ETSI TR 119 476-1 V1.3.1 (2025-08)

TECHNICAL REPORT

Electronic Signatures and Trust Infrastructures (ESI); Selective disclosure and zero-knowledge proofs applied to Electronic Attestation of Attributes; Part 1: Feasibility study

Reference
RTR/ESI-0019476-1v131

Keywords
identity, trust services

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

BLUETOOTH® is a trademark registered and owned by Bluetooth SIG, Inc.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Electronic Signatures and Trust Infrastructures (ESI).

The present document is part 1 of a multi-part deliverable covering policy and security requirements for selective disclosure and zero-knowledge proofs applied to Electronic Attestation of Attributes, as identified below:

Part 1: "Feasibility study";

Part 2: "Implementation in EUDI Wallet";

Part 3: "EUDI Wallet Unit Attestation".

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The eIDAS2 regulation [i.103] defines regulatory requirements on selective disclosure and unlinkability for the EUDI Wallet. While the corresponding Architecture and Reference Framework (ARF) is not legally binding, it serves as a reference for the harmonized implementation of the EUDI Wallet. As such, it provides guidance, among other aspects, how to implement selective disclosure.

In contrast, the present document has a broader scope regarding data minimisation. It provides a general yet comprehensive analysis of signature schemes, formats and protocols with different degrees of maturity that cater for selective disclosure, unlinkability, and predicate proofs. More specifically, the present document includes an analysis of how certain data minimisation techniques can be applied to eIDAS2 and the EUDI Wallet.

The term selective disclosure means that a user should be capable of presenting a subset of attributes from at least one, but potentially multiple, (Qualified) Electronic Attestations of Attributes ((Q)EAAs). For example, a user should be able to only present their birth date from an attestation resembling an ID-card.

The term unlinkability means that different parties should not be able to connect the user's selectively disclosed attributes beyond what is disclosed. There are different categories and degrees of unlinkability, and the present document focuses both on verifier unlinkability and full unlinkability. Verifier unlinkable means that one or more verifiers cannot collude to determine if the selectively disclosed attributes describe the same identity subject, whilst fully unlinkable means that no party can collude to determine if the selectively disclosed attributes describe the same identity subject.

Predicate proofs are verifiable computations on information included in attestations, where only the result of the computation and none of the underlying inputs is shared. This includes Boolean assertions (true or false) about properties of attributes in a (Q)EAA without disclosing the attribute value itself. For example, a user could derive a proof that they are above the age of 20 from their birth date and show only this proof as opposed to the birthdate itself. On the other hand, the predicate could also be the sum of two attributes included in a (Q)EAA or a verifiable pseudonym computed from cryptographic metadata associated with the (Q)EAA and information provided by the relying party. Predicate proofs are often employed in Zero-Knowledge Proof (ZKP) systems aimed at limiting information disclosure or fine-tuning it (e.g. in the case of verifiable pseudonyms [i.245]).

In general, data minimisation in verifiable presentations is not easy to achieve because the required degree of verifiability relies on digital signatures, which become invalid upon any modification ("blackening") of information in a (Q)EAA. Selective disclosure can be relatively easy to achieve, whereas full unlinkability and predicate proofs require advanced to very advanced cryptographic constructions.

The selective disclosure signature schemes described in the present document are divided in the following categories:

- **Atomic (Q)EAA schemes**. An atomic electronic attribute attestation is a (Q)EAA with a single attribute claim, which can be issued by a (Q)TSP upon request or as part of a batch to an EUDI Wallet. The atomic (Q)EAAs can be selected by the user and be included in a verifiable presentation to a verifier.

- **Salted attribute hashes**. The general concept of this category is to combine each attribute with a salt, hash the combined values, and insert the resulting salted attribute hashes in a list that is signed. The user presents a selection of attributes to the verifier, which can validate them against the list of salted attribute hashes. The following schemes, based on salted attribute hashes, are described: HashWires, Authentic Chained Data Containers (ACDC), and Gordian Envelopes.

- **Multi-message signature schemes**. The category of multi-message signature schemes has the capability of proving the knowledge of a signature while selectively disclosing any subset of the signed messages. By definition, multi-message signature schemes also cater for full unlinkability. The following schemes in this category are described: BBS/BBS+/BBS#, Camenisch-Lysyanskaya (CL) signatures, Mercurial signatures, and Pointcheval-Sanders Multi-Signatures (PS-MS). ISO/IEC have standardized parts of BBS and PS-MS in ISO/IEC 20008 [i.184], and have taken the initiative to standardize BBS+ and PS-MS in ISO/IEC 24843 [i.185] and ISO/IEC CD 27565 [i.191]. Furthermore, there are cryptographic research projects, such as MoniPoly, where undisclosed attributes have no impact on the proof size.

- **Proofs for arithmetic circuits (programmable/general-purpose ZKPs)**. This category of ZKP protocols enable the user to prove to the verifier that a certain statement is true, without revealing any additional information beyond the truth of the statement itself. The discussion of proofs for arithmetic circuits is currently focused on zk-SNARKs because this type of programmable ZKPs has matured rapidly in recent years, arguably to their broad adoption in cryptocurrencies, decentralised finance and industry blockchain projects.

The present document also includes descriptions of (Q)EAA formats that can be used with selective disclosure. The (Q)EAA formats are divided in the following categories:

- **Atomic (Q)EAA formats**. These (Q)EAA formats are based on the category of atomic (Q)EAA formats. The following (Q)EAA formats in this category are described: PKIX X.509 attribute certificate with atomic attribute and W3C® Verifiable Credential with atomic attribute.

- **(Q)EAAs with salted attribute hashes**. This category of (Q)EAA formats is based on the concept of salted attribute hashes. These (Q)EAA formats specify in detail how the attributes are combined with the random salts and hashed, inserted in a list, which is signed. The following (Q)EAA formats of this category are described: IETF SD-JWT and ISO/IEC 18013-5 [i.181] Mobile Security Object (MSO).

- **Multi-message signature (Q)EAA formats**. This category of (Q)EAA formats is based on multi-message signature schemes. Mainly W3C and Hyperledger have specified such formats to be used for privacy preserving features. The following (Q)EAA formats in this category are described: W3C VC Data Model with ZKP, W3C VC Data Integrity with BBS Cryptosuite, and Hyperledger AnonCreds (format).

- **JSON container formats**. This category of generic JSON container formats allows for combining and presenting a mix of selective disclosure signature schemes. The following JSON container formats are described: IETF JSON WebProof (JWP), JSON Web Zero Knowledge (JWZ), W3C Data Integrity ECDSA Cryptosuites v1.0, and W3C JSON Web Proofs For Binary Merkle Trees.

Furthermore, the present document describes systems and protocols with selective disclosure capabilities. The systems and protocols are divided in the following categories:

- **Atomic attribute (Q)EAA presentation protocols**. This category of protocols is designed to present the atomic attribute (Q)EAA formats. The atomic attribute (Q)EAAs may be issued on demand to the user, upon request by a verifier. The following protocols in this category are described: PKIX X.509 attribute certificates with single attributes and VC-FIDO for atomic (Q)EAAs.

- **Salted attribute hashes-based protocols**. These solutions and protocols are designed to present selectively disclosed attributes based on salted attribute hashes. The OpenAttestation solution of Singapore's Smart Nation is described in the present document. Furthermore, ISO mDL MSOs can be shared over the proximity protocols described in ISO/IEC 18013-5 [i.181] or over the Internet by using ISO/IEC CD 23220-4 [i.187]. The SD-JWTs can be presented with different protocols, such as OID4VP (OpenID for Verifiable Presentations), ISO/IEC CD 18013-7 [i.182] or ISO/IEC CD 23220-4 [i.187].

- **Multi-message signature protocols and solutions**. This category of protocols is based on multi-message signature schemes, such as BBS+ and CL-signatures, and are used to present selected attributes of the (Q)EAAs. The following protocols and solutions in this category are described: Hyperledger AnonCreds (protocols) and Direct Anonymous Attestation (DAA) used with Trusted Platform Modules (TPMs); the TPMs have been deployed in personal computers at a large scale.

- **Solutions based on proofs for arithmetic circuits (programmable/general-purpose ZKPs)**. The solutions that are based on proofs for arithmetic circuits intend to use ZKP schemes such as zero-knowledge non-interactive arguments of knowledge (zk-NARKs) and succinct forms of these (zk-SNARKs) (to facilitate data-minimising verifiable presentations based on existing digital identity infrastructures). In particular, they can provide selective disclosure, unlinkability, and arbitrary predicate proofs. As proof generation involves substantial overhead, these schemes are often combined with new formats for attestations that make ZKP operations relatively efficient. As examples, constructions suggested by [i.14] and the Iden3 protocols are covered. On the other hand, the recent progress in designing and implementing efficient programmable ZKPs now also allows for compatibility with legacy formats. The following projects are covered in the present document: Cinderella (zk-SNARKs used with X.509 certificates), zk-creds (zk-SNARKs used with ICAO passports), FIDO-AC (zk-SNARKs used with the FIDO protocol and ICAO passports), Crescent (zk-SNARKs used with X.509 certificates and JWTs), and anonymous credentials from ECDSA (zk-NARKs used with ISO mdoc).

- **Anonymous attribute-based credentials systems**. These solutions are implementations of existing multi-message signature schemes such as BBS+ or CL-signatures, with the purpose to present anonymous credentials ((Q)EAAs) to a verifier. The following solutions in this category are described: Idemix (Identity Mixer), U-Prove, ISO/IEC 18370 [i.183] (blind digital signatures), and Keyed-Verification Anonymous Credentials (KVAC).

NOTE: In the academic literature, the terms "anonymous credentials" and "attribute-based credentials" are often used synonymously. Both refer to the construction of digital certificates ((Q)EAA) and corresponding presentation protocols that disclose only the minimum amount of information requested by the relying party while still giving assurances of all the expected validity and consistency properties.

- **ISO mobile driving license (ISO mDL)**. The ISO mDL standard [i.181]specifies various flows for selective disclosure of attributes. In the present document, the following ISO mDL flows are described: ISO/IEC 18013-5 [i.181] (device retrieval flow), ISO/IEC 18013-5 [i.181] (server retrieval flows), ISO/IEC 18013-7 [i.182] (unattended flow) and ISO/IEC 23220-4 [i.187] (operational protocols). mDL describes the specific driver's license document or application, whereas mdoc is used to describe the general mechanism for documents or applications residing on a mobile device. For the rest of the present document, mdoc is used to refer to the general mechanism or format for digital identity documents, whereas mDL can be seen as a special case of an mdoc.

The ARF proposes two protection mechanisms for the PID, which support selective disclosure but not unlinkability (unless batch issued):

- ISO/IEC 18013-5 [i.181] (ISO mDL). The mdoc contains 2 general structures, the issuer signed part called, especially the MSO and when presenting a device signed part. During issuance, the issuer provides all attributes of a user in an issuer signed part next to the MSOs whilst the MSO contains the corresponding salted attribute hashes and other information that is signed over by the issuer like a validity period. During presentation, the device signed structure is used to present the attributes that are released.

- IETF SD-JWT in conjunction with IETF SD-JWT VC. The JWT contains the user attributes, whilst the SD-JWT contains the corresponding salted attribute hashes.

The present document includes an extensive analysis of mdoc and SD-JWT and how the formats comply with the eIDAS2 requirements on selective disclosure and unlinkability.

The mdoc and the SD-JWT formats, and related presentation protocols, cater for selective disclosure based on the concept of salted attribute hashes. Furthermore, the mdoc and SD-JWT formats support SOG-IS approved cryptographic algorithms and can also be used with quantum-safe cryptography for future use. The conclusion is thus that mdoc and SD-JWT meet the eIDAS2 regulatory and technical requirements on selective disclosure.

As stated, mdoc and SD-JWT are not fully unlinkable, although they can provide verifier unlinkability with certain operational measures. In order to achieve verifier unlinkability, batches of MSOs or SD-JWTs need to be issued to each EUDI Wallet. In this case, the random salts in the MSO and SD-JWT should be unique, meaning that refreshed MSOs and SD-JWTs are presented to a relying party. Furthermore, the user public keys used for holder binding, if presented (in non-proximity scenarios), need to be unique, too.

There are many similarities between the ISO mDL issuers and the eIDAS2 compliant PID Providers (PIDPs) or QTSPs. The PIDPs/QTSPs can issue PIDs/(Q)EAAs to EUDI Wallets as follows to cater for selective disclosure:

- The PIDP/QTSP issues mdoc and/or JWT as PID/(Q)EAAs to the EUDI Wallet.

- The PIDP/QTSP issues MSOs and/or SD-JWTs batchwise to the EUDI Wallet. The MSOs are associated with the mdoc, and the SD-JWTs with the JWT. Random salts are used for the salted attribute hashes in each MSO or SD-JWT. This will cater for verifier unlinkability when the MSOs or SD-JWTs are presented to and validated by a relying party.

- The EUDI Wallet selectively discloses certain attribute(s) of an mdoc or JWT. One MSO or SD-JWT is selected from the batch in the EUDI Wallet, and is associated with the disclosed attribute(s).

- The relying party can use the eIDAS2 trust list (which is equivalent to an ISO mDL VICAL) to retrieve the QTSP/PIDP trust anchor (which is equivalent to the IACA trust anchor). The relying party validates the MSOs or SD-JWTs signatures by using the QTSP/PIDP trust anchor. The relying party also verifies that the presented selected attribute hash is present in the MSO or SD-JWT.

These recommendations could be considered for the upcoming ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97] that will standardize the issuance policies and profiles of (Q)EAAs.

Multi-message signature schemes such as BBS+, BBS#, Camenisch-Lysyanskaya (CL) signatures, Mercurial signatures, and Pointcheval-Sanders Multi-Signatures (PS-MS) cater for full unlinkability, although they are not yet fully standardized. Hence, ISO/IEC 24843 [i.185] intends to standardize BBS+ and PS-MS with blinded signatures, which may allow for a future standard that could be used in compliance with the EUDI Wallet requirements on selective disclosure and unlinkability in eIDAS2. The BBS# scheme can also be implemented for ISO MSO, W3C VCDM and IETF SD-JWT, which caters for full unlinkability for these formats.

There are also systems based on programmable ZKPs in the form of zk-SNARKs, such as Cinderella, zk-creds, and zk-mdoc, that can achieve both selective disclosure and unlinkability with existing digital identity infrastructures such as X.509 certificates, ISO mDL, or ICAO passports. Such systems can generate pseudo-certificates that share selected attributes from the (Q)EAAs and attest holder binding and non-revocation without exposing linkable cryptographic identifiers, as well as implement arbitrary predicates. Anonymous credentials based on programmable ZKPs can, therefore, in particular be made compatible with deployed secure hardware and are easily extendable. However, these projects are still in the research phase and face a high degree of complexity. On the other hand, they can be considered future-proof as some forms (e.g. the NARKs used in [i.113]) are plausibly post-quantum secure owing to their sole reliance on cryptographic hash functions, and as opposed to established and more efficient multi-message signature schemes, they can flexibly adapt to new (e.g. post-quantum secure) signature schemes and novel (Q)EAA formats. Hence, they may be considered for the EUDI Wallets and eIDAS2 relying parties.

Furthermore, there are recommendations on how to store such (Q)EAA formats in the EUDI Wallet, and how to present selectively disclosed attributes to eIDAS2 relying parties. These recommendations can be considered for the upcoming ETSI TS 119 462 [i.95] on EUDI Wallet interfaces.

The present document also analyses the privacy aspects of revocation schemes and validity status checks. In order to achieve privacy preserving features for revocation and validity status checks it is recommended to use OCSP in Must-Staple mode, implement Revocation Lists or validity Status Lists with additional privacy techniques such as Private Information Retrieval or Private Set Intersection, and use cryptographic accumulators where possible given the associated complexity. If programmable ZKP schemes (such as zk-(S)NARKs) are combined with existing credentials (such as X.509) and revocation or status lists, the status validity checks are performed at the EUDI Wallet, and only the relevant information (revocation state) without any linkable cryptographic identifiers is disclosed to the verifier.

The present document also includes an analysis of attacks facilitated by a quantum computer on cryptographic schemes with selective disclosure capabilities. More specifically, the salted attribute hashes-based formats, such as mdoc and SD-JWT, can be signed with quantum-safe cryptographic algorithms. Also the atomic (Q)EAA formats can be secured with post-quantum safe signatures. The multi-message signature schemes, such as BBS+ and CL-signatures, have the following characteristics in a post-quantum world: an attacker can use a quantum computer to forge proofs and signatures (i.e. to violate soundness guarantees), but an attacker will not be able to break data minimisation, meaning that undisclosed attributes are safe in a post-quantum world, as are undisclosed signature values etc. This unconditional data minimisation guarantee is also provided by the programmable ZKPs. However, it depends on the design of the arithmetic circuit proof if soundness holds in the presence of a quantum computer. For example, the hash-based NARKs used in [i.113] are considered post-quantum secure in this regard, while others (like the ones used in [i.65], [i.182], [i.269]) are not as they rely on elliptic curves. There are also research projects on lattice-based anonymous credentials schemes, which are plausibly post-quantum safe.

Furthermore, there is an annex (annex F) with business models, which discusses how a QTSP can be able to invoice a Relying Party, even if the EUDI Wallet has shared the (Q)EAA/PID anonymously with the Relying Party. ETSI TR 119 479-2 [i.92] and anonymous usage data aggregation propose solutions to this business model.

Finally, there is an annex (annex C) with research projects about innovative ZKP schemes. One such approach is to design cryptographic ZKP schemes based on quantum physics. Quantum Key Distribution (QKD), quantum physics applied to the graph 3-colouring ZKP scheme, and ZKPs using the quantum Internet (based on Schnorr's algorithm) are described in annex C. The ZKP schemes based on quantum physics are still in the research phase, but may be considered for the future. There are also cryptographic research initiatives on post-quantum safe (lattice-based) anonymous credentials, which cater for privacy-preserving signature schemes. The most recent research in this field is related to efficient anonymous credentials that are post-quantum safe, yet with small signature sizes.

While the present document aims to comprehensively explore cryptographic techniques to ensure selective disclosure, unlinkability, and predicate proofs, it does not discuss in depth the data-minimising capabilities of authenticated channels facilitated by secure hardware (this approach is prominently used in the German eID) and the corresponding challenges, such as the tradeoff between unique device identifiers and shared risks. Similarly, while selective disclosure, unlinkability, and predicate proofs can be easily implemented via the use of remote attestation in trusted execution environments on the holder or relying party side (see, e.g. [i.121]), corresponding architectures and their risks are not covered regarding trust in manufacturers and side channel attacks. Furthermore, while data minimization in the cryptographic parts of verifiable presentations is necessary to achieve a high degree of data protection and avoid over-identification, it is not sufficient. Yet, the present document does not cover further linkable data, e.g. on the networking layer (IP addresses) or how to determine which identity attributes a relying party should be allowed to request.

Introduction

A historical perspective

To facilitate an understanding of the concepts in the present document, the present clause begins with a brief account of the history of selective disclosure and Zero-Knowledge Proofs (ZKPs), the problems they were introduced to address, their applications, and their potential uses in electronic attestations of attributes. The present document also discusses related concepts where required.

A pioneer in the field of privacy was the American cryptographer David Chaum who published the scientific paper Blind Signatures for Untraceable Payments [i.60] in 1982, which described anonymized digital money (DigiCash) for the first time. The concept of Blind Signatures was designed to ensure complete privacy of users who wanted to conduct online transactions.

Cryptographic schemes for selective disclosure, unlinkability via blind signatures or ZKPs, and predicate proofs (in particular, range proofs) have been researched and developed since the 1980s. The first ZKP scheme was published in a paper 1985 by the researchers Shafi Goldwasser, Silvio Micali, and Charles Rakoff [i.120]. The abstract of this paper defines ZKP: "*Zero-Knowledge Proofs are defined as those proofs that convey no additional knowledge other than the correctness of the proposition to the question*".

The present document on selective disclosure can be linked to the broader work on signatures that allows to reveal parts of or statements derived from signed documents, while maintaining verifiability.

It is important to note that ZKP is not a selective disclosure scheme in and of itself, but rather a property of a proof system. Thus, ZKPs are not limited to selective disclosure or proofs of knowledge of a signature in the context of electronic attestations of attributes. On the contrary, Brassard et al. demonstrated in their paper "Minimum disclosure proofs of knowledge" [i.34] that everything that has a proof also has a ZKP version of that proof.

Put differently, every proof of a statement about a signed object - like a digital certificate - has a ZKP version of that proof. In particular, selective disclosure can be achieved with a ZKP. But it is incorrect to state that every selective disclosure scheme is done using ZKP, or that every ZKP is used for selective disclosure. ZKPs matter because usually, in digital identification, holders share substantially more information than the verifier asks for, e.g. superfluous identity attributes, unique cryptographic information (signatures, public keys, revocation IDs, expiration dates), following the verifier's implicit request. Using a ZKP, the holder only proves what the verifier wants to know (precisely the required identity attributes, i.e. selective disclosure; that the attributes are signed by the issuer without revealing the linkable digital signature (unlinkability), that an attribute has a required property without sharing it (predicates such as range proofs). As such, ZKPs can be considered as the "gold standard" for meeting the GDPR's data minimisation principle, as they reflect a mathematical notion of data minimisation relative to what the relying party needs to know explicitly.

Electronic attestations of attributes represent a context in which several features, such as selective disclosure or proofs about knowledge of states like a valid signature value, have been implemented with the ZKP property. Among the earliest work here was done by Feige, Fiat, and Shamir (1987) [i.229] who demonstrated how ZKP can be used in identification schemes by a user demonstrating knowledge as opposed to directly demonstrating the validity of assertions. Since then, ZKPs have been widely deployed in many of the privacy focused selective disclosure capable electronic attestation of attribute solutions.

In 2002, Steinfeld, Bull, and Zheng published their paper "Content Extraction Signatures" (CES) [i.239]. In it, the authors present a way to perform the delete operation without knowledge of the signer's private key. The authors argue that this would allow a user "to disclose only certain parts of a document" as opposed to "forcing the document holder to disclose all of its contents to a third party for the signature to be verifiable". The authors then go on to present the idea of context extraction, i.e. "the extraction of certain selected portions of a signed document" in cases where a user "does not wish to pass on the whole document to a third (verifying) party". Their method is based on signing digests of data subsets. Relatedly, Johnson et al. (2002) [i.275] presented their work on redactable signatures, which are conceptually very similar to CES. In fact, the proposed schemes in the papers overlap, together detailing four different schemes for CES. Two of these rely on commitment vectors, and two on the homomorphic properties and batching of RSA respectively.

Brands (2002) directly applies these concepts to electronic attestations of attributes. In his 2002 paper "A Technical Overview of Digital Credentials" [i.32] Brands discusses the "selective disclosure properties of data fields" in digital credentials. In that paper, Brands presents the idea to "hash attributes [...] using a collision-intractable hash function; to disclose these attributes, Alice discloses the preimages of the corresponding [attributes]". Interestingly, Brands proposed design also relies on a proof of knowledge of the digital signature, which is among the first references to the use of ZKP for enhancing privacy when presenting electronic attestations of attributes. Brands' paper is also among the earliest work on the use of predicates in electronic attestations of attributes. In essence, Brands' work was based on commitment vectors and the algebraic manipulations (e.g. addition and multiplication) of these commitments, allowing proofs containing logical AND, OR, and NOT operations between attributes and for a single attribute.

The above mentioned work laid the groundwork for the concept of selective disclosure and unlinkability. Ongoing work presented workarounds to discovered vulnerabilities in some of the proposed schemes, and introduced more advanced features that further improved privacy e.g. by enabling multi-show unlinkable selective disclosures (defined in clause 3.1 and for additional details see "Anonymous Credentials" [i.43] by Camenisch and Lysyanskaya in 2003). Notable early examples of implementations of this work focused on enhanced privacy include AnonCreds and Idemix (both based on Camenisch-Lysyanskaya signatures as detailed herein under clause 4), as well as U-Prove (based on Brands' work). A more recent example of a multi-message signature scheme capable of selective disclosure is the BBS+ signature scheme (detailed in clause 4.4 and is based on group signatures and the work of Boneh, Boyen, and Shacham, 2004 [i.33]). However, as noted in Camenisch et al. (2013) [i.43], real-world deployments of cryptographic primitives, schemes and protocols in electronic attestations of attributes have been slow due to them being hard to understand and "very difficult to use" as they often require advanced cryptography and the combination of several protocols to achieve the desired privacy goals. In a survey, Asghar (2011) [i.11] lists some of these often employed mechanisms, including blind signatures (Chaum, 1983 [i.60]), ZKPs (Goldwasser, Micali, and Rakoff, 1985 [i.120]), group signatures, commitment schemes (formalized in Brassard, Chaum, and Crépeau, 1988 [i.34]), and multi-message signing; which often need to be employed in tandem to reach privacy goals important for selective disclosure including multi-show unlinkability, blinding, and the ability to present a subset of the signed attestation.

In contrast to the focus on increasing privacy, others sought more performant schemes with lower but still acceptable levels of privacy. A notable example here is the early work of Bull, Stanski, and Squire (2003) [i.37], who presented a way to "enable selective disclosure of verifiable content" using a randomized salt to blind the attribute disclosures, using an identifier for each disclosable attribute, and the principle of signing the hash digests of attributes. To disclose the desired attributes, a user would simply present a subset of the attestation to the verifier, together with the attributes and salts to disclose. Variations of this salted hash digest based approach are used both in ISO/IEC 18013-5 [i.181] and in the IETF SD-JWT specification [i.155]. Note that these techniques do not achieve the same levels of privacy as their more advanced counterparts (e.g. U-Prove, AnonCreds, Idemix) because they lack unlinkability and support for selected predicates, but they are easier to use and more performant.

The academic research of cryptographic schemes for selective disclosure, unlinkability, and predicates have continued from the mid 2010s until present day: Bulletproofs [i.38] and Pointcheval-Sanders Multi-Signatures [i.223] provide range proofs over committed values, whilst zk-(S)NARKs (clause 4.5.2) are advanced protocols for fully programmable ZKPs. More information about those cryptographic schemes is described in clause 4 of the present document.

The Internet standardization organizations Hyperledger, IETF and W3C® have followed the academic cryptographic research by creating Internet standards for selective disclosure, unlinkability, and predicates. Hyperledger has specified AnonCreds [i.131]. IETF has specified the BBS Signature Scheme [i.177], JSON WebProofs [i.152], PKIX attribute certificates [i.158], and SD-JWT [i.155]. W3C has specified the BBS Cryptosuite and the Verifiable Credentials Data Model describes ZKPs [i.264]. Furthermore, ISO/IEC 18013-5 [i.181] specifies selective disclosure for the mobile driving license by introducing the Mobile Security Object (MSO) for the device retrieval use case. Clauses 5 and 6 in the present document describe the mentioned standards in more detail.

Overview and use cases

An overview of various use cases is provided in Figure 1 to illustrate the concepts of selective disclosure, unlinkability, and predicate proofs.

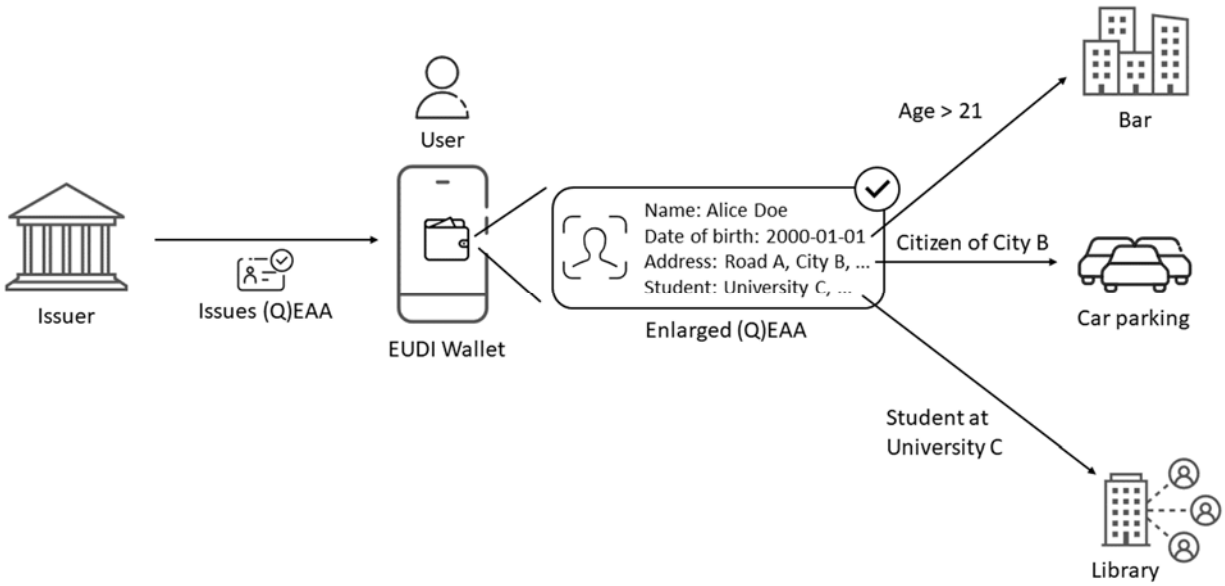


Figure 1: Overview of selective disclosure

First, an issuer creates and issues a (Qualified) Electronic Attestation of Attribute (EAA) (denoted as (Q)EAA) to a user, whereupon the (Q)EAA is stored in the user's EUDI Wallet.

EXAMPLE 1: The (Q)EAA contains the attributes name (first name and last name), date of birth, address (street, city, zip code, etc.), and student information (university, exams, course, etc.).

NOTE 1: The issuer may also issue a Person Identification Data (PID) with the same attributes, but a (Q)EAA is used for readability in this particular example.

The (Q)EAA that is stored in the user's EUDI Wallet is also associated with cryptographic keys that are necessary for the cryptographic scheme's selective disclosure capabilities. In order to access the private keys, the user needs to authenticate with PIN-code or biometrics. Clauses 6.3 and 6.5.3 in the Architecture and Reference Framework (ARF) [i.71] provide more information on the EUDI Wallet security architecture and the supported cryptographic keys management systems.

Now, the user can use its EUDI Wallet to present selected attributes of the (Q)EAA to various relying parties. A user may present multiple attributes to each verifier and is not limited to present only a single attribute claim. The user may also be able to create a presentation that includes claims from at least two (Q)EAAs even if these are issued by different issuers (herein referred to as combined presentation).

When borrowing a book at the university library, the user may only present that she is taking Course D at University C to prove that she is eligible to borrow the course literature.

NOTE 2: This is an example of selective disclosure of a single attribute. The EUDI Wallet contains detailed student information (university, degrees, courses, etc.), but the EUDI Wallet only presents the single claim that user studies at University C.

When going to a bar, the user may even further minimise data disclosure by only presenting a proof that she is over the age of 21 years instead of selectively revealing the date of birth.

NOTE 3: This is an example of a predicate proof. The EUDI Wallet contains the user's actual date of birth (2000-01-01), but the EUDI Wallet could only present a proof that $21 \leq age$ (more realistically, the difference between the UNIX timestamp corresponding to the date of birth is more than the number of seconds in 18 years smaller than today's date, also as a UNIX timestamp).

NOTE 4: This example can also be achieved using selective disclosure of a single attribute. The EUDI Wallet could contain an attestation with the key value pair `"age_over_21": "True"`. This is much simpler from a technical perspective but less flexible.

When parking the car in City B, the user may present a proof that she is a citizen of City B in order to get a discount when paying for the parking ticket. Unlinkability here helps prevent behavioural profiling and the user presents evidence that the (Q)EAA is valid, without disclosing it.

NOTE 5: This can be achieved using a ZKP. The EUDI Wallet only presents a ZKP of knowledge of a valid signature without disclosing said signature (the signature is linkable data). Analogously, a proof of holder binding without revealing the holder's linkable public key may be needed, which can also be given with a ZKP. Further assertions (which can all be considered predicate proofs) include a proof of non-expiration, without revealing the underlying linkable expiration date and revocation ID.

The concept of verifier unlinkability relates to the amount of additional information that colluding verifiers can discover about the user. A high degree of unlinkability means that the colluding verifiers learn little in addition to what the verifier explicitly requested to be disclosed (for instance, a verifier should typically not care about the linkable signature value but only that the (Q)EAA is valid). Similarly, a single verifier cannot collect multiple selectively disclosed attributes and link them to the same user beyond what is possible solely based on the disclosed attribute values. This requires removing correlatable data (such as the signature) in the presentation to each verifier.

EXAMPLE 2: If presentations are unlinkable, then the bar (who knows that the user is over 21 years) cannot cooperate with the car parking (who knows that the user lives in City B) to link the user's age to the citizenship.

EXAMPLE 3: If presentations are unlinkable, then the user may visit the university library multiple times and present proofs of different courses (Course D, Course E, etc.) over time. The university library cannot link these presentations to the same user beyond what they already know about combinations of courses among students.

The concept of issuer unlinkability means that the issuer cannot collude with one or more verifiers to discover where the user is using the issued (Q)EAA. Most ZKP-based systems discussed in the present document provide full unlinkability, i.e. verifier unlinkability and issuer unlinkability. In contrast, batch issuance can only provide verifier unlinkability.

Descriptions of selective disclosure and unlinkability

The preceding text introduced the terms 'selective disclosure' and 'unlinkability' without providing precise definitions. These terms often have varied interpretations, and these interpretations significantly influence the choice of an appropriate privacy-enhancing technology such as ZKPs. Despite their apparent similarity, selective disclosure and unlinkability are distinct concepts, and their relationship to privacy is complex:

- Selective disclosure involves revealing specific attributes, or claims about these attributes, from a larger dataset. Selective disclosure, on its own, does not guarantee the highest privacy guarantees but may be a key part of a privacy preserving solution. In particular, selective disclosure used without associating it to unlinkability may provide a false sense of privacy to the user where the advantages of selective disclosure might be inverted through correlation attacks.

- Unlinkability relates to the difficulty or cost of linking multiple electronic attestation of attribute presentations. Unlinkability does not inherently ensure privacy but can be a vital element thereof.

Furthermore, the two concepts (selective disclosure and unlinkability) are not binary; they exist on a spectrum or scale, where various degrees or levels exist. And different privacy-enhancing technologies are required at different degrees or levels. For selective disclosure, it is possible to understand these levels through a set of requirements:

1) The ability to selectively disclose a minimum of one attribute from a single (Q)EAA.

2) The ability to selectively disclose a minimum of two attributes from at least two distinct (Q)EAAs, with at least one attribute from each (Q)EAA. This ability is sometimes referred to as 'combined presentation'.

3) The user can disclose statements about one or several attributes rather than the attributes themselves. This ability is sometimes referred to as support for predicates or predicate proofs. These attributes may even be associated with different (Q)EAA.

Note that the attributes disclosed do not necessarily have to describe the identity subject. For instance, a disclosure can disclose the EAA type to reveal only that the user has a certain attestation (e.g. passport) without revealing any attribute about the identity subject. On the other hand, predicates can also involve the cryptographic meta-data, e.g. determine membership of the issuer's public key in a list specified by the relying party or derive a pseudonym from the holder's public key. Furthermore, the above three requirements relate to other requirements to ensure important capabilities like holder binding (e.g. the verifier has to be assured that the: a) presented attributes cannot be combined in ways that make them appear to be part of another (Q)EAA than they originally were, b) presented attributes describe the same identity subject, and c) identity subject is the same entity as is presenting the attributes) and unlinkability. In this sense, a proof of knowledge of a valid (Q)EAA that asserts all of the common checks and selectively discloses attributes can also be considered a predicate proof. As a consequence, predicate proofs can be seen as a generalisation of selective disclosure.

Relatedly, unlinkability can be understood through a set of requirements. The general requirement relates to the ability to determine whether at least two (Q)EAA presentations describe the same identity subject. More precisely, presentations (p1, p2) are unlinkable if a set of entities cannot decide, with a non-negligible probability better than pure guessing based on the presentations and attributes received, whether the two presentations describe the same identity subject. The following cases are possible as unlinkability criteria:

1) The set is a single verifier who seeks to learn whether the attributes describe the same identity subject.

2) The set consists of at least two colluding verifiers who share the respective presentations they received in order to determine whether the attributes describe the same identity subject.

3) The set consists of signers (issuers) and verifiers, who share information to determine if the attributes describe the same identity subject.

4) The set consists of signers, verifiers, or any other party, who share information to determine if the attributes describe the same identity subject.

Throughout the rest of the present document, criteria 1 and 2 above will be combined and referred to as verifier unlinkable, whilst criteria 3 and 4 will be combined and referred to as fully unlinkable.

Neither the requirements for selective disclosure nor unlinkability are exhaustive; they are meant to clarify the non-binary nature of these concepts. What matters is the extent to which the technical solutions and formats presented in the present document can fulfil some or all of the above requirements.

Furthermore, the relationship between selective disclosure, unlinkability, and privacy is not straight forward. It is incorrect to assume that a (Q)EAA capable of selective disclosure also has to be privacy preserving. Similarly, it is not necessarily so that a (Q)EAA with unlinkability features guarantees that the privacy is preserved. If the verifier requires certain information for business or regulatory reasons, privacy may not be possible but minimizing the amount of information conveyed by the user may still be desirable to technically maximize privacy within the boundaries of the use case. Consider the following examples:

EXAMPLE 4: A user discloses that they are below the age of 65, and that they have a tertiary education. The verifier is able to determine that these two attributes describe the same identity subject. The user's privacy is still protected because the verifier does not have enough information to learn the user's identity (roughly 32 % of citizens aged 25 - 74 years in the EU have a tertiary education).

EXAMPLE 5: A user discloses that their first name is Peter, that they live in Sweden, and that they are below the age of 21 in three separate presentations. Each attribute roughly represents 10 million possible entities. If any party is able to learn that these three attributes represent the same identity subject (i.e. is able to link them) they can narrow down the candidates to about 300. Unlinkability here is crucial to prevent a subset of attributes from becoming personally identifying.

EXAMPLE 6: A doctor books a physical meeting with a patient, and when the patient arrives, they selectively disclose only the meeting time and meeting location. The user did not reveal any identifying or linkable information. The verifier can still easily identify the patient through the context of the presentation.

EXAMPLE 7: The verifier has access to user data sufficient for a behavioural profile in another context, e.g. browsing data over time. The user then presents unrelated data to the verifier that allows the verifier to quantify similarities in sequential data and thus identify the user.

These examples serve as a transition to a more insightful approach to understanding privacy beyond the capacity for selective disclosure or unlinkability. It delves into quantifying the extent to which each presentation diminishes the uncertainty surrounding the identity subject. Both selective disclosure and unlinkability can contribute to privacy, but their effectiveness depends on the extent of uncertainty reduction, which often is influenced by other factors. And it is unlikely that technical solutions alone can eliminate all such factors, especially considering the rapid evolution of behavioural profiling and identification techniques.

As established, user control and privacy are influenced by factors extending beyond the technical aspects of selective disclosure, unlinkability, or even predicates. Nonetheless, it is the legal text that guides the choice of privacy-preserving techniques and when and how selective disclosure and unlinkability will be supported.

Legal definitions in eIDAS2 about selective disclosure, unlinkability, and ZKP

The provisional agreement on the amending Regulation (EU) No 910/2014 (hereafter called eIDAS2) [i.103] mandates support for privacy in Recital 15 and article 5a.4(a) and provides the following definition of selective disclosure in recital 59:

"Selective disclosure is a concept empowering the owner of data to disclose only certain parts of a larger data set, in order for the receiving entity to obtain only such information as is necessary for the provision of a service requested by a user. The European Digital Identity Wallet should technically enable the selective disclosure of attributes to relying parties. It should be technically possible for the user to selectively disclose attributes, including from multiple, distinct electronic attestations, and to combine and present them seamlessly to relying parties. This feature should become a basic design feature of European Digital Identity Wallets, thereby reinforcing convenience and the protection of personal data, including data minimisation."

The definition in eIDAS2 recital 59 clarifies that disclosed information may come from multiple distinct electronic attestations of attributes, similar to the second selective disclosure requirement for combined presentations. This scenario requires additional considerations related to holder binding and proper pairing of attributes as compared to single attestation disclosures.

Moreover, the definition specifies the ability to disclose a subset of a larger data set as disclosing only such information that is necessary for the provision of a service. It is possible to interpret this clarification as a requirement that users are able to assert and prove statements about their attributes without disclosing the actual attribute data. This interpretation is aligned with Recital 14 [i.103] that states that "*cryptographic methods should allow a relying party to validate that a given statement based on the person's identification data and attestation of attributes is true, without revealing any data this statement is based on*". If this interpretation holds true, it aligns with the concept of the third selective disclosure requirement concerning predicate support. One method for implementing predicate support is through the utilization of ZKP-capable attestations, although alternatives exist. ZKPs could also be used to prove the equality (a predicate) of highly linkable identity attributes (e.g. name and date of birth or a cryptographic public key) from different attestations without revealing the identity attributes, thus increasing holder binding guarantees without reducing privacy.

Relatedly, eIDAS2 article 5a.16 lists the requirements related to unlinkability as follows:

"The technical framework of the European Digital Identity Wallet shall:

(a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorised by the user;

(b) enable privacy preserving techniques which ensure unlinkability, where the attestation of attributes does not require the identification of the user."

This article elucidates the specific entities whose linking efforts the solution aims to make more difficult. Note how (a) encompasses all parties, including issuers, verifiers, and third parties. Note also how and when the requirement in (b) mandates privacy preserving techniques to ensure unlinkability. Together, (a) and (b) seemingly correspond to either the third or fourth unlinkability requirement, which mandates unlinkability even in cases of collusion between an issuer (who signs the attestation) and a verifier (who sees a presentation of the attestation) or any other party. No salted attribute digest based solution can satisfy this unlinkability requirement as issuers are always able to link user behaviour through the disclosure of the highly linkable issuer's digital signature.

Moreover, (b) appears to suggest that unlinkability is only obligatory when the (Q)EAA does not require user identification. One plausible interpretation is that unlinkability may not be obligatory in cases where an (Q)EAA presentation includes user identifying attributes.

It is not clear if (a) is a restriction to the acquisition of data, or if it is a requirement that the data are unlinkable. If the article is a restriction on the acquisition of data, then contractual terms that prevent data sharing may be enough even in cases where the data are linkable (e.g. using salted attribute hashes approach such as mdoc and SD-JWT). Conversely, if the data has to be unlinkable then technical solutions are required that ensure unlinkable (Q)EAAs. This may require that issuers issue a (Q)EAA in such a way that even a coalition of colluding issuers and verifiers has no ability of linking together attribute presentations on the basis of the data shared with a greater probability than pure guessing (e.g. using signature blinding and ZKP of valid signature).

It is also possible that the legal text intended unlinkable data without fully considering its technical feasibility or the relationship between unlinkable data and privacy. For instance, consider recital 14:

"Member States should integrate different privacy-preserving technologies, such as zero knowledge proof, into the European Digital Identity Wallet. Those cryptographic methods should allow a relying party to validate whether a given statement based on the person's identification data and attestation of attributes is true, without revealing any data on which that statement is based, thereby preserving the privacy of the user."

There are two main issues with this recital and the strong focus on unlinkable data. Firstly, the recital presumes that cryptographic unlinkability can ensure privacy. Cryptographic methods can only guarantee unlinkability of the data itself, and do not guarantee anything with regards to the unlinkability of an identity subject. While unlinkability of data can be achieved using cryptographic operations, the unlinkability of the identity subject requires that the user's presentation is devoid of any information (contextual or auxiliary) that reduces the verifier's uncertainty of who the identity subject is. Secondly, advanced ZKP schemes (see clause 4.5) are not yet standardized in a way that can be referenced by the eIDAS2 implementing acts. Moreover, eIDAS2 article 5a.14 states:

"Users shall have full control of the use of and of the data in their European Digital Identity Wallet. The provider of the European Digital Identity Wallet shall neither collect information about the use of the European Digital Identity Wallet which is not necessary for the provision of European Digital Identity Wallet services, nor combine person identification data or any other personal data stored or relating to the use of the European Digital Identity Wallet with personal data from any other services offered by that provider or from third party services which are not necessary for the provision of European Digital Identity Wallet services, unless the user has expressly requested otherwise."

Hence, eIDAS2 article 5a.14 puts a requirement on the EUDI Wallet Providers to not gather unnecessary personal data, which in turn could be used for issuer collusion of linkable user information.

In conclusion, selective disclosure and unlinkability are potential components in a privacy-by-design solution. Their impact on privacy depends, however, on an entity's ability to reduce uncertainty about a user's identity from the attribute presentation. When an entity relies solely on linking attributes to reduce uncertainty (and few do), selective disclosure and unlinkability are vital. However, when the entity controls the context or requires user identifying attributes for service provision, non-technical measures (e.g. contractual, economic, and/or regulatory) may be necessary to ensure user privacy and data control.

No technical solution can offer complete control over data and privacy, which requires a more comprehensive approach. Determined, potentially malicious, and well-resourced entities can identify a user and map their behaviour regardless of technical countermeasures employed. This stems from the inherently leaky nature of (Q)EAA presentations, even presentations that do not contain identity subject attributes. For instance, in the context of the European Digital Identity Wallet, a presentation and the associated flow reveals, among other things, that the user has a certified and capable device, often an IP address, attestation issuance dates, identifies an actor the user has been in contact with, and reveals an attestation type the user is eligible to request.

The above regulatory discussion notwithstanding, the present document focuses on various technical solutions that can increase the cost associated with uncertainty reduction (and thus e.g. on verifier and issuer unlinkability). Any (Q)EAA solution that seeks to ensure user privacy has to consider these technical solutions as part of a more comprehensive approach.

Identity matching in eIDAS2

The proposed eIDAS2 regulation [i.103] also includes recitals and articles on identity matching. Recital 55 in eIDAS2 defines identity matching as follows:

"'identity matching' means a process where person identification data, or electronic identification means are matched with or linked to an existing account belonging to the same person;"

Furthermore, eIDAS2 article 11a.2 states:

"Member States shall provide for technical and organisational measures to ensure a high level of protection of personal data used for identity matching and to prevent the profiling of users."

High level protection of personal data for identity matching can be achieved with selective disclosure of attributes.

EXAMPLE 8: Assume that a relying party requests a user to get identified based on the attributes Name, Date of birth and Place of birth. The relying party will need these attributes only to perform identity matching. Instead of revealing the entire PID, which will provide superfluous person identification data to the relying party, the user can select to disclose only the requested attributes Name, Date of birth and Place of birth, which the relying party can use to perform the identity matching.

Descriptions of selective disclosure and unlinkability in the ARF

The ARF [i.71] also defines the term selective disclosure as follows in clause 2:

"The capability of the EUDI Wallet that enables the User to present a subset of attributes provided by the PID and/or (Q)EAAs."

Furthermore, in the ARF outline [i.70] the term unlinkability is also introduced as follows in clause 5:

"The Wallet shall ensure an appropriate level of privacy, implementing policies about non-traceability and unlinkability of user's activities for third parties as appropriate considering:

- *the applicable legal context for identity providers and attestation providers;*
- *the need to retain evidence for dispute resolution purpose;*
- *the right for the user to be informed of the use of their EUDI Wallet".*

More specifically, the ARF [i.71] mandates ISO/IEC 18013-5 [i.181] Mobile Security Object (MSO) and IETF SD-JWT to enable selective disclosure of the EUDI Wallet PID formats. In the ARF [i.71] section 5.1.2 "Issuing requirements for PID" it is stated:

"PID attestation MUST enable Selective Disclosure of attributes by using Selective Disclosure for JWTs (SD-JWT) and Mobile Security Object (ISO/IEC 18013-5) scheme according to the data model."

The mdoc and IETF SD-JWT are mandatory as PID selective disclosure mechanisms in use cases where the Relying Party relies on LoA High as defined in EU CIR 2015/1502 [i.99], to enable cross border identification using PID attributes at LoA High. Hence, the requirements in EU CIR 2015/1502, in conjunction with Regulation (EU) No 1025/2012 on European standardisation [i.105] and the SOG-IS catalogue of approved cryptographic algorithms [i.237], have resulted in this restricted selection of PID formats for the EUDI Wallet.

However, the ARF also specifies the EUDI Wallet support for additional (Q)EAA formats and proof mechanisms, which aims at enabling flexibility and additional feature support for use cases that cannot be met by ISO mDL MSO and IETF SD-JWT (such as in the areas of health, education credentials, etc.). Hence, the EUDI Wallet allows for other selective disclosure techniques based on multi-message signature schemes or proofs for arithmetic circuits but does not mandate support for these.

It should be observed that the ARF holds no legal value and does not prejudge the forthcoming legislative process and the final mandatory legal requirements for EUDI Wallets. Nor does it discuss unlinkability to the same extent as selective disclosure. Only the finally adopted eIDAS2 regulation [i.103], and the implementing and delegated acts adopted under that legal basis, will be mandatory. The ARF will be aligned to the final adoption of eIDAS2. Hence, the ARF provides guidelines to the present document for the PID formats to be analysed with respect to selective disclosure in the context of eIDAS2, although the present document may also provide recommendations for additional selective disclosure and ZKP schemes for future versions of the ARF or to be considered for further ETSI standardization.

1 Scope

The present document analyses cryptographic schemes for selective disclosure and their potential application for privacy of electronic attestation attributes in line with the expected requirement of the proposed regulation amending Regulation (EU) No 910/2014 (commonly called eIDAS2) [i.103].

NOTE 1: The term selective disclosure is a collective term that may also include various concepts of unlinkability, and predicates such as range proofs, depending on the context of the specific cryptographic scheme. The scope of the present document is primarily to describe selective disclosure and unlinkability properties of each analysed cryptographic scheme.

NOTE 2: Range proofs, and more general predicate proofs as well as general-purpose ZKPs are out of scope in the ARF [i.71]. If an analysed cryptographic scheme relies on any of these features, they will be described in the context of that particular cryptographic scheme.

The present document aims at providing a comprehensive overview of existing cryptographic schemes for selective disclosure and the formats and protocols associated with these cryptographic schemes.

The aim of the present document is first to provide input to ETSI standardization relating to how selective disclosure may be applied to the eIDAS2 (Qualified) Electronic Attribute Attestations ((Q)EAA) and Person Identification Data (PID). More specifically, the present document may serve as input to (Q)EAA issuance policies as being specified in ETSI TS 119 471 [i.96] and (Q)EAA profiles as being specified in ETSI TS 119 472-1 [i.97].

Second, the present document will also analyse the policy requirements for (Q)TSPs and PID providers issuing (Q)EAAs or PIDs with selective disclosure capabilities to EUDI Wallets.

Third, the present document analyses how the user of an EUDI Wallet can present selected attributes of a (Q)EAA or PID to relying parties (or (Q)TSPs acting as relying parties). Consequently, the present document can highlight needs that may require future standardization efforts.

The present document analyses the concepts of selective disclosure, unlinkability, and predicates (including range proofs) in the following main clauses:

- Selective disclosure signature schemes (clause 4): This clause describes the academic research of the cryptographic algorithms and schemes that shape the foundation for selective disclosure signature schemes.
- Selective disclosure (Q)EAA formats (clause 5): This clause describes the (Q)EAA formats that have been developed and standardized based on the aforementioned selective disclosure signature schemes.
- Selective disclosure protocols and systems (clause 6): This clause describes the complete protocols and /or systems that have been developed and standardized based on the aforementioned selective disclosure signature schemes and (Q)EAA formats.

Since the ARF [i.71] specifies the PID to be issued to an EUDI Wallet as mdoc [i.181] (with MSO for selective disclosure) or W3C Verifiable Credentials (with SD-JWT for selective disclosure), these formats and protocols are analysed in more detail in clause 7.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] Adida: "Helios: Web-based Open-Audit Voting".

[i.2] ANSSI, BSI et al: "Securing Tomorrow, Today: Transitioning to Post-Quantum Cryptography".

[i.3] Paquin, Zaverucha: "U-Prove Cryptographic Specification V1.1".

[i.4] Alikhani, Brunner, Crépeau et al: "Experimental relativistic zero-knowledge proofs".

[i.5] Altmann: "A third party repudiable ZKP-based PoA".

[i.6] Altmann: "Inequality tests in salted attribute digest based attestations".

[i.7] Ames, Hazay, Ishai, Venkitasubramaniam: "Ligero: Lightweight Sublinear Arguments Without a Trusted Setup".

[i.8] Arapinis, Cortier, Kremer, Ryan: "Practical Everlasting Privacy".

[i.9] Argo, Güneysu, Jeudy et al: "Practical Post-Quantum Signatures for Privacy".

[i.10] Argo, Jeudy, Land: "Lattice Anonymous Credentials".

[i.11] Asghar: "A Survey on Blind Digital Signatures".

[i.12] Au, Susilo, Mu: "Constant-size dynamic k-TAA".

[i.13] Au, Tsang, Susilo, Mu: "Dynamic Universal Accumulators for DDH Groups and Their Application to Attribute-Based Anonymous Credential Systems".

[i.14] Babel, Sedlmeir: "Bringing data minimization to digital wallets at scale with general-purpose zero-knowledge proofs".

[i.15] Barki, Brunet, Desmoulins, Traoré: "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials".

[i.16] Bellare, Rogaway: "Random oracles are practical: A paradigm for designing efficient protocols".

[i.17] Ben-Sasson, Bentov, Horesh, Riabzev: "Scalable, transparent, and post-quantum secure computational integrity (zk-STARK)".

[i.18] Ben-Sasson, Bentov-Horesh, Riabzev: "Scalable zero-knowledge with No Trusted Setup".

[i.19] Ben-Sasson, Chiesa, Genkin, Tromer, Virza: "SNARKs for C: Verifying Program Executions Succinctly and in zero-knowledge".

[i.20] Ben-Sasson, Chiesa, Riabzev, Spooner: "Aurora: Transparent Succinct Arguments for R1CS".

[i.21] Ben-Sasson, Tromer: "Succinct Non-Interactive zero-knowledge for a von Neumann Architecture".

[i.22] Benhamouda, Lepoint, Loss et al: "On the (in)Security of ROS".

[i.23] Benjumea, Lopez, Montenegro, Troya: "A First Approach to Provide Anonymity in Attribute Certificates".

[i.24] Bennett, Brassard: "Quantum cryptography: Public key distribution and coin tossing".

[i.25] Bitcoin: "BIP-32 Bitcoin Improvement Proposal 32".

[i.26] Boneh, Bortz et al: "Private Information Retrieval".

[i.27] Boneh, Boyen-Shacham: "Short Group Signatures".

[i.28] Boneh, Lynn-Shacham: "Short Signatures from the Weil Pairing".

[i.29] Bootle, Lyubashevsky-Nguyen-Sorniotti: "A Framework for Practical Anonymous Credentials from Lattices".

[i.30] Bowe: "BLS12-381: New zk-SNARK Elliptic Curve Construction".

[i.31] Bowe, Grigg, Hopwood: "Recursive Proof Composition without a Trusted Setup".

[i.32] Brands: "A Technical Overview of Digital Credentials".

[i.33] Brands: "Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy".

[i.34] Brassard, Chaum, Crépeau: "Minimum disclosure proofs of knowledge".

[i.35] Broadbent, Ji-Song, Watrous: "Zero-knowledge proof systems for QMA".

[i.36] BSI TR-03110: "Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS token".

[i.37] Bull, Stanski, Squire: "Content extraction signatures using XML digital signatures and custom transforms on-demand".

[i.38] Bünz, Bootle, Boneh: "Bulletproofs: Short Proofs for Confidential Transactions and More".

[i.39] Bünz, Fisch, Szepieniec: "Transparent SNARKs from DARK Compilers".

[i.40] Camenisch, Drijvers, Lehmann: "Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited".

[i.41] Camenisch, Drijvers, Lehmann-Neven-Towa: "Short Threshold Dynamic Group Signatures".

[i.42] Camenisch, Lysyanskaya: "A Signature Scheme with Efficient Protocols".

[i.43] Camenisch, Lysyanskaya: "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation".

[i.44] Camenisch, Lysyanskaya: "Dynamic accumulators and application to efficient revocation of anonymous credentials".

[i.45] Camenisch, Lysyanskaya: "Signature Schemes and Anonymous Credentials from Bilinear Maps".

[i.46] Camenisch, Mödersheim, Sommer: "A Formal Model of Identity Mixer".

[i.47] Camenisch, Shoup: "Practical verifiable encryption and decryption of discrete logarithms".

[i.48] Camenisch, Stadler: "Efficient group signature schemes for large groups".

[i.49] Campanelli, Fiore, Querol: "LegoSNARK: Modular Design and Composition of Succinct Zero-Knowledge Proofs".

[i.50] Canard, Coisel, Jambert, Traoré: "New Results for the Practical Use of Range Proofs".

[i.51] Carney: "On Zero-Knowledge Proofs over the Quantum Internet".

[i.52] Castryck, Galbraith, Farashahi: "Efficient arithmetic on elliptic curves using a mixed Edwards-Montgomery representation".

[i.53] Celi, Levin, Rowell: "CDLS: Proving Knowledge of Committed Discrete Logarithms with Soundness".

[i.54] CEN TC/224 WG17: "EN 419 211: Protection profiles for secure signature creation device", (produced by CEN).

[i.55] CEN TC/224 WG20: "New work item: PID onboarding technical standard", (produced by CEN).

[i.56] Chadwick: "The Use of FIDO2 and Verifiable Credentials".

[i.57] Chairattana, Apirom, Harding, Lysyanskaya, Tessaro: "Server-Aided Anonymous Credentials".

[i.58] Chalkias, Cohen, Lewi, Moezinia, Romailler: "HashWires: Credential-Based Range Proofs".

[i.59] Chase, Meiklejohn, Zaverucha: "Algebraic MACs and keyed-verification anonymous credentials".

[i.60] Chaum: "Blind signatures for untraceable payments".

[i.61] Chaum, Pedersen: "Wallet Databases with Observers".

[i.62] Chaum, van Heyst: "Group Signatures".

[i.63] Chen, Page, Smart: "On the Design and Implementation of an Efficient DAA Scheme".

[i.64] Chen: "A DAA scheme requiring less TPM resources".

[i.65] Chiesa, Bitansky, Canetti: "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again".

[i.66] Chiesa, Hu, Maller, Mishra: "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS".

[i.67] Chiesa, Ohja, Spooner: "FRACTAL: Post-Quantum and Transparent Recursive Proofs from Holography".

[i.68] Circom: "Circuit Compiler".

[i.69] Circom-ECDSA: "Implementation of ECDSA operations in Circom".

[i.70] Common Union Toolbox for a Coordinated Approach Towards a European Digital Identity Framework: "The European Digital Identity Wallet Architecture and Reference Framework, Outline".

[i.71] Common Union Toolbox for a Coordinated Approach Towards a European Digital Identity Framework: "The European Digital Identity Wallet Architecture and Reference Framework".

NOTE: The European Digital Identity Wallet Architecture and Reference Framework is commonly called the ARF.

[i.72] Costello, Fournet, Howell et al.: "Geppetto: Versatile Verifiable Computation".

[i.73] Crites: "Delegatable Anonymous Credentials from Mercurial Signatures".

[i.74] Crites, Lysyanskaya: "Mercurial Signatures for Variable-Length Messages".

[i.75] CRYSTALS: "Dilithium digital signature scheme".

[i.76] Damgård and Triandopoulos: "Supporting Non-membership Proofs with Bilinear-map Accumulators".

[i.77] Delignat, Lavaud, Fournet, Kohlweiss, Parno: "Cinderella: Turning Shabby X.509 Certificates into Elegant Anonymous Credentials with the Magic of Verifiable Computation".

[i.78] Desmoulins, Dumanois, Kane Traoré: "Making BBS Anonymous Credentials eIDAS 2.0 Compliant".

[i.79] Diamond, Posen: "Polylogarithmic Proofs for Multilinears over Binary Towers".

[i.80] DIF: "Blind Signatures extension of the BBS Signature Scheme".

[i.81] DIF: "Presentation Exchange 2.0.0".

[i.82] DIF: "Wallet Security Working Group".

[i.83] Dutto, Margaria, Sanna, Vesco: "Toward a Post-Quantum Zero-Knowledge Verifiable Credential System for Self-Sovereign Identity".

[i.84] Eaton, Lepoint, Wood: "Security Analysis of Signature Schemes with Key Blinding".

[i.85] Eberhardt, Tai: "ZoKrates - Scalable Privacy-Preserving Off-Chain Computations".

[i.86] European Banking Association (EBA): "Register of payment and electronic money institutions under PSD2".

[i.87] Ebrahimi: "Post-quantum Efficient Proof for Graph 3-Coloring Problem".

[i.88] ETSI EN 319 162-1: "Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC); Part 1: Building blocks and ASiC baseline containers".

[i.89] ETSI EN 319 401: "Electronic Signatures and Trust Infrastructures (ESI); General Policy Requirements for Trust Service Providers".

[i.90] ETSI EN 319 411-1: "Electronic Signatures and Trust Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements".

[i.91] ETSI TR 119 001: "Electronic Signatures and Infrastructures (ESI); The framework for standardization of signatures; Definitions and abbreviations".

[i.92] ETSI TR 119 479-2: "Electronic Signatures and Trust Infrastructures (ESI); Technological Solutions for the EU Digital Identity Framework; Part 2: EAA Extended Validation Services Framework and Application".

[i.93] ETSI TS 119 495: "Electronic Signatures and Trust Infrastructures (ESI); Sector Specific Requirements; Certificate Profiles and TSP Policy Requirements for Open Banking".

[i.94] ETSI TS 119 612: "Electronic Signatures and Trust Infrastructures (ESI); Trusted Lists ".

[i.95] ETSI TS 119 462: "Electronic Signatures and Trust Infrastructures (ESI); Wallet interfaces for trust services and signing".

[i.96] ETSI TS 119 471: "Electronic Signatures and Trust Infrastructures (ESI); Policy and Security requirements for Providers of Electronic Attestation of Attributes Services".

[i.97] ETSI TS 119 472-1: "Electronic Signatures and Trust Infrastructures (ESI); Profiles for Electronic Attestation of Attributes; Part 1: General requirements".

[i.98] European Banking Authority: "Regulatory Technical Standards on strong customer authentication and secure communication under PSD2".

[i.99] Commission Implementing Regulation (EU) 2015/1502 of 8 September 2015 on setting out minimum technical specifications and procedures for assurance levels for electronic identification means pursuant to Article 8(3) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market.

[i.100] Commission Implementing Decision (EU) 2015/1505 of 8 September 2015 laying down technical specifications and formats relating to trusted lists pursuant to Article 22(5) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market.

[i.101] Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC.

NOTE: The Directive (EU) 2015/2366 is commonly called PSD2.

[i.102] Regulation (EU) No 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

[i.103] European Parliament legislative resolution of 29 February 2024 on the proposal for a regulation of the European Parliament and of the Council amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity (COM(2021)0281 - C9-0200/2021 - 2021/0136(COD)).

NOTE 1: The Proposal for a regulation of the European Parliament and of the Council amending Regulation (EU) No 910/2014 is commonly called eIDAS2.

NOTE 2: The European Commission issued the first proposal of eIDAS2 in June 2021. The European Council issued an amended edition of eIDAS2 in December 2022 and the European Parliament issued another amended edition of eIDAS2 in February 2023. The eIDAS2 proposal, which is based on the agreement in the eIDAS2 trialogue, was published in December 2023. Finally, the EU Parliament voted to approve the eIDAS2 regulation in February 2024. Unless stated otherwise, the eIDAS2 adopted text, which was issued in February 2024, is by default the referenced version of the eIDAS2 regulation.

[i.104] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.

NOTE: The Directive (EU) 910/2014 is commonly called eIDAS.

[i.105] Regulation (EU) No 1025/2012 of the European Parliament and of the Council on European standardisation, amending Council Directives 89/686/EEC and 93/15/EEC and Directives 94/9/EC, 94/25/EC, 95/16/EC, 97/23/EC, 98/34/EC, 2004/22/EC, 2007/23/EC, 2009/23/EC and 2009/105/EC of the European Parliament and of the Council and repealing Council Decision 87/95/EEC and Decision No 1673/2006/EC of the European Parliament and of the Council.

[i.106] Eurosmart PP-0117: "Protection Profile for Secure Sub-System in System-on-Chip (3S in SoC)".

[i.107] Evans, Angeris: "Succinct Proofs and Linear Algebra".

[i.108] Faz, Hernández, Ladd, Maram: "ZKAttest: Ring and Group Signatures for Existing ECDSA Keys".

[i.109] Federal Public Key Infrastructure Policy Authority: "X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework".

[i.110] FIDO Alliance: "Fast Identity Online v2 (FIDO2)".

[i.111] Fleischhacker, Krupp, Malavolta, Schneider: "Efficient Unlinkable Sanitizable Signatures from Signatures with Re-Randomizable Keys".

[i.112] Flamini, Ranise, Sciarretta et al: "Public Key Accumulators for Revocation of Non-Anonymous Credentials".

[i.113] Frigo, Shelat: "Anonymous credentials from ECDSA".

[i.114] Frymann, Gardham, Kiefer et al: "Asynchronous Remote Key Generation: An Analysis of Yubico's Proposal for W3C WebAuthn".

[i.115] Frymann, Gardham, Manulis: "Asynchronous Remote Key Generation for Post-Quantum Cryptosystems from Lattices".

[i.116] Gabison, Williamson, Ciobotaru: "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge".

[i.117] Garcia, Rodriguez, Moreno, Bernabe, Skarmeta: "Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures".

