ETSI GS E4P 006 V1.1.1 (2021-05)

Europe for Privacy-Preserving Pandemic Protection (E4P); Device-Based Mechanisms for pandemic contact tracing systems

Disclaimer

The present document has been produced and approved by the Europe for Privacy-Preserving Pandemic Protection ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/E4P-006

Keywords
covid, eHealth, emergency services, identity,
mobility, pandemic, privacy, security, smartphone

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 Web server (https://ipr.etsi.org/).

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™** and **LTE™** 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 Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Europe for Privacy-Preserving Pandemic Protection (E4P).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**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.

Introduction

The COVID-19 pandemic has generated significant challenges for many countries and their citizens and showed that digital technologies could play an important role in addressing this pandemic and future pandemics. Various applications, services and systems for contact tracing (identification and notification of those who come in contact with a carrier) have been developed in different regions.

Despite the similar goal of automated detection of COVID-19 exposure as a complementary solution to manual tracing (interviews with people diagnosed with COVID-19 to track down their recent contacts), their functionality, technology, scale, required data and limitations are different and may not interoperate.

These systems are currently being deployed in different countries and many more are expected in the near future. In particular, mobile devices with their contact tracing applications can support public health authorities in controlling and containing the pandemic. In that purpose, E4P has been created to provide a technical answer to pandemic crisis not limited to COVID-19 by specifying interoperable contact tracing systems.

1 Scope

The present document sets out device based features which pandemic contact tracing systems should meet to ensure their performance and compliance with the ETSI E4P system requirements and interoperability framework defined in ETSI GS E4P 003 [1] and ETSI GS E4P 007 [i.21]. Systems based on a smartphone with cellular and Bluetooth® connectivity will be studied as the first priority and other solutions could be considered later.

In the context of tracing persons potentially infected with a transmittable virus such as SARS-CoV-2, the ISG E4P develops a framework and consistent set of specifications for proximity tracing systems, to enable the development of applications and platforms, and to facilitate international interoperability as defined in ISG E4P Terms of Reference [i.1]. The present document describes device based mechanisms for the Pandemic Tracing Systems.

In particular, the present document specifies various Proximity Detection Methods for Pandemic contact tracing systems, including:

- Proximity detection of contacts
- Anonymous identification of contacts
- Storage requirements for proximity data of contacts
- User experience and usability

Solutions are specified in technical detail so that means of interoperability between different systems and methods can also be readily defined in ETSI GS E4P 007 [i.21] "Pandemic proximity tracing systems: Interoperability framework". Each method is characterized (e.g. in a table) by its degree of compatibility with the ETSI GS E4P 003 [1].

The present document relates to ETSI GR E4P 002 [i.20] "Comparison of existing pandemic contact tracing systems" ETSI GS E4P 003 [1] "Requirements for Pandemic Contact Tracing Systems using mobile devices" and ETSI GS E4P 008 [2] "Back-End mechanisms for pandemic contact tracing systems". In addition, it will be used as an input to ETSI GS E4P 007 [i.21] "Pandemic proximity tracing systems: Interoperability framework".

2 References

2.1 Normative 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.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

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 are necessary for the application of the present document.

[1] ETSI GS E4P 003 (V1.1.1): "Europe for Privacy-Preserving Pandemic Protection (E4P); High level requirements for pandemic contact tracing systems using mobile devices".

[2] ETSI GS E4P 008 (V1.1.1): "Europe for Privacy-Preserving Pandemic Protection (E4P); Back-End mechanisms for pandemic contact tracing systems".

[3] Bluetooth® Core Specification V5.2.

[4] ETSI EN 301 549: "Accessibility requirements for ICT products and services".

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 are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI ISG E4P Terms of Reference (ToR) for "Europe for Privacy-Preserving Pandemic Protection (E4P)", Version 1.1, 8 May 2020.

[i.2] Google API for Exposure Notification - Exposure Notification BLE attenuations.

NOTE: Available at https://developers.google.com/android/exposure-notifications/ble-attenuation-overview.

[i.3] "The Strength of Friendship Ties in Proximity Sensor Data", Vedran Sekara, Sune Lehmann, Published: July 7, 2014.

NOTE: Available at https://doi.org/10.1371/journal.pone.0100915.

[i.4] "Exposure Notification Bluetooth® Specification", v1.2 April 2020, Google Apple.

NOTE: Available at https://blog.google/documents/70/Exposure_Notification_-_Bluetooth_Specification_v1.2.2.pdf.

[i.5] "Exposure Notification Cryptography Specification", v1.2 April 2020, Google Apple.

NOTE: Available at https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-CryptographySpecificationv1.2.pdf?1.

[i.6] Decentralized Privacy-Preserving Proximity Tracing GitHub repository.

NOTE: Available at https://github.com/DP-3T/bt-measurements/tree/master/figures.

[i.7] CTIA Test Plan for Wireless Device Over-the-Air Performance, Method of Measurement for Radiated RF Power and Receiver Performance, Version 3.8.1, October 2018.

NOTE: Available at https://api.ctia.org/wp-content/uploads/2019/04/CTIA_OTA_Test_Plan_3_8_2.pdf.

[i.8] Google API for Exposure Notification - Exposure Notifications BLE RSSI calibration procedure.

NOTE: Available at https://developers.google.com/android/exposure-notifications/ble-attenuation-procedure.

[i.9] Android™ API Reference Bluetooth® Low Energy.

NOTE 1: Available at https://developer.android.com/reference/android/bluetooth/le/package-summary.

NOTE 2: Android is a trademark of Google LLC.

[i.10] Apple Developer Documentation Core Bluetooth® Framework.

NOTE: Available at https://developer.apple.com/documentation/corebluetooth.

[i.11] Apple Documentation Archive - Core Bluetooth® Background Processing for iOS Apps.

NOTE 1: Available at https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/CoreBluetoothBackgroundProcessingForIOSApps/PerformingTasksWhileYourAppIsInTheBackground.html#//apple_ref/doc/uid/TP40013257-CH7-SW1.

NOTE 2: IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used by Apple under license.

[i.12] PRIVATICS team, Inria, France, Fraunhofer AISEC, Germany, "ROBERT: ROBust and privacy-presERving proximity Tracing, version 1.1", May 31st, 2020.

NOTE: Available at https://github.com/ROBERT-proximity-tracing/documents, https://hal.inria.fr/hal-02611265/en/.

[i.13] C. Castelluccia, N. Bielova, A. Boutet, M. Cunche, C. Lauradoux, D. Le Metayer, V. Roca, "DESIRE: A Third Way for a European Exposure Notification System Leveraging the best of centralized and decentralized systems, version 1.0", May 2020.

NOTE: Available at https://github.com/3rd-ways-for-EU-exposure-notification/project-DESIRE, https://hal.inria.fr/hal-02570382/en/.

[i.14] J-M. Gorce, M. Egan, R. Gribonval, "An efficient algorithm to estimate Covid-19 infectiousness risk from BLE-RSSI measurements".

NOTE: Available at https://hal.inria.fr/hal-02641630/en/.

[i.15] G. Kessibi, M. Cunche, A. Boutet, C. Castelluccia, C. Lauradoux, D. Le Metayer, V. Roca, "Analysis of Diagnosis Key distribution mechanism in contact tracing applications based on Google-Apple Exposure Notification (GAEN) framework (version 1.2)", September 2020.

NOTE 1: Available at https://hal.inria.fr/hal-02899412/en/.

NOTE 2: https://github.com/3rd-ways-for-EU-exposure-notification/project-DESIRE.

[i.16] M. Cunche, A. Boutet, C. Castelluccia, C. Lauradoux, V. Roca, "On using Bluetooth®-Low-Energy for contact tracing (version 1.3)", September 2020.

NOTE: Available at https://github.com/3rd-ways-for-EU-exposure-notification/project-DESIRE and https://hal.inria.fr/hal-02878346/en/.

[i.17] Swiss Confederation, "Replay Attacks", June 14th, 2020, section "Unmasking users by eavesdropping EphIDs".

[i.18] Tijmen Schep, "Corona Detective".

NOTE: Available at https://www.coronadetective.eu/.

[i.19] O. Seiskari, "BLE contact tracing sniffer PoC".

NOTE: Available at https://github.com/oseiskar/corona-sniffer.

[i.20] ETSI GR E4P 002: "Europe for Privacy-Preserving Pandemic Protection (E4P); Comparison of existing pandemic contact tracing systems".

