ETSI TR 133 871 V12.0.0 (2014-10)

Universal Mobile Telecommunications System (UMTS);
LTE;
Study on Security for WebRTC IMS Client access to IMS
(3GPP TR 33.871 version 12.0.0 Release 12)

Reference
DTR/TSGS-0333871vc00

Keywords
LTE,SECURITY,UMTS

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, 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.

Foreword

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**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.

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, I.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The goal of WebRTC IMS Client access to IMS is to significantly expand the pool of clients able to access IMS.
The present document contains a study on security issues following the potential modifications of the IMS architecture and stage 2 procedures as required by the support of WebRTC IMS Client access to IMS.

For this purpose the present document is addressing:

- WebRTC IMS Client authentication mechanisms, including the re-use of existing IMS authentication mechanisms from WebRTC IMS Clients;
- Required enhancements to IMS media plane security;
- Control plane security related aspects.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 22.228: "Service requirements for the Internet Protocol (IP) multimedia core network subsystem (IMS); Stage 1".

[3] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[4] 3GPP TR 23.701: "Study on the Support of WebRTC IMS Client access to IMS".

[5] 3GPP TS 33.203: "3G security; Access security for IP-based services".

[6] 3GPP TS 33.328: "IP Multimedia Subsystem (IMS) media plane security".

[7] W3C Web Real-Time Communications Working Group, http://www.w3.org/2011/04/webrtc-charter.html

[8] IETF Real-Time Communication in WEB-browsers Working Group, http://tools.ietf.org/wg/rtcweb/

[9] IETF RFC 5763: "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)".

[10] draft-ietf-rtcweb-security: "Security Considerations for WebRTC".

[11] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".

[12] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)".

[13] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[14] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

[15] 3GPP TS 29.228: "IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signalling flows and message contents".

[16] 3GPP TS 24.292: "IP Multimedia (IM) Core Network (CN) subsystem Centralized Services (ICS); Stage 3".

[17] IETF RFC 5764: "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)".

[18] draft-ietf-rtcweb-data-protocol: "RTCWeb Data Channel Protocol ".

[19] draft-ejzak-dispatch-webrtc-data-channel-sdpneg: "SDP-based WebRTC data channel negotiation".

[20] RFC 6714: "Connection Establishment for Media Anchoring (CEMA) for the Message Session Relay Protocol (MSRP)".

[21] RFC 2617: "HTTP Authentication: Basic and Digest Access Authentication".

[22] 3GPP TR 33.830: "Study on Firewall traversal (Stage 2)".

[23] IETF RFC 4169: "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA) Version-2".

[24] IETF RFC 3310 (2002): "HTTP Digest Authentication Using AKA". April, 2002.

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply.
A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Web Real-Time Communications (WebRTC): A set of browser extensions enabling web applications to define real-time services.

WebRTC IMS Client (WIC): A WebRTC-capable browser running a JavaScript application that allows a user to access IMS services**.**

3.2 Symbols

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

Cx	Reference Point between a CSCF and an HSS
Gm	Reference Point between a UE and a P-CSCF or between an IP-PBX and a P-CSCF
Iq	Reference Point between the IMS Application Level Gateway (ALG) (IMS-ALG) and the IMS Access Gateway (IMS-AGW)
Mb	Reference Point between a UE and IP network services used for user data transport
Mw	Reference Point between a CSCF and another CSCF
W1	Reference Point between a WIC and WWSF
W2	Reference Point between a WIC and eP-CSCF
W3	Reference Point between a WIC and eIMS-AGW

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply.
An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

DTLS-SRTP	Datagram Transport Layer Security SRTP
eP-CSCF	P-CSCF enhanced for WebRTC
eIMS-AGW	IMS-AGW enhanced for WebRTC
ICE	Interactive Connectivity Establishment
NAT	Network Address Translation
P-CSCF	Proxy CSCF
RTP	Real-time Transport Protocol
S-CSCF	Serving CSCF
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SRTP	Secure RTP
WebRTC	Web Real-Time Communication
WIC	WebRTC IMS Client
WWSF	WebRTC Web Server Function
WAF	WebRTC Authorization Function

4 Overview

4.1 WebRTC

4.1.1 General

Web Real-Time Communication (WebRTC) is specified by the W3C WebRTC WG [7] in collaboration with the IETF RTCWeb WG [8]. Although it is still work in progress, the technology has already been implemented in many different browsers. As W3C specifies the API and IETF the protocols, the IETF specifications are likely to be more relevant for the WebRTC IMS Client access to IMS work.

4.1.2 WebRTC control plane

The WebRTC control plane is sent over HTTP/WebSocket and is controlled by the WebRTC IMS Client (WIC) application of the UE. While HTTP is a request-response protocol, WebSocket provides a full-duplex communication channel over TCP. Current WebRTC specifications [7] do not specify any control-plane protocol to establish the connection between WIC peers. The WIC application can implement any signalling protocol such as SIP or RESTful HTTP or JSON over WebSocket. The WIC exchanges WebRTC User plane parameters with the peer WIC over the chosen signalling protocol. These parameters are retrieved from the WebRTC compliant browser in the form of SDPs.

WebRTC compliant browser shall support the following security requirements specified by the IETF RTCWeb WG (see [10]):

- DTLS-SRTP shall be used. DTLS certificate fingerprint are shared with the peer DTLS endpoint. This fingerprint binds the DTLS key exchange in the media plane to the WebRTC control plane.
- ICE shall be used. ICE candidates are needed for the WebRTC media plane to traverse through firewalls and NATs.
- The WebRTC control plane that transports SDP between two W2 end points shall be integrity-protected.

4.1.3 WebRTC user plane

The WebRTC user plane consists of media channels for audio and video and data channels for peer-to-peer communication of arbitrary data. The user plane is controlled by the WebRTC compliant browser and therefore much more standardized. Some security relevant requirements (see [10]) on the WebRTC compliant browser are:

- All channels shall use STUN/TURN/ICE to traverse NATs and firewalls;
- Media channels shall use SRTP with keys provided by DTLS-SRTP;
- Data channels shall use SCTP over DTLS.

An overview of the WebRTC protocol layers for the user plane can be seen in Figure 4.1.3-1 and Figure 4.1.3-2.

Figure 4.1.3-1 shows the protocol architecture in the scenario that the restrictive firewall is not existent between WIC and the peer, and all the user plane packets are transported over the UDP. In this architecture, STUN and ICE layer are used to support NAT traversal, STUN layer performs WIC public address collection and connectivity check, ICE layer handles the IP address changes during session without needing interaction with the upper protocol layer.

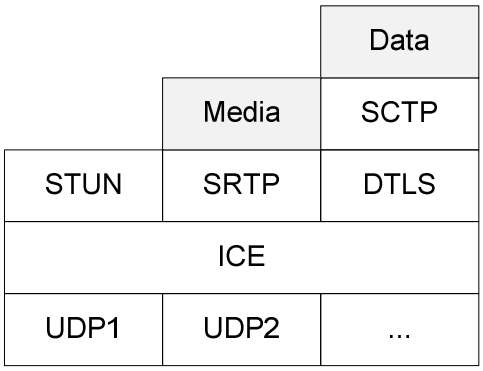

Figure 4.1.3-1: WebRTC user plane protocol layers for supporting NAT traversal

The Figure 4.1.3-2 shows the protocol architecture in the scenario that the user plane packets are blocked by the restrictive firewall between WIC and the peer, and the TURN server is used for relay the media. In this architecture, TURN layer performs the relay path creation and encapsulates both media and data UDP packets into TURN payloads, the TURN packets are transported over TCP or TLS.

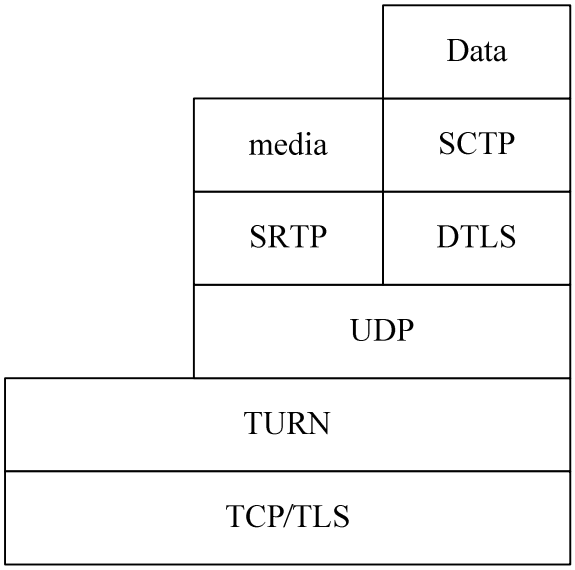

Figure 4.1.3-2: WebRTC user plane protocol layers for supporting restrictive firewall traversal

4.2 WebRTC IMS Client access to IMS

4.2.1 Overview

A WIC (WebRTC IMS Client) is a WebRTC-capable browser running a web application that allows a user to access IMS services. The web application (written in HTML/CSS/JavaScript) is offered by the IMS operator or by a third party. The support of WebRTC IMS Client access to IMS significantly expands the pool of clients able to access IMS.