[i.118] Global Platform: "TEE Protection Profile".

[i.119] Goldreich, Micali, Wigderson: "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems".

[i.120] Goldwasser, Micali, Rackoff: "The knowledge complexity of interactive proof systems".

[i.121] Google®: "Using A Trusted Execution Environment As A Trusted Third Party Providing Privacy For Attestation".

[i.122] Grassi, Khovratovich, Rechberger, Schofnegger: "POSEIDON: A New Hash Function for Zero-Knowledge Proof Systems".

[i.123] Grontas, Pagourtzis, Zacharakis, Zhang: "Towards everlasting privacy and efficient coercion resistance in remote electronic voting".

[i.124] Groth: "On the Size of Pairing-based Non-interactive Arguments".

[i.125] Groth: "Short pairing-based non-interactive zero-knowledge arguments".

[i.126] Grover: "A fast quantum mechanical algorithm for database search".

[i.127] Guo, Feng, Wu, Li: "Benchmarking ZK-Friendly Hash Functions and SNARK Proving Systems for EVM-compatible Blockchains".

[i.128] Haines, Gritti: "Improvements in Everlasting Privacy: Efficient and Secure Zero Knowledge Proofs".

[i.129] Haines, Mosaheb, Müller, Pryvalov: "SoK: Secure e-voting with everlasting privacy".

[i.130] Heath, Yang, Devecsery, Kolesnikov: "Zero Knowledge for Everything and Everyone: Fast ZK Processor with Cached ORAM for ANSI C Programs".

[i.131] Hyperledger Foundation: "AnonCreds Specification v1.0".

[i.132] Hyperledger Foundation: "Hyperledger Aries".

[i.133] Hyperledger Foundation: "Hyperledger Fabric".

[i.134] Hyperledger Foundation: "Hyperledger Indy".

[i.135] Hyperledger Foundation: "Hyperledger Ursa SDK".

[i.136] IBM® Research: "Identity Mixer (IDEMIX)".

[i.137] ICT Trust and Security Research: "Attribute based Credentials for Trust (ABC4Trust)".

[i.138] Iden3: "BJJSignature2021".

[i.139] Iden3: "Iden3 Documentation".

[i.140] Iden3: "Iden3SparseMerkleTreeProof".

[i.141] Iden3: "JSON Web Zero-knowledge".

[i.142] IETF: "Authentic Chained Data Containers (ACDC)".

[i.143] IETF: "SD-JWT-based Verifiable Credentials (SD-JWT VC)".

[i.144] IETF: "Self-Addressing IDentifier (SAID)".

[i.145] IETF: "The Gordian Envelope Structured Data Format".

[i.146] IETF CFRG: "Asynchronous Remote Key Generation (ARKG) algorithm".

[i.147] IETF CFRG: "libZK: a zero-knowledge proof library".

[i.148] IETF: "Key Blinding for Signature Schemes".

[i.149] IETF IESG: "JOSE and COSE Encoding for Post-Quantum Signatures".

[i.150] IETF JOSE: "JSON Proof Algorithms".

[i.151] IETF JOSE: "JSON Proof Token and CBOR Proof Token".

[i.152] IETF JOSE: "JSON Web Proof (JWP)".

[i.153] IETF OAUTH: "OAuth Status List".

[i.154] IETF OAUTH: "SD-JWT-based Verifiable Credentials with JSON payloads (SD-JWT VC)".

[i.155] IETF OAUTH: "Selective Disclosure for JWTs (SD-JWT)".

[i.156] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[i.157] IETF RFC 5652: "Cryptographic Message Syntax (CMS)".

[i.158] IETF RFC 5755: "An Internet Attribute Certificate Profile for Authorization".

[i.159] IETF RFC 6066: "Transport Layer Security (TLS) Extensions: Extension Definitions".

[i.160] IETF RFC 6960: "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP)".

[i.161] IETF RFC 6818: "Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[i.162] IETF RFC 7049: "Concise Binary Object Representation (CBOR)".

[i.163] IETF RFC 7515: "JSON Web Signature (JWS)".

[i.164] IETF RFC 7516: "JSON Web Encryption (JWE)".

[i.165] IETF RFC 7519: "JSON Web Token (JWT)".

[i.166] IETF RFC 7748: "Elliptic Curves for Security".

[i.167] IETF RFC 8152: "CBOR Object Signing and Encryption (COSE)".

[i.168] IETF RFC 8235: "Schnorr Non-interactive Zero-Knowledge Proof".

[i.169] IETF RFC 8259: "JavaScript Object Notation (JSON) Data Interchange Format".

[i.170] IETF RFC 8610: "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures".

[i.171] IETF RFC 9162: "Certificate Transparency Version 2.0".

[i.172] IETF RFC 9380: "Hashing to Elliptic Curves".

[i.173] IRMA: "Revocation".

[i.174] IRTF CFRG: "BBS per Verifier Linkability".

[i.175] IRTF CFRG: "Blind BBS Signatures".

[i.176] IRTF CFRG: "Hierarchical Deterministic Keys".

[i.177] IRTF CFRG: "The BBS Signature Scheme".

[i.178] IRTF CFRG: "Pairing-Friendly Curves".

[i.179] ISO/IEC 9796 series: "Information technology — Security techniques — Digital signature schemes giving message recovery".

[i.180] ISO/IEC 14888-3: "IT Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms".

[i.181] ISO/IEC 18013-5: "Personal identification — ISO-compliant driving licence — Part 5: Mobile driving licence (mDL) application".

[i.182] ISO/IEC CD 18013-7: "Personal identification — ISO-compliant driving licence — Part 7: Mobile driving licence (mDL) add-on functions".

[i.183] ISO/IEC 18370 series: "Information technology — Security techniques — Blind digital signatures".

[i.184] ISO/IEC 20008 series: "Information technology — Security techniques — Anonymous digital signatures".

[i.185] ISO/IEC 24843: "Privacy-preserving attribute-based credentials".

[i.186] ISO/IEC 23220-3: "Cards and security devices for personal identification — Building blocks for identity management via mobile devices — Part 3: Protocols and services for installation and issuing phase".

[i.187] ISO/IEC CD 23220-4: "Cards and security devices for personal identification — Building blocks for identity management via mobile devices — Part 4: Protocols and services for operational phase".

[i.188] ISO/IEC CD 23220-6: "Cards and security devices for personal identification — Building blocks for identity management via mobile devices — Part 6: Mechanism for use of certification on trustworthiness of secure areas".

[i.189] ISO/IEC 27001: "Information security, cybersecurity and privacy protection — Information security management systems — Requirements".

[i.190] ISO/IEC 27002: "Information security, cybersecurity and privacy protection — Information security controls".

[i.191] ISO/IEC CD 27565: "Guidelines on privacy preservation based on zero knowledge proofs".

[i.192] Jeudy, Roux, Langlois, Sanders: "Lattice Signature with Efficient Protocols, Application to Anonymous Credentials".

[i.193] Kampanakis, Panburana, Daw, Van Geest: "The Viability of Post-Quantum X.509 Certificates".

[i.194] Kosba, Papadopoulos, Papamanthou, Song: "MIRAGE: Succinct Arguments for Randomized Algorithms with Applications to Universal zk-SNARKs".

[i.195] Kosba, Papmanthou, Shi: "xJsnark: A Framework for Efficient Verifiable Computation".

[i.196] Lapon, Kohlweiss, Decker, Naessens: "Analysis of Revocation Strategies for Anonymous Idemix Credentials".

[i.197] Libert, Ling, Mouhartem et al: "Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions".

[i.198] Maller, Bowe, Kohlweiss, Meiklejohn: "Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings".

[i.199] Menezes: "An Introduction to Pairing-Based Cryptography".

[i.200] Microsoft® Research: "Spartan: High-speed zkSNARKs without trusted setup".

[i.201] Microsoft® Research: "U-Prove".

[i.202] Morales, Agudo, Lopez: "Private set intersection: A systematic literature review".

[i.203] Mouris, Tsoutsos: "Zilch: A Framework for Deploying Transparent Zero-Knowledge Proofs".

[i.204] Nguyen: "Accumulators from Bilinear Pairings and Applications".

[i.205] Nitulescu: "zk-SNARKs: A Gentle Introduction".

[i.206] NIST: "Digital Identities - Mobile Driver's License (mDL)".

[i.207] NIST FIPS 204: "Module-Lattice-Based Digital Signature Standard".

[i.208] NIST FIPS 205: "Stateless Hash-Based Digital Signature Standard".

[i.209] NIST IR 8547 initial public draft: "Transition to Post-Quantum Cryptography Standards".

[i.210] NIST: "Post-Quantum Cryptography (PQC)".

[i.211] OpenAttestation: "Document Integrity".

[i.212] OpenID Foundation: "OpenID Connect Core 1.0".

[i.213] OpenID Foundation: "OpenID for Verifiable Credentials Issuance".

[i.214] OpenID Foundation: "OpenID for Verifiable Presentations".

[i.215] OpenID Foundation: "OpenID4VC High Assurance Interoperability Profile with SD-JWT VC".

[i.216] OpenID Foundation: "Self-Issued OpenID Provider v2".

[i.217] Orange™: "Trust Model: Securing digital identity with advanced cryptographic algorithms".

[i.218] Orrù, Tessaro, Zaverucha, Zhu: "Oblivious issuance of proofs".

[i.219] Paquin, Policharla, Zaverucha: "Crescent: Stronger Privacy for Existing Credentials".

[i.220] Parno, Howell, Gentry et al: "Pinocchio: Nearly Practical Verifiable Computation".

[i.221] Patrick Amrein: "BBS Device Binding using conventional P256 Signature", 2025.

[i.222] Petkus: "Why and How zk-SNARK Works: Definitive Explanation".

[i.223] Pointcheval, Sanders: "Short Randomizable Signatures".

[i.224] PrimeLife: "Identity Mixer".

[i.225] Proos, Zalka: "Shor's discrete logarithm quantum algorithm for elliptic curves".

[i.226] Pussewalage, Oleshchuk: "An Efficient Multi-Show Unlinkable Attribute Based Credential Scheme for a Collaborative E-Health Environment".

[i.227] Radboud University Nijmegen: "IRMA project".

[i.228] Ramos: "Evaluation of trust service and software product regimes for zero-knowledge proof development under eIDAS 2.0".

[i.229] Rivest, Shamir: "PayWord and MicroMint: Two simple micropayment schemes".

[i.230] Roetteler, Naehrig, Svore, Lauter: "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms".

[i.231] Rosenberg, White, Garman, Miers: "zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure".

[i.232] Sanders: "Efficient Redactable Signature and Application to Anonymous Credentials".

[i.233] Sanders: "Improving Revocation for Group Signature with Redactable Signature".

[i.234] Setty: "Spartan: Efficient and general-purpose zkSNARKs without trusted setup".

[i.235] Shor: "Algorithms for quantum computation: discrete logarithms and factoring".

[i.236] SLIP-0010: "Universal private key derivation from master private key".

[i.237] SOG-IS Crypto Working Group: "SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms".

[i.238] SPHINCS+: "Stateless hash-based signature scheme".

[i.239] Steinfeld, Bull, Zheng: "Content Extraction Signatures".

[i.240] Tan-Groß: "MoniPoly - An Expressive q-SDH-Based Anonymous Attribute-Based Credential System".

[i.241] Thaler: "SNARK Security and Performance".

[i.242] Trusted Computing Group: "TPM 2.0 Library".

[i.243] U.S. Congress H.R.7535: "Quantum Computing Cybersecurity Preparedness Act", 117th Congress (2021-2022).

[i.244] U.S. Department of Homeland Security: "Cryptography Review of W3C Verifiable Credentials Data Model (VCDM) and Decentralized Identifiers (DIDs) Standards and Cryptography Implementation Recommendations".

[i.245] Verheul: "Attestation Proof of Association".

[i.246] Verheul: "SECDSA: Mobile signing and authentication under classical 'sole control'".

[i.247] Vitto, Biryukov: "Dynamic Universal Accumulator with Batch Update over Bilinear Groups".

[i.248] Wang, Hazay, Venkitasubramaniam: "Ligetron: Lightweight Scalable End-to-End Zero-Knowledge Proofs Post-Quantum ZK-SNARKs on a Browser".

[i.249] Wahby, Setty, Ren, Blumberg, Walfish: "Efficient RAM and Control Flow in Verifiable Outsourced Computation".

[i.250] Wahby, Tzialla, Shelat: "Doubly-Efficient zkSNARKs Without Trusted Setup".

[i.251] Watrous: "Zero-knowledge against quantum attacks".

[i.252] Webber, Elfving, Weidt, Hensinger: "The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime".

[i.253] Woo: "Efficient Proofs of Possession for Legacy Signatures".

[i.254] W3C®: "Bitstring Status List v1.0 - Privacy-preserving status information for Verifiable Credentials".

[i.255] W3C®: "Data Integrity BBS Cryptosuites v1.0".

[i.256] W3C®: "Data Integrity ECDSA Cryptosuites v1.0".

[i.257] W3C®: "Decentralized Identifiers (DIDs) v1.0".

[i.258] W3C®: "Json Web Proofs for Binary Merkle Trees".

[i.259] W3C®: "Merkle Disclosure Proof 2021".

[i.260] W3C®: "Remove securing JSON, VC-JWT issue #88".

[i.261] W3C®: "Securing Verifiable Credentials using JOSE and COSE".

[i.262] W3C®: "Universal Wallet 2020".

[i.263] W3C®: "Verifiable Credentials Data Integrity 1.0".

[i.264] W3C®: "Verifiable Credentials Data Model v1.1".

[i.265] W3C®: "Verifiable Credential Data Model v2.0 (working draft)".

[i.266] W3C®: "Web Authentication: An API for accessing Public Key Credentials Level 2".

[i.267] W3C® CCG: "BBS Cryptosuite v2023".

[i.268] WhiteHat, Baylina, Bellés: "Baby Jubjub Elliptic Curve".

[i.269] Yeoh, Kepkowski, Heide, Kaafar, Hanzlik: "Fast IDentity Online with Anonymous Credentials (FIDO-AC)".

[i.270] ZKPassport: "Privacy-preserving identity verification using passports and ID cards".

[i.271] ZKProof: "HashWires: Range Proofs from Hash Functions".

[i.272] zk-regex: "A library to compile regex verification in Circom".

[i.273] Zhang, Xie, Zhang, Song: "Transparent Polynomial Delegation and Its Applications to zero-knowledge Proof".

[i.274] Zhao, Liu, Wu et al: "A Tutorial on Quantum Key Distribution".

[i.275] Johnson et al: "Homomorphic signature schemes."

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI TR 119 001 [i.91], ETSI EN 319 401 [i.89] and the following apply:

atomic (Q)EAA: (Qualified) Electronic Attestation of Attribute with a single attribute claim

attribute: feature, characteristic or quality of a natural or legal person or of an entity, in electronic form

NOTE: As defined in the ARF [i.71].

authentic source: repository or system, held under the responsibility of a public sector body or private entity, that contains attributes about a natural or legal person and is considered to be the primary source of that information or recognized as authentic in national law

NOTE: As defined in the ARF [i.71].

blind signature: type of digital signature in which the content of a message is disguised (blinded) before it is signed

EXAMPLE: The concept of blind signatures can be exemplified by a voting system in the physical world. The voter encloses an anonymous ballot in a carbon envelope with the voter's name written on the outside. An official verifies the voter's identity and signs the envelope, such that the ballot inside the carbon envelope gets signed with the official's signature. The voter moves the signed ballot to a new unmarked envelope. Hence, the signing official does not see the content of the vote, but a third party can later verify its signature and know that the vote is valid.

NOTE 1: Blinded signatures cater for unlinkability, since the verifier cannot link the signed messages back to the user.

NOTE 2: The U-Prove scheme (clause 6.6.2) utilizes blinded signatures when issuing the credentials.

NOTE 3: Blind signatures are specified in the ISO/IEC 18370 series [i.183], which allow a user to obtain a digital signature as specified in the ISO/IEC 9796 series [i.179]. ISO/IEC 18370-1 [i.183] also introduces a model of selectively disclosing attributes by using blind signatures.

NOTE 4: Sometimes blind signature schemes leverage ZKPs to ensure the signer that the blindly signed content is well-formed (adheres to some requirements).

Electronic Attestation of Attributes (EAAs): attestation in electronic form that allows the authentication of attributes

NOTE: As defined in the ARF [i.71].

EUDI Wallet Instance: instance of an EUDI Wallet Solution belonging to and which is controlled by a user

NOTE: As defined in the ARF [i.71].

EUDI Wallet Provider: organization, public or private, responsible for the operation of a eIDAS-compliant EUDI Wallet Solution that can be instantiated, e.g. through installation and initialization

NOTE: As defined in the ARF [i.71].

EUDI Wallet Solution: entire product and service owned by an EUDI Wallet Provider, offered to all users of that solution

NOTE 1: As defined in the ARF [i.71].

NOTE 2: An EUDI Wallet solution can be certified as being EUDI-compliant by a CAB.

ISO mobile Driving License (ISO mDL): According to ISO/IEC 18013-5 [i.181] and ISO/IEC CD 18013-7 [i.182].

Issuing Authority Certification Authority (IACA): certification authority in the context of ISO mDL that issues certificates for the creation of ISO mDL MSOs and auxiliary certificates for revocation services or securing online services (such as TLS servers)

issuer: issuing authority that is accredited or supervised for issuing certificates, attested attributes, ISO mDL or credentials

NOTE 1: In the context of eIDAS2, the issuer can be a Person Identification Data Provider issuing PIDs or a (Qualified) Trust Service Provider issuing (Q)EAAs (as defined in the ARF [i.71]).

NOTE 2: In the context of ISO mDL, the issuer is an IACA that issues certificates for the creation and operation of ISO mDL MSOs.

mdoc: ISO mobile document as a credential format that carries information about a user and supports selective disclosure via mechanisms like the MSO

Mobile Security Object (MSO): ISO mobile driving license Mobile Security Object (MSO), with salted attribute hashes of the user's elements in the ISO mDL mdoc

Person Identification Data (PID): set of data enabling the identity of a natural or legal person, or a natural person representing a legal person to be established

NOTE: As defined in the ARF [i.71].

Person Identification Data Provider (PIDP): Member State or legal entity providing Person Identification Data to users

NOTE: As defined in the ARF [i.71].

predicate proof: verifiable computation on attributes or cryptographic meta-data included in a (Q)EAA, where the result of the computation is shared with the relying party without disclosing the claim value itself

EXAMPLE 1: Predicate proofs are often in the form of range proofs (greater than, less than), equal to, set member, etc.

EXAMPLE 2: A user can prove to a verifier that he/she is an EU citizen, without revealing in which Member State.

NOTE: Predicate proofs are often employed in ZKP systems aimed at limiting information disclosure.

Qualified Electronic Attestations of Attributes (QEAAs): Electronic Attestation of Attributes, which is issued by a Qualified Trust Service Provider and meets the requirements laid down in eIDAS Regulation amendment proposal Annex V [i.103]

NOTE: A (Qualified) Electronic Attestation of Attribute is abbreviated as (Q)EAA, and is a collaborative term that is used when either a QEAA or an EAA could be applicable for the context.

Quantum-Safe Cryptography (QSC): cryptographic algorithms (typically public-key algorithms) that are expected to be secure against a cryptanalytic attack by a quantum computer

NOTE 1: NIST conducts a research program [i.210] to identify candidates for QSC algorithms that can be standardized. The signature scheme finalists (December 2023) are FIPS 204 [i.207] (based on CRYSTALS Dilithium [i.75]) and FIPS 205 [i.208] (based on SPHINCS+ [i.238]). Unless stated otherwise, FIPS 204 [i.207] and FIPS 205 [i.208] are referred to as QSC signature schemes throughout the present document.

NOTE 2: The term Post-Quantum Cryptography (PQC) is sometimes used in other literature, and is equivalent to the term Quantum-Safe Cryptography (QSC) that is used throughout the present document.

NOTE 3: The post-quantum world is the era when quantum computers are expected to be capable of breaking asymmetric cryptographic algorithms based on the Discrete Logarithm Problem (DLP) or the difficulty of factoring large composite numbers. Asymmetric cryptographic algorithms that are plausibly vulnerable to such attacks are RSA, SDH, ECDSA, ECSchnorr, etc.

NOTE 4: The pre-quantum world is the era when quantum computers are not (yet) capable of breaking asymmetric cryptographic algorithms based on the DLP or the difficulty of factoring large composite numbers.

NOTE 5: Plausible quantum-safe cryptographic systems, protocols or signature schemes may be implemented either by introducing quantum-safe components, and/or by selecting a quantum-safe signature method like FIPS 204 [i.207] and FIPS 205 [i.208].

range proof: method by which the user (prover) can prove to the relying party (verifier) that a number is in a given range (lower and upper bound) without disclosing the actual number

EXAMPLE: A 21-year-old user can prove to a verifier that he/she is older than 18 years, without revealing their actual age.

NOTE: Range proofs are an example of predicate proofs. A range proof for inclusion in an interval is typically generated by using two inequality tests, one for each boundary.

Selective Disclosure JSON Web Token (SD-JWT): W3C® Verifiable Credential (VC) used in conjunction with a SD-JWT [i.155] with a list of salted hash values of the user's claims in the W3C VC

selective disclosure: capability of the EUDI Wallet that enables the user to present a subset of attributes provided by the PID and/or (Q)EAAs

NOTE 1: As defined in the ARF [i.71].

EXAMPLE: Assume that a user's EUDI Wallet includes a (Q)EAA with the attributes first name, last name, birth date, and address. The user can for example selectively disclose only its first name.

NOTE 2: ISO mdoc (clause 7.2) and IETF SD-JWT (clause 7.3) can present selectively disclosed attributes based on the design of salted attribute hashes.

unlinkability: lack of information required to connect the user's selectively disclosed attributes beyond what is disclosed

NOTE 1: Verifier unlinkable means that one or more verifiers cannot collude to determine if the selectively disclosed attributes describe the same identity subject.

NOTE 2: Issuer unlinkable means that one or more issuers cannot collude to determine if the selectively disclosed attributes describe the same identity subject.

NOTE 3: Fully unlinkable means that no party can collude to determine if the selectively disclosed attributes describe the same identity subject.

NOTE 4: Multi-show unlinkability means that a (Q)EAA can be used for multiple presentations, which cannot be used to connect the user's selectively disclosed attributes.

NOTE 5: The opposite of multi-show unlinkability means that a (Q)EAA can only be used once for a presentation, since the (Q)EAA will thereafter reveal information that can be used for linkability.

EXAMPLE 1: Assume that a user's EUDI Wallet includes a (Q)EAA with the attributes first name and last name. The user can disclose its first name to one relying party, and its last name to another relying party. The relying parties cannot exchange any information that allows them to link the user's first name disclosure to the last name disclosure.

EXAMPLE 2: The same principle applies if the user discloses its first name to a relying party and later discloses its last name to the same relying party and the single relying party cannot link the user's first name disclosure to its last name disclosure.

EXAMPLE 3: The same principle applies if the issuer colludes with the verifier without being able to link the user's first name disclosure to its last name disclosure.

user: natural or legal person using an EUDI Wallet

NOTE 1: As defined in the ARF [i.71].

NOTE 2: In the context of selective disclosure, the user is also the prover of the attributes it presents from its EUDI Wallet.

NOTE 3: The user is sometimes also denoted as holder in other specifications.

Verified Issuer Certificate Authority List (VICAL) provider: ISO mDL provider that can compile, operate and provide trust anchors (such as IACA trust anchors) in the form of a service to mDL participants

Zero-Knowledge Proof (ZKP): method by which the user (prover) can prove to the relying party (verifier) that a given statement is true while the user does not provide any additional information apart from the fact that the statement is true

NOTE 1: There are special-purpose ZKPs that can only prove very specific statements (e.g. knowledge of a pre-image of a hash or knowledge of a signature under a specific digital signature scheme) and general-purpose or programmable ZKPs that allow to prove any statement. Programmable ZKPs usually involve a compiler from some programming language that describes the statement to be proved (e.g. program returns a certain public value upon correct execution on a private input) into a ZKP proving and verification program.

NOTE 2: A ZKP protocol should meet the following three criteria: Completeness (if the statement is true then a user can convince a verifier), soundness (a fraudulent user can not convince a verifier of a false statement beyond negligible probability - how small is a parameter choice, e.g. 2^{-128}), and zero-knowledge (the interaction only reveals if a statement is true and nothing else beyond what can trivially be inferred from the statement itself). The definition of "zero-knowledge" is quite intricate and formalized by means of a simulator. Plastically speaking, the simulator is an algorithm that does not have access to the prover's private information but instead has the capability to "time travel", i.e. to anticipate the verifier's challenges. If the simulator can manage to convince the verifier in this way that the communication with the simulator cannot be from the communication with the prover, it is intuitive that the verifier cannot learn anything about the prover's private information from the transcript, as the simulator does not even have access to this private information.

NOTE 3: A programmable ZKP system supports selective disclosure, unlinkability and predicate proofs and arbitrary predicates per definition, provided the verifier does not specifically ask for all (Q)EAA or linkable data.

EXAMPLE: zk-SNARKs (clause 4.5.2) are examples of programmable ZKP protocols, whereas CL-signatures and BBS+ are examples of special-purpose ZKP protocols.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TR 119 001 [i.91] and the following apply:

3S	Secure Sub-System
AA	Attribute Authority

ABC	Attribute Based Credentials
AIR	Algebraic Intermediate Representation
ARF	Architecture and Reference Framework
ARKG	Asynchronous Remote Key Generation
BBS	Boneh-Boyen-Shacham
BLE	Bluetooth® Low Energy
BLS	Barreto-Lynn-Scott

NOTE: Pairing-friendly elliptic curves.

BIP-32	Bitcoin Improvement Proposal 32
BSI	Bundesamt für Sicherheit in der Informationstechnik
CBOR	Concise Binary Object Representation
CCG	Credentials Community Group
CD	Committee Draft
CDDL	Concise Data Definition Language
CES	Content Extraction Signatures
CFRG	Crypto Forum Research Group
CIR	Commission Implementing Regulation
CL	Camenisch-Lysyanskaya
CLRSA	Camenisch-Lysyanskaya signatures based on RSA
CMS	Cryptographic Message Syntax
COSE	CBOR Object Signing and Encryption
CPT	CBOR Proof Token
CRL	Certificate Revocation List
CRQC	Cryptographically Relevant Quantum Computer
CRYSTALS	Cryptographic Suite for Algebraic Lattices
CS	Computationally Sound
CWT	CBOR Web Tokens
DAA	Direct Anonymous Attestation
DAG	Directed Acyclic Graph
DIF	Digital Identity Foundation
DLP	Discrete Logarithm Problem
DLREP	Discrete Logarithm Representation
dp-ABC	distributed privacy-preserving Attribute Based Credentials
EAA	Electronic Attestation of Attributes
EBA	European Banking Association
ECDL	Elliptic Curve Discrete Logarithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDSA-SD	Elliptic Curve Digital Signature Algorithm with Selective Disclosure
ECSDSA	EC Schnorr DSA
eMRTD	electronic Machine Readable Travel Document
EPID	Enhanced Privacy ID
EUDI	European Union Digital Identity
EUDIW	European Union Digital Identity Wallet
FIDO	Fast IDentity Online
FIPS	Federal Information Processing Standards
FPKIPA	Federal Public Key Infrastructure Policy Authority
FRI	Fast Reed Solomon Interactive oracle proof
G3C	Graph 3-Colouring
HAIP	High Assurance Interoperability Profile
HDK	Hierarchical Deterministic Key
HNDL	Harvest Now Decrypt Later
HSM	Hardware Security Module
IACA	Issuing Authority Certification Authority
ICAO	International Civil Aviation Organization
IDEMIX	Identity Mixer
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IOP	Interactive Oracle Proof
IRTF	Internet Research Task Force
JAdES	JSON Advanced Electronic Signatures

JOSE	JSON Object Signing and Encryption
JPA	JSON Proof Algorithm
JPT	JSON Proof Token
JSON	JavaScript Object Notation
JSON-LD	JSON for Linking Data
JWP	JSON Web Proof
JWK	JSON Web Key
JWS	JSON Web Signature
JWT	JSON Web Token
JWZ	JSON Web Zero-knowledge
KBSS	Key Blinding for Signature Schemes
KDF	Key Distribution Function
k-TAA	k-Times Anonymous Authentication
KVAC	Keyed-Verification Anonymous Credentials
KZG	Kate Zaverucha Goldberg
LLVM	Low Level Virtual Machine
MAC	Message Authentication Code
MAC_BBS	Message Authentication Code based Boneh-Boyen-Shacham signatures
mDL	mobile Driving License
MSO	Mobile Security Object
NCCoE	National Cybersecurity Center of Excellence
NP	Nondeterministic Polynomial-time
NTRU	Number Theory Research Unit
OCSP	Online Certificate Status Protocol
OID4VC	OpenID for Verifiable Credentials
OID4VCI	OpenID for Verifiable Credentials Issuance
OID4VP	OpenID for Verifiable Presentations
OIDC	OpenID Connect
p-ABC	privacy-preserving Attribute Based Credentials
PCP	Probabilistically Checkable Proofs
PCS	Polynomial Commitment Scheme
PID	Person Identification Data
PIDP	Person Identification Data Provider
PII	Personal Identifiable Information
PIOP	Polynomial Interactive Oracle Proof
PIR	Private Information Retrieval
PKD	Public Key Directory
PKIX	Public-Key Infrastructure (X.509)
PQC	Post-Quantum Cryptography
PSD2	Payment Services Directive v2
PSI	Private Set Intersection
PS-GS	Pointcheval-Sanders Group Signatures
PS-MS	Pointcheval-Sanders Multi-Signatures
PWI	Preliminary Work Item
QAP	Quadratic Arithmetic Program
QEAA	Qualified Electronic Attestation of Attributes
QKD	Quantum Key Distribution
QMA	Quantum Merlin Arthur
QSC	Quantum-Safe Cryptography
qSDH	q-Strong Diffie-Hellman
QTSP	Qualified Trust Service Provider
QWAC	Qualified Website Authentication Certificate
RDF	Resource Description Framework
RL	Revocation List
ROM	Random Oracle Model
ROS	Random inhomogeneities in a Overdetermined Solvable system of linear equations
RSAREP	RSA Representation
RTS	Regulatory Technical Standard
R1CS	Rank-1 Constraint System
SAID	Self-Addressing IDentifier
SD	Selective Disclosure
SDH	Strong Diffie-Hellman

SD-JWT	Selective Disclosure JSON Web Token
SECDSA	Split-ECDSA
SEP	Signatures with Efficient Protocols
SIOP2	Self-Issued OpenID Provider v2
SL	Status List
SLIPS	SatoshiLabs Improvement Proposals
SMT	Sparse Merkle Tree
SoC	System on Chip
SOG-IS	Senior Officials Group Information Systems Security
SSP	Square Span Program
TCG	Trusted Computing Group
TLS	Transport Layer Security
TPM	Trusted Platform Module
UI	User Interface
UUID	Universal Unique IDentifier
VC	Verifiable Credential
VCDI	Verifiable Credential Data Integrity
VCDM	Verifiable Credential Data Model
VDR	Verifiable Data Registry
VICAL	Verified Issuer Certificate Authority List
VP	Verifiable Presentation
W3C	World Wide Web Consortium
WG	Working Group
XAdES	XML Advanced Electronic Signatures
YAML	Yet Another Multicolumn Layout
ZKARG	Zero-Knowledge Argument
ZKP	Zero-Knowledge Proof
zk-SNARK	zero-knowledge Succinct Non-Interactive Argument of Knowledge
zk-STARK	zero-knowledge Scalable Transparent Argument of Knowledge
zkVM	zero-knowledge Virtual Machine

4 Selective disclosure signature schemes

4.1 General

The present clause provides an analysis of a set of selective disclosure signature schemes.

The topics for the analysis of each selective disclosure signature scheme are:

- Underlying cryptographic algorithms for selective disclosure, unlinkability and optionally ZKP.
- Maturity of the selective disclosure signature scheme's specification and deployment.
- Cryptographic aspects, more specifically if the cryptographic algorithms used for the selective disclosure signature schemes are approved by SOG-IS and allows for QSC algorithms for future use.

There exist four main categories to enable selective disclosure:

- The first category is using atomic (Q)EAAs, which is described in clause 4.2.
- The second category is signing a collection of salted attribute digests; this category is described in clause 4.3.
- The third category is using a selective disclosure capable multi-message signature scheme, which typically relies on commitments. This category is explained in clause 4.4.
- There is also a fourth category of methods that can ensure the privacy of any computable proof (e.g. Bulletproofs, zk-SNARKS, zk-STARKS, etc.). This category is elaborated in clause 4.5. These methods could support additional selective disclosure mechanisms beyond the three main ones listed above.

NOTE: An argument can be made for a selective disclosure mechanism that relies on trusted components for storage and computation. It is possible to store unsigned attribute claims on trusted storage and transport only the requested claims over a secure messaging channel. It is also possible in these setups to associate each storage partition with a unique key and only store a single (Q)EAA per partition in order to ensure the proper pairing of attributes. A solution based on these principles is detailed in BSI TR-03110 [i.36]. The solutions described in the present document, however, include only signature based selective disclosure schemes.

Each of the four main ways are described in clauses 4.2, 4.3, 4.4 and 4.5.

4.2 Atomic (Q)EAAs schemes

An atomic Electronic Attribute Attestation is a (Q)EAA with a single attribute claim, which can be issued by a (Q)TSP upon request or as part of a batch to an EUDI Wallet. The atomic (Q)EAAs can be selected by the user and be included in a verifiable presentation that is presented to a verifier.

An example of a solution based on atomic (Q)EAAs is illustrated in Figure 2. In this scenario, the user needs a parking ticket to enter a car parking. For that purpose, the user enrols for atomic (Q)EAAs from a transport authority (with the car registration number), from a civil registry (with the address), and from a payment service provider (with the paid amount). The user's EUDI Wallet can then combine these atomic (Q)EAAs into a verifiable presentation, which is the parking ticket that is presented to the car parking clerk.

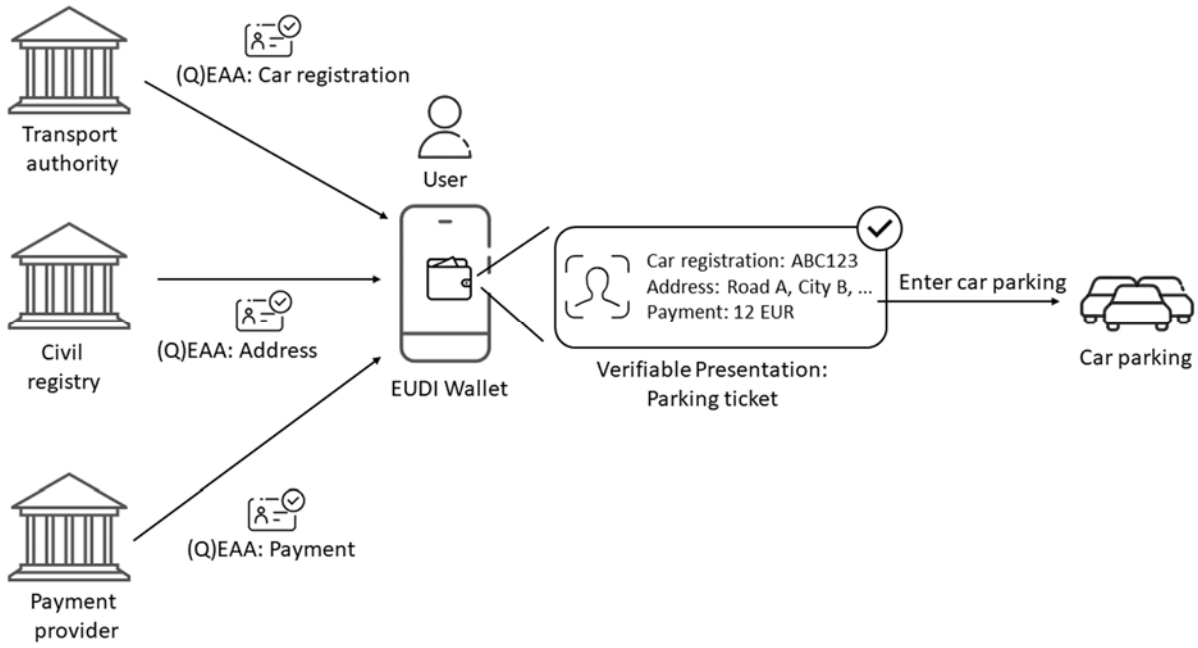


Figure 2: Example of atomic attribute credentials

The underlying cryptographic algorithms depend on the (Q)TSPs' signing algorithms of the (Q)EAAs and the proof key when signing the verifiable presentation. Hence, it is possible to select signature algorithms that are approved by SOG-IS and/or allow for QSC. (More information on the specific (Q)EAA formats X.509 attribute certificates and W3C Verifiable Credentials is available in clauses 5.2.1 and 5.2.2).

By enrolling for atomic (Q)EAAs on demand it is possible to achieve verifier unlinkable attestations which results in an unused set of (Q)EAAs with new signatures that cannot be correlated with any previous signatures. Fully unlinkable (Q)EAAs are, however, not possible.

NOTE 1: If the atomic (Q)EAAs are issued batchwise to an EUDI Wallet, it is recommended to keep track of the atomic (Q)EAAs that have been used for presentations, and replace them with new atomic (Q)EAAs.

NOTE 2: Atomic attribute credentials cannot alone guarantee that the claims are paired properly in a presentation. For instance, if the user has a credential from the civil registry with an address, and one for their company they are the legal representative of, there is nothing preventing the user from creating a presentation that improperly pairs the company's address with the user's private car registration. Verifiers cannot trust that verifiable presentations containing multiple atomic attribute credentials are properly paired without additional mechanisms preventing improper pairing.

4.3 Salted attribute hashes

4.3.1 Overview of salted attribute hashes

Salted attribute hashes are a widely deployed concept in many solutions capable of selective disclosure. The salted hash approach computes a cryptographic digest over at least one attribute and an attribute specific random salt, e.g. a SHA256 digest over a concatenation of a salt and an attribute, SHA256 (salt||attribute).

In the context of a (Q)EAA, each attribute is salted and a hash digest is included as a value in the attestation. The specific way to include the digest in the attestation varies between various solutions. Some include salted attribute hashes in an indexed list, others in an array, others structure these as a Directed Acyclic Graph (DAG). Common to all is that the issuer needs to issue the (Q)EAA with the attributes in clear text, along with the logical ordering of salted attribute hashes.

An illustrative example of salted attribute hashes is illustrated in Figure 3.

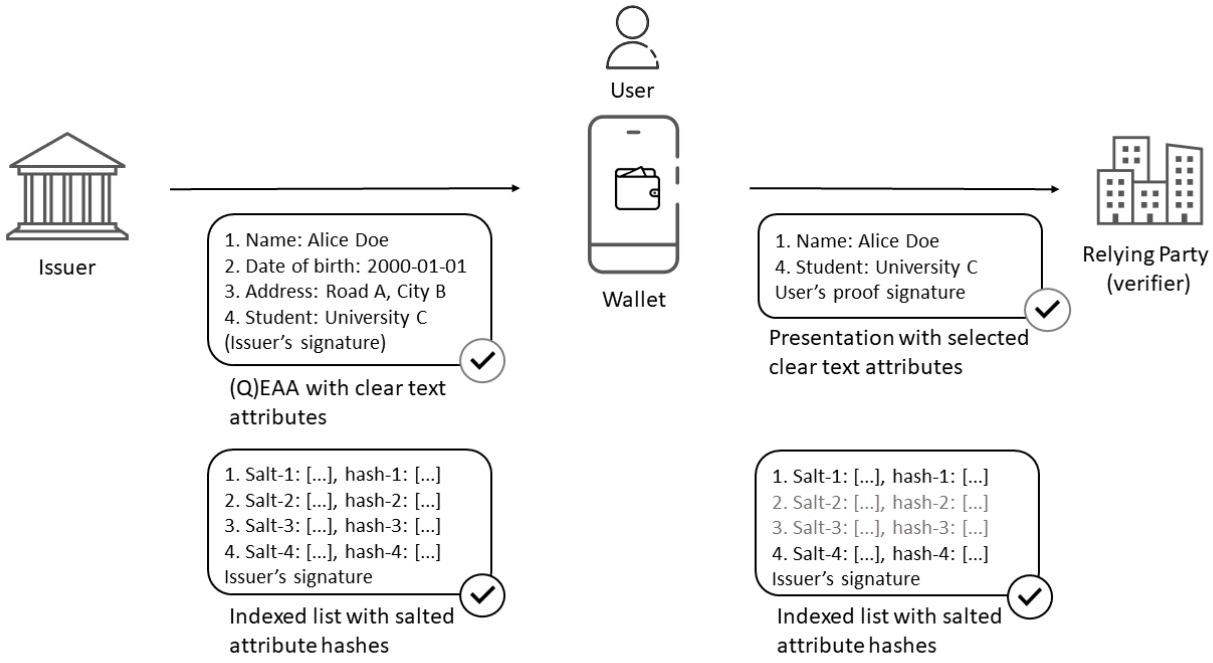


Figure 3: Illustrative example of salted attribute hashes

In the example above, the issuer issues a (Q)EAA with all attributes in clear text. The issuer also issues an indexed hash list in which each (Q)EAA attribute is represented as a key (index), a random salt, and a hash value over the salt and attribute. The (Q)EAA and indexed hash list are signed by the issuer.

NOTE 1: Exactly how the random salts are combined with the attributes and hashed, and how the lists of salted attributes hashes are signed by the issuer, differs between various specifications and standards. The relevant standards that are described and analysed in the present document are ISO mDL MSO (see clause 7.2) and IETF SD-JWT (see clause 7.3).

NOTE 2: The salts may be included in the indexed list with salted attribute hashes, or be provided separately from the indexed list. If the salts are provided separately (on a need to know basis) to the verifier, this is the most privacy preserving option.

NOTE 3: The (Q)EAA can be either signed or unsigned depending on the technical standard.

EXAMPLE 1: The mdoc (with the attributes) is unsigned, whilst the corresponding MSO (with the salted attribute hashes) is signed by the issuer.

EXAMPLE 2: The W3C Verifiable Credentials (with the attributes) is signed, and the corresponding IETF SD-JWT (with the salted attribute hashes) is also signed.

The (Q)EAA and indexed hash list are stored in the user's wallet. The user selects the attributes to disclose to a relying party, and the wallet generates a presentation with the disclosed attributes; the user signs the presentation with its proof key.

The wallet submits the presentation with selected attributes (in clear text) along with the indexed hash list. The relying party parses out the salted hashes from the indexed hash list, and compares them with the salted hashes of the presented attributes. Unique salts prevent cross-verifier linkage using digests but enables issuer-level tracking.

Solutions based on the concept of salted attribute hashes have been standardized as IETF SD-JWT and ISO mDL MSO. More information on the specific formats IETF SD-JWT and ISO mDL MSO that use salted attribute hashes for selective disclosure is available in clauses 5.3.2 and 5.3.3.

4.3.2 Issuance phase

The issuance phase of this selective disclosure scheme is in principle based on the following algorithm:

1) Parse out each attribute from a user's (Q)EAA.

2) Concatenate each attribute set with a salt, denoted as (salt||attribute).

3) Hash each (salt||attribute), denoted as hash(salt||attribute).

4) Order all the hash(salt||attribute) values and the salts in e.g. an indexed hash list (could also be an array, DAG etc.), which is signed. The indexed hash list can be expressed as this formula: signed({key-1, salt-1, hash(salt-1||attribute-1)}, … {key-*n*, salt-*n*, hash(salt-*n*||attribute-*n*)}).

5) Store the (Q)EAA in an EUDI Wallet along with the indexed list from step 4.

NOTE 1: The hash algorithm used in step 3 should be listed in the SOG-IS list of approved hash algorithms [i.237], such as SHA-256 or higher.

NOTE 2: The signature algorithm used in step 4 should be listed in the SOG-IS list of approved signature algorithms [i.237], such as ECDSA with BrainpoolP256r1.

NOTE 3: The signature format used in step 4 should allow for QSC algorithms. For example, JOSE and COSE allows for QSC algorithms.

NOTE 4: Salted hash digest-based attestations are inherently verifier-unlinkable. Issuer unlinkability can be added using blind signatures: the user generates *N* blinded attestations, the issuer verifies openings for *N-1*, and signs the last if all openings are valid. This probabilistic method is impractical in practice.

4.3.3 Presentation and verification phase

When presenting selective disclosed attributes in the (Q)EAA along with the indexed list, the relying party can perform the following verification process:

1) The EUDI Wallet parses out the disclosed attribute with key-*x* from the (Q)EAA.

2) The EUDI Wallet submits the disclosed (Q)EAA attribute with key-*x* from step 1 along with the indexed hash list to the relying party. The indexed hash list has the format: signed({key-1, salt-1, hash(salt-1||attribute-1)}, … {key-*n*, salt-*n*, hash(salt-*n*||attribute-*n*)}).

3) The relying party verifies the signature of the indexed hash list from step 2. If the signature check fails, the verification process is stopped, else it continues at step 4.

4) The relying party parses out salt-*x* from the indexed hash list.

5) The relying party parses out hash(salt-*x*||attribute-*x*) from the indexed hash list.

6) The relying party concatenates the disclosed (Q)EAA attribute from step 2 with the corresponding salt-*x* from step 4, and hashes the result.

7) The relying party checks if the result in step 6 is equal to the hash(salt-*x*||attribute-*x*) from step 5. If the values match, the verification process has succeeded.

4.3.4 Salted attribute hashes and unlinkability

4.3.4.1 General criteria of unlinkability for salted attribute hashes and associated challenges

Salted-hash approaches are typically used with traditional digital signature schemes with inherent linkability risks. Issuers,verifiers, and third parties, can link disclosures and attestations through signature values or any salt. Workarounds can offer verifier unlinkability at added cost for issuers.

Verifier unlinkability requires two main criteria. First, each salt has to be randomly generated, unique, and presented only once. Second, each attestation has to include a distinct public key that is used only once during Proof of Possession (PoP).

NOTE 1: An issuer can identify a user from a single unique salt or public key. Thus, salted attribute hashes prevent verifier linkability only and should only be used when issuers are assumed to be honest (i.e. follow protocol and do not attempt to learn more than allowed).

NOTE 2: Issuer unlinkable attestations using salted hashes and conventional cryptography are unpractical. Using a cut-and-choose approach, a user would generate *N* attestations and create a commitment to each, then the issuer randomly picks *N-1* to unblind and verify before signing the remaining one. Assuming users are malicious makes it difficult to optimize the cut-and-choose.

With conventional cryptography, distinct public keys per attestation require either request-based issuance (i.e. when needed) or batch issuance (many single-use keys issued simultaneously). This amplifies two challenges:

1) Key management. Secure hardware has to manage multiple single-use keys.

2) Proof of Association (PoA). Users have to be able to prove that distinct keys that appear unrelated to a third party, are in fact associated; potentially tied to the same secure hardware.

Key management can be addressed by either (i) generating fresh keys for each attestation, or (ii) deriving new keys through key derivation functions. Fresh key generation is straightforward: the user's device generates the required key pairs, communicating each public key to the issuer (although PoA is a lot more challenging as will be discussed later). Key derivation requires further elaboration.

Key Derivation Functions (KDFs) are designed to derive child keys that appear cryptographically unrelated to an external party. Using a KDF enables a user to securely derive keys from a single hardware-protected seed value. The key derivation can be either local (requiring only user input) or remote (requiring both user and issuer input). The remote case, and its suitability for a wallet context, is prominently exemplified by BIP-32 [i.25] and SLIP-0010 [i.236] (both used to deterministically derive new keys for cryptocurrency wallets). Although informative, BIP-32 and SLIP-0010 require adaptation for the EUDIW context due to differing derivation paths, input requirements, and issuer-user interactions.

Key derivation specifications suitable for PID/(Q)EAAs are under development. One notable example is the IETF individual draft "The Asynchronous Remote Key Generation (ARKG) algorithm" [i.146]. The ARKG draft describes suitable algorithms, actors, and interactions. It is possible to create ARKG applications bespoke for the EUDIW context, with one example being Hierarchical Deterministic Keys (HDKs).

Next, ARKG and HDK are described, followed by a discussion on PoP and PoA for single-show attestations.

4.3.4.2 The Asynchronous Remote Key Generation (ARKG) algorithm

The ARKG is an abstract algorithm enabling key derivation. The specification details:

1) A generic key derivation algorithm using abstract primitives. Detailed in section 2 of [i.47], this allows for the derivation of child keys using a parent key.

2) Concrete instantiations of abstract primitives. Detailed in section 3 of [i.47], it details generic formulas for instantiating individual ARKG parameters used to define concrete ARKG instantiations.

3) An initial set of fully specified concrete ARKG instances in the subsequent sections.

The ARKG specification requires two parameters: (i) an asymmetric Key Blinding (BL) scheme, and (ii) a Key Encapsulation Mechanism (KEM). The BL enables the derivation of child keys from a parent key. The KEM contributes entropy from multiple parties into the derivation function and ensures that the derived keys appear unrelated to any third party.

Child public keys can be derived by any party in possession of a parent public key without requiring access to the corresponding private key. The derivation can be done asynchronously in an unsecured environment following initial input from the private key holder. Assuming that the parent private key is secured, a child private key can only be derived by the party who controls the parent private key.

The party in control of the parent private key is called the *delegating* party, and the party who performs the public key derivation is called the *subordinate* party. These can, but are not required to, be two different entities (e.g. the delegating party may be a device's secure hardware and the subordinate party may be an application running on the same device). The two interact over three procedures:

1) Initialization. The delegating party, i.e. the parent private key holder, generates a seed pair consisting of a private seed and a public seed. The private seed is a tuple of a private Key Encapsulation Mechanism (KEM) key and a private parent key from which child private keys are derived. The public seed is a tuple of a public KEM key and a public parent key from which child parent keys are derived.

2) Public key derivation. The subordinate party uses the public seed to derive both child public keys and a key handle. Optionally, the subordinate party also enables a remote third party to derive the child public keys.

3) Private key derivation. The delegating party uses the key handle from 2) and the private seed to derive the corresponding child private keys.

The three procedures essentially use a KEM to create a shared secret between the delegating party and the subordinate party, where the shared secret is input to a function that derives a blinding factor, $\boldsymbol{\tau}$. This blinding factor is then combined (by leveraging some homomorphism that exists) with both the parent public key, K_P, and the parent private key, k_P, yielding a blinded version of each.

The function deriving the blind factor can take as input a key index, allowing the derivation of multiple child keys from a single parent key, with all keys appearing unrelated to a third party. Keys appear unrelated because a third party faces a decomposition problem when seeing the derived key, $K_{C1} = K_P + [\boldsymbol{\tau}_1]G$. Learning about the relationship requires: (i) knowledge of either K_p or $\boldsymbol{\tau}_1 G$, (ii) correctly guessing that the issuer derived the key and how the key was derived, and (iii) finding a colluding party who has seen K_P. By itself, knowledge of K_p and $\boldsymbol{\tau}_1 G$ does not reveal a cryptographic relation beyond what is already known (i.e. that any point can be expressed as the sum of other points).

With the concrete ARKG instantiation ARKG-P256ADD-ECDH, $K_{C1} = K_P + [\boldsymbol{\tau}_1]G$ is the BL scheme EC point addition with $\boldsymbol{\tau}$ being a random element in the EC scalar field computed using a hash_to_field function as specified in IETF RFC 9380 [i.172]. In turn, the hash function input is computed using a ECDH based KEM. The generic ARKG algorithm is modular and allows for substitution of either the BL scheme or the KEM, provided they are secure.

The ARKG algorithm is useful in multiple use cases, one being generating single-show asymmetric keys for PIDs/(Q)EAAs. Here, the Hierarchical Deterministic Key (HDK) individual draft specification [i.176] aims to detail how ARKG can be leveraged to enable both local and remote derivation of single-use keys. In HDK, both the local and remote derivation implements the delegating party in such a way that the secure hardware manages and protects the seed's BL (non-exportable) private key. The device's software implements the ARKG subordinate party, and additional functionality if the secure hardware does not natively support ARKG.

For remote derivation, the issuer determines the key index, *i = [0, 1, …, n]*, which is input to the hash_to_field function (together with the parent public key) yielding the blinding factor τ_i. The seed's BL keypair may be blinded in multiple iterations, each using a different *i*. Local derivation is a special case of remote derivation where the key index is determined locally in software and requires no further explanation.

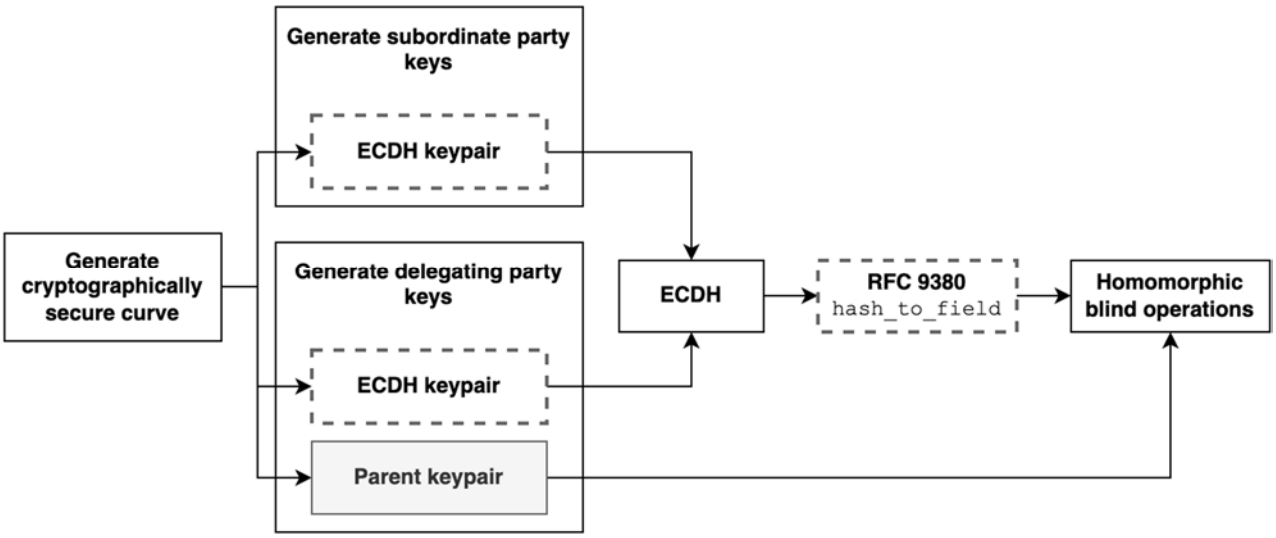
Using the concrete instantiation ARKG-P256ADD-ECDH as an illustrative example, the process can be depicted as shown in Figure 4. Here, the BL leverages the homomorphism that exists between EC point addition, and scalar addition. Specifically, child keys are derived by adding a blinding factor to the parent keys.

NOTE: There are two non-exclusive ways to ensure unique child keys:

i) in the KEM step, vary the ECDH shared secret by the KEM recipient, sender, or both, using ephemeral ECDH key pairs upon each invocation; and

ii) in the BL step, vary the info parameter in the hash_to_field. In ARKG, the main approach is to rely on ephemeral invocation specific ECDH keypairs.

Relatedly, the way HDK applies ARKG is to change the info parameter to enable repeated derivation of unique child keys using a single ECDH keypair. Both result in child keys that appear unrelated to third parties.



NOTE: Dotted boxes are ways to ensure that the child keypair appears unrelated to the parent key pair.

Figure 4: Illustrative example of key derivation building blocks using ARKG-P256-ECDH

Having discussed the key management challenge of batch issuance, the PoP and PoA challenge will be described next.

4.3.4.3 Batch issuance and Proof of Possession / Association

Users have to be able to generate a PoP for any public key used in an attestation, including multiple single-use keys. During batch issuance, the issuer has to be certain that the public keys are associated, e.g. protected by an individual piece of secure hardware. While PoP is straightforward, PoA requires further consideration.

There are multiple general PoA types including:

1) Session-based PoA.

2) Claim-based PoA.

3) Signature-based PoA.

4) Related-key PoA.

Session-based PoA is principally based on hardware protection of attestations (possibly unsigned) where the association is ensured through a secure session. A session-based PoA is arguably of limited value for the EUDIW context and will not be discussed further here.

Claim-based PoA relies on equivalence proofs between claims. For example, an attribute set (e.g. a single unique identifier or claims composition) may serve as PoA input. To ensure verifier unlinkability, the claims have to be blinded to appear unrelated to colluding verifiers. Accordingly, each attestation includes a blinded claim statement, and the user may reveal these - with an optional software-generated PoA - only when explicitly required. Suitable techniques include Discrete Log Equivalence proofs, such as a Chaum-Pedersen proof [i.61].

NOTE 1: User-to-issuer PoA proofs can be simpler than user-to-verifier PoA proofs, particularly for identified users. Issuers can confirm key association via multiple methods, and verifiers who trust this process may not need to verify PoA on their own.

NOTE 2: Requiring verifiers to validate PoA raises privacy concerns, as it reveals associations between attestations. For example, a user might not wish to disclose that their medical prescription is linked to their PID. Appendix A of Altmann's "A third party repudiable ZKP-based PoA" [i.5] illustrates generating third-party repudiable PoA using an interactive Chaum-Pedersen ZKP.

Signature-based PoA employs at least two keypairs to prove their association. It relies on certified hardware to enforce signing only when the hardware controls both keys. Options for EUDIW adaptation include countersignatures, cross-signing, and multi-signature schemes. However, any asymmetric signature-based PoA is inherently linkable, creating a non-repudiable link between attestations.

NOTE: Alternatively, a symmetric PoP is possible using e.g. ECDH-MAC.

Related-key PoA exploits the homomorphism between private and public keys, e.g. private scalars and public EC points. Two approaches have been developed for the EUDIW context:

1) **Private key ratios**. Verheul "Attestation Proof of Association" [i.245] proposes computing an association key, z, as the ratio of two private key scalars, (p_1, p_2), i.e. $z = p_2 / p_1$. Since z's multiplicative structure is preserved in the public key domain, a PoA is possible (see Algorithms 1 and 2 in Verheul's paper [i.245]).

2) **Derived associations**. ARKG enables implicit association through a homomorphic blind operation on an existing public key, ensuring only the controller of the parent private key can derive the child private key. While not required, explicit association proofs are possible using a DLEQ proof like Chaum-Pedersen.

NOTE 1: Related-key PoA are actively debated and have known security risks and require appropriate safeguards (see [i.148]). There are also potential patent concerns (see [i.121]). They thus require careful and extensive examination.

NOTE 2: The above two approaches are not the only ones possible, but are two that specifically target PoA in the EUDIW context.

4.3.5 Cryptographic analysis

The (Q)EAA and indexed hash list are separate objects that can be signed with cryptographic algorithms that are approved by SOG-IS [i.237]. In other words, there are no specific requirements on ECC curves for bilinear pairings.

This concept also caters for the (Q)EAA and indexed hash list to be signed in the future with QSC algorithms as discussed in the IETF report "JOSE and COSE Encoding for Post-Quantum Signatures" [i.149].

The security proof for ARKG is available in Frymann et al. [i.114] and Frymann, Gardham and Manulis [i.115] and are proven to produce unforgeable signatures for challenge-response protocols.

The security of DLEQ proofs requires publicly verifiable random base points; when used in attestation PoAs, verifiers need to use issuer signed DLEQ inputs.

4.3.6 Predicates based on computational inputs

Salted attribute hashes do not inherently support dynamic calculation of predicates (e.g. to compute a proof for age over 18 given only the birth date and current date). A current approach is to include Boolean claims such as `"age_over_NN": "True"`. However, using static age variables are problematic for several reasons, including, but not limited, to:

1) Requires tight issuance timing to be valid. A claim like `age_over_18: False` may flip to `True` the next day. This creates pressure to issue credentials with precise timing and may require validity metadata like not-before values. However, exposing issuance or validity windows can indirectly leak the user's date of birth, undermining privacy.

2) Vulnerable to insecure verifier implementations. A user who presents an `age_over_NN: False` claim may be above the age NN at the time of presentation. Verifiers have to check timestamps carefully in addition to validating the attestation and processing the attributes.

3) Inflexible - requires pre-issuance of many claims. Since verifiers may ask for different thresholds (e.g. `age_over_18`, `age_over_21`, `age_over_65`), the credential has to either preemptively include a large set of `age_over_NN` claims or be reissued for each new use case. This bloats the credential and introduces unnecessary dependence on the issuer.

4) Unclear and cumbersome support for age ranges. Proving that a user's age lies within a specific range (e.g. $18 < v < 65$) is not straightforward with static `age_over_NN` claims as it is non-trivial to check for the absence of a claim. Alternatively, the issuer would have to include all `age_over_NN` values that are false too (which leads to issuance timing problems as detailed in 1).

5) The granularity of years is not fine enough. The use of whole-year thresholds (`age_over_NN`) is too coarse for a value that can change daily. A credential that flips from invalid to valid overnight has to be anchored with finer precision, which `age_over_NN` cannot express.

One alternative is for the issuer to sign the parameters and the inputs to an inequality test. While not solving all the aforementioned issues with `age_over_NN`, this would enable the user and the verifier to compare numbers and perform range proofs. For an (Q)EAA system, there is normally a) a trusted issuer, and b) a limited need to perform operations between hashed values.

It is normally interesting to prove that an attribute claim satisfies a threshold or inequality and nothing else. Furthermore, there is a trusted issuer and there is also only the need to hide the exact amount of the values. Thus, the ZKP property may not be strictly necessary.

To prove that an attribute claim satisfies a threshold or inequality, it is necessary to transform that problem into one that is easier to privacy preserve. Many such transformations exist, with three examples detailed below.

EXAMPLE 1: Transforming the inequality test to a find pre-image problem. The issuer could compute the digest s = H (seed) and assign this to the user's birth year. The issuer then computes the chain $c = H^k (salt | s)$, which is k repeated iterations of H. The value for k can be computed e.g. based on the maximum year supported in the calculation. The issuer shares s and includes c in the signed attestation both as disclosures (the user should never reveal s, only c). The user can now generate an age over 18 proof by constructing a hash chain where the length of the chain equals the k iterations used to arrive at the signed commitment c if and only if the user is above a certain age. Example code is provided in Appendix B. Research on efficient protocols for hash chain based range proofs is underway with one notable example being HashWires [i.271]. And variations of the technique exist that would allow a user to generate a valid age_over_N proof from an age_over_M proof where M > N. The algorithm for HashWires in combination with salted attribute hashes is described in clause B.1.

EXAMPLE 2: Transforming a range proof to an inner product proof. The issuer could compute the commitment $\mathbf{V} = v\mathbf{G} + \gamma\mathbf{B}$ where v is the integer value, γ is a blind and (**G, B**) a pair of EC points with unknown discrete logarithm relationships. The user can then use a Bulletproof inner product ZKP to generate a range proof for the lower value *l* and upper value *u* (i.e. $l \leq v < u$). This can be done by showing that (a) the value *v*-*l* fits in a predetermined n-bit range $[0, 2^n)$, and (b) the value *u*-*v* fits in the same range. Put differently, the user proves that the difference between their age and the lower bound is non-negative, and that the difference between the upper bound and their age is also non-negative. Together, these two conditions establish that the user's age lies within the specified range, without revealing the actual age.

EXAMPLE 3: Transforming the inequality test to a set intersection cardinality proof. Using Lin-Tzeng 0-encoding and 1-encoding, it is possible to encode binary representations of values into sets. The intersection of these sets determines the result of the inequality. Specifically, using a Private Set Intersection Cardinality (PSI-CA) protocol, such as Epione, enables computing the comparison in a privacy-preserving way.

Equipped with a privacy-preserving method for comparing two values, constructing range proofs becomes straightforward. A range check such as $a < x < b$ can be expressed as two separate inequality evaluations - each of which can be carried out privately using techniques like hash chains, set intersection cardinality, or ZKPs like Bulletproof inner product proofs.

More generally, many privacy-preserving protocols follow a two-step model: (1) a commitment to a hidden value, and (2) a proof or evaluation over that commitment, alternatively revealing non-sensitive inputs. In the context of verifiable credentials, the issuer can include the commitment as a signed attribute within the attestation, allowing the holder to later prove statements about the hidden value without revealing the value itself.

Hashwires, an approach using hash chains is detailed in clause 4.3.7.

4.3.7 HashWires

4.3.7.1 Introduction

In their 2021 paper "HashWires: Hyperefficient Credential-Based Range Proofs", Chalkias et al. [i.58] present a hash based protocol for performing inequality tests (and by extension range proofs) in contexts where a trusted issuer can sign commitments to computational inputs. The computational inputs in HashWires are a commitment c to a hash chain, and the parameter is the hashing algorithm used to create the chain.

HashWires are inherently less flexible than general ZKP inequality tests and range proofs, and do not support homomorphic operations on commitments. However, the commitment and proof conditions, together with the adversarial assumptions in their deployed contexts (e.g. cryptocurrencies), often makes ZKP inequality tests and range proofs unsuitable for resource constrained environments and unnecessarily complex given the presence of a trusted PID/(Q)EAA Provider (as opposed to self signed claims). Put differently, many existing ZKP inequality tests and range proofs were designed to cater for highly adversarial cryptocurrency contexts without any trusted parties or central authorities, and where the user self issues a signed intent to perform a certain transaction. In contrast, HashWires were designed to specifically cater for the needs of the issuer-holder-verifier model. The authors introduce the concept of "Credential-based range proofs" to distinguish these inequality tests and range proofs from their ZKP counterparts.