[i.21] ETSI GS E4P 007 (V1.1.1): "Europe for Privacy-Preserving Pandemic Protection (E4P); Pandemic proximity tracing systems: Interoperability Framework".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

diagnosis key: secret code from which ephemeral identifiers for a given period of time can be derived with the help of a cryptographic function

ephemeral identifier: unique device identifier exchanged with another device during a proximity event

proximity event: event recorded by the software on a mobile device corresponding to proximity to other device with an active interoperable application, and which meets the predefined criteria for an event to be recorded (e.g. duration)

3.2 Symbols

For the purposes of the present document, the following symbols apply:

dB	decibel

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ADV_IND	Advertisement Indication
AES	Advanced Encryption Standard
API	Application Programming Interface
Bluetooth® LE	Bluetooth Low Energy
CID	Content IDentifier
CT	Clearance Time
DB	Device - Backend
DD	Device - Device
DP-3T	Decentralized Privacy-Preserving Proximity Tracing
DPA	Data Protection Authority
DUT	Device Under Test
E4P	Europe for Privacy-Preserving Pandemic Protection
EBID	Ephemeral Bluetooth IDentifier
ECC	Encrypted Country Code
FSPL	Free Space Path Loss
GAEN	Google Apple Exposure Notification
GDPR	General Data Protection Regulation
GR	Group Report
GRX	receiving Bluetooth device antenna gain [dB]
GS	Group Specification
GTX	transmitting Bluetooth device antenna gain [dB]
HKDF	Hashed Key Derivation Function
iOS	iOS Operating System
ISG	Industry Specification Group
ISM	Industrial Scientific Medical band
LE	Low Energy
MAC	Medium Access Control
NTP	Network Time Protocol
PDU	Protocol Data Unit
PET	Private Encounter Token
PRF	PseudoRandom Function
PRG	PseudoRandom Generator
QR	Quick Response (code)
RF	Radio Frequency
ROBERT	ROBust and privacy-presERving proximity Tracing
RPI	Rolling Proximity Identifier
RSSI	Received Signal Strength Indicator
SID	EBID Slice IDentifier
SIG	Special Interest Group
TEK	Temporary Exposure Key
TRP	Total Radiated Power
TRP	Total Radiated Power
TX	Transmit
UD	User - Device
UUID	Universal Unique IDentifier
XOR	eXclusive OR

4 General description

4.1 Reference device architecture

The generic reference E4P device architecture is depicted in Figure 1. It describes a user device and its interactions with other components of the system such as its users, Back-End system (described in ETSI GS E4P 003 [1] as part of the Infrastructure) and other devices. It also introduces three corresponding external reference points and interfaces as follows:

a) **Reference point UD** (User - Device) - User interface.

b) **Reference point DB** (Device - Back-End System) - Back-End interface.

c) **Reference point DD** (Device - Device) - Contact proximity detection interface.

The contact tracing protocols are based on decentralized or centralized design approach as defined in ETSI GS E4P 003 [1]). Based on the model of digital contact tracing system defined in ETSI GS E4P 003 [1], main internal device functions implementing pandemic contact tracing Mobile Application as defined in ETSI GS E4P 003 [1] are described in the following clauses of the present document and are also shown in Figure 1.

NOTE: Optional architecture elements and functions are shown in dotted lines.

These include:

a) **User interface** - describes requirements related to interaction between the user and the device (as defined in ETSI GS E4P 003 [1]).

b) **Contact proximity detection** - describes proximity detection method requirements as defined in ETSI GS E4P 003 [1].

c) **Anonymous contact identification** - describes anonymous contacts identification requirements. This function shall be present in the Device for decentralized approach only as it shall be implemented in Back-End for the centralized approach.

d) **Contact data storage** - describes requirements for data storage and contact tracing protocol between the device and the Back-End. This function may also include Ephemeral IDs generation sub function if it is not implemented in the Back-End.

Detailed architecture of the Back-End system is out of scope of the present document and is described in ETSI GS E4P 008 [2]. High level requirements related to unique device identifiers exchanged on DD interface (Ephemeral IDs) are defined in ETSI GS E4P 003 [1].

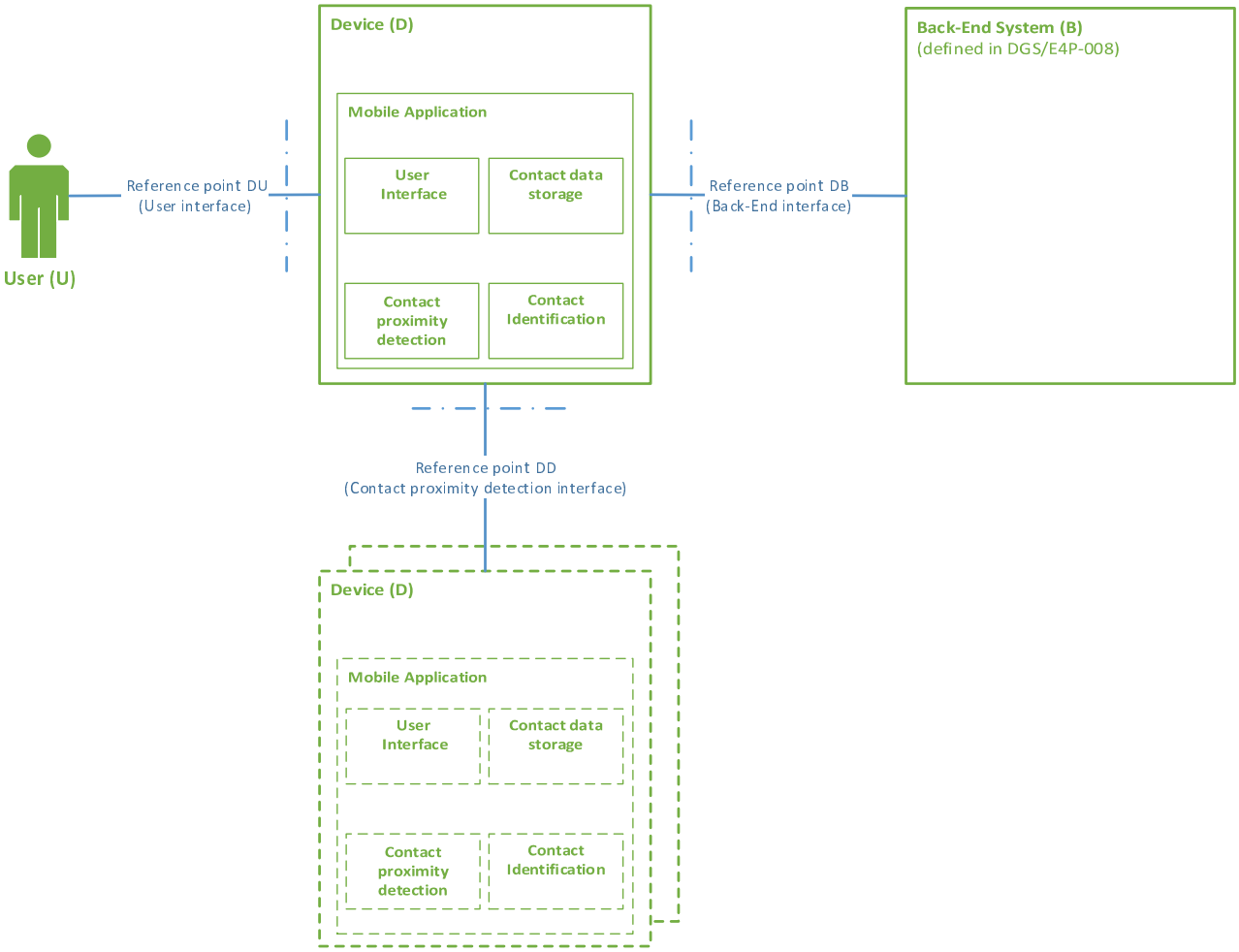

Figure 1: Reference E4P device architecture

5 Device-based mechanisms for pandemic contact tracing systems

5.1 Contact proximity detection

5.1.1 Contact proximity detection technical options

Almost all mobile device-based proximity tracing protocols, both centralized and decentralized, proposed so far suggest using radiated transmit power of advertisement packets or signal attenuation of Bluetooth® Low Energy (LE) signals as a substitute for spatial proximity. This information combined with the duration of exposure for the past few days is used for infection risk score calculation and subsequent notification based on exceeding a threshold.

The Bluetooth LE Received Signal Strength Indicator (RSSI) measurements can be affected by a number of factors such as antenna sensitivity, device orientation and environmental conditions which makes it challenging to use it for distance estimation. The signal attenuation is defined as the difference between the transmission power at the sending device and RSSI. The difficulty in measuring the distance based on the attenuation arises due to signals being noisy. The multipath and shadowing effects due to walls, objects and people also affect signal propagation, however the authors of Decentralized Privacy-Preserving Proximity Tracing (DP-3T) [i.3] have conducted experiments to show that rarely do these hurdles decrease the attenuation and use this fact to propose a mechanism for distance estimation between two devices.

The authors show that below 50 dB of attenuation there is a good chance that devices are within 2 meters of each other, which is a suggested minimum distance to avoid contacting infection. Further calibration is applied at both the sending and receiving device to account for the variance in transmission power and RSSI measurements across device models. The key goal of the approach presented by the authors is not to accurately measure the distance between devices but to estimate the breach of the 2-meter threshold, accurate distance measurement is rather challenging considering the diversity of operation environments and low Bluetooth LE beaconing frequency.