The WebRTC client authenticates to the IMS via the WebRTC control plane function, using either traditional IMS credentials e.g. SIP Digest username/password, or some form of web credentials e.g. OAuth access token.
In the latter case, the WebRTC control plane function will verify the web credentials and then authenticate to the IMS core on behalf of the user.

4.2.2 Architecture

Figure 4.2.2-1 shows the architecture for WebRTC IMS Client access to IMS as described in TR 23.701 [4].
The WWSF (WebRTC web server function) is the first web server contacted by the user (generally by clicking on a link or entering a URL into the browser). The P-CSCF enhanced for WebRTC (eP-CSCF) is the endpoint for the signalling connection.

Instead of authenticating the IMS client directly using existing IMS authentication methods, the IMS network may choose to authenticate the WIC indirectly using a third party authentication service. In this case the WWSF obtains an authorization token from the WAF (WebRTC Authorization Function) which asserts the user's identity.
The WWSF forwards the token to the WIC which in turn includes it in the SIP register request for verification by the eP-CSCF. The WAF can either authenticate the user itself as part of the token issuance process, or it trusts the user identity supplied by the WWSF. In the latter case the WWSF is assumed to have authenticated the user prior to sending the token request.

The functional split between the WWSF and WAF is consistent with the OAuth 2.0 architecture, where the WWSF corresponds to the client and the WAF corresponds to the authorization server. The ownership of the WWSF and WAF can be split so that the WWSF is maintained by the third party while the WAF is under control of the operator. This is beneficial from a security point of view but involves an administrative overhead for the operator.
Another option is to place both the WWSF and WAF in the third party domain, in which case the functional entities can be physically co-located.

Editor's Note: In the current architecture description in TR 23.701, the WWSF is responsible for providing the web page as well as issuing authorization tokens. By splitting out the token generation functionality into separate function, the WAF, the architecture becomes more in line with the OAuth 2.0 architecture. The functional separation also makes it possible to split the ownership of the WWSF and WAF between the third party and the operator, which is beneficial from a security point of view. The architecture description in TR 23.701 should therefore be updated to include the WAF and the interface W4.

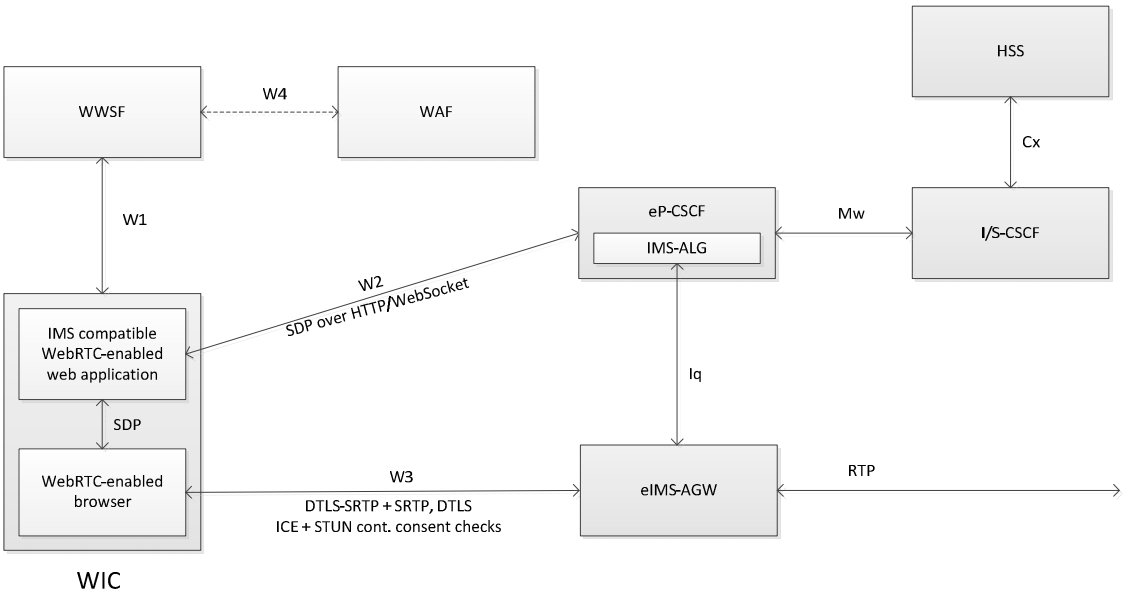

Figure 4.2.2-1: Architecture of WebRTC IMS Client access to IMS

Editor's Note: The ownership of the WAF should be reflected in Figure 4.2.2-1.

5 Assumptions, Risks and Security requirements

5.1 Assumptions

Editor's Note: If needed, this clause will define the underlying assumptions of the work.

Editor's Note: For all registration scenarios, there is the following NOTE:

> "The eP-CSCF can verify that the web-page establishing the signalling connection comes from a trusted domain by inspecting the value of Origin header. This header is inserted by the browser in the WebSocket handshake and in every HTTP request (requires the use of CORS, http://www.w3.org/TR/cors/).
> The protection mechanism works under the assumption that the browser is not under the attacker's control, which means that the contents of the Origin header can be trusted."

It is FFS whether this NOTE should rather be translated into a security assumption on the browser and a requirement on the verification of the origin of the JavaScript code constituting the WIC.

5.2 Risks

5.2.1 Impact of security breach at WWSF on arbitrary IMS subscribers

This subclause deals with a potential security breach affecting the web authentication scheme operated by a third party WWSF.

In the registration scenario described in clause 6.1.2, it would become possible that an attacker in control of a compromised WWSF or authorization server could assert having authenticated a user with an IMPU of the attacker's choice, providing that this IMPU relates to an existing IMS subscription. In this way, the attacker could initiate a WebRTC call with this IMPU as originator and hence impersonate the user legitimately associated with this IMPU. This could have not only financial implications for the user and/or the IMS service provider, but could also damage their reputation or result in legal prosecution, depending on the destination and content of the call. While it is true that the eP-CSCF is tasked with verifying that the WWSF is authorized to allocate IMS identities that it assigns to a WIC the eP-CSCF could not stop this impersonation as any IMPU relating to an existing IMS subscription could be assigned by the WWSF, unless there are restrictions on the IMPUs a WWSF is allowed to assign and the eP-CSCF knows about them.

The impersonation could affect any IMS subscriber, even if they had no business relationship with any third party (e.g. a social network) operating a WWSF, or would not even use WebRTC.

A requirement to address this risk is REQ 2.1 in clause 5.3.

5.2.2 Lack of means to identify potentially compromised WWSF in the IMS core

For the registration scenario described in clause 6.1.2, assume that there is a security breach at one WWSF, or that the behaviour of WebRTC clients authenticated by one WWSF shows some anomalies. The IMS service provider has an interest to isolate the impacts of the security breach without affecting clients associated with other WWSFs. However, this is not possible if the IMS core is lacking relevant information, I.e. the identity of the WWSF that authorized the user to access the IMS core. With such information lacking there is therefore the risk that the IMS core cannot adequately address a potential compromise at the WWSF.

A requirement to address this risk is REQ 2.2 in clause 5.3.

5.2.3 Risks relating to the determination of IMS identities by the WWSF

It has to be assumed for the registration scenario described in clause 6.1.2 that the WWSF can securely authenticate the user's web identity via some web authentication scheme. But even under this assumption, there is a risk that the linkage between the private identity of the user's existing IMS subscription (IMPI) and the authenticated web identity has not

been securely established by the WWSF. (The IMPU(s) are less critical as they are associated with the IMPI in the HSS, so the IMPI-IMPU association cannot be spoofed.) Any technical means addressing how this linkage, once established, could be stored and accessed does not help in establishing this linkage. If the IMPI is determined wrongly then the user will be associated by the IMS core with somebody else's subscription, and the IMS core will have no means of knowing as all the knowledge about the user is embodied in the authentication information received from the WWSF via the WIC. Furthermore, if the user can influence the choice of the IMPI associated by the WWSF with the user's web identity, then the user can impersonate another IMS subscriber in a targeted way.