HashWires is based on the core idea that the trusted third party, i.e. the PID/(Q)EAA Provider, generates and signs the commitment needed for an inequality test. The idea to rely on a trusted third party to sign a commitment can be traced back to Rivest and Shamir's 1996 work on micro-payments. In their paper "PayWord and MicroMint: Two simple micropayment schemes" [i.229], Rivest and Shamir describe how issuer signed hash chains type commitments can be used for payments. A description of their original idea follows in clause B.1.

4.3.7.2 Cryptographic analysis of HashWires

HashWires are considered as plausible quantum safe since they are based on hash chains. If the used hash functions are designed as QSC, the HashWires scheme becomes quantum-safe.

Since the HashWires scheme is based on chained salted attribute hashes, it can be designed to be unlinkable for verifier(s) collusion, but is not fully unlinkable (see clause 4.3.4).

4.3.8 Authentic Chained Data Containers (ACDCs)

Authentic Chained Data Containers (ACDCs) are verifiable data structures designed to cater for (Q)EAAs with selective disclosure requirements based on Directed Acyclic Graphs (DAGs). While a detailed account of ACDC would require describing a suite of related specifications and standards (that cover key management topics, identifier systems, protocols for introduction and exchange, encoding, proofs, schemas, and the use of various event logs), the text herein focuses on the selective disclosure mechanism that are detailed in the IETF ACDC draft specification [i.142], more specifically in sections 2, 5 and 13.

Every salted attribute hash based approach relies on some form of logical ordering or structuring of the salted attributes that are included in an attestation. In ACDC, that structure is a Directed Acyclic Graph (DAG), where a knowledge graph expresses the attributes of the identity subject. A user may disclose various parts of such a graph, e.g. a vertex identifier, without disclosing any attribute values contained in the vertex, and/or the entire vertex.

The IETF ACDC draft specification [i.142] offers multiple different, but closely related, disclosure mechanisms. To understand these mechanisms it is helpful to distinguish between mechanisms that offer contractual protection of the disclosure (i.e. mechanisms that detail permissions), and mechanisms that are primarily technical in nature (i.e. mechanisms that allow the recipient to obtain the plaintext attribute).

In ACDC, the contractual mechanisms can be expressed in legal terms as the value to a key, "l". This allows the user to specify certain terms and conditions associated with a potential disclosure of attributes, and the ACDC can present a set of such contractual terms under its rule attribute, "r". These mechanisms are not in place to enable disclosures of data for privacy purposes, but instead to protect the identity subject from the unauthorized exploitation of the disclosed data. While essential for a comprehensive grasp of ACDC's contributions, the intricate details of its contractual mechanisms are beyond the scope of the present document. Interested parties should refer to sections 2 and 5 of the IETF ACDC draft specification [i.142] for a comprehensive examination. Of particular relevance herein is that these contractual agreements are designed to be both machine-readable and cryptographically verifiable, and that they play a role in interactions where disclosures are successive and depend on agreements that enable yet additional disclosures.

The IETF ACDC draft specification [i.142] outlines several technical mechanisms to enable sharing only the minimum amount of information about the identity subject that the verifier needs. These mechanisms do not represent different selective disclosure techniques; rather they detail what of the DAG is revealed to a verifier. Three options are detailed:

1) The verifier obtains only a cryptographic digest of a set of key value pairs. These digests are referred to as "compact disclosures". These can be considered as a type of cryptographic commitment to a future disclosure.

2) The verifier obtains a set of key value pairs, and this disclosure contains correlatable information. This mechanism is referred to as "partial disclosure".

3) The verifier obtains a set of key value pairs, and this disclosure is not correlatable to any other yet undisclosed but disclosable key value pair. This mechanism is referred to as "selective disclosure".

Option 1 is used to enable Options 2 and 3. Option 2 is closely linked with successive disclosures where a user can disclose information over time following the acceptance of contractual agreements (e.g. first a commitment, then a schema, then a full disclosure of all attributes in a particular attestation). In contrast, Option 3 allows a user to disclose only a subset of key value pairs without any correlation handles such as an issuer signature over the entire salted attribute hash set. The ability of Option 3 to do so in turn relates to the DAG structure of ACDC and how an ACDC compliant attestation needs to be understood as a graph (section 4 of the IETF ACDC draft specification [i.142] provides additional details).

The content of an ACDC depends on its particular type, but for the purposes of explaining the selective disclosure mechanism employed the following example of a so called "private compact" variant is used with two properties important for understanding selective disclosure highlighted in green:


```
{
  "v":"ACDC10JSON00011c_",
  "d":"EBdXt3gIXOf2BBWNHdSXCJnFJL5OuQPyM5K0neuniccM",
  "u":"0ANghkDaG7OY1wjaDAE0qHcg",
  "i":"did:keri:EmkPreYpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPM",
  "ri":"did:keri:EymRy7xMwsxUelUauaXtMxTfPAMPAI6FkekwlOjkggt",
  "s":"E46jrVPTzlSkUPqGGeIZ8a8FWS7a6s4reAXRZOkogZ2A",
  "a":"EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
  "e":"ERH3dCdoFOLe71iheqcywJcnjtJtQIYPvAu6DZIl3MOA",
  "r":"Ee71iheqcywJcnjtJtQIYPvAu6DZIl3MORH3dCdoFOLB"
```

```
}
```

The example is private because it contains a property "u", which is a unique high entropy unique salt. This salt effectively blinds the digest commitment to the ACDC so that an entity cannot derive any of an ACDC's content knowing only its identifier (i.e. the value of "d", which is a content addressable and self referential identifier, called UUID, as specified in the IETF Self-Addressing IDentifier (SAID) draft specification [i.144]). Note that if an ACDC attribute set does not include an UUID, then its content is not private, and consequently it does not make much sense to discuss disclosure of attributes that an entity can derive using a rainbow table attack.

The example is compact because only commitments to other key value pair sets are included. For instance, in the above example, the key "a" is the unique identifier for a set of attributes but the attributes themselves are omitted.

A user can disclose the above ACDC by presenting ("u":"0ANghkDaG7OY1wjaDAE0qHcg"), i.e. a verifiable UUID, to a verifier and then disclosing the rest of the attributes in the ACDC. The verifier can then use the rest of the attributes to compute the value of "u" and compare it with the previously disclosed commitment. Relatedly, the user can further disclose identity related attributes by presenting the uncompacted private attribute key value set.

```
{
  "a":{
    "d":"EgveY4-9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PY",
    "u":"0AwjaDAE0qHcgNghkDaG7OY1",
    "i":"did:keri:EpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPmkPreYA",
    "score":96,
    "name":"Jane Doe"
  }
}
```

Note how disclosure of attributes in "a" discloses the entire set. A user who wants to disclose individual identity attributes needs to use a selective disclosable attribute ACDC. There, each attribute is blinded individually as follows:

```
{
  "A":[
    {
      "d":"ErzwLIr9Bf7V_NHwY1lkFrn9y2PYgveY4-9XgOcLxUde",
      "u":"0AqHcgNghkDaG7OY1wjaDAE0",
      "i":"did:keri:EpZfFk66jpf3uFv7vklXKhzBrAqjsKAn2EDIPmkPreYA"
    },
    {
      "d":"ELIr9Bf7V_NHwY1lkgveY4-Frn9y2PY9XgOcLxUderzw",
      "u":"0AG7OY1wjaDAE0qHcgNghkDa",
      "score":96
    },
    {
      "d":"E9XgOcLxUderzwLIr9Bf7V_NHwY1lkFrn9y2PYgveY4-",
      "u":"0AghkDaG7OY1wjaDAE0qHcgN",
      "name":"Jane Doe"
    }
  ]
}
```

Note how each attribute is selectively disclosable independently. Note also the capital "A" as key.

As with any salted attribute hash based approach to selective disclosure, ACDC only offers selective disclosure ability and does not offer inherent protection against verifiers colluding and correlating the users use of an ACDC. The UUID is a perfect correlation handle that any entity can use to track the user's behaviour. To protect against such correlation, the IETF ACDC draft specification [i.142] discusses bulk issuance, where correlation handles are removed (see section 13.5.2 of IETF ACDC draft specification [i.142]). Note that such an approach does not protect against malicious issuers that wish to track the user. Succinctly put, ACDC is verifier unlinkable but not fully unlinkable.

ACDC is considered as being plausible quantum safe since they are based on hashes in a Directed Acyclic Graph. If the used hash functions are designed as QSC, the ACDC scheme becomes quantum-safe.

4.3.9 Gordian Envelopes

The Gordian Envelope [i.145] is a structured format for verifiable hierarchical data. The approach relies on a graph to logically order and structure salted attributes included in an attestation. Hence, it can be used to create Directed Acyclic Graphs (DAGs) through references within or between Envelopes. Claims can be structured as subject-predicate-object triplets (the predicate and the object are in turn envelopes), e.g. subject:Alice, predicate:knows, object:Bob.

The envelope itself is not limited to such triplets. An Envelope can enclose various types of data, ranging from basic plaintext messages to ciphertext to semantic graphs. These can then be represented in different ways in an envelope. The ways include nodes, leaves, nestled structures among others; common to all is that the envelope is meant to contain deterministically encoded identity subject claims that may or may not be encrypted, compressed, or made disclosable. The user has multiple ways to limit disclosures:

1) A single part of the triplet can be hidden: subject:Alice, predicate:knows, object:.

2) Multiple parts of the triplet can be hidden: subject:Alice, predicate:, object:.

3) The existence of the claim can be hidden.

Each envelope produces a unique and content determined digest, meaning that envelopes that are semantically identical produce the same digest. By extension, an identical identity subject with an identical claims set will yield the same digest tree every time the (Q)EAA is enveloped. As with other salted attribute hash approaches, the issuer signs the digests, which allows the user to later reveal claims associated with the digests. In the case of Gordian Envelopes, selective disclosure is possible by revealing only those objects required to traverse a path of interest and to calculate the Merkle root that is involved in the verification of the attestation.

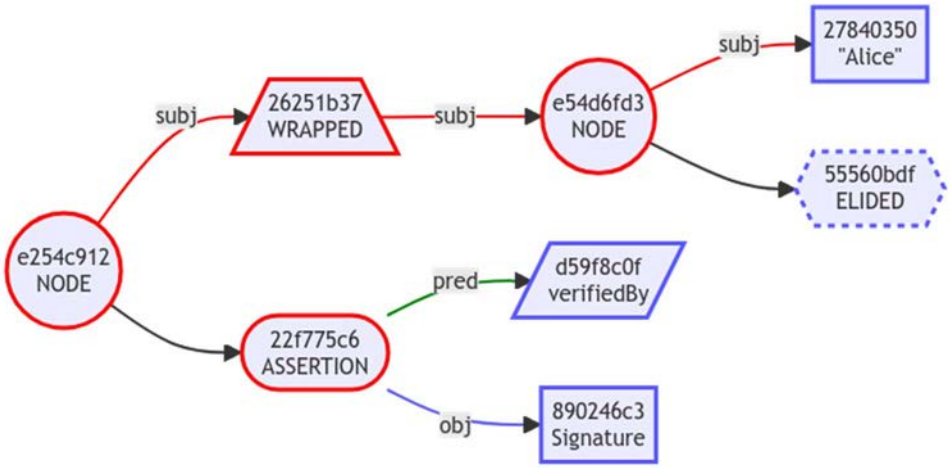


Figure 5: An example of a verifiable graph that selectively discloses only the subject

As with any salted attribute digest based approach to selective disclosure, a Gordian Envelope only offers selective disclosure ability and does not offer inherent protection against verifiers colluding and correlating the users based on the attestations they see. To prevent verifier collusion, Gordian envelopes support salting. Specifically, a unique salt is added as a predicate with a random number as the corresponding object to every envelope. As with any salted attribute hash approach, adding salts requires batch issuance, and does not protect against a malicious colluding issuer. In other words, Gordian Envelopes are verifier unlinkable but not fully unlinkable.

Gordian Envelopes are considered as being plausible quantum safe since they are based on hashes in a Directed Acyclic Graph. If the used hash functions are designed as QSC, the Gordian Envelopes scheme becomes quantum-safe.

4.4 Multi-message signature schemes

4.4.1 Camenisch-Lysyanskaya (CL) signatures

4.4.1.1 Introduction to CL-signatures

In their paper "A signature scheme with efficient protocols" [i.42] (2002), Camenisch and Lysyanskaya introduce the CL-signature. The authors explicitly sought to design signature schemes that would be "suitable as building blocks for other applications".

Of particular relevance to this text is that the CL-signature allows for the implementation of two additional protocols. The first protocol is a secure multiparty computation protocol that allows an issuer to issue a signed attestation to the user, without the issuer learning all the message content or the final signature value. The ability for a signer to obliviously sign a user provided commitment to a message is enables, among other things, the user to convenience a verifier that two attestations were issued to the same identity subject simply by providing an equality proof between the two (blinded) commitments in the two attestations. Relatedly, it allows the user to generate a proof of possession of the commitment value in a privacy preserving way. The second protocol enables the user to prove possession of a, potentially hidden and blinded, message-signature pair (in CL-signatures, this proof is done in a ZKP manner). This ability for the user to present different looking presentations based on the same underlying issuer signed attestation is an important property when seeking to achieve privacy across distinct authentications.

Together, the two protocols above are introduced to achieve what Camenisch and Lysyanskya describe as an anonymous credential system. Such a system has two important requirements:

1) The user is required to demonstrate to a verifier that they possess the right attributes for a specific service, without the verifier being able to infer anything other than the fact that the user has the right attributes.

2) The user is required to obtain attribute attestations without revealing their identity to the issuer (in the paper "A signature scheme with efficient protocols" [i.42], the authors consider the user's secret key to be equivalent to the user's identity).

A signature scheme that can meet the above two requirements is one that allows the design of protocols that can prove statements in the form of "I have a valid signature" and where these signatures are over blinded committed values.

4.4.1.2 The CL-signature scheme

CL-signatures enable the signing of messages without affecting the message's algebraic structure; a property that allows a user to prove statements about messages even if these messages are hidden in some way (e.g. using a commitment).

For key generation, the first CL-scheme relies on a special RSA modulus n = pq, where (p, q) are safe primes, and the quadratic residues mod n (a,b,c). The public key is (n,a,b,c) and the secret key is (p). The message space consists of the integers in range $[0, 2^{l\_m})$ for the parameter l_m. The signing algorithm takes as input a message m, selects a random prime number e and a random value s of suitable lengths (the paper "A signature scheme with efficient protocols" [i.42] details how to select the proper parameters) and computes the value v such that $v^e = a^m b^s c \ (mod\ n)$. The signature verification is done using the tuple (e,s,v), where it is the user that completes the value for s based on input from the issuer, and the message m by checking that $v^e = a^m b^s c \ (mod\ n)$ and that e is within the suitable range.

Later versions rely on bilinear pairings and are more efficient.

As aforementioned, the CL-signature scheme preserves the message's algebraic structure. As such, when signing a block of messages, $(m_1, m_2, \ldots, m_L)$ it is not permitted to simply sign the hash over the block of messages $H(m_1, m_2, \ldots, m_L)$ as this would make it impossible with Schnorr proofs to both prove relations among the message components, the oblivious signature demand, and to prove predicates. Instead, the previous signing algorithm is modified to allow for multi-message signing as follows:

$$v^e = a_1^{m_1} a_2^{m_2} \ldots a_L^{m_L} b^s c \ (mod\ n)$$

As such, in a sense, each message is signed with an individual key by the issuer, and all the signatures are combined to a single one. Next it will be described how the CL-signature scheme enables selective disclosure.

4.4.1.3 The CL-signature scheme and selective disclosure

In essence, the CL-signature includes a commitment vector of messages $a_1^{m_1} a_2^{m_2} \ldots a_L^{m_L}$. The following characteristics can now be observed:

- All the quadratic residues are public.
- The commitment $a^m b^s \ (\mathrm{mod\ n})$ prevents the verifier from learning m as long as solving the DLP in that group is hard. This kind of commitment is called a Pedersen commitment with a message m that is committed and a blinding factor s.
- The user can present any combination of the commitment and the cleartext message.

The last point is what enables selective disclosure. Basically, the user will present in cleartext all the messages they wish to reveal, and the commitments to the messages they wish to keep secret. For instance, if a user wants to present m_1 but keep m_2 hidden, the user would present $((a_1, m_1), a_2^{m_2})$.

4.4.1.4 The CL-signature scheme, predicates, and knowledge proofs

Since the algebraic structure of the messages is preserved, it is possible to generate various proofs using CL-signatures.

In their original paper, Camenisch and Lysyanskya list the following protocols known to be secure under the strong RSA assumption:

- Proof of knowledge of discrete logarithm representation modulo a composite. Under specific conditions, this can be used to prove knowledge of exponents $(m_1, m_2, \ldots, m_L)$ in the commitments $a_1^{m_1} a_2^{m_2} \ldots a_L^{m_L}$ without revealing the exponents.
- Proof of knowledge of equality of representation modulo two (possibly different) composite moduli. This one is similar to the one above, but can prove knowledge of exponents even if the bases are different and the composite moduli are different.
- Proof that a committed value, $g^{ab} h^{r_3} (mod\ n)$, is the product of two other committed values, ($g^a h^{r_1} (mod\ n)$, $g^b h^{r_2} (mod\ n)$), without revealing any of the values.
- Proof that a committed value, $g^x h^r (mod\ n)$, lies in a given integer interval $a \leq x \leq b$. This builds on other known proofs that a committed value is a square (i.e. a positive number) and greater than or equal to proofs.

The above support the various predicate proofs that attestation systems based on CL-signatures are capable of, set (non-) membership tests, enable the property where the user can provide a proof of a valid signature as opposed to presenting the signature itself, and allows the user to request a signature over blinded messages. By extension, these properties provide unlinkability for the user as issuer and verifiers cannot collude to track use of an attestation.

EXAMPLE: A positive number proof can be easily constructed using other proofs. Lagrange's four-square theorem states that every natural number can be represented as the sum of four non-negative integer squares. Remember that there exists a way for the user to prove that a committed value is a square. A user could then send over the commitments to the square values, together with their corresponding proofs. The verifier can then easily check that another number is a positive number using the four commitments of a square number proof.

4.4.1.5 Cryptographic analysis of the CL-signature scheme

Since the first CL-signature scheme is based on the strong RSA assumption, and later versions are based on bilinear-pairings, they are not considered as being plausible quantum-safe in a post-quantum world. The CL-signature schemes are also not possible to construct using SOG-IS approved inputs. As with BBS+ signatures, the data confidentiality properties of a CL signatures remain safe even against a computationally unbounded attacker, but such an attacker can recover the signer's private key and forge signatures and proofs. For a more general discussion on the Post Quantum Computer implications, see clause 9.

The CL-Signature scheme is fully unlinkable.

4.4.2 The BBS, BBS+ and BBS# signature schemes

4.4.2.1 Background: Boneh-Boyen-Shacham (BBS04) signature scheme

Initially, the term group signatures was introduced in 1991 by Chaum and van Heyst in their paper "Group signatures" [i.62] as a scheme that provides anonymity for signers. This means that any member of the group can sign a message, but the resulting signature keeps the identity of the signer secret. The Stanford cryptography researchers Boneh, Lynn and Shacham continued the research on group signatures with respect to bilinear pairings, and published the results in their paper "Short signatures from the Weil pairing" [i.28] in 2001, where the Weil pairing refers to elliptic curve bilinear pairings [i.199].

Three years later the BBS04 signature scheme was published 2004 in the paper "Short Group Signatures" [i.27] by Boneh, Boyen and Shacham, who also named the BBS04 signature scheme after their initials. The BBS04 is a group signature scheme that is based on the Strong Diffie-Hellman assumption in conjunction with bilinear groups called the Decision Linear assumption.

4.4.2.2 Introducing the BBS+ signature scheme

Based on the BBS04 signature scheme, the cryptographic research has continued with BBS+, which allows for multi-messages to be selectively disclosed and signed with group signatures. One major contribution was Camenisch and Lysyanskaya and their 2004 work on signature schemes and anonymous credentials from bilinear maps [i.45]. The BBS+ signature scheme was described for the first time in 2006 by Au et al. in the paper "Constant-size dynamic k-TAA" [i.12]. Furthermore, the BBS+ signature scheme is proven to be secure in the type-3 pairing setting in the paper "Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited" [i.40] published by Camenisch et al. in 2016.

The BBS+ signature scheme is a multi-message digital signature protocol that proves knowledge of a signature while selectively disclosing any subset of the signed messages. Similar to CL-signatures (see clause 4.4.1.2), BBS+ signatures preserve the algebraic structure of the messages and rely on commitments. Specifically, the message $M = (m_1, m_2, \ldots, m_L)$ is used in a commitment as follows:

$A = (g_1 h_1^{m_1} h_2^{m_2} \ldots h_L^{m_L})^{(1/(x+e))}$, where *h_1*, …, *h_L* are generators of the group *G_1*.

NOTE 1: The present report uses the multiplicative notation for point operations here as is usual for pairing based constructions. Generally speaking, the notation in the present document follows the one from the paper "Constant-size dynamic k-TAA" [i.12].

The signature on *M* is *(A,e)*. The proof generation and verification then involves disclosing the messages and generators that the user wishes to present.

NOTE 2: The IRTF CFRG BBS draft [i.177] differs from the above in subtle ways but the core selective disclosure mechanism is the same.

The BBS+ scheme allows for signing multiple messages whilst producing a single, constant size, digital (group) signature. BBS+ Signatures allow for an efficient ZKP protocol, hence the BBS+ proofs do not reveal any information about the undisclosed messages or the original signature. A user who possesses a signature is able to generate multiple, unlinkable proofs that selectively disclose subsets of the originally signed messages, yet preserving the authenticity and integrity of the messages.

A user can generate a ZKP proof of knowledge of a valid BBS+ signature, which makes BBS+ signatures suitable in cases that seek to prevent linkability through the issuer's signature.

BBS+ also allows for splitting the prove operation into two operations as introduced in Chen's paper "A DAA scheme requiring less TPM resources" [i.64] and following work, where simple operations are performed in the Secure Element, whereas the computationally expensive operations (e.g. pairing operations) happen in the host system. This construction allows for an efficient way to achieve secure device-binding without having to implement complex pairing operations in restricted systems like SEs or TPMs. This option is discussed in more detail in clause 4.4.2.5.

4.4.2.3 Overview of BBS+

The BBS+ signature scheme is illustrated in Figure 6.

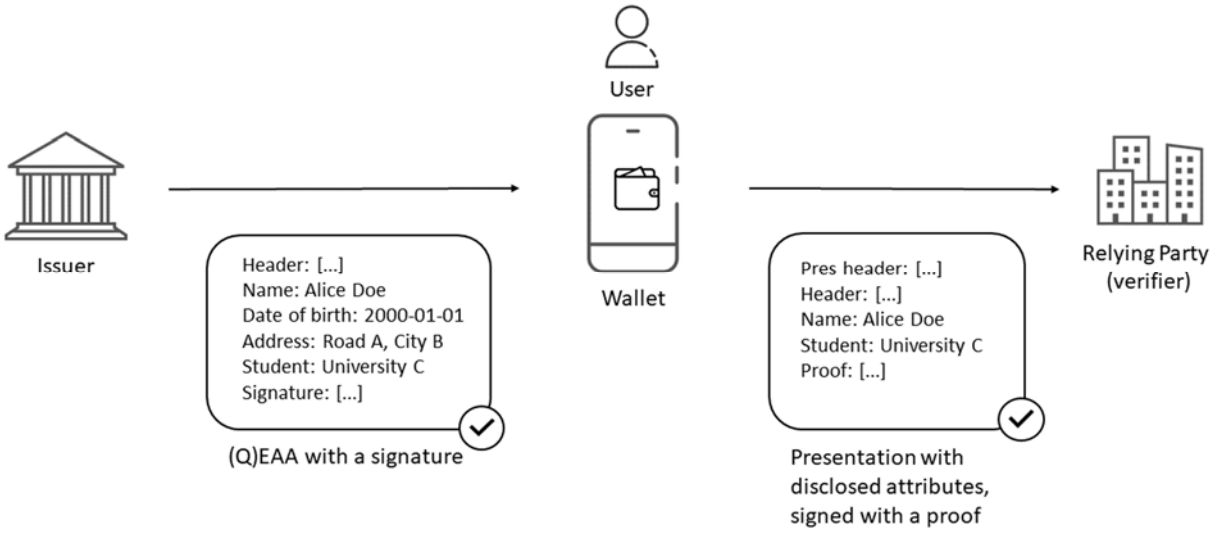


Figure 6: Overview of the BBS+ signature scheme

The issuer issues a (Q)EAA, with a header and a complete set of attributes, which is signed by the issuer. The (Q)EAA is stored in the user's wallet.

The user selects the attributes to disclose to a relying party, and the wallet generates a presentation with the disclosed attributes. The presentation contains a presentation header, the original header, the selectively disclosed attributes, and a proof. The proof reveals the user's knowledge of the original signature, but does not reveal the actual signature.

4.4.2.4 IRTF CFRG BBS specification

The IRTF Crypto Forum Research Group (CFRG) has created the internet draft specification "The BBS Signature Scheme" [i.177], which specifies an internet profile of the BBS+ scheme. The specification describes the following topics:

- Scheme Definition ([i.177], clause 3) defines the core operations and parameters for the BBS+ signature scheme.
- Utility Operations ([i.177], clause 4) defines utilities used by the BBS+ signature scheme.
- Security Considerations ([i.177], clause 5) describes a set of security considerations associated with the signature scheme.
- Ciphersuites ([i.177], clause 6) define the format of a ciphersuite.

More specifically, the IRTF CFRG BBS+ draft specifies pairing-friendly ECC curves [i.178] alongside a concrete ciphersuite based on the BLS12-381 curve.

NOTE: The IRTF CFRG draft specification [i.177] has the title "The BBS Signature Scheme", although it describes the BBS+ scheme. There has been some ambiguity on what exactly the "+" in BBS+ means. Within the present document, the term BBS+ is used to describe the multi-message signature scheme, whilst the term BBS04 describes the original single-message signature scheme.

Furthermore, IRTF CFRG has published two more BBS drafts:

- IRTF CFRG Blind BBS Signatures [i.175]. The present document defines an extension to the BBS Signature scheme that supports blind digital signatures, i.e. signatures over messages not known to the Issuer.
- IRTF CFRG BBS per Verifier Linkability [i.174]. The present document presents the use of pseudonyms with BBS proofs. A pseudonym, is a value that will remain constant each time an EUDI Wallet presents a BBS proof to the same Relying Party, but will be different (and unlinkable), when the EUDI Wallet interacts with a different Relying Party. This provides a way for a recipient (Relying Party) to track the presentations intended for them, while also hindering them from tracking the EUDI Wallet's interactions with other Relying Parties.

The three BBS drafts mentioned above have been adopted by the IRTF CFRG and are on track to become IETF RFCs.

4.4.2.5 Device Binding Options for BBS+

One of the core problems for the deployment of BBS+ within a reasonable amount of time seems to be the holder/device binding. Common denominator for trusted hardware (or Secure Elements) currently seems to be curve P-256 and changes to such hardware would likely take time, unless many mobile phone app manufacturers announce the opening of their secure elements to certain developer activity. There seem to be 2 approaches towards a secure device binding of BBS+ based credentials:

- A split signature using the native BBS+ construction (as described in [i.184], [i.40], [i.50] and [i.59]).
- Binding to a P-256 private key.

While split signatures BBS+ are reasonably well understood and would only require minimal operations in the trusted hardware, at the very least support for a Schnorr Proof on a pairing friendly curve would be required. This would mean the addition of a new curve and exposing operations on this curve via the external interfaces to Secure Elements. While this would technically be the cleanest and easiest device binding for BBS+, it would likely take several years until such a construction would see enough support in smartphones to become feasible in the real world. The traditional way for this split operation would also require the trusted hardware to hold state during the proof generation.

Another construction to realize a secure device binding with BBS+ based signatures is to leverage ECDSA based signatures over P-256 and construct a proof over a committed public key. A concrete instantiation of a protocol based on such a binding could look like this:

- The BBS+ messages contain a commitment to a P-256 public key.
- The trusted hardware creates an ECDSA based proof of possession by signing over a message that includes a verifier-provided challenge.
- The holder reveals the message and creates a ZKP that proves that they know a valid signature over said message that belongs to the committed public key.

The idea behind such proofs has been originally proposed in the context of ring- and group signatures as zkAttest [i.108] by Faz-Hernández et al. and subsequently been improved by Celi et al. [i.53] as CDLS. Combining this approach with BBS+ has been proposed and implemented by Amrein/Ubique in the SPRIND EUDI Funke Innovation Challenge [i.221]. Those constructions are based on sigma protocols and computationally expensive (~800 ms for the prove operation) and will require further analysis, but might be a useful alternative to other schemes.

There is also more recent work using zkSNARKs over ECDSA based legacy credentials that include efficient variants of this general scheme. The protocol proposed by Woo et al. [i.253] is a promising candidate for zero-knowledge Proof of Possession using curve P-256 and ECDSA, but comes with its current construction at the cost of losing post-quantum privacy guarantees. Similarly, the construction of Frigo et al. (see clause 6.5.4) contains such a proof-of-possession part that could in principle also be used in combination with BBS+.

Constructions leveraging a combination of BBS+ and circuit-based approaches could significantly reduce the complexity of the circuits involved since the proof would only need to be over a committed public key and a signature. The data that is signed over can be revealed in the case of a proof of possession which would reduce a lot of the complexity of the currently proposed circuit-based solutions (which are usually proving over the original credential while hiding its raw content instead of only a proof of possession / key binding).

An alternative to these direct device binding constructions would be the involvement of the issuer during each presentation that allows for significantly more performant constructions at the cost of additional communication with the issuer. Such constructions are proposed in clause 4.4.3 as BBS# and in the work by Chairattana-Apirom et al. [i.57].

4.4.2.6 Cryptographic analysis of the BBS+ signature scheme

In a post-quantum world, SDH algorithms based on bilinear pairings are vulnerable against quantum computing attacks [i.244]. This is an identified weakness of the BBS+ signature scheme, which has been described in a cryptographic review [i.244] prepared for the U.S. Department of Homeland Security. The report [i.244] claims that BBS+ signatures are not standardized by NIST, and are unlikely to be standardized, since they rely on ECC with BLS12-381 curves that are not considered quantum-safe in a post-quantum world. The European standardization organization SOG-IS has not approved the BLS12-381 [i.30] curves either. The U.S cryptographic review [i.244] gives the following recommendations for the IRTF CFRG BBS draft specification [i.177] to move closer to government compliance: use the SHAKE256 hash function from SHA-3 and an approved random number generator in the BBS+ signature implementation.

While the strong Diffie-Hellman assumption is not quantum resistant, the threat from an attacker utilizing a quantum computer is more difficult to assess. In general, the parts of a BBS+ secured (Q)EAA that are ZKP are secure against a computationally unbounded adversary, whereas the parts that can be attacked based on public knowledge (e.g. a signature or a public key) need to either be frequently rotated, used once only, or replaced with quantum resistant alternatives. Put differently, an attacker can use a quantum computer to reveal the signer's private key from the public key and thereafter forge proofs and signatures. But an attacker will not be able to break data confidentiality, meaning that undisclosed messages are safe in a post-quantum world, as are undisclosed signature values. For a more general discussion on the Post Quantum Computer implications, see clause 9.

The BBS+ signature scheme is fully unlinkable (i.e. to issuers, verifiers, and any other party).

4.4.3 The BBS# signature scheme

4.4.3.1 Introduction to the BBS# protocol

BBS# [i.78] is a variant of BBS/BBS+ that has been designed to meet several stringent requirements put forth in the eIDAS 2.0. regulation. More precisely, BBS# removes the need for pairings and pairing-friendly curves (which are not standardized and not supported by trusted phone hardware) and can be combined with SOG-IS sanctioned protocols for the implementation of the holder binding feature. This feature states that only the legitimate holder of a credential can be able to perform transactions with that credential. In practice, this is achieved by binding that credential to a private key stored in a trusted hardware (or Secure Element) of the credential holder's mobile device and making presentations of such a credential impossible without that private key.

Unlike BBS, BBS# can therefore be used with conventional elliptic curves (such as the NIST P-256 curve) and enables a credential to be bound to a hardware-protected device key without requiring any change in that hardware or in the algorithms it supports.

4.4.3.2 BBS# underlying signature schemes

4.4.3.2.1 General

BBS# makes use of two different types of signature schemes: on the user's side, BBS# requires (for the holder binding feature) a signature scheme that supports key blinding also known as key randomization [i.84] and [i.111] whereas on the issuer's side, a pairing-free variant of BBS, sometimes called MAC_{BBS}, is used.

4.4.3.2.2 Holder's signature scheme

Signatures schemes that support key blinding have the advantage that one can randomize or blind a key pair (sk, pk) to a key-pair (sk', pk') and sign a message m with the seemingly unrelated key (sk'). Of course, in the specific context of a verifiable presentation of a (Q)EEA (bound to a hardware-protected device key sk), the user will have to prove (in ZK) that pk' is a randomized version of a public key pk that has been certified by a given issuer.

The main goal of this randomization is to ensure that neither a verifier nor even an issuer will be able to trace a user from their public key and the signatures they issued (as with the ISO mobile driving license). In other words, the verifier will not be able to distinguish between two signatures using two fresh keys obtained from the randomization of the same long-term key sk and two signatures using two fresh keys obtained from the randomization of two distinct long-term keys sk and sk*. In the same way, the issuer will not be able to recognize any of the public key it certified or signature it generated.

BBS# supports two signature schemes with key blinding: ECSDSA (a.k.a. ECSchnorr) as specified in ISO/IEC 14888-3 [i.180] with additive blinding and ECDSA as specified in ISO/IEC 14888-3 [i.180] with multiplicative blinding.

4.4.3.2.3 Issuer's signature scheme

The signature scheme used on the issuer's side is a pairing-free variant of BBS, called MAC_{BBS}, which, like BBS, also preserves the algebraic structure of the messages and relies on commitments.

NOTE 1 : The following notation, introduced by Camenisch and Stadler [i.48] will be used in the sequel to denote a zero-knowledge proof of knowledge (PoK): π := PoK{α,β : statements about α,β} where the Greek letters α and β correspond to the knowledge of the prover. For example, π := PoK{α,β : $y = g^{\alpha} \wedge z = g^{\beta}$} denotes a proof of knowledge of secrets α, β, verifying the statement on the right hand side of the colon.

Setup:

Let G denote a cyclic group of prime order p, $\tilde{g}$, g, h_1, h_2, ⋯, h_L, $L+2$ random generators of G, x is the issuer's private key and $PK_1 = \tilde{g}^x$ is the corresponding public key.

Signing:

To sign a message $M = (m_1, m_2, \cdots, m_L)$, the issuer first chooses a random value e in $[1, \cdots, p]$ and computes $A = (g h_1^{m1} h_2^{m2} \cdots h_L^{mL})^{\frac{1}{x+e}}$.

If $C_M = g h_1^{m_1} h_2^{m_2} \ldots h_L^{m_L}$ and $B = C_M A^{-e}$ then $B = C_M A^{-e} = A^x$.

The issuer then computes a ZKP π_{DLEQ} proving that the discrete logarithm of B in the base A is equal to the discrete logarithm of PK_1 in the base $\tilde{g}$ [i.61]: $\pi_{DLEQ} := PoK\{\alpha : B = A^{\alpha} \wedge PK_1 = \tilde{g}^{\alpha}\}$.

The signature (also called a tag) on M is the pair (A, e) along with the proof π_{DLEQ}.

Verification:

To check whether the tag (A, e, π_{DLEQ}) is valid on M, the verifier first computes $C_M = g h_1^{m_1} h_2^{m_2} \ldots h_L^{m_L}$ and $B = C_M A^{-e}$ and then verify the validity of π_{DLEQ}. The tag is valid if the proof π_{DLEQ} is valid.

Comment:

A feature of MAC_{BBS} is that a tag can be randomized. The user can choose a random value r in [1,⋯, p] and compute $(\underline{A}, \underline{B}) = (A^r, B^r)$. The randomization still preserves the equality $\underline{B} = \underline{A}^x$.

NOTE 2 : In the following and for readability, the tag will either consist of the pair (A, e) or of the pair (A, B), which is an equivalent formulation.

Given $(\underline{A}, \underline{B})$, a randomized version of (A, B), no one, including the issuer, will be able to determine whether $(\underline{A}, \underline{B})$ is a randomized version of the tag (A, B) or of another tag (A', B').

Another feature of MAC_{BBS} tags is that the corresponding discrete logarithm equality proof π_{DLEQ} can be requested anonymously and issued obliviously (by the issuer) on a randomized version $(\underline{A}, \underline{B})$ of the tag (A, B) [i.218].

More precisely, the proof will be issued in such a way that the issuer will be unable to link $(\underline{A}, \underline{B}, \underline{\pi}_{DLEQ})$ to its respective issuance (obliviousness), where $\underline{\pi}_{DLEQ} := PoK\{\alpha : \underline{B} = \underline{A}^{\alpha} \wedge PK_1 = \tilde{g}^{\alpha}\}$. In a nutshell, the user first re-randomizes the pair $(\underline{A}, \underline{B})$, and obtains a new pair $(\hat{A}, \hat{B}) = (\underline{A}^l, \underline{B}^l)$ for a random value l in [1,.., p], and sends $(\hat{A}, \hat{B})$ to the issuer. The issuer then computes a proof $\hat{\pi}_{DLEQ} := PoK\{\alpha : \hat{B} = \hat{A}^{\alpha} \wedge PK_1 = \tilde{g}^{\alpha}\}$ and transmits $\hat{\pi}_{DLEQ}$ to the user who (knowing l) can 'de-randomize' $\hat{\pi}_{DLEQ}$ to retrieve $\underline{\pi}_{DLEQ}$. The issuer will be unable to link $(\underline{A}, \underline{B}, \underline{\pi}_{DLEQ})$ to $(\hat{A}, \hat{B}, \hat{\pi}_{DLEQ})$.

An example on how to obliviously obtain the proof $\underline{\pi}_{DLEQ}$ is given in Appendix E of [i.78]. These two features (randomization of the tags and obliviousness of the discrete logarithms equality proofs) will be useful to prevent linkability of an issuer's signature / tag during the verifiable presentation of a (Q)EEA.

4.4.3.3 Overview of the BBS# protocol

4.4.3.3.1 General

The BBS# protocol is illustrated in Figure 6, with the addition of the holder binding public key pk in the (Q)EEA).

Let sk denote the user's hardware-protected device key and $pk = h_1^{sk}$ the corresponding public key.

4.4.3.3.2 Issuance

The issuer creates a MACBBS authentication tag σ on the user's public pk (of a signature scheme supporting key blinding / randomization) and on their attributes $\{m_i\}_{i=1}^{n}$. The tag σ=(A,e) represents the user's credential and authenticates both the user's attributes and their public key pk, where:
$A = (gpkh_2^{m2}h_3^{m3} \cdots h_L^{mL})^{\frac{1}{x+e}} = (gh_1^{sk}h_2^{m2} \cdots h_L^{mL})^{\frac{1}{x+e}}$.

Finally, the issuer transmits σ=(A,e), along with the corresponding proof of validity π_{DLEQ}, to the user's EUDI Wallet.

4.4.3.3.3 Selective disclosure

During a Verifiable Presentation (VP) of a user's attributes (or a subset of them) to the relying party (verifier), the user will first randomize their public key pk (either additively if ECSDSA is used on the user's secure cryptographic device or multiplicatively in the case of ECDSA) as well as their tag (i.e. their verifiable credential) σ. These randomized versions are denoted by pk_{Blind} and $\sigma_{Blind} = (\underline{A}, \underline{B})$ respectively.

The user will then first generate a signature σ_{HB}, using the private key associated with pk_{Blind}, on a nonce generated by the verifier (to guarantee the freshness of the VP) and then a ZKP $\pi_{Validity}$ proving knowledge of (a) two random factors (r, r'), (b) a credential σ and (c) of a public pk such that:

1) σ_{Blind} is a randomized version of σ under the random factor r;
2) pk_{Blind} is a randomized version of pk under the random factor r'; and
3) σ is a valid MACBBS authentication tag on the disclosed attributes requested by the verifier.

The signature σ_{HB} is a proof that the VP originates from the user holding the underlying credential σ on the attributes disclosed to the verifier (holder binding).

The VP consists of the following elements: $VP = (\{m_i\}_{i\in D}, pk_{Blind}, \sigma_{HB}, \sigma_{Blind} = (\underline{A}, \underline{B}), \pi_{Validity})$, where $\{m_i\}_{i\in D}$represents the disclosed attributes. Since both the user's public key pk and their credential σ have been randomized, no one, including the issuer, will be able to trace a user or link their VPs from these two values or from σ_{HB} (full unlinkability).

4.4.3.3.4 Verification

Upon receipt of $VP = (\{m_i\}_{i\in D}, pk_{Blind}, \sigma_{HB}, \sigma_{Blind} = (\underline{A}, \underline{B}), \pi_{Validity})$ the verifier first checks that the signature σ_{HB} is valid, using pk_{Blind}, and then verifies the validity of the ZKP $\pi_{Validity}$ using PKI.
The VP is considered valid if both verifications are successful and if the following equality $\underline{B} = \underline{A}^x$holds or not.

In BBS/BBS+ based anonymous credentials schemes, pairings are used by the verifier to check if the latter equality holds. Since BBS# is pairing-free, three options are therefore proposed to let any verifier perform this check.

Option 1:

The first option is to let the verifier ask the issuer to check whether the equality $\underline{B} = \underline{A}^x$ holds or not. As $\underline{A}$ and $\underline{B}$ have been randomized by the user (they consist of the randomization of his credential values A and B), the issuer cannot trace back the user from these values. Obviously, the issuer should prove to the verifier whether this equality holds or not. This can be done for example by using the classical Chaum-Pedersen ZKP of discrete logarithms equality π_{DLEQ} [i.61] when the equality holds and by using for example the discrete logarithm inequality proof of Camenisch and Shoup otherwise [i.47].

A similar approach has been adopted in the card payment sector to enable a point-of-sale terminal to check the validity of a smart card transaction online with the issuer (the cardholder's bank).

NOTE 1: The smart card transaction roughly consists of a MAC computed by the card on the payment data elements such as the transaction amount and transaction date.

Option 2:

The second option is to let the user request anonymously from the issuer, during the presentation protocol, a blind proof (also known as an oblivious proof [i.218]), π_{DLEQ} showing that $\underline{B} = \underline{A}^x$, that will be sent, along the VP, to the relying party (verifier).

By blind, it is meant that the issuer, although contributing to the generation of this proof (as only they know x), will be unable to later link a given tuple $(\underline{A}, \underline{B}, \underline{\pi}_{DLEQ})$ to its respective issuance. The proof $\underline{\pi}_{DLEQ}$ can be verified by anyone using solely the issuer's public key.

This approach (option 2) is similar to the one used in the context of centralized / federated Identity Management Systems (IMSs). In fact, in a federated IMS when a user wants to authenticate at a Relying Party (or prove that they hold the attributes requested by that Relying Party), the user is redirected to their Identity Provider (issuer) in order to obtain a token (signed by the issuer), which the user can present to the Relying Party as a proof that they have authenticated to the issuer (or that they hold the requested attributes). However, unlike federated IMS, with option 2, neither the issuer nor the verifier (even if they collude) will be able to track or link the user's activity. Indeed, since the user anonymously requests the blind proof $\underline{\pi}_{DLEQ}$, a time-correlation attack for example will not work.

This option is much less efficient than option 1 because the necessary blinding to anonymize the holder requires much more computation than option 1 where the verifier is doing the request directly.

Option 3:

The user generates ahead of time several pairs $(A_i = A^{r_i})$, $(B_i = B^{r_i})$ and requests from the Issuer (in batch), blind proofs π_{DLEQ} showing that $B_i = A_i^x$ and stores these blind proofs for future use (and only use them in the rare cases where both the user and the verifier are offline).

This approach (option 3) is similar to that described in the ISO mobile driving license, where a user can obtain several verifiable credentials at once (in batch) to prevent colluding verifiers from tracing them. However, this option provides full unlinkability, unlike the ISO mobile driving license batch credential issuance approach.

NOTE 2: A different credential has to be used for each new VP, resulting in batch issuance of credentials. Obtaining several verifiable credentials may be expensive (as the user will have to be strongly authenticated to obtain new credentials) or no longer allowed by the issuer. Unlike option 2, where blind proofs can be issued without the user being authenticated.

This option is also much less efficient than option 1 because the necessary blinding to anonymize the holder requires much more computation than option 1 where the verifier is doing the request directly.

4.4.3.4 Cryptographic analysis of the BBS# protocol

BBS#, which is proven secure in the random oracle model [i.16], retains the well-known security property (unforgeability of the credentials under the (gap) q-SDH assumption) and anonymity properties (full unlinkability and statistical anonymity of presentation proofs) of BBS/BBS+.

Since BBS# is a pairing-free MMS, it removes the main security and certification related issue associated with the other listed MMS as it can leverage the widely deployed ECDSA infrastructure for security while losing nothing of the privacy properties linked to the BBS/BBS+ protocol suite.

As the (Gap) q-SDH assumption is not quantum-safe, an attacker in a post-quantum world will be able to forge BBS# credentials. However, the anonymity of BBS# presentation proofs will be preserved even against a computationally unbounded attacker.

4.4.4 Mercurial signatures

Mercurial signatures [i.74] cater for privacy preserving schemes, such as anonymous credentials, delegatable anonymous credentials, and related applications. They allow a signature s0 on a message m0 under a public key pk0 to be transformed into a signature s1 on an equivalent message m1 under an equivalent public key pk1. For example, pk0 and pk1 may be unlinkable public keys of the same user, and m0 and m1 may be unlinkable pseudonyms of a user to whom some capability is delegated. Mercurial signatures were presented by Crites-Lysyanskaya [i.73] in 2019.

Mercurial signatures are based on Decisional Diffie-Hellman (DDH) over equivalent groups, and are therefore not considered as plausible quantum-safe cryptography in a post-quantum world. Mercurial signatures can however be considered to be secure in a pre-quantum world, and the ZKP of knowledge of Mercurial signatures that are generated in a pre-quantum world will also remain plausible quantum-safe in a post-quantum world (see clause 4.4.2.6).

The Mercurial signature scheme is fully unlinkable when blinded.

4.4.5 Pointcheval-Sanders Multi-Signatures (PS-MS)

Pointcheval-Sanders Multi-Signatures (PS-MS) [i.223] is another multi-message signature scheme based on Bilinear Pairings with efficient Zero Knowledge Proofs. Its construction has some properties that make it preferable to alternatives like BBS+ in some special settings like for example threshold signing.

PS-MS signatures have certain properties that can be used for distributed privacy-preserving Attribute Based Credentials (dp-ABC). The PS-MS signatures are based on a variant of CL-signatures with pairing-friendly curves such as BLS12-461. There is a formal definition of PS-MS signatures by Camenisch et al. in the paper "Short Threshold Dynamic Group Signatures" [i.41] (2020), which are secure under bilinear group model and random oracle model.

An dp-ABC scheme based on PS-MS signatures has been designed by García-Rodríguez et al. in their paper "Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures" [i.117] (2021).

The workflow of a dp-ABC scheme is illustrated in Figure 7.

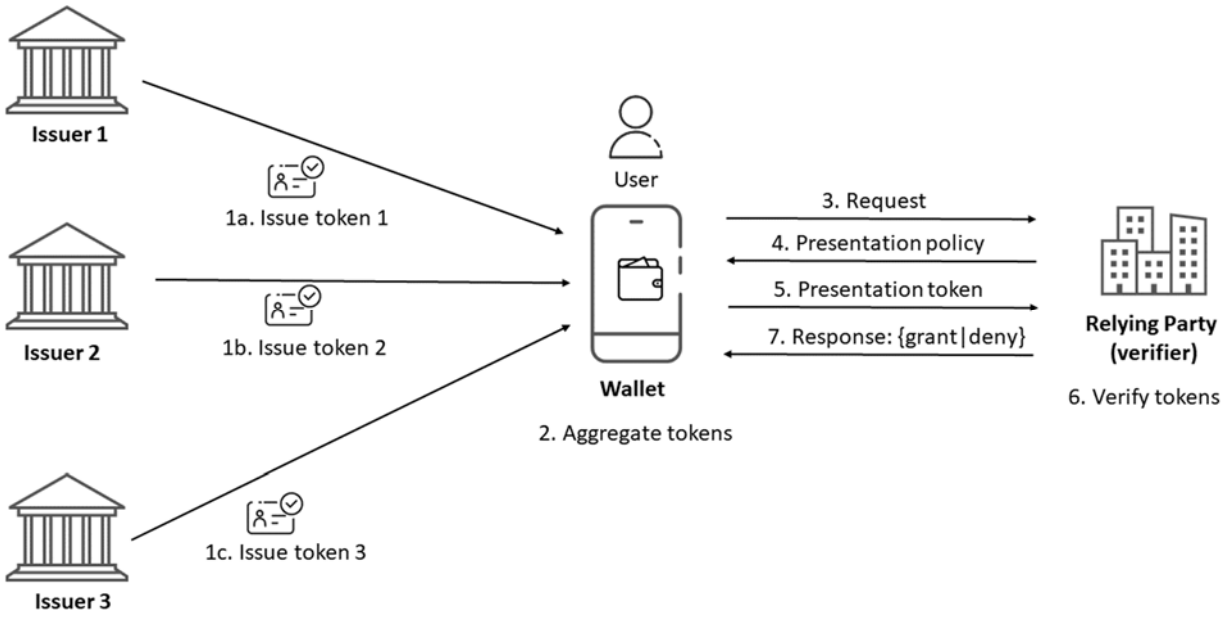


Figure 7: Overview of PS-MS signatures used for dp-ABC flow

More specifically, the PS-MS signatures are used when aggregating the issued tokens in step 2. Selective disclosure and unlinkability is an integral feature of the PS-MS signatures.

NOTE: The identity systems Idemix (clause 6.6.1) and U-Prove (clause 6.6.2) are also based on p-ABC schemes, however, they are based on CL-Signatures and the DLP.

Since the PS-MS signature scheme is based on bilinear-pairings, it is not approved by SOG-IS or considered as being plausible quantum-safe cryptography in a post-quantum world. ZKP of knowledge of PS-MS signatures can however be considered to be secure in a pre-quantum world, and the ZKP of knowledge of PS-MS signatures that are generated in a pre-quantum world will also remain plausible quantum-safe in a post-quantum world (see clause 4.4.2.6).

The PS-MS signature scheme is fully unlinkable.

4.4.6 ISO standardisation of multi-message signature schemes

4.4.6.1 ISO/IEC 20008 - Anonymous digital signatures

The ISO/IEC 20008 series [i.184] specify anonymous digital signature mechanisms (algorithms) as follows:

- ISO/IEC 20008-1 [i.184] specifies a general model with principles, entities, processes, and requirements for anonymous digital signature mechanisms.
- ISO/IEC 20008-2 [i.184] specifies anonymous digital signature mechanisms, for which a verifier can use a group public key to verify a digital signature. For each mechanism, this part of the standard specifies the processes for generating group member signature keys, producing signatures, verifying signatures, opening signatures, linking signatures, and revocation of group members.
- ISO/IEC 20008-3 [i.184] extends ISO/IEC 20008-2 [i.184] by specifying anonymous digital signature mechanisms using multiple public keys.
- ISO/IEC 20008-2/AMD1 [i.184] and ISO/IEC 20008-2/AMD2 [i.184] are amendments to ISO/IEC 20008-2 [i.184] with additional details about certain mechstandardizationanisms.

More specifically, ISO/IEC 20008-2 [i.184] mechanism 3 specifies the cryptographic primitives of a qSDH scheme, which corresponds to BBS04 with single messages as described in 2004 by Boneh, Boyen and Shacham in their paper on short group signatures [i.27]. Since ISO 20008-2 [i.184] mechanism 3 is designed as a single message signature scheme, it requires an extension to support multi-message signature protocols.

BBS+ is an extension of BBS04 (including the Pedersen commitments) to cater for a multi-message signature scheme. Formally, BBS+ relies upon the same security model as the qSDH assumption that is described in ISO 20008-2 [i.184] mechanism 3. More precisely, it is shown (for example in [i.15]) that if an attacker can forge BBS+ signatures then it can also forge BBS04 signatures. In other words, if the BBS04 cryptographic primitives are deemed secure as specified in ISO 20008-2 [i.184], so is BBS+.

Furthermore, the Pointcheval-Sanders Group Signature scheme (PS-GS) [i.223] is specified in ISO 20008-2 [i.184] amendment 2.

4.4.6.2 ISO/IEC 24843 - Privacy-preserving attribute-based credentials

The ISO/IEC Preliminary Work Item (PWI) 24843 [i.185] was approved in March 2025 and a new project on Attribute-Based Credentials has been launched (ISO/IEC 24843 [i.185]). This future standard will specify several attribute-based credential (ABC) mechanisms including those of the PS and BBS/BBS+ families.

In other words, the future ISO/IEC 24843 [i.185] standard will have the potential to result in an ISO standardized version of BBS+ as well as other multi-message signature schemes capable of both selective disclosure and full unlinkability.

4.4.6.3 ISO/IEC CD 27565 - Guidelines on privacy preservation based on ZKP

In addition to the aforementioned ISO standards on anonymous digital signatures and the PWI on privacy-preserving attribute-based credentials, ISO/IEC JTC 1/SC 27 are also working on the common draft ISO/IEC CD 27565 [i.191] "Guidelines on privacy preservation based on zero knowledge proofs". This draft document provides guidelines for how to use ZKPs to improve privacy by minimizing unnecessary information disclosure when sharing personal data between organizations and users.

More specifically, Annex C of ISO/IEC CD 27565 [i.191] includes an example of selective disclosure by using BBS+, with a reference to the IRTF CFRG BBS draft specification [i.177].

4.4.7 Extensions of multi-messages signature schemes

The multi-messages signature schemes described in clauses 4.4.1 to 4.4.5 are based on the classic approach for building (Q)EAAs from a set of advanced cryptographic mechanisms such as BBS+, CL or PS-MS signatures. While this approach does support selective disclosure, it comes with the cost of concealing the undisclosed attributes in a zero-knowledge proof whose complexity grows linearly with the number of such attributes. In order to minimize the size of the (Q)EAAs and their verifiable presentations, more elaborate approaches have been proposed for BBS+ and PS-MS, where undisclosed attributes have no impact on the proof size, which is beneficial for selective disclosure. Below are three cryptographic research papers that describes such approaches:

- "MoniPoly: An Expressive q-SDH-Based Anonymous Attribute-Based Credential System" [i.240] published by Syh-Yuan Tan and Thomas Gross (2020).
- "Efficient Redactable Signature and Application to Anonymous Credentials" [i.232] published by Olivier Sanders (2020).
- "Improving Revocation for Group Signature with Redactable Signature" [i.233] published by Olivier Sanders (2021).

4.5 Proofs for arithmetic circuits (programmable ZKPs)

4.5.1 General

Arithmetic circuits can represent any computational logic. Consequently, proofs for arithmetic circuits are "programmable ZKPs": As every statement can be translated into an arithmetic circuit, a ZKP for any statement can be constructed. The programmable ZKPs are often designed and implemented as zk-SNARKs, which are further described in clause 4.5.2.

4.5.2 zk-SNARKs

4.5.2.1 Introduction to zk-SNARKs

The abbreviation zk-SNARK stands for "Zero-Knowledge Succinct Non-interactive ARgument of Knowledge", and is a collaborative term for a specific category of ZKP protocols. At the time of writing (in August 2025), eighteen zk-SNARK protocols have been published by cryptographic researchers; see clause A.4 for a list of all zk-SNARK protocols.

The zk-SNARK characteristics can be broken down as follows:

- zero-knowledge: As defined earlier, the proof gives no information beyond that the statement is correct, and any information that can be trivially derived from the statement (e.g. a ZKP that the statement that a holder is older than 19 is correct trivially proves also that the holder is older than 18).
- Succinct: the proof size grows sublinearly with the statement's size (e.g. logarithmically or even independent of statement size (constant proof size)).
- Non-interactive: randomness is not provided by the verifier (but by a random oracle). Consequently, a single message from the prover suffices to convince any verifier.
- ARgument: Cryptographic evidence (that relies on some battle-tested computational hardness assumptions such as DLP, as opposed to a full mathematical proof).
- of Knowledge: the proof demonstrates the user's knowledge of data (a witness) that proves the statement (not just its existence).

NOTE 1: A zk-SNARK system provides selective disclosure, unlinkability, and predicate proofs by design.

NOTE 2: The succinctness property is not necessary for privacy in verifiable presentations. In contrast, as shorter proof sizes and verification complexity typically require more resources on the prover side, and the hardware running the prover is commonly a mobile phone, which is much more resource-constrained than a relying party's server, the succinctness is probably not a desirable property in an implementation of general-purpose ZKPs for digital wallets. This resonates well with more recent and efficient constructions that use (non-succinct) zk-NARKS. Besides proving time, memory requirements can pose an issue on mobile devices. While many zk-(S)NARK constructions have very high memory requirements (scaling with the size of the overall transcript of the witness verification algorithm to be proved), some constructions, such as Ligero, allow for garbage collection and hence scale memory requirements only with the maximum memory required during running the witness verification algorithm.

The concept of zk-SNARK was initially described by Alessandro Chiesa et al. in a paper [i.65] in 2012, which in turn was based on Jens Groth's work [i.125] from 2010. The first general or programmable zk-SNARK protocol Pinocchio [i.220] was designed and implemented in 2013. Hence, a zk-SNARK that is correctly executed (e.g. with a C program) can efficiently create specific ZKPs for any statement.

There is an important distinction between zk-SNARK proving systems that require a program (circuit)-specific preprocessing. So far, mainly preprocessing SNARKs have been used in practice (blockchain privacy and scaling projects) because they tend to have higher proving performance as they can be hand-optimized to the program. However, for different programs (e.g. patches) the preprocessing needs to be conducted again. On the other hand, so-called zero-knowledge Virtual Machines (zkVMs) can dynamically prove the correct execution of any program (represented by an instruction set received through compilation, e.g. a C or Rust program compiled with LLVM). Promising candidates that would allow the dynamic execution of certificate verification include a zk-WASM (not yet a fully-fledged VM) based on the Ligero [i.7] proof system, called Ligetron.

NOTE 3: In the zkVM case, there is also a preprocessing step, but it is only instruction set specific and, therefore, not program-specific.

A zk-SNARK protocol can be based on a trusted setup or as a transparent setup, as further described in clauses 4.5.2.2 and 4.5.2.3.

4.5.2.2 Trusted setup of zk-SNARKs

The trusted setup of a zk-SNARK involves three algorithms *KeyGen*, *CP*, *CV* as illustrated in Figure 8.

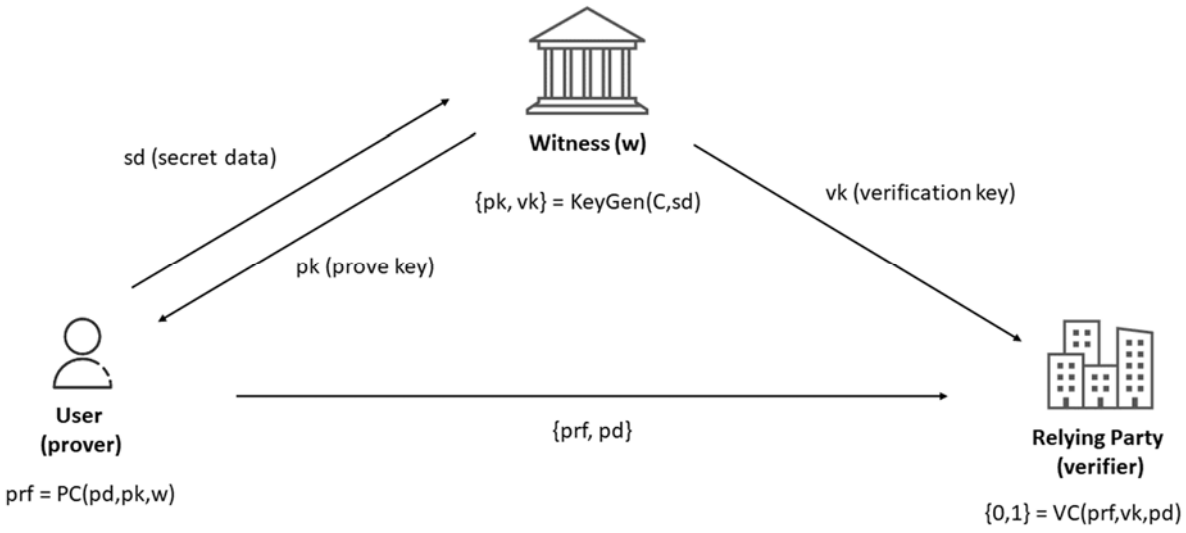


Figure 8: Overview of zk-SNARK with trusted setup

The key generator *KeyGen* takes a secret parameter *sd (secret data)*, also called "toxic waste", and the program *C* for which correct execution should be proven (the statement), and generates two publicly available keys, the user's proving key *pk*, and the relying party's verification key *vk*. These keys are public parameters that need to be generated once for a specific program *C*.

NOTE 1: The parameter *sd* used in the generator is a secret value. If this parameter is known to an attacker, it can generate fake proofs, i.e. without knowing a valid witness *w*. In other words, the soundness guarantees of the zk-SNARK would not be satisfied any more. However, the zero-knowledge property is not conditional on the secrecy of sd. In the context of digital attestations, even a citizen that does not trust the entity that ran the trusted setup need not to be afraid of a loss of privacy guarantees.

NOTE 2: To make sure that sd cannot be leaked, many projects (particularly on blockchains where whoever runs the trusted setup will unlikely be trusted by everyone), the trusted setup is usually operated in a multi-party computation by many entities, such that *sd* is only leaked if all of these entities collude. As such, if a verifier trusts only a single entity involved in the trusted setup, soundness of the zk-SNARK system is guaranteed, i.e. no fake proofs can be practically created.

NOTE 3: In principle, each relying party (verifier) could run their own trusted setup and distribute the corresponding pk to the holder: If the verifier protects their sd, they do not need to be afraid of receiving fake proofs. However, there are two significant drawbacks: pk tends to be large for practical presentations (tens to hundreds of MB), so real-time distribution is impractical and a pk that all verifiers accept is more desirable (particularly because different presentations correspond to different programs and, therefore, require different *pk*). Furthermore, as the holder cannot check the setup conducted by the verifier, additional certification of the pk to make sure it is derived from the correct program (and not some other program that outputs more information than stated), allowing a user to trust in the privacy guarantees.

The user executes the algorithm *CP* with the following input parameters: its (static) proving key *pk*, a (dynamic) public input *pd (public data)*, and a private witness *w*. The algorithm *CP* generates the proof value *prf = CP(pk, pd, w)*, as evidence that the user knows a witness *w*.

EXAMPLE 1: The public data *pd* could be the statement, for example that the user's age is above 18. It will also likely involve a nonce to avoid replay attacks and a set of public keys for accepted issuers against which the signature of the user's attestation (which represents part of the witness) is verified in the zk-SNARK.

The verifying relying party calculates the algorithm *CV(vk, pd, prf)* which returns true if the proof is correct and false otherwise. Hence, the function *CV* returns true if the user knows a witness *w* that satisfies the function *C(sd,w) = true*.

EXAMPLE 2: zk-SNARK protocols with trusted setup are Pinocchio [i.220], Geppetto [i.72], and TinyRAM [i.19]. For a complete list of zk-SNARK protocols with trusted setups, see table A.4 in clause A.4.

NOTE 4: Most zk-SNARKs with trusted setup actually involve a two-step trusted setup: one that is not dependent on C and a second one that is dependent on C. In 2019, PLONK [i.116] was introduced as a universal zk-SNARK protocol. In this approach, only the first step which is independent of C involves toxic waste that may compromise soundness; and the second, C-dependent step - while involving a computationally intensive preprocessing step - does not involve toxic waste anymore but only relies on the output of the first step. However, the "complexity" of the programs C that can be covered is bounded by the sizes covered by the first step.

Universal trusted setup: In 2019, PLONK [i.116] was introduced as the universal zk-SNARK protocol.

4.5.2.3 Transparent setup zk-SNARKs

In a transparent (public) setup of zk-SNARK there is no need for a trusted setup. Yet, to achieve succinctness, a computationally and memory-intensive preprocessing step is still required.

EXAMPLE: zk-SNARK protocols with transparent (public) setups are SuperSonic [i.198], Hyrax [i.250] and Halo [i.31]. For a complete list of zk-SNARK protocols with transparent setups, see table A.4 in clause A.4. Moreover, hash-based zk-(S)NARK, such as Ligero, are often not succinct but still sublinear in proof size and/or verification time and hence require a transparent setup.

NOTE: If the general-purpose ZKP is transparent and not succinct, the transparent setup may be as simple as specifying the cryptographic hash function used in the construction.

4.5.2.4 Cryptography behind zk-SNARKs

The cryptography that underpin the zk-SNARK schemes is highly complex and differs from protocol to protocol.