ISG E4P also propose using the Google Apple Exposure Notification (GAEN) approach, which exchanges advertisements with neighbouring devices every 2,5 to 5 minutes, to estimate the proximity using a probabilistic measure of distance by observing the received beacon attenuation level. Thus, allowing a receiver to estimate the overall duration of time it was exposed at an attenuation level using a set of beacons and corresponding attenuations received.

5.1.2 Bluetooth® LE usage

5.1.2.1 Bluetooth® LE usage requirements

[DCP-01]: Bluetooth LE is a subset of Bluetooth. Its full description can be found in the clause 4.5 of the Bluetooth specification [3]. The contact proximity detection function in a mobile device of a contact tracing system can use Bluetooth LE to do a contact proximity detection. For this purpose, the Bluetooth component of the mobile device shall comply with clause 4.5 of Bluetooth specification [3].

[DCP-02]: The contact proximity detection function in a mobile application shall be able to use the Bluetooth LE capabilities via the Bluetooth LE API of the mobile device. All Bluetooth LE functions and parameters shall be accessible via the Bluetooth LE API of a mobile device.

NOTE 1: The instructions how to use the Bluetooth LE API by a mobile application are described in [i.9] for Android mobile devices and in [i.10] for IOS mobile devices.

NOTE 2: Bluetooth LE is also used in a dedicated API called GAEN [i.4] that can be used in the contact proximity detection function of the decentralised contact tracing system application to do all the contact proximity detection functions. The Bluetooth LE setup is defined in the GAEN API and cannot be modified.

5.1.2.2 Bluetooth® LE API usage

Bluetooth LE API shall be used in any centralized approach as the GAEN API is not applicable However, the Bluetooth LE API can also be used in a decentralized approach without relying on GAEN API.

The use of the Bluetooth LE API can lead to some restriction to the Bluetooth LE functions on Apple mobile devices when the application is running in background mode or in suspended mode [i.11]. These restrictions are not inherent to Bluetooth LE, but the result of choices made in iOS to prevent malicious applications that leverage on Bluetooth LE. These restrictions most likely could be modified in iOS at any time. In the meantime, workarounds are required to bypass up to a certain point these restrictions. One working workaround is the presence of a Bluetooth LE beacon or Android device broadcasting which leads to wakening up of the app running on iOS. With the French StopCovid/TousAntiCovid application, the presence of Android versions of the application is required to wakeup iOS versions of the application. These workarounds are very specific and are not further described in the present document.

5.1.3 Bluetooth® LE advertisement mode

[DCP-04]: Bluetooth LE can be used in the contact proximity detection function to make continuous advertisement of a given contact tracing service and, in the same time, to share a specific payload used to assess proximity with a user that shall be anonymised.

This clause describes the advertisement modes and recommended advertisement payload for contact proximity detection.

[DCP-05]: Two Bluetooth LE advertisement modes are available for the contact proximity detection function and the mobile application can use either or both modes. They are defined as follows:

a) **Bluetooth LE Broadcast mode** - consists of broadcasting advertisement together with payload required to make the proximity detection. The broadcast mode shall use PDU type ADV_NONCONN_IND as described in Bluetooth Sig Vol.6 Part B, clauses 2.3 and 2.3.1.3 of [3].

b) **Bluetooth LE Connected mode** - consists of broadcasting in the first step only the advertisement of the service and in the second step, connecting with another device to share contact data required to make the proximity detection. The broadcast mode shall use PDU type ADV_ IND as described in Bluetooth Sig Vol.6 Part B, clauses 2.3 and 2.3.1.2 of [3].

[DCP-06]: Both Bluetooth LE advertisement modes in the contact proximity detection function shall use the physical layer LE1 as described in Bluetooth Sig Vol.1 Part A, clause 1.2 of [3]. Advertising physical channel Protocol Data Unit (PDU) is described in Bluetooth Sig Vol.6 Part B, clause 2.3 of [3].

[DCP-07]: A Universally Unique Identifier (UUID) shall be used by the contact proximity detection function to allow service discovery of the contact tracing service. The UUID shall be registered to Bluetooth SIG.

5.1.4 Bluetooth® LE RSSI measurement suitability for proximity detection

The reported Bluetooth LE RSSI to estimate the distance between the two Bluetooth devices is based on a number of parameters, which are described in the following Free Space Path Loss (FSPL) equation 1:

$$FSPL(dB) = 20log(d) + 20log(f) - 27{,}55dB - GTX - GRX \quad (1)$$

where:

- d- distance between the two Bluetooth LE devices [m];
- f - communication frequency [MHz];
- GTX - transmitting Bluetooth device antenna gain [dB];
- GRX - receiving Bluetooth device antenna gain [dB].

The frequency f is known as it is based on the Industrial, Scientific and Medical (ISM) bands used in Europe in the frequency range 2,4 - 2,5 GHz. GTX and GRX are not known a priori even if they are based on the technical specifications of the device (brand and model). In addition, the Bluetooth LE device may introduce another correction factor in the conversion between the received power and the RSSI provided to the application layer. The GTX, GRX and the correction factor are all unknowns which can negatively impact the proper evaluation of the distance from the reported RSSI.

The position of the Bluetooth LE devices (e.g. mobile phones) determine different values of GTX and GRX on the basis of the radiation patterns of the receiver and transmitter Bluetooth device. Such radiation patterns are also different for each polarization used in the wireless communication Bluetooth standard.

If the coordinated system is used as described in Figure 2, the radiation pattern of various models of phones were measured in a calibrated shielded anechoic chamber at the Joint Research Centre facilities of the European Commission in Ispra, Italy. An example of such radiation pattern is reported in Figure 3.

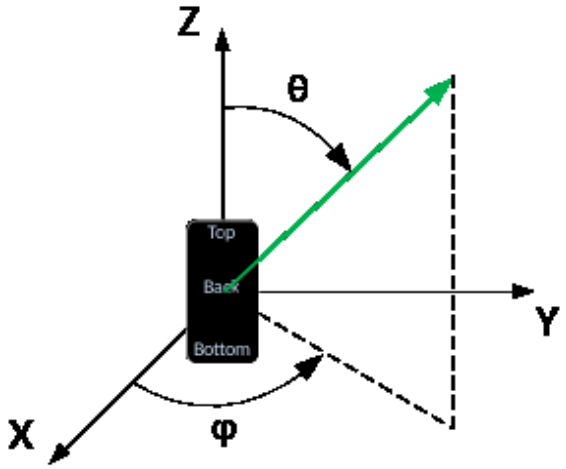

Figure 2: Position of the Bluetooth® LE device (smartphone) in the coordinate system

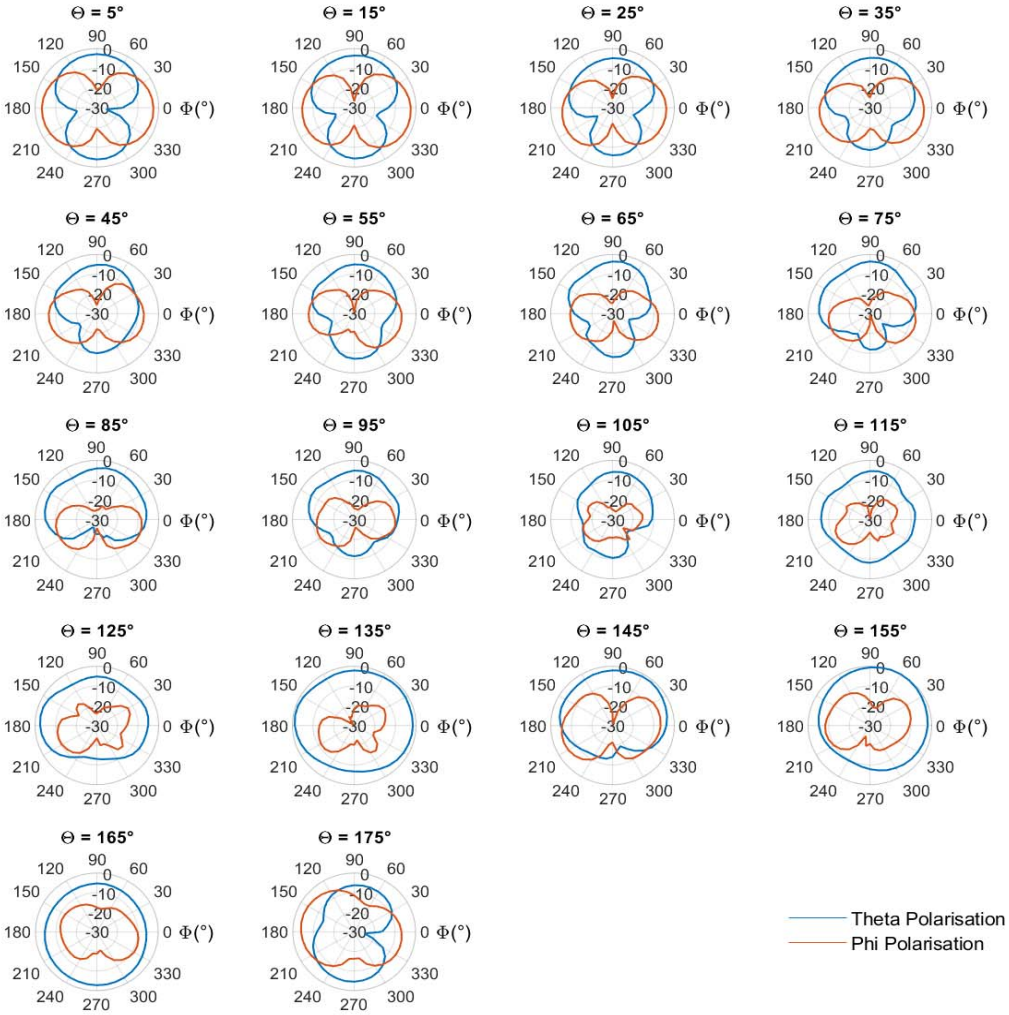

Figure 3: An example of a reported radiation pattern for two different polarizations for a smart phone model