It is therefore not acceptable that the user simply tells the WWSF about his IMPI as the user could purposefully lie about it. For the same reason, it is not acceptable that the user simply tells the IMS provider about his web identity.

In order to counter the security risk described above, security requirement REQ 2.5 in clause 5.3 shall be fulfilled in any deployment of the registration scenario described in clause 6.1.2.

5.2.4 Risks relating to assignment of IMS identities to WebRTC IMS Client from pool of IMS subscriptions held by WWSF

This risk relates to the third registration scenario described in TS 23.228 [3] and in clause 6.1.3 of the present document. In that scenario, the IMS subscriber is the WWSF, not the user. There is no linkage between the user's web identity that may be authenticated by a third party authentication service and the assigned IMS identities.
This implies that the risks described for the second registration scenario do not apply to the third registration scenario in the same way.

However, the following risks remain:

- A WIC may falsely claim that the WIC is allowed to use one of the IMS subscriptions from the pool owned by the WWSF.

- A (potentially compromised) WWSF may falsely claim to own an IMS subscription, and the related IMS identities, and may issue false authorization information to a WIC allowing the WIC to use this IMS subscription.

5.3 Potential security requirements

Requirements for Support of WebRTC IMS Client access to IMS are specified by SA1 in 3GPP TS 22.228 [2]. Additional potential architectural aspects identified by SA2 are stated in 3GPP TR 23.701 [4].

The following security requirements apply to the first registration scenario described in clause 6.1.1:

- REQ 1.1: For the reference interface W1 (WIC to WWSF), one way authentication (WIC needs to authenticate WWSF) is required. For the interface W2 (WIC to eP-CSCF), mutual authentication is required.

The following security requirements apply to the second registration scenario described in clause 6.1.2:

- REQ 2.1: An IMS service provider shall ensure that a authentication service authenticating a WebRTC IMS Client (WIC) and authorizing it to register with an IMS network using certain IMS identities has been granted the right to do so by the IMS subscriber owning these IMS identities. In case of a potential security breach affecting that authentication service, IMS subscribers that did not grant any right to that authentication service shall not be affected.

- REQ 2.2: An IMS service provider should be able to identify and mitigate security anomalies or security breaches at one entity providing a authentication service selectively, without affecting clients associated with other entities providing a authentication service.

- REQ 2.3: To prevent a third party authentication service from providing authorization information to a WebRTC IMS Client (WIC) without having been authorized by the IMS service provider to do so, an IMS service provider shall be able to identify the granting third party authentication service each time the IMS subscriber registers with the IMS network through the W2 interface. The identity of the third party authentication service shall be determined from the authorization information securely received by the IMS network over W2.

- REQ 2.4: An IMS service provider relying on an authentication service for WebRTC IMS Clients (WIC), shall securely determine from the received authorization information the IMPI and IMPU of the authenticated WIC attempting to register with the IMS network.

NOTE: In a use-case where IMPI is associated with multiple IMPUs, IMPI to IMPU association check when I-CSCF User Registration Query is processed by the HSS, is not enough. For ex., a user who has authenticated to the WWSF as sip:bob-impu1@operator.com but changes "To" field in the W2 REGISTER message to sip:bob-impu2@operator.com, will not be detected by the IMS network. It is therefore necessary to determine IMPU and IMPI of the authenticated user from the received authorization information.

- REQ 2.5: It shall be ensured that the authentication service has enough information to guarantee that the user is entitled to use the IMS private identity IMPI determined from the user's web identity authenticated by the authentication service.

- REQ 2.6: For the interface W1 (WIC to WWSF) mutual authentication is required, unless the user's web identity is authenticated by the WAF, in which case only one-way authentication is required. For the interfaces W2 (WIC to eP-CSCF), and W4, if present, (WWSF to WAF), mutual authentication is required.

The following security requirements apply to the third registration scenario described in clause 6.1.3:

- REQ 3.1: The authentication service shall provide authorization information to the eP-CSCF (possibly via the WIC) that allows the IMS core to ascertain that the WIC in possession of this authorization information is authorized to access IMS using the associated public and private IMS identities presented during registration or retrieved from the authentication service through undefined means.

- REQ 3.2: An IMS service provider shall ensure that the private IMS identity provided in the authorization information from REQ 3.1 belongs to an IMS subscription in the pool of IMS subscriptions uniquely assigned to the WWSF.

- REQ 3.3: For the interfaces W2 (WIC to eP-CSCF), and W4, if present, (WWSF to WAF), mutual authentication is required. For the W1 interface, mutual authentication is required, except for the case of anonymous user. In the case of anonymous user, one way authentication (WIC needs to authenticate WWSF) is required.

Editor's Note: This clause will define additional potential security requirements.

Editor"s Note: It is necessary to make corresponding changes in the requirements of 33.203 Annex WebRTC part.

6 Solutions

6.1 Authentication and Authorization

Editor's Note: This clause is split into two sub-clauses to reflect the use cases mentioned in SA1 TS 22.228 [2] "The authentication of the subscriber can be performed via the WebRTC IMS Client or by a WebRTC server on behalf of a user."

Editor's Note: TR 23.701 describes a third authentication/registration solution in which the eP-CSCF acts as an IP-PBX in static mode of operation. Whether SA3 should study this solution as well depends on the outcome of the SA2 discussions. From a security perspective this solution appears similar to the solution described in 6.1.2.

Editor's Note: SA3 shall validate the registration scenarios and provide additional details related to security aspects of the architecture. In particular, SA3 should verify for all scenarios the security properties of at least the following aspects: the use of TLS, WSS and CORS at the relevant reference points; the use of IMS digest, TNA, and/or potentially other IMS authentication mechanisms; how to provide IMS digest authentication and registration information to the WIC; the required trust relationships between functional entities for the scenarios; and whether there are any constraints on network locations of the functional entities of the architecture in the scenarios.

Editor's Note: The feasibility of the solutions should take into account the dependencies on the browser limitations.

6.1.1 Authentication of WebRTC IMS Client with IMS subscription re-using existing IMS authentication mechanisms

6.1.1.1 General

Editor's Note: It is assumed that the WebRTC IMS Client has access to IMS credentials and uses these to authenticate to the IMS.

In this scenario it is assumed that the user has a subscription with an individual IMPU and uses an IMS authentication mechanism (e.g. IMS digest) to authenticate with IMS. The eP-CSCF is assumed to relay the authentication information so that the message flows are unchanged.

Editor's Note: Access to the (U)SIM and the AKA algorithm from JavaScript is currently not supported in today's browsers (without requiring browser modifications or installation of proprietary plug-ins).

6.1.1.2 Use of SIP Digest credentials

In this scenario that the WebRTC IMS Client implements the SIP Digest algorithm and sends the authentication information to the eP-CSCF. The use of SIP Digest in IMS is specified in Annex N of TS 33.203 [5].

NOTE 1: The use of SIP Digest breaks the 3GPP security requirement mandating IMS AKA to connect to IMS when using a 3GPP access network, see 3GPP TS 33.203 [5].

Figure 6.1.1.2-1 shows the registration flow. In this figure SIP over secure WebSocket is used between the WebRTC IMS Client and the eP-CSCF. Other protocols (e.g. HTTP RESTful or JSON over WebSocket) can also be used as long as it is able to relay the IMPI and the digest challenge, challenge-response, and auth-info values.

It is recommended to maintain a clear separation between WebRTC IMS Clients and regular IMS UEs.
A user accessing IMS from a WebRTC IMS Client should be assigned a separate subscription in the HSS with a unique IMPI and SIP Digest password. In this way a compromised password will have an isolated impact and only affect the WebRTC IMS Client.

The solution requires that the IMS identity and SIP Digest password are made available to the JavaScript in the WebRTC IMS Client.

The entities that have access to the IMS identity and SIP Digest password, and thus needs to be trusted by the operator, are the user, the browser, the WWSF, and the IMS core network. SIP Digest is therefore only intended to be used when the WWSF is controlled by the operator or a 3rd party trusted by the operator.