In brief, the zk-SNARK protocols can be constructed based on the following cryptographic building blocks [i.222]:

- Fiat-Shamir Heuristics, which in turn can be broken down into Sigma-Protocols, Random Oracle Models (ROM) and Fiat-Shamir-Compatible Hash Functions.
- Probabilistically Checkable Proofs (PCP): Merkle Trees and Hash Functions, Kilian Interactive Argument of Knowledge, and Micali's Computationally Sound (CS) Proof.
- Quadratic Arithmetic Programs (QAPs) and Square Span Programs (SSPs).
- Linear Interactive Proofs (LIPs).
- Polynomial Interactive Oracle Proofs (PIOPs).

A common construction involves three steps:

1) Arithmetization: Representing the program *C* as a sequence of simple algebraic operations, such as additions and multiplications. Common representations are Rank-1 Constraint Systems (R1CS), PLONKish, and Algebraic Intermediate Representation (AIR).

2) This representation is translated into one or multiple polynomials, such that knowledge of a witness, corresponding to a valid execution trace of *C*, corresponds to certain properties of the polynomials (e.g. roots at certain positions or equalities between one polynomial and a product of two other polynomials). Challenging this equality under the assumption of a truthfully answering prover corresponds to an Interactive Oracle Proof (IOP). The IOP is an information-theoretic object, i.e. it does not rely on cryptographic hardness assumptions. Because of the good error-amplification of polynomial encodings following the Schwartz-Zippel lemma (polynomials of low degree in a large field will either be equal or different in almost every point), few spot checks are sufficient, with the corresponding points for the spot checks determined using the Fiat-Shamir heuristic.

3) Using a cryptographic Polynomial Commitment Scheme (PCS), the prover can be forced to answer truthfully to queries of these polynomials (which are not shared by the prover). The PCS is responsible for the transparency properties of the setup (trusted or transparent) and the reason why a "proof" based on a PCS becomes an "argument".

NOTE 1: Depending on the IOP and PCS, some zk-SNARKs are not post-quantum secure, i.e. soundness guarantees rely on hardness assumptions such as DLP. As for the toxic waste, the zero-knowledge property is, by contrast, unconditional.

NOTE 2: Bulletproofs [i.38] - developed by Bünz et al. - are a family of zk-SNARKs with reduced succinctness properties (proof size is sublinear, but verification time is not). See clause 4.5.4 for more information about Bulletproofs.

NOTE 3: zk-STARKs [i.17] and [i.203] - developed by Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev [i.18] - are a family of transparent zk-SNARKs that are plausibly post-quantum secure, i.e. soundness guarantees plausibly hold against an adversary with a quantum computer. They are instantiated with a specific arithmetization (AIR) and IOP-PCS combination (Fast Reed Solomon IOP - FRI) that relies on low-degree testing of polynomials and Merkle trees for opening polynomials on small subgroups. Because of their FRI-based construction, proof sizes of zk-STARKs are around 100 to 1 000 times higher than proof sizes of the shortest zk-SNARKs. See clause 4.5.3 for more information about zk-STARKs.

Given the vast literature of zk-SNARK algorithms, a complete description of the cryptography for zk-SNARKs goes beyond the scope of the present document. For further reading about the cryptographic algorithms behind the zk-SNARK protocols, the following papers are recommended: Nitulescu "zk-SNARKs: A Gentle Introduction" [i.205], Petkus "Why and How zk-SNARK Works: Definitive Explanation" [i.222], and Evans "Succinct Proofs and Linear Algebra" [i.107].

4.5.2.5 Implementations

As regards to implementations, zk-SNARK was implemented in 2016 for the blockchain protocol ZeroCash for cryptocurrency ZCash, for which zk-SNARK caters for four different transaction types: private, shielding, deshielding, and public. Hence, zk-SNARK allows the users to determine how much data to be shared with the public ledger for each transaction. The blockchain Ethereum zk-Rollups also utilizes zk-SNARKs to increase its scalability. In doing so, they do not make use of the zero-knowledge property but the succinctness property, so some zk-rollups, in fact, are based on SNARKs and not on zk-SNARKs. Furthermore, zk-SNARKs have been implemented as general-purpose ZKP schemes in combination with existing digital identities, as described in clause 6.5.

4.5.2.6 Cryptographic analysis

Whether a zk-SNARK protocol is quantum-safe or not depends on the underlying cryptographic algorithms, as described in table A.4. The zk-SNARK protocols Aurora [i.20], Ligero [i.7], Spartan [i.200], and Virgo [i.273] are considered as plausible quantum-safe (related to soundness), whilst the others in table A.4 are not considered as quantum-safe.

It is possible to implement presentations of (Q)EAA using zk-SNARKs that support fully unlinkable attestations.

NOTE 1: Succinct proofs can typically be turned into ZKPs quite easily through adding blinding factors, since a succinct proof already eliminates a lot of superfluous information ("there cannot be much sensitive information left"). In the context of the EUDIW, the succinctness property is arguably not very relevant because the complexity of the statement to be proved is low enough to be handled directly by a mobile phone. Hence, it makes a lot of sense to look into programmable ZKPs beyond zk-SNARKs. Yet, because of their limited computational power, the focus of the blockchain project has lied on succinct proofs, such that progress and industry-grade tooling is arguably most advanced there.

NOTE 2: It is possible to combine ZKPs based on CL-signatures or BBS(+) with proofs for arithmetic circuits. For instance, BBS can be used for a proof of knowledge of the issuer's signature and reveal commitments to selected attributes. Then, a programmable ZKP (e.g. a zk-SNARK) can be used to prove certain properties of the identity attribute (the preimage of the revealed hash), e.g. to compute a complex predicate. A well-known construction that follows this paradigm is LegoSNARK [i.49], implemented in the context of digital attestations, among others, by dock.io.

4.5.3 zk-STARKs

4.5.3.1 Introduction to zk-STARK

The abbreviation zk-STARK stands for "Zero-Knowledge Succinct Transparent Arguments of Knowledge", and is a collaborative term for a specific category of Zero-Knowledge Proof protocols. The zk-STARK protocols fulfil the criteria of a Zero-Knowledge Proof system, which enables one party (the prover) to prove to another party (the verifier) that a certain statement is true, without revealing any additional information beyond the truth of the statement itself. Furthermore, zk-STARKs are succinct, such that they allow for the creation of short proofs that are easy to verify, and they are transparent, meaning that anyone can verify the proof without needing any secret information.

The zk-STARK characteristics can be broken down as follows (based on the initials S-T-ARK) to cater for zero-knowledge (zk):

- Scalable: the prover algorithm is typically implemented with repeated functions (e.g. several hash functions).
- Transparent: the prover and verifier keys are generated verifiably in a trustless manner (i.e. without the need of a trusted setup).
- ARgument of Knowledge: a proof system that demonstrates the user's knowledge of data (not just its existence).

NOTE: A zk-STARK system provides predicate proofs, selective disclosure and unlinkability by design.

The concept of zk-STARK was initially described by Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev in a paper [i.18], 2018. At the time of writing (in August 2025), two zk-STARK protocols have been published by cryptographic researchers:

- the zk-STARK protocol [i.17] in 2019; and
- Zilch [i.203] in 2021.

4.5.3.2 Setup of zk-STARK

Unlike the zk-SNARK frameworks, which in several cases require a trusted setup, the zk-STARK protocols are designed to be used without a trusted setup. Hence, the zk-STARK protocols are considered to be both transparent and universal: a transparent protocol is defined as it does not require any trusted setup and uses public randomness, and a universal protocol is defined as it does not require a separate trusted setup for each circuit.

4.5.3.3 Cryptography behind zk-STARK

The cryptography behind the zk-STARK schemes is based on Interactive Oracle Proofs (IOP) with scalable proofs.

A Zero-Knowledge system based on IOP (ZK-IOP) [i.17] is a common generalization of the Interactive Proofs (IP), Probabilistically Checkable Proofs (PCP) and Interactive PCP (IPCP) models that were previously introduced for zk-SNARKs (see clause 4.5.2).

The zk-STARK protocols are typically implemented using standard hash functions. As in the PCP model, the IOP verifier does not need to read all prover messages, but can rather query them at random locations; as in the IP model, prover and verifier interact over several rounds. Hence, a ZK-IOP system can be converted into an interactive ARgument of Knowledge (ARK) model, assuming a family of collision-resistant hash functions can be turned into a non-interactive argument in the random oracle model, which is typically realized using a standard hash function.

Given the complexity of zk-STARK algorithms, a complete description of the cryptography for zk-STARK goes beyond the scope of the present document. For further reading about the cryptographic algorithms behind the zk-STARK framework, the following paper is recommended: Ben-Sasson et al. "Scalable, transparent, and quantum-safe computational integrity" [i.17].

4.5.3.4 Implementations

While the zk-STARKs developed by StarkWare are a prominent instantiation, they have been criticised for relatively low security (Starkware's construction amplifies security by a proof of work [i.241]) - among other reasons because the concrete choice of security level is based on an additional unproven conjecture to allow for fast proving times. Moreover, the FRI-based IOP and the Merkle tree-based PCS scheme are not the only way to construct transparent, post-quantum secure zk-SNARKs. Other protocols also use different IOP paradigms or PCS. For example, Aurora [i.20] is a zk-SNARK based on Interactive Oracle Proofs (IOPs) over Rank-1 Constraint Systems (R1CS), using the sum check protocol. Like the traditional zk-STARKs, it relies on polynomial commitments via Merkle trees, and hence does not require a trusted setup and is considered plausibly post-quantum secure. FRACTAL [i.67] is another transparent zk-SNARK that achieves both post-quantum security and recursive composition efficiency. It is also based on FRI as an IOP but uses different encoding techniques and soundness amplification strategies compared to zk-STARKs. Lastly, while constructions like Spartan and Ligero are not succinct, they still involve relatively small proofs (square root scaling, as opposed to poly-logarithmic) and share the transparency and post-quantum security characteristics of zk-STARKs based on FRI + Merkle-tree based polynomial commitments. These examples show that zk-STARKs represent a well-developed point in a larger design space of transparent and plausibly post-quantum secure zk-SNARKs - underscoring that transparency and post-quantum security are properties that can be achieved in multiple ways, not just via the zk-STARK construction lineage.

Potentially, zk-STARKs could replace zk-SNARKs for various applications in the future. For example, zk-STARKs could be used for the privacy and confidentiality of ZeroCash protocol, which is currently implemented with zk-SNARK. However, zk-SNARKs are roughly 1 000 times shorter than zk-STARK proofs, so replacing zk-SNARKs with zk-STARKs would require more research to either shorten proof length, or aggregate and compress several zk-STARK proofs using incrementally verifiable computation [i.17].

4.5.3.5 Cryptographic analysis

It makes sense to consider zk-STARKs as a special category of zk-SNARKs because they fulfil the same fundamental purpose - namely, enabling succinct ("scalable"), non-interactive zero-knowledge proofs - but with a distinct set of design trade-offs, particularly in terms of cryptographic assumptions, proof system architecture, and transparency. zero-knowledge Scalable Transparent ARguments of Knowledge (zk-STARKs) differ from traditional zk-SNARKs mainly in that they forgo trusted setup (i.e. they are transparent) and are constructed from information-theoretic rather than algebraic assumptions, relying only on collision-resistant hash functions for designing the polynomial commitment scheme. Since they avoid assumptions such as knowledge of exponent underlying Groth16 or elliptic curve pairings underlying the KZG commitment scheme underlying many of the popular zk-SNARKs, they are also plausibly post-quantum secure (in terms of soundness, as the privacy guarantees are discussed broadly above). These properties position zk-STARKs as a natural subclass of zk-SNARKs, with additional guarantees, rather than a completely separate lineage. Grouping them this way emphasizes their role within the broader zk-SNARK family, defined by succinctness and non-interactivity, while allowing meaningful differentiation based on the underlying cryptographic assumptions and protocols.

The zk-STARK schemes are considered as plausible quantum-safe, since they are based on a machinery of hash functions for implementing the IOP. If the used hash functions are designed as QSC, the zk-STARK scheme becomes quantum-safe.

4.5.4 ZK Bulletproofs

In their paper, "Bulletproofs: Short Proofs for Confidential Transactions and More" [i.38], Bünz et al. (2017) introduce a non-interactive ZKP protocol aimed to address the issue of transaction size and verification time in existing privacy preserving protocols. Specifically aiming to improve upon proposals for confidential transactions in cryptocurrencies, bulletproofs support aggregation of range proofs and require no trusted setup.

A Bulletproof is a zero knowledge inner product argument. Specifically, it enables a prover to prove the correct computation of an inner product of two vectors a = [a1, …, an], and b = [b1, …, bn] such that v = ⟨a, b⟩ = a1b1 +...+ anbn. The prover can do so optionally hiding the vectors or the inner product result. The verifier receives Pedersen commitments to the input vectors and their resulting inner product, together with a proof, π, it can use to verify the commitments and the honest and correct computation.

By computing multiple inner products, it is possible to compute proofs for R1CS formatted circuits directly using only EC point addition. The size of the circuit can be limited in many contexts improving the performance of bulletproofs significantly. For instance, in the context of age verification, an 8-bit circuit is enough. Here, the inner product of the bit vector representation of the users age, a, and the power of two vectors equals the users age. And it is easy to see that the inner product v = ⟨[a1, …, a7], [1, 2, 4, 8, 16, 32, 64, 128]⟩ effectively constraints this value to the range $v \in [0,255]$.

The inner workings of a Bulletproof is rather lengthy to detail here due to various optimizations used, but the core building block is the Pedersen commitment. Herein, it is enough to state that using a series of Pedersen commitments, the prover can prove a commitment to a polynomial and its correct evaluation at some value u. This is then used to create a ZKP of polynomial multiplication, which is important because it provides a way to create a ZKP of scalar multiplication as follows:

1) With commitment A to a and b, and commitment V to v, where v = ab, it is possible to prove that A and V are committed as claimed without revealing a, b, or v.

2) Prover choses sL and sR randomly and adds these as linear terms, i.e. a becomes a + sLx and b becomes b + sRx.

3) Multiplication of ab is given by the polynomial multiplication (a + sLx)(b + sRx) = ab + linear term + quadratic term.

4) Verification is then done using the prover supplied commitment A and V.

Again, the details are rather lengthy. Suffice to say is that there exists an easy way to create a ZKP of scalar multiplication using a ZKP of polynomial multiplication; specifically by only focusing on the constant terms of the polynomial. Equipped with a ZKP of multiplication, it is easy to create a ZKP for the inner product by changing the coefficients from scalars to vectors, and commitments from scalar commitments to vector commitments.

That is the essence of the Bulletproof ZK inner vector argument. The precise steps look a lot more complex as they include an optimization to create logarithm-sized proofs of knowledge for inner products. Furthermore, when using Bulletproof ZK inner product argument for a range proof, multiple inner products are proven with one proof using a random linear combination. There are also additional constraints to enforce that the user honestly and correctly computes the inner product of a bit vector and the power of two vectors. All of these add complexity.

In the context of the EUDIW and evaluating the relational predicate m <x < n in a privacy preserving way, Bulletproof ZK inner product proofs can be used in the following way:

1) The issuer creates the commitment $V = vG + \gamma B$ where G and B are two EC points with unknown discrete log relationships, v is the value (e.g. user age in days), and γ is a random blind (shared secret between issuer and user).

2) The issuer or the user creates commitments to the binary vector representation of v (and the associated proof that the vector truly is binary), and commitments to the required vector polynomials.

3) The verifier responds with a random challenge pair (can be made non-interactive using Fiat-Shamir) that the prover uses to combine the three inner products into one.

4) The prover shares Pedersen commitments to the linear and quadratic coefficients.

5) The verifier responds with a random challenge u that the prover uses to evaluate the polynomial.

6) The verifier can now check that the inner product of the binary vector and the power of two vector equals the value commitment V, and that the evaluation is correct and honest. This serves as a proof that the committed value lies in the range [0, 2n).

For some applications, range proofs of the form $v \in [0, 2^n - 1]$ need to be adjusted to instead have some lower positive bound and an upper bound that is not a power of two. This requires shifting the commitment value as follows:

- Given a lower bound, l, the lower bound shift can be accomplished by $V - lG = (v - l)G + \gamma B$. The prover can now use the shifted committed value to prove the range $(v - l) \in [0, 2^n)$.

- Given an upper bound,$2^n - u$, the upper bound shift can be accomplished by adding u to the initial commitment $V + uG = (v + u)G + \gamma B$. The prover can now use the shifted committed value to prove the range $(v + u) \in [0, 2^n)$.

- Taken together, a tighter combined range is accomplished as $v \in [l, 2^n - u)$. The two bound proofs can be aggregated into a single proof as described in Bünz et al. [i.38] section 4.3.

The primary goal of Bulletproofs is to provide a compact and efficient way of proving the correctness of a transaction, while hiding the specific details of the transaction itself. Bünz et al. [i.38] do mention other uses, including support for arithmetic circuits, verifiable shuffles (i.e. to prove that one list of committed values is a shuffle of another list of committed values), and privacy preserving smart contracts in public blockchains. Each of these uses, however, can be done more efficiently in contexts with different contextual characteristics than those of decentralized cryptocurrencies. It is not immediately apparent if Bulletproofs are relevant for electronic attestations of attributes/person identification data given that more performant options like Hashwires exist. Further exploration or analysis may be needed to fully understand how Bulletproofs could be directly applicable to electronic attestations of attributes or person identification data.

5 (Q)EAA formats with selective disclosure

5.1 General

The present clause provides an analysis of a set of formats for selective disclosure.

The topics for the analysis of each selective disclosure (Q)EAA formats are:

- Signature scheme(s) used for selective disclosure and optionally unlinkability, when applicable with references to clause 4.

- Encoding of the (Q)EAAs used for selective disclosure.
- Maturity of the (Q)EAA format's specification and deployment.
- Cryptographic aspects, more specifically if the cryptographic algorithms used for the selective disclosure (Q)EAA formats are approved by SOG-IS and allows for QSC algorithms for future use.

The (Q)EAA formats are categorized according to three of the main cryptographic schemes for selective disclosure:

- Atomic (Q)EAA formats, see clause 5.2. These (Q)EAA formats correspond to the (Q)EAA signature schemes described in clause 4.2.
- Multi-message signature (Q)EAA formats, see clause 5.4. These (Q)EAA formats correspond to the multi-message signature schemes described in clause 4.4.
- (Q)EAAs with hashes of salted attributes, see clause 5.3. These (Q)EAA formats correspond to the multi-message signature schemes described in clause 4.3.

NOTE 1: There is also a type of generic JSON container format (JSON WebProofs), which allows for a mix of the selective disclosure signature schemes in clause 4, and is therefore treated as a separate category of (Q)EAA formats.

NOTE 2: The proofs for arithmetic circuits (such as zk-SNARKs) do not rely upon (Q)EAA formats per se, as they can prove the correct execution of any credential verification program in zero-knowledge. Hence, proofs for arithmetic circuits are out of scope for the present clause, which describes (Q)EAA formats. However, clause 6.5 describes solutions that are implemented based on a combination of programmable ZKPs (such as zk-SNARKs) with existing credentials (such as X.509 certificates).

5.2 Atomic (Q)EAA formats

5.2.1 Introduction to atomic (Q)EAA formats

The concept of atomic (Q)EAAs was introduced in clause 4.2. There are numerous (Q)EAA formats that can be issued with a single claim, so in principle a selective disclosure scheme based on atomic claims can be designed for a variety of types of (Q)EAA formats (ICAO DTCs, IETF JWTs, W3C Verifiable Credentials, X.509 certificates, etc.).

Clauses 5.2.2 and 5.2.3 are however focusing in more detail on two (Q)EAA formats that are used for atomic (Q)EAA schemes: PKIX X.509 attribute certificates and W3C Verifiable Credentials.

5.2.2 PKIX X.509 attribute certificate with atomic attribute

The PKIX X.509 Attribute Certificate (AC) profile is specified in IETF RFC 5755 [i.158]. An attribute certificate may contain attributes that specify group membership, role, security clearance, or other authorization attributes associated with the user. The attribute certificate is a signed set of attributes, although it does not contain a public key. Instead, the attribute certificate is linked to a X.509 Public Key Certificate (PKC), which can be used by the user for authentication. In order to preserve the user's privacy, the X.509 public key certificate may only include a pseudonym in the subject field.

The attribute certificates are issued by an Attribute Authority (AA), and they may be issued with a short lifetime and with an atomic (single) attribute. These characteristics make short-lived attribute certificates with atomic credentials suitable for an access control service with selective disclosure features.

A description of how to use PKIX X.509 attribute certificates for selective disclosure with an access control system is available in clause 6.2.1.

The X.509 attribute certificates are ASN.1/DER encoded as described in IETF RFC 5755 [i.158].

X.509 certificates can be signed by the QTSP using cryptographic algorithms (RSA with proper key lengths or ECC with approved curves) that are published by SOG-IS [i.237]. For future use, the X.509 certificates can be signed with quantum-safe cryptographic algorithms [i.193].

The maturity of X.509 attribute certificates can be considered as high, given that the IETF RFC 5755 [i.158] is a mature PKIX standard.

5.2.3 W3C Verifiable Credential with atomic attribute

As a preparation for enrolment of W3C Verifiable Credentials with atomic attributes, the EUDI Wallet would need to be equipped with Credential templates for the W3C Verifiable Credentials. The W3C Verifiable Credentials Data Model v1.1 [i.264] distinguishes between a Credential as "*a set of one or more claims made by an issuer*" and a Verifiable Credential as "*a verifiable credential is a tamper-evident credential that has authorship that can be cryptographically verified*". Put differently, a Verifiable Credential can be a signed Credential. Hence, the Credential(s) in the EUDI Wallet can consist of templates with the attribute properties that should be used for the enrolment of attribute values.

NOTE: The W3C Verifiable Credentials Data Model v1.1 [i.264] is a conceptual data model rather than a specific credential format. In this context of atomic attributes, however, the scope of W3C Verifiable Credentials can be limited to the JWT format.

A description of how to use the FIDO standard [i.56] as an authentication protocol in conjunction with Verifiable Credentials with atomic attributes for selective disclosure is available in clause 6.2.2.

The encoding of the W3C Verifiable Credentials is specified as JWT or JSON-LD in the W3C Verifiable Credentials Data Model v1.1 [i.264].

W3C Verifiable Credentials can be signed by the QTSP using cryptographic algorithms (RSA with proper key lengths or ECC with approved curves) that are published by SOG-IS [i.237]. For future use, the W3C Verifiable Credentials can be signed with quantum-safe cryptographic algorithms as described in the IETF report on JOSE signatures with QSC algorithms [i.149].

The maturity of W3C Verifiable Credentials can be considered as high, given the wide deployment of issued W3C Verifiable Credentials.

5.3 Formats of (Q)EAAs with salted attribute hashes

5.3.1 General

The general concept of selective disclosure based on salted attribute hashes is described in clause 4.3. As regards to credentials within this category, there are several noteworthy formats. The formats that are described more in-depth in the present document are:

- IETF SD-JWT, which is further described in clause 5.3.2.
- ISO mDL Mobile Security Object (MSO), which is elaborated in clause 5.3.3.

NOTE: ETSI EN 319 162-1 [i.88] specifies the Associated Signature Containers (ASiC), which is an XML-formatted manifest that binds together a number of hashed file objects into one single digital container. The principle of combining hashed objects in an ASiC manifest is similar to the IETF SD-JWT and ISO mdoc credentials with salted attribute hashes. There are however two main differences:

  - ETSI ASiC is intended for combining file objects in a signature container manifest, whilst IETF SD-JWT and ISO mDL MSO are designed for selective disclosure.
  - Furthermore, the ETSI ASiC hashes are not salted, whilst the hashed attributes in IETF SD-JWT and ISO mDL MSO are salted to cater for unlinkability. Hence, the comparison with ETSI ASiC is observed, but nevertheless out of scope for the present clause.

In addition to the above two formats, the present document also includes a mention of disclosure mechanisms based on proof mechanisms detailed in JSON Web Proofs and describes a proposal that relies on Directed Acyclic Graphs (DAG).

5.3.2 IETF SD-JWT and SD-JWT VC

5.3.2.1 IETF SD-JWT

To support selective disclosure in JWT or JWS, IETF has specified Selective Disclosure JSON Web Token (SD-JWT) [i.155]. The specification introduces two primary data formats, an SD-JWT which is a composite structure consisting of a JWS plus optionally disclosures, and an SD-JWT+KB which is a composite structure of an SD-JWT and a Key Binding JWT (KB-JWT) that is used as a proof of possession for a private key corresponding to a public key embedded in the SD-JWT.

At its core, an SD-JWT is a digitally signed JSON document that can contain salted attribute hashes that the user can selectively disclose using disclosures that are outside the SD-JWT document. This allows the user to share only those attributes that are strictly necessary for a particular service. The technique of SD-JWT is based on salted attribute hashes as described in clause 4.3.

Each SD-JWT contains a header, payload, and signature and optionally disclosures. The header contains metadata about the token including the type and the signing algorithm used. The signature is generated using the issuer's private key. The payload includes the proof object that enables the selective disclosure of attributes. Each disclosure contains a salt, a cleartext claim name, and a cleartext claim value. The issuer then computes the hash digest of each disclosure and includes each digest in the attestation it signs and issues.

NOTE: The JOSE [i.169] signature format allows for SOG-IS approved cryptographic algorithms [i.237] and QSC algorithms [i.149] for future use.

During presentation, a Holder selects the disclosures they want to reveal, if the SD-JWT is bound to a key, produces a proof of possession that also signs over the revealed disclosures (KB-JWT), and presents the composite of SD-JWT and KB-JWT to the verifier.

The SD-JWT specification is still a draft, yet SD-JWT has been selected in the ARF [i.71] as the JSON-format for selective disclosure and is in the final stages of the IETF standardization process.

A thorough analysis of SD-JWT and how it can be applied for selective disclosure of the PID/(Q)EAA for the EUDI Wallet is available in clause E.1.

5.3.2.2 IETF SD-JWT VC

While SD-JWT defines the general container format, SD-JWT-based Verifiable Credentials (SD-JWT VC) defines a data format and validation rules to express JSON based Credentials based on SD-JWT. This is a usual pattern where a general container format is defined (e.g. JWT) and based on that container format concrete data formats are defined (e.g. Access Token, ID Token).

SD-JWT VC defines a set of mandatory and optional claims that have not to be selectively disclosable to enable different additional features such as a way to resolve additional issuer metadata, a credential type mechanism, and a status (revocation) mechanism. SD-JWT VC does not fundamentally change the underlying mechanisms of SD-JWT, but allows for the creation of a digital credential ecosystem on top of it by adding essential mechanisms for such ecosystems that allow for type based filtering, credential revocation, issuer key discovery, and additional display information.

5.3.3 ISO/IEC 18013-5 Mobile Security Object (MSO)

The Mobile Security Object (MSO) is specified in clause 9.1.2.4 of ISO/IEC 18013-5 [i.181] and contains the following attributes encoded in a CDDL [i.170] structure:

- digestAlgorithm: Message digest algorithm
- valueDigests: Array of digests of all data elements
- deviceKey: Device key in COSE_Key as defined in IETF RFC 8152 [i.167]
- docType: DocType as used in Documents
- validityInfo: validity of the MSO and its signature

The valueDigests are issued as IssuerSignedItems, which are the hash values of the ISO mDL attributes combined with random values (see ISO/IEC 18013-5 [i.181], clause 9.1.2.4). In other words, the MSO is a selective disclosure standard based on salted hashes of attributes (see clause 4.3), where the random values are the salts.

The deviceKey contains the mdoc Authentication Key (see clause 7.2.2), which is protected by the user's PIN-code or biometrics (see clause 7.6).

The MSO is signed by the mDL Issuer Authority, which is an IACA X.509 CA (see clause 7.2.1.4), and the signature is COSE formatted.

NOTE 1: ISO/IEC 18013-5 [i.181], Table B.3 "Document signer certificate" lists the ECDSA curves BrainpoolP256r1, BrainpoolP384r1 and BrainpoolP512r1, which are also approved by SOG-IS [i.237].

NOTE 2: The COSE [i.162] signature format also allows for QSC algorithms [i.149] for future use.

An example of an MSO data structure is provided in ISO/IEC 18013-5 [i.181], annex D.5.2.

The MSO is stored and protected in the device's SE/TEE. The MSO is included in the mDL Response for the device retrieval flow (see clause 7.2.3).

ISO/IEC 18013-5 [i.181] is considered mature, and several device retrieval solutions with MSOs have been deployed in production, for example in a number of states in the US.

A thorough analysis of ISO mDL MSO and how it can be applied for selective disclosure of the PID/(Q)EAA for the EUDI Wallet is available in clause 7.2.

5.4 Multi-message signature (Q)EAA formats

5.4.1 W3C VC Data Model with ZKP

The W3C Verifiable Credentials (VC) Data Model v1.1 [i.264] contains clause 5.8 "Zero-Knowledge Proofs", which describes a data model that supports selective disclosure with the use of Zero-Knowledge Proof (ZKP) mechanisms.

The W3C Verifiable Credentials Data Model states two requirements for Verifiable Credentials when they are to be used in ZKP systems:

- The Verifiable Credential contains a proof, so that the user can derive a verifiable presentation that reveals only the information that the holder intends to reveal.
- The credential definition (if being used) is defined in the JSON credentialSchema property, so that it can be used to perform various cryptographic operations in zero-knowledge.

The following cryptographic schemes that support selective disclosure while protecting privacy across multiple presentations have been implemented for the W3C Verifiable Credentials Data Model [i.264]: IRTF CFRG BBS [i.177], CL Signatures [i.42], Idemix [i.136], Merkle Disclosure Proof 2021 [i.259], Mercurial Signatures [i.45], PS Signatures [i.223], U-Prove [i.3] and Spartan [i.234].

More specifically, the W3C Verifiable Credentials Data Model standard includes examples of how to use Camenisch-Lysyanskaya (CL) signatures (see clause 4.4.1) with a W3C Verifiable Credential and a W3C Verifiable Presentation; see examples 24 and 25 in W3C Verifiable Credentials Data Model [i.264] for examples of these data structures.

An example of how to combine two W3C Verifiable Credentials into a W3C Verifiable Presentation with selected attributes is shown in Figure 9.

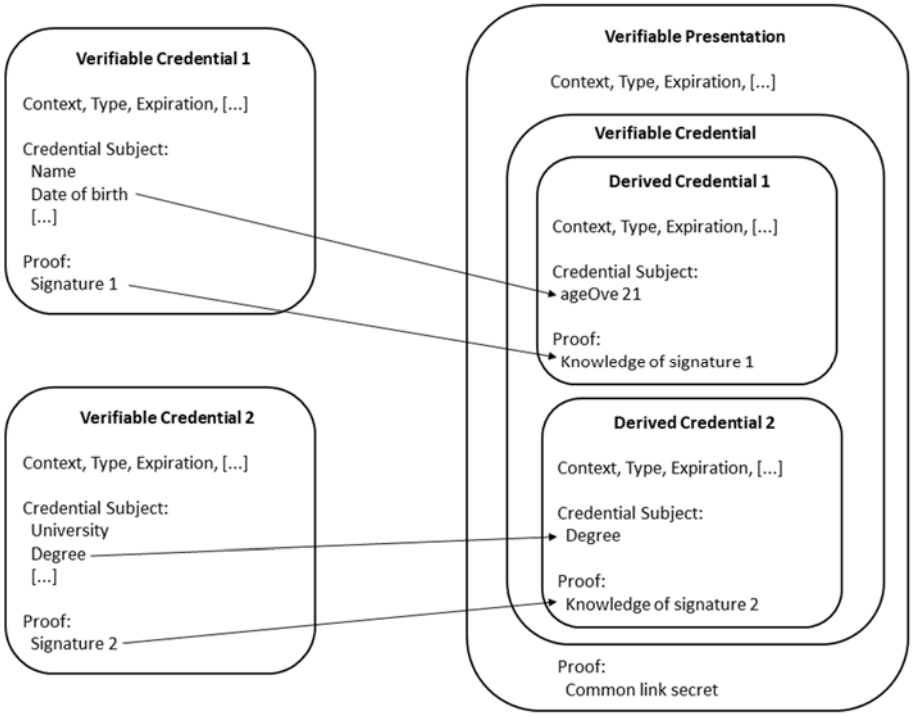


Figure 9: W3C Verifiable Credentials presented using ZKP

In Figure 9, selectively disclosed attributes from W3C Verifiable Credential 1 and W3C Verifiable Credential 2 are combined into a W3C Verifiable Presentation. CL-signatures are used in the Verifiable Presentation to create the proofs of knowledge of the original W3C Verifiable Credential signatures.

5.4.2 W3C VC Data Integrity with BBS Cryptosuite

5.4.2.1 W3C BBS Cryptosuite v2023

W3C BBS Cryptosuite v2023 [i.267] is an experimental draft specification, which defines a set of cryptographic suites for the purpose of creating, verifying and deriving proofs for the IRTF CFRG BBS [i.177] draft signature scheme that specifies BBS+ (see clause 4.4.2.4). The BBS+ signatures are compatible with any pairing friendly elliptic curve, however the cryptographic suites defined in the W3C BBS Cryptosuite specification allow the usage of the BLS12-381 curve for interoperability purposes.

NOTE: The W3C draft specification has the title "W3C BBS Cryptosuite v2023", although it describes the BBS+ scheme. The term BBS+ is however used throughout the present document to describe the multi-message signature scheme, whilst the term BBS04 describes the original single-message signature scheme.

W3C BBS Cryptosuite v2023 [i.267] can be used in conformance with the W3C Verifiable Credentials Data Integrity v1.0 specification [i.263], which in turn describes mechanisms for ensuring the authenticity and integrity of JSON-LD encoded credentials according to W3C Verifiable Credentials Data Model v2.0, especially through the use of digital signatures and related cryptographic proofs.

As a result, the IRTF CFRG BBS signature scheme (clause 4.4.2.4) can be applied on W3C Verifiable Credentials v2.0 and W3C Verifiable Presentations in order to disclose selected attributes, which are signed by the user's proofs without revealing the entire W3C Verifiable Credentials and their original signatures.

5.4.2.2 W3C VC Data Integrity with ISO standardized BBS04/BBS+

In the present clause it is analysed whether the ISO/IEC standardization efforts of BBS04/BBS+ (see ISO/IEC 20008-2 [i.184], ISO/IEC 24843 [i.185] and ISO/IEC CD 27565 [i.191], clause 4.4.6) are compatible with W3C BBS Cryptosuite v2023 and W3C Verifiable Credentials Data Integrity v1.1.

At the time of writing (August 2025), ISO/IEC 20008-2 [i.184] mechanism 3 is thus far the only ISO standard that specifies the qSDH cryptographic primitives for BBS04. However, ISO 20008-2 [i.184] mechanism 3 is designed for single messages and is therefore neither compatible with W3C BBS Cryptosuite v2023 nor W3C Verifiable Credentials Data Integrity v1.1. It has been proven [i.15] that BBS+ with multi-messages has the same security features as BBS04 with single messages, although BBS+ is not yet standardized by ISO.

If the ISO/IEC 24843 [i.185] is approved to standardize privacy-preserving attribute-based credentials schemes, the potentially new ISO standard may include a standardized version of BBS+ that has the potential to be compatible with W3C BBS Cryptosuite v2023 and W3C Verifiable Credentials Data Integrity v1.1.

Furthermore, ISO/IEC CD 27565 [i.191] refers to IRTF CFRG BBS (clause 4.4.2.4), whilst W3C BBS Cryptosuite v2023 also refers to IRTF CFRG BBS, so both ISO/IEC CD 27565 [i.191] and W3C BBS Cryptosuite v2023 share IRTF CFRG BBS as a common reference for the BBS+ scheme.

Hence, if ISO/IEC 24843 [i.185] and/or ISO/IEC CD 27565 [i.191] will standardize BBS+ according to IRTF CFRG BBS in conjunction with DIF draft "Blind Signatures extension of the BBS Signature Scheme" [i.80], then W3C BBS Cryptosuite v2023 can be enhanced to reference such an ISO standard. In such a scenario, the W3C Verifiable Credential Data Integrity 1.0 specification will refer to an ISO compliant version of W3C BBS Cryptosuite v2023. Finally, the W3C Verifiable Credentials Data Model v2.0 can be deployed with W3C Verifiable Credential Data Integrity 1.0, which is underpinned with an ISO standardized version BBS+.

NOTE 1: W3C Verifiable Credentials Data Model v2.0 with JSON-LD encoding has the potential to be underpinned by an ISO standardized version BBS+.

NOTE 2: W3C Verifiable Credentials Data Model v1.1 with JWT encoding does not refer to W3C Verifiable Credential Data Integrity 1.0, and can therefore not be supported by an ISO standardized version of BBS+.

5.4.3 W3C Data Integrity ECDSA Cryptosuites v1.0

The W3C "Data Integrity ECDSA Cryptosuites v1.0" [i.256] specification describes a data integrity cryptosuite for use when generating a digital signature using the Elliptic Curve Digital Signature Algorithm (ECDSA). The data integrity cryptosuites are in conformance with the W3C Verifiable Credentials Data Integrity [i.263] specification.

More specifically, selective disclosure is described in generalized terms according to the ECDSA-SD-2023 functions. The function createDisclosureData is used to generate a derived proof. The inputs include a JSON-LD document, an ECDSA-SD base proof, an array of JSON pointers to use to selectively disclose statements, and any custom JSON-LD API options (such as a document loader). The disclosure data object is produced as output, which contains the selectively disclosed fields of the JSON-LD document along with the ECDSA-SD proof.

5.4.4 Hyperledger AnonCreds (format)

The Hyperledger AnonCreds [i.131] credentials are JSON-formatted according to public AnonCreds objects, which in turn are defined by Schemas, CredDefs, Revocation Registry Definitions and Rev_Reg_Entrys. These objects are published by the issuers to repositories called Verifiable Data Registries (VDRs), which are accessible to users and verifiers to enable presentation generation and verification. AnonCreds can also be issued in accordance with the W3C Verifiable Credentials Data Model.

AnonCreds are bound to the user with a non-correlatable secret only known to the user itself called a link secret. The link secret as a blind attribute that is sent to the issuer during credential issuance. The issuer signs every claim (including the blinded link secret) individually, enabling selective disclosure. The Pedersen Commitment is used for the link secret. It means the issuer does not know the exact value of the link secret, and the holder can prove the ownership of credentials to a verifier without disclosing a persistent identifier. A user can link two attestations by generating a proof that the two exponents in the Pedersen Commitments are equal, i.e. they contain the same link secret.

The cryptographic signature scheme used by AnonCreds is CLRSA-signatures (see clause 4.4.1), which caters for selective disclosure and full unlinkability.

More information about the AnonCreds protocols is available in clause 6.4.1.

5.4.5 Cryptographic analysis

The maturity of W3C Verifiable Credentials can be considered as high, given the wide deployment of issued W3C Verifiable Credentials. However, BBS+, CL signatures and ECDSA are not secure against quantum-safe cryptographic algorithms [i.244] (see also clause 9), and they are additionally not standardized by NIST in the US or by SOG-IS in the EU. Furthermore, since AnonCreds are based on CLRSA-signatures, the cryptographic algorithms are not considered as quantum-safe nor SOG-IS approved.

5.5 JSON container formats

5.5.1 IETF JSON WebProof (JWP)

The JOSE [i.152] standard is a widely adopted container format for JSON-formatted Keys (JWK), Signatures (JWS), and Encryption (JWE). For example, JWTs with JOSE-containers are used by the OpenID Connect standard and by W3C's Verifiable Credentials.

However, JOSE is not designed to cater for the growing number of selective disclosure and ZKP schemes. Most of these emerging cryptographic schemes require additional transforms, are designed to operate on subsets of messages, and have more input parameters than traditional signature algorithms.

Examples of selective disclosure signature schemes that would benefit from a more flexible JSON container format are:

- BBS+ [i.177];
- CL Signatures [i.42];
- Idemix [i.136];
- Merkle Disclosure Proof 2021 [i.259];
- Mercurial Signatures [i.45];
- PS Signatures [i.223];
- U-Prove [i.3]; and
- Spartan [i.234].

They adhere to the same principles of collecting multiple attributes and binding them together into a single issued token, which is transformed into a presentation that reveals only a subset of the original attributes, predicate proofs, or proofs of knowledge of the attribute.

In order to address these issues, the IETF JSON working group has drafted the JSON WebProof (JWP) specification [i.152]. The JWP specification defines a new JSON container format similar in design to JSON Web Signature (JWS). However, JWS only integrity-protects a single payload, whilst JWP can integrity-protect multiple payloads in one message. JWP also specifies a new presentation form that supports selective disclosure of individual payloads, enables additional proof computation, and adds a protected header to prevent replay and support binding mechanisms.

The JWP payload can contain JSON Proof Tokens (JPTs) [i.151]. JSON Proof Token (JPT) is a compact, privacy-preserving representation of attributes. The attributes in a JPT are encoded as base64url-encoded JSON objects, allowing them to be digitally signed and selectively disclosed in the JWP payload. JPTs also support reusability and unlinkability when being used for Zero-Knowledge Proofs (ZKPs). A CBOR-based representation of JPTs is also defined in the JPT draft, called a CBOR Proof Token (CPT). It has the same properties of JPTs, but uses the JSON Web Proof (JWP) CBOR Serialization, rather than the JSON-based JWP Compact Serialization.

Furthermore, the JSON Proof Algorithms (JPA) specification [i.150] defines IANA registries for the cryptographic algorithms and identifiers to be used with the JSON Web Proof, JSON Web Key (JWK), and COSE specifications.

5.5.2 W3C JSON Web Proofs For Binary Merkle Trees

In hash-based cryptography, the Merkle signature scheme is a digital signature scheme based on Merkle trees and one-time signatures such as the Lamport signature scheme. It was developed by Ralph Merkle in the late 1970s and is an alternative to traditional digital signatures such as DSA or RSA. An advantage of the Merkle signature scheme is that it is plausible quantum-safe. Note that SPHINCS+ [i.238] can be considered an evolution of Lamport signature schemes that is more efficient and also not one-time but that can be used multiple (yet a limited number of) times.

The JSON Web Proofs For Binary Merkle Trees specification [i.258] defines a generic encoding of merkle audit paths that is suitable for combining with JWS to construct selective disclosure proofs. The specification is suitable for more generic applications and formats such as W3C Verifiable Credentials [i.264] and W3C Decentralized Identifiers [i.257].

JSON Web Proofs (see clause 5.5.1) are used as formats for the encoding binary merkle trees.

Selective disclosure is defined as the same as full disclosure with the exception that the rootNonce is not encoded in the compressed representation. The rootNonce is omitted in order to ensure that a selective disclosure proof does not reveal information that can be used to brute force siblings of disclosed members.

Merkle proofs are already being used to provide certificate transparency in IETF RFC 9162 [i.171]. The JSON Web Proofs For Binary Merkle Trees specification [i.258] is however independent of the certificate transparency specification.

5.5.3 JSON Web Zero Knowledge (JWZ)

JSON Web Zero-knowledge (JWZ) [i.141] is an open standard for representing messages proven by zero-knowledge technology.

A JWZ message consists of three parts:

- Header - defines the features of the JWZ token.
- Payload message - contains the message that will be shared with the relying party (verifier).
- Signature - represents a zero-knowledge authentication proof.

The parts are Base64-encoded and separated by dots in the JWZ message.

The JWZ header consists of the following parameters:

- alg - this is a zero-knowledge algorithm that is used for proof generation.
- circuitId - this is a circuit that is used for proof generation.
- crit - describes the list of header keys that the verifier has to support.
- typ - is the MIME type of the message. In the JWZ case, it is the protocol type of a packed message application/iden3-zkp-json.

The JWZ payload can be any arbitrary message, for example a DIDcomm message in the Iden3 protocol.

The JWZ signature is a zero-knowledge proof, which is based on a specific auth circuit. An auth circuit is a programmable zero-knowledge circuit that generates a ZK proof based on a set of inputs of the message. The JWZ zero-knowledge proof could for example use "groth16" as alg and "authV2" as circuitId.

The JWZ messages are used with the Iden3 protocol, which is described in clause 6.9.

More information about JWZ and complete examples are available in [i.141].

6 Selective disclosure systems and protocols

6.1 General

The present clause provides an analysis of a set of systems and protocols for selective disclosure.

The topics for the analysis of each selective disclosure protocol are:

- Signature scheme(s) used for selective disclosure and optionally Zero-Knowledge Proofs, when applicable with references to clause 4.
- (Q)EAA format(s) for selective disclosure, when applicable with references to clause 5.
- Protocol(s) for presentation of the user's (Q)EAAs to a relying party (relying party).
- Maturity of the protocol's specification and deployment.
- Cryptographic aspects, more specifically if the cryptographic algorithms used for the selective disclosure protocol are approved by SOG-IS and allows for QSC algorithms for future use.

The protocols are first categorized according to the four main cryptographic schemes for selective disclosure:

- Atomic (Q)EAA protocols, see clause 6.2. These protocols correspond to the (Q)EAA signature schemes described in clause 4.2 and formats in clause 5.2.
- Multi-message signature protocols, see clause 6.4. These protocols correspond to the multi-message signature schemes described in clause 4.4 and formats in clause 5.4.
- Salted attribute hashes protocols, see clause 6.3. These protocols correspond to the multi-message signature schemes described in clause 4.3 and formats in clause 5.4.
- Proofs for arithmetic circuits protocols, see clause 6.5. These protocols correspond to the proofs for arithmetic circuits described in clause 4.5.

In addition to the traditional categories listed above, the following systems are described, which are based on a mix of selective disclosure schemes:

- Anonymous attribute based credentials systems, see clause 6.6.
- ISO mobile driving license (ISO mDL), see clause 6.7.

6.2 Atomic attribute (Q)EAA presentation protocols

6.2.1 PKIX X.509 attribute certificates with single attributes

An access control system based on PKIX X.509 certificates with atomic attributes is illustrated in Figure 10.

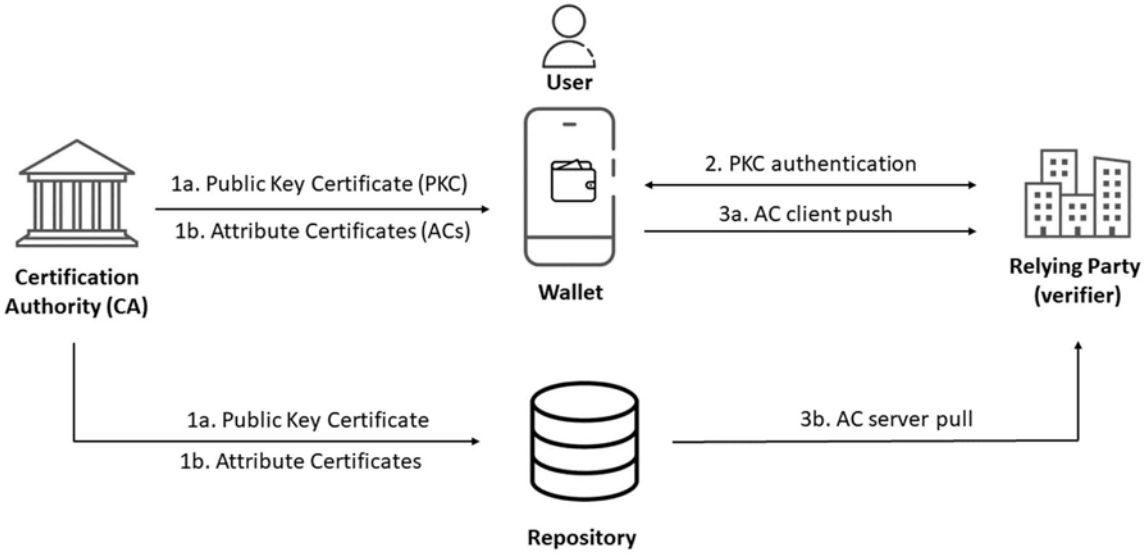


Figure 10: Overview of attribute certificate authorization

First, the system is configured by a Certification Authority (CA) that issues a PKIX X.509 public key certificate to a user's wallet. The user has a corresponding private key protected in the wallet, such that the user can be authenticated with the public key certificate. The public key certificate may only contain a pseudonym. The Certification Authority also issues short-lived PKIX X.509 attribute certificates with atomic attributes. The attribute certificates are associated with the public key certificate, and they may be stored in the user's wallet and/or in a central repository.

Second, the user authenticates to a relying party (with an access control system) by using the public key certificate. For example, TLS/SSL could be used for this authentication. If the public key certificate only contains a pseudonym of the user, the authentication protocol does not reveal the user's identity.

Third, the user's attribute certificate(s) are submitted to the relying party's access control system. The attribute certificate(s) may either be pushed from the client to the relying party, or pulled from the repository by the relying party.

For more information about attribute certificate architectures, see the IETF RFC 5755 [i.158].

An alternative design of using attribute certificates for anonymous authorization is described in the paper "A First Approach to Provide Anonymity in Attribute Certificates" [i.23] from 2004.

The PKIX X.509 certificates can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use, meaning that the attribute certificate access control solution meets the SOG-IS requirements on cryptographic algorithms.

6.2.2 VC-FIDO for atomic (Q)EAAs

Another example of a protocol for selective disclosure based on atomic (Q)EAAs is the VC-FIDO [i.56] integration that was invented at Kent University. The used atomic (Q)EAA format is W3C Verifiable Credential, which is described in clause 5.2.3.

In order to issue the atomic W3C Verifiable Credentials to an EUDI Wallet, the user needs to be identified or authenticated to a QTSP. The VC-FIDO integration is based on the W3C WebAuthn protocol in the FIDO2 standard. The WebAuthn [i.266] stack is extended with a W3C Verifiable Credentials enrolment protocol, resulting in a client that can enrol for multiple atomic short-lived W3C Verifiable Credentials based on W3C Credential templates. These atomic short-lived W3C Verifiable Credentials can then be (temporarily) stored in an EUDI Wallet, and be combined into a Verifiable Presentation that is presented to the relying party (verifier). Selective disclosure is achieved since the user can enrol for the atomic attributes it needs for a specific use case, and present only those atomic (Q)EAAs to a Relying Party.

The VC-FIDO integration was presented by David Chadwick at SHACK2020 [i.56]. This presentation explains the VC-FIDO architecture diagrams and shows a demo of how the client enrols for three atomic W3C Verifiable Credentials (address, driving license, and credit card) that are combined into a Verifiable Presentation as a parking ticket. The VC-FIDO integration is still a prototype, which is deployed as a pilot at National Health Services (NHS) in the UK.

The W3C Verifiable Credentials can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use, meaning that the VC-FIDO solution meets the SOG-IS requirements on cryptographic algorithms.

6.3 Salted attribute hashes protocols

6.3.1 OpenAttestation (Singapore's Smart Nation)

OpenAttestation, which is part of Singapore's Smart Nation initiative and developed within the GovTech's Government Digital Services, is an open source framework for verifiable documents and transferable records.

OpenAttestation allows a user to prove the existence and authenticity of a digital document. It makes use of smart contracts on the Ethereum blockchain to store cryptographic proof of individual documents. As an alternative to using the Ethereum blockchain, OpenAttestation can also be used to create verifiable documents using digital signatures.

More specifically, OpenAttestation provides Document Integrity [i.204] based on a target hash of salted attribute hashes. An overview of the OpenAttestation Document Integrity flow is illustrated in Figure 11.

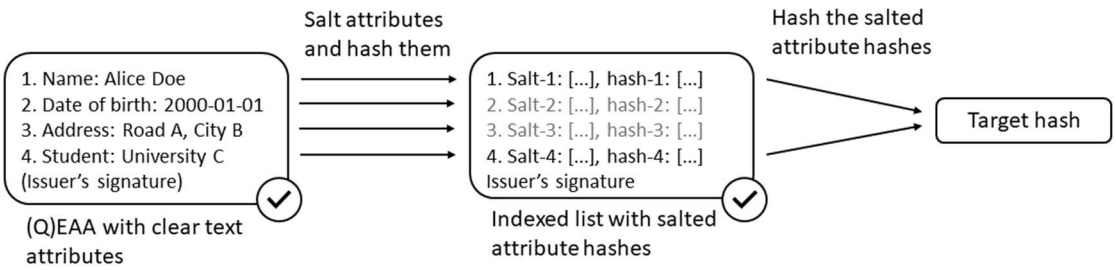


Figure 11: Overview of the OpenAttestation scheme

The target hash of the digital document is calculated as follows: Sort the selected salted attribute hashes from the previous step alphabetically and hash them all together. To compute the target hash the KECCAK256 algorithm is used.

During verification of the digital document, the same exact steps are performed again, and the result is compared to the target hash. If the two hash values match, the digital document integrity is intact.

Since the OpenAttestation scheme is based on salted attribute hashes, which can be signed with QSC algorithms, it can be considered as plausible quantum safe.

6.4 Multi-message signature protocols and solutions

6.4.1 Hyperledger AnonCreds (protocols)

The Hyperledger AnonCreds (Anonymous Credentials) specification [i.131] is based on the open source verifiable credential implementation of Hyperledger AnonCreds that has been in use since 2017. The Hyperledger AnonCreds software stack was initially implemented as a combination of the Hyperledger Aries [i.132] protocols, the Hyperledger Indy [i.134] credentials, and the Hyperledger Ursa [i.135] SDK with features for public/private key pair management, signatures and encryption. Since 2022 all Hyperledger AnonCreds features have been merged in the Hyperledger AnonCreds project. The Hyperledger AnonCreds credential format is described in clause 5.4.4.

Hyperledger AnonCreds are widely deployed, and are for example used by organizations such as the Government of British Columbia, IDunion, and the IATA Travel Pass.

6.4.2 Direct Anonymous Attestation (DAA) used with TPMs

Direct Anonymous Attestation (DAA) is a cryptographic protocol which enables remote authentication of a trusted computer yet preserving the privacy of the user.

ISO/IEC has standardized the DAA protocol in ISO/IEC 20008 [i.184]. The DAA protocol has been adopted by the Trusted Computing Group (TCG) in the Trusted Platform Module (TPM) v2.0 specification [i.242] to ensure the integrity of the computer yet addressing privacy concerns. Furthermore, Intel® has also adopted DAA in the Enhanced Privacy ID (EPID) 2.0 specification.

Since the ISO/IEC 20008 [i.184] standard specifies the cryptographic primitives of BBS and PS-MS, DAA is essentially based on BBS and PS-MS credentials with device binding and pseudonyms.

The primary scope of a TPM is to ensure the integrity of a computer and its operating system. The purpose is to ensure that the boot process starts from a trusted combination of hardware and software, and continues until the operating system has fully booted and applications are running in a trusted state. A computer that is running in a trusted state can be better controlled with respect to software licences and protection against computer viruses and malware.

The DAA eco-system consists of three entities: the DAA Member (i.e. TPM platform or EPID-enabled microprocessor), the DAA Issuer, and the DAA Verifier. The Issuer verifies the TPM platform during the Join step and issues a credential to the platform. The Member presents the credential to the Verifier during the Sign step; the Verifier can, based on a zero-knowledge proof, verify the credential without violating the platform's privacy. The DAA protocol also supports a blocklist such that Verifiers can prevent attestation attempts from TPMs that have been compromised.

Furthermore, the DAA protocol splits the signer role in two parts. In brief, a principal signer (a TPM) signs messages in collaboration with an assistant signer (the standard computer into which the TPM is embedded). This split aims to combine the high level of security provided by the TPM, and extend it by using the high level of computational and storage ability offered by the computing platform. Chen et al. have specified the DAA protocol based on an ECC scheme [i.63] using Barreto-Naehrig curves, which is implemented by both TPM 2.0 and EPID 2.0.

The DAA protocol standardized in ISO/IEC 20008 [i.184], and implemented according to the TPM 2.0 and EPID 2.0 specifications, is considered mature and has been deployed at computers at a very large scale. Since the DAA protocol is based on an ECC scheme, it is however not considered as plausible quantum safe.

6.5 Proofs for arithmetic circuits solutions

6.5.1 Anonymous (Q)EAAs from programmable ZKPs and existing digital identities

6.5.1.1 Overview

This category is based on the principle of deriving anonymous (Q)EAAs by combining existing digital identities (such as X.509 certificates) with zero-knowledge proofs generated by general-purpose ZKP schemes (such as zk-SNARKs).

A generalized model of such systems is described in the paper "Bringing data minimization to digital wallets at scale with general-purpose zero-knowledge proofs" [i.14] by Babel and Sedlmeir. The solution, which can be divided in three phases, is illustrated in Figure 12.

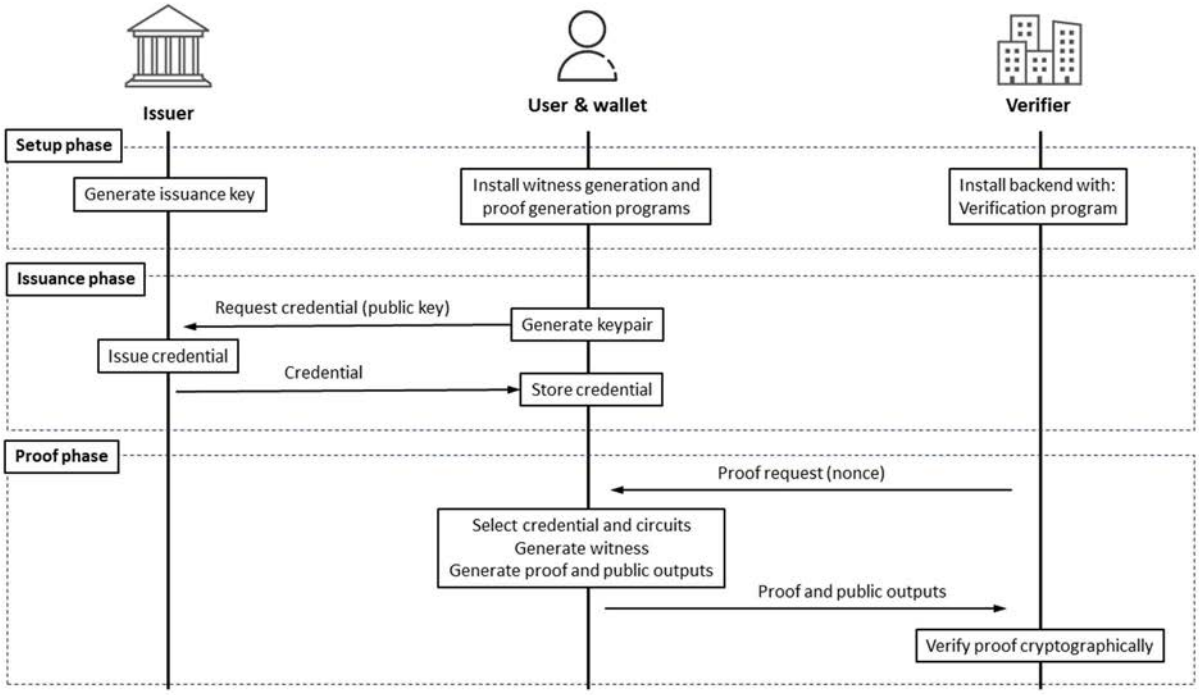


Figure 12: Overview of proofs used with credentials

6.5.1.2 Setup phase

In the setup phase, the issuer generates the issuance key. This could for example be a PKIX CA that issues X.509 certificates, or a PKD compliant CA that issues ICAO eMRTDs. The credential format, revocation scheme, etc., are typically also specified and implemented in this phase.

The digital wallet is provided with a witness generation program and a proof generation program, which implements the proofs for arithmetic circuits. Typically, the zk-SNARK circuits are integrated with the digital wallet by using a circuit compiler.

The verifier's backend is provided with the server-side circuits of the zk-SNARK scheme, which allows the verifier to validate the ZKPs generated by the digital wallet. The verifier in this scenario is equivalent to a relying party in the eIDAS2 context.

6.5.1.3 Issuance phase

During the issuance phase the digital wallet generates a key-pair and submits the public key in a credential request to the issuer. The issuer creates and signs the credential, for example an X.509 certificate, and returns it to the digital wallet where it is installed. The issuance phase can for example be performed as described in the ETSI EN 319 411-1 [i.90] standard for trust service providers issuing certificates.

6.5.1.4 Proof phase

The proof phase is initiated by the verifier, who submits a proof request (including a nonce) to the digital wallet. The user selects the credentials to be used for verification, and the digital wallet runs the verification algorithm using the locally stored credentials. The verification algorithm depends on the credentials framework, which could for example be a PKIX CA, ICAO PKD, or SSI type issuer of W3C VCs. The digital wallet also creates a ZKP that this verification algorithm was run correctly, without providing any further information than the statement provided by the verifier.

EXAMPLE: If a PKIX CA is used for issuance of X.509 certificates, the validation process should check that the user possesses the private key associated with the X.509 certificate, and that the X.509 certificate is valid (properly signed). The X.509 certificate status can be checked with respect to CA signature, expiry date, and revocation checks using OCSP.

The digital wallet executes the programmable ZKP scheme with the selected credential and its validity as private inputs. The digital wallet generates the witness, proof and public outputs and sends the ZKP result to the verifier. Hence, the digital wallet can use the ZKP scheme to submit the credential's verification result and selected attributes or predicates that need to be disclosed to the verifier. In order for the verifier to trust the verification result, the digital wallet also creates a ZKP that certifies the correct execution of the verification program, yet without sharing any details about the inputs or the results of the credential verification algorithm. Hence, the ZKP scheme can prove that the verification algorithm that was locally executed by the digital wallet resulted in the shared statement. The verifier can use the ZKP to check that the digital wallet has a credential that was indeed issued by a particular CA, and that the user possesses the private key associated with the public holder binding key referenced in the credential.

6.5.2 Cinderella: zk-SNARKs to verify the validity of X.509 certificates

The Cinderella project is described in the paper "Cinderella: Turning Shabby X.509 Certificates into Elegant Anonymous Credentials with the Magic of Verifiable Computation" [i.77] by Delignat-Lavaud et al. As indicated by the title, the project is an implementation of how to validate X.509 certificates locally at the digital wallet, and share the results with a verifier by using a ZKP scheme.

More specifically, the Cinderella project implemented a new format for application policies by composing X.509 templates, and provided a template compiler that translates C code for validating X.509 certificates within a given policy into an arithmetic circuit that allows for the generation of proving and verification programs. In order to produce a zero-knowledge verifiable computation scheme based on the Pinocchio [i.220] zk-SNARK, the Geppetto [i.72] cryptographic compiler was used.

The Cinderella project was evaluated by two real-world applications: a plug-in replacement for certificates within TLS [i.159], and access control for the Helios [i.1] voting protocol. Fine-grained validation policies were implemented for TLS with revocation checking and selective disclosure of certificate contents, which turn X.509 certificates into anonymous credentials. For Helios, additional privacy and verifiability guarantees for voters equipped with X.509 certificates were obtained, such as those currently available from certain national ID cards.

Rather than modifying the TLS standard and implementations, the X.509 certificate chains communicated during the TLS handshake were replaced with a single X.509 pseudo-certificate that carries a short-lived ECDSA public key and a proof that this key is properly signed with a valid RSA certificate whose subject matches the peer's identity. Also OCSP stapling can be communicated via the Cinderella version of TLS. National eID smartcards with X.509 certificates issued in Belgium, Estonia, and Spain have been evaluated with the Cinderella version of TLS.

One immediate issue is proving performance. Since the resulting Cinderella pseudo-certificates can take up to 9 minutes to generate for complex policies on a computer, it is recommended that they are generated offline and refreshed typically on a daily basis. Once the setup is configured or refreshed, online verification of the Cinderella pseudo-certificates and their embedded proof takes less than 10 ms. Yet, progress in zk-SNARK proving performance - e.g. lookup table with PLONKish arithmetization, assembly provers for mobile platforms, and tolerance of "bigger" proofs (hundreds of kilobytes) would arguably make a re-implementation of Cinderella practical on mobile phones, with proving times in the low double-digit seconds range.

NOTE: A vulnerability [i.130] in the Geppetto compiler that was found later would also require another toolchain to compile C-code to a ZKP (e.g. zk-SNARK) proving and verification algorithm.

6.5.3 zk-creds: zk-SNARKs used with ICAO passports

The zk-creds protocol was introduced in the paper "zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure" [i.231] by Rosenberg et al. The zk-creds protocol uses programmable ZKPs in the form of zk-SNARKs to:

- Remove the need for credential issuers to hold persistent signing keys. Instead, credentials can be issued to a bulletin board instantiated as a transparency log, a Byzantine system, or a blockchain.
- Convert existing identity documents into anonymous credentials without modifying documents or coordinating with their issuing authority.
- Allow for flexible, composable, and complex identity statements over multiple credentials.

The second use case has been implemented by generating ZKPs of ICAO compliant eMRTDs (passports) to create anonymous credentials for accessing age-restricted videos. More specifically, the eMRTDs were NFC-enabled and issued by the US State Department, which signs a hash tree of the eMRTD data with a raw RSA signature. The ZKP is essentially generated based on the eMRTD's Data Group 1 (DG1), which contains the textual information available on the eMRTD's data page and the Machine Readable Zone: name, issuing state, date of birth, and passport expiry. It is worth mentioning that the ZKP-based verification of the RSA signature is not a standard part of the zk-creds construction owing to the process of reissuance to a bulletin board with more ZKP-friendly primitives.