It can be seen from Figure 3 that the radiation patterns vary significantly with the value of the angles. As a result, the power received by a mobile phone for the proximity detection from another phone will vary significantly for different positions and orientations of the phone. This observation is confirmed with the result of experiments conducted on different phone models, which are reported in Table 1 below:

Table 1: Peak and average gain for different polarizations for different phone models

Mobile Phone Model	Peak Gain Theta Polarization (dBi)	Peak Gain Phi Polarization (dBi)	Median Gain Theta Polarization (dBi)	Median Gain Phi Polarization (dBi)
Model 1	-3,8	-6,2	-10,8	-16,7
Model 2	-10,6	-10,9	-18,4	-20,8
Model 3	-1,0	-0,7	-8,2	-12,3
Model 4	-7,4	-7,4	-16,5	-18,1
Model 5	-10,7	-11,0	-19,9	-18,5
Model 6	-12,0	-5,8	-24,5	-20,0
Model 7	+4,5	+0,9	-2,9	-10,2
Model 8	+4,0	+3,3	-3,9	-7,4
Model 9	-7,0	-7,8	-14,6	-21,0
Model 10	-4,4	-4,9	-17,0	-17,2

As it can be seen from Table 1, different models have significant differences even in the peak and average gain, which means that even in ideal positions of the Bluetooth LE devices, the calculated distance for the proximity detection may not be easily derived from the reported RSSI and a correction factor shall be introduced, which depends on the orientation of the Bluetooth device. Finally, even the total radiated power of a Bluetooth LE device can have variations among different models.

The Total Radiated Power (TRP) is calculated from equation 2 [i.7].

$$TRP \cong \frac{\pi}{2NM} \sum_{i=1}^{N-1} \sum_{j=0}^{M-1} \left[EiRP_{\theta}(\theta_i, \phi_j) + EiRP_{\phi}(\theta_i, \phi_j) \right] sin(\theta_i) \quad (2)$$

where:

- TRP - Total Radiated Power calculated for a complete sphere;
- N - number of theta intervals with even angular spacing;
- M - number of phi intervals with even angular spacing;
- EiRP - Effective Isotropic Radiated Power [dBi].

Using the same experimental data set used for the previous results, the total radiated power was calculated for the same models of Bluetooth LE devices (i.e. mobile phones) listed before. The results are shown in Table 2.

Table 2: Total radiated power for different models

Mobile Phone Model	Tx Power (dBm)	TRP (dBm)	Efficiency (dB)
Model 1	-2	-9,4	-7,4
Model 2	+1	-12,5	-13,5
Model 3	-1	-4,7	-3,7
Model 4	-1	-12,3	-11,3
Model 5	+1	-12,5	-13,5
Model 6	+1	-11,3	-12,3
Model 7	-1	-0,15	+0,85
Model 8	0	+0,74	+0,74
Model 9	+12	+1,25	-10,8
Model 10	+12	+1,8	-10,2

Table 2 indicates two different observations. The first is that the total radiated power can still vary significantly across models. This can create problems in the calculation of the distance from the RSSI as each Bluetooth LE device will generate the Bluetooth signals with different levels of power. Then, a correction factor is needed for each phone model. The second observation is that the efficiency is relatively low for mass market mobile phones. Then, presence of obstacles, which can create higher levels of attenuation, can impact the correctness of the calculation of the distance for the proximity detection even more.

Limitations of Bluetooth LE signals as a ranging technology are shown in Figure 4 which shows the limited correlation between the measured Bluetooth signal attenuation and the actual distance [i.6]. Especially for the shorter distance range below 4 meters typically used for the contact proximity detection, the corresponding attenuation range is significant which may impact the distance evaluation accuracy and cause proximity detection errors.

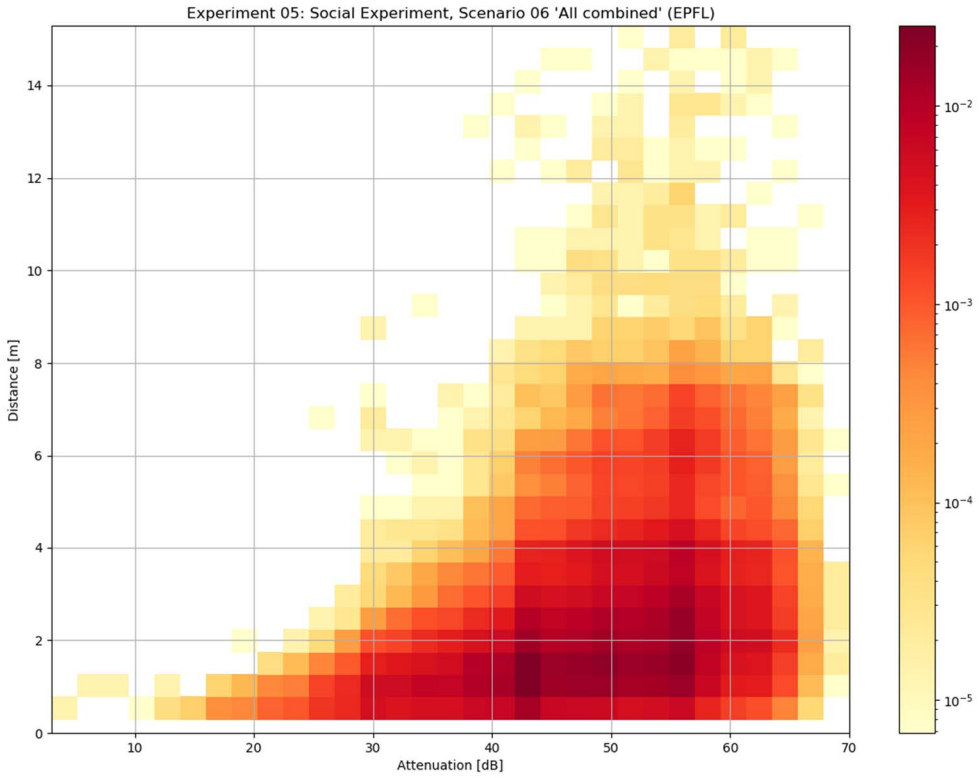

Figure 4: Measured correlation between Bluetooth® LE signal attenuation and distance [i.6]

5.1.5 Bluetooth® LE calibration

Bluetooth LE signal attenuation is not the best indicator of proximity due to being noisy but can offer some benefit if used carefully. Each user device using contact proximity detection interface (reference point DD in the reference architecture) based on Bluetooth will have a unique Bluetooth RF performance, based upon its Bluetooth chipset, RF Front End and antenna design. This means that in order to utilise the reported Bluetooth RSSI to estimate the distance between the two Bluetooth devices, there is a need to understand the difference in performance between each device. Once the performance difference is understood it is possible to calculate a calibration factor to normalise the reported RSSI with a known offset, typically measured on a 'reference' device. This will then allow the user device to calculate approximated distance between devices from a lookup table of calibration values per device type. It is noted that 3rd parties including Google have been working on solutions for calibration methods and publication of calibration values and how to apply them within the device.

5.1.6 Decentralized approach

5.1.6.1 Calibration in Google Apple Exposure Notification (GAEN)

At present (August 2020) for Android Bluetooth LE devices Google has made their work public, see [i.2]. The Google Apple Exposure Notification (GAEN) approach proposes calibrating devices using a Bluetooth RSSI Calibration Tool app and generating a per-device TX_power and RSSI_correction for increased accuracy. The procedure includes collecting data with mobile devices in different orientations - portrait, landscape and facing skyward (12 device orientations in total) and using it to predict calibration values. It is important to test various orientations as the readings can vary by about 10 dB only based on the orientation.