NOTE 2: It is assumed that the credentials are entered by the user via the web GUI or retrieved from the WWSF over HTTPS. Note that the latter option requires that WWSF has authenticated the user previously.

NOTE 3: Unless the SIP Digest password or the intermediate hash value H(A1) (see RFC 2617 [21]) is stored in the WIC, the password needs to be re-obtained each time a re-registration is performed. If the password is entered manually and if re-registrations occur often, this will result in a negative user experience. This can be avoided by storing the SIP Digest password or H(A1) in the WIC after the initial registration procedure. Ensuring the confidentiality of the SIP Digest password or H(A1) during storage is at the discretion of the implementation and is outside the scope of 3GPP.

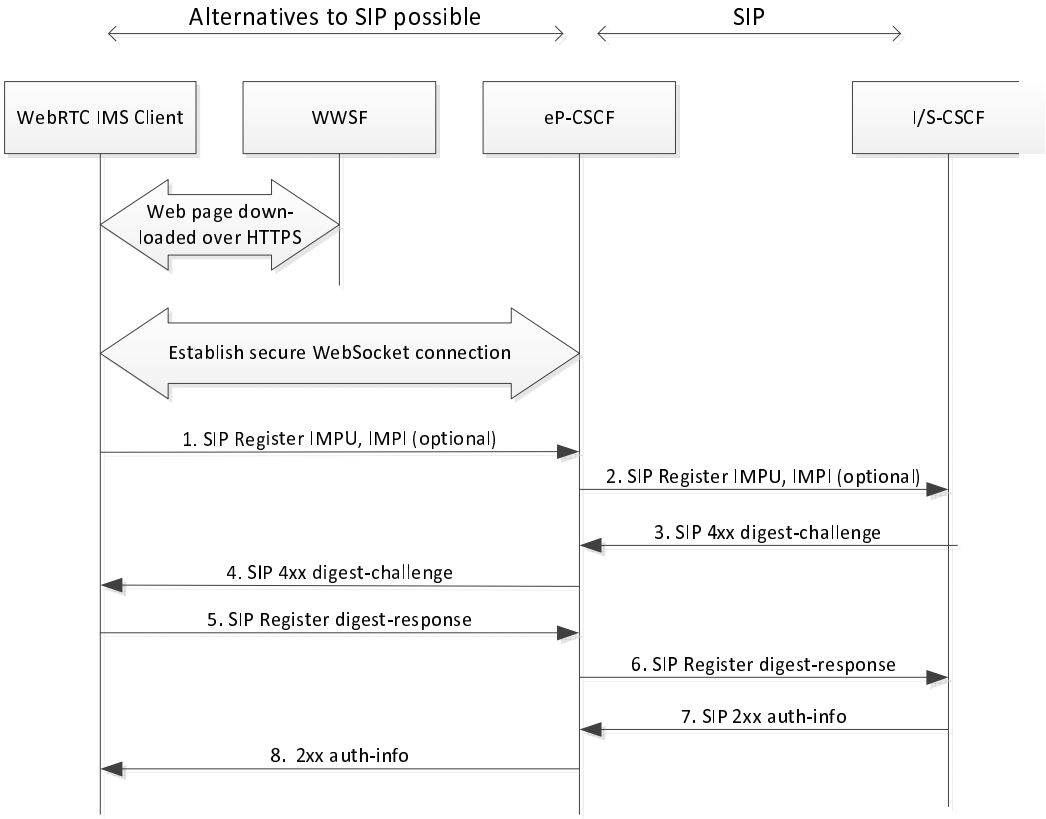

Figure 6.1.1.2-1: WebRTC client authentication using SIP Digest

NOTE 4: The eP-CSCF can verify that the web-page establishing the signalling connection comes from a trusted domain by inspecting the value of Origin header. This header is inserted by the browser in the WebSocket handshake and in every HTTP request (requires the use of CORS, http://www.w3.org/TR/cors/). The protection mechanism works under the assumption that the browser is not under the attacker's control, which means that the contents of the Origin header can be trusted.

NOTE 5: When WIC sends IMPU for the REGISTER request, the IMPI can be derived from the IMPU (refer to Note 2a in Annex N.2 of TS 33.203 [5]), so it is optional that WIC sends the IMPI to the eP-CSCF.

6.1.1.3 Use of IMS AKA

When the WIC has access to the USIM/ISIM in the UE, IMS AKA scheme is used for authenticating WebRTC IMS Client, as described in figure 6.1.1.3-1. The IMS AKA procedure is performed as specified in TS 33.203 [5] clause 6.1, with the usage of HTT Digest AKAv2 as defined in [23] (instead of HTTP Digest AKA) and without security association set-up. The protection of IMS signalling between the WIC and the eP-CSCF is provided by the secure WebSocket connection. The WebRTC IMS Client forwards necessary IMS AKA information to the UICC application in charge of the IMS AKA authentication for WebRTC. This UICC application sends back the results of the AUTHENTICATE command executed to perform the IMS AKA authentication, as defined in section 8 of 3GPP TS 33.203 [5].

Editor's Note: Due to the non-setup of security association in the re-use of IMS AKA for WebRTC, the description of WebRTC access to IMS in 3GPP CT1 and CT4 specifications will detail the changes applied to the SIP messages parameters used to set up the security mode in regular IMS.

It is optional to have in the UICC an ISIM application that would be dedicated to WebRTC usage in order to maintain a clear separation between WebRTC Client and regular IMS UEs. This ISIM application dedicated to WebRTC could have separate subscription in the HSS (with unique IMPI and key K). In this way an attack will have an isolated impact and only affect the WebRTC IMS Client.

The ME needs to be able to apply access control policy to the WIC before granting the access to the UICC application in charge of the IMS AKA authentication for WebRTC.

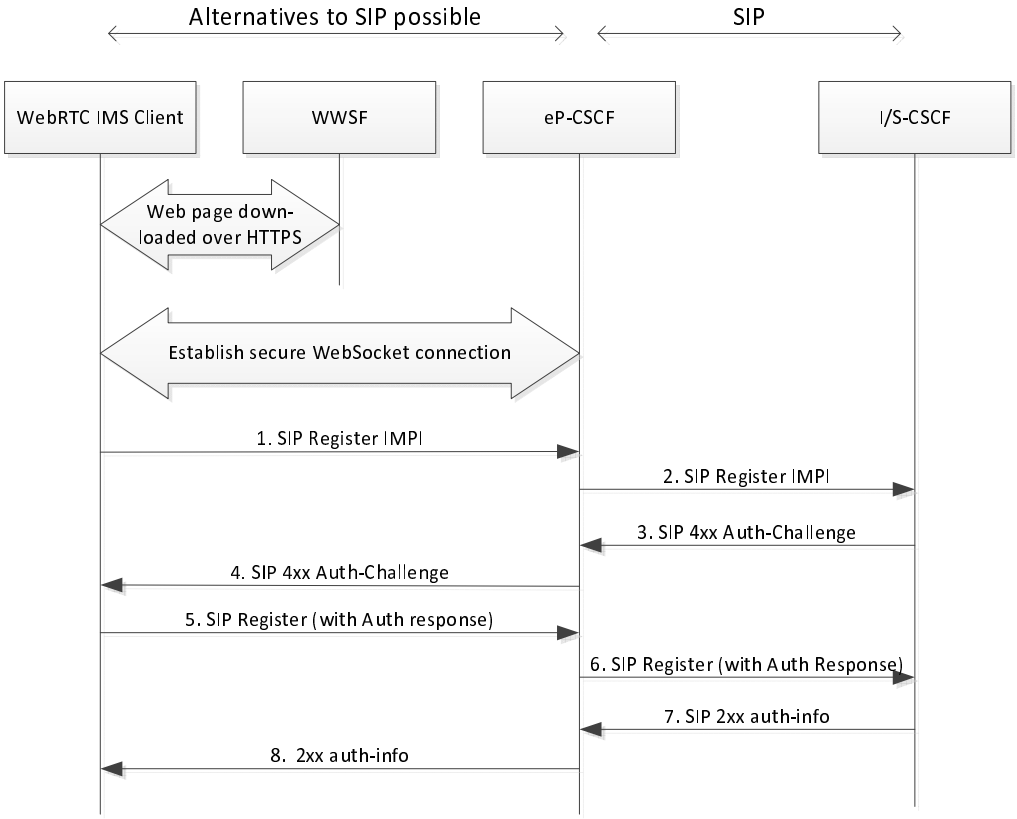

Figure 6.1.1.3-1: WebRTC client authentication using IMS AKA

- Web page download from WWSF

From within a WebRTC-enabled browser, the user accesses a URI to the WWSF to initiate an HTTPS connection to the WWSF. The TLS connection provides one-way authentication of the server based on the server certificate. The browser downloads and initializes the WIC from the WWSF.