6.5.4 Anonymous credentials from ECDSA

6.5.4.1 Overview of the research paper

Similar to the Cinderella research paper by Delignat-Lavaud [i.77], Matteo Frigo and Abhi Shelat have designed circuit-based (general-purpose) zero-knowledge proofs to construct proofs for the correct verification of digital certificates compatible with legacy formats and hardware-based key storage. Frigo's and Shelat's results are published in the research paper "Anonymous credentials from ECDSA" [i.113]. While Delignat-Lavaud [i.77] focused on X.509 certificates and RSA-based holder binding, Frigo and Shelat consider the mdoc standard with ECDSA-based holder binding. As for the Cinderella paper, the main challenge Frigo and Shelat need to resolve is that the circuits for verifying SHA256 hashing, ECDSA signatures, and parsing (ASN.1 versus CBOR) have a substantial number of constraints and, therefore, require high compute and memory resources on the prover side. For instance, implementing the verification circuit in Circom (see also Crescent in clause 6.5.5 and FIDO-AC in clause 6.6.5) would correspond to several million constraints and a proving time on the order of a few minutes when implemented on mobile phones via the rapid snark assembly prover.

To overcome this issue, Shelat and Frigo introduce a variety of optimizations, including a novel ZKP system that combines the sum check protocol and Ligero proof system (which is known to be very memory-efficient) [i.7] and the avoidance of simulating foreign arithmetic (e.g. the P256 field) in the circuits. Thereby, Anonymous credentials from ECDSA achieve a more than 100x speedup for proving an ECDSA signature, which gets it to within one order of magnitude overhead compared to generating BBS proofs. Moreover, it achieves a 20x speedup of SHA256 hashing compared to Ligero, and similar performance as Binius [i.79], a recent ZKP system that is heavily optimized for binary operations.

Through these optimizations, the main bottleneck of verifying MDOC presentations is the hashing of the MDOC document. The proof of possession of a valid MDOC presentation, including a few simple predicates (expiration check, age proof) takes around 1,2 seconds on a high-end mobile phone (such as a Google Pixel 6). While the implementation does not implement revocation checks, the presentation of the paper [i.113] at the Real World Crypto (RWC) conference included an outlook how this could be implemented efficiently via signed pairs of adjacent revocation IDs for revoked certificates, leveraging the efficiency of verifying ECDSA signatures compared to hashing. However, through the performance improvements achieved for SHA256 hashing, also other approaches, such as constructing an accumulator in the form of a SHA256-based Merkle tree over a bitstring-based revocation registry (Babel and Sedlmeir [i.14]), would become practical, as the amount of data to be hashed in a corresponding proof would be comparable to the MDOC.

The construction by Frigo and Shelat has met great interest by the scientific community and led to standardization efforts as well as attempts to further improve on proving time by reducing the hashing operations in MDOCs (particularly considering that with ZKPs, selective disclosure via salted hashes becomes obsolete). However, while their optimizations make use of several well-established mathematical constructions in ZKP research, it is nevertheless a highly complex construction leveraging multiple complex optimizations that may take standardization and certification bodies a long time to familiarize with. On the other hand, it is probably fair to say that as of now, their work by far remains the most performant in terms of proof generation for legacy-compatible credentials. Nevertheless, it should be mentioned that compared to other approaches based on circuit-based ZKPs, such as the Iden3 protocol, the ZKPs have substantially higher proof sizes and, therefore, impose a higher verification effort for servers. However, this well reflects the typical case for available resources on the holder and relying party side, Moreover, other benefits, such as transparent setup and plausible post-quantum security which are not met by constructions based on, e.g. the Groth16 proof system, arguably by far outweigh this aspect in the bilateral electronic identification use case.

6.5.4.2 Implementation and standardization

The results in the publication "Anonymous credentials from ECDSA" [i.113] can be applied on the ISO/IEC 18013-5 [i.181] mobile driving license (ISO mDL). The ISO mDL presentation protocol is modified so that the user instead produces a zero-knowledge proof, which proves that their mdoc verifies with respect to the requested attributes. The following public attributes are shared from the EUDI Wallet's ISO mDL application with the relying party: the public key of the identity-issuer, an attribute value to disclose (such as the "age over 18" boolean), the name of this attribute (as specified in the MSO), the liveness transcript, and the tnow parameter to verify that the mdoc has not expired. The rest of the mdoc values in the statement are hidden, in the sense that the relying party is convinced that such values exist, but does not learn them through the interaction.

This zero-knowledge argument (ZKARG) implementation for the ISO mDL application is subject for standardization within the ISO/IEC JTC 1/SC 17 WG10 ZKP.

In addition to the standardization initiative in ISO/IEC JTC 1/SC 17 WG10, IETF CFRG has published the draft specification "libZK: a zero-knowledge proof library" [i.147]. The IETF CFRG draft specification specifies the ZK proof system that is described by Frigo and Shelat in "Anonymous credentials from ECDSA" [i.113]. This ZK proof system consists of two major components: the outer proof is a Ligero ZKP that checks a property on a committed transcript, whilst the committed transcript corresponds to a proof for a bespoke verifiable-computation scheme. The document [i.147] defines an algorithm for generating a succinct non-interactive zero-knowledge argument that for a given input x and a circuit C, there exists a witness w, such that $C(x,w)=0$.

6.5.5 Crescent: Stronger Privacy for Existing Credentials

Similar to the work by Frigo and Shelat [i.113], Paquin et al. [i.219] argue that compatibility with legacy credentials is essential for fast adoption, that circuit-based ZKPs can achieve such compatibility, and that proving time remains the key challenge of circuit-based ZKPs, particularly on mobile phones. This work considers two legacy credential formats, JWTs and mdoc. The key trick applied by Paquin et al. [i.219] follows a similar idea as discussed already in the Cinderella paper [i.77], namely, to separate parts of the statements to be proved that are repetitive (e.g. verification of the signature on the certificate) and one that changes in every interaction (e.g. verification of the holder's signature on the relying party's unique challenge, as well as verification of non-expiration according to a fresh timestamp supplied by the relying party). The circuit size for the one-time creation of the pseudo-certificate is similar in size as the circuit in Cinderella [i.77] and as discussed by [i.14] - around 2 million R1CS constraints, dominated by hashing and ECDSA verification.

Taking aside the one-time work to create a proof of the repetitive statements (the result is often called a pseudo-certificate, which include a derived certificate as well as a ZKP of its correct derivation), the work for the user (prover) that needs to be performed for every verification process reduces to around 20 ms by making use of a sigma-protocol for possession of an ECDSA signature where the committed public key coincides with the one committed in the pseudo-certificate. In terms of combining sigma-protocols with zk-SNARKs, the approach shares some ideas with LegoSNARK [i.49], yet the use is different as LegoSNARK would use circuit-based ZKPs for more complex predicates that cannot be covered in BBS credentials, such as binding to an ECDSA key.

As such, there are fewer performance optimizations necessary, and the implementation relies on Circom [i.68], which has seen several years of adoption and auditing in blockchain projects (see also clause 6.9 about the Iden3 protocol). While the Groth16 proofs supported by Circom involve a trusted setup and are not post-quantum secure, they are shorter (around 1 kB) and faster to verify (around 15 ms) than the ones constructed by Frigo and Shelat [i.113]. For hardware binding, Crescent makes use of recent work by Woo et al. [i.253], which is itself an optimized circuit-based ZKP that accelerates proof generation for ECDSA-based signatures by almost an order of magnitude. If further acceleration is needed, Woo et al. also devise a pre-computation that leaves only a sigma protocol involving the verifier's challenge to be executed. As such, the construction can achieve similar performance to creating BBS proofs. While all components used to achieve this performance are quite well established, their composition nevertheless introduces a substantial amount of complexity that is difficult to compare with, e.g. the construction by Frigo and Shelat [i.113].

6.5.6 Analysis of systems based on programmable ZKPs

The protocols that combine general-purpose ZKP schemes and digital identities provide some valuable characteristics:

- The existing digital identity infrastructures can be re-used as is, more specifically the eIDAS2 framework of X.509 certificates. This covers secure hardware for issuers' signing keys, secure hardware in mobile phones as commonly used with FIDO2. In particular, the issuance process would not need to be changed at all if the hardware attestation chain for the holder binding keypair is checked by the issuer in this step (which should usually be the case).

- The existing validation algorithm and revocation checking schemes can be executed in the digital wallet. That is, all the checks that the verifier usually does (verification of the (Q)EAA's signature, holder binding, expiration date, OCSP status or signature on the CRL and inclusion/exclusion of the (Q)EAA, etc.) will now be executed in the wallet app, and only the result of the verification, the explicitly requested attributes, and a ZKP of the correctness of these results will be shared with the relying party.

NOTE 1: Some of the "comparison" values, such as the challenge for holder binding check or threshold values for range proofs (expiration, age), may need to be communicated from the verifier to the prover, and then be disclosed as public output of the ZKP.

- Only the relevant information about the credential's validity and selected attributes or predicates need to be shared with the verifier because the holder also shares a zk-SNARK of correct local verification with the verifier.

- Both the credentials and zk-SNARK protocol can be designed with cryptographic algorithms that are plausible quantum-safe.

- Features such as very general predicates (e.g. proof of location within a certain region based on coordinates included in the (Q)EAA or verifiable pseudonyms derived from the holder's public key) and designated verifier proofs that can improve both security and privacy guarantees are easy to implement. Furthermore, equality proofs across different (Q)EAA can help achieve consistency (i.e. that all (Q)EAA belong to the same subject or wallet, without the need to disclose a subject- or wallet-specific cryptographic identifier. Another interesting predicate is membership of the issuer among a list of trusted issuers, as this can further improve herd anonymity if multiple issuers use the same (Q)EAA format. Lastly,certificate chaining falls into this category: A holder can prove the validity of a certificate chain, only disclosing the issuer of the top ("root") certificate. In a large-scale system of (Q)EAA, where, e.g. every municipality can issue (Q)EAA yet the permissions to do so are managed on a federal / state level, this can substantially contribute to unlinkability.

- General predicates may be particularly useful to facilitate data-minimizing (Q)EAA issuance. For instance, a holder could ask an issuer to bind a new (Q)EAA to the same public key and name as another (Q)EAA previously presented to the issuer, without disclosing the raw name or public holder binding key. This addresses a key issue in the AnonCreds' Link Secret, which relies on the honesty of holders at the time of issuance.

- Designated verifier properties that are challenging to achieve concurrently with unlinkability and non-interactiveness can be easily implemented. Designated verifier proofs allow the holder to make sure that only the designated recipient is convinced of the correctness of the verifiable presentation, mitigating risks of monetization of sensitive, attested (Q)EAA and of man-in-the-middle attacks.

However, the anonymous credential schemes described in the present clause are still under research and development, and have not been deployed at scale. Hence, the maturity can be considered as low, although they provide a promising option for zero-knowledge proofs for the future of eIDAS2 and the EUDI Wallet. Moreover, yet, arithmetic circuits for commonly used cryptographic primitives, such as SHA256, RSA, and ECDSA are very complex and involve higher proving times than common digital signature schemes such as ECDSA. Proving time may be even worse for lattice-based post quantum secure digital signatures. The programmable ZKP systems that are most mature (zk-SNARKs) add some pronounced tradeoffs, e.g. the generality of preprocessing versus performance aspects.

NOTE 2: Among the key benefits of general-purpose ZKPs is their flexibility to adapt to legacy formats and cryptographic constructions, and to be able to flexibly accommodate future updates (e.g. switching to plausibly post-quantum secure signature schemes for holder binding and issuance). Nevertheless, as the proving resources are still considered a bottleneck, it may make sense to make minor modifications to existing certificate formats (e.g. to reduce the number of hashing operations). Another approach that has sometimes been brought up is using formats (e.g. salted Merkle trees) that are efficient both within a general-purpose ZKP but also can be used in a "hybrid" scheme that only facilitates selective disclosure in a transition period where low-end mobile devices do not have enough computational power to generate ZKPs with low latency and, thus, a high degree of usability.

NOTE 3: As a generalization of ZKPs, Multi-Party Computation (MPC) could also facilitate data-minimizing (Q)EAA presentations. Yet, ZKPs can be considered more mature, and as a specific form of MPC, it is also unlikely that a general-purpose MPC protocol applied to the specific case of a (Q)EAA presentation will be more performant than the best general-purpose ZKP constructions. On the other hand, (Q)EAA presentations based on fully homomorphic encryption are conceivable, which relies on quite different cryptographic constructions. Yet, thus far, his direction seems to have received little research.

NOTE 4: The high degree of complexity of general-purpose ZKP construction, paired with its still fast progress may pose a substantial barrier to its standardization. Even if regulators cannot agree on the certification of a ZKP scheme, the private sector may do so with the "layered" approach that can accommodate existing (Q)EAA formats. For an analysis of the corresponding accountabilities, see [i.228].

6.6 Anonymous attribute based credentials systems

6.6.1 Idemix (Identity Mixer)

The Idemix (Identity Mixer) technology [i.136] was invented by IBM® Research in 2008. The Idemix system caters for strong authentication that is privacy preserving based on Attribute Based Credentials (ABC).

In summary, the Idemix scheme contains two protocols: Issuing the credential to a user and presenting it when accessing a relying party. An overview of the Idemix ABC scheme is illustrated in Figure 13.

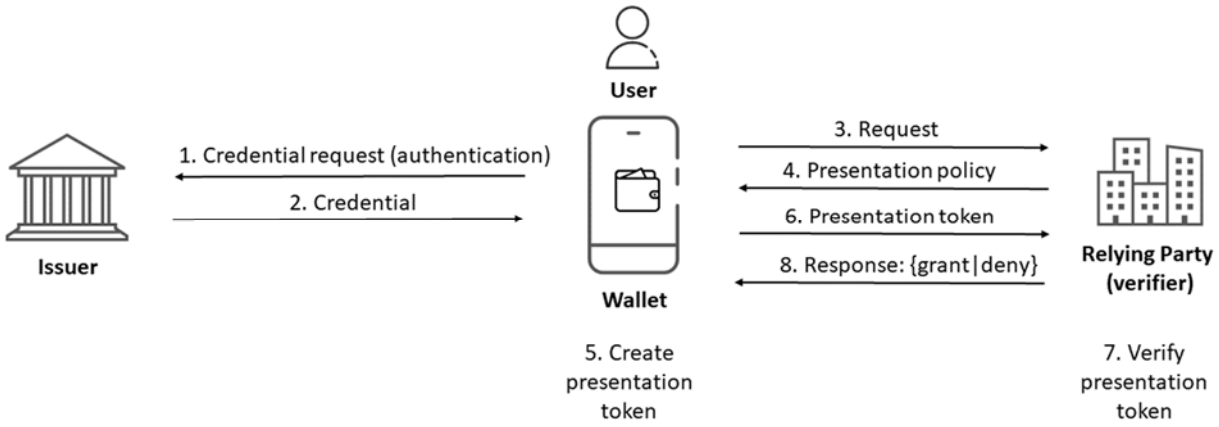


Figure 13: Overview of the Idemix ABC scheme

The Idemix system supports selective disclosure based on unlinkable Zero-Knowledge Proofs, such that users can prove that they are over 18 years old without revealing their name or birthdate. Idemix uses the pairing-based CL-signature scheme (clause 4.4.1) to prove knowledge of a signature in a Zero-Knowledge Proof.

NOTE 1: CL-signatures are not SOG-IS approved and not plausible quantum-safe.

The Idemix solution has been implemented by IBM® Identity Mixer [i.136], Hyperledger Fabric [i.133], Radboud University Nijmegen's IRMA project [i.227], and the EU-project PrimeLife [i.224]. The Idemix system was also selected as an ABC solution by the EC-funded project Attribute based Credentials for Trust (ABC4Trust) [i.137].

NOTE 2: Idemix is similar to the U-Prove (see clause 6.6.2) in the sense that both protocols are based on privacy-preserving ABC technology, although the iterations in the issuance phase and the underlying cryptographic algorithms differ.

NOTE 3: Idemix caters for multi-show unlinkability, whilst U-Prove does not [i.226].

The Idemix ABC system has been formalized by Camenisch et al. in the paper "A Formal Model of Identity Mixer" [i.46] and the Idemix revocation mechanisms are discussed by Lapon et al. in the paper "Analysis of Revocation Strategies for Anonymous Idemix Credentials" [i.196].

6.6.2 U-Prove

The U-Prove scheme is based on Attribute Based Credentials (ABC), which in turn relies upon Stefan Brand's cryptographic research on selective disclosure and blinded signature schemes in the book "Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy" from 2000 [i.33]. Brands founded a company to implement the U-Prove ABC scheme, and this company was later acquired by Microsoft®. In 2013, Microsoft® Research released the Identity Metasystem with support for U-Prove ABC to cater for anonymous credentials [i.201]. The U-Prove ABC system was also selected by the EC-funded project Attribute based Credentials for Trust (ABC4Trust) [i.137].

In summary, the U-Prove scheme contains two protocols: issuing the credential to a user and presenting it when accessing a relying party. The U-Prove scheme is illustrated in Figure 14.

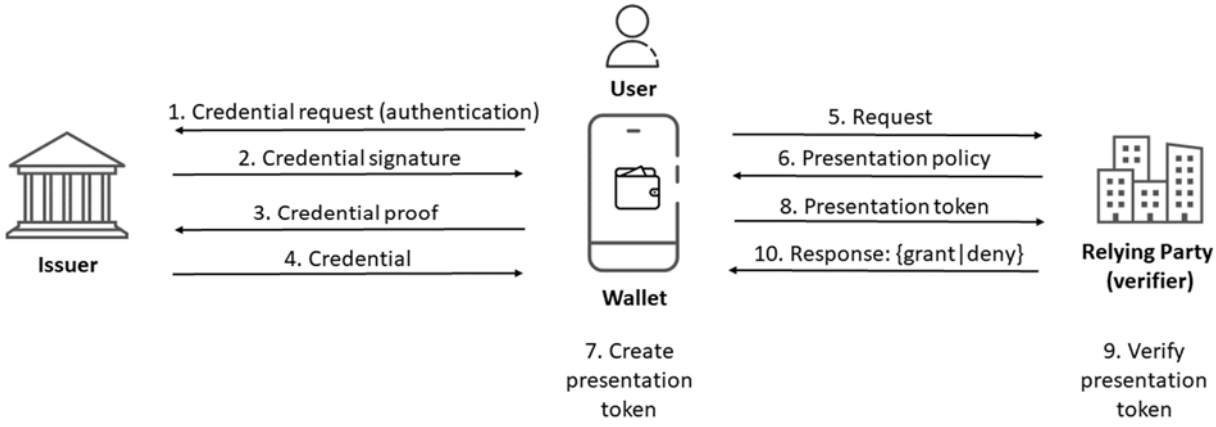


Figure 14: Overview of the U-Prove ABC scheme

The U-Prove issuing protocol is performed between the issuer and the user. The objective of this protocol is for the user to receive a credential, such that it can later present a selected set of attributes to access a relying party. The issuer basically applies a blind signature to the credential with attributes. In other words, the issuer verifies the validity of the attributes and applies a signature without seeing the resulting signature. Since the issuer does not store the result of the issuing protocol, the user cannot be tracked when using the credential, i.e. the processes of issuing and presenting are unlinkable.

The U-Prove presentation phase is based on a selective disclosure protocol between the user and the relying party. Based on the relying party's presentation policy, the user selects those attributes that it is willing to present from the issued credential. All the other attributes can be proved by the user to be unchanged in the credential. By the end of the interaction the relying party receives a presentation token with all the revealed attributes and the intact issuer's signature on the whole set of attribute values.

NOTE 1: U-Prove is similar to the Idemix (see clause 6.6.1) in the sense that both protocols are based on privacy-preserving ABC technology, although the iterations in the issuance phase and the underlying cryptographic algorithms differ.

The U-Prove scheme is based on the DLP and the credentials are issued as DLREP-based certificates as well as for RSAREP-certificates.

NOTE 2: Since U-Prove is based on algorithms using the DLP, the scheme cannot be considered as quantum-safe.

6.6.3 ISO/IEC 18370 (blind digital signatures)

The ISO/IEC 18370 series [i.183] standardize blind digital signature protocols. Whereas, ISO/IEC 18370-1:2016 describes an overview of blind digital signature solutions, ISO/IEC 18370-2:2016 specifies discrete logarithm based mechanisms.

More specifically, section 8 of ISO/IEC 18370-2:2016 specifies a DLP-based blind signature protocol with selective disclosure capabilities. Actually, mechanism 4 described in section 8 of ISO/IEC 18370-2:2016 is a standardization of Microsoft® U-Prove anonymous credential system (see clause 6.6.2 of the present document).

Since ISO/IEC 18370 [i.183] is an international standard, which has the potential status to be referenced by EU regulations. This begs the question if ISO/IEC 18370 [i.183] could serve as a standardized selective disclosure protocol for the EUDI Wallet. There are however two critical issues associated with ISO/IEC 18370 [i.183].

The first critical issue with mechanism 4 described in section 8 of ISO/IEC 18370-2:2016 (i.e. U-Prove) is that it does not provide multi-show unlinkability. In other words, it is only possible to present a U-Prove credential once, thereafter additional presentations of the U-Prove credential are linkable.

The second issue is that the U-Prove scheme is broken under certain conditions, as described in the article "On the (in)Security of ROS" [i.22]. Provided that the U-Prove issuance protocol is executed concurrently, it is possible to forge a U-Prove credential. However, U-Prove will remain secure if the issuance protocol is only executed sequentially, but this would not be practical nor user-friendly.

Since ISO/IEC 18370-2:2016 is based on algorithms using the DLP, the scheme cannot be considered as quantum-safe.

Hence, the ISO/IEC 18370 [i.183] standard on blind signatures is not recommended to be considered as a selective disclosure protocol for the EUDI Wallet.

6.6.4 Keyed-Verification Anonymous Credentials (KVAC)

The anonymous credentials systems Idemix (clause 6.6.1) and U-Prove (clause 6.6.2) are based on public key primitives. A different approach, that is based on algebraic Message Authentication Codes (MACs) in prime-order groups, was proposed by Chase et al. in the paper "Algebraic MACs and keyed-verification anonymous credentials" [i.59]. The paper describes two anonymous credentials systems called "Keyed-Verification Anonymous Credentials (KVAC)" as they require the verifier to know the issuer secret key. The KVAC system is based on two algebraic MACs in prime-order groups, along with protocols for issuing credentials, asserting possession of a credential, and proving statements about hidden attributes (e.g. the age of the user). The performance of the KVAC schemes is comparable to U-Prove and faster than Idemix. However, the presentation proof, for n unrevealed attributes, is of complexity O(n) in the number of group elements.

In order to address the complexity issue, a new KVAC system has been designed that provides multi-show unlinkability of credentials and is of complexity O(1) in the number of group elements. This enhanced KVAC scheme was described by Barki et al. in the paper "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials" [i.15]. A new algebraic MAC_BBS+ scheme based on a pairing-free variant [i.50] of BBS [i.27] is also described in the paper.

This KVAC system is suitable for resource constrained environments like SIM-cards, and MAC_BBS+ has been implemented as a prototype on standard SIM-cards. Only the verification process differs between the MAC_BBS+ and BBS+ versions but all other operations remain the same (such as credentials issuance and generation of verifiable presentations). The MAC_BBS+ signatures are therefore equivalent to BBS+ signatures for the KVAC system as a whole. Hence, the verification of a MAC_BBS+ verifiable presentation can be done more efficiently and without pairings, provided that the verifier and the issuer are the same entity and therefore share the issuance private key. This could for example be the case for instance in e-voting or public transportation use cases, where the voting authority respectively public transportation authority manages the virtual ballot box server respectively turnstiles/validators. The BBS+ variant of the KVAC system, which can be seen as the public-key variant of MAC_BBS+, is described in clause 4.4 in the paper "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials" [i.15].

The main drawback of KVAC systems is that they are tailored to specific settings in which the issuer also acts as a verifier, as in the case of e-government or public transportation. They are not suited to the more general setting, envisioned in eIDAS2, in which the issuer and the verifier are two distinct entities that do not necessarily share the issuance private key.

To address this issue, a new KVAC system, called BBS# [i.78], has been designed to provide publicly verifiable and multi-show anonymous credentials from pairing-free elliptic curves. In addition, BBS# allows a credential to be bound to a hardware-protected device key without requiring any change in that hardware or in the algorithms it supports. It leverages MAC_{BBS+} (as described in [i.15]), thus its given name.

In contrast to conventional (pairing-free) KVACs, BBS# enables publicly verifiable showing of credentials, and this is achieved either by allowing the verifier to request the issuer to check the validity of a specific part of a VP or by allowing the holder to interact with the issuer (during a VP or ahead of time) to generate additional ZK proofs.

Requests to or interactions with the issuer preserve the user's anonymity. In particular, interactions with the issuer are entirely oblivious [i.218]: the issuer does not need to authenticate or verify anything about the user with whom it is interacting and can neither link the interaction to another performed by the same user, nor learn anything about the user's credential attributes.

BBS# has been implemented as a prototype on several smartphones using different secure execution environments. [i.78] reports that the time required to generate and verify a presentation is less than 70 ms in high-end mobile devices (using Android StrongBox) and that the size of a BBS# presentation proof is 416 + U × 32 bytes, where U denotes the number of hidden attributes.

More information about BBS# is available in clause 4.4.3.

6.6.5 Fast IDentity Online with Anonymous Credentials (FIDO-AC)

The anonymous credential approaches presented above, with the exception of BBS#, are usually motivated by first considering anonymous credentials and then devising mechanisms to make them compatible with legacy credential formats and hardware, particularly for holder binding. Yeoh et al. have published the research paper "Fast IDentity Online with Anonymous Credentials (FIDO-AC)" [i.269] that is motivated by the converse direction, yet leading to similar constructions: They observe that the FIDO2 standard [i.110] that is used for passwordless authentication and strongly connected to legacy signature mechanisms supported by secure hardware (RSA and ECDSA) is not privacy-friendly, and so is its combination with revealing digital certificates bound to the corresponding cryptographic keys. As such, Yeoh et al. [i.269] enrich FIDO-based authentication with the presentation of attributes of a credential connected to the same cryptographic keypair in a privacy-oriented way that facilitates unlinkability and predicates. Thereby, from a functional perspective, it corresponds to a construction concerning compatibility with legacy formats: There is compatibility with legacy secure hardware, which commonly supports the FIDO2 protocol, and the anonymous credential is constructed from ICAO-compliant electronic passports, which differ from the mdoc standard (ISO/IEC 18013-5 [i.181]) yet have large-scale adoption with around a billion issued electronic passports. Creating anonymous credentials from electronic passports has more broadly received attention, e.g. in the zk-creds paper [i.231] and projects like zkPassport [i.270] - however, it should be noted that owing to the lack of active authentication (involving a PIN) in most electronic passports, this is not suitable for meeting a substantial level of assurance in remote identification owing to the lack of a second factor.

As such, the work not only focuses on cryptographic constructions but also on how the FIDO protocol would need to be augmented to facilitate the selective presentation of attributes and predicates of associated digital certificates, akin to how OID4VP would need to be extended to facilitate ZKP-based presentations of (Q)EAA (see also clause 6.8.2). As in Paquin et al. [i.219], the Groth16 ZKP system is chosen to implement the anonymous credential capabilities, which leads to a proving time of around 3 seconds on a Google Pixel 6 (Frigo and Shelat [i.113] use the same device for benchmarking). The authors also claim that preprocessing can further reduce the time for creating the ZKP, yet they do not further specify how the relying party's random challenge is considered. The core contribution of the paper, therefore, seems to be rather the provably secure integration of presenting anonymous credentials from legacy formats into the FIDO protocol than designing a novel cryptographic algorithm or performance optimizations of circuit-based ZKPs. With the affordances of the FIDO2 protocol (that caters for strong authentication and phishing protection) being largely covered by the means of presenting (Q)EAA at least at a substantial level of assurance, the relevance of this work ([i.269]) for data minimization in presentations with the EUDI Wallets may, therefore, be limited.

6.7 ISO mobile driving license (ISO mDL)

6.7.1 Introduction to ISO/IEC 18013-5 (ISO mDL)

The ISO mobile driving license (ISO mDL) is specified in the ISO/IEC 18013-5 [i.181] standard, which on a high level can be divided in the device retrieval flow (see clause 6.7.2) and the server retrieval flows (see clause 6.7.3) for selective disclosure of the user's mdoc.

ISO/IEC CD 18013-7 [i.182] is a draft specification that extends the ISO/IEC 18013-5 [i.181] standard with unattended flows (see clause 6.7.3), which are online protocols for selective disclosure of the user's mdoc to a web hosted mdoc reader.

6.7.2 ISO/IEC 18013-5 (device retrieval flow)

The device retrieval flow is described in ISO/IEC 18013-5 [i.181], clauses 6.3.2, 6.3.2.1 (as flow 1) and 6.3.2.4.

The credential format is the mdoc, which contains the attributes about the user, in conjunction with the Mobile Security Object (MSO). The MSO is a signed object that contains a list of salted attribute hashes of the user's attributes. The MSO caters for selective disclosure based on the salted attribute hashes as described in clause 5.3.2.

The selected attributes of the mdoc and the MSO are presented by the user's mdoc app to an mdoc reader by using BLE, NFC or WiFi. The mdoc reader verifies the MSO and the selectively disclosed attributes (see clause D.2 for more information on the device retrieval flow).

ISO/IEC 18013-5 [i.181] is considered mature, and several device retrieval solutions have been deployed in production, for example in a number of states in the US.

The MSO and DeviceSignedItems can be signed with cryptographic algorithms that are currently approved by SOG-IS [i.237]. Since the MSO and DeviceSignedItems are signed with a COSE-formatted signature, this caters for MSOs to be signed in the future with QSC algorithms as discussed in the IETF report "JOSE and COSE Encoding for Post-Quantum Signatures" [i.149].

NOTE: Although DeviceSignedItems can be signed with candidate quantum-safe signatures, the issue of having a quantum-safe key agreement mechanism to secure the communication channel remains. The ephemeral session keys between the mdoc device and the reader are currently exchanged using the ECKA-DH key agreement, which is vulnerable to quantum computing attacks. Furthermore, MAC signatures are mentioned in ISO/IEC 18013-5 [i.181] as offering better privacy guarantee, but the MAC secret is derived from an ECKA-DH key agreement, which is exposed to the quantum computing vulnerability. An extensive analysis of the session key exchange goes beyond the scope of the present document, however, but this quantum computing vulnerability should be observed.

The device retrieval flow has been selected as a PID protocol for the EUDI Wallet as specified in the ARF [i.71].

An extensive analysis of the device retrieval flow, and how it can be applied for eIDAS2 QTSPs and EUDI Wallet PID/(Q)AEE, is available in clause 7.2.3.

6.7.3 ISO/IEC 18013-5 (server retrieval flows)

The server retrieval flows are described in ISO/IEC 18013-5 [i.181], clause 9.2.

The server retrieval flow can be initialized as a hybrid device/server process (see annex D.2.2) or as a server process (see annex D.2.3). Once the server retrieval flow has been initialized, it continues with either the WebAPI flow or the OpenID Connect (OIDC) flow.

In the WebAPI flow the mdoc Reader submits a server retrieval WebAPI Request with a list of requested DataElements to the Issuing Authority. Upon the user's consent, the Issuing Authority will reply with the mdoc Response with the selected and disclosed DataElements (see clause D.2.4 for more information).

In the OIDC flow the mdoc Reader (OIDC client) submits a server retrieval OIDC Request with the requested data elements (JWT claims) to the Issuing Authority, which operates an OIDC Authorization Server. This activates the OIDC authorization code flow [i.212]. Based on the user's consent, the Issuing Authority (OIDC Authorization Server) will reply to the mdoc Reader (OIDC client) with the OIDC Token with the selected and disclosed JWT claims about the user (see clause D.2.5 for more information).

ISO/IEC 18013-5 [i.181] and OIDC standards are considered mature, and several server retrieval solutions have been deployed in production, for example in a number of states in the US.

The WebAPI and OIDC tokens are JWTs that can be signed with cryptographic algorithms that are currently approved by SOG-IS [i.237]. Since the WebAPI and OIDC tokens are signed with a JOSE-formatted signature, this caters for those JWTs to be signed in the future with QSC algorithms as discussed in the IETF report "JOSE and COSE Encoding for Post-Quantum Signatures" [i.149].

An extensive analysis of the server retrieval flow, and how it can be applied for eIDAS2 QTSPs and EUDI Wallet PID/(Q)AEE, is available in clause D.2.4.

6.7.4 ISO/IEC 18013-7 (unattended flow)

ISO/IEC CD 18013-7 [i.182] draft standard extends ISO/IEC 18013-5 [i.181] with the unattended flow, i.e. the online flow whereby an mdoc app connects directly to an mdoc reader that is hosted as a web server application. ISO/IEC CD 18013-7 [i.182] is backward compatible with the protocols specified in ISO/IEC 18013-5 [i.181].

ISO/IEC CD 18013-7 [i.182] unattended flow is based on the following protocols:

- Device Retrieval from an mdoc app to a web server application by using REST APIs over HTTPS POST; this flow is described in clause D.3.1.
- OpenID for Verifiable Presentations (OID4VP) [i.214] in conjunction with Self-issued OpenID Provider v2 (SIOP2) [i.216]; this flow is described in clause D.3.2.

Both protocols for the unattended flow transmit the selectively disclosed mdoc attributes in conjunction with the MSO from the mdoc app to the mdoc reader. The mdoc attributes and the MSO are verified according to the same principles as for the mdoc device retrieval flow (see clause 7.2.3).

As described in clause 6.7.1, the MSO can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use.

ISO/IEC CD 18013-7 [i.182] is still a draft, so there are no real deployments in production. NIST NCCoE will carry out interoperability tests [i.206] with an ISO/IEC CD 18013-7 [i.182] compatible reader during the course of 2023 and 2024.

The proximity unattended flow has been selected as a PID protocol for the EUDI Wallet as specified in the ARF [i.71].

An extensive analysis of the unattended flow, and how it can be applied for eIDAS2 QTSPs and EUDI Wallet PID/(Q)AEE, is available in clause D.3.

6.7.5 ISO/IEC 23220-4 (operational protocols)

ISO/IEC CD 23220-4 [i.187] is a draft specification describing operational (presentation) protocols for a digital wallet. The specification expands on ISO/IEC 18013-5 [i.181] with reader engagement, internet online connections to a reader, and bridges to additional standards for user authorization such as OID4VP [i.214] and credential formats such as W3C Verifiable Credentials [i.264].

ISO/IEC CD 23220-4 [i.187] presentation protocols are based on the following protocols:

- Device Retrieval from a digital wallet to a web server application by using REST APIs over HTTPS POST.
- OpenID for Verifiable Presentations (OID4VP) [i.214] in conjunction with Self-issued OpenID Provider v2 (SIOP2) [i.216].

More specifically, Annex B in ISO/IEC CD 18013-7 [i.182] draft specification refers to ISO/IEC CD 23220-4 [i.187] for the OID4VP/SIOP2 profile to be used for presentation of the mdoc in an ISO/IEC CD 18013-7 [i.182] unattended flow. As described in clause 6.7.1, the MSO can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use.

Furthermore, Annex B in ISO/IEC CD 23220-4 [i.187] WD9 describes how to present W3C Verifiable Credentials [i.264] in conjunction with IETF SD-JWT [i.155] for selective disclosure. The SD-JWT can be signed with SOG-IS approved cryptographic algorithms and allows for QSC algorithms for future use (see clause 7.3).

In order to secure the HTTPS connection to an online reader (relying party), ISO/IEC CD 23220-4 [i.187] recommends the use of QWACs.

ISO/IEC CD 23220-4 [i.187] is still a draft, so there are no real deployments in production. However, the ARF [i.71] refers to ISO/IEC CD 23220-4 [i.187] as an alternative attestation exchange REST API protocol.

6.8 OpenID for Verifiable Credentials (OpenID4VC)

6.8.1 OpenID for Verifiable Credential Issuance (OpenID4VCI / OID4VCI)

OpenID for Verifiable Credential Issuance specifies an OAuth-protected API for the issuance of Verifiable Credentials of different formats. To enable secure digital credential issuance and provisioning across different platforms and providers, a standardized protocol is essential. The protocol provides support for different options and features to meet the requirements of different ecosystems and Issuers:

- Initiation of flows by the Wallet or the Issuer
- User information and authorization via an Authorization Server or out-of-band mechanisms
- Immediate or deferred issuance for cases where the Issuer cannot issue credentials immediately
- Batch Issuance - a way to request and receive batches of credentials bound to different keys
- Securely binding credentials to secret keys generated by the Wallet
- Key Attestations that provide additional trust in the key storage (WSCD)
- Wallet Attestations that provides additional trust in the Wallet application and device
- Display metadata that allows for a customized display of credentials within a Wallet

Describing the whole protocol and all of its features in detail would be too much for the scope of the present document. Instead, the present clause will focus on some key features around key derivation, batch issuance and general support for ZKPs.

OpenID4VCI has built-in support for batch issuance. The flow starts with discoverable issuer metadata by which an Issuer can signal if batch issuance is supported and if so, what maximum batch sizes can be requested in one interaction. Upon discovering the support for batch issuance and after successful authorization, a Wallet can choose how many credentials it wants to receive by sending the appropriate amount of proof objects (either a proof of possession or a key attestation) for key-bound credentials. The Issuer checks those proofs for correctness and responds with an array of credentials bound to the keys that were provided. Batch issuance and the issuance of a single credential do not differ in the general flows, just in the amount of key proofs and credentials.

OpenID4VCI has a defined extension point for proving possession of private keys in a credential request by the Wallet. There are currently 3 defined types, a jwt proof that is a simple proof of possession by which the Wallet demonstrates possession of a private key that belongs to a public key that a credential gets bound to that is used for mdoc and SD-JWT, a proof of possession for W3C based credentials using Data Integrity proofs, and key attestations where the key storage of a Wallet makes a trusted statement that specific keys were generated in hardware.

To better support key derivation mechanisms like ARKG, another variant could be added to allow a single proof of possession for a public master key and the issuance of a batch of credentials bound to derived keys. While this is currently not part of the OpenID4VCI specification, it is part of ongoing discussions and would be easy to add leveraging the defined extension points.

6.8.2 OpenID for Verifiable Presentations (OpenID4VP / OID4VP)

OpenID for Verifiable Presentations is a mechanism to request and deliver presentations of digital credentials of different credential formats. OID4VP is built on top of OAuth2.0 and extends it by introducing a new return type called VP Token that serves as a container holding one or more presentations that a Wallet provides to a Relying Party. Similar to OD4VCI, OID4VP was built in a way to be credential format agnostic and pre-defines credential format profiles for mdoc, SD-JWT VC, and W3C VCDM based credentials.

OID4VP supports two different flows, the same device flow where the wallet is invoked from another application (e.g. the browser) on the same device, or a cross device flow where the wallet is invoked from a different device. It describes a profile that allows OID4VP to work with the Digital Credentials (DC) API that allows browsers to natively invoke wallets using the DC API and improves the security of cross device flows by leveraging the underlying transport of the Client to Authenticator Protocol (CTAP). OID4VP provides for different options to authenticate a Relying Party by introducing a new client ID Prefix mechanism that allows different ecosystems to use different trust mechanisms.

OID4VP also introduces a transaction data mechanism where a credential presentation can also be bound to a specific transaction authorization, allowing for authorization for use-cases like digital document signatures, and payment transactions.

OID4VP introduces its own JSON based query language called Digital Credentials Query Language (DCQL) that:

- is credential format agnostic;
- allows for queries spanning multiple credentials;
- allows for different options to fulfil a request (A or B type queries);
- allows to query trusted authorities - matching issuers or trust frameworks.

DCQL was designed in a way that most parts of the query language do not depend on the credential format, but some parts like type matching are defined per credential format. This allows for the overall query language to stay relatively simple but cater for the different requirements and payloads of the different credential formats. DCQL allows to express requests on individual attribute level and also supports optional filtering on expected values. A response within OID4VP allows for an easy matching from the different presentations provided in a VP token to the specific query parts in the request they belong to. This is especially important for queries that have some level of optionality.

While DCQL was not built explicitly for Zero-Knowledge Proofs, it can be used for simple queries for ZKPs and there are extension points for more complex constructions like predicate proofs or composite proofs that can be added once those requirements are fully understood.

6.8.3 OpenID4VC High Assurance Interoperability Profile (HAIP)

The OpenID4VC High Assurance Interoperability Profile (HAIP) defines a profile of the OID4VCI and OID4VP protocols and mdoc and SD-JWT VC credential formats. The aim of the profile is to define a subset of features and a set of mandatory requirements for those specifications to create interoperable implementations for use-cases where a high level of security and privacy is required. The profile does not define trust management, but mandates support for:

- X.509 based verifier authentication (signed requests);
- Response Encryption (for both OID4VCI, and OID4VP);
- specific signature and encryption algorithms.

The core goal of HAIP is to narrow down choices of the different protocols and credential formats to create a manageable subset of choices that will guarantee interoperable and secure implementations.

6.9 The Iden3 protocol

6.9.1 Introduction to the Iden3 protocol

Iden3 [i.139] has its origin in the development of Circom [i.68], which is a language to implement constraint systems for ZKPs, and SnarkJS, a library for generating and verifying zk-SNARKs based on the Groth16 proof system. Circom has been benchmarked in the report "Benchmarking ZK-Friendly Hash Functions and SNARK Proving Systems for EVM-compatible Blockchains" [i.127] and SnarkJS is the first general-purpose zk-SNARK prover capable of running on edge devices. As such, the work underlies the implementation of Crescent [i.219] and the proposal by Babel and Sedlmeir [i.14]. Circom has undergone major improvements since its first release and is one of the more popular frameworks for implementing ZKPs in the Web3 space, with an active development community that implemented complex libraries for, e.g. ECDSA signature verification [i.69] and regular expression verification [i.272].

6.9.2 Cryptography behind the Iden3 protocol

The Iden3 protocol uses zk-SNARKs to conduct efficient verification for regular and blockchain (smart contract) applications completely removing the need for verifier to communicate with issuer to perform verification of zero-knowledge proofs of predicates and selective disclosure, as well as verification of credential non-revocation.

Iden3 utilizes a combination of the Groth16 zk-SNARK proving scheme (on the BN254 curve), Sparse Merkle Trees, BabyJubJub EdDSA signatures, and the Poseidon hash function [i.122].

- Groth16 [i.124] is a widely used zk-SNARK, especially in the blockchain space, because of its performance to on-chain verification cost ratio. The Iden3 protocol is designed to be pluggable such that it can utilize different proof systems. Plonk is another zk-SNARK system, which the Iden3 protocol is capable of utilizing to generate ZK proofs.

- Iden3 utilizes Sparse Merkle Trees (SMTs), which are a variation of Merkle Trees allowing not only proving set inclusion, but also non-inclusion as part of its cryptographic infrastructure to facilitate data integrity and selective disclosure. By using SMTs, Iden3 can verify data elements in large sets of data by hashing them into a single root hash. The Iden3SparseMerkleTreeProof [i.140] proves that the specific issuer has issued this verifiable credential by a Merkle tree proof that this claim is included in the issuer's Claims Merkle tree, and is therefore in the issuer's state. This state is published by the issuer to a trusted ledger. Since this algorithm does not use any kind of signature it is stronger against potential quantum attacks, versus other signature algorithms.

- BabyJubJub [i.268] is an elliptic curve, which can be used inside any zk-SNARK circuit, utilizing a BN254 pairing friendly elliptic curve.

To verify a zk-SNARK proof, it is necessary to use an elliptic curve. The basic curve is alt_bn128 (also referred to as BN254), which has prime order *r*. But while it is possible to generate and validate proofs of *F_r* -arithmetic circuits with BN254, it is not possible to use BN254 to implement elliptic-curve cryptography within these circuits. Baby Jubjub is an elliptic curve defined over the finite field *F_r* which can be used inside any zk-SNARK circuit, allowing for the implementation of cryptographic primitives that make use of elliptic curves, such as the Pedersen Hash, Poseidon Hash or the Edwards Digital Signature Algorithm (EdDSA). Baby Jubjub curve satisfies the SafeCurves criteria. Baby Jubjub is a twisted Edwards curve birationally equivalent to a Montgomery curve [i.52]. The algorithm chosen for generating Baby Jubjub is based on the criteria defined in IETF RFC 7748 [i.166].

In the context of ZKPs, SMTs enable Iden3 to allow users to generate predicate proofs and selectively disclose specific VC attributes while maintaining the privacy of other elements. This data structure is also used to implement one of the ways to issue credentials, privacy-preserving credential revocation methods and hiding a user's real identity behind identity "Profiles" (pseudonymous identifiers with strong cryptographic identity holder binding).

6.9.3 Implementation aspects of the Iden3 protocol

The implementation of PrivadoID / Billions Network leverages the tooling that was originally developed for a blockchain-based rollup, which also includes optimizations of the Rapidsnark assembly prover that has been extended from servers to mobile phone instruction sets to improve client-side proof generation performance.

While the Iden3 implementation is based on a circuit-based ZKP, the current implementation is not focused on compatibility with legacy formats. The main motivation for doing so follows the line of reasoning of the zk-creds paper [i.231]: With ZKP-friendly hashes like Poseidon being fast to prove in a circuit-based ZKP compared to digital signatures, verification can leverage a blockchain-based list of hashes or a cryptographic accumulator, e.g. a Merkle tree) that cryptographically anchors the certificate. This list can then also be used as a revocation registry; yet, the tradeoff is that offline verification without revocation checks is not possible anymore in this setting. Alternatively, the Iden3 implementation offers a signature-based verification based on EdDSA - essentially, a Schnorr signature on the ZKP-friendly BabyJubJub curve [i.268]. EdDSA is also used by Babel and Sedlmeir [i.14] to obtain a faster proving speed. While binding to legacy hardware via RSA or ECDSA signatures could be implemented by leveraging existing tooling for Circom or corresponding improved variants in Crescent [i.219] or the work "Anonymous credentials from ECDSA" by Frigo and Shelat [i.113], the main application of the Iden3 protocol thus far has been the on-chain verification of credentials in blockchain projects, which typically tend to require less compatibility with legacy systems and at the same time often prioritize proof size over proof generation effort, such that Groth16 [i.124] is a suitable choice. The credential format used by Iden3 is JSON Web Zero-knowledge (JWZ) [i.141], which is described in clause 5.5.3. Yet, it should be noted that the Iden3 implementation also features an implementation of "anonymous Aadhaar" that showcases compatibility with legacy certificate formats involving RSA signatures (imported from the zk-email project).

Notably, while the Iden3 stack focuses on developing several smart contracts for identity verification, the use in bilateral settings is straightforward and the corresponding verifier components are required artifacts in the development of the corresponding smart contracts for blockchain-based verification. The circuits provide different means of credential presentation, including the selective disclosure of single attributes and corresponding range proofs. Moreover, the implementation is compatible with the emerging verifiable credentials standard, although it should be mentioned that the cryptographic representation of the certificate is different to avoid the need for costly parsing. Besides the cryptographic implementation, a valuable feature offered by the Iden3 stack is the development of a query language that simplifies the formulation of checks expected by the relying party. The corresponding modular design of the circuits hence obviates the need for storing a high number of proving keys (generated during the trusted setup) in the EUDI Wallet.

7 Implications of selective disclosure on standards for (Q)EAA/PID

7.1 General implications

The purpose of clause 7 is to analyse the implications of selective disclosure and unlinkability on ETSI standards for (Q)EAAs and PIDs.

More specifically, the (Q)EAA/PID credentials discussed in the following clauses 7.2 and 7.3 are scoped to ISO/IEC 18013-5 [i.181] mdoc and SD-JWT, because these formats are explicitly specified as selective disclosure formats for PIDs in the ARF [i.71]. The main reason why mdoc and SD-JWT were selected in the ARF [i.71] as (Q)EAA/PID credentials is that they can be signed with cryptographic algorithms that are currently approved by SOG-IS [i.237], and that the credentials also allow for being signed with Quantum-Safe Cryptography (QSC) algorithms for future use. More technical details on how the issuer may apply such signatures on mdoc and SD-JWT are discussed in clauses 7.2.1 and 7.3.1 respectively.

Furthermore, clause 7.4 analyses the possibilities of using BBS+ credentials as (Q)EAA/PID. The reason for analysing BBS+ is due to the emerging ISO standardization of BBS+, which may be used with W3C VCDM in conjunction with W3C VCDI. Since BBS+ with blinded signatures is fully unlinkable, it would be a viable alternative from a privacy preserving perspective. This in turn may cater for BBS+ to be referenced in a future version of the ARF and/or the ETSI TS 119 472-1 [i.97] standard on (Q)EAAs profiles.

Also, clause 7.5 analyses solutions that utilize programmable ZKPs such as zk-SNARKs in conjunction with existing digital infrastructures. The reason for analysing such solutions is that they can provide fully unlinkable presentations that provide selectively disclosed attributes and revocation information, based on existing eIDAS X.509 QCs and the forthcoming eIDAS2 (Q)EAAs/PIDs. This in turn may cater for zk-SNARK based solutions to be referenced in a future version of the ARF and/or the ETSI TS 119 462 [i.95] standard on EUDI Wallet interfaces.

The analysis in clause 7 is primarily focused on selective disclosure and unlinkability since those characteristics are defined in eIDAS2 [i.103] and the ARF outline [i.70]. Predicates are described on a high level, with proposals on how to implement them for the selected PID credentials mdoc and SD-JWT.

The selected (Q)EAA/PID credentials are analysed with respect to the issuance by a QTSP/PIDP, how the credentials are stored in the EUDI Wallet, and how selected attributes are presented to a relying party.

Firstly, it is analysed how the QTSP or PID provider may issue (Q)EAAs/PIDs with capabilities for selective disclosure. This analysis also describes the PKI trust models for the issuance process and whether EU Trusted Lists (EU TLs) can be applied. Furthermore, it is described how the (Q)EAAs/PIDs should be issued to cater for unlinkability. The recommended policies and practices for such QTSP/PIDP issuance processes are discussed for mdoc in clause 7.2 and SD-JWT in clause 7.3.

Secondly, it is analysed how the (Q)EAAs/PIDs with capabilities for selective disclosure and unlinkability are stored in the EUDI Wallet. This analysis also describes the associated cryptographic keys used for proving the user's ownership of the (Q)EAAs/PIDs. The implications for storing the (Q)EAAs/PIDs with selective disclosure in an EUDI Wallet are discussed for mdoc in clause 7.2 and SD-JWT in clause 7.3.

Thirdly, it is analysed how the selected attributes can be presented to a relying party, yet sustaining unlinkability. The recommended policies and practices for presenting the (Q)EAAs/PIDs with an EUDI Wallet are discussed for mdoc in clause 7.2 and SD-JWT in clause 7.3.

7.2 Implications for mdoc with selective disclosure

7.2.1 QTSP/PIDP issuing mdoc

7.2.1.1 General

The mdoc, as specified in ISO/IEC 18013-5 [i.181], is composed by the the user's elements, the authentication key, and the Mobile Security Object (MSO) with a signed list of salted hash values of these elements. The MSO is a CBOR-encoded [i.170] object, which is signed by the issuer with a COSE-formatted signature [i.167].

ISO/IEC 18013-5 [i.181] describes the Issuing Authority Certification Authority (IACA) that is the root CA that used for issuing subordinated certificates, which in turn are used for signing the user's MSOs, signing revocation data (OCSP-responses and CRLs), and securing online services (JWS and TLS).

Clauses 7.2.1.2 to 7.2.1.6 compare and map the requirements on ISO mDL compliant IACAs into considerations for eIDAS2 compliant QTSPs/PIDPs when issuing ISO mDL with capabilities for selective disclosure and (predetermined) predicates. Clauses 7.2.1.2 to 7.2.1.6 also provide a summary of the ISO mDL and its Issuing Authorities, but it is recommended to have studied ISO/IEC 18013-5 [i.181] before to have an understanding of the ISO mDL ecosystem.

7.2.1.2 Certificate profiles

The IACA's trust anchor is a DER-encoded X.509 certificate that should be issued according to the certificate profile in ISO/IEC 18013-5 [i.181], Annex B.1. ISO/IEC 18013-5 [i.181], Annex B.1.1 declares that all X.509 certificates are DER-encoded and specifies the generic certificate requirements on certificate extensions and subjects. The IACA certificate profile also defines the cryptographic algorithms that are approved by ISO/IEC 18013-5 [i.181].

In the context of eIDAS2, the cryptographic algorithms used in the QTSP/PIDP CA certificates are required to comply with the SOG-IS list of EU approved cryptographic algorithms [i.237]. Hence, the QTSP/PIDP CA certificates used for issuing ISO mDLs are required to comply with the intersection of IACA's and SOG-IS' requirements on cryptographic algorithms.

EXAMPLE 1: SOG-IS [i.237], section 4.3 "Discrete Logarithm in Elliptic Curves" lists the following approved ECC curves: BrainpoolP256r1, BrainpoolP384r1, and BrainpoolP512r1.

EXAMPLE 2: ISO/IEC 18013-5 [i.181], Table B.3 "Document signer certificate" lists the following approved ECC curves: BrainpoolP256r1, BrainpoolP320r1, BrainpoolP384r1, BrainpoolP512r1, Curve P-256, Curve P-384, and Curve P-521.

The IACA trust anchor is used for issuing the following subordinated certificates in an IACA PKI:

- mDL MSO signer certificate (ISO/IEC 18013-5 [i.181], Annex B.1.2).
- JWS signing certificate (ISO/IEC 18013-5 [i.181], Annex B.1.3.1).
- TLS server certificate issuing authority (ISO/IEC 18013-5 [i.181], Annex B.1.6).
- TLS client authentication certificate (ISO/IEC 18013-5 [i.181], Annex B.1.8).
- OCSP signer certificate (ISO/IEC 18013-5 [i.181], Annex B.1.9).

Furthermore, the ISO mDL IACA CRL profile is specified in Annex B.2 in ISO/IEC 18013-5 [i.181].

An eIDAS2 QTSP/PIDP that issues ISO mDLs should adhere to the IACA PKI and the certificate and CRL profiles described above.

One more alternative could be for ETSI to assign a specific QC extension to be used for trust anchor certificates that are used by accredited QTSPs to issue ISO mDLs.

7.2.1.3 Trusted Lists

According to article 22(1) of eIDAS [i.104], each EU Member State is required to publish a Trusted List (TL) with all QTSPs in that EU Member State. All information referred to in eIDAS article 22(3), including the location and signing certificates of the TLs, is compiled in the EU List Of Trusted Lists (LOTL). Furthermore, the Commission Implementing Directive (CID) 2015/1505 [i.100] mandates the use of ETSI TS 119 612 [i.94] for the implementation of the trusted lists. ETSI TS 119 612 [i.94] specifies the format and mechanisms for establishing, locating, accessing and authenticating trusted lists. The EU TLs and EU TOTL are XML-encoded according to specific XML schemas and signed with XAdES-signatures as specified in ETSI TS 119 612 [i.94].

ISO/IEC 18013-5 [i.181] has introduced a similar concept called Verified Issuer Certificate Authority List (VICAL), which contains the trustworthy IACA's that issue certificates for creating and operating ISO mDLs. An ISO mDL VICAL can be formatted and signed either in CDDL [i.170] or CMS [i.157] format. The ISO mDL VICAL Providers publishes the VICALs. ISO/IEC 18013-5 [i.181], Annex C specifies the policy and security requirements and technical and procedural controls for a VICAL Provider.

NOTE: ISO/IEC 18013-5 [i.181], Annex C refers to ETSI EN 319 411-1 [i.90] and FPKIPA X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework [i.109] for the operations of an ISO mDL VICAL Provider.

Hence, there are synergies between the EU TLs and the ISO mDL VICALs, in the sense that both trusted lists contain trust anchors. The main differences are the encodings and signature formats (EU TL XML/XAdES versus ISO mDL VICAL CDDL/CMS). In order to bridge this gap, ETSI TS 119 612 [i.94] may specify a CDDL/CMS profile of the EU TL that is compatible with the ISO mDL VICAL, or ISO/IEC 18013-5 [i.181] may be extended to specify an XML profile of the VICAL that is compatible with the ETSI EU TLs. In such a scenario, an eIDAS2 accredited QTSP/PIDP could issue CA certificates that are included in an EU TL, which in turn could be trusted as a VICAL in the ISO mDL ecosystem.

In summary, transposing ISO/IEC 18013-5 [i.181], Annex C to an eIDAS2 context results in the following recommendations:

- The ISO mDL Issuing Authority corresponds to the eIDAS2 QTSP/PIDP.
- The IACA trust anchor should be issued as a trust anchor by the eIDAS2 QTSP/PIDP that issues ISO mDL as (Q)EAA/PID.
- The eIDAS2 QTSP/PIDP should ensure that its IACA trust anchor is published in the EU TL, which is issued by the supervisory body in the applicable EU Member State.
- ETSI TS 119 612 [i.94] may specify an additional CDDL/CMS profile of the EU TL that is compatible with the ISO mDL VICAL, or ISO/IEC 18013-5 [i.181] may be extended to specify an XML profile of the VICAL that is compatible with the ETSI EU TLs.
- The EU TLs may include a specific extension for the QTSPs that are authorized to issue QEAAs that also are compliant with ISO mDL; the EU TL extension can reference the ISO mDL VICAL where the QTSP is also listed.

7.2.1.4 Issuance of mdocs

An mdoc, which has been issued to the user's EUDI Wallet on a device, is essentially composed of the user's data elements and the MSO, which are associated with the mdoc authentication key (see clause 7.2.2).

The data elements inside an mdoc consist of an unsigned list of the user's elements belonging to the nameSpace "org.iso.18013.5.1", as defined in ISO/IEC 18013-5 [i.181] for mDL.

The MSO (mobile security object) is defined in ISO/IEC 18013-5 [i.181], section 9.1.2.4 as a signed object, which contains the mDL authentication public key and a list of salted attribute hashes of the user's elements. The MSO is signed with a COSE-formatted signature, by the IACA's MSO signer certificate.

NOTE 1: In the context of eIDAS2, a QTSP/PIDP will issue an MSO signer certificate with cryptographic algorithms that are approved by both SOG-IS [i.237] and ISO/IEC 18013-5 [i.181].

NOTE 2: Since the MSO's signature is COSE-formatted, QSC algorithms can also be considered for the future according to the IETF IESG report [i.149].

According to section E.8.4 of ISO/IEC 18013-5 [i.181] and sections E.8.4 and E.5 of ISO/IEC CD 23220-3 [i.186] it is recommended that the mdoc authentication keys and related MSOs are updated frequently to achieve unlinkability when presenting the ISO mDL elements multiple times. Hence, the QTSP/PIDP should establish processes for issuing multiple MSOs to the user's EUDI Wallet, typically in batches prior to the device retrieval use of the MSOs. The EUDI Wallet may also signal to the QTSP/PIDP when it is necessary to refresh the MSOs. When issuing a new MSO, the random salts in IssuerSignedItems for the hash calculations should be unique such that the random salted hash values differ for each MSO, even if the user's data elements remain the same.

EXAMPLE 1: Assume that the user's GivenName in the mdoc is "Smith". If the GivenName is combined with random salt S1 and hashed, the resulting hash value becomes H1 in the first MSO. If the same GivenName name is combined with another random salt S2 and hashed, the resulting hash value becomes H2 in the second MSO.

mdoc does not support predicates in the sense that Zero-Knowledge Proofs or range proofs can be dynamically derived based on the elements in the mdoc. However, ISO/IEC 18013-5 [i.181], section 7.2.5 specifies the possibility to insert predetermined Boolean elements as "age_over_NN" in the ISO mDL.

EXAMPLE 2: The Boolean statement "age_over_18" could be an element in the mdoc.

NOTE 3: It is possible to include signed computational inputs and parameters to enable dynamic predicates (see clause B.1).

In order to achieve (predetermined) predicates, the issuing QTSP/PIDP should establish processes to identify the relevant Boolean statements and insert them as elements in the mdoc.

7.2.1.5 Comparison with ETSI certificate profiles for Open Banking (PSD2)

ETSI TC ESI has specified certificate profiles and TSP policy requirements for Open Banking in the sector specific ETSI TS 119 495 [i.93]. The scope of ETSI TS 119 495 [i.93] is:

- Specifies requirements for qualified certificates for electronic seals and website authentication, to be used by payment service providers in order to meet needs of Open Banking including the EU PSD2 [i.101] Regulatory Technical Standards (RTS) [i.98].
- Specifies additional TSP policy requirements for the management (including verification and revocation) of additional certificate attributes as required by the above profiles.

In summary, a QTSP can issue PSD2 compliant certificates (QWACs or QCert for eSeal), using the certificate profile specified in ETSI TS 119 495 [i.93] as follows. The PSD2 specific attributes are checked by the (Q)TSP as part of the identity proofing, as specified in the ETSI TS 119 495 [i.93], REG-6.2.2-1, which states: *"The TSP shall verify the Open Banking Attributes (see clauses 5.1 and 5.2) provided by the subject using authentic information from the Competent Authority (e.g. a national public register, EBA PSD2 Register, EBA Credit Institution Register, authenticated letter)."* The European Banking Association (EBA) maintains a register of payment institutions [i.86], which can be used for that purpose. As a result, a QCStatement extension with Open Banking attributes is included in the PSD2 certificate, which proves its compliance with the PSD2 RTS.

A relying party intending to validate a PSD2 certificate usually performs a two step validation approach:

1) The relying party validates the qualified status of the certificate using the EU TLs.
2) The relying party confirms the correctness of the PSD2 attributes included in the certificate QCStatement using either the national public registers, or the EBA register. The relying parties need to have out-of-band knowledge of where to retrieve the EBA register.

The ETSI TS 119 495 [i.93] requirements for (Q)TSPs issuing PSD2 certificates may partially be re-used also for the issuance of mdocs, but with the following differences:

- The format will be (Q)EAA for mdoc instead of X.509 certificates.
- The relying party will confirm that the QTSP having issued the (Q)EAA is authorized to issue this specific type of (Q)EAA by looking into a domain-specific list, i.e. the ISO mDL VICAL.

- To facilitate the validation of (Q)EAAs being used ISO mDLs, EU TLs could be used to point towards the domain-specific VICAL list where a QTSP is listed as being authorized for a specific scope. Alternatively, an URI for accessing this domain-specific VICAL list could be included in the mdoc (Q)EAA itself, although this may be too static as this URI may change over time.

7.2.1.6 Mapping of mdoc and eIDAS2 terms

As discussed in clauses 7.2.1.1 to 7.2.1.5, there are several equivalences between the terms in ISO/IEC 18013-5 [i.181] and the terms in eIDAS2 [i.103] and the ARF [i.71].

Table 1 provides a mapping of eIDAS2 and ARF terms with the syntax used in ISO/IEC 18013-5 [i.181].

Table 1: Mapping of eIDAS2/ARF and ISO/IEC 18013-5 [i.181] terms

Terms in eIDAS2 and the ARF	Terms in ISO/IEC 18013-5 [i.181] (mDL)
End users of EUDI Wallets	mdoc Holder
EUDI Wallet issuers	Technology Providers
Person Identification Data Providers	Issuing Authorities
Providers of registries of trusted sources (e.g. EU TL)	Verified Issuer Certificate Authority List (VICAL) Providers
Qualified and non-qualified electronic attestation of attributes (qEAA) providers	Issuing Authorities
QTSPs for issuing qualified and non-qualified certificate for electronic signature/seal providers	Issuing Authority Certification Authority (IACA)
Providers of other trust services	Not defined
Authentic sources	Governmental authoritative source
Relying parties	mdoc Reader, operated by a mdoc verifier
Conformity Assessment Bodies (CAB)	Auditing Bodies following ISO/IEC 27001 [i.189] and ISO/IEC 27002 [i.190]
Supervisory bodies	Auditing Bodies following ISO/IEC 27001 [i.189] and ISO/IEC 27002 [i.190]
Device manufacturers and related subsystems providers	Technology Providers
Catalogue of attributes and schemes for the attestations of attribute providers	mdoc namespace

7.2.2 EUDI Wallet mdoc authentication key

The mdoc authentication key is used to prevent cloning of the mdoc and to mitigate man in the middle attacks. The mdoc authentication key pair consists of a public and a private key denoted as (SDeviceKey.Priv, SDeviceKey.Pub). The mdoc authentication public key is stored as the DeviceKey element in the MSO, and the corresponding mdoc authentication private key is used for signing the response data contained in the DeviceSignedItems structure (see ISO/IEC 18013-5 [i.181], sections 9.1.3, 9.1.2.4 and 9.1.3.3 for more information).

Hence, the mdoc authentication key is used by the EUDI Wallet for authentication of selectively disclosed data elements that are presented to a relying party (see clause 7.2.3).

More information on how to store the data elements, MSO, and the mdoc authentication key is available in clause 7.6.

See also clause 4.3.4.2 on the possibility to use Hierarchical Deterministic Key derivation functions where the MSO issuer can issue a batch of MSOs, each with a unique and unlinkable DeviceKey element derived from a single DeviceKey element.

7.2.3 EUDI Wallet used with ISO mDL flows

How the EUDI Wallet can be used with the different types of ISO mDL flows is described in Annex D.

7.3 Implications for SD-JWT selective disclosure

7.3.1 Analysis of using SD-JWT as (Q)EAA format applied to eIDAS2

An analysis of the IETF SD-JWT formats applied to an eIDAS2 context results in the following observations and recommendations:

- The present document recommends using SD-JWT VC as a standalone attestation format where selective disclosure is required. When verifier unlinkability is required, it is possible to rely on a batch issuance approach where each SD-JWT VC contains unique salts. Each attestation in a batch should also contain a unique public key that the user needs for the holder binding JWT. Clause 4.3.4.2 describes the possibility to use Hierarchical Deterministic Key derivation functions where the SD-JWT VC issuer can issue a batch of SD-JWT VCs, each with a unique and unlinkable public key value derived from a single user controlled public key.