The calibration software collects the uncalibrated RSSI that a stationary reference device (an Android phone) measures when the Device Under Test (DUT) emits packets at ULTRA_LOW_POWER, LOW_POWER, MEDIUM_POWER and HIGH_POWER to calculate TX_power.

The reference device is kept at a distance of 1 meter from the DUT after which the calibration app takes the median of 10 RSSI measurements at each power level. The RSSI_correction is also computed in a similar fashion, by taking the median of 10 uncalibrated RSSI measurements at different power levels. The data is collected for both the DUT and a reference DUT (another Pixel 4 phone) so that the transmit power and the receive sensitivity can be compared.

The transmitted and received power calibrations are configurations that are regularly updated and pushed to the devices as more calibration results are received. The Exposure Notifications framework scans for advertised beacons in the background (for 4 seconds every 5 minutes) on multiple channels.

However, due to the way Bluetooth LE is implemented in Android scanning does not always happen in all three channels. The observed identical Bluetooth beacons are grouped into a `ScanInstance` and are represented by the minimum attenuation and the average attenuation of the beacons in the group, the `ScanInstances` are exposed through the Exposure Notifications API. `ExposureInformation` is used to expose a three-bucket histogram of attenuations, `getAttenuationDurationsInMinutes()` returns the durations for which the minimum attenuation of exposures fell in three different buckets, below the low threshold, between the low threshold and the high threshold and above the high threshold. The thresholds can be configured using `ExposureConfiguration`.

This can be used to compute the exposure risk. `ExposureInformation` also provides `attenuationValue` which is a duration-weighted average of the `ScanInstance` minimum attenuation values. A summary over all exposures is represented by `ExposureSummary` while `getAttenuationDurationsInMinutes`() summarizes attenuations over all exposures as the aggregate of all `ExposureInformation.getAttenuationDurations()`.

5.1.7 Centralized approach

5.1.7.1 Calibration in the French TousAntiCovid/ROBERT digital exposure notification tool

Report [i.14] describes the algorithm used to predict the Covid-19 infectiousness risk from physical contacts through Bluetooth LE RSSI measurements. In order to derive a robust risk estimator, the proposed algorithm relies on known physical wireless propagation effects and on technical properties of current Bluetooth LE interfaces. The proposed algorithm has been tested on the data acquired by the German teams from the Fraunhofer institute in the PEPP-PT European project as well as on data acquired from May 18 to 20, 2020 in France during field tests (outside, within a building, within public transportation).

5.1.7.2 Calibration in DESIRE protocol

DESIRE, as a 'third way' protocol (see clause 5.2 for more details about the 'third way' meaning), currently shares the calibration technique of the French TousAntiCovid/ROBERT tool.

[DCP-08]: Device using contact proximity detection function based on Bluetooth LE technology shall use the calibration procedure to ensure RSSI measurements consistency.

NOTE: Bluetooth LE calibration procedure example is presented in [i.8] (GAEN) and [i.12] (ROBERT/DESIRE).

5.2 Anonymous contact identification

5.2.1 Contact identification protocols

While all protocols propose exchanging some sort of anonymous identifiers to record the proximity of a contact for tentative future processing, how these identifiers are crafted, varies significantly based on the role of the backend server and the protocol design choices.

Whereas the decentralized protocols conduct all processing on the local device to estimate exposure to an infection as well as the underlying risk, the centralized protocols use a Back-End system for this task. Additionally, the contact graph of an individual remains local at all times in the decentralised approach.

In the centralised approach [i.12], a list of contacts potentially at risk (rather than a graph) is uploaded by a user tested positive, and this list of contacts is mixed with the other uploaded lists before being processed. The Back-End is in charge of this functionality (see ETSI GS E4P-008 [2]). The process is guaranteed by regular Data Protection Authority (DPA controls).

Another difference between the two protocols lies in the protection of the "diagnosed status": whereas a centralised protocol like ROBERT [i.12] never collects any such information (it remains on the user's smartphone), a decentralised protocol like GAEN exchanges the "diagnosed keys" publicly in a shared database freely accessible on Internet [i.15]. A database containing health sensitive data that can be easily de-anonymised via trivial Bluetooth LE scanning [i.17], [i.18] and [i.19]. This weakness is the result of a design choice to protect the social graph over user infection status, although this might be sensitive data under GDPR.

Third way approaches like DESIRE, that offer several deployment scenarios (centralized, decentralized, intermediate), provide some more flexibility with respect to the previously mentioned trade-offs since additional considerations can be taken into account (e.g. confidence of users in their DPA and institutions).

These are points of fundamental difference in the two primary approaches. Anonymous contact identification and diagnosed user's anonymity are two fundamental requirements for all proximity tracing protocols as they build trust in a system by allowing the users to report infections confidently without the risk of being stigmatised.

Additionally, many countries have legal regulations which safeguard such privacy. Public trust is an important aspect in self-reporting infections and large-scale adoption of digital proximity tracing are key points on which its success depends. Another aspect is the need for informed user consent (who need to understand the risks in sharing sensitive information) when the legal basis for the sharing of diagnosed keys (decentralized) or contact history (centralized) is the user consent. The following clauses describe in detail with examples how anonymous contact discovery is implemented in these two broad categories of protocols.

5.2.2 Decentralized approach

5.2.2.1 Anonymous contact identification in GAEN

5.2.2.1.1 The GAEN protocol

GAEN follows the above-mentioned approach to generate Rolling Proximity Identifiers (RPI) which are exchanged when the devices are in close proximity of each other. A 16-byte Temporary Exposure Key (TEK) which is rotated every 24 hours (TEKRollingPeriod) is generated locally on the device using a cryptographic random number generator, $tek_i \leftarrow CRNG(16)$.

The Temporary Exposure Key (TEK) is used to generate RPI Keys using a Hashed Key Derivation Function (HKDF), $RPIK_i \leftarrow HKDF(tek_i, NULL, UTF8(\text{EN-RPIK}), 16)$. The RPI Key is then used to generate RPIs as $RPI_{i,j} \leftarrow AES128(RPIK_i, PaddedData_j)$, where j is the Unix Epoch Time at the moment the RPI is rolled over and $ENIN_j \leftarrow ENIntervalNumber(j)$, where i is the ENIntervalNumber that maps the current time to current interval of operation.

The RPIs are rotated every 10 to 20 minutes (based on specification), the same frequency at which the Bluetooth randomized address is changed, to prevent linkability and wireless tracking. Each participating device records a list of RPIs it encounters. Anonymous identifier and Bluetooth payload generation process is described in [i.5].

[DCI-01]: Device using anonymous contact identification function based on GAEN protocol should use the anonymous identifier and Bluetooth payload generation process defined in [i.5].

5.2.2.1.2 Bluetooth® message structure in GAEN

GAEN Exposure Notification Service payload consists of three clauses, Flags Section which contains Bluetooth Low Energy general discoverable mode set to 1, the complete 16-bit Service UUID Section and Service Data 16-bit UUID Section. The Service Data section contains a 16 byte Rolling Proximity Identifier and a 4 byte Associated Encrypted Metadata, one byte each of the metadata is used for versioning and transmit power level, while the remaining two bytes are reserved for future use.

The Table 3 below shows the structure in detail as described in GAEN Bluetooth Specification [i.4].

Table 3: Bluetooth® advertising payload in GAEN

Flags			**Complete 16-bit Service UUID**			**Service Data - 16 bit UUID**				
Length	**Type**	**Flags**	**Length**	**Type**	**Service UUID**	**Length**	**Type**	**Service Data**		
0x02	0x01 (flag)	0x1A	0x03	0x03 (Complete 16-bit Service UUID)	0xFD6F (Exposure Notification Service)	0x17	0x16 (Service Data - 16 bit UUID)	0xFD6F (Exposure Notification Service)	**16 bytes Rolling Proximity Identifier**	**4 bytes Associated Encrypted Metadata**

[DCI-02]: Device using anonymous contact identification function based on GAEN protocol should use the Bluetooth message structure defined in [i.4].

5.2.2.2 Decentralized Privacy-Preserving Proximity Tracing (DP-3T)

DP-3T protocol (which GAEN was inspired of) also uses a similar principle to detect contacts anonymously. The participating smartphones locally generate pseudo-random ephemeral identifiers (EphIDs) and broadcast them using Bluetooth LE beacons. Simultaneously, the smartphones also listen for these beacons and store the broadcast EphIDs with a timestamp and signal attenuation to estimate exposure. In line with this approach three designs described below are proposed with varying trade-offs between privacy and computation cost - Low-cost, Unlinkable and Hybrid:

a) **Low-cost design** - starts by generating a random initial daily seed SK_t for the current day t. The secret day seed is rotated daily and the new seed is set as the hash of the previous day seed, $SK_t = H(SK_{t-1})$, where H is a cryptographic hash function. The daily secret seed is used to generate a list of ephemeral ids (EphIDs) with the lifetime of a few minutes L, known as epoch (e.g. 15 minutes).