- Establishment of secure Web socket connection between WIC and eP-CSCF

The WIC opens a WSS (secure Web Socket) connection to the eP-CSCF. The TLS connection provides one-way authentication of the server based on the server certificate.

NOTE 1: The eP-CSCF can verify that the web-page establishing the signalling connection comes from a trusted domain by inspecting the value of Origin header. This header is inserted by the browser in the WebSocket handshake and in every HTTP request (requires the use of CORS, http://www.w3.org/TR/cors/). The protection mechanism works under the assumption that the browser is not under the attacker's control, which means that the contents of the Origin header can be trusted.

NOTE 2: Precision on how the ME could apply access control policy to restrict access to UICC is at the discretion of the ME implementation and is left out of scope of the present 3GPP release.

- **IMS AKA Procedure** (from Step 1 to Step 8)

The IMS AKA procedure is performed as specified in section 6.1 of 3GPP TS 33.203 [5] with the usage of HTTP Digest AKAv2 as defined in RFC 4169 [23] (instead of HTTP Digest AKA defined in RFC 3310 [24]) and without security association set-up.

The WebRTC IMS Client forwards necessary IMS AKA information to the UICC application in charge of the IMS AKA authentication for WebRTC.

The ME applies access control policy to the WIC before granting the access to the UICC application in charge of the IMS AKA authentication for WebRTC.

This UICC application sends back the results of the AUTHENTICATE command executed to perform the IMS AKA authentication, as defined in section 8 of 3GPP TS 33.203 [5]. After successful execution of the AUTHENTICATE command, the ME securely derives the HTTP Digest password as described in RFC 4169 [23] using algorithm name equal to AKAv2-SHA-256 and associated pseudo-random function (PRF) as defined in RFC 4169 [23]. The algorithm value equals to SHA-256 in RFC 3310[24]. The WebRTC IMS Client uses this HTTP Digest password to provide the authentication response in the SIP Register message. The WIC shall not have access to the keys CK and IK.

The S-CSCF shall also derive the HTTP Digest password as described in RFC 4169 [23] using algorithm name equal to AKAv2-SHA-256 and associated pseudo-random function (PRF).

Editor's Note: It is ffs how solution 2 in X.2.3 can co-exist with the authentication schemes currently taken into account in Annex P. In particular, it is ffs how the P-CSCF shall react with respect to integrity protection indicators and how the S-CSCF can know that Digest AKAv2 is required.'

6.1.2 Authentication of WebRTC IMS Client with IMS subscription using web credentials

6.1.2.1 General

Editor's Note: It is assumed that the user does not have access to IMS credentials and that the eP-CSCF authenticates to the IMS on behalf of the user. The user may use some other form of credentials to authenticate to the eP-CSCF.

In this scenario it is assumed that the user has a subscription with an individual IMPU but uses a web identity and authentication scheme to authenticate with a third party authentication service.

NOTE 1: The third party authentication service is the function that performs authentication of the user and provides the token to the user. This term does not imply anything about a function split among WWSF, authorization server, etc. in providing this service.

NOTE 2: There is a pre-condition that the authentication service should have a mapping between the user's web identity and IMS identity and synchronize the mapping of users' web identities and IMS identities with the operator in advance. For example, both of them can perform the mapping according to the agreement they made.

The third party authentication service in turn issues authentication information to the WebRTC IMS Client (WIC) that the WIC presents to the eP-CSCF. The WWSF determines the IMS identities of the user based on a linkage of the IMS identities and the user's web identity (e.g. via database lookup or other translation means). The eP-CSCF verifies the

authentication information. Provided the validation of the authentication information is successful, the eP-CSCF performs the IMS registration on behalf of the user.

The linkage of the IMS identities and the user's web identity carries a security risk that is described in clause 5.2.3 of the present document. Clause 5.3 contains a security requirement countering this risk. Examples of mechanisms realizing this security requirement can be found in Annex C. These examples are not subject to 3GPP Technical Specifications.

6.1.2.2 Use of Trusted Node Authentication (TNA)

The scenario allows applying Trusted Node Authentication (TNA) specified for IMS in Annex U of TS 33.203 [5]. While TNA was specified mainly for interworking with the CS access domain, the technology is access and protocol independent. The requirements include that the trusted node (I.e. eP-CSCF) can authenticate the user by means of authentication information received from the third party authentication services, that the trusted node can provide interworking between the IMS domain and the other domain, in which the WWSF resides, if necessary, and as the name applies, that the operator trusts the WWSF and the authentication provided by the third party authentication service. It is clear that the operator trusts the eP-CSCF, performing the role of trusted node in TNA, as the eP-CSCF resides in the operator network, according to TR 23.701 [4].

The token is sent to the WebRTC IMS Client which includes it in the initial registration request to the eP-CSCF. Provided the token verification is successful, the e-PCSCF will proceed with the IMS registration of the user using TNA.

The signalling flow for when the Trusted Node performs registration on behalf of the WebRTC IMS Client is shown in Figure 6.1.2.2-1. In this figure SIP over secure WebSocket is used between the WebRTC IMS Client and the eP-CSCF. Other protocols (e.g. HTTP RESTful or JSON over WebSocket) can also be used. The signalling between the Trusted Node and the rest of the IMS core is unchanged from the signalling flow in Annex U of TS 33.203 [5] in figure 6.1.2.2-1. The REGISTER message may, however, have to be enhanced with an additional parameter to satisfy the requirements from clause 5 of the present report.

OAuth 2.0 (IETF RFC 6749 [13]) may be used an example authentication protocol between the WebRTC IMS Client and the eP-CSCF. Annex D specifies the mapping of OAuth 2.0 roles to IMS WebRTC Architecture. In the following paragraphs of this section, role assignment based on Model A is assumed for the purpose of discussion. In model A, the user corresponds to the resource owner, the WWSF corresponds to the client, the WAF corresponds to the authorization server, and the IMS network (eP-CSCF) corresponds to the resource server.

In the OAuth 2.0 protocol the WWSF first obtains an access token from the WAF which authorizes it to access the user's IMS account. The token is then sent to the WebRTC IMS Client which includes it in the initial registration request to the eP-CSCF. Provided the token verification is successful, the e-PCSCF will proceed with the IMS registration of the user using TNA.

The access token is associated with a specific user and WWSF and has a certain lifetime and scope. This authorization information can either be encoded into the token itself and verifiable through a signature or MAC (so called self-contained token), or retrieved as part of the validation response if the validation is performed against the WAF.
If the token is self-contained and has a signature or MAC, the eP-CSCF can verify the token using the public key or pre-shared key of the WAF. If the token is a handle and has no signature or MAC, eP-CSCF needs to send token validation message to the WAF and verify the response from the WAF. The token validation protocol and interface is not defined in this release.

NOTE 1: In the present 3GPP release only the W2 interface is specified; how the WWSF obtains the token and how it is made available to the WebRTC IMS Client is left out of scope of the present 3GPP release.

NOTE 2: In the present 3GPP release the token format and verification procedure is left out of scope. It is assumed that the eP-CSCF can check the validity of the token and obtain the IMPI, WWSF identity, lifetime, and scope parameters.

NOTE 3: To protect against token disclosure, the W1 and W2 interfaces shall be integrity and confidentiality protected using TLS. This is a mandatory requirement in the OAuth bearer token specification RFC 6750 [14].