- Another option to achieve unlinkability afforded by HAIP is for the user to request specific claims they need to present to a verifier and for the issuer to issue only these claims in the attestation; an approach that fits particularly well with the logic of short lived attestations.

- The SD-JWT VC issuer corresponds to a QTSP and/or a PIDP.

- The SD-JWT VC verifier corresponds to an eIDAS2 relying party (that will validate the SD-JWT as a (Q)EAA/PID).

- The eIDAS2 relying party should use the eIDAS2 EU TL to retrieve the QTSP/PIDP trust anchor and verify with the corresponding x5c header parameter of the SD-JWT VC.

- The eIDAS2 relying party should validate the attestation (submitted by the EUDI Wallet) according to the principles described in annex E; the issuer's signature should be validated by using the QTSP/PIDP trust anchor.

- The SD-JWT VCs in the EUDI Wallet should all use unique salts as described in annex E to cater for verifier unlinkability when validated by the relying party.

NOTE 1: Hence, the QTSP/PIDP would need to issue batchwise SD-JWT VCs in order to cater for multi-show verifier unlinkability. Batch issuance will require an operational procedure of issuing multiple SD-JWT VCs to each device on a regular basis, which may result in an additional operational cost for the QTSP/PIDP. Clause 4.3.4.2 describes an approach where the issuer can derive multiple unique user controlled public keys on the basis of a single user controlled public key.

NOTE 2: SD-JWT does not satisfy the requirements of full unlinkability.

- The SD-JWT VC is signed by the QTSP/PIDP with a JOSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.237] and for QSC for future use [i.149].

- The SD-JWT VC may be signed with an ETSI JAdES signature if supported by the relying party. Thus, the JAdES signature format may contain additional information about revocation information, CA-chains and time-stamps.

These observations and recommendations should be considered with respect to selective disclosure for the ETSI work items ETSI TS 119 462 [i.95], ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97], where also a mapping algorithm for the PID could be proposed.

7.4 Feasibility of BBS+ and BBS# applied to eIDAS2

7.4.1 General

The present clause provides an analysis of the feasibility of BBS+ and BBS# applied to eIDAS2. The BBS+ and BBS# schemes are of interest since they cater for issuer and verifier unlinkability, which could support privacy for a user's EUDI Wallet that shares selectively disclosed attributes. The BBS# scheme is of interest since it both leverages a SOG-IS sanctioned protocol (ECDSA or ECSDSA) for holder binding and caters for issuer and verifier unlinkability, which could support privacy for a user's EUDI Wallet that shares selectively disclosed attributes.

The following aspects are in scope of the analysis:

- The standardization status of BBS+ and BBS#, and if the schemes can be considered for the eIDAS2 regulation [i.103].
- Whether or not a standardized version of BBS+ and BBS# can be applied to the W3C Verifiable Credentials Data Model (VCDM) and ISO mobile driving license (ISO mDL).
- Post-quantum aspects of BBS+ and BBS#.
- Conclusions of how BBS+ and BBS# may be applied to QTSPs/PIDPs and EUDI Wallets operating under eIDAS2.

7.4.2 Standardization of BBS+ and BBS#

7.4.2.1 Standardization of BBS+

In order for BBS+ to be considered for the EUDI Wallet, it would have to be standardized by CEN, ETSI or ISO as declared in the EU regulation 1025/2012 [i.105].

As described in clause 4.4.6.1, a set of anonymous digital signatures schemes are specified in the ISO/IEC 20008 series [i.184]. More specifically, ISO/IEC 20008-2 [i.184] mechanism 3 specifies the cryptographic primitives of a qSDH scheme, which corresponds to BBS04 with single messages [i.27]. BBS04 with single messages is however not practically sufficient for most attestation formats, including the W3C Verifiable Credentials Data Model and SD-JWT VC, which require BBS+ with multi messages.

BBS+, which supports multi messages, is however not yet fully standardized. IRTF CFRG is currently in the process of specifying BBS+ in the following IRTF CFRG BBS draft standards: The BBS Signature Scheme [i.177], Blind BBS Signatures [i.175], BBS per Verifier Linkability [i.174]. In parallel, DIF is drafting a specification for blind signatures extension of BBS+ [i.80]. But even when the IETF and DIF standards are finalized they will not have the status such that they can be referenced by the eIDAS2 regulation [i.103].

In order to bridge this gap, ISO/IEC has initiated the standardization work on ISO/IEC 24843 [i.185] on privacy-preserving attribute-based credentials. One objective of ISO/IEC 24843 [i.185] is to formally standardize the multi-message signature scheme version of ISO/IEC 20008-2 [i.184], i.e. BBS+.

ISO/IEC are also working on the common draft ISO/IEC CD 27565 [i.191]. More specifically, Annex C of ISO/IEC CD 27565 [i.191] includes an example of selective disclosure by using BBS+, with a reference to the IRTF CFRG BBS draft specification.

Hence, the ISO/IEC 24843 [i.185] future standard, possibly in conjunction with ISO/IEC CD 27565 [i.191], has the potential to result in an ISO standardized version of BBS+ as well as other multi-message signature schemes. If these ISO standards on BBS+ will materialize, they may be referred by the eIDAS2 regulation [i.103] and its implementing acts. When such standards become available, the various attestation formats can also detail how BBS+ can be used as a proof mechanism.

7.4.2.2 Standardization of BBS#

BBS# is currently being standardized by AFNOR (the French Standardization Association). Also note that a new standard on Attribute-Based Credentials has been launched by ISO/IEC SC 27 (ISO/IEC AWI 24843 - Information security - Attribute-Based Credentials). Orange and Austrian Institute of Technology (AIT) will be the editor of this new project which might include the BBS/BBS# family of protocols.

7.4.3 Feasibility of using BBS+ or BBS# with W3C VCDM and mdoc

7.4.3.1 BBS+ applied to W3C VCDM

The analysis in clause 5.4.2.2 concludes that if ISO/IEC 24843 [i.185] and/or ISO/IEC CD 27565 [i.191] will standardize BBS+ according to IRTF CFRG BBS, then W3C BBS Cryptosuite v2023 [i.267] can be enhanced to reference such an ISO standard. In such a scenario, the W3C Verifiable Credential Data Integrity 1.0 specification [i.263] would refer to an ISO compliant version of W3C BBS Cryptosuite v2023. That would in turn mean that the W3C Verifiable Credentials Data Model v2.0, in conjunction with W3C Verifiable Credential Data Integrity 1.0, would be underpinned with an ISO standardized version BBS+.

It should however be observed that the ARF [i.71] requires the JSON PID to be compliant with the W3C Verifiable Credentials Data Model v1.1 with JWT encoding. Since an ISO standardized version of BBS+ would require W3C Verifiable Credentials Data Model v2.0 [i.265] with JSON-LD encoding, it will not be compatible with the ARF.

NOTE: It is not entirely clear what the ARF text requires in terms of W3C VCDM compliance. Section 6.2.2, Table 3 in the ARF text [i.71] requires that the *presentation* of an attestation is compliant with W3C VCDM 1.1, which means that the presentation includes verifiable statements about subject-predicate-value triplets that can be modelled as a graph. Section 7.5.3 of [i.71] requires that the *issuance* is compliant with the W3C VCDM 1.1. However, section 7.5.3 of [i.71] also requires that attestations are JWT based (optional support only for JSON-LD) and secured using SD-JWT. It is not clear how this compliance is to be achieved, i.e. whether enveloping and/or mapping is intended, and how enveloping would work with selective disclosure. The present document recommends using SD-JWT VC and relying on a mapping approach to ensure VCDM 1.1 compliance. If SD-JWT VCs are used, it is not clear how BBS+ can secure such attestations.

Hence, in order to support an ISO standardized version of BBS+, it is recommended to update the ARF to allow for W3C Verifiable Credentials Data Model v2.0 or preferably specify such format in the forthcoming ETSI TS 119 472-1 [i.97] standard on (Q)EAAs profiles.

Note that for a conversion between JSON or JSON-LD based document and multi-message signature schemes (such as BBS), choices have to be made that have an impact on the complexity of the overall signature scheme and possibly also Zero-Knowledge features.

The W3C Data Integrity BBS Cryptosuites v1.0 [i.255] has made such choices for the data transformation that currently optimize for the 3 variants of the IETF BBS drafts. Following the transformation of W3C DI BBS, the individual messages in the BBS signature scheme hold the full RDF canonicalized messages (see clause E.1.1 for examples on n-quads) such as:

```
[
…
  "_:b3 <https://windsurf.grotto-networking.com/selective#boards> _:b2 .\n",
  "_:b3 <https://windsurf.grotto-networking.com/selective#sailNumber> \"Earth101\" .\n"
…
]
```

This choice encodes the semantics (n-quads) together with the values, which works well with the current set of features of the IETF BBS drafts, but would make more complex proofs over values like range proofs and equality proofs significantly more difficult to construct and implement. Those trade-offs should be carefully taken into consideration when choosing data and container formats for BBS+ based credentials. Ecosystems should understand the more advanced features they want to implement before making those choices.

There are other options for container formats for BBS+ like Json Web Proofs (see clause 5.5.1 for more details on JWP).

7.4.3.2 BBS# applied to mdoc

BBS# can be made compatible with the ISO mDL device retrieval flow, for which selective disclosure is based on salted attribute digests. The use of BBS# on mdoc requires slight modifications to the BBS# issuance and selective disclosure protocols described in clause 4.4.3.

A summary of how BBS# can be applied toMSO is described below.

The issuer creates a MAC_{BBS} authentication tag σ on the user's mdoc authentication key *pk* and on the *L* digests $\{H_i\}_{i=1}^{L}$. The user's Mobile Security Object (MSO), in the terminology of ISO/IEC 18013-5 [i.181], consists of its public key *pk*, the digests $\{H_i\}_{i=1}^{L}$ and the MAC_{BBS} authentication tag σ on these data: $MSO = (pk, \{H_i\}_{i=1}^{L}, \sigma)$.

During the ISO mDL device retrieval flow, which involves selective disclosure, the user will create a signature σ_{HB} on the set of data referred to as "DeviceAuthenticationBytes" in the ISO/IEC 18013-5 [i.181] standard.

The signature σ_{HB} is a proof that the MSO originates from the user, which is holding the underlying MAC_{BBS} authentication tag σ, on the attributes disclosed to the relying party.

A complete description of how BBS# can be applied to the ISO mDL device retrieval flow is available in Annex G.

7.4.3.3 BBS# applied to W3C VCDM

BBS# is considered to be compatible with W3C Verifiable Credentials Data Model (VCDM) v2.0, given the following requisites:

- W3C VCDM v2.0 is compatible with BBS/BBS+ as declared in clause 7.4.3.1, and the credentials format can be preserved for BBS#.
- W3C VCDM v2.0 leverages Data Integrity BBS Cryptosuite for proofs, which can be extended to BBS#.
- Additional BBS# specific attributes, such as non-revocation proofs, can be defined as extensions to W3C VCDM.

7.4.4 Post-quantum considerations for BBS+ and BBS#

As discussed in clause 4.4.2.6, and as further elaborated on in clause 9, BBS+ multi-message signatures and disclosures that are generated in a pre-quantum world will remain confidential in a post-quantum world. As regards to BBS#, and as discussed in clause 4.4.3.4, the (Gap) *q*-SDH assumption is not quantum-safe, so an attacker in a post-quantum world will be able to forge BBS# credentials. Put differently, a computationally unbounded attacker will not be able to reveal neither undisclosed BBS+/BBS# messages nor the hidden signature value.

In a post-quantum world, however, neither BBS+ nor BBS# can maintain data integrity and authenticity. An attacker with a quantum computer can reveal the signer's private key from the public key and forge new signatures and proofs. Clause 9 discusses the prerequisites of this attack, its potential impact, and how to protect against it in greater detail.

7.4.5 Conclusions of using BBS+ and BBS# applied to eIDAS2

7.4.5.1 Conclusions of applying BBS+ to eIDAS2

An analysis of the BBS+ scheme applied to an eIDAS2 context results in the following observations and recommendations:

- The BBS+ algorithm would need to be standardized according to ISO/IEC 24843 [i.185] in order to comply with the EU regulation 1025/2012 [i.105] on standardization.
- A standardized profile of W3C BBS Cryptosuite v2023 would need to reference the ISO standardized version of BBS+. It is recommended that ETSI TC ESI standardize such a profile.
- A standardized (Q)EAA/PID profile of W3C Verifiable Credentials Data Model (VCDM) v2.0 in conjunction with W3C Verifiable Credential Data Integrity (VCDI) 1.0 would need to be specified, and reference the standardized W3C BBS Cryptosuite v2023. It is recommended that ETSI TC ESI standardizes profiles if attestation formats are to be W3C VCDM compliant and secured using BBS+.

- The issuing QTSPs/PIDPs would need to implement such ETSI standards in order to issue (Q)EAAs/PIDs compliant to the ARF and signed with the BBS+ algorithm.
- The BBS+ signature verifier corresponds to an eIDAS2 relying party (that will validate the BBS+ multi message signatures generated by the (Q)EAA/PID).
- The eIDAS2 relying party should use the eIDAS2 EU TL to retrieve the QTSP/PIDP trust anchor.
- The eIDAS2 relying party should validate the BBS+ multi message signature (finalized by the EUDI Wallet) according to the principles described in the IRTF CFRG BBS specification (or the future ISO standard on BBS+); the issuer's signature should be validated by using the QTSP/PIDP trust anchor.

NOTE: The BBS+ algorithm would cater for full unlinkability.

- The EUDI Wallets need to support the BBS+ algorithm in cryptographic keys management systems as specified in clause 6.5.3 of the ARF [i.71]. As described in clause 7.6, such cryptographic keys management systems with support for BBS+ could preferably be remote HSMs (with BBS+ support) or SIM-cards with support for BBS_MAC/BBS+ (see clause 6.6.4).
- A long term (Q)EAA/PID based on BBS+ should be used in a pre-quantum world only. The QTSP/PIDP should plan for migrating to quantum-safe cryptograhic algorithms in a post-quantum world.

These observations and recommendations should be considered with respect to selective disclosure for ETSI TS 119 462 [i.95], ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97].

7.4.5.2 Conclusions of applying BBS# to eIDAS2

An analysis of the BBS# scheme applied to an eIDAS2 context results in the following observations and recommendations:

- The BBS# algorithm would need to be standardized in order to comply with the EU regulation 1025/2012 [i.105] on standardization.
- BBS# could be made compatible with the mdoc and SD-JWT formats, and if standardized, this would bring full unlinkability to the associated protocols.
- The issuing QTSPs/PIDPs would need to implement such ETSI standards in order to issue (Q)EAAs/PIDs compliant to the ARF and signed with the BBS# algorithm.
- The BBS# signature verifier corresponds to an eIDAS2 relying party (that will validate the BBS# multi message signatures generated by the (Q)EAA/PID).
- The eIDAS2 relying party should use the eIDAS2 EU TL to retrieve the QTSP/PIDP trust anchor.
- BBS# is compatible with the existing security infrastructure (secure elements and HSMs) that could be used for WSCDs. BBS# optimizes the performance and security when deployed in combination with HSM based WSCD.
- BBS# can either use pairing friendly curves in which case it does not require additional interactions with the issuer for the proof or pairing free curves. Hence, BBS# leverages a SOG-IS approved holder binding cryptographic protocol (ECDSA), yet preserving all privacy properties of BBS/BBS+.

The BBS# signature scheme would meet the requirements of the EUDI Wallet cryptographic keys management systems as specified in clause 6.5.3 of the ARF [i.71]. Any ECDSA capable certified WSCD would suffice, combined with the proper software support in the EUDI Wallet.

BBS# is not forgery safe in a post-quantum world but supports everlasting privacy which offers a reasonably long window of opportunity. A long term (Q)EAA/PID based on BBS# should be used in a pre-quantum world only. The QTSP/PIDP should plan for migrating to quantum-safe cryptographic algorithms in a post-quantum world.

At this stage, BBS# matches the ZKP scheme security requirements that is at the same time efficient and non-circuit based. Furthermore, BBS# optimizes the deployment with HSM based WSCDs, which seems to be the preferred solutions for several EUDI Wallets across the EU Member States.

BBS# still lacks standardisation but would be easily amenable to the mdoc format and is already used commercially by dock.io. An overview of the implementation of a trust model based on BBS# is described in [i.217].

7.5 Feasibility of programmable ZKPs applied to eIDAS2 (Q)EAAs

7.5.1 Background and existing solutions

As discussed in clause 6.5, there exist two implementations of ZKP schemes (zk-SNARKs) that are utilized for sharing selectively disclosed attributes and revocation status information.

The Cinderella project (see clause 6.5.2) has integrated support for zk-SNARKs in TLS software libraries, which allows for Cinderella pseudo-certificates with selected attributes and optional OCSP stapled responses to be communicated over the TLS handshake. More specifically, the Belgian, Estonian, and Spanish national eID smartcards with X.509 QCs have been successfully tested with the Cinderella TLS implementation. Hence, the existing eIDAS PKI infrastructure without modifications is re-used. Configuring or refreshing the Cinderella pseudo-certificates can take up to nine minutes, and should therefore be performed offline, but the online verification takes only 10 ms.

The zk-creds project (see clause 6.5.3) has implemented anonymous credentials by using ZKP for credentials derived from ICAO compliant eMRTDs (passports). The ZKP is essentially generated based on the eMRTD's Data Group 1, which contains the textual information available on the eMRTD's data page and the Machine Readable Zone: name, issuing state, date of birth, and passport expiry.

Hence, the Cinderella and zk-creds projects have demonstrated with their prototypes that ZKP schemes can be used with existing digital identity infrastructures to share selected attributes of X.509 certificates and ICAO eMRTDs.

7.5.2 Extensions to EUDI Wallets, relying parties and protocols

In order for an EUDI Wallet to use zk-SNARKs with existing credentials (such as X.509 certificates), a circuit compiler (such as the Geppetto compiler) is needed to integrate the zk-SNARK client circuits into the EUDI Wallet. Furthermore, the authentication protocol (such as TLS) needs to be enhanced in order to generate pseudo-certificates that can be validated by the relying party (TLS server). The EUDI Wallet would need to download the trusted roots based on the EU Trusted List (TL) in order to validate the status of the X.509 certificate and the optional OCSP-response.

The relying party needs to be extended in order to validate the pseudo-certificates and the proof of the OCSP response. The Cinderella project has demonstrated that this is feasible with TLS and X.509 certificates. In a similar fashion, the zk-creds project has demonstrated that it is possible to share selected attributes of an ICAO eMRTD by using ZKP schemes.

Since the ARF specifies mdoc and mandates W3C VCDM compliance for the PID formats, it would be of interest to investigate if the EUDI Wallet could be extended with zk-SNARK client circuits policy templates that can generate selected attributes of pseudo-versions of mdocs and/or W3C VCDM compliant VCs (e.g. SD-JWT VC with mapping) and optional stapled revocation information.

Furthermore, the ARF [i.71] specifies OID4VP [i.214] as the presentation protocol for the EUDI Wallet. Hence, it would be of interest to specify a profile of OID4VP with a DIF Presentation Definition (OID4VP request) [i.81] and DIF Presentation Submission (OID4VP response) [i.81] that could use programmable ZKP schemes to present selected attributes of pseudo-versions of mdocs and/or W3C VCDM compliant VCs and optional stapled revocation information.

Since zk-SNARKs can cater for full unlinkability, this feature would be inherited for the EUDI Wallets as well. Also, it is recommended to select zk-SNARKs that are plausible quantum computing safe (see Table A.4).

7.5.3 Conclusions of programmable ZKPs applied to eIDAS2 (Q)EAAs

An analysis of the ZKP scheme applied to (Q)EAAs, QCs or PIDs in an eIDAS2 context results in the following observations and recommendations:

- The EUDI Wallets would need to be extended with programmable ZKP circuits and policy templates in order to generate pseudo-credentials with selected attributes of (Q)EAAs, QCs or PIDs and optional stapled revocation information. The EUDI Wallet should use the eIDAS2 EU TL to retrieve the QTSP/PIDP trust anchor. The zk-SNARK trusted roots would need to be configured as well.

- The issuing QTSPs/PIDPs can re-use the existing eIDAS framework and related ETSI standards in order to issue QCs. The eIDAS2 framework and planned ETSI standards for issuance of (Q)EAAs/PIDs can also be used without modifications. The QTSP/PIDP trust anchor can be published at an eIDAS2 EU TL.

- The verifier corresponds to an eIDAS2 relying party (that will validate zk-SNARK proofs and pseudo-credentials generated out of the (Q)EAA/QC/PID). The eIDAS2 relying parties would need to be extended with zk-SNARK circuits and policy templates in order to validate the pseudo-credentials and stapled revocation information.

NOTE: The zk-SNARK scheme would cater for full unlinkability.

- The zk-SNARKs that are plausible quantum computing safe (see Table A.4) should be used.

- OID4VP would need to be extended in order for an EUDI Wallet to present the pseudo-credentials with selected attributes and stapled revocation information to a relying party.

These observations and recommendations should be considered with respect to selective disclosure for ETSI TS 119 462 [i.95], ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97]. Implementations of the programmable ZKP schemes in the EUDI Wallets and relying parties may be implemented and evaluated as part of the eIDAS2 large scale pilots.

7.6 Secure storage of PID/(Q)EAA keys in EUDI Wallet

7.6.1 General

The mdoc authentication key and SD-JWT holder binding keys should be protected in the device's Trusted Execution Environment (TEE) or a Secure Element (SE). The user should be able to access the mdoc authentication key and SD-JWT holder binding key by authentication with a PIN-code or the use of biometrics. There exist implementations and large scale deployments of mdoc for Apple iOS® and Google Android®, which utilize Secure Elements that protect the mdoc authentication key. Several mdoc and SD-JWT data elements are PII and should therefore be stored securely. Encryption at rest of the SD-JWT is recommended, and if possible the SE/TEE should be used to perform the encryption, with keys protected by the SE/TEE, or else the mdoc and SD-JWT should be stored in the SE/TEE. Alternatively, the ISO MSO or SD-JWT keys could be protected in a remote HSM or external device, which are the other cryptographic keys management systems as specified in clause 6.5.3 of the ARF [i.71]. The ARF [i.71], clause 6.5.3 and Table 5 also specify how to store and access the PID/(Q)EAA cryptographic keys in a device used by the EUDI Wallet.

Since BBS+ is not (yet) selected to be used for any PID format, there is no specification in the ARF about storage or access to BBS+ credentials and keys. However, the research paper "Improved Algebraic MACs and Practical Keyed-Verification Anonymous Credentials" [i.15] describes how to efficiently implement a BBS_MAC/BBS+ variant on a SIM-card, which can be considered as an external cryptographic device that can be accessed by a mobile device. It is also plausible that HSMs in a near future will be equipped with the BBS+ algorithm, which would then cater for the EUDI Wallets to access BBS+ credentials and keys in a remote HSM. It is however unlikely that BBS+ will be implemented in embedded Secure Elements in the near future.

BBS# can leverage any ECDSA (or ECSDSA) signature capable WSCD making it suitable to the same widely available security infrastructures deployed today as any other ECDSA based solution, be it a secure element in or next to a phone or an HSM deployed in a secure environment. As noted above, BBS# is particularly well suited to HSM deployments. In addition to the WSCD though, randomization keys have to be managed inside the EUDI Wallet's WSCA or WSCI. Some keys are ephemeral and do not need to be stored in EUDI Wallet post transactions (typically for presentations) while others need to be stored for a longer time (typically those used to randomize the keys provided to issuers) as they have to be retrieved later on to perform proofs. This means that a key generation mechanism has to be managed in the EUDI Wallet's WSCA or WSCI, typically leveraging smartphone capabilities as TEE or secure elements when available (irrespective of the WSCD location). As explained above, this part of the key is necessary for security but it is not critical, contrary to the part stored in the WSCD.

From a regulatory perspective, the eIDAS2 [i.103] article 5c specifies the legal requirements on an EUDI Wallet certification, which will be defined in a Commission Implementing Regulation (CIR). This CIR will in turn refer to ENISA's EU Cybersecurity Certification (EUCC) scheme, which may regulate the certification requirements on protection of the PID/(Q)EAA as mdoc and SD-JWT.

Furthermore, CEN TC/224 WG17 may specify Common Criteria Protection Profiles (CC PP) on how to protect the PID/(Q)EAA and associate cryptographic keys related to the ENISA EU-CC; such EUDI Wallet CC PP may be based on TC/224 WG17 [i.54]. Also, TC/224 WG20 [i.55] are specifying how to onboard the PID to an EUDI Wallet, which involves the associated cryptographic key protection as well.

Other certification standards that may underpin the ENISA EU-CC scheme are Global Platform TEE Protection Profile [i.118] and Eurosmart PP-0117 Protection Profile for Secure Sub-System in System-on-Chip (3S in SoC) [i.106].

Additional recommendations on how to store and protect credentials and the associated cryptographic keys in a digital wallet are available in the DIF Wallet Security [i.82], ISO/IEC CD 23220-6 [i.188] and W3C Universal Wallet [i.262] specifications.

NOTE: Complete descriptions about storage of PID/(Q)EAA, protection of cryptographic keys and EUDI Wallet certifications go beyond the scope of the present document, but an overview is provided in the present clause since the cryptographic keys are of relevance to selective disclosure of PID/(Q)EAA in the formats of mdoc and SD-JWT.

7.6.2 Key splitting technique (relevant for BBS#)

The theory behind the BBS# key splitting technique is described in clause 5.2 of "Making BBS Anonymous Credentials eIDAS 2.0 Compliant" [i.78]. The present clause analyzes how BBS# key splitting can be applied to the EUDI Wallet.

The splitting technique of BBS# is similar to SECDSA except that the "blinded"/"randomized" public key, called pk_{Blind}, is included for security reasons in the message to be signed by the WSCD. Without pk_{Blind}, the BBS# splitting technique would be vulnerable to a "simple related-key attack" (which unfortunately applies to SECDSA and therefore means that SECDSA is insecure).

The splitting technique pf BBS# basically distributes the keys and the associated computing between on one side the WSCD and on the other side the EUDI Wallet's WSCA/WSCI. The WSCD part is the Wallet Secure Cryptographic Device, which has to be AVA_VAN.5 certified. The WSCD hence protects the private key (or its cryptographic primitives). The WSCA/WSCI parts are responsible for randomization, which ensures unlinkability both at issuance - each VC can have its own unique public key - and at presentation. The WSCA/WSCI also create an additional security layer for "centralized" type WSCDs (like HSMs) by descaling the effects of a potential HSM takeover by forcing the attacker to also need to retrieve each of the random keys of each of the VCs from each of the EUDI Wallets' WSCA/WSCI. While these keys might not be as well protected as the WSCD keys, the bare fact that they reside on other platforms breaks the global reach of a successful HSM takeover.

Finally, whenever a "combined" proof needs to be performed leveraging multiple VCs in a single VP, as all VPs use the same unique WSCD key, the calculation can be performed only once and the rest of the computations are then diversified locally on the wallet. This avoids the complex issue of having to authorize multiple transactions on a single WSCD with a single unlocking action from the end users and also reduces the load on the HSM when performing combined proofs, thus increasing its scalability.

7.7 The proportionality of privacy goals

7.7.1 General

The present clause examines the complexity costs and practical implications of key privacy goals, structured around the two principal events where privacy preservation is most relevant: issuance and presentation. It focuses on core privacy objectives (issuer and verifier unlinkability, selective disclosure, pseudonymity, and unlinkable revocation) and discusses whether these are proportionate given the practical feasibility of the technical approaches used to achieve them. Rather than providing exhaustive coverage, the aim is to support a systematic evaluation of the trade-offs between privacy and practical feasibility across a set of representative scenarios.

Different Levels of Assurance (LoA) significantly influence the cost of achieving privacy goals. For example, many PID issuers operating at LoA High face legal and operational constraints that limit the use of certain privacy-preserving technologies. Relying on salted attribute digests for selective disclosure can make it prohibitively expensive to meet legal requirements for full unlinkability through technical means alone. In contrast, private actors issuing at LoA Substantial encounter significantly lower complexity costs. Efforts to reduce the cost of achieving key privacy goals in LoA High issuance contexts (such as standardization, broader hardware vendor support, and changes to operational and legal frameworks) are underway.

The results presented below are subject to change as cost-reduction efforts advance and should be interpreted within the appropriate LoA context.

7.7.2 Issuance

Issuance requires identity verification and Proof of Possession (PoP) of a hardware-protected key (the hardware protection is especially burdensome at LoA High). The PoP inherently links the attestation to the user's identity as the issuer knows who receives which attestation and when.

Issuer unlinkability (assuming issuer-verifier collusion) requires that no value in the attestation reduces uncertainty about the user's identity beyond the disclosed identity attributes. Any value in the attestation - timestamps, salts, the issuer's signature - can be linked to the previously identified user. While many of these values can be blinded, achieving issuer unlinkability at LoA High incurs prohibitive complexity costs if limited to conventional cryptography due to the requirement of cut-and-chose based issuance (costs are high even if more recent schemes and solutions are considered, e.g. BBS+ or relying on ZKP layering on top of the core wallet formats, as this requires certification, hardware support, and/or standardization efforts).

Consequently, preventing issuer linkability - especially under collusion - is presently infeasible at LoA High (but available for private actors operating at LoA Substantial). Consequently, most deployments of PID issuance have to accept this limitation, rely on regulatory mechanisms to prevent issuer collusion, and focus privacy protections on the presentation phase instead.

NOTE: Two main approaches are being explored to address the privacy limitations of core wallet formats. One approach is to layer ZKPs on top of an attestation. The other is to rely on private actors who can issue identity attestations at LoA Substantial, based on an underlying PID. Both approaches have distinct strengths and limitations, and efforts are ongoing to definitively assess their relative advantages or suitability within the EUDIW context.

With the above in mind, and before elaborating on why issuer unlinkability is presently practically infeasible at LoA High, there are two primary issuance models to consider:

1) Request-based issuance, in which a user explicitly requests a credential from an issuer following successful authentication.

2) Scheduled issuance, where credentials are issued proactively - e.g. after enrollment or at regular intervals - without a direct request for each individual credential.

In request-based issuance, the attestation is tailored to a specific verifier's request. This model increases some privacy risks but simplifies others:

- Revealed beyond attributes: Request timing and attribute set reduce uncertainty about the target service and may uniquely identify it via auxiliary data (e.g. service registries).

- Issuer unlinkability: Requires removing all correlation handles (salts, timestamps, signature, public PoP key). No practical solution achieves this presently at acceptable cost at LoA High, but ongoing efforts show promise. Solutions exist at LoA Substantial.
- Verifier unlinkability: Achievable with short-lived, single-use attestations (assuming no issuer-verifier collusion) at LoA High. Achievable at LoA Substantial even assuming issuer-verifier collusion.
- Selective disclosure: Not needed. The attestation is scoped to the verifier's request, unless such a scoping introduces a privacy risk necessitating selective disclosure.
- Pseudonyms: Easy to implement but of limited value if colluding issuers are assumed when the users authenticate with identifying information.
- Validity status mechanism: Privacy can be preserved by using explicit validity periods, which reduce linkability compared to mechanisms such as status lists or online revocation services. However, there is currently no standardized or widely deployed method to maintain privacy when explicit revocation is required. That said, several research initiatives show promise and are approaching trial readiness.
- PoA mechanism: The issuer will always include the user's PoP key in the attestation and the PoA links every key to a user with one and the same verified identity. While blinding the PoP key is technically feasible (but requires great care), it only achieves issuer unlinkability if all other correlation handles are blinded as well.

NOTE 1: The above purposefully ignores non-technical measures such as audits or organizational mechanisms where each eliminated correlation handle has a positive impact.

NOTE 2: While possible to use long lived attestations in a request-based model, the benefits are unclear.

These considerations highlight a core issue: issuer unlinkability cannot be achieved by blinding a single value in isolation. While blinding individual values is often feasible, the issuer observes the full authenticated context, and any remaining correlation handle can enable linkage. This risk is further exacerbated by the issuer's ability to structure value combinations to increase linkability - for example, using unique combinations of `nbf`, `iat`, and `exp`. Consequently, issuer unlinkability requires blinding all handles and standardizing certain attributes (like `nbf`, `iat`, and `exp`), which is likely prohibitively costly at LoA High using only conventional cryptography. Thus, legal requirements for issuer unlinkability cannot be met without additional measures, such as ZKP layering or issuing identity attestations from a PID using more suitable signature schemes at LoA Substantial.

In scheduled issuance, the attestation includes all user attributes. This increases some privacy risks while reducing others:

- Revealed beyond attributes: The issuer cannot infer the target service from the timing or the attribute set.
- Issuer unlinkability: Same as request-based; eliminating all correlation handles is impractical at LoA High but otherwise achievable.
- Verifier unlinkability: Achievable with short-lived, single-use attestations (assuming no issuer-verifier collusion), possibly using batch issuance at LoA High. Long-lived, multi-show use requires cryptographic protections (e.g. ZKPs) to avoid reuse of correlatable values, but are only effective if all correlation handles are eliminated.
- Selective disclosure: Required. Salted attribute hashes offer a practical compromise for verifier unlinkability. Advanced schemes can support issuer unlinkability if all correlation handles are eliminated.
- Pseudonyms: Same as in request-based issuance; limited utility under identifying authentication.
- Validity status mechanism: Short-lived attestations can use embedded validity periods which limits linkability. Long-lived ones require a practical privacy-preserving revocation that can scale, which remains an open challenge (trials of promising candidates are planned).
- PoA mechanism: Same as request-based; unlinkability depends on blinding all correlation handles.

In summary, issuer linkability remains a fundamental challenge due to the reliance on an authenticated context during issuance. The issuer can use any value in the attestation - keys, timestamps, attributes - either in isolation or jointly to reduce uncertainty about the identity subject. Full unlinkability would require eliminating all correlation handles, which is practically impossible using conventional cryptography at LoA High.

When discussing these correlation handles, important considerations include, but are not limited to:

- Issued set vs. eligible set. The issuer's identification scope is limited to users who have received attestations, reducing uncertainty from the broader eligible population (e.g. all individuals eligible for a driver's license) to the actual issuance set. This allows the issuer to perform re-identification within a smaller, more tractable group.
- Crafted attribute combinations: The issuer can construct unique identifiers by combining multiple attributes with high variability. For example, using a 60-second resolution for temporal fields such as `nbf`, `exp`, and `iat` yields $60^3 = 216\,000$ possible combinations. Additional claims like `aud`, `scope` etc., (or even custom attributes) further increase identifiability. Mitigations may be costly.
- Structured randomness. Fields intended to appear random can be structured to encode user-identifying information. The issuer can create specific bit patterns to represent users in values that are indistinguishable from random.

To conclude: while request-based issuance offers privacy advantages by avoiding the use of unique salts or commitments and eliminating the need for revocation checks (both sources of linkability) the cost of using technical mechanisms to eliminate all correlation handles is disproportionate to the privacy benefits offered at LoA High, but achievable at LoA Substantial.

7.7.3 Presentation

Presentation differs from issuance in two main ways that impact privacy. First, user authentication does not necessarily reveal an identity to the verifier (in contrast to issuance where the issuer identifies the user prior to issuance). Second, presentation occurs in the context of service access, which can expose behavioural patterns and enable profiling. It is therefore essential that verifier unlinkability is achieved.

If issuers and verifiers collude, privacy is as difficult to preserve as during issuance at LoA High, due to the infeasibility of issuer unlinkability discussed in the previous clause. However, under a more realistic trust model - where issuers do not collude with verifiers, but verifiers may collude with each other - several privacy goals become achievable. In this setting, techniques such as selective disclosure, zero-knowledge proofs, and short-lived attestations are particularly effective in limiting linkability and preserving user privacy.

At LoA Substantial, actors have several cost effective opportunities that work even under the more adversarial trust assumption of issuer-verifier collusion.

Several verifier-side steps during presentation carry privacy implications:

1) Wallet instance validation: If verifiers have to validate the wallet, this step has to be privacy-preserving. Short-lived attestations mitigate the issue if the verifier can rely on issuer-side validation. In contrast, long-lived attestations require wallet validation mechanisms that avoid introducing correlation handles. While such mechanisms exist, they increase complexity and may outweigh the benefits of long-term credentials.

2) Parsing the presentation: Disclosed attributes, metadata, and validity information can all enable correlation if any value - such as salts, PoP keys, or issuer signatures - is reused. Verifier unlinkability requires eliminating all correlation handles and minimizing auxiliary linkability. For example, index assignment in validity status lists should be randomized to avoid correlation based on sequential ordering. Similarly, static boolean values (e.g. age_over_18: true) require external validity context (e.g. `nbf`) to be meaningful; a major privacy risk when using long-lived attestations. The validity context can be blinded, but the techniques used for blinding the context could just as well be adopted to instead blind the age value (e.g. a Bulletproof range proof).

3) Proof of Association (PoA): When combining attributes from multiple attestations or linking PoP keys, the association proof has to be unlinkable and preferably third-party deniable. This can be achieved using, for example, interactive discrete log equivalence (DLEQ) ZKPs.

4) Attribute value verification: Privacy-preserving techniques like Bulletproofs or other range proofs can be used to verify attribute values. Issuers have to ensure that all blinded value commitments and cryptographic hash digests are unique to ensure verifier unlinkability.

5) Validity checks: Validity checks have to be privacy-preserving to avoid introducing stable identifiers (e.g. revocation list positions). The simplest approach is to use short-lived attestations with embedded expiration, avoiding online revocation checks entirely. For long lived attestations, several privacy preserving revocation solution proposals approach trial readiness.

7.7.4 Prioritizing privacy goals given the costs

Verifier unlinkability is achievable today at reasonable complexity both at LoA High and Substantial with conventional cryptography, making the presentation phase the primary point for enforcing privacy in a digital identity system. This shifts some burden to issuance, which has to ensure unique salts, signatures, and PoP keys.

Issuer unlinkability remains particularly challenging at LoA High, but approaches using ZKP and advanced signature schemes show promise. However, eliminating one or more values in isolation is ineffective as long as other potential correlation handles remain (both at the attribute level and through structured combinations).

At LoA High, the recommendation is, as shown in Table 2, to prioritize verifier unlinkability first, and pursue more advanced privacy goals as supporting technologies mature and the understanding of privacy risks from structured combinations improves.

Table 2: Privacy goals and their feasibility at LoA High

Privacy Goal	Problem Source(s)	Feasible today?	Recommended Approach
Issuer unlinkability	Authenticated context; any unique value identifies user	No	Mitigate through regulation or policy. Verifiers can leak malicious issuer behaviour.
Verifier unlinkability	Reuse of static values (PoP keys, salts, unique IDs, revocation data)	Yes, with short-lived attestations (revocation) remains challenging	Use short-lived, single-use attestations; eliminate reused values. ZK overlays (e.g. over ECDSA and range proofs).
Association unlinkability	Non-deniable binding between disjoint attestations	Yes	Use ephemeral or interactive ZKPs (e.g. DLEQ); avoid persistent proofs.
Minimal disclosure	Full disclosure of attributes, inflexible values	Yes	Use selective disclosure and range proofs with unique computational inputs (e.g. Pedersen commitments or hash digests). Signature schemes with inherent disclosure capabilities.
Prevent service inference	Request timing or highly specific attribute sets	Yes	Use scheduled issuance; avoid public service catalogues with request-based flows.

At LoA Substantial, where actors operate under fewer constraints, privacy goals are significantly more attainable using solutions such as ZKP layering or advanced signature schemes (e.g. BBS+ or BBS#). It remains unclear whether ZKP layering is feasible at LoA High, or when advanced signature schemes will be acceptable for issuing LoA High attestations.

The challenges associated with privacy preserving validity status checks are discussed next.

8 Privacy aspects of revocation and validity checks

8.1 Introduction to revocation and validity checks

Given that eIDAS2 article 5a.16(a) as well as recitals 14, 15, and 59 require that selective disclosures and unlinkability are done in ways that prevent data linkability, then the data unlinkability requirement has to be extended to validity status checks. Herein, the focus includes only options that fall under "state of the art" (solutions that have been deployed on a market) as stipulated in GDPR [i.102] articles 25, 26, and 32, and those approaches that are "experimental" (solutions where technical feasibility has been demonstrated but where market deployments are still lacking). In addition to this, eIDAS2 article 5a.16 should be considered, where it is stated:

"*The technical framework of the European Digital Identity Wallet shall: (a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorised by the user;*"

Hence, revocation services and validity status check services should avoid collecting revocation information about the EUDI Wallet and its (Q)EAAs.

Furthermore, a validity status check (e.g. due to revocation) can be conceptualized as a set (non-)membership proof, and alternatives that limit correlation handles and uncertainty reduction are discussed. For completeness, the text also mentions well known options that may not be suitable as a validity status check approach.

NOTE 1: Both (Q)EAAs or PIDs may be considered with respect to revocation and validity status checks; only the term (Q)EAA is used for readability throughout clause 8.

NOTE 2: (Q)EAAs or PIDs may contain unique identifiers or serial numbers; only the term identifier is used for readability throughout clause 8.

NOTE 3: Issuers can use explicit validity periods as an alternative to the techniques mentioned below.

8.2 Online certificate status protocol (OCSP)

The online certificate status protocol (OCSP) is an internet protocol specified in IETF RFC 6960 [i.160] that is designed to obtain and check the current validity status of a digital X.509 PKIX certificate.

However, OCSP was not designed with privacy in mind and therefore it lacks certain privacy aspects. The OCSP protocol submits the unique identifier of a (Q)EAA to an OCSP responder, which checks revocation status of the X.509 PKIX certificate against a revocation database and returns an OCSP response with status 'good', 'revoked', or 'unknown'. So, from a privacy perspective, OCSP risks revealing more information with the OCSP responder than the user intended.

However, OCSP could work for (Q)EAAs containing an identifier or serial number, specifically with respect to:

- OCSP Must-Staple. In an OCSP stapling scenario, the EUDI Wallet itself would query the OCSP responder at regular intervals in order to obtain a signed and time-stamped OCSP response for the user's (Q)EAA. Then the EUDI Wallet would need to append the OCSP response when presenting the (Q)EAA to the verifier. OCSP stapling is supported by TLS in the Certificate Status Request extension (see section 8 in IETF RFC 6066 [i.159]).

8.3 Revocation lists

A Revocation List (RL) is a mature and widely utilized validity status check mechanism. For detailed examples see IETF RFC 5280 [i.156] that specifies the Certificate Revocation List (CRL) profile for PKIX X.509 certificates and IETF RFC 6818 [i.161] that updates IETF RFC 5280 [i.156]. Commonly, a RL is a signed list of identifiers or serial numbers associated with the (Q)EAAs that have been revoked before they expired. Since the identifiers are unique and thus perfectly correlates with the associated (Q)EAAs, any solution that relies on a RL need to consider the following privacy aspects:

- Single-show attestations, whereby each (Q)EAA has a unique identifier or serial number. This concept is equivalent to atomic (Q)EAAs that are described in clause 4.2. Hence, the RL will contain different identifiers for the user's set of atomic (Q)EAAs.
- Range requests, which depends on the size of the RL. The privacy provided by a RL is proportionate to the size of the RL. In the extreme case with one revoked identifier in a RL, the RL provider will be able to identify what (Q)EAA the verifier or user needs to check. The larger the RL is, the more difficult it is for a RL provider to correlate the user's (Q)EAA with the requests to the RL provider.

Additionally, a RL needs to also consider the event where a batch of (Q)EAAs change status at once. In such a scenario, verifiers can collude and compare the (Q)EAA identifiers with the simultaneous validity status changes to learn more about which (Q)EAAs describe the same subject. Cryptographic techniques such as Private Set Intersection (PSI) or Private Information Retrieval (PIR) may prove helpful as solutions:

- Private Set Intersection [i.202] is a secure multiparty cryptographic technique that allows two parties holding sets to compare encrypted versions of these sets in order to compute the intersection. In this scenario, neither party reveals anything to the counterparty except for the elements in the intersection.
- Private Information Retrieval [i.26] is a protocol that allows a client to retrieve an element of a database without the owner of that database being able to determine which element was selected.

8.4 Validity status lists

A validity Status List (SL) is a bit vector that is issued and signed by an issuer (QTSP in eIDAS2 terms). The validity status of a (Q)EAA is represented using either a single bit or multiple bits in the SL bit vector. The (Q)EAA identifier is mapped to an index in the status list. The validity status check of the (Q)EAA is performed by checking the binary value of the bit(s) that is indexed in the status list bit vector. If the binary value of the indexed position in the status list is 1 (one), the (Q)EAA is revoked, else if it is 0 (zero) it is not revoked.

EXAMPLE: The (Q)EAA with the identifier 49361 is mapped to the status list index 136547. In the status list bit vector, the indexed position 136547 is a binary value of 0 (zero). Hence, the (Q)EAA is not revoked in this example.

The W3C Verifiable Credentials working group has specified "Bitstring Status List v1.0 - Privacy-preserving status information for Verifiable Credentials" [i.254] with details on how to issue status lists and check the validity status of Verifiable Credentials. IETF has specified "OAuth Status List" [i.153] that defines status list data structures for representing the status of JSON Web Tokens (JWTs) and CBOR Web Tokens (CWTs).

Status lists have the following features:

- The validity status list bit vector per se does not reveal any information about the (Q)EAA's identifier, which is a privacy preserving feature. (PKIX CRLs contain the serial numbers of the revoked PKIX X.509 certificates).
- The size of a status list can be relatively small. The size after compression depends on the final revocation rate and whether or not the index assignments are random. Uncompressed, a status list for 100 000 (Q)EAAs is roughly 12,5 kB in size. This is beneficial for performance and bandwidth reasons when a verifier downloads the status list. (PKIX CRLs contain more metadata about the revoked PKIX X.509 certificates and are therefore considerably larger).
- A verifier can retrieve the entire status list without revealing what index it will check, which is a privacy preserving feature. (An OCSP request contains the PKIX X.509 certificate serial number, which reveals what certificate a verifier needs to check).

As with RLs, the identifier is a unique correlation handle. Consequently, any solution that relies on a SL need to also consider the following privacy preserving aspects:

- Single-Show attestations, range requests, and/or PSI (cardinality), possibly ZKP, as described for RLs.
- Randomized index assignment. The index associated with each (Q)EAA is randomly assigned over the entire set of possible (Q)EAAs. Consequently, chunks of the status list cannot be derived based on e.g. issuance or expiration time.
- Hiding of still valid (Q)EAAs. Status list sizes that equal the number of issued (Q)EAAs allows an attacker to learn information about still valid (Q)EAAs.

As with RL, a SL does also consider events where a batch of (Q)EAAs change status at once. Private Set Intersection and Private Information Retrieval techniques are therefore recommended to be considered.

8.5 Cryptographic accumulators

A cryptographic accumulator allows the aggregation of many values into a fixed-length digest called the accumulator value. Furthermore, and in contrast to cryptographic hash functions, it is possible to verify whether an element is accumulated or not. Asymmetric accumulators rely on a so-called (non-)membership witness. Symmetric accumulators do not require a witness for membership testing. Negative accumulators support non-membership witnesses: positive ones support membership witnesses, and universal ones support both.

A Bloom filter is an append-only data structure that can be used for a set of (non-)membership tests without any witness. These tests allow for false positives but not for false negatives. Put differently, a Bloom filter test will either yield that the tested element is possibly in the set, or that it is definitely not in the set. Multiple Bloom filters can be chained so that the false positives are included in a second Bloom filter that tests for the opposite value (e.g. the first Bloom tests for revocation; the second is a non-revocation test). This process can be repeated indefinitely to create a Bloom filter cascade with a sufficiently low false-positive rate.

In contrast to RL and SL, a Bloom filter does directly reveal information about the set elements. Any validity status change is probabilistic, which means that colluding entities cannot know if the changes reflect a simultaneous validity status change (e.g. a revocation of a batch issued (Q)EAA) or a false positive. However, the probabilities depend on the Bloom filter and it has to be designed with care as colluding verifiers can use any Bloom filter based approach that has a sufficiently low false-positive rate to link together an attestation batch in the event of a validity status change.

Many other cryptographic accumulators exist beside Bloom filters. This text mentions Bloom filters specifically due to the focus on market deployed techniques.Yet, the alignment of Blook filters with general-purpose ZKPs to achieve unlinkability remains unexplored, other examples of market deployed solutions exist, e.g. the accumulator scheme used in Hyperledger AnonCreds [i.131] and by the IRMA [i.173] project, which is an implementation of the Idemix [i.136] attribute-based credential scheme. It is also worth mentioning more recent work that demonstrates how the witness updates can be done in a privacy friendly batch update, meaning that the witness update is the same for all users.

Camenisch and Lysyanskaya introduced the concept of dynamic accumulators in their paper "Dynamic accumulators and application to efficient revocation of anonymous credentials" [i.44] in 2002. A dynamic accumulator allows for dynamically adding or deleting a value, such that the cost of adding or deleting is independent of the number of accumulated values. The paper also provides a construction of a dynamic accumulator and an efficient zero-knowledge proof scheme, which can be proven secure under the strong RSA assumption. Such construction of dynamic accumulators enables efficient revocation of anonymous credentials and membership revocation for group signature and identity escrow schemes.

Furthermore, the first dynamic universal accumulator was introduced in 2009 in a paper by Au, Tsang, Susilo and Mu that describes how dynamic universal accumulators for DDH groups can be applied to attribute-based anonymous credential systems [i.13].

Moreover, Nguyen described accumulators from bilinear pairings and applications in a paper published in 2005 [i.204], which was extended in 2008 by Damgård and Triandopoulos in their paper "Supporting Non-membership Proofs with Bilinear-map Accumulators" [i.76]. Recently, in 2022, the research in this field was extended by Vitto and Biryukov in their paper "Dynamic Universal Accumulator with Batch Update over Bilinear Groups" [i.247].

Pairing free accumulators also exist that function with the same kind of scheme as BBS#. The BBS# scheme could be mutualized with a flow from the verifier to the issuer, which is described in option 1 in clause 4.4.3.3.3. Option 1 is recommended for performance reasons and because the holder can be offline. This setup assumes that the issuer is also the accumulator issuer (which should be the case in most if not all situations).

Hence, cryptographic accumulators, and dynamic accumulators and universal dynamic accumulators are worth considering for revocation schemes when privacy requirements are high. Recent work has focused specifically on how accumulators can be used for revocation of the core EUDI Wallet formats [i.112].

8.6 Using programmable ZKP schemes for revocation checks

As described in clause 6.5.1, it is possible to design anonymous credentials from programmable ZKPs (typically zk-SNARKs) and existing digital identities (such as X.509 certificates). Furthermore, the revocation and validity status can be performed at the digital wallet, whilst the validation results, selected attributes and predicates are shared with the verifier. Hence, any type of revocation verification protocol, even OCSP, can be implemented at the digital wallet, yet providing privacy for the user.

8.7 Conclusions on validity status checks

The present clause introduces the topic of revocation and validity status checks in the context of selective disclosure capable and unlinkable (Q)EAAs. If explicit (and short) validity periods are not used as an alternative, then it is recommended that the validity status check employed does not introduce a correlation handle in cases where selective disclosure and unlinkability are required. Concretely put, long lived (Q)EAAs that support selective disclosure and unlinkability using the mechanisms described in the present document:

- Are recommended to use OCSP in Must-Staple mode.
- May use validity Status List bit vectors rather than CRLs.
- Cannot rely on Revocation Lists or validity Status Lists without additional privacy considerations as detailed above. Seemingly, the use of Revocation Lists or Status Lists requires some form of Private Information Retrieval or Private Set Intersection techniques not to undermine selective disclosure and unlinkability.
- Can use cryptographic accumulators where possible given the associated complexity. Bloom filters represent an easy first step, whereas universal dynamic accumulators with public batch witness updates represent an interesting possibility for the future development of validity status checks of anonymized credentials and zero knowledge proofs. There is also work focused specifically on accumulators for EUDI Wallet core formats [i.112].
- May be combined with ZKP schemes (such as zk-SNARK) such that the status validity checks are performed at the digital wallet, and only the relevant information is disclosed with the verifier.

NOTE: Revocation checks can be considered as a predicate - a computation on the revocation ID included in the (Q)EAA's cryptographic meta-data and additional public information about the revocation registry. However, there may be further non-public inputs going into this computation provided by the holder to perform an inclusion check, e.g. a Merkle proof.

Ultimately, there is no suitable validity status mechanism that is both simple, mature in terms of standards, and that matches unlinkability requirements of (Q)EAAs capable of selective disclosure and data unlinkability.

Where selective disclosure and unlinkability is required, it is presently advisable to rely on short lived (Q)EAAs with explicit validity periods. Where users are identified, and/or when using formats based on salted attribute hashes where full unlinkability guarantees cannot be made, standard solutions like RL and SL are suitable.

9 Post-quantum considerations

9.1 General remarks

The recent years have witnessed significant advances in the area of quantum computing, which led to reconsider the threats posed by quantum algorithms such as the one devised by Shor [i.235] in 1994. The latter algorithm could indeed be used to attack the mathematical problems underlying most of the current asymmetric cryptographic algorithms, including several of those presented in the present document.

While there is still a lot of uncertainty surrounding the advent of a Cryptographically Relevant Quantum Computer (CRQC), it can be noted that the questions of whether a CRQC will be built, and when, have become less crucial, for at least two reasons. The first one is that the confidentiality of current data is already at risk because the data encrypted today using non-quantum-safe cryptographic algorithms could be stored and then decrypted by a CRQC in the future. This attack is broadly known as "store now decrypt later" and questions the interest of precisely predicting the date of the Q-day (the advent of CRQC). Indeed, knowing whether the problem will occur in 2030, 2032, etc. is only relevant if it is considered, for example, that leaking sensitive data is acceptable in 2032 but not in 2030. On this note, the NIST IR 8547 "Transition to Post-Quantum Cryptography Standards" [i.209] has declared that ECDSA, EdDSA and RSA are disallowed after 2035.

As most use-cases are unlikely to provide such a granularity for data shelf life it needs to be considered that every data with long-term sensitivity should be protected as of now. The second reason is that most cybersecurity agencies worldwide urge to initiate the transition to quantum-safe cryptography (also known as post-quantum cryptography) as soon as possible (see, e.g. [i.2] and [i.209]). This has already become mandatory for some systems in the US [i.243]. Transition is thus likely to become necessary for compliance and interoperability reasons, regardless of the actual advances of quantum computing.

In this regard, it is important to assess the impact of quantum computers on the QEAAs systems described in the present document. To this end, it is first needed to clarify the actual consequences of the Shor's algorithm. In particular, the fact that the latter solves the main mathematical problems underlying elliptic curves, finite fields and RSA cryptography does not mean that every security assurance provided by a cryptographic mechanism implemented in these settings is lost. Indeed, a security property of such a cryptographic mechanism may rely on a different problem or even be proved unconditionally, that is, regardless of the computational power of the adversary. In such cases, the security property remains even in the presence of a quantum computer. This is fortunately the case of many QEAAs constructions presented in the present document and, while every construction will require a dedicated quantum risk assessment, the following general comments can be made:

1) QEAA systems based on multi-message signature schemes often achieve unconditional privacy which means that their privacy is not affected by quantum algorithms. This is for example the case for anonymous credentials based on BBS+ (clause 4.4.2), BBS# (clause 4.4.3), CL (clause 4.4.1) and PS-MS (clause 4.4.5) signature schemes. This also holds true for some of the extensions discussed in the present document such as [i.240] and [i.232] and for the KVAC scheme in [i.15]. This property can however be lost by some variants thereof, such as the DAA systems presented in clause 6.4.2. It is therefore important to understand that unconditionally privacy is not a property inherent to these multi-message signature schemes but only results from a careful design of the QEAA system. Any modification of the latter (e.g. to add a new feature) might then affect this property.

2) In QEAA systems based on salted attributes hashed, the privacy of non-disclosed attributes is protected by the salt entropy which prevents exhaustive search. While a quantum computer could theoretically improve this exhaustive search by running Grover's algorithm, it can be noted that the actual performance of the latter is still unclear. In the worst case, it would only lead to a quadratic speedup, which means that doubling the salt size would be sufficient to retain the same security assurances as the one these systems enjoy today against non-quantum adversaries.

3) Conversely, for all these systems, an adversary equipped with a quantum computer will be able to forge valid attestations by solving the underlying mathematical problem.

In other words, a quantum adversary will be able to break the authenticity of QEAA systems but not (in most cases) their privacy. This subtlety is far from being insignificant as it means that all QEAA systems achieving unconditional privacy are immune to the store now decrypt later attack and so could postpone their transition as long as it is completed before the Q-day.

Finally, it can be noted that several cybersecurity agencies recommend the use of so-called hybrid mechanisms, that is, mechanisms combining current cryptographic algorithms and post-quantum ones. In such a case, the systems presented in the present document will not have to be discarded but simply completed with post-quantum solutions.

9.2 Post-quantum computing threats

A quantum computer capable of cryptanalysis remains a speculative prospect for a remote future despite the current level of trepidation. While a remote risk, the emergence of one with the computational power to execute algorithms like Shor [i.235] or Grover [i.126] could significantly affect the proposed solutions. To fully realize the impact of quantum computers, it is important to understand three things:

1) when they become a threat;

2) how quickly an attack is performed; and consequently

3) what they threaten.

One way to assess when a quantum computer can be a threat is to look at the requirements for launching a particular attack. These requirements can be expressed as logical qubits (a collection of physical qubits to protect against errors, where each logical qubit acts as the unit of information analogous to a classical bit). Proos and Zalka 2008 [i.225] show that computing the ECDL on an elliptic curve of order n field requires roughly 6*n* qubits without degradation and error rates. However, due to degradation and error rates, it makes more sense to discuss logical qubits and estimate the number of physical qubits for various degradation and error rates. For one reasonable estimate, Roetteler et al. 2017 [i.230] conclude that the ECDL on an elliptic curve defined over an *n*-bit prime field can be computed with at most $9n + 2 \times \text{ceil}(\log2(n)) + 10$ qubits. This means that 2330 logical qubits are required to perform NIST P-256 point addition and the full Shor algorithm on NIST P-256 would require 1,26 × 10^{11} universal gates. A final, but important consideration relating to the when, is that once a malicious and extremely well-resourced entity is equipped with a quantum resource it has to choose what to employ this resource on.

Another important consideration is to estimate how quickly the attack, once possible, can be performed. This is important because the time frame for the attack determines both the required size of the quantum computer and what threat it poses. It is thus incorrect to assume that the emergence of a quantum computer capable of cryptanalysis immediately renders all classical cryptography obsolete; an attacker will carefully deploy their quantum computers and each attack takes time. It is difficult to provide an exact size estimation for a given time frame given the many assumptions that need to be made about how a future quantum computer may operate. But with reasonable assumptions, Webber et al. 2022 [i.252] estimate that breaking a 256-bit elliptic curve cryptography within a day would require 13 million physical qubits and a quantum computer capable of running Shor's algorithm [i.235].

After examining the conditions under which a quantum computer could pose a threat and the associated timeframes, the next crucial consideration is to identify the specific targets such a quantum computer would jeopardize within a defined timeframe. This elucidates the threats posed to (Q)EAAs and provides insights into potential countermeasures that prospective (Q)EAA issuers and users can take.

The most significant threat, the Harvest Now, Decrypt Later (HNDL) threat, arises when a quantum computer is utilized on the sensitive ciphertext. In this scenario, an attacker monitors the key agreement between two actors, collects the ciphertext, and employs their quantum computers to find the negotiated symmetric decryption key. The threat here is one against confidentiality, i.e. the extraction of information about the signed message that the signer did not intend to disclose or the signature value itself in ZKP-capable signature schemes. The timeframe for such an attack can span the duration during which the encrypted data retains its sensitivity. Where an (Q)EAA contains information at risk of an HNDL attack, the risk of quantum computers necessitates that the (Q)EAA Provider abstains from using encryption schemes, and/or key sizes, where quantum computers pose a threat. An (Q)EAA Provider has many possible alternatives they could rely on, such as quantum-safe algorithms, zero-knowledge proofs that are quantum resistant (e.g. those based on cryptographic hash functions), increased key sizes, or Oblivious Pseudo-Random Functions, to name a few. However, Providers are recommended to take great care in the mitigating steps they take and be entirely sure that these protect against a HNDL attack.

Another risk is that of signature and proof forging, which is arguably more relevant to the topic of the present document. Here, the risk is relatively much lower due to the time frames involved. Note that an attacker cannot begin the attack without knowledge of some public material (e.g. a public key) derived from the sensitive cryptographic material. The threat here is one against integrity and authenticity, i.e. that the attacker would need to forge signatures, disclosures, and/or proofs. Note also that the attacker does not have the same time frames at their disposal as in the case of an HNDL attack as the attack target is not a decryption key that can be used on pre-collected sensitive ciphertext. Actors may deploy frequent key rotation and rely on short-lived attestations to mitigate the quantum threat. The potential use of one-time signing and proof keys provides excellent protection against an attacker with a quantum computer. Frequent key rotation, or even one-time use of keys, is likely viable for the foreseeable future given existing development trajectories. Once the threat level is sufficiently high, actors can move to alternative signature algorithms (e.g. CRYSTALS Dilithium) and post-quantum safe zero-knowledge solutions.

EXAMPLE: The complexity of forging documents that have been digitally signed in a pre-quantum world can be illustrated by this example. Assume that Alice digitally signs a document in the pre-quantum world. The signed document is also time-stamped by a trusted time-stamping authority. She stores the digitally signed document in an archive, which has an audit log where each log entry is digitally signed and each signed log entry is added to a chain of hashes of previous log entries. In a post-quantum world, the attacker Bob will be able to derive Alice's private key from her public key in the X.509 certificate. Hence, he can create a forged document and sign this with her private key and certificate. However, in order to replace the existing signed document, which is archived, Bob would also need to attack the time-stamping authority to generate a forged time-stamp (with a rewinded clock). He would also need to attack the archive to delete the existing document, replace it with the forged document, and finally forge the signed audit log and hash chain of log entries. Such an attack is utterly complicated to perform, even with the use of quantum computers.

The related concept of everlasting privacy, which is typically applied to e-voting schemes, aims at ensuring the electronic votes will remain secret and secure also in the future. For more information on everlasting privacy the following research papers are recommended: "Practical Everlasting Privacy" [i.8] by Arapinis et al., "Towards everlasting privacy and efficient coercion resistance in remote electronic voting" [i.123] by Grontas et al, "Improvements in Everlasting Privacy: Efficient and Secure Zero Knowledge Proofs" [i.128] by Haines et al. and "SoK: Secure e-voting with everlasting privacy" [i.129] by Haines et al.

9.3 Post-quantum computing solutions

Although (Q)EAA systems are not immediately threatened by quantum computing, as explained in clause 9.2, they will eventually have to migrate to post-quantum cryptography, at least before the Q-day.

In the case of salted attributes hashes, the main component vulnerable to quantum computers is the signature scheme used to sign the hash values. Transition to post-quantum cryptography will then mostly consist in replacing this signature scheme by a post-quantum counterpart such as the NIST standard FIPS 204 [i.207].

The case of (Q)EAA based on multi-message signature schemes is more complex as post-quantum variants for these particular signature schemes will be needed, but also for the related zero-knowledge proof systems. This is today a very active research area whose main advances are presented in clause 9.4.

9.4 Lattice-based anonymous credentials schemes

9.4.1 Background

The transition to post-quantum cryptography is an enormous challenge for cryptographers and the IT-security industry as a whole. There have been significant enhancements such as the future NIST standards on Post-Quantum Safe (PQS) cryptography. However, these NIST standards have so far only been focusing on general cryptographic mechanisms, such as digital signatures or key exchange, whilst there are not yet any similar PQS standardization efforts for blind signatures, group signatures, and anonymous credentials.

Nevertheless, there are cryptographic research initiatives in the field of PQS multi-message signatures and anonymous credentials. In 2016, Libert et al. published the research paper "Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions" [i.197]. The result of this research indicated that anonymous credential schemes, which are based on plausibly PQS cryptography using lattices, generate signature and proof sizes in the magnitude of several hundreds of MB. This lattice-based scheme is however outdated, and the research to improve the performance and proof sizes has continued as described in clause 9.4.2.

Another option is to apply PQS zk-SNARKs to the Cinderella project (see clause 6.5.2), whereby PQS ZKPs can be derived from X.509 certificates. Potential PQS zk-SNARKs for such a setup are Spartan [i.200], Virgo [i.273] or Ligero [i.20]. Furthermore, the X.509 certificates would need to be signed with PQS cryptographic algorithms, such as CRYSTALS Dilithium [i.75]. There are also programmatic issues to be resolved with such an integration, such as patching the vulnerability in the Gepetto compiler.

Hence, until recently there have essentially been two alternatives to achieve a plausible PQS ZKP system: a system with large signature and proofs that rely upon cryptographic algorithms, or a system based on ad-hoc integrations of PQS zk-SNARKs. The research of how to improve the performance and proof sizes of PQS ZKP systems has however progressed in recent years, which is further described in clause 9.4.2.

9.4.2 Research on effective lattice-based anonymous credentials

In order to address the issues with large sized signatures, cryptographic research is currently being performed on PQS anonymous credentials with small signature sizes.

In 2022, Jeudy et al. published the cryptographic research paper "Lattice Signature with Efficient Protocols, Application to Anonymous Credentials" [i.192]. The paper introduced a new construction that is based both on standard lattices and structured ones, which resulted in significant performance improvements. In particular, the size of a signature proof was reduced to less than 650 KB.