   Each day t begins with the device generating SK_t and then using it to generate $n = \frac{24\times60}{L}$ EphIDs as $EphID_1 \vee ... \vee EphID_n = PRG\big(PRF(SK_t, \text{broadcastkey})\big)$; where PRG is a pseudorandom generator such as AES in counter mode, PRF is a pseudorandom function such as HMAC-SHA256 and "broadcast key" is a fixed, public string. The EphIDs are then broadcast in a random order.

b) **Unlinkable design** - provides more robust privacy properties by changing the generation of EphIDs for each epoch i by picking a random 32-byte per-epoch $seed_i$ and then computing the $EphID_i = LEFTMOST128\big(H(seed_i)\big)$. This prevents an adversary from linking EphIDs after an infection is broadcast as in Low-cost design.

c) **Hybrid design** - tries to combine the best of both worlds by grouping consecutive epochs into a time window w which is a multiple of L. At the start of each time window w, the devices pick a random 16-byte $seed_w$ and compute EphIDs as $EphID_{w,1} \vee ... \vee EphID_{w,n} = PRG\big(PRF(seed_w, \text{DP3T-HYBRID})\big)$, where "DP3T-HYBRID" is a fixed, public string. The EphIDs are then broadcast in a random order for each time window. This approach can be seen as the GAEN approach with w set as 24 hours.

5.2.3 Centralized approach

5.2.3.1 ROBERT

5.2.3.1.1 ROBERT protocol

This clause describes the ROBERT [i.12] protocol. Applications interact with the system through the four following procedures:

- **Initialization:** When a user wants to use the service, he/she installs the application, App, from an official application marketplace. The App then registers to the server that generates a permanent identifier (ID) and several Ephemeral Bluetooth Identifiers (EBIDs). The Back-End maintains a table (IDTable) that keeps an entry for each registered ID (see ETSI GS E4P-008 [2]). The stored information is "anonymous" and, by no mean, associated to a particular user (no personal information is stored in IDTable).

- **Proximity Discovery:** After registering to the service, the App broadcasts HELLO messages over its Bluetooth interface and collects HELLO messages from other devices, running the same application, in the vicinity. These HELLO messages contain several fields, and in particular, an Ephemeral Bluetooth Identifier. The collected HELLO messages are stored, together with the time of reception (and possibly other information such as the strength of the Bluetooth signal or the user's speed) into a local list of the application, the LocalProximityList.

- **Diagnosed User Declaration:** When an individual is tested and diagnosed COVID-positive, and after an explicit user consent and authorisation (from the medical services), his/her smartphone's application uploads its LocalProximityList to the authority server, Srv. Srv then flags as "exposed" all IDs of IDTable of which at least one EBID appears in the uploaded LocalProximityList. It is important to note that:

  - The server does not learn the identifiers of the diagnosed user's App but only the EBIDs contained in its LocalProximityList (list of Ephemeral Bluetooth Identifiers he/she was in proximity with).

  - Given any two random identifiers of IDTable that are flagged as "exposed", the server Srv cannot tell whether they appeared in the same or in different LocalProximityList lists (the proximity links between identifiers are not kept and, therefore, no proximity graph can be built).

- **Exposure Status Request:** App queries (pull mechanism) the "exposure status" of its user by probing regularly the server with its EBIDs. The server then checks how many times the App's EBIDs were flagged as "exposed" and computes a risk score from this information (and possibly other parameters, such the exposure duration or the user's speed/acceleration during the contact). If this score is larger than a given threshold, the bit "1" ("at risk of exposure") is sent back to the App and his/her account is deactivated, otherwise the bit "0" is sent back. Upon reception of this message, a notification is displayed to the user that indicates the instructions to follow (e.g. go the hospital for a test, call a specific phone number, stay in quarantine, etc.).

5.2.3.1.2 Bluetooth® message structure in ROBERT

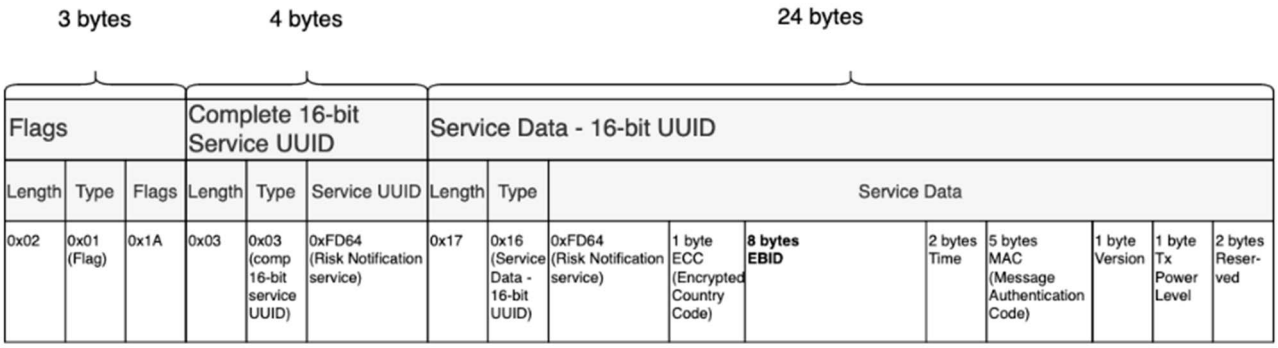

Figure 5: Bluetooth® advertising payload in ROBERT

The Bluetooth LE HELLO message structure of ROBERT is depicted in Figure 5. It is close to that of GAEN (Table 3), relying on the same Bluetooth LE broadcast approach. It is organised as follows [i.16]:

- Flags (3 bytes): Not specific to exposure notification, but included in every advertising packet.
- Complete 16-bit Service UUID (4 bytes): carry the UUID of the dedicated risk notification service (0xFD64).
- Service Data - 16-bit UUID (24 bytes): carry the data associated to the risk notification service along with the risk notification UUID:
  - UUID of the dedicated risk notification service (0xFD64).
  - ECC (1 byte): encrypted country code.
  - EBID (8 bytes): the temporary identifier used by the ROBERT protocol.
  - Time (2 bytes): encodes the current epoch.
  - MAC (5 bytes): message authentication code computed from EBID and Time using HMAC-SHA256.
  - Metadata (4 bytes): version on 1 byte, txPowerLevel on 1 byte, and 2 additional bytes reserved for future use.

5.2.3.1.3 Underlying assumptions for ROBERT: adversarial model

A centralised protocol such as ROBERT assumes the following adversarial model:

- Users can be malicious. They can, for example, perform active and passive attacks, eavesdrop, inject bogus messages, modify messages, pollute users' contact lists and develop their own applications.
- The authority running the system, in turn, is "honest-but-curious". Specifically, it will not deploy spying devices or will not modify the protocols and the messages. However, it might use collected information for other purposes such as to re-identify users or to infer their contact graphs. It is assumed that the Back-End system is secure, and regularly audited and controlled by external trusted and neutral authorities (such as Data Protection Authorities and National Cybersecurity Agencies) see ETSI GS E4P-008 [2].

The security assumptions (conversely adversarial model) behind a ROBERT is key and requires the citizen trust their institutions and their Data Protection Agency (DPA).

A centralised approach also creates significant robustness requirements on the backend server that gathers some sensitive data essentially the "at risk" status of each user, along with some keying material that are used to generate all the EBIDs of a user. A careful design is required as well as continuous security and privacy audits. However, many online services share similar requirements.

Decentralized protocols are safeguard against abuse of power by the backend server, but they are prone to other attacks which centralized protocols are not.

5.2.3.2 DESIRE

5.2.3.2.1 DESIRE protocol

DESIRE changes the way the EBIDs are generated: rather than using public device pseudonyms, it relies on private encounter pseudonyms, a 32-byte value called Private Encounter Token (PET). A PET changes over time (every 15 minutes) and is computed in a private manner by the users who met, and only them (e.g. Alice and Bob in Figure 5), using well-known, robust, Diffie-Hellman cryptography. This approach of generating EBIDs was previously proposed to mitigate privacy risks of the decentralised DP-3T protocol and was more elaborated e.g. in the Pronto-C2 protocol.

In this approach, Alice generates a random value X and computes $g^X \bmod p$, which constitutes her EBID, and Bob does the same with $g^Y \bmod p$ (p being a prime number and g a generator of the cyclic group Z_p^*). Alice and Bob then exchange their 32-bytes EBIDs during their encounter, using Bluetooth LE broadcast messages (see clause 5.2.3.2.2 for Bluetooth LE details).

Upon receiving Bob's EBID, Alice can locally derive the encounter $PET=H((g^Y)^X \bmod p)=H(g^{X.Y} \bmod p)$ because she is the only one to know the secret X, and the same for Bob (see [i.13] for more details on cryptographic considerations).