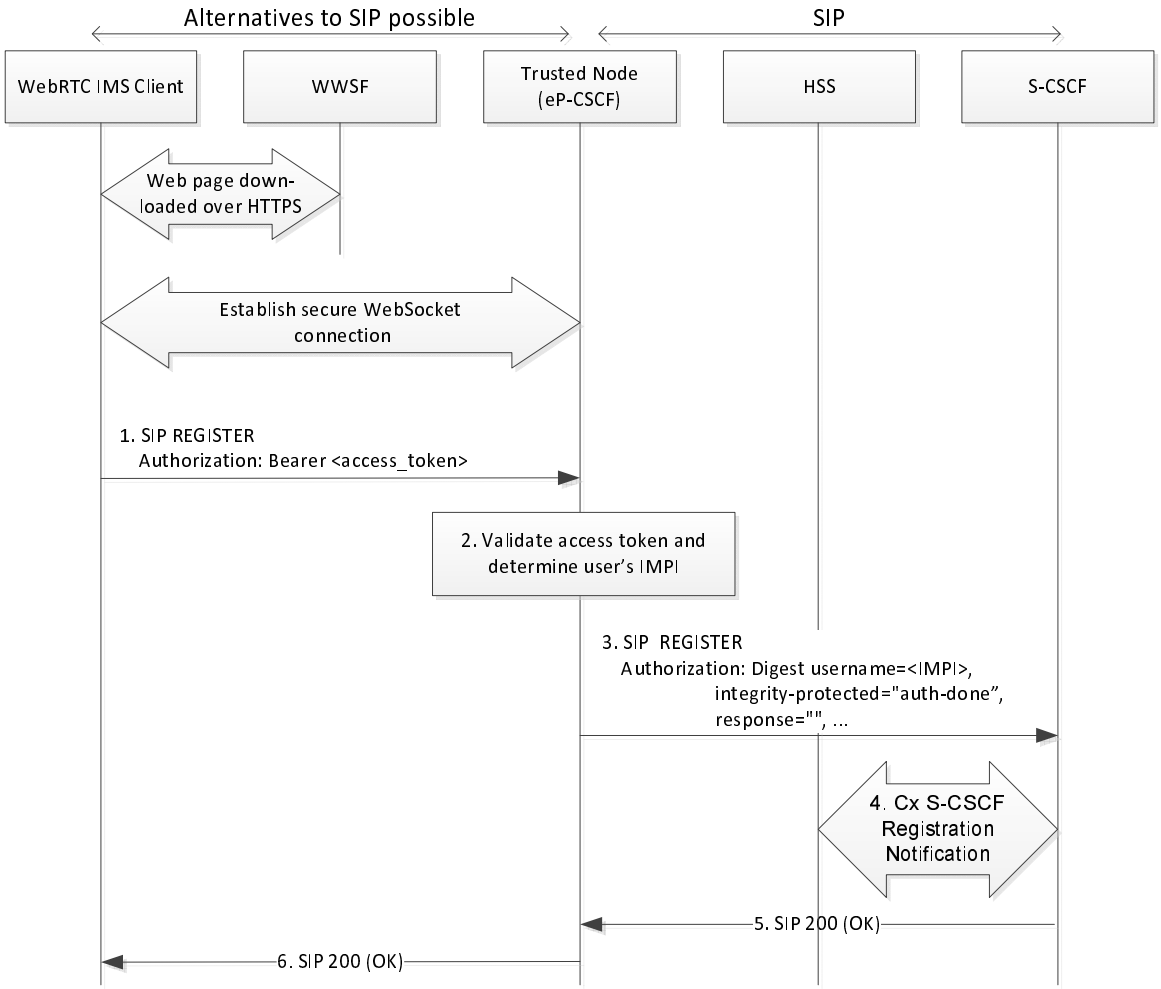

Figure 6.1.2.2-1: Trusted Node performs registraton on behalf of the WebRTC client

The details of the signalling flows are as follows:

NOTE 4: If the WAF resides in the operator domain, we can use the WAF control to authorize the WWSF. If WAF is not in the operator domain, then we can use the authorization of the WWSF by S-CSCF and HSS.

0) Token request (WWSF to WAF)

If the WAF resides in the operator domain, when the WAF receives a token request, it can either authenticate the user itself as part of the token issuance process, or it trusts the user identity supplied by the WWSF which is assumed to have authenticated the user prior to sending the token request. Only if this check is successful will the WAF return the authorization token; otherwise, the WAF rejects the request of the WWSF. And if a WWSF is suspected of a security breach the WAF will block all token requests originating from that WWSF.

1) REGISTER request (WebRTC IMS Client to Trusted Node)

The WebRTC IMS Client establishes a secure WebSocket connection with the eP-CSCF and sends a REGISTER request. The Authorization header includes the OAuth 2.0 access token which the WebRTC IMS Client has previously obtained. The access token is of the so called "bearer" token type; see RFC 6750 [14].

NOTE 5: OAuth bearer tokens can be used with signalling protocols that supports the Authorization header defined in RFC 2617 [21], for example SIP and HTTP.

2) Validation of security token at eP-CSCF

The eP-CSCF extracts the access token and validates it in some unspecified manner. If the token is still valid the eP-CSCF obtains the associated authorization information, including the IMPIU of the associated user, the WWSF identity, and the token scope. The eP-CSCF verifies that the scope includes the value "webrtc-ims-client-access-to-ims".

NOTE 6: The realm value "webrtc-ims-client-access-to-ims" is just a placeholder. The final syntax will be defined in the stage 3 specification.

Editor Note: It shall consider how the eP-CSCF can validate the token from the WIC when there is no interface between the WWSF and the trusted node (eP-CSCF). It is desirable to define the token category to solve the problem, and it is FFS in Rel-13.

3) REGISTER request (Trusted Node to S-CSCF)

Provided that the validation in the previous step was successful, the eP-CSCF replaces the Authorization header with a TNA Authorization header and forwards the request to the S-CSCF (via the I-CSCF). The format of the TNA Authorization header is specified in TS 24.292, Clause 6.2 [15], and contains, among others, the user's IMPI, an integrity-protected directive set to auth-done, and an empty response directive. Furthermore, the eP-CSCF includes the identity of the WWSF if the WAF belongs to the third party network.

4) Cx: S-CSCF Registration Notification

Based on the presence of the "integrity-protected" directive set to indicate that authentication has already been performed, the S-CSCF knows that the subscriber has already been authenticated by the Trusted Node.
The S-CSCF informs the HSS that the user has been registered. Upon being requested by the S-CSCF, the HSS will also include the user profile in the response sent to the S-CSCF.
For detailed message flows see TS 29.228 [16].

If the WAF is not inside of the operator domain, the HSS further includes a list of identities of WWSFs outside the IMS provider"s domain allowed for this IMS subscription. When the S-CSCF received an identity of the WWSF from the eP-CSCF, it checks whether the WWSF identity received from the eP-CSCF and the HSS respectively match. If it is, the S-CSCF proceeds with the next step; otherwise, it rejects the registration.
And if a WWSF is suspected of a security breach the S-CSCF will block all registration attempts involving assertions from that WWSF.

5) 200 (OK) response (S-CSCF to eP-CSCF)

The S-CSCF sends a 200 (OK) response to the eP-CSCF (via I-CSCF) indicating that Registration was successful.

Similar to the registration procedure for SIP Digest with TLS, the eP-CSCF associates the IMPI and all successfully registered IMPUs with the TLS Session ID when the 200 (OK) is received.

6) 200 (OK) response (eP-CSCF to WebRTC IMS Client)

The eP-CSCF forwards the 200 (OK) response to the WebRTC IMS Client indicating that Registration was successful.

NOTE 7: The eP-CSCF can verify that the web-page establishing the signalling connection comes from a trusted domain by inspecting the value of Origin header. This header is inserted by the browser in the WebSocket handshake and in every HTTP request (requires the use of CORS, http://www.w3.org/TR/cors/).
The protection mechanism works under the assumption that the browser is not under the attacker's control, which means that the contents of the Origin header can be trusted.

Editor's Note: It is desirable for 3GPP to provide a security mechanism for the interface between WIC and eP CSCF in Rel-12, but it is FFS whether this goal can be achieved in Rel-12. Furthermore, it is FFS, which authentication mechanism to specify. It is also FFS whether this security mechanism should be mandatory to implement, but not mandatory to use, or whether it should just be an example security mechanism. It is agreed that, if SA2 does not provide a full specification of the signalling interface as mandatory to implement, then it only makes sense to have an example security mechanism in SA3. It is not intended to make it mandatory to use. The advantages of such a 3GPP-defined security mechanism for the interface between WIC and eP-CSCF would include ensuring interoperability between WICs and eP CSCFs from a security point of view and ensuring a minimum level of security.