Based on Jeudy's research, Dutto et al. proposed a PQS ZKP scheme in their paper "Toward a Post-Quantum Zero-Knowledge Verifiable Credential System for Self-Sovereign Identity" [i.83], which describes PQS variants of BBS+ and CL-signatures based on a lattice-based scheme.

The research by Jeudy et al. was continued in 2024 by Argo et al. who published their research paper "Practical Post-Quantum Signatures for Privacy" [i.9] that proposes privacy-preserving Signatures with Efficient Protocols (SEP). The SEP is lattice-based and generates short-sized signatures that are PQS. Furthermore, the SEP has been integrated with an anonymous credential system, resulting in anonymous credentials of less than 80 KB. The source code of this project is published at the repository "Lattice Anonymous Credentials" [i.10].

Furthermore, Bootle et al. published the research paper "A Framework for Practical Anonymous Credentials from Lattices" [i.29] in 2023. Their paper introduces a framework for practical anonymous credential schemes based on a new family of lattices. The security of this lattice scheme is based on the difficulty to generate a pre-image for an element given short pre-images of random elements in a set. Such a framework can be used to implement efficient privacy-preserving cryptographic primitives for blind signatures, anonymous credentials, and group signatures.

Hence, there are several cryptographic research initiatives that aim at inventing anonymous credentials and privacy-preserving signature schemes that are PQS with efficient and small-sized signature proofs.

10 Conclusions

The eIDAS2 regulation and the Architecture and Reference Framework (ARF) define regulatory requirements on selective disclosure and unlinkability for the EUDI Wallet. The present document provides a comprehensive analysis of signature schemes, credential formats and protocols that cater for selective disclosure, unlinkability, and predicates.

Since the ARF specifies that a PID Provider can issue any PID in both the format specified in ISO/IEC 18013-5 [i.181] and the SD-JWT VC format, the present document analyses ISO mDL and SD-JWT VC.

The ISO mDL specified mdocand the IETF SD-JWT formats and related presentation protocols cater for selective disclosure using a hashed salted attributes approach. Both MSO and SD-JWT support SOG-IS approved cryptographic algorithms and can also be used with quantum-safe cryptography for future use. The conclusion is thus that MSO (as detailed in ISO mDL) as well as the SD-JWT approach meet the eIDAS2 regulatory and technical requirements on selective disclosure when defined as revealing at least one attribute from a single PID or (Q)EAA. Neither format supports selective disclosure of at least two attributes from multiple distinct PID/(Q)EAAs. Neither format supports predicates, although the present document also proposes a new approach to calculate predicates based on hash chains in conjunction with salted attribute hashes, which can be used for dynamically deriving statements about the user without revealing the attribute values.

In addition to limited selective disclosure capabilities, the major drawback with mdoc and SD-JWT is the lack of unlinkability. Neither of the formats supports issuer unlinkability or full unlinkability, and verifier unlinkability encumbers the issuer. In order to achieve verifier unlinkability, batches of MSOs or SD-JWTs need to be issued to each EUDI Wallet. When the PID Provider (PIDP) or QTSP supports batch issuance with unique salts, both MSO and SD-JWT can support verifier unlinkability. In order to achieve verifier unlinkability, the random salts in the MSO and SD-JWT should be unique, meaning that refreshed MSOs and SD-JWTs are presented to a relying party.

The present document gives recommendations on how eIDAS2 compliant PIDPs or QTSPs can issue PID/(Q)EAAs in the form of mdoc and/or SD-JWT that cater for selective disclosure. The present document notes that SD-JWT can provide selective disclosure capability also for attestations that use JSON-LD and linked data proofs but advises against it (support for data integrity proofs is lacking and there exist security concerns with polyglot parsing).

There are many similarities between the mdoc issuers and the eIDAS2 QTSPs or PID providers, which could be harmonized in ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97] that will standardize the issuance policies and profiles of (Q)EAAs. More specifically, the MSO could be issued by an eIDAS2 QTSP certification authority, meaning that the EU trusted lists can be used to retrieve revocation information and trust anchors when validating the MSO signature. ETSI TS 119 495 [i.93], which specifies certificate profiles and TSP policies for Open Banking and PSD2, may partially be re-used for the issuance of ISO mdocs as (Q)EAAs. The same principles could be applied on QTSPs and PID providers that will issue PIDs/(Q)EAAs in conjunction with SD-JWT, although the existing specifications do not specify the issuance policies in detail.

Furthermore, there are recommendations on how to store MSO and SD-JWT VC compliant representation for JWT in the EUDI Wallet, and how to present selectively disclosed attributes to eIDAS2 relying parties. The presentation protocols for the ISO mDL and OID4VP are specified in the ARF, and the present document describes how to use these protocols for selective disclosure of attributes in mdoc and SD-JWT.

The multi-message signature schemes on the other hand are designed to provide selective disclosure and full unlikability. Such multi-message signature schemes are BBS+, CL-Signatures, PS-MS signatures and Mercurial signatures. However, such signature schemes are based on pairing-based elliptic curve cryptographic algorithms that are not yet fully standardized. So far, ISO/IEC 20008 [i.184] has standardized single-message signature schemes that underpin BBS and PS-MS, but they are not sufficient for PID formats and (Q)EAAs that require multi-message signature schemes. However, ISO/IEC 24843 [i.185] intends to standardize BBS+ with blinded signatures, which may allow for a future standard that could be used in compliance with the EUDI Wallet requirements on selective disclosure and unlinkability in eIDAS2. Furthermore, there are cryptographic research projects, such as MoniPoly, where undisclosed attributes have no impact on the proof size.

BBS# [i.78] is a variant of BBS/BBS+ that has been designed to meet several stringent requirements put forth in the eIDAS 2.0. regulation. More precisely, BBS# removes the need for pairings and pairing-friendly curves (which are not standardized and not supported by trusted phone hardware) and can be combined with SOG-IS sanctioned protocols for the implementation of the holder binding feature. The BBS# scheme can be made format compatible to mdoc and SD-JWT, thus catering for full unlinkability of mdoc and SD-JWT.

Another interesting approach to achieve solutions for the EUDI Wallet with selective disclosure and full unlinkability are the systems that combine ZKP schemes (such as zk-SNARKs) with existing digital identity infrastructures (such as X.509 certificates or ICAO eMRTD). There are existing research projects, such as Cinderella, Crescent and zk-creds, that have succeeded to implement prototypes where zk-SNARKs are used to generate pseudo-certificates that share selected attributes from the (Q)EAAs and derived revocation information. Furthermore, the research of "Anonymous credentials from ECDSA" ("zk-mdoc") provides a ZKP solution for the existing ISO mDL protocols. These projects are still in the research phase, but may be considered for the EUDI Wallet and eIDAS2 relying parties.

In order to achieve privacy preserving features for revocation and validity status checks it is recommended to use OCSP in Must-Staple mode, implement Revocation Lists or validity Status Lists with additional privacy techniques such as Private Information Retrieval or Private Set Intersection, and use cryptographic accumulators where possible given the associated complexity. If ZKP schemes (such as zk-SNARKs) are combined with existing (Q)EAAs (such as X.509), the status validity checks are performed at the EUDI Wallet, and only the relevant information is disclosed with the verifier.

Annex A:
Comparison of selective disclosure mechanisms

A.1 Selective disclosure signature schemes

Table A.1 provides a comparison of the investigated selective disclosure signature schemes.

Table A.1: Comparison of selective disclosure signature schemes

Signature scheme	Cryptography	Plausible quantum-safe	Unlinkability	Predicates	Reference
Category: Atomic attribute (Q)EAAs					
Atomic attribute (Q)EAAs	Conditional: depends on the signature on the credential	Yes, the (Q)EAAs can be signed with QSC algorithms.	Verifier unlinkable attestations can be achieved. Fully unlinkable (Q)EAAs are not possible.	No dynamic predicates are supported. Workaround: enrol for atomic attributes with Boolean attributes.	See clause 4.2
Category: Salted attribute hashes					
Salted attribute hashes	**Salted attribute hashes**, signed with RSA, ECC, or QSC	Yes, the (Q)EAAs can be signed with QSC algorithms.	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not protected against issuer linkability.	No dynamic predicates are supported. Workaround: set Boolean attributes in the PID/(Q)EAA.	See clause 4.3
ACDC	Salted attribute hashes structured in a Directed Acyclic Graph	Yes	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not fully unlinkable.	No dynamic predicates are supported. Workaround: set Boolean attributes in the PID/(Q)EAA.	See clause 4.3.8
Gordian Envelopes	Salted attribute hashes structured in a Directed Acyclic Graph	Yes	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not fully unlinkable.	No dynamic predicates are supported. Workaround: set Boolean attributes in the PID/(Q)EAA.	See clause 4.3.9
HashWires	Salted attribute hashes structured in a chain of hashes	Yes	Verifier unlinkability can be achieved if unique salts are used when creating the salted attribute hashes, but the schemes are not fully unlinkable.	HashWires supports range proofs that can be combined with selectively disclosed salted hashes of attributes (see clause 4.3.7).	See clause 4.3.7

Signature scheme	Cryptography	Plausible quantum-safe	Unlinkability	Predicates	Reference
Category: Multi-message signature schemes					
BBS+ signatures	Multi-message signature scheme based on ECC bilinear pairings	ZKPs generated pre-quantum will remain plausible safe post-quantum. BBS+ is plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.4.2
BBS# signatures	Multi-message signature scheme based on conventional elliptic curves (such as the NIST P-256 curve).	ZKPs generated pre-quantum will remain plausible safe post-quantum. BBS+ is plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.4.3
Camenisch- Lysyanskaya (CL) signatures	Multi-message signature scheme based on strong RSA assumption	ZKPs generated pre-quantum will remain plausible safe post-quantum. CL-signatures are plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.4.1
Mercurial Signatures	Multi-message signature scheme based on Decisional Diffie-Hellman (DDH)	ZKPs generated pre-quantum will remain plausible safe post-quantum. MS is plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.4.4
Pointcheval- Sanders Multi-Signatures (PS-MS)	Multi-message signature scheme based on improved CL-signatures	ZKPs generated pre-quantum will remain plausible safe post-quantum. PS-MS is plausible vulnerable in a post-quantum world.	Fully unlinkable with blinded signatures.	Yes (in theory)	See clause 4.4.5
Category: Proofs for arithmetic circuits (programmable ZKPs)					
Bulletproofs	Proofs for arithmetic circuits based on Fiat-Shamir heuristics	No	Yes	Yes	See clause 4.5.4
zk-SNARKs	Proofs for arithmetic circuits based on various mechanisms in clause A.4	Some zk-SNARK schemes are QSC, see Table A.4.	Yes	Yes	See clauses 4.5.2 and A.4
zk-STARKs	Proofs for arithmetic circuits based on various mechanisms	Yes	Yes	Yes	See clause 4.5.3

A.2 (Q)EAA formats with selective disclosure

Table A.2 provides a comparison of the investigated credential formats with selective disclosure.

Table A.2: Comparison of credential formats with selective disclosure

(Q)EAA format	Scheme	Encoding	Maturity	Reference
Category: Atomic attribute credentials				
IETF X.509 attribute certificates	Atomic attribute (Q)EAAs	ASN.1/DER	X.509 attribute certificate (IETF RFC 5755 [i.158]) is an IETF PKIX standard	See clause 5.2.2
W3C Verifiable Credentials	Atomic attribute (Q)EAAs	JSON-LD or JWT	W3C VC Data Model [i.264] is a standard	See clause 5.2.3
Category: Salted attribute hashes				
IETF SD-JWT	Salted attribute hashes	JSON (JWT)	IETF SD-JWT draft standard [i.155], several reference implementations	See clause 5.3.2.1
IETF SD-JWT VC	Salted attribute hashes	JSON (JWT)	IETF SD-JWT VC draft standard [i.143], several reference implementations	See clause 5.3.2.2
ISO/IEC 18013-5 [i.181] Mobile Security Object (MSO)	Salted attribute hashes	CBOR/CDDL (COSE)	ISO/IEC 18013-5 [i.181], implemented in several wallets, deployed in the US	See clause 5.3.3
Category: Multi-message signature schemes				
Hyperledger AnonCreds	CLRSA-signatures	JSON (JWS)	Deployed in Government of British Columbia, IDunion, and the IATA Travel Pass	See clause 5.4.4
W3C VC with ZKP	Various MMS schemes, CL-signatures explicitly referenced	JSON (LD)	W3C VC Data Model [i.264], implemented in several wallets	See clause 5.4.1
W3C VC Data Integrity with BBS+ signatures	BBS+ signatures	JSON (LD)	W3C VC Data Integrity [i.263]	See clause 5.4.2
W3C VC Data Integrity with ECDSA-SD	ECDSA-SD signatures	JSON (LD)	W3C VC Data Integrity [i.263]	See clause 5.4.3
Category: JSON container formats				
IETF JSON Web Proof	Flexible: CL-signatures, BBS+, etc.	JSON (JWS)	IETF JSON Web Proof draft standard [i.90]	See clause 5.5.1
W3C JSON Web Proofs For Binary Merkle Trees	Merkle trees	JSON Web Proofs	W3C draft specification	See clause 5.5.1
JSON Web Zero Knowledge (JWZ)	Zero-knowledge proofs, for example Groth-16	JSON Web Proofs	Part of Iden3 protocol stack, several reference implementations	See clause 5.5.3

A.3 Selective disclosure systems and protocols

Table A.3 provides a comparison of the investigated selective disclosure protocols.

Table A.3: Comparison of selective disclosure systems and protocols

Protocol	Credentials	Protocol	Maturity	Reference
Category: Atomic attribute (Q)EAAs				
IETF X.509 attribute certificate (protocol)	IETF X.509 attribute certificates	Attribute certificate authorization protocol	X.509 attribute certificate [i.158] is an IETF PKIX standard	See clause 6.2.1
VC-FIDO	W3C Verifiable Credentials	VC-FIDO	Deployed as a prototype at NHS in the UK	See clause 6.2.2
Category: Salted attribute hashes protocols				
Singapore's Smart Nation OpenAttestation	Document Integrity credentials	OpenAttestation protocol [i.211]	Deployed at the Singapore's Smart Nation	See clause 6.3.1
Category: Multi-message signature schemes				
Hyperledger AnonCreds (protocol)	AnonCreds [i.131] based on CLRSA-signatures	Hyperledger Aries protocol [i.132] in conjunction with Hyperledger AnonCreds SDK [i.131]	Deployed in Government of British Columbia, IDunion, and the IATA Travel Pass	See clause 6.4.1
Direct Anonymous Attestation (DAA)	DAA credentials	ISO/IEC 20008-2 [i.184]	Deployed at large scale by TCG in TPM 2.0 and Intel® in EPID 2.0	See clause 6.4.2
Iden3	W3C Verifiable Credentials with Iden3 Signature Schemes	Verifiable Credentials with BJJ Signature [i.138] and Verifiable Credentials with SMT Signature [i.140]	Web2 and Web3 projects performed at organizations such as the Ethereum Foundation, Deutsche Bank, HBSC, Kaleido, Rarimo, and others are using the Iden3 stack	See clause 6.9
Category: Proofs for arithmetic circuits solutions				
Cinderella	X.509 certificates	zk-SNARK (Pinocchio)	In research phase	See clause 6.5.2
zk-creds	ICAO eMRTDs	zk-SNARK (Pinocchio)	In research phase	See clause 6.5.3
Anonymous credentials from ECDSA	mdoc [i.181]	ECDSA	Implemented in a prototype of Google® Wallet	See clause 6.5.4
Crescent	JWT and mdoc [i.181]	Sigma-protocols combined with zk-SNARK (Groth16)	In research phase	See clause 6.5.5
Category: ABC (Attribute Based Credentials)				
Idemix	Idemix ABC credentials [i.136] based on CL-signatures	Idemix ABC protocol [i.136]	Implemented by IBM®, Hyperledger Fabric [i.133], IRMA project [i.227], and the EU-projects PrimeLife [i.224] and ABC4Trust [i.137]	See clause 6.6.1
U-Prove	U-Prove ABC credentials [i.201]	U-Prove ABC protocol [i.201]	Implemented in Microsoft® Identity Metasystem and the EU-project ABC4Trust [i.137]	See clause 6.6.2
ISO/IEC 18370 [i.183]	U-Prove ABC credentials [i.201]	ISO/IEC 18370 [i.183]	Implemented in U-Prove solutions, security flaws detected	See clause 6.6.3
Keyed-Verification Anonymous Credentials (KVAC)	Keyed-Verification Anonymous Credentials	BBS_MAC+ [i.15]	Implemented as a prototype on SIM-cards	See clause 6.6.4
FIDO-AC	ICAO eMRTDs	FIDO2 (WebAuthn)	In research phase	See clause 6.6.5

Protocol	Credentials	Protocol	Maturity	Reference
Category: ISO mobile driving license (ISO mDL)				
ISO/IEC 18013-5 [i.181] (device retrieval)	ISO/IEC 18013-5 [i.181] mDL/MSO [i.181]	ISO mDL/MSO over BLE/NFC	ISO standard, implemented in several wallets, deployed in the US	See clause 6.7.2
ISO/IEC 18013-7 [i.182] (unattended)	ISO/IEC 18013-5 [i.181] mDL/MSO [i.181]	SIOP2 [i.216], OID4VP [i.214]	Draft ISO/IEC CD 18013-7 [i.182] standard, correlated with ISO/IEC CD 23220-4 [i.187]	See clause 6.7.4
ISO/IEC 23220-4 [i.187]	ISO mDL [i.181], SD-JWT [i.155], etc.	SIOP2 [i.216], OID4VP [i.214]	Draft standard, correlated with ISO/IEC CD 18013-7 [i.182]	See clause 6.7.5
ISO/IEC 18013-5 [i.181] (server retrieval)	OpenID Connect ID-Token [i.212]	OpenID Connect (OIDC) Core [i.212]	ISO standard, implemented in several wallets, deployed in the US	See clause 6.7.3
Category: OpenID for Verifiable Credentials (OpenID4VCI)				
OpenID for Verifiable Credential Issuance (OpenID4VCI)	ISO mDL [i.181], SD-JWT [i.155], etc.	OpenID4VCI [i.213]	Draft standard, implemented in several wallets and pilot projects	See clause 6.8.1
OpenID for Verifiable Presentations (OpenID4VP)	ISO mDL [i.181], SD-JWT [i.155], etc.	OpenID4VP [i.214]	Draft standard, implemented in several wallets and pilot projects	See clause 6.8.2
OpenID4VC High Assurance Interoperability Profile (HAIP)	ISO mDL [i.181], SD-JWT [i.155], etc.	HAIP [i.215]	Draft standard, implemented in several wallets and pilot projects	See clause 6.8.3

A.4 zk-SNARK protocols

Table A.4 provides a comparison of the different zk-SNARK protocols.

The comparison is made based on transparency, universality, and plausible quantum-safety. A transparent protocol is defined as it does not require any trusted setup and uses public randomness. A universal protocol is defined as it does not require a separate trusted setup for each circuit. A plausibly quantum-safe protocol is one that is not considered to be vulnerable to attacks by quantum computing algorithms.

Table A.4: Comparison of zk-SNARK protocols

Protocol	Published	Transparent	Universal	Quantum-safe
Pinocchio [i.220]	2013	No	No	No
Geppetto [i.72]	2015	No	No	No
TinyRAM [i.19]	2013	No	No	No
Buffet [i.249]	2015	No	No	No
ZoKrates [i.85]	2018	No	No	No
xJsnark [i.195]	2018	No	No	No
vnTinyRAM [i.21]	2014	No	Yes	No
MIRAGE [i.194]	2020	No	Yes	No
Sonic [i.198]	2019	No	Yes	No
Marlin [i.66]	2020	No	Yes	No
PLONK [i.116]	2019	No	Yes	No
Spartan [i.200]	2019	No	Yes	Yes
SuperSonic [i.39]	2020	Yes	Yes	No
Hyrax [i.250]	2018	Yes	Yes	No
Halo [i.31]	2019	Yes	Yes	No
Virgo [i.273]	2020	Yes	Yes	Yes
Ligero [i.4]	2017	Yes	Yes	Yes
Aurora [i.20]	2019	Yes	Yes	Yes
Groth16 [i.124]	2016	No	No	No
Ligetron [i.248]	2024	No	Yes	Yes

Annex B:
Hash wires

B.1 HashWires applied on inequality tests

B.1.1 Using a hash chain for inequality tests

A fundamental building block in HashWires is hash chains. Given two collision-resistant hash functions (H, G), a maximum integer value N, and a random value r, the issuer computes the commitment $c = H^k(G(r))$. Here, $H^k(\cdot)$ represents k iterations of the function H such that the digest of H^i is the pre-image to H^{i+1}. The issuer signs c and sends (c, r) to the user (optionally also k). The user can now produce a hash chain of the same length as a threshold t by computing the range proof $\pi = H^{k-t}(G(r))$. The user signs a presentation containing (π) and the verifier checks if $c = H^t(\pi)$. If the check passes, the verifier knows that c is the commitment to some value $t \leq x$ but does not learn k.

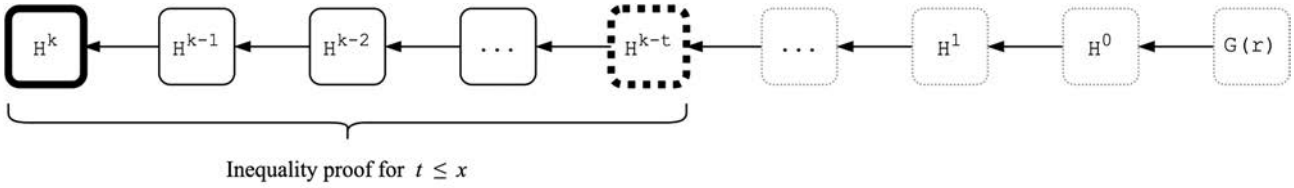


Figure B.1: A hash chain based inequality test

In Figure B.1, the issuer signs the leftmost bold box representing the commitment $c = H^k(G(r))$. The user presents the dotted bold lined box representing the threshold value $\pi = H^{k-t}(G(r))$. The verifier accepts π as a proof for the inequality $t \leq x$. Note that for an age proof, the value H^0 should represent the user's actual age k at the time of issuance and that H^k represents the minimum age value 0.

NOTE 1: The hash functions (H, G) should be listed in the SOG-IS table of agreed hash functions [i.237].

NOTE 2: The digital signature scheme should be listed in the SOG-IS table of agreed signature schemes [i.237].

NOTE 3: The use of digital signatures that are QSC should be possible.

NOTE 4: The verifier does not learn the value k, $G(r)$ and any $H^m(\cdot)$ where $m > t$.

NOTE 5: A single hash function with two different salts, or a keyed HMAC with two keys, are both alternatives to (H, G).

When considering non-negative integers, one obvious representation is that the H^0 digest represents the maximum value, and each subsequent digest represents a decrement by 1. The problem with that approach is that it does not scale. Take for instance age over or equal to proofs. Here, the user should be able to prove that their age is equal to or above 18 the very day they turn 18, but not before. A hash chain for 18 years in days requires roughly 6 575 digests. This is further exacerbated by the batch issuance requirement for PIDs and (Q)EAAs to prevent verifier collusion (the Provider would need to create a new hash chain for every attestation since the commitment would be correlatable even with a salt). Also, each verifier needs to recompute the threshold length of the chain at every presentation. With ~450 million EU citizens, and potentially multifold more inequality tests for age based services, optimization is required.

B.1.2 Using multiple hash chains for inequality tests

The optimization presented in the HashWires paper ensures that the commitment generation, proof and verification, and proof size all scale well even for very large *n*-digit numbers. The core idea is to rely on multiple hash chains. However, instead of representing decrements starting from the maximum number, each digest represents the commitment to the digits x_i of a number $x = x_n \cdot 10^n + x_{n-1} \cdot 10^{n-1} + \ldots + x_1 \cdot 10^1 + x_0$.

For instance, using the commitments to the coefficients in $22 = 2 \cdot 10^1 + 2 \cdot 10^0$ a user could generate a proof for the inequality $x \geq 10$. Note, however, that the user would not be able to use that commitment to prove $x \geq 13$ without revealing a lot more information than necessary (more specifically, the user would need to reveal commitments to 20).

Chalkias et al. [i.58] here describe the idea of Minimum Dominating Partitions (MDP) to address the above problem. In the HashWires paper, there is a formal definition of MDP, which relies on the idea that a number x dominates another number y if each digit $x_i \geq y_i$. The authors present an algorithm that takes a non-negative integer as input and outputs one or more non-negative integers that represent numbers that dominate other numbers, where the collection of numbers output can dominate any other number in the entire range of the requested inequality.

A simpler explanation is that the MDP is generated using a recursive function that takes as input a number, and outputs the first number that the input cannot dominate. That new output number then becomes the new input number, and the MDP outputs the value it cannot dominate. For instance, using base 10, the number 84 can dominate $\{84,83,82,81,80\}$ but not 79. Subsequently, 79 can dominate all numbers down to 0. So the $MDP(84) = \{84,79\}$. Similarly, $MDP(3413) = \{3413,3409,3399,2999\}$.

Given a set of MDP partitions, the user can use hash chains to dominate any number that up to and including the first element by simply picking the element that can dominate the requested threshold value. For instance, given $MDP(3413) = \{3413,3409,3399,2999\}$ the user can use the $\{2999\}$ element to prove $x \geq 376$. When the user can use more than a single element from the MDP to dominate the threshold number, the user picks the number that reveals the least amount of information.

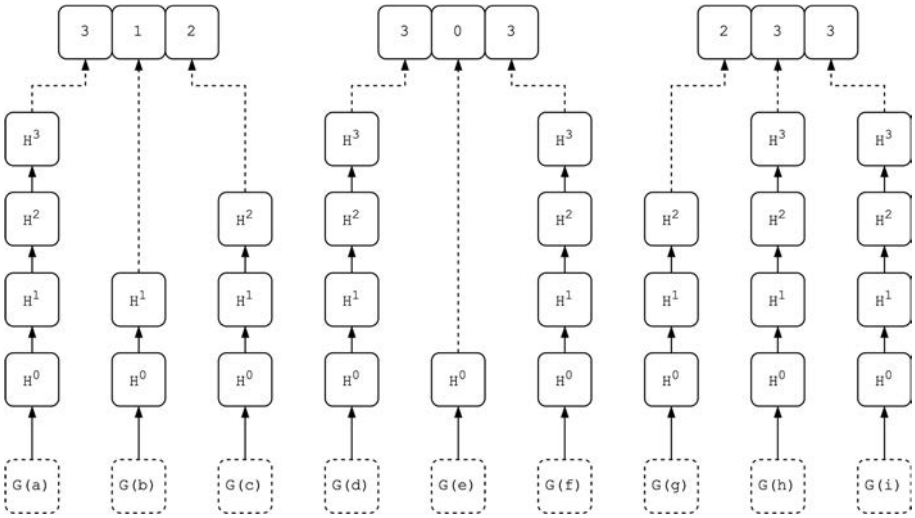


Figure B.2: Basic HashWires commitment

Figure B.2 illustrates a basic HashWires commitment to the number 312 in base 4 with $MDP_4(312) = \{312,303,233\}$. Each hash chain represents a commitment to a specific digit in each MDP partition.

A further optimization can be made by reusing the same hash chain for multiple different commitments. The idea here is to generate one hash chain per digit in the largest number, with the length of the hash chain being the largest value of any digit in any MDP partition.

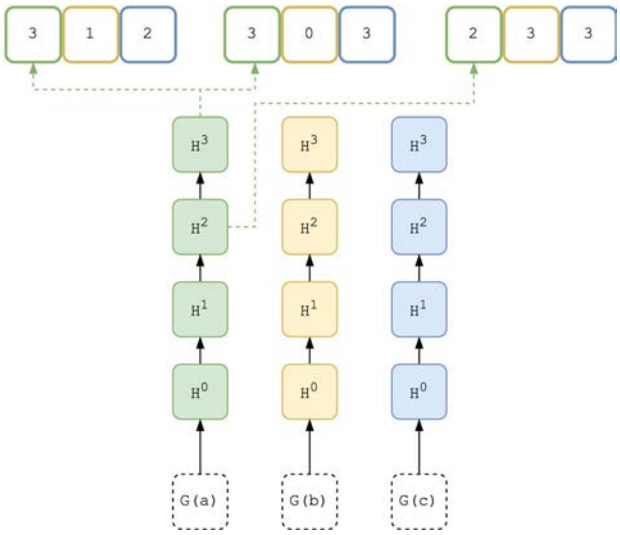


Figure B.3: Optimized HashWires commitment

Figure B.3 shows an optimized HashWires commitment to the number 312 in base 4 with $MDP_4(312) = \{312,303,233\}$. Each hash chain represents the commitments to the digit values of each partition. Green dotted line illustrates how the values are sourced for the third digit in each MDP partition. Hash chains are coloured to correspond to their commitments, i.e. the second digit in each MDP partition would source their commitment from the middle hash chain, and the first digit in each partition would source commitments from the rightmost hash chain.

The optimized HashWires approach is orders of magnitude more efficient than using a single hash chain. Specifically, the $MDP(6575) = \{6575,6569,6499,5999\}$ (18 years in days), requires $3 + 6 + 9 + 9 + 9 = 36$ hash operations (three for the seeds, and then 6 for the fourth digit, and then 9 for each subsequent digit). In fact, using base 10, the maximum possible number of hash chains will never exceed the number of digits multiplied by 10.

One concern with the optimized HashWires approach is that it may leak information about the partitions, and thus reveal the users actual number. To avoid such leaks, the authors of the HashWires paper suggest the use of an accumulator that can hide the actual commitments. While the use of an accumulator addresses the concern, it is also not necessary when the attestation format is capable of selectively disclosing the particular commitment that the user needs to prove the inequality, and when attestations are batch issued and used only once (that is not to say that the issuer cannot select to include the accumulator value as a selectively disclosable value).

B.1.3 Protecting optimized HashWires with SD-JWT or MSO

The MDP partitions leak information about the number in several ways. Therefore, it is important that the user only reveals the exact commitment that is required for the request threshold inequality proof. The original HashWires paper achieves this using an accumulator, but it is also possible to rely on the selective disclosure capabilities of SD-JWT and MSO. For reasons of readability, illustrative examples will be done using SD-JWT and without an accumulator, but the concept is equally applicable for MSO and every other salted attribute hashes based approach.

NOTE: Combining HashWires range proofs with selectively disclosed salted hashes of attributes is suggested by Peter Lee Altmann (Swedish Digitalization Agency) and Sebastian Elfors (IDnow) to the present document. The idea is not peer reviewed and is meant primarily to illustrate the idea of a PID/(Q)EAA Provider signing computational inputs and parameters to enable dynamic predicates e.g. inequality tests. With modifications, the proposal could enhance the mdoc [i.181] and IETF SD-JWT [i.155] standards to cater for predicate proofs in addition to selectively disclosing claims.

Consider an optimized HashWire for an *n*-digit number, $HW = \{[c_n, c_{n-1}, \ldots, c_0], [r_n, r_{n-1}, \ldots, r_0]\}$ where c_i denotes the hash chain root for digit position *i* in each MDP partition for a value *x* and r_i denotes the seed used in $G(\cdot)$ to generate the first value of the hash chain for each digit position *i*. Each MDP partition is a combination of hash roots.

For instance, the $MDP(6575) = \{6575,6569,6499,5999\}$ would require four seeds, resulting in four hash chains, one for each digit. The corresponding hash chains lengths for $MDP(6575)$ are $6 \cdot 10^3 + 9 \cdot 10^2 + 9 \cdot 10^1 + 9$.
More precisely:

- 6575 requires the commitment: $H^6(G(r_3)), H^5(G(r_2)), H^7(G(r_1)), H^5(G(r_0))$

- 6569 requires the commitment: $H^6(G(r_3)), H^5(G(r_2)), H^6(G(r_1)), H^9(G(r_0))$
- 6499 requires the commitment: $H^6(G(r_3)), H^4(G(r_2)), H^9(G(r_1)), H^9(G(r_0))$
- 5999 requires the commitment: $H^5(G(r_3)), H^9(G(r_2)), H^9(G(r_1)), H^9(G(r_0))$

Each commitment is required to be included in a disclosure, and then signed as part of the SD-JWT or MSO. The PID/(Q)EAA Provider is required to also include a number of decoy digests to hide the number of MDP partitions, or alternatively commit only an accumulator value (e.g. a Merkle Tree as proposed in the original HashWires paper or the digest over the concatenation of all the decoys and commitments). In Figure B.3, and in the example below, the commitments are included as separate disclosures for illustrative purposes only.

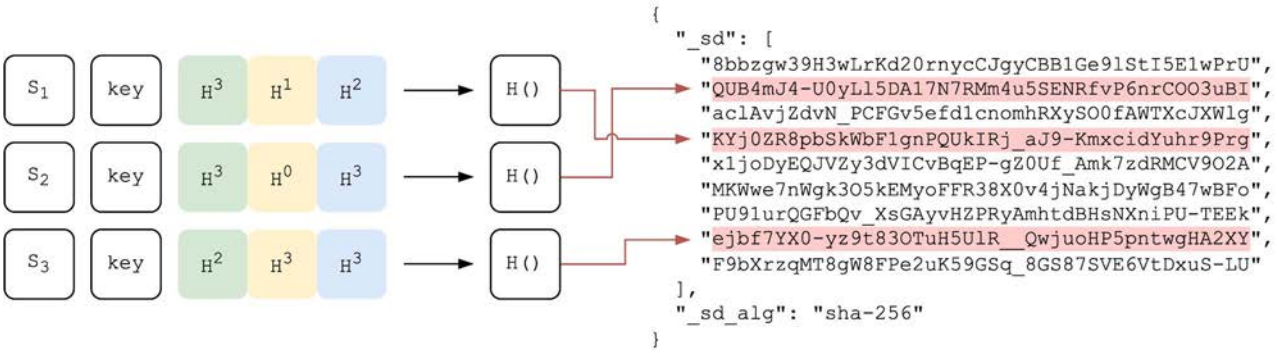


Figure B.4: Optimized HashWires commitment using SD-JWT

Figure B.4 illustrates an optimized HashWires commitment to the number 312 in base 4 protected by the `_sd` object suitable for an SD-JWT. Each commitment to the three partitions is salted (box with S), contain a MDP partition identifier key, and the hash chain roots for each MDP partition. The hash over the salt, key, and commitment is included in the `_sd` (red highlights). The other digests in the `_sd` object are decoys to hide the number of MDP partitions the user has. Each commitment is included as a disclosable value for illustrative purposes. Optionally, an issuer could instead add the commitments to an accumulator, which would be disclosable. This is an illustration of HashWires, although implementations may differ.

EXAMPLE: The random values needed to initiate each hash chain with $G(\cdot)$. The values are not sent to the verifier.

```
{
  "10^0": "f6a23b90b9f07f34f33dfd4e5de87adab167b6ea9eb060163e741ac26f16edc1",
  "10^1": "3026950fd2d2c6c7e23c8a8b0a80928d5cdac0f953699a96e02c1033379ed392",
  "10^2": "d942fdb1d9c3274a257154ef2f6f66161ea5872163dbb8daa40c7496e5365242",
  "10^3": "ba0acaf18a6a966a3eecbb791e9e22bc45d3a1183ff47342ab9cbde4635a828c",
  "10^4": "f32da5b457d45e0e6113d744fff316a1882f77fbf6ef5f92456faf84dfc8bd02"
}
```

The disclosure of the commitment to the partition 13699 using the format `["salt", "key", <value>]`.

```
["TpPrKdZ73ZR7JoUU-FCiTYvlQ4-QQ5ab9V2Z-cXze8E", "0",
["927eb07e71c648f73bec94e03d29cb41a0efc4f247a999d49f1318e3e8afbb84",
"b4b2a297499d63dd1ae5ee64c1aa21667b43b8974be3b3e17273005951413a56",
"854983f72c56c0102cac32edcce8b7c52365edc793cdba37d5603221b21d0a95",
"040be38408070da03bd6ca9e63999fac072adc20e1ba6f4513861db317a82a54",
"ad1a9492c27be7d33c7d00e33b0ca223e02a07440394b4036ded6f1f2c990c7a"]]
```

The base64url encoded SHA256 digest included in the `_sd`:

```
"zDHz3CX-akEjrDddMc8RYemeUCmEN0yjT1JIM_KXJd4"
```

NOTE 1: The user is required to only disclose the particular partition it uses to generate the inequality proof.

NOTE 2: The issuer can combine the disclosure digests into a single value using an accumulator or by concatenating the disclosure digests and the decoys. Implementation specific profiles are required.

The user, given a threshold value *t*, is required to select the partition that can generate the hash chains required for the inequality $x \geq t$. The user sends the disclosure of the commitment required for the inequality test, and the threshold values for each digit. The verifier can compute the hash chain using the threshold value for each digit and compares the root hash with the issuer signed commitments in the SD-JWT or MSO. If the signature is verified, the verifier accepts the inequality test.

B.1.4 Less than or equal to and range proofs

Any range proof, $a \leq x \leq b$, can be constructed using two inequality tests, one proving the inequality at the lower bound and the other at the upper bound. The above demonstrates an inequality test of type $a \leq x$. To generate a less than or equal to $x \leq b$ proof, it is necessary to extend the above described approach. Using whole number *K*, the issuer can generate a commitment to the inequality $K - x \geq K - b$. Both inequality tests rely solely on hash digests and combined they can generate any valid range proof using issuer signed commitments.

EXAMPLE:

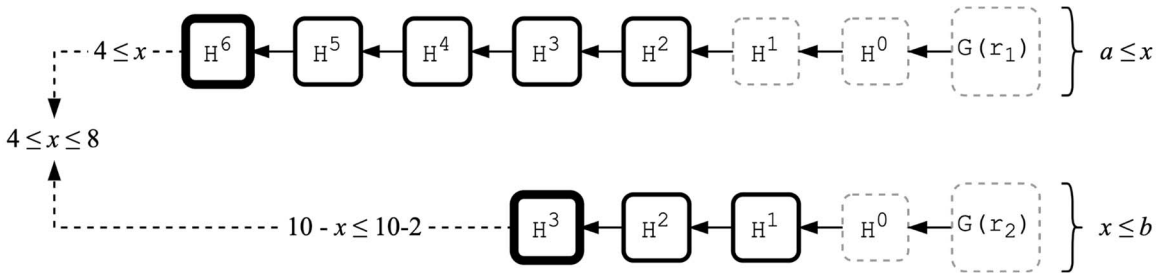


Figure B.5: Hash chain based range proof

Figure B.5 illustrates a hash chain based range proof for the range $4 \leq x \leq 8$. The issuer signs the bold commitments to both the lower bound test $4 \leq x$ and the upper bound test $x \leq 8$. The user presents both inequality tests to the verifier. The verifier combines the two proofs for inequality tests into range proof and accepts the range proof if the issuer's signature over the commitments is valid.

NOTE 1: For a range proof, the issuer is required to sign the parameter *K* used for the inequality $K - x \geq K - b$.

NOTE 2: The attestation issuance date impacts the proof that the user generates. A user generates a proof on an inequality test not for the request threshold, *t*, but subtracts the difference between the issuance date and the presentation date. A similar logic applies for age under or equal to proofs, as well as for range proofs.

HashWires represent an efficient way to generate inequality tests and range proofs using only SHA256. Running 70 000 loops on a dual core 2,2 GHz processor, it takes 72 µs ± 5,58 µs to generate the commitment for a 3 digit inequality test, and 156 µs ± 31,7 µs for a 6 digit one. The proof size is constant and the verification is faster than the generation.

B.2 Hash chain code example

This annex contains a Python code example of how to use hash chains to calculate a predicate of a user's age.

```
import secrets
from hashlib import sha256

# Get the user's age
while True:
  try:
    age = int(float(input("Enter your age: ")))
    if age < 0:
      raise ValueError
    break
  except ValueError:
    print("Enter a non negative number.")

# The issuer generates a seed and the commitment the user will need.
seed = secrets.token_bytes()
```

```
commitment = sha256(seed)
hash_chain = [commitment.hexdigest().encode('ascii')]

# The issuer then generates the hash chain.
for i in range(age):
  commitment = sha256(commitment.hexdigest().encode('ascii'))
  hash_chain.append(commitment.hexdigest().encode('ascii'))

# The hash chain is reversed so that the index values equal age
hash_chain.reverse()

# The issuer includes the following claim in the signed attestation
age_is_zero = hash_chain[0]

# The verifier wants a proof for age_over_n
n = 10
age_proof = None

# The user has to generate the following age proof
assert isinstance(n, int) and n >= 0, "The value is a non-negative integer."
try:
  age_proof = hash_chain[n] if n != 0 else age_is_zero
  print(f"The proof value is: {age_proof}")
  print(f"Copy this value for the next cell's input prompt: {age_proof.decode('ascii')}")
except IndexError:
  print(f"The user does not have a long enough hash chain for the required age proof of {n}")

# The user sends the age proof to the verifier, who verifies the chain length
age_proof_test = input("Copy paste the provided value from the previous cell: ")
age_proof_test = age_proof_test.encode('ascii')

above_n = False
if n == 0 and age_proof_test == age_is_zero:
  above_n = True
else:
  for i in range(n):
    age_proof_test = sha256(age_proof_test).hexdigest().encode('ascii')
  above_n = True if age_proof_test == age_is_zero else False

print(f"The user provided valid proof for the age is equal to or greater than {n} test: {above_n}")
```

B.3 HashWires for SD-JWT and MSO

Code examples in Python and descriptions on how to use HashWires for inequality tests for SD-JWT and MSO have been provided by Peter Lee Altmann at the repository [i.6].

Annex C:
Post-quantum safe zero-knowledge proofs and anonymous credentials

C.1 General

This annex describes research and innovations of new types of ZKP schemes. These types of innovative ZKP schemes are still being researched at an academic level and are not yet standardized, so they cannot be considered for the EUDI Wallet at the time of writing (August 2025). Nevertheless, the research on ZKP schemes is described in this annex since they may be implemented and standardized, which could be of interest for future standardization of the EUDI Wallet.

C.2 Quantum physics applied on ZKP schemes

C.2.1 Background

The advent of quantum computers is typically considered a disruption for classic cryptography. In 1994 Peter Shor published the paper "Algorithms for quantum computation: discrete logarithms and factoring algorithm" [i.235] that described how quantum computers can use certain algorithms for finding discrete logarithms and factoring integers. As a consequence, classic asymmetric cryptographic algorithms such as RSA and ECDSA, which are based on the discrete logarithm problem, are vulnerable against quantum computing attacks in a post-quantum world.

One countermeasure is to employ Quantum-Safe Cryptography (QSC) algorithms, i.e. cryptographic algorithms (typically public-key algorithms) that are expected to be secure against a cryptanalytic attack by quantum computers. NIST conducts a research program [i.210] to identify candidates for QSC algorithms that can be standardized. The signature scheme finalists (December 2023) are FALCON [i.75], FIPS 204 [i.207] (based on CRYSTALS Dilithium [i.75]) and FIPS 205 [i.208] (based on SPHINCS+ [i.238]).

Furthermore, Dutto et al. has published the paper "Toward a Post-Quantum Zero-Knowledge Verifiable Credential System for Self-Sovereign Identity" [i.83], which analyses quantum-safe variants of BBS+ and CL-signatures based on a lattice-based scheme. The paper also identifies the open issues for achieving VCs suitable for selective disclosure, non-interactive renewal mechanisms, and efficient revocation.

NOTE: The countermeasures above describe lattice-based or hash-based algorithms that are executed in classic computers with the intention to protect against quantum computing attacks with Shor's algorithm, but the QSC algorithms per se are not designed for quantum computers.

On the contrary to quantum computing attacks on classic cryptography, quantum physics and quantum computers can be used as an advantage when designing cryptographic protocols for a post-quantum world. There exist Quantum Key Distribution (QKD) protocols and quantum-based ZKP schemes, which are described in the following clauses.

C.2.2 Quantum Key Distribution (QKD)

The most mature quantum cryptographic application is Quantum Key Distribution (QKD), which utilizes quantum mechanics to share a random secret key with two parties, which then can be used to encrypt and decrypt messages. A unique property of quantum key distribution is the ability to detect if any third party has tried to eavesdrop on the communication channel between the two parties. The first QKD scheme was BB84 [i.24] that was invented by Charles Bennett and Gilles Brassard in 1984. BB84 is based on Heisenberg's uncertainty principle and uses the polarization state of photons to encode key bits, which means that the quantum data encoded as photons cannot be copied or measured without disturbing the key exchange protocol. There exist several commercial products that implement QKD schemes, which can be used for example to share symmetric AES keys. A tutorial on QKD with more information on this subject is published by IEEE [i.274].

C.2.3 Quantum physics applied to the graph 3-colouring ZKP scheme

The graph 3-colouring ring (G3C) problem is a classic problem that was introduced already in 1856. The graph 3-colouring problem takes as input a graph (G) and decides whether it can be coloured using only three (3) colours, such that no two adjacent vertices (nodes) have the same colour. The graph 3-colour problem is proven to be NP-complete.

The graph 3-colouring problem can be used as a ZKP scheme as described below.

Let G be a graph with n vertices and define the set of vertices as $V = \{v_1, ..., v_n\}$. Also define the set of edges as $E = \{e_{i,j}\}$. where $e_{i,j}$ is the edge between vertices v_i and v_j. The graph G is known to both parties. The prover's private knowledge is the 3-colouring of the graph G. whilst the verifier only knows the graph shape (with black "hidden" colours). The protocol is executed as follows:

1) Prover: Randomly permute the 3-colours of graph G. Commit to the permutation of the colours of all vertices, such that $c_i = P(v_i, \textit{colour of } v_i)$.

2) Prover: Share the graph G (with black "hidden") colours to the verifier.

3) Verifier: Select edge $e_{i,j}$ and send $e_{i,j}$ to the prover.

4) Prover: Open c_i and c_j.

5) Verifier: Accept if $c_i \neq c_j$, else reject.

The protocol is illustrated with Figures C.1 and C.2.

In step 1, the prover permutes the colours of a graph G as illustrated in the figure below. Two permutations are shown in Figure C.1, and the prover commits to permutation P_2 in this example.

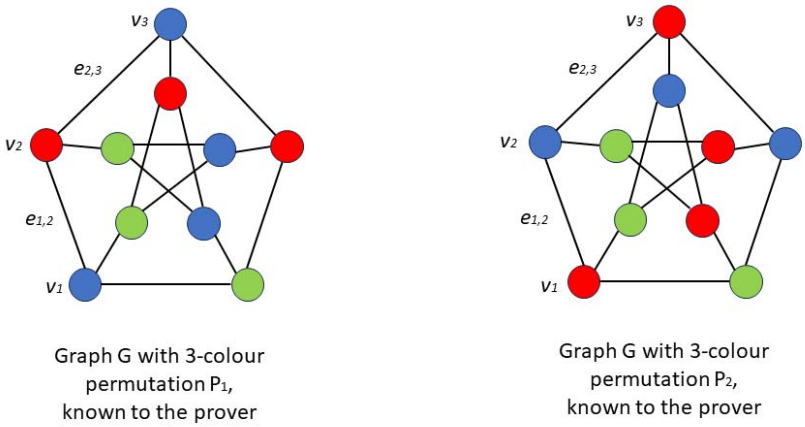


Figure C.1: Examples of 3-coloured graphs

The prover shares the graph G (with hidden colours) with the verifier, as shown to the left in Figure C.2. The verifier selects edge $e_{1,2}$ whereupon the prover opens vertices v_1 and v_2. Since v_1 is red and v_2 is blue, i.e. the colours are different, the verifier can accept the proof.

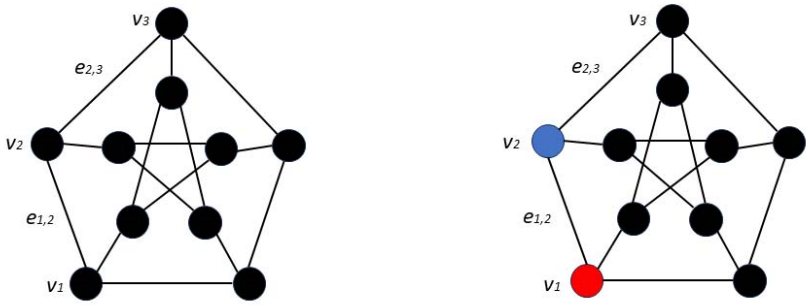


Figure C.2: Example of 3-coloured graph ZKP

Hence, the prover's knowledge is the 3-colouring permutation of the graph, and can prove this for each edge of the graph to the verifier. The prover's zero-knowledge proofs are the vertices that are opened to the verifier.

A formal description of the graph 3-colouring ZKP scheme is described as Zero-Knowledge Protocol for Graph Isomorphism in the paper "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems" [i.119] published in 1991 by Goldreich et al.

The classic graph 3-colouring ZKP scheme can be transposed to the quantum world. Simply put, large entangled quantum states are utilized for a graph in a quantum computer, equivalent to how the colour permutations are computed on a graph in a classic computer. The quantum graphs may also be shared between the prover and verifier by using the quantum key distribution as described in the previous clause. The paper "Experimental relativistic zero-knowledge proofs" [i.4] describes how the graph 3-colouring ZKP can be implemented in a way that is theoretically quantum computing safe:

- The quantum cryptography behind the graph 3-colouring ZKP schemes goes beyond the scope of the present document. For further reading the following research papers are recommended: "Zero-knowledge against quantum attacks" [i.251] by Watrous, "Post-quantum Efficient Proof for Graph 3-Coloring Problem" [i.87] by Ebrahimi, and "Zero-knowledge proof systems for QMA" [i.35] by Broadbent et al.

C.2.4 ZKP using the quantum Internet (based on Schnorr's algorithm)

Another quantum ZKP scheme is based on Schnorr's algorithm on non-interactive zero-knowledge proof [i.168].

Assume that the prover wants to prove that it knows the secret value x such that $Y = g^x \bmod p$, for prime p and generator g, with g, p, and Y public. Schnorr's algorithm can then be performed as follows:

1) The prover chooses the value r and calculates $t = g^r \bmod p$. The prover sends value t to the verifier.

2) The verifier sends the random value c to the prover.

3) The prover calculates $s = r + cx$, and sends the value s to the verifier.

4) The verifier checks that $g^s \equiv t \times Y^c \bmod p$.

Schnorr's algorithm can be proven as follows:

$$t \times Y^c \equiv g^r \times (g^x)^c \bmod p$$

$$\equiv g^{(r+cx)} \bmod p$$

$$\equiv g^s \bmod p$$

Carney [i.51] has described how to replace the use of the generator g in Schnorr's scheme for a quantum mechanical qubit rotation, and how to perform zero-knowledge proofs using quantum algorithms over the quantum Internet. The applied quantum cryptography goes beyond the scope of the present document, but for further reading the paper "On Zero-Knowledge Proofs over the Quantum Internet" [i.51] is recommended.

C.2.5 Conclusions on quantum ZKP schemes

Quantum cryptography takes advantage of quantum computers to design new cryptographic protocols for a post-quantum world.

The Quantum Key Distribution (QKD) schemes are rather mature and are implemented in several commercial products. Hence, the QKD schemes may be used for sharing keys between two parties using classic ZKP schemes.

Several quantum cryptographic algorithms for use with ZKP are also being developed. The classic graph 3-colouring scheme and Schnorr's algorithm have been transposed into quantum cryptographic algorithms. There are also relativistic quantum ZKP protocols [i.4] with promising applications for identification tasks and blockchain applications such as cryptocurrencies or smart contracts.

The quantum ZKP schemes are still being researched at an academic level and are not yet standardized, so they cannot be considered for the EUDI Wallet yet. It is however worthwhile to monitor the research and development of quantum ZKP schemes: if the quantum ZKP schemes get standardized and implemented in commercial products they could be considered for a future revision of the eIDAS regulation.

Annex D:
EUDI Wallet used with ISO mDL flows

D.1 EUDI Wallet used with ISO mDL device retrieval flow

D.1.1 Overview of the ISO mDL device retrieval flow

The scope of the present clause is to describe how the EUDI Wallet can present ISO mDL selectively disclosed elements over the ISO mDL device retrieval flow, and how eIDAS2 trust services can be used to support this process.

NOTE: The ISO mDL device retrieval flow is mandatory for the EUDI Wallet according to the ARF [i.71].

The ISO mDL device retrieval flow is described in ISO/IEC 18013-5 [i.181], sections 6.3.2, 6.3.2.1 (as flow 1) and 6.3.2.4. The present clause will not repeat the entire ISO mDL device retrieval process, although a brief summary is provided below for readability with references to the ISO/IEC 18013-5 [i.181].

The ISO mDL device retrieval flow is illustrated in Figure D.1.

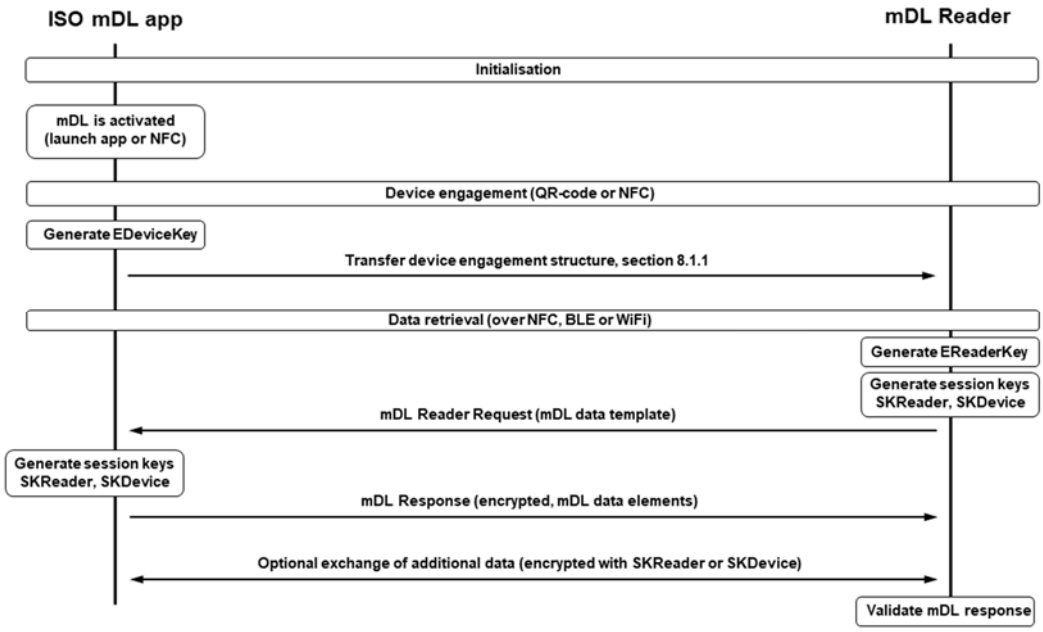


Figure D.1: Overview of the ISO mDL device retrieval flow

On a high level, the ISO mDL device retrieval flow can be divided in the following phases, where the ISO mDL reader is equivalent to an attended eIDAS2 relying party:

- Initialization phase, whereby the ISO mDL app is activated either by the user or triggered by NFC contact with the ISO mDL reader (see ISO/IEC 18013-5 [i.181], section 6.3.2.2 for more information).
- Device engagement phase, whereby the ephemeral device key EDeviceKey is generated, and the device engagement structure is transferred over NFC or as QR-code. The device engagement structure contains parameters for device retrieval transfer options TransferMethod and TransferOptions (see ISO/IEC 18013-5 [i.181], sections 6.3.2.3, 9.1.1, 8.2.1, 8.2.2 and 8.2.1.1 for more information).
- Data retrieval phase, whereby the EReaderKey, SKReader and SKDevice keys are generated to establish an encryption session. The ISO mDL reader then transmits the mDL Reader Request and the ISO mDL replies with the mDL Response (see ISO/IEC 18013-5 [i.181], sections 9.1, 9.1.1, 8.3.2.1.2 and 8.3.2.2.2 for more information).

As regards to selective disclosure, the mDL Reader Request contains a list of the DataElements the mDL Reader requests from the mDL app. Upon the user's consent, the mDL app will reply with the mDL Response with the selected DataElements in the DeviceSignedItems. The DeviceSignedItems object is signed by the mDL Authentication Key, to which the user is authenticated with a PIN-code or biometrics (see ISO/IEC 18013-5 [i.181], sections 8.3.2.1.2 and 8.3.2.2.2 for more information).

The selected DataElements will be hashed at the mDL reader, and be compared with the corresponding hash values in the MSO. ISO/IEC 18013-5 [i.181], section 9.1.2.3 describes how the relying party validates the MSO signature and how to check that the hashed mDL mdoc elements match the hash values in the MSO.

More specifically, ISO/IEC 18013-5 [i.181], section 9.1.2.3 specifies in detail how the mDL reader validates the certificate chain of the IACA trust anchor and the Issuing Authority's MSO signer certificate. ISO/IEC 18013-5 [i.181], Annex C describes the ISO mDL VICAL, which points to the IACA trust anchor and revocation information.

D.1.2 Analysis of the ISO mDL device retrieval flow for eIDAS2

An analysis of the ISO mDL device retrieval flow applied to an eIDAS2 context results in the following observations and recommendations:

- The ISO mDL app should be part of an EUDI Wallet.
- The ISO mDL Issuing Authority corresponds to a QTSP, PIDP and/or an EUDI Wallet provider.
- The mDL Reader corresponds to an device retrieval eIDAS2 relying party (that will validate the ISO mDL as an (Q)EAA/PID).
- The recommendations should be observed in clause 7.2.1 on how a QTSP/PIDP supervised under eIDAS2 can operate as an ISO mDL IACA.
- The recommendations should be observed in clause 7.2.1 on how an eIDAS2 EU TL should be formatted to be compatible as an ISO mDL VICAL or vice versa.
- The eIDAS2 relying party should use the eIDAS2 EU TL (which is equivalent to an ISO mDL VICAL) to retrieve the QTSP/PIDP trust anchor (which is equivalent to the IACA trust anchor).
- The eIDAS2 relying party should validate the MSO (submitted by the ISO mDL app in the mDL Response) according to the principles in ISO/IEC 18013-5 [i.181], section 9.1.2.3, by using the QTSP/PIDP trust anchor.
- The MSOs in the EUDI Wallet ISO mDL app should be unique as described in clause 7.2.1 to cater for verifier unlinkability when validated by the relying party.

NOTE 1: ISO mDL MSO does not enable unlinkability; it only enables selective disclosure.

NOTE 2: While issuer unlinkability is impossible to achieve, verifier unlinkability can be achieved by having the QTSP/PIDP issue batches of MSOs, each with unique salts, signatures, and DeviceKey elements. This will require an operational procedure of issuing multiple MSOs to each device on a regular basis, which may result in an additional operational cost for the QTSP/PIDP. Operational costs may be lessened by relying on a HDK function as described in clause 4.3.4.2 whereby the issuer only needs to keep track of a single DeviceKey element and use it to derive unique per MSO DeviceKey elements that the user can derive the corresponding private key for.

- The MSO is signed by the QTSP/PIDP with a COSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.237] and for QSC for future use [i.149].

These observations and recommendations should be considered with respect to selective disclosure for ETSI TS 119 462 [i.95], ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97].

D.2 EUDI Wallet used with ISO mDL server retrieval flow

D.2.1 Overview of the ISO mDL server retrieval flows

The scope of the present clause is to describe how the EUDI Wallet can present ISO mDL selectively disclosed elements over the ISO mDL server retrieval flow, and how eIDAS2 trust services can be used to support this process.

NOTE: This ISO mDL server retrieval flow is NOT mentioned by the ARF, but may need to be used by national or specific implementations that need to be interoperable with ISO mDL.

The ISO mDL server retrieval flow can be initialized as a hybrid device/server process (see clause D.2.2) or as a server process (see clause D.2.3). Once the ISO mDL server retrieval flow has been initialized, it continues with either the WebAPI (see clause D.2.5) or the OpenID Connect (OIDC) flow (see clause D.2.7). Clause D.2 will not repeat the entire ISO mDL server retrieval process, although a brief summary is provided below for readability with references to ISO/IEC 18013-5 [i.181].

D.2.2 ISO mDL flow initialization

The initialization of the ISO mDL device and server retrieval flows are described in ISO/IEC 18013-5 [i.181], sections 6.3.2, 6.3.2.1 (as flow 2) and 6.3.2.4.

The ISO mDL device/server data retrieval flow is illustrated in Figure D.2.

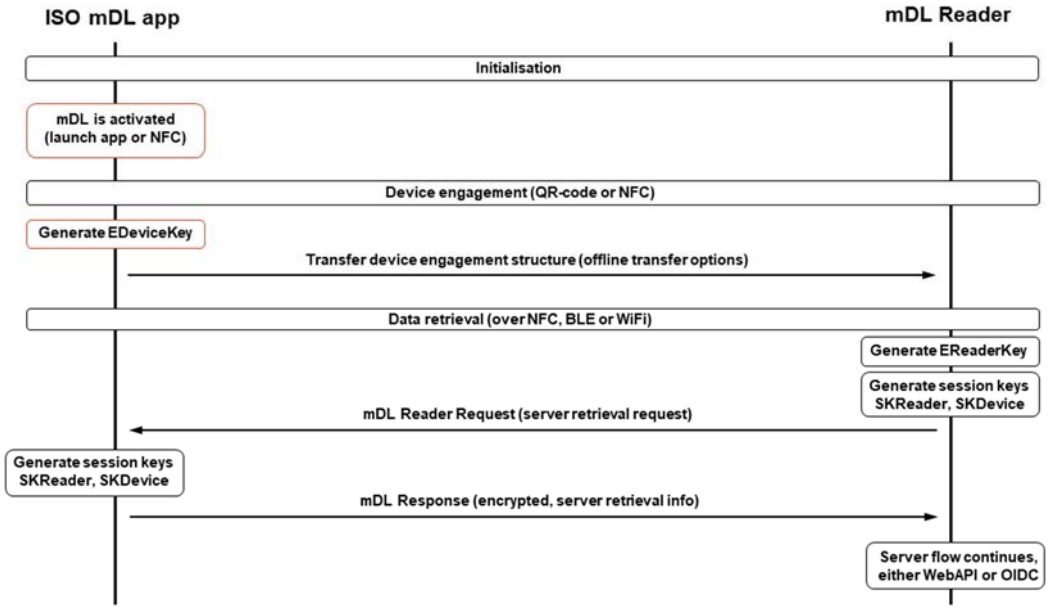


Figure D.2: ISO mDL flow initialization

On a high level, the ISO mDL device/server retrieval flow can be divided in the following phases (where the ISO mDL reader is equivalent to an eIDAS2 relying party):

- Initialization phase, whereby the ISO mDL app is activated either by the user or triggered by NFC contact with the ISO mDL reader (see ISO/IEC 18013-5 [i.181], section 6.3.2.2 for more information).
- Device engagement phase, whereby the ephemeral device key EDeviceKey is generated, and the device engagement structure is transferred over NFC or as QR-code (see ISO/IEC 18013-5 [i.181], sections 6.3.2.3, 9.1.1, 8.2.1 and 8.2.2 for more information).
- Data retrieval phase, whereby the EReaderKey, SKReader and SKDevice keys are generated to establish an encryption session. The ISO mDL reader then transmits the mDL Reader Request including the server retrieval request and the ISO mDL replies with the mDL Response including the server retrieval information (see ISO/IEC 18013-5 [i.181], sections 9.1, 9.1.1, 8.3.2.1.2.1 and 8.3.2.1.2.2 for more information).

The ISO mDL online data retrieval flow continues with either the WebAPI (see clause D.2.5) or OIDC (see clause D.2.7).

D.2.3 ISO mDL server retrieval flow initialization

The ISO mDL server retrieval flow initialization is described in ISO/IEC 18013-5 [i.181], sections 6.3.2 and 6.3.2.1 (as flow 3) and 6.3.2.4.

The ISO mDL server retrieval flow initialization is illustrated in Figure D.3.

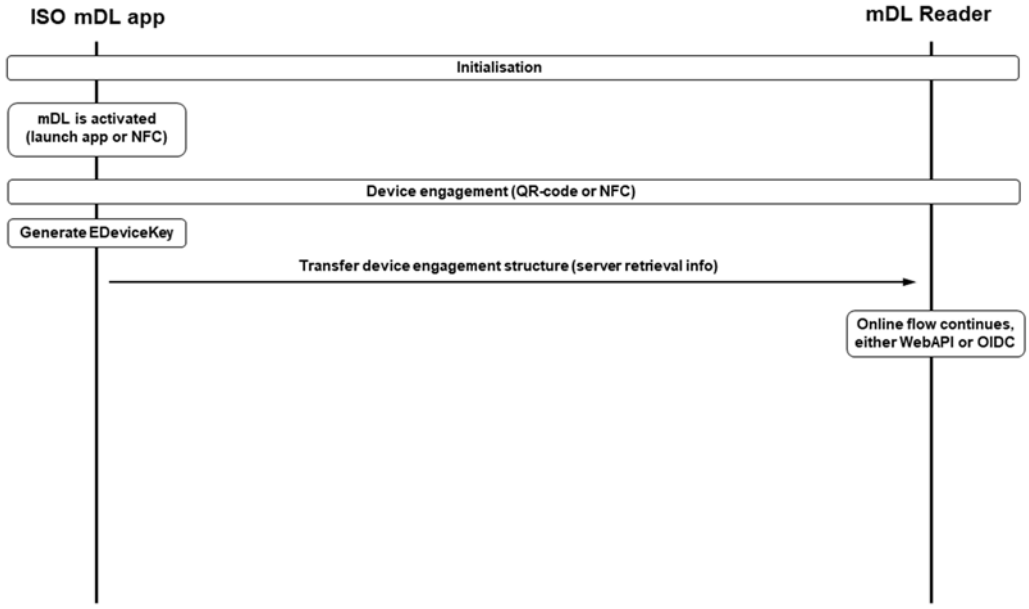


Figure D.3: ISO mDL server retrieval flow initialization

On a high level, the ISO mDL server retrieval flow can be divided in the following phases (where the ISO mDL reader is equivalent to an eIDAS2 relying party):

- Initialization phase, whereby the ISO mDL app is activated either by the user or triggered by NFC contact with the ISO mDL reader (see ISO/IEC 18013-5 [i.181], section 6.3.2.2 for more information).
- Device engagement phase, whereby the ephemeral device key EDeviceKey is generated, and the device engagement structure is transferred over NFC or as QR-code. The device engagement structure contains parameters for online transfer options WebAPI or OIDC (see ISO/IEC 18013-5 [i.181], sections 6.3.2.3, 9.1.1, 8.2.1, 8.2.2 and 8.2.1.1 for more information).