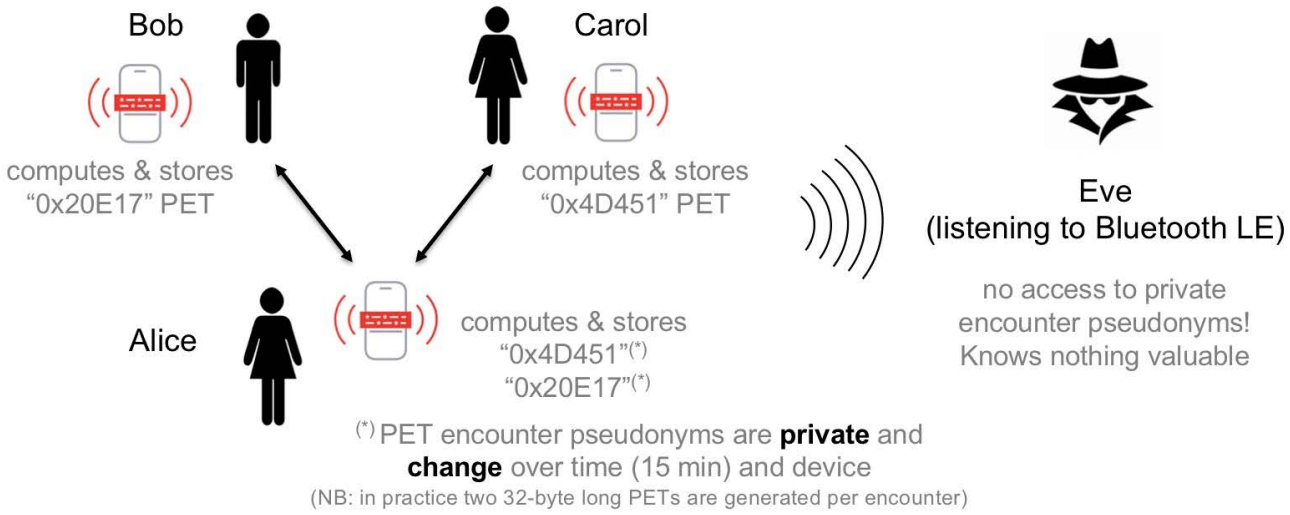

Figure 6: DESIRE PET principle and robustness in front of passive eavesdropping

Using Private Encounter Token changes the situation from the security and privacy viewpoints: an eavesdropper who passively listens to Bluetooth LE traffic is left powerless, because he/she cannot compute the PET between Alice and Bob, and he/she will never be in position to know that a given PET refers to this Alice/Bob encounter that day at that precise time, or to any other encounter, or even that this is a valid PET rather than a random value. Nevertheless, PETs do not solve certain privacy problems if the attacker is not only listing but does actively participate in the communication.

In practice two 32-byte PETs are derived per encounter to avoid linkability: one PET is used by Alice during a Status Request and the other one during the upload of Alice exposed PET list if she is tested COVID+, while Bob uses the second PET during a Status Request and the first PET during an upload of Bob exposed PET list. Doing so prevents a server to link Alice and Bob and determine they met when they both query for their status (see [i.13] for more details).

Another key benefit of DESIRE is that it enables either a centralized (like ROBERT), or decentralized (like GAEN) risk evaluation, or something in-between. And this architectural choice does not impact interoperability: all DESIRE deployments fully interoperate, seamlessly. It means that each country can choose what to do, in a sovereign manner. For instance:

- if users trust their DPA and institutions, a centralized deployment is recommended (improved robustness in front of re-identification attacks);
- if users do not trust their institutions and are not afraid of re-identification risks (and what it may trigger), a decentralized deployment is recommended.

5.2.3.2.2 Bluetooth® message structure in DESIRE

Since Bluetooth LE Advertising frames have a limited payload capacity (16 bytes plus 4 extra bytes), a 32 byte EBID is spread over four slices:

- slice 0: the low order 12 bytes of the EBID;
- slice 1: the following 12 bytes of the EBID;
- slice 2: the high order 8 bytes of the EBID, prepended with 4 bytes padding (set to zero) in order to align to 12 bytes (as with other slices);
- slice 3: 12 bytes corresponding to the XOR sum of the three other slices in order to enable a terminal to reconstruct an EBID after receiving any subset of 3 slices out of the 4 that are transmitted.

The Bluetooth LE message structure is globally similar to that of ROBERT (Figure 5), yet the content of the 16 byte payload differs as follows (Figure 7):

- EBID Slice Identifier (SID) (2 bits): 2 bits that identify the EBID slice contained in the Bluetooth LE payload.
- Content Identifier (CID) (30 bits): random value that identifies the EBID. Note that the source Bluetooth LE MAC address cannot be reliably used to identify the EBID since it may change during an epoch, which advocates for a separate identifier.
- EBID slice: the 12 bytes long slice of the original EBID. Slice 3 allows the application to reconstruct one other slices in case an advertisement is missing.

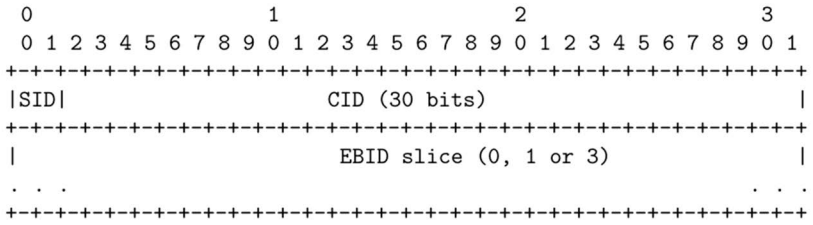

Figure 7: Bluetooth® LE Advertising frame payload for UUID value 0xFD64 and slices 0, 1 or 3

5.3 Contact data storage

5.3.1 General considerations

Both decentralised and centralised protocols identify contacts by exchanging anonymous identifiers. The identifiers are exchanged when the devices are in close proximity of each other. This clause describes contact data storage and ephemeral IDs generation requirements.

5.3.2 Decentralized approach

In the decentralized protocol the participating devices locally generate pseudo-random ephemeral identifiers and broadcast them using Bluetooth LE beacons. In parallel, the devices also listen for these beacons and store the broadcast ephemeral identifiers of mobile devices in proximity with a timestamp and signal attenuation to estimate exposure.

In the event of an infection the user can choose to self-report the infection to the mobile application. When reported, the mobile application will self-upload the representation of the user's ephemeral identifiers (diagnosis keys) for the last 14 days to the Back-End server. The Back-End server will then collect and group these keys and sends them periodically to other participating devices or upon a request from the participating device, so these could do a risk assessment of infection see ETSI GS E4P-008 [2].

[DCD-01]: For decentralized contact tracing protocol, contact data storage function shall store diagnosis keys and ephemeral IDs received from other devices with recorded proximity event.

[DCD-02]: For decentralized contact tracing protocol, contact data storage function shall store data received from other devices with recorded proximity event including their ephemeral ID, timestamp and Bluetooth LE signal attenuation.

[DCD-03]: For decentralized contact tracing protocol, contact data storage function shall send their own diagnosis keys to the Back-End System (see ETSI GS E4P-008 [2]) when the authorised user has given its consent.

[DCD-04]: For decentralized contact tracing protocol, contact data storage function after receiving diagnosis keys of diagnosed users from the Back-End System (see ETSI GS E4P-008 [2]) shall derive the corresponding ephemeral IDs of diagnosed users and compare them with ephemeral IDs from the stored proximity events to determine whether the user was in proximity with an diagnosed user, and use the timestamp and Bluetooth LE signal attenuation to evaluate the risk of infection.

[DCD-05]: For decentralized contact tracing protocol, contact data storage function based on this information shall also store the report of contact diagnosis calculation. These data should be stored for a predefined period, e.g. 14 days.

[DCD-06]: For decentralized contact tracing protocol, contact data storage function shall be isolated from different applications, and shall meet security and privacy requirements.

5.3.3 Centralized approach

5.3.3.1 Robert

In the centralized protocols the participating devices get pseudo-random ephemeral identifiers from the Back-End (see ETSI GS E4P-008 [2]) and broadcast those using Bluetooth LE beacons. In parallel, the devices also listen for the beacons of mobile devices in proximity and store the related broadcast ephemeral identifiers with a timestamp and the signal attenuation to estimate the exposure.

In the event of an infection, the user can decide to report his/her infection in the mobile application. When reported, the mobile application would self-upload the representation of the stored ephemeral identifier of mobile devices in proximity for the last 14 days on the back end server. The back end server will then assess the risk of infection according to the epidemiological criteria for the ephemeral identifier uploaded. And where the risk is confirmed, it will send a notification to the mobile application that was using the related ephemeral identifier.

5.3.3.2 ROBERT protocol

During registration and then regularly, i.e. every M epochs (value to be defined), each registered application App_A connects to the server in order to obtain a list of the M $\{EBID_{A,i}, ECC_{A,i}\}$ pairs for the M following epochs. This list is stored locally on the device and is used for HELLO message broadcasting.