Example countermeasures to satisfy REQ 2.1 from clause 5 are:

The three example countermeasures require that the third party WWSF is only authorized to assign IMS identities from a well-defined set of IMS subscribers that have chosen the option to access the IMS via this third party's web authentication scheme. The countermeasures differ in the enforcement points:

- Control by eP-CSCFs: TR 23.701 [4], Annex A.1.3.3, states: ""The eP-CSCF verifies that the WWSF is authorized to allocate IMS identities that it assigns to a WIC"." This text suggests control by eP-CSCFs.
  In order to enable this verification all eP-CSCFs that may receive assertions (in the form of authorization tokens) issued by a certain third party authentication service have to be provided with the list of the IMS identities that a third party authentication service is authorized to assign. But, considering that several eP-CSCFs can receive assertions issued by one third party authentication service, one eP-CSCF can receive assertions issued by several third party authentication services operated by different third parties, and that these lists would have to be updated dynamically, this solution may be difficult to manage and not scale well. In view of these disadvantages one may want to look at using a different enforcement point, cf. next paragraph.

- The evaluation of the countermeasure control by eP-CSCFs:

  1) All eP-CSCFs shall maintain a list which contains plenty of IMS identities corresponding to the different third party authentication service, the eP-CSCFs may be equipped with an extra entity (e.g. database) to store the list.

  2) When the user roams to the coverage of the different eP-CSCF and registers with this eP-CSCF initially, the eP-CSCF should update the list dynamically.

- Control by S-CSCF and HSS: For each IMS subscription, an HSS entry indicates, which third party authentication service is authorised to assign a given IMS identity. The HSS is the natural repository for subscription-related information. This information is sent to the S-CSCF over Cx during registration.
  The eP-CSCF sends the identity of the third party authentication service to the S-CSCF with the REGISTER message. The S-CSCF can then check whether the third party authentication service identities received from the eP-CSCF and the HSS respectively match.

- The evaluation of the countermeasure control by S-CSCF and HSS:

  1) The HSS is the natural repository for subscription-related information, there is no need to set an extra database to process.

  2) The HSS has stored all users' subscription information so that it has no need to update its database dynamically.

  3) The register message from the WIC to the eP-CSCF contains a new parameter so-called the WWSF identity which means that the SIP signalling needs to change.

  4) The diameter interface between the S-CSCF and HSS needs to be extended as well.

- Control by WAF maintained by operator: This countermeasure assumes that the WWSF resides in the third party domain while the WAF resides in the operator domain. When the WAF receives a token request from the WWSF specifying a user identity the WAF verifies that the WWSF is authorized to access this particular user's IMS account. Only if this check is successful will the WAF return the authorization token. The verification itself can be done by consulting a subscriber database (e.g. the HSS or a custom one) and verifying that the WWSF is among the list of user authorized WWSFs. If the WAF also authenticates the user as part of the token issuance process (using e.g. the OAuth 2.0 authorization code flow), another option is that the user is asked to authorize the WWSF in a separate authorization step after the user authentication. This latter approach is commonly used by OAuth protected web services.

The details of the countermeasures are selected as follows:

If WAF is deployed inside of the operator, then we can use the countermeasure control by WAF maintained by operator, if WAF belongs to the 3rd party, then we can use the countermeasure control by S-CSCF and HSS.

Figure 6.1.2.2-2 shows an example registration flow illustrating the case when the control is enforced by S-CSCF and HSS. The new parameters are shown in figure.

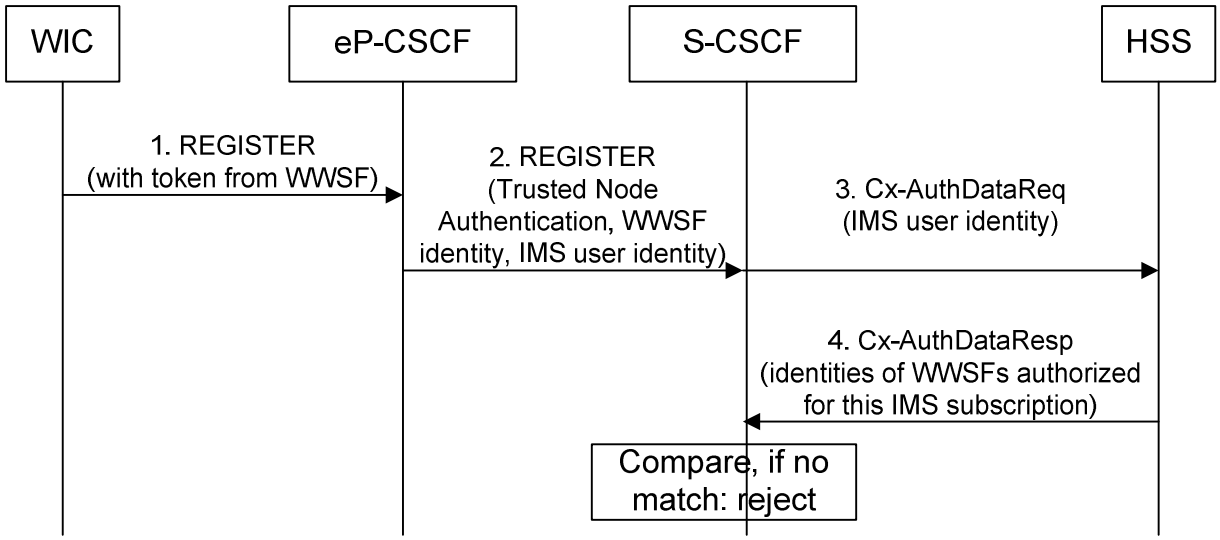

Figure 6.1.2.2-2: Example registration flow satisfying REQ 2.1

Example countermeasures to satisfy REQ 2.2 from clause 5 are:

- Control by eP-CSCFs: When a third party authentication service is under suspicion of a security breach an eP-CSCF can block all registration attempts involving assertions from that third party authentication service. All eP-CSCFs that can receive assertions from the third party authentication service under suspicion would have to be provided with the information, which third party authentication service to block. If authorization token are verified by public key signatures, this can for example be done by revoking the third party certificate and using a mechanism such as OCSP or CRLs.

- The evaluation of the countermeasure control by eP-CSCFs:

  1) Each eP-CSCF shall maintain an up-to-date list of authorized third party authentication services. If authorization tokens are verified using public key signatures, the eP-CSCF can determine if a third party authentication service has been blocked by checking the revocation status of its certificate, using e.g. CRLs or OCSP. Similar revocation mechanism for symmetric keys could be used if the authorization token is verified using MACs instead of public key signatures.
  If key revocation is not a suitable for some reason, another option is to store the identities of the third party authentication services in a white or black list. However, this option involves additional complexity and administration.

- Control by S-CSCF and HSS: The eP-CSCF has to explicitly send the identity of the third party authentication service to the S-CSCF with the REGISTER message. (The mechanism from the countermeasures to satisfy REQ1 could be re-used.) Then the S-CSCF can block all registration attempts involving assertions from that third party authentication service. All involved S CSCFs would have to be provided with the information, which third party authentication service to block, either by OAM or from the HSS.

- The evaluation of the countermeasure control by S-CSCF and HSS:

  1) The HSS is the natural repository for subscription-related information, there is no need to set an extra database to process.

  2) The HSS has stored all users' subscription information so that it has no need to update its database dynamically.

  3) The register message from the WIC to the eP-CSCF contains a new parameter so-called the WWSF identity, which means that the SIP signalling needs to change.

  4) The diameter interface between the S-CSCF and HSS needs to be extended as well.

- Control by WAF maintained by operator: This countermeasure assumes that the WWSF resides in the third party domain while the WAF resides in the operator domain. If a WWSF is suspected of a security breach the WAF

will block all token requests originating from that WWSF. In this way attacks are prevented at the earliest possible stage, even before the SIP registration procedure has started.

The details of the countermeasures are selected as follows:

If WAF is deployed inside of the operator, then we can use the countermeasure control by WAF maintained by operator, if WAF belongs to the 3rd party, then we can use the countermeasure control by S-CSCF and HSS.

6.1.2.3 Example of web authentication using IMS AKA credentials

This is an example of web authentication for scenario 2. The solution relies on IMS AKA credentials thanks to GBA mechanism as defined in normative Annex D of 3GPP TS 33.222 [11].

In this solution, it is assumed that:

- The UE re-uses IMS AKA credentials;
- The WebRTC IMS Client implements GBA features as defined in normative Annex D of 3GPP TS 33.222 [11];
- The WWSF is a NAF that implements the associated GBA features described in normative Annex D of 3GPP TS 33.222 [11].