The ISO mDL server retrieval flow continues with either the WebAPI (see clause D.2.5) or OIDC (see clause D.2.7).

D.2.4 ISO mDL server retrieval WebAPI flow

The ISO mDL server retrieval flow is described in ISO/IEC 18013-5 [i.181], section 8.3.2.2 and the WebAPI calls are specified in ISO/IEC 18013-5 [i.181], section 8.3.2.2.2.

The ISO mDL WebAPI server retrieval flow is illustrated in Figure D.4.

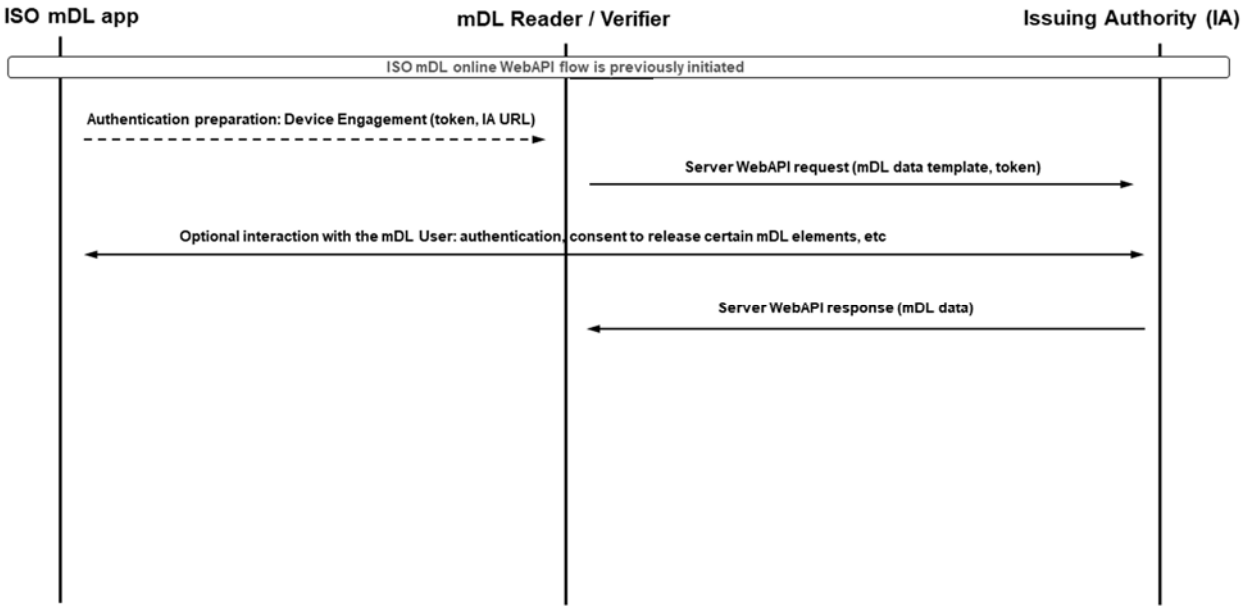


Figure D.4: ISO mDL server retrieval WebAPI flow

As regards to selective disclosure, the mDL Reader submits a server retrieval WebAPI Request with a list of requested DataElements to the Issuing Authority. Upon the user's consent, the Issuing Authority will reply with the mDL Response with the selected and disclosed DataElements (see ISO/IEC 18013-5 [i.181], section 8.3.2.2.2 for more information).

D.2.5 Analysis of the ISO mDL server retrieval WebAPI flow for eIDAS2

An analysis of the ISO mDL WebAPI server retrieval flow applied to an eIDAS2 context results in the following observations and recommendations:

- The ISO mDL app should be part of an EUDI Wallet.
- The ISO mDL Issuing Authority corresponds to a QTSP, PIDP and/or an EUDI Wallet provider.
- The mDL Reader corresponds to an eIDAS2 relying party, which will connect to the ISO mDL Issuing Authority over the WebAPI to request information about the user.

NOTE 1: eIDAS2 [i.103] Article 5a.14 states: "The provider of the European Digital Identity Wallet shall neither collect information about the use of the European Digital Identity Wallet which is not necessary for the provision of European Digital Identity Wallet services, nor combine person identification data or any other personal data stored or relating to the use of the European Digital Identity Wallet with personal data from any other services offered by that provider or from third-party services which are not necessary for the provision of European Digital Identity Wallet services, unless the user has expressly requested otherwise." If the ISO mDL Issuing Authority also has the role as an eIDAS2 European Digital Identity Wallet provider, the statement in eIDAS2 article 5a.14 may require additional privacy considerations when the server retrieval is used.

NOTE 2: eIDAS2 [i.103] Article 5a.16 states: "The technical framework of the European Digital Identity Wallet shall: (a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorized by the user". If the ISO mDL Issuing Authority also has the role as an eIDAS2 QTSP/PIDP, the statement in eIDAS2 article 5a.16(a) may imply that server retrieval is not possible unless explicitly approved by the user.

- The ISO mDL Issuing Authority may deploy QWACs in order to prove its authenticity over TLS to the connecting relying parties.

- The WebAPI token is a JWT that is signed by the ISO mDL Issuing Authority OIDC Authorization Server. The JWT signer certificate should be issued by an IACA, which in the eIDAS2 context is also a QTSP.
- The ISO mDL Reader, which is an eIDAS2 relying party, should use the ISO mDL VICAL (EU TL) to retrieve the IACA trust anchor (QTSP trust anchor).
- The WebAPI JWT is signed by the QTSP/PIDP with a JOSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.237] and for QSC for future use [i.149].

These observations and recommendations should be considered with respect to selective disclosure for ETSI TS 119 462 [i.95], ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97].

D.2.6 ISO mDL server retrieval OIDC flow

The ISO mDL server retrieval flow is described in ISO/IEC 18013-5 [i.181], clause 8.3.2.2 and the OIDC calls are specified in ISO/IEC 18013-5 [i.181], section 8.3.3.2.2.

The ISO mDL OIDC server retrieval flow is illustrated in Figure D.5.

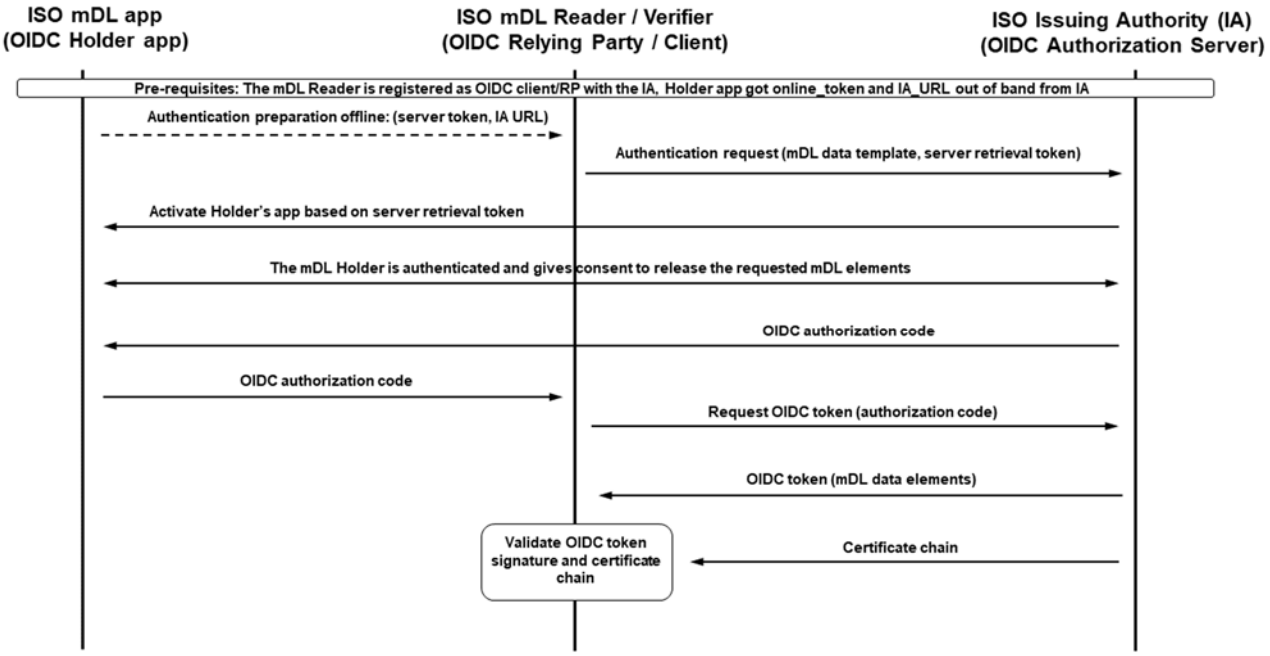


Figure D.5: ISO mDL server retrieval OIDC flow

As regards to selective disclosure, the mDL Reader (OIDC client) submits an server retrieval OIDC Request with the requested data elements (JWT claims) to the Issuing Authority, which operates an OIDC Authorization Server. This activates the OIDC authorization code flow [i.212]. Based on the user's consent, the Issuing Authority (OIDC Authorization Server) will reply to the mDL Reader (OIDC client) with the OIDC Token with the selected and disclosed JWT claims about the user (see ISO/IEC 18013-5 [i.181], section 8.3.3.2.2 and Annex D.4.2.2 for more information about the OIDC workflow).

D.2.7 Analysis of the ISO mDL OIDC server retrieval flow applied to eIDAS2

An analysis of the ISO mDL OIDC server retrieval flow applied to an eIDAS2 context results in the following observations and recommendations:

- The ISO mDL app should be part of an EUDI Wallet.
- The ISO mDL Issuing Authority corresponds to a QTSP, PIDP and/or an EUDI Wallet provider.

- The ISO mDL Issuing Authority operates an OIDC Authorization Server, which supports the OIDC authorization code flow.
- The mDL Reader corresponds to an eIDAS2 relying party, which is registered as an OIDC client to the ISO mDL Issuing Authority OIDC Authorization Server. The mDL Reader will connect to the ISO mDL Issuing Authority over OIDC to request information about the user.

NOTE 1: eIDAS2 [i.103] Article 5a.14 states: "The provider of the European Digital Identity Wallet shall neither collect information about the use of the European Digital Identity Wallet which is not necessary for the provision of European Digital Identity Wallet services, nor combine person identification data or any other personal data stored or relating to the use of the European Digital Identity Wallet with personal data from any other services offered by that provider or from third-party services which are not necessary for the provision of European Digital Identity Wallet services, unless the user has expressly requested otherwise." If the ISO mDL Issuing Authority also has the role as an eIDAS2 European Digital Identity Wallet provider, the statement in eIDAS2 article 5a.14 may require additional privacy considerations when the server retrieval is used.

NOTE 2: eIDAS2 [i.103] Article 5a.16 states: "The technical framework of the European Digital Identity Wallet shall: (a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorized by the user". If the ISO mDL Issuing Authority also has the role as an eIDAS2 QTSP/PIDP, the statement in eIDAS2 article 5a.16(a) may imply that server retrieval is not possible unless explicitly approved by the user.

- The ISO mDL Issuing Authority may deploy QWACs in order to prove its authenticity over TLS to the connecting relying parties.
- The OIDC Token is a JWT that is signed by the ISO mDL Issuing Authority OIDC Authorization Server. The JWT signer certificate should be issued by an IACA, which in the eIDAS2 context is also a QTSP.
- The ISO mDL Reader, which is an eIDAS2 relying party, should use the ISO mDL VICAL (EU TL) to retrieve the IACA trust anchor (QTSP trust anchor).
- The OIDC token JWT is signed by the QTSP/PIDP with a JOSE formatted signature, which allows for SOG-IS approved cryptographic algorithms [i.237] and for QSC for future use [i.149].

These observations and recommendations should be considered with respect to selective disclosure for ETSI TS 119 462 [i.95], ETSI TS 119 471 [i.96] and ETSI TS 119 472-1 [i.97].

D.3 EUDI Wallets used with ISO/IEC 18013-7 for unattended flow

D.3.1 Overview of the ISO/IEC 18013-7 flows

ISO/IEC CD 18013-7 [i.182] draft standard extends ISO/IEC 18013-5 [i.181] with the unattended flow, i.e. the server retrieval flow whereby an ISO mDL app connects directly to an mDL reader that is hosted as a web server application. ISO/IEC CD 18013-7 [i.182] is backward compatible with the protocols in ISO/IEC 18013-5 [i.181].

NOTE: Since the ISO mDL app connects directly to the web hosted mDL reader without involving any issuer, this flow preserves the user's privacy as required in eIDAS2 [i.103], Article 5a.16.

ISO/IEC CD 18013-7 [i.182] unattended flow is designed based on the following protocols:

- Device Retrieval from an ISO mDL app to a web server application over HTTPS POST; this flow is described in clause D.3.2.
- OpenID for Verifiable Presentations (OID4VP) [i.214] in conjunction with Self-issued OpenID Provider v2 (SIOP2) [i.216]; this flow is described in clause D.3.3.

D.3.2 ISO/IEC 18013-7 Device Retrieval flow

The general data retrieval architecture is described in ISO/IEC 18013-5 [i.181], section 6.3.2.4.
ISO/IEC CD 18013-7 [i.182] draft standard describes device retrieval of data for unattended (i.e. online web application) use cases. The ISO mDL app and the ISO mDL reader support device retrieval using the mDL request and response as specified in ISO/IEC 18013-5 [i.181], section 8.3.2.1.

ISO/IEC CD 18013-7 [i.182] adds Annex A that specifies the Reader Engagement phase, which takes place before the Device Engagement phase in ISO/IEC 18013-5 [i.181]. The Reader Engagement struct contains the parameter RetrievalOptions, which in turn includes the RestApiOptions that defines the URI and REST API parameters for the HTTPS connection to the web hosted mDL Reader.

ISO/IEC CD 18013-7 [i.182] unattended online retrieval flow is illustrated in Figure D.6.

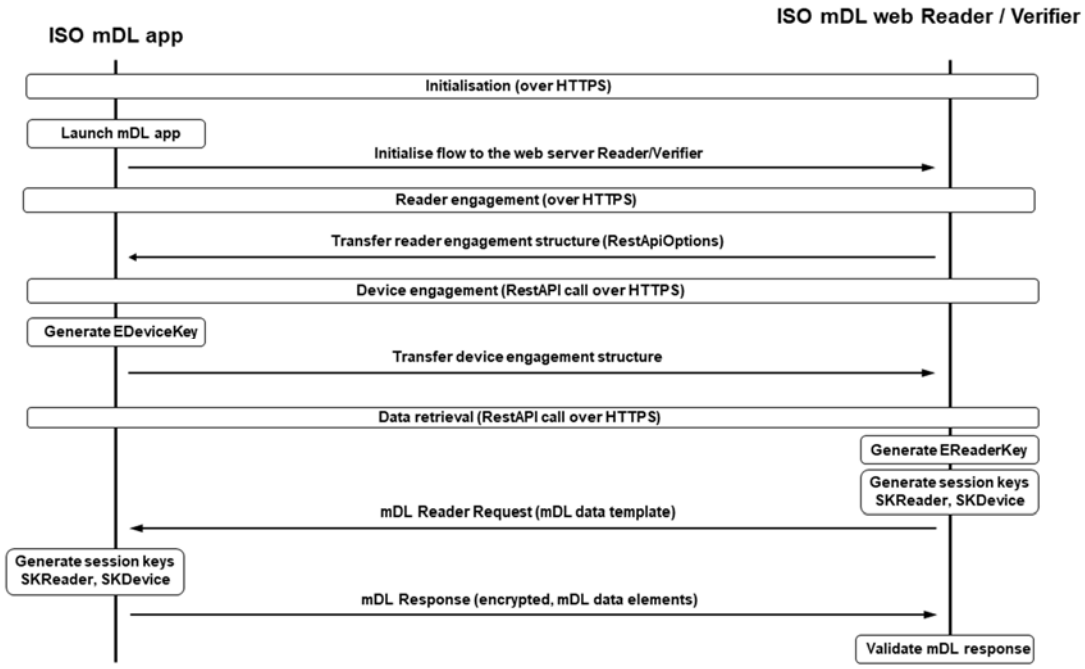


Figure D.6: ISO mDL unattended Device Retrieval flow

When the mDL Response has been retrieved and parsed by the ISO mDL reader/verifier, the mDL selected attributes and MSO are verified according to the same process as the ISO mDL device retrieval flow (clause 7.2.3).

As regards to selective disclosure for the ISO mDL unattended Device Retrieval flow, the same principles and recommendations apply as for the ISO mDL device retrieval flow (clause 7.2.3). However, the ISO/IEC CD 18013-7 [i.182] specification is not referred to by the ARF [i.71], although the associated specification ISO/IEC CD 23220-4 [i.187] is mentioned in the ARF.

D.3.3 ISO/IEC 18013-7 OID4VP/SIOP2 flow

As an alternative to the unattended Device Retrieval flow, ISO/IEC CD 18013-7 [i.182] specifies an unattended (online) flow based on OID4VP [i.214] with SIOP2 [i.216]. The OID4VP/SIOP2 flow is defined in Annex B of ISO/IEC CD 18013-7 [i.182]. Furthermore, the OID4VP/SIOP2 protocol is based on the ISO/IEC CD 23220-4 [i.187] profile for presentations of ISO mDL. Note that the present clause is about a mDL presentation with OID4P, see clause 6.7.2 for a general description of OID4VP.

ISO/IEC CD 18013-7 [i.182] unattended OID4VP/SIOP2 flow is illustrated in Figure D.7.

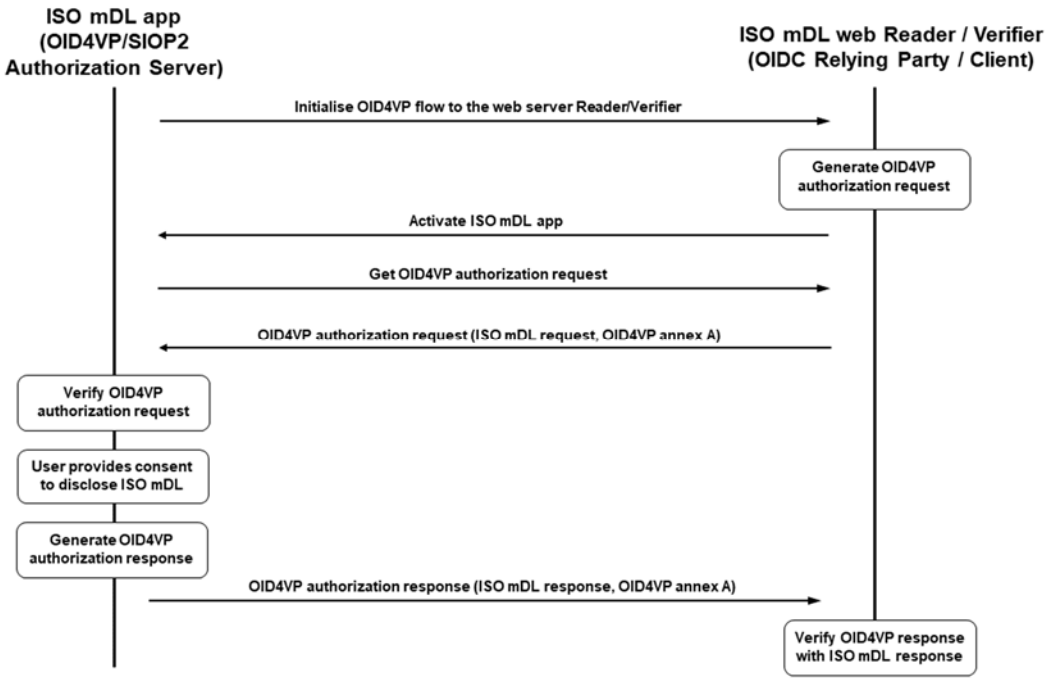


Figure D.7: ISO mDL unattended OID4VP/SIOP2 flow

When the OID4VP Response, which contains the mDL Response, has been retrieved and parsed by the ISO mDL reader/verifier, the mDL selected attributes and MSO are verified according to the same process as the ISO mDL device retrieval flow (clause 7.2.3).

As regards to selective disclosure for the ISO mDL unattended OID4VP/SIOP2 flow, the same principles and recommendations apply as for the ISO mDL device retrieval flow (clause 7.2.3). However, the ISO/IEC CD 18013-7 [i.182] specification is not referred to by the ARF [i.71], although the associated specification ISO/IEC CD 23220-4 [i.187] is mentioned in the ARF.

NOTE: ISO/IEC CD 23220-4 [i.187] is mentioned as a target in the ARF [i.71], but not mandatory since not yet published. If ISO/IEC CD 23220-4 [i.187] will include ISO/IEC 18013-5 [i.181] proximity as well as OID4VCI and OID4VP then 23220-4 is likely to be mandatory in a future version of the ARF.

Annex E:
A primer on W3C VCDM & SD-JWT VC

E.1 Overview of W3C Verifiable Credential Data Model (VCDM)

E.1.1 W3C VC, JSON-LD, data integrity proofs, and linked data signatures

The W3C Verifiable Credential Data Model (VCDM) is a way to express verifiable electronic attestation of attributes on the Web. At its core, a W3C Verifiable Credentials (VC) is a standardized digital format for presenting and exchanging verifiable claims (in essence statements expressed using subject-property-value relationships) about individuals, organizations, or things. These claims can be expressed as attributes in an electronic attestation of attributes. Specifically designed for the Web, the W3C VCDM aims to enable users to present attribute assertions from potentially different issuers and about potentially different identity subjects. These assertions can be organized into information graphs expressing subject-property-value relationships (e.g. Credential-type-DrivingLicense).

The W3C Verifiable Credentials Data Model (VCDM) is an open standard and is designed to be interoperable across different systems and platforms and to support a wide range of applications. The W3C VCDM v1.1 [i.264] describes a issuer-holder-verifier based model for digital "verifiable credentials" (defined as a "set of one or more claims made by an issuer" that are also "tamper-evident [with] authorship that can be cryptographically verified"). Specifically, the VCDM v1.1 aims to improve the ease of expressing digital credentials while also ensuring a high degree of privacy.

EXAMPLE: A trusted authority, such as a PID Provider, could construct a W3C VCDM compliant attestation containing the PID attributes and sign these with their private key. The user (assumed herein to be the identity subject of the VC) can then create a Verifiable Presentation (VP) using one or more VCs and present attributes to a verifier. The resulting W3C VC is verifiable to any verifier who has access to the required cryptographic keys. The proof mechanism could then support privacy features such as selective disclosure and/or unlinkable verifiable presentations.

The VCDM 1.1 text mandates that claims about a subject can be made tamper evident, that these claims are expressed in the form of subject-property-value relationships, and that it is possible to organize these claims into an information graph. However, it is not required that the claims or the proof is expressed as a graph in the attestation. To date, the VCDM 1.1 text has principally focused on JSON-LD type attestations. W3C VCDM Support for JSON only has been limited. The lack of JSON only support is problematic since the ARF prohibits the use of linked data proofs for the PID and only optionally supports JSON-LD. The ARF text mandates that the PID is issued as a JWT and that it is secured using SD-JWT.

After the publication of VCDM v1.1, the W3C VC WG has been working on VCDM 2.0 to make the standard more flexible and able to support multiple formats and signature algorithms. Work was ongoing to support the representation of verifiable claims in multiple ways including JSON, JSON-LD, or using any other data representation syntax capable of expressing the data model such as XML, YAML, or CBOR, as long as there is a mapping defined back to the base data model defined in the VCDM document (which relies on JSON-LD). This work was ongoing as several outstanding issues remained unsolved.

However, recently the W3C VC WG has argued strongly in favour of removing securing JSON and non linked data formats from the specification (see W3C VC WG issue #88 [i.260]). This means that the W3C VCDM is likely to evolve in a direction that will not address outstanding issues with the underspecified JSON sections, which includes key details such as how to do the required transformations or mappings. By extension, it is likely also that the proposed W3C work on how to secure a (W3C) VC using JSON [i.169] will be postponed until further notice. It is worth noting that the W3C VC WG charter does not specify specific media types, but that there does not exist a consensus with the WG to pursue JSON.

Regardless of the debate outcome, each VC and VP includes fields for specifying the signature schemes used to sign the claim or the presentation of a claim respectively (i.e. whether the verification of the proof is calculated against the data transmitted or against a transformation such as another data model or an information graph). Since the debate outcome is presently unknown, the text herein describes the solutions presently mentioned by VCDM v1.1, which are JSON Web Token and Data Integrity Proofs. Each will be described, with illustrations for possible solutions to still outstanding issues for the JWT based approach. The data integrity proofs will only be briefly explained to help readers understand why some of the ideological differences may make it difficult to secure a W3C VC using SD-JWT without a proper specification on how to secure a W3C VC using JSON.

Finally, the potential of relying only on SD-JWT VC for the attestation and use case specific mapping to VCDM 1.1 will be discussed as it represents the most suitable selective disclosure alternative considering the ongoing debates.

E.1.2 W3C VC, JSON-LD, data integrity proofs, and linked data signatures

There are many concepts surrounding the W3C VCDM v1.1, including JSON-LD, data integrity proofs, and linked signatures. The first, JSON-LD, will be explained in detail below, but it is helpful to explain how the other two relate to JSON-LD.

Data integrity proofs are defined by the W3C as "a set of attributes that represent a digital proof and the parameters required to verify it." Put differently, a data integrity proof provides information about the proof mechanism, parameters required to verify that proof, and the proof value itself. This information is provided using Linked Data vocabularies in a JSON-LD formatted attestation.

Linked data signatures are a proposed way to sign data expressed in linked data formats such as a JSON-LD. Linked data signatures sign the underlying information graph as opposed to the payload itself. More specifically, the graph is normalized into a byte stream that is signed. The corresponding verification can be of the graph of information, and not necessarily the syntax specific content itself meaning that the same digital signature would validate information expressed in multiple compatible syntaxes without necessitating syntax specific proofs (see W3C VC Data Integrity v1.0 where this idea is explored in detail).

To understand how a W3C VCDM v1.1 compliant attestation would look like, it is necessary to understand its core format, JSON-LD. Being similar to JSON, a key difference is that JSON-LD uses a property called `"@context"` to link attributes to descriptions that provide semantic clarity on how to unambiguously interpret each attribute. Each attribute is expressed in the form of subject-predicate-object triples that essentially describe an information graph.

Consider the following example of a JSON-LD document describing a person. The attributes `name` and `jobTitle` are mapped to concepts in the schema.org vocabulary as detailed in the `"@context"`.

```
{
  "@context": "http://schema.org/",
  "@id": "https://me.example.com",
  "@type": "Person",
  "name": "John Doe",
  "jobTitle": "ETSI TR editor"
}
```

The @context allows the JSON-LD to be mapped to an Resource Description Framework (RDF) model and thus an information graph. The information graph for the above looks as follows:

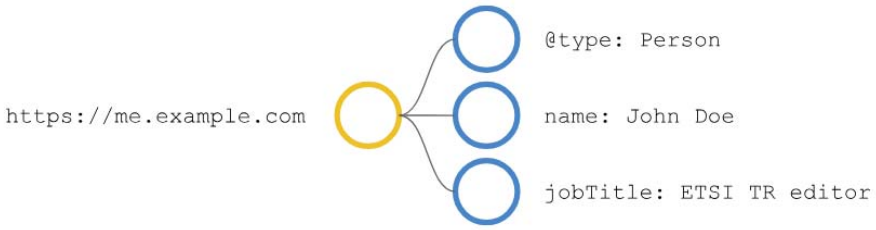


Figure E.1: Example of W3C VCDM v1.1 graph

And the W3C VCDM v1.1 graph triples are as follows:

Table E.1: Example of W3C VCDM v1.1 graph triples

Subject	Predicate	Object
`https://me.example.com`	`http://www.w3.org/1999/02/22-rdf-syntax-ns#type`	`http://schema.org/Person`
`https://me.example.com`	`http://schema.org/jobTitle`	`ETSI TR editor`
`https://me.example.com`	`http://schema.org/name`	`John/Jane Doe`

And the associated N-Quads (a syntax for RDF datasets) are:

1) <https://me.example.com> <http://schema.org/jobTitle> "ETSI TR editor".

2) <https://me.example.com> <http://schema.org/name> "John/Jane Doe".

3) <https://me.example.com> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/Person>.

The benefit with the above is that it does not matter what syntax is used to describe the underlying information graph as they would all describe the same model and thus enable a mapping to the exact same N-Quads.

NOTE: Since data integrity proofs sign the N-Quads containing triples as opposed to only the object, they do not fully support predicates that rely on the algebraic manipulations of the object. For instance, while it is possible to check for message equality, it is not possible to check whether one value is larger than another. Consequently, the signature scheme used to sign the N-Quads may support additional predicates than the N-Quads allow (e.g. a range proof may be supported by the signature scheme but the N-Quad may limit the predicate to an equality test).

To enable selective disclosure of a W3C VCDM v1.1 using data integrity proofs and linked data proofs, an issuer would need a proof mechanism that can logically order the N-Quads in such a way that the verifier knows that the presented attributes are properly paired. One way is to use the N-Quad message digests as leaf nodes to a Merkle tree and include the Merkle root in the attestation. Another, assuming that the issuer is comfortable with using JSON-LD and linked data proofs only, is to include N-Quad messages as selectively disclosable values in a SD-JWT `"_sd"` array (see clause 7.3.1.2 for a detailed description of how to generate a disclosure in [i.155] (IETF OAUTH: "Selective Disclosure for JWTs (SD-JWT)") and let the user present only the parts of the information graph that the verifier needs.

To date, the most well developed solution relies on the `bbs-2022` cryptosuite, which supports JSON-LD + data integrity proofs + linked data proof. Including triples in SD-JWT is not entirely straight forward and would require additional specification.

To conclude, JSON-LD is a way to express linked data and JSON-LD based attestations may include data integrity proofs that also rely on linked data for their verification. When also using linked data proofs, issuers can issue (Q)EAAs that are highly optimized for semantic interoperability. However, it is not entirely clear how selective disclosure and predicates would work in the context of PID/(Q)EAAs. Supporting crypto suites like `bbs-2022` are based on primitives that the public sector is unlikely to use since they are not considered as being plausible quantum safe. Solutions like SD-JWT can support linked data proofs but it is not entirely clear how they could be combined with data integrity proofs (and what the benefits would be) as SD-JWT was designed with JWT based attestations in mind.

Having described how W3C VCDM v1.1 compliant attestations can be secured using SD-JWT also for JSON-LD and linked data signatures, attention now turns to JWT based W3C VCs and SD-JWT.

E.1.3 JWT based W3C VC

One popular proof format that is actively used in several implementations is the JSON Web Token (IETF RFC 7519 [i.165]). A JWT encodes claims as a JSON object contained in a JSON Web Signature (JWS) (IETF RFC 7515 [i.163]) or JWE (IETF RFC 7516 [i.164]). A user could present a VP with the VC claims using JWT as described in example 32 of the W3C VC Data Model [i.264]. The decoded JWT contains the presentation as exemplified next.

```
{
   ...,
   "verifiableCredential":[
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImRpZDpleGFtcGxlOmFiZmUxM2...QGbg"
   ]
```

```
}

The VC contained within (highlighted above in yellow) contains the following information about the
identity subject.
{
   ...,
   "credentialSubject":{
      "degree":{
         "type":"BachelorDegree",
         "name":"<span lang='fr-CA'>Baccalauréat en musiques numériques</span>"
      }
   }
}
```

The VC contains the attribute in cleartext. Typically, a signed JWT containing identity data cannot support use cases where the JWT is issued once and then presented multiple times by the user who seeks to disclose only the attributes necessary for the service. In and of itself, the W3C VC standard only supports, but does not enforce, selective disclosure by design. The standard is flexible and supports multiple selective disclosure techniques. However, until recently these selective disclosure techniques have relied on multi-message signature schemes like `bbs-2022` suite.

NOTE: The text below assumes that there is a way to secure JSON for W3C VCDM v1.1 and ignores the ongoing debate on the topic within the W3C VC WG.

E.2 SD-JWT based attestations

E.2.1 General

To support selective disclosure in JWTs, Fett, Yasuda, and Campbell (2023) specify Selective Disclosure JSON Web Token (SD-JWT) in the Internet Engineering Task Force (IETF) draft document [i.155] entitled "Selective Disclosure for JWTs (SD-JWT)". At its core, an SD-JWT is a digitally signed JSON document that can contain salted attribute hashes that the user can selectively disclose using disclosures that are outside the SD-JWT document. This allows the user to share only those PID attributes that are strictly necessary for a particular service.

NOTE 1: SD-JWT is generally applicable to selective disclosure of JWTs that are not bound to the W3C VCDM v1.1. A W3C VCDM v1.1 contains sections that describe how a VC can be JSON encoded in a JWT and then protected using JWS/JWE. Correspondingly, the SD-JWT specifies how any JWT can support selective disclosure. But the joint utilization of the two is not straightforward.

NOTE 2: An SD-JWT supports selective disclosure solutions that require a clear logical ordering of data. It does not support algebraic manipulations of data.

Each SD-JWT contains a header, payload, and signature. The header contains metadata about the token including the type and the signing algorithm used. The signature is generated using the PID Provider's private key. The payload includes the proof object that enables the selective disclosure of attributes. Each disclosure contains a salt, a cleartext claim name, and a cleartext claim value. The issuer then computes the hash digest of each disclosure and includes each digest in the attestation it signs and issues.

Using the proof object and the user shared disclosures, the verifier can verify that the disclosed claims were part of the original attestation. To do so, the verifier first verifies the issuer's signature over the entire SD-JWT. The verifier then calculates the digest over the shared disclosures and checks that the digest is included in the signed SD-JWT. Since the SD-JWT includes only digests of disclosable attributes, the verifier can only learn about claim names and claim values that are disclosed by the user or that are included as clear-text claims. The verifier cannot learn about any other claim names or values as these are included in the SD-JWT as salted attribute digests.

The IETF SD-JWT draft specification 07 [i.155] of 2023-12-11 details the exact process of creating a disclosure in section 5.2. In essence, for each disclosable claim, the issuer generates and associates a random salt with each key value pair, and encodes the byte representation of these as base64url. An example of a disclosure is shown in Figure E.2.

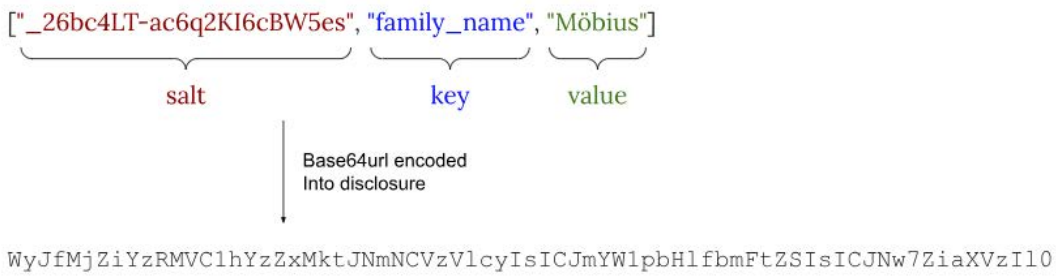


Figure E.2: Example of SD-JWT disclosure

Figure E.2 illustrates an example with the byte representation of the JSON-encoded array containing the salt, key, and value, is base64url-encoded into the disclosure.

NOTE: A linked data signature could be included in the `_sd` array but it is not entirely clear how to handle triples in the disclosure. One option could be to set the subject to the sub property in the attestation and to only include predicates in the disclosures as: `[<salt>, <predicate>, <object>]`.

To embed a disclosure in the SD-JWT, the issuer hashes each disclosure using a specified hash algorithm. The base64url encoded bytes of the digest, and not the disclosure, is then included in the SD-JWT as an array in the claim `_sd`, which includes only an array of strings, each being the digest of a disclosure or a random number (used to hide the original number of disclosures). This array is randomized so that the order of attribute disclosures is not always the same.

The SD-JWT specification supports selectively disclosable claims in both flat and more complex nested data structures. The issuer can therefore decide for each key individually, on each level of the JSON, whether or not the key should be selectively disclosable. The `_sd` claim is included in the SD-JWT at the same level as the original claim. Selectively disclosable claims can in turn include other objects with selectively disclosable claims.

Below, this text only exemplifies the flat and the nested data structure examples, but others are possible too.

Table E.2: Example of SD-JWT using a flat data structure

Contents	`["imQfGj1_M0El76kdvf7Daw", "address", {"street_address": "Schulstr. 12", "locality": "Schulpforta", "region": "Sachsen-Anhalt", "country": "DE"}]`
Disclosure	`WyJpbVFmR2oxX00wRWw3NmtkdmY3RGF3IiwgImFkZHJlc3MiLCB7InN0cmVldF9hZGRyZXNzIjogIlNjaHVsc3RyLiAxMiIsICJsb2NhbGl0eSI6ICJTY2h1bHBmb3J0YSIsICJyZWdpb24iOiAiU2FjaHNlbi1BbmhhbHQiLCAiY291bnRyeSI6ICJERSJ9XQ`
Digest	`FphFFpj1vtr0rpYK-14fickGKMg3zf1fIpJXxTK8PAE`
_sd value	`{ "_sd": [ "FphFFpj1vtr0rpYK-14fickGKMg3zf1fIpJXxTK8PAE" ], ..., "_sd_alg": "sha-256" }`

Table E.3: Example of nested SD-JWT with the sub-claim country in cleartext

Contents	["QSNIhu_n6a1rI8_2eNARCQ", "street_address", "Schulstr. 12"], ["QPkblxTnbSLL94I2fZIbHA", "locality", "Schulpforta"], ["jR-Yed08AEo4gcogpT5_UA", "region", "Sachsen-Anhalt"]
Disclosures	WyJRU05JaHVfbjZhMXJJOF8yZU5BUkNRIiwgInN0cmVldF9hZGRyZXNzIiwgIlNjaHVsc3RyLiAxMiJd, WyJRUGtibHhUbmJTTEw5NEkyZlpJYkhBIiwgImxvY2FsaXR5IiwgIlNjaHVscGZvcnRhIl0, WyJqUi1ZZWQwOEFFbzRnY29ncFQ1X1VBIiwgInJlZ2lvbiIsICJTYWNoc2VuLUFuaGFsdCJd
Digests	"G_FeM1D-U3tDJcHB7pwTNEElLal9FE9PUs0klHgeM1c", "KlG6HEM6XWbymEJDfyDY4klJkQQ9iTuNG0LQXnE9mQ0", "ffPGyxFBnNA1r60g2f796Hqq3dBGtaOogpnIBgRGdyY"

_sd value	{
	  "address": {
	  "_sd": [
	  "G_FeM1D-U3tDJcHB7pwTNEElLal9FE9PUs0klHgeM1c",
	  "KlG6HEM6XWbymEJDfyDY4klJkQQ9iTuNG0LQXnE9mQ0",
	  "ffPGyxFBnNA1r60g2f796Hqq3dBGtaOogpnIBgRGdyY"
	  ],
	  "country": "DE"
	  },
	  ...,
	  "_sd_alg": "sha-256"
	}

The QTSP/PIDP will have to send the raw claim values contained in the SD-JWT, together with the salts, to the EUDI Wallet user. The SD-JWT standard requires that data format for sending the SD-JWT and the disclosures to the EUDI Wallet user is a series of base64url-encoded values in what is called the Combined Format for Issuance, which looks like follows: `<JWT>~<Disclosure 1>~<Disclosure 2>~...~<Disclosure n>~<optional Holder Binding JWT>`. Note the separation of between the values using ~. The specific ways the ~ character should be used is defined under section 5 in the SD-JWT v.07 specification.

When the EUDI Wallet user receives the attestation from the QTSP/PIDP, the SD-JWT standard requires that the user verifies the disclosures. The user does so by extracting the disclosures and the SD-JWT from the Combined Format for Issuance, hashing each disclosure, and accepts the SD-JWT only if each resulting digest exists in the `_sd` array.

Relatedly, during presentation, the user sends the SD-JWT and the *n* disclosures to the verifier as a series of base64url encoded values in what is called the Combined Format for Presentation (also called SD-JWT+KB), which looks as follows: `<JWT>~<Disclosure 1>~<Disclosure 2>~...~<Disclosure n>~<optional Holder Binding JWT>`

The verifier checks that the issuer's signature is valid over the SD-JWT, that the disclosure digests are part of the SD-JWT, and if applicable that the Holder binding is valid (for specific steps see section 8 in the SD-JWT 07 specification).

Having described JSON secured W3C VCs and how SD-JWT can ensure selective disclosure of JWT based attestations, the text next discusses the potential joint utilization of both W3C VCs and SD-JWT, and why it is not as straightforward as it may appear.

E.2.2 SD-JWT VC

The IETF SD-JWT VC draft specification [i.143] provides a format that is optimized for the transport of the credential including the disclosures without further encoding. It is not designed to be embedded into any envelopes. It is arguably better to simply rely on JSON only claims for SD-JWT VC and recreate the W3C VCDM using a mapping algorithm. This option does not require the issuer to use linked data proofs (the ARF text does not allow the use of linked data proofs for the PID attestation), includes identity subject claims in an SD-JWT VC, and where a transformation is used to map the SD-JWT VC claims to a W3C VCDM 1.1 compliant information graph. Relying on SD-JWT VC and mapping would circumvent the aforementioned four difficulties and also adhere strictly to the design logic of a particular solution approach.

An example is provided next.

```
{
  "alg": "ES256",
  "typ": dc+sd-jwt,
  <other header info>
}
.
```

```
{
  "iss":"https://example.com/issuers/14",
  "nbf": 1262304000,
  "iat": 1262304000,
  "vct": "eu.europa.ec.eudiw.pid.se.1",
  "_sd":[
    "2cj...szs",
    "H03...iVY",
    "RKE...omY",
    "S7e...uDc"
   ],
   "_sd_alg":"sha-256"
}
```

Figure E.3: Example of a SD-JWT VC where W3C VCDM compliance relies on mapping

The example in Figure E.3 shows an SD-JWT VC secured attestation (not using JSON-LD) with the mandatory and disclosable PID attributes highlighted in blue. The `"_sd"` is here included as a root claim. This SD-JWT VC can be consumed, without prior processing, by any compliant SD-JWT VC library. Further evaluation can be done using standard JWT payload processing algorithms. In the example in Figure E.3.

- The JOSE header indicates the type.
- The claims in the credential are standard JWT claims. Applications can use predefined and established JWT claims from the "JWT Claims Registry", like `"sub"` for user identifiers. They can also use more complex claim structures such as those defined by OpenID Connect for Identity Assurance for providing information about provenance and level of assurance. This means existing JWT-based implementations can consume such VC payloads directly.
- The vct communicates to the verifier how to interpret any disclosed claim and there is no need for a separate `@context`.

A presentation is constructed using the combined format for presentation as defined in the SD-JWT specification.

NOTE 1: The present document recommends using the IETF October 23 2023 version of SD-JWT without Appendix A4 and A5 to understand the selective disclosure mechanism. Relatedly, to understand how to use SD-JWT VC as an attestation format, see the 2023-10-23 version of "SD-JWT-based Verifiable Credentials (SD-JWT VC)" [i.154].

NOTE 2: It should also be observed that SD-JWT VC is referenced by the OpenID4VC High Assurance Interoperability Profile (HAIP) [i.215], which is a profile of OpenID for Verifiable Credentials.

E.2.3 SD-JWT and multi-show unlinkable disclosures

Because every SD-JWT disclosure contains a unique salt, this unique salt acts as an identifier for the entire SD-JWT. Put differently, it is enough for a malicious issuer to receive a single disclosure from a colluding verifier for the issuer to uniquely identify the identity subject. Similarly, colluding verifiers could compare salt values to link together presentations from the same user (see clause 9.4 in the SD-JWT [i.155] specification for additional details).

While it is impossible to prevent issuers from identifying the user based on the unique salt in the salted attribute hashes approach, it is possible to enable multi-show verifier unlinkable disclosures even if verifiers collude or if a single curious verifier attempts to learn more about the user than what is disclosed in each presentation. To achieve complete multi-show unlinkability it is required that:

1) each SD-JWT VC contains only unique salts (even for the same claim); and
2) each SD-JWT VC is associated with a unique cryptographic key material used for device binding and/or holder binding (denoted as "holder binding key" in the context of SD-JWT).

Consequently, issuers are required to rely on batch issuance of SD-JWT to the EUDI Wallet if device retrieval functionality is desired (in an online scenario, the user can request a new SD-JWT on demand).

NOTE: To reduce the burden on issuers, it is possible to introduce a limit on the number of uses of each SD-JWT. The user's SD-JWTs would then be linkable in a portion of their presentations.

EXAMPLE: A user is given 10 PID attestations as SD-JWT VCs. The user presents the first 9 SD-JWT VCs once and the 10th twice. Out of the 11 presentations, two are linkable.

E.2.4 Predicates in SD-JWT

Similar to MSO, an SD-JWT was not designed to support predicates that can be dynamically computed (e.g. to compute an age over proof from the birth date). Here too, the recommendation is to use static claims with Boolean values such as `"age_over_NN": "True"`. However, as presented above in clause 4.3.6, it is possible to rely on issuer signed computational inputs and parameters to enable dynamic predicate support in SD-JWT.

E.3 W3C VCDM 2.0 with SD-JWT

There are currently two constructions that combine W3C VCDM or certain aspects of it and SD-JWT:

- W3C: Securing Verifiable Credentials using JOSE and COSE
- OpenID4VP: SD-JWT VCLD profile

The W3C specification "Securing Verifiable Credentials using JOSE and COSE" [i.261] defines how to secure credentials and presentations conforming to W3C VCDM 2.0 with JSON Object Signing and Encryption (JOSE), CBOR Object Signing and Encryption (COSE) and SD-JWT. This specification provides a proof mechanism that describes how to use SD-JWT to secure documents conforming to W3C CDM 2.0 using SD-JWT. The payload can be used as is without any kind of transformation algorithms. At the time of writing the present document (in August 2025), the SD-JWT section is marked as "Features related to [SD-JWT] are at risk and will be removed from the specification if the IETF standardization process occurs after this specification's timeline for reaching a Proposed Recommendation…", it is unclear if the SD-JWT section will remain in the final version of the specification.

SD-JWT VCLD is a profile defined in Appendix B.3.7 of OID4VP [i.214] that extends the SD-JWT VC credential format and allows for the incorporation of JSON-LD based payloads (such as W3C VCDM), but keeps some of the core mechanisms of SD-JWT VC. The core idea of this profile is to have sequential processing rules by first applying the SD-JWT VC processing rules and then the JSON-LD processing rules on the output of the SD-JWT VC processing. This construction aims to introduce a clear separation between the security relevant (SD-JWT VC) and the business-logic relevant (JSON-LD) parts by introducing a new claim "ld" that contains all JSON-LD payload. This allows for existing SD-JWT VC implementations to be extended with JSON-LD payloads in a clearly defined manner.

Annex F:
Business models and unlinkability

F.1 General

In a digital identity ecosystem it is often the case that the QTSP needs to invoice the relying party for the digital transactions it consumes.

EXAMPLE: An QTSP issues a Qualified Certificate to a user. The relying party is a bank with whom the user wants to sign a digital agreement. Hence, the user signs the digital agreement with a Qualified Electronic Signature by using its Qualified Certificate. Next, the relying party verifies the Qualified Electronic Signature and the corresponding Qualified Certificate. In order to check the status of the Qualified Certificate, the relying party sends an OCSP request to the QTSP. The QTSP counts the OCSP transactions from the relying party and can invoice the relying party accordingly.

The example above illustrates how QTSPs under eIDAS1 have been able to keep track of the usage of its issued Qualified Certificates and have been able to invoice the relying parties accordingly.

The legal conditions have however changed under eIDAS2, as article 5a.16 states:

"The technical framework of the European Digital Identity Wallet shall:

(a) not allow providers of electronic attestations of attributes or any other party, after the issuance of the attestation of attributes, to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorised by the user;"

More specifically, with full unlinkability it is not possible for the QTSPs to, on their own, keep track of how the (Q)EAAs are shared and with which relying parties.

This boils down to one question: How can the QTSPs invoice the relying parties without knowing how attestations are used and when?

The clauses below present various options of how to design a business model for QTSPs that operate under eIDAS2 with (Q)EAAs being shared with full unlinkability.

F.2 ETSI TR 119 479-2

ETSI ESI has released an early draft of ETSI TR 119 479-2 "EAA Extended Validation Services Framework and Application" [i.92]. The draft proposes a technical solution intended to enable QTSPs to invoice relying parties while claiming to preserve full unlinkability of (Q)EAAs/PIDs. This solution, termed "Cyphered VC Presentation", is further described in [i.92].

F.3 Anonymous usage data aggregation

F.3.1 General

To enable accurate billing and fair compensation for issuers in the EUDIW ecosystem, it is essential to collect data on attestation usage. At the same time, this collection has to uphold strict user privacy guarantees. While various anonymous data aggregation techniques exist, many rely on security assumptions or adversarial models that do not align with the EUDIW context, or they fail to scale efficiently and introduce complex flows that require protocol changes. The method outlined here is purpose-built to balance privacy, integrity, and performance, making it suitable for practical deployment at the required scale.

Accurate aggregate usage counts are required to support billing models, which may involve differentiated pricing based on attestation type, verifier, or a combination of both. Ideally, aggregation should rely on parties without incentives to misreport and where auditability or certification can replace complex cryptographic guarantees. Both issuers and verifiers have financial motives to distort data: issuers to inflate usage, and verifiers to downplay it. In contrast, users operate certified wallet devices that provide a trusted execution environment for computing usage data, which they have no incentive to falsify.

Users, however, require strong privacy guarantees. Usage data has to be collected in a way that preserves anonymity and prevents linkability. Output privacy is not essential, as billing data is shared only with the relevant parties, and there is no requirement to publish aggregate statistics.

One viable approach that achieves both scalability and privacy-preserving aggregation is a multiparty private sum protocol. This method provides strong privacy under minimal trust assumptions (requires a single honest server) and supports high performance. However, it only enables efficient and accurate aggregation when all participants behave honestly. Importantly, an accurate aggregation result does not guarantee that the underlying data is truthful; for instance, an issuer may manipulate EUDIW interactions to artificially inflate usage counts. Consequently, usage data aggregation, regardless of approach, is most suitable for deployments with certified wallet software, audited issuers, and regulated aggregators.

While it is technically feasible to limit the impact of malicious users (incl. issuers pretending to be users) and/or servers, the associated performance overhead - particularly in billing models with price differentiation by attestation type or service pair - can make the approach impractical at scale.

F.3.2 The billing model and private sum process

It is beneficial to list three different pricing structures as these impact performance:

- Flat-rate models require only aggregate usage counts per issuer and verifier. This significantly simplifies private sum computations as there is no longer a need to keep track of tuples.
- Type-based pricing necessitates tracking usage per issuer and possibly per attestation type (e.g. driver's licence, PID etc.). This makes the private sum computation slightly more cumbersome but is entirely manageable.
- Pair-specific pricing sets different prices per (issuer, verifier) pair. This should be minimized to the extent possible to avoid complexity and overhead.

Assuming a manageable amount of pair-specific pricing, the core mechanism of the private sum is additive secret sharing. Specifically:

1) The wallet logs a usage event as an (issuance_info, service_id) tuple and increments a local counter for that tuple.
2) When required, the wallet splits the counter into *N* random shares and sends each share to a distinct aggregation server, tagged with the corresponding tuple.
3) Aggregation servers independently collect shares across users, optionally applying tuple-specific pricing policies.
4) Aggregation servers then reveal the sums of the shares for each tuple and combine their results to compute the total usage counts.
5) Each aggregation server can then prepare billing information, and if required add further privacy enhancing measures to protect user privacy where required.
6) Each aggregation server can also compute aggregate statistics such as total count, most used service, average users per service etc.

User privacy is preserved as long as at least one server remains honest and each tuple is used by at least 100 users. If rare use of tuples presents a privacy concern, it is possible to add carefully selected decoy tuples with count 0 to enable deniability. Accurate totals require both correct user-submitted shares and honest behaviour by all aggregation servers (e.g. a regulated clearing house, the wallet provider, or member state appointed actor). Ensuring correct user reports can be achieved with relatively low overhead, but defending against malicious servers entails a significant performance cost.

To illustrate the core of the private sum protocol, consider three users who wish to compute the sum of their local counts using three trusted aggregation servers, under a prime modulus *p*.

Each user performs the following steps:

1) Let the user's private input be $u \in [0, p-1]$.

2) Generate two random values $(s_1, s_2) \in [0, p-1]$.

3) Compute the third share as: $s_3 = (u - s_1 - s_2) \bmod p$.

4) Sends each of the three shares, (s_1, s_2, s_3), to a distinct aggregation server.

The remaining users repeat the same process with their respective inputs. Each server receives one share from each user and sums them locally. Once all servers broadcast their local sums to each other, the final result is computed by summing the three totals modulo *p*.

This final sum is correct because the random values cancel out, leaving only the true sum of the original user inputs. At the same time, privacy is preserved: each server sees only one randomized share per user and cannot infer individual input values from the aggregated data.

F.3.3 Alternative approach optimized for compatibility

The private sum method requires coordination among multiple parties, which can be challenging to reach an agreement on, and does not fully leverage the trust assumptions already present in the EUDIW ecosystem. Specifically, if any single party - such as the EUDIW Provider - can be trusted, then a simpler and more autonomous approach becomes feasible.

Importantly, this alternative works with existing protocols and can be implemented independently by each member state. Assuming the EUDIW Provider can act as a trusted aggregation server, the following approach supports scalable usage reporting with reasonable privacy:

1) The wallet logs a usage event as an (issuance_info, service_id) tuple and increments a local counter for that tuple.

2) It then obtains a PID from the PID Provider, including a single-use Proof-of-Possession (PoP) key with no attribute disclosures.

3) The wallet authenticates to the EUDIW Provider via a dedicated endpoint used solely for usage reporting.

4) The EUDIW Provider verifies the PID and requests the usage data recorded in step 1.

The above approach enables aggregation using a trusted party, where the submitting user's privacy is reasonably protected (the EUDIW Provider only uses the PID as proof that the user is valid and sees only a single-use key). Aggregation is now possible with the user submitted values. Additionally, it is now easier to detect possibly fraudulent usage reports since the EUDIW Provider sees the usage numbers in the clear (and the EUDIW Provider can submit PID keys to the PID Issuer for identification).

Annex G:
BBS# applied to ISO mDL

G.1 General

BBS# can be made compatible with ISO mDL or IETF SD-JWT. However, this requires slight modifications to the issuance and selective disclosure protocols described in clause 4.4.3. The present clause describes the modifications necessary for applying BBS# to achieve selective disclosure for the ISO mDL device retrieval flow.

NOTE: The same principles can also be used for applying BBS# to the holder binding key used for signing IETF SD-JWT.

G.2 Setup

Let G denote a cyclic group of prime order p, g̃, g, h1, h2, ⋯, hL, L+3 random generators of G, x is the issuer's private key and PK1 = g̃x is the corresponding public key.

Furthermore, let's it is assumed that the issuer has published L public values, randomly chosen from [1,…,p] and denoted $\{K_i\}_{i=1}^{L}$ as well as another integer (also public), randomly selected from [1,…,p] and denoted as Ud (for "undisclosed") in the following.

NOTE: These public values (which can be the empty value) will be used for all VC issuances carried out by this issuer.

Let sk denote the user's hardware-protected device key, $pk = h^{sk}$, the corresponding public key and (a1, a2, a3, … , aL), their attributes (known to the issuer).

This pair of keys (sk, pk) corresponds to the mDL authentication key in the ISO mDL terminology.

G.3 Issuance

The issuer first computes L digests (cryptographic hashes), one for each attribute. Each of these L digests will be labelled with a unique digest identifier denoted as HIDi. The digest, denoted Hi, is computed for each attribute using its digest identifier (HIDi), the attribute identifier (denoted as ID_{a_i}), the value of the attribute (ai) and the public value (Ki) generated and associated with this attribute during the set-up of this credential schema:
$H_i = Hash(HID_i \,||\, ID_{a_i} \,||\, a_i \,||\, K_i)$ where Hash denotes a cryptographic hash function producing digests in [1,…, p] (such as SHA-256, for example).

NOTE: The ISO/IEC 18013-5 [i.181] standard requires this representation of attributes. It is understood that BBS# would also work with any other representation of attributes.

The issuer creates a MACBBS authentication tag σ on the user's mDL authentication key pk (of a signature scheme supporting key blinding or randomization) and on the L digests $\{H_i\}_{i=1}^{L}$. The tag $\sigma = (A, e)$ represents the user's credential and authenticates both the user's attributes and its mDL authentication key where:

$$A = (gpkh_1^{H1} \cdots h_L^{mL})^{\frac{1}{x+e}} = (gh^{sk}h_1^{H2} \cdots h_L^{mL})^{\frac{1}{x+e}}$$

The issuer then transmits σ to the user. The user's Verifiable Credential (VC) (or Mobile Security Object - MSO in the terminology of ISO/IEC 18013-5 [i.181]) consists of their public key pk, the digests $\{H_i\}_{i=1}^{L}$ and the tag σ on these data: $VC = (pk, \{H_i\}_{i=1}^{L}, \sigma)$. The secret data associated with this VC is sk.

G.4 Selective disclosure

In the following, D will denote the list of indices of the attributes requested by the relying party, which is also the verifier. For example, D={1,5,7} will mean that the relying party wants the user to reveal the attributes a1, a5 and a7. For each attribute not belonging to D, the user will send the value Ud to the relying party.

During a Verifiable Presentation (VP) of the user's attributes (or a subset of them) to the relying party, the EUDI Wallet will first randomize their mDL authentication key pk (either additively if ECSDSA is used on the user's secure cryptographic device or multiplicatively in the case of ECDSA) as well as their tag (i.e. their verifiable credential) σ. These randomized versions are denoted as pkBlind and $\sigma_{Blind} = (\underline{A}, \underline{B})$ respectively.

To guarantee the freshness of the VP, the user will then create a signature σ_{HB}, using the private key associated with pkBlind, on the set of data referred to as "DeviceAuthenticationBytes" in the ISO/IEC 18013-5 [i.181] standard and denoted mDAB in the following. Furthermore, a ZKP $\pi_{Validity}$ proving knowledge is calculated of (a) two random factors (r, r'), (b) a credential σ and (c) of a public pk such that: (1) σ_{Blind} is a randomized version of σ under the random factor r, (2) pkBlind is a randomized version of pk under the random factor r', and (3) σ is a valid MACBBS authentication tag on the disclosed attributes requested by the verifier.

NOTE: The DeviceAuthenticationBytes includes the nonce (or any other equivalent element specific to the current session with the relying party, which helps prevent replay attacks of VPs), possibly the set of data disclosed to the relying party, and other contextual data. However, the ISO/IEC 18013-5 [i.181] standard is not very explicit about the exact content of the "DeviceAuthenticationBytes".

The signature σ_{HB} is a proof that the VP originates from the user holding the underlying credential σ on the attributes disclosed to the verifier.

The VP consists of the following elements:

$$VP = (\{H_i\}_{i\in D}, Ud_{i\in D}, D_{disclosed}, pk_{Blind}, \sigma_{HB}, \sigma_{Blind} = (\underline{A}, \underline{B}), \underline{\pi}_{DLEQ}, \pi_{Validity})$$

where $D_{disclosed} = \{HID_i \,||\, ID_{a_i} \,||\, a_i \,||\, K_i\}_{i\in D}$ represents the set of disclosed attributes, with their associated verification values, and $\underline{\pi}_{DLEQ} := PoK\{\alpha : \underline{B} = \underline{A}^{\alpha} \wedge PK_1 = \tilde{g}^{\alpha}\}$.

Finally, denote $\sigma_{blind}^{Issuer} = (\sigma_{Blind} = (\underline{A}, \underline{B}), \underline{\pi}_{DLEQ}, \pi_{Validity})$. Then $VP = (D_{Disclosed}, \{Ud\}_{i\in D}, pk_{Blind}, \sigma_{HB}, \sigma_{blind}^{Issuer})$.

G.5 Verification

Upon receipt of $VP = (\{H_i\}_{i\in D}, \{Ud\}_{i\notin D}, D_{Disclosed}, pk_{Blind}, \sigma_{HB}, \sigma_{blind}^{Issuer})$, the relying party first computes $H'_i = Hash(HID_i \,||\, ID_{a_i} \,||\, a_i \,||\, K_i)$ for each $i \in D$ and verifies that $H'_i = H_i$ for each $i \in D$. The relying party then checks that the signature σ_{HB} is valid on mDAB, using pk_{Blind}, and then verifies the validity of σ_{blind}^{Issuer} on the $\{H_i\}_{i\in D}$ and pk_{Blind} using PKI.

This last verification consists in verifying that the ZKPs $\underline{\pi}_{DLEQ}$ and $\pi_{Validity}$ are both valid, using the corresponding verification algorithms of these two ZKPs.

If all these checks are successful, this proves that the attributes $\{a_i\}_{i\in D}$ have been certified by the issuer and that the VP indeed originates from the user whose attributes are the $\{a_i\}_{i\in D}$. This verification phase proceeds exactly as in the ISO/IEC 18013-5 [i.181] standard.

Annex H: Bibliography

- Bellare-Goldreich-Goldwasser: "Incremental Cryptography: The Case of Hashing and Signing".
- Ben-or-Goldwasser-Shafi-Kilian-Wigderson: "Multi prover interactive proofs: How to remove intractability assumptions".
- Camenisch-Dubovitskaya-Lehmann: "Concepts and Languages for Privacy-Preserving Attribute-Based Authentication".
- ENISA: "Cybersecurity Certification: Candidate EUCC Scheme V1.1.1".
- ETSI EN 319 102-1: "Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation".
- ETSI EN 319 403-1: "Electronic Signatures and Infrastructures (ESI); Trust Service Provider Conformity Assessment; Part 1: Requirements for conformity assessment bodies assessing Trust Service Providers".
- ETSI EN 319 411-2: "Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 2: Requirements for trust service providers issuing EU qualified certificates".
- ETSI TR 103 619: "CYBER; Migration strategies and recommendations to Quantum Safe schemes".
- ETSI TS 119 182-1: "Electronic Signatures and Infrastructures (ESI); JAdES digital signatures; Part 1: Building blocks and JAdES baseline signatures".
- Fuentes-González-Olvera-Veseli: "Assessment of attribute-based credentials for privacy-preserving road traffic services in smart cities".
- IETF RFC 6749: "OAuth 2.0 Authorization Framework".
- OASIS: "PKCS #11 Cryptographic Token Interface Base Specification Version 2.40".
- OpenID Foundation: "OpenID for Verifiable Credential Issuance".
- W3C®: "JSON-LD 1.1 - A JSON-based Serialization for Linked Data".
- W3C® Working Draft 21 July 2023: "Securing Verifiable Credentials using JOSE and COSE".
- Zhang-Genkin-Katz-Papadopoulos: "vRAM: Faster Verifiable RAM with Program-Independent Preprocessing".

Annex I:
Change history

Date	Version	Information about changes
August 2023	V1.1.1	Publication
January 2024	V1.1.2	Stable draft with updates made according to ESI(23)000072 "Comments on ETSI TR 119 476 V1.1.1 for the revision to ETSI TR 119 476 v1.2.1".
February 2024	V1.1.3	Stable draft with updates made according to ESI(24)082054 "Resolved collated comments on ETSI TR 119 476 v1.1.2".
March 2024	V1.1.4	Editorial edits based on feedback from ETSI's directorate.
April 2024	V1.1.5	Final draft with updates made according to ESI(24)82b004 "Resolved collated comments on ETSI TR 119 476 v1.1.4".
April 2025	V1.2.2	Early draft with updates made according to ESI(25)000077 "Collated resolved comments on ETSI TR 119 476 v1.2.1".
June 2025	V1.2.3	Stable draft with updates made according to ESI(25)000376 "Resolved comments on ETSI TR 119 476 v1.2.2".
June 2025	V1.2.4	Final draft with editorial edits based on feedback from ETSI's directorate.
June 2025	V1.2.5	Final draft with editorial edits based on additional feedback from ETSI's directorate.

History

Document history		
V1.1.1	August 2023	Publication as ETSI TR 119 476
V1.2.1	July 2024	Publication as ETSI TR 119 476
V1.3.1	August 2025	Publication