App_A also continuously collects HELLO messages sent by nearby devices running the same application. Upon receiving $HELLO_{B,i}$, the App_A: parses $HELLO_{A,i}$ to retrieve ecc_B (8 bits), $ebid_B$ (64 bits), $time_B$ (16 bits) and mac_B (40 bits); obtains a 32-bit NTP "Seconds" timestamp, $time'_A$; 3. checks that $time_B$ and $time'_A$ are sufficiently close; and stores in its LocalProximityList the following pair: $(HELLO_{B,i}, time'_A)$. Entries in LocalProximityList are automatically deleted after CT Days (the value CT needs to be defined with the health authority).

Not all entries in the LocalProximityList will be uploaded to the server in case the user is tested positive. Entries corresponding to contacts that are significantly too short or too far will be filtered out, in order to comply with the data minimisation principle. Only those entries that can be useful to compute a risk exposure are uploaded.

5.3.3.3 DESIRE protocol

Unlike ROBERT, an application periodically generates a random value X and computes g^X that constitutes its EBID during that period. This is a purely local process that is performed at the beginning of each 15 minutes period.

The application then:

- collects EBIDs of encountered devices;
- computes PET tokens from collected EBIDs if certain conditions are satisfied (e.g. in terms of contact length and received signal strength, etc.): two PETs are generated per encounter, one for exposure request, the other one for an upload of exposed PETs if user is tested COVID+;
- stores the generated PET tokens in two local lists: the RTL list, used for status requests and the ETL gathering exposed PETs if user is tested COVID+. In case of the ETL list, metadata (timing, received signal strength, transmission gain, etc.) is stored along with each PET.

The PETs contained in the RTL list are transmitted during each exposure status request (potentially they could be stored in the backend server to avoid uploading a PET several times, depending on the implementation choices).

The PETs contained in the ETL are uploaded, along with metadata, in case the user is tested COVID+.

For both the RTL and ETL lists, PETs corresponding to encounters that are more than 14 days old, are automatically removed, along with the metadata.

5.4 User experience and usability

This clause describes user experience and usability requirements for the user experience function defined as part of the mobile application in the device, applicable for both decentralized and centralized design approach.

NOTE: User experience function requirements applicability could be limited by the device capabilities.

[DUI-01]: User experience function in the mobile application shall inform the user that the automatic mobile application update should be enabled in the device settings.

[DUI-02]: User experience function in the mobile application shall inform the user when an automatic mobile application update took place.

[DUI-03]: User experience function in the mobile application shall implement accessibility requirements defined in ETSI EN 301 549 [4].

[DUI-04]: User experience function in the mobile application shall notify the user if the battery level is below the limit, which can guarantee the service for at least another hour.

[DUI-05]: Mobile application should be available on the most common mobile device operating systems.

[DUI-06]: User experience function in the mobile application may inform users identified at risk, if required by Health Authority.

[DUI-07]: User experience function in the mobile application shall inform the user about the data sharing policy and it shall request a consent for data sharing.

[DUI-08]: User experience function in the mobile application shall inform users identified at risk with a maximum delay of 12 hours, if required by Health Authority (see [DUI-07]).

[DUI-09]: User experience function in the mobile application shall inform the user about potential infection due to a close contact with an diagnosed user and advise on the next steps. This shall be determined by the Health Authority and may include information on self-quarantine, what symptoms to look for and what to do if symptoms develop.

[DUI-10]: User experience function in the mobile application, shall inform the user at the moment of the mobile application installation or at first use, that it may be contacted by the Health Authority.

[DUI-11]: User experience function in the mobile application, shall inform the user at the moment of the mobile application installation or at first use, about the data which will be collected by the mobile application and could be transmitted to the Back-End system and how and why their data is processed and used.

[DUI-12]: User experience function in the mobile application shall make use of the user authentication methods available in the mobile device (e.g. user authentication to unlock the screen or access sensitive data).

[DUI-13]: User experience function in the mobile application shall notify the user if the version of the device operating system is not suitable to support the mobile application and it shall recommend to update the operating system if possible.

[DUI-14]: User experience function in the mobile application shall implement user friendly methods for a confirmed case data input to ensure high usability and avoid errors (e.g. using QR code).

[DUI-15]: User experience function in the mobile application shall provide a procedure for reporting errors and vulnerabilities in the contact tracing system.

[DUI-16]: User experience function in the mobile application shall notify the user if the epidemic is considered to be over by the Health Authority and then shall erase all its data stored on the mobile device and disable automatically.

[DUI-17]: User experience function in the mobile application shall allow the user to uninstall or disable the mobile application at any time.

[DUI-18]: User experience function in the mobile application shall not allow its user to discover the identity of any diagnosed user.

[DUI-19]: User experience function in the mobile application shall not allow its user to discover close contact with a diagnosed user in the present time.

[DUI-20]: For the centralized contact tracing protocol, if the device uses more than one phone number (e.g. more than one Subscriber Identity Module is used in the device), the user experience function in the mobile application should indicate the phone number used for the authentication.

[DUI-21]: For the centralized contact tracing protocol, the user experience function in the mobile application shall provide the user the capability to upload ephemeral identifiers to the contract tracing system.

6 Requirements mapping to device functions and interfaces

Based on the discussion of different device mechanisms in previous clauses and corresponding definitions of technical device requirements, this clause maps those technical requirements to the device functions and interfaces introduced in the reference device architecture (clause 4).

Table 4: Requirement Mapping Table

Device function	Requirement number	Device interface (Reference point - UD, DB, DD)	Described in clause
Contact proximity detection	[DCP-01] - [DCP-08]	Contact proximity detection interface (DD)	5.1
Anonymous contact identification	[DCI-01] - [DCI-02]	Contact proximity detection interface (DD)	5.2
Contact data storage	[DCD-01] - [DCD-06]	Back-End interface (DB)	5.3
User interface	[DUI-01] - [DUI-21]	User interface (UD)	5.4

Annex A (informative): Matching with ETSI GS E4P 003 'Requirements for Pandemic Contact Tracing Systems using mobile devices'

Table A.1

Device requirement	System requirement (from ETSI GS E4P 003 [1])	Decentralized approach	Centralized approach	
		Applicable to GAEN	Applicable to ROBERT	Applicable to DESIRE
Contact proximity detection				
[DCP-01]	[HL-MD-01]	Yes		
[DCP-02]	[HL-MD-01]	No		
[DCP-03] VOID				
[DCP-04]	[HL-MA-03]	Yes		
[DCP-05]	[HL-MA-03]	Yes	Yes	
[DCP-06]	[HL-MA-03]	Yes		
[DCP-07]	[HL-MA-03]	Yes		
[DCP-08]	[HL-MD-01], [HL-MD-02]	Yes	Yes	Yes
Anonymous contact identification				
[DCI-01]	[HL-MA-06]	Yes		
[DCI-02]	[HL-MA-06]	Yes		
Contact data storage				
[DCD-01]	[HL-MA-05]	Yes		
[DCD-02]	[HL-MA-07]	Yes		
[DCD-03]	[HL-PV-09]	Yes		
[DCD-04]	[HL-MA-05]	Yes		
[DCD-05]	[HL-PV-08], [HL-PV-11]	Yes		
[DCD-06]	[HL-SE-11], [HL-PV-08], [HL-PV-12]	Yes		
User interface				
[DUI-01]	[HL-GE-02], [HL-GE-03], [HL-GE-07]	Yes	Yes	Yes
[DUI-02]	[HL-GE-07]	Yes	Yes	Yes
[DUI-03]	[HL-UX-01]	Yes	Yes	Yes
[DUI-04]	[HL-UX-02]	Yes	Yes	Yes
[DUI-05]	[HL-GE-09]	Yes	Yes	Yes
[DUI-06]	[HL-MA-13]	Yes	Yes	Yes
[DUI-07]	[HL-PV-02]	Yes	Yes	Yes
[DUI-08]	[HL-MA-15]	Yes	Yes	Yes
[DUI-09]	[HL-MA-16], [HL-MA-17]	Yes	Yes	Yes
[DUI-10]	[HL-MA-19]	Yes	Yes	Yes
[DUI-11]	[HL-MA-20]	Yes	Yes	Yes
[DUI-12]	[HL-MA-21]	Yes	Yes	Yes
[DUI-13]	[HL-MA-25]	Yes	Yes	Yes
[DUI-14]	[HL-IN-02], [HL-SE-13]	Yes	Yes	Yes
[DUI-15]	[HL-SE-20]	Yes	Yes	Yes
[DUI-16]	[HL-PV-01]	Yes	Yes	Yes
[DUI-17]	[HL-PV-03]	Yes	Yes	Yes
[DUI-18]	[HL-PV-13]	Yes	Yes	Yes
[DUI-19]	[HL-PV-13]	Yes	Yes	Yes
[DUI-20]	[HL-MA-21], [HL-SE-06]	No	Yes	Yes
[DUI-21]	[HL-PV-09]	No	Yes	Yes

History

Document history		
V1.1.1	May 2021	Publication