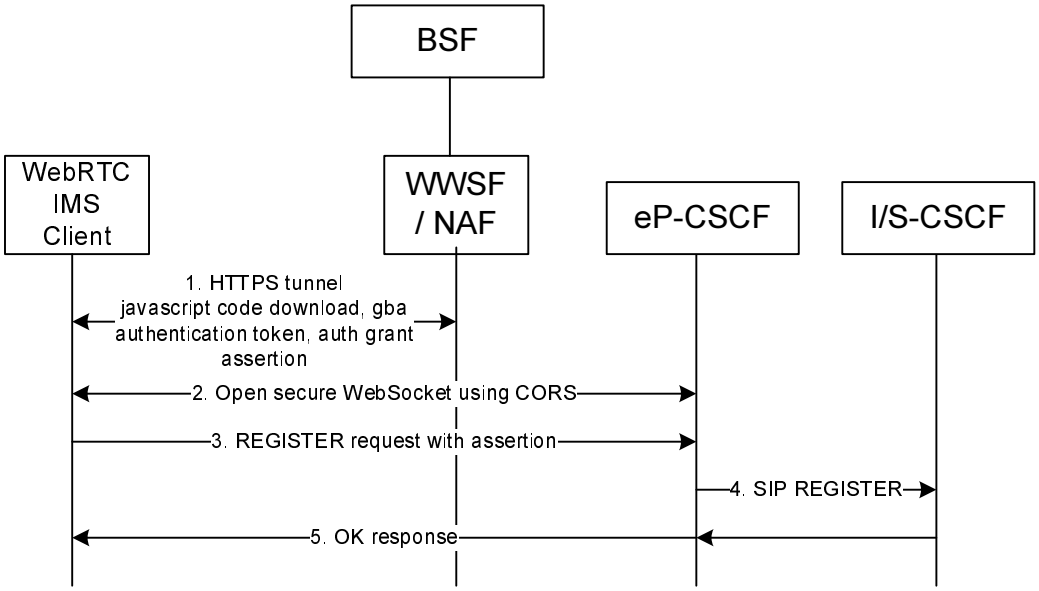

Figure 6.1.2.3-1: WebRTC IMS Client authentication relying on IMS AKA credentials

1) From within a WebRTC-enabled browser, the user accesses a URI to the WWSF to initiate an HTTPS connection to the WWSF. The TLS connection provides one-way authentication of the server based on the server certificate. The browser downloads and initializes the WIC from the WWSF.
The WWSF sends gba-related javascript code (gba.js) and authenticates the WIC by means of authentication token Ks_js_NAF, as described in Annex A of this document. After successful GBA-based authentication relying on IMS AKA credentials, the WWSF determines the IMPI and IMPU assigned to the user (the IMPI and IMPU are received from the BSF), issues a security token for the user (e.g. where the security token is a JSON Web Token) and returns the IMS identities as claims within the security token to the WIC.

2) The WIC opens a WSS connection to the eP-CSCF using CORS procedures to ensure that the WIC originated from a WWSF authorized to access this eP-CSCF.

3) The WIC sends a REGISTER request to the eP-CSCF via the WSS connection. The request includes the user identity extracted from the claims in the security token, as well as the security token received from the WWSF as an attachment to the request.

4) The eP-CSCF validates the contents of the security token and confirms that the IMS identities being registered are authorized by the security token. The eP-CSCF then forwards the authorized REGISTER request to IMS to initiate authentication-less IMS registration using TNA procedures, with an indication that the authentication has already been carried out.

5) IMS returns an OK response to the WIC to confirm the successful IMS registration.

The Web identity used to perform the web authentication is the B-TID value defined by GBA procedure. The B-TID was specified to bind the subscriber identity to the IMS AKA credentials of the IMS subscription. Consequently, the Web identity relying on B-TID is trustable. Moreover, there is a secure linkage linking between Web identity (B-TID)

and the IMPI/IMPU of IMS subscription since those identities are associated to the same IMS subscription and under the control of the operator via the HSS and the BSF.

GBA-based web authentication relies on existing normative mechanism, applies to common IMS and provides linkage between Web identity and IMPI/IMPU of IMS subscription.

6.1.2.4 Use of direct authentication between WIC and eP-CSCF

In this solution, user uses the web credentials to register with IMS directly, which WIC executes the SIP Digest mechanism to perform the authentication procedures. It is assumed the mapping of web identity and IMPU is stored in HSS, Web identity is acted as the IMPI function used for authenticating the subscription, and as usual, the IMPU is used to identify the user in the session.

Editor's note: It is FFS whether this solution is in the scope of scenario 2 described in Clause U.2.1.3 of the TS 23.228.

As shown in Figure 6.1.2.4-1, the signalling flows can be demonstrated as follows:

1) REGISTER request (WebRTC IMS Client to eP-CSCF)

The WebRTC IMS Client establishes a secure WebSocket connection with the eP-CSCF and sends a REGISTER request. The REIGISTER request message includes the user's Web identity and the WWSF identity.

NOTE: WWSF identity is used to distinguish the Web service provider, and serves for the proxy to find the AVs. If the Web identity is in the format of bob@socialnet.com which has already clearly shown the specific web service provider, then the WWSF identity may not be needed.

2) eP-CSCF forwards the REGISTER Request

The eP-CSCF and the I-CSCF respectively forwards the REIGSTER Request towards the S-CSCF as described in Annex N of TS 3.203 [5]. The I-CSCF queries the HSS to find the address of the S-CSCF. At this stage, the HSS shall check the user's Web identity maps with a specific IMPU (e.g. via database of the mapping relationship between user's Web identity and IMS identity). The S-CSCF checks whether the REGISTER Request is related to SIP Digest in terms of the rules in Annex P of TS 33.203 [5]. If it is, the procedures apply below.

Editor's Note: Routing of incoming and outgoing calls is FFS. One of the possible methods is that the Web identity as the temporary IMPU or WIC/eP-CSCF generating a special temporary IMPU sent to the IMS networks for initial registration, but this temporary IMPU is barred for the session, and the IMPU linking with the Web identity is implicitly registered. In this way the IMPU can be used for session.

Editor's Note: Whether inter domain calls (I.e. between two IMS operators) are possible when web identities are used for addressing is FFS.

3) Authentication Vector Request

After receiving the REGISTER request, the S-CSCF shall send the Authentication Vector Request (AV Req) towards HSS. The AV Req message includes the user's Web identity and the WWSF identity.

4) Authentication Vector Response

Upon receiving the request from the S-CSCF, the HSS forwards the request towards the proxy. The proxy sends the AV consists of Web identity, qop value, the algorithm, realm, and H(A1) based on the user's Web identity and the WWSF identity to the HSS, H(A1) consists of the Web identity, realm, and password. And the HSS passes the AV to the S-CSCF.

NOTE: The proxy is the intermediate entity between the IMS and the authentication service and it's a component of the HSS. The proxy serves for sending the AV which has been synchronized for each Web user with the authentication service offline in advance. How does the synchronization process is left out of scope.

5) 401 Auth_Challenge (S-CSCF to eP-CSCF)

The S-CSCF sends a SIP 401 Auth_Challenge to the eP-CSCF, including the Web identity, realm, qop, algorithm, and new generated random number nonce. Meanwhile, the S-CSCF stores H(A1).

6) 401 Auth_Challenge (eP-CSCF to WebRTC IMS Client)

The eP-CSCF forwards the 401 Auth_Challenge to the WebRTC IMS Client.

7) REGISTER Digest-Response (WebRTC IMS Client to eP-CSCF)

Upon the receipt of the challenge, the WIC generates a cnonce, and uses the cnonce as well as parameters in the receive challenge to compute an authentication response refer to RFC 2617 [21].

8) eP-CSCF forwards the Digest-Response

The eP CSCF forwards the authentication response to the I CSCF, which queries the HSS to find the address of the S CSCF. And then, the I CSCF forwards the authentication response to the S CSCF. The S-CSCF calculates the expected response using the previously stored H(A1) and stored nonce together with other parameters (e.g. cnonce, nonce-count, qop, as specified in RFC 2617 [21]) and uses this to check against the response sent by the WIC.

9) 200 (OK) response (S-CSCF to eP-CSCF)

The S-CSCF sends a 200 (OK) response to the eP-CSCF (via I-CSCF) indicating that Registration was successful.

10) 200 (OK) response (eP-CSCF to WebRTC IMS Client)

The eP-CSCF forwards the 200 (OK) response to the WebRTC IMS Client indicating that Registration was successful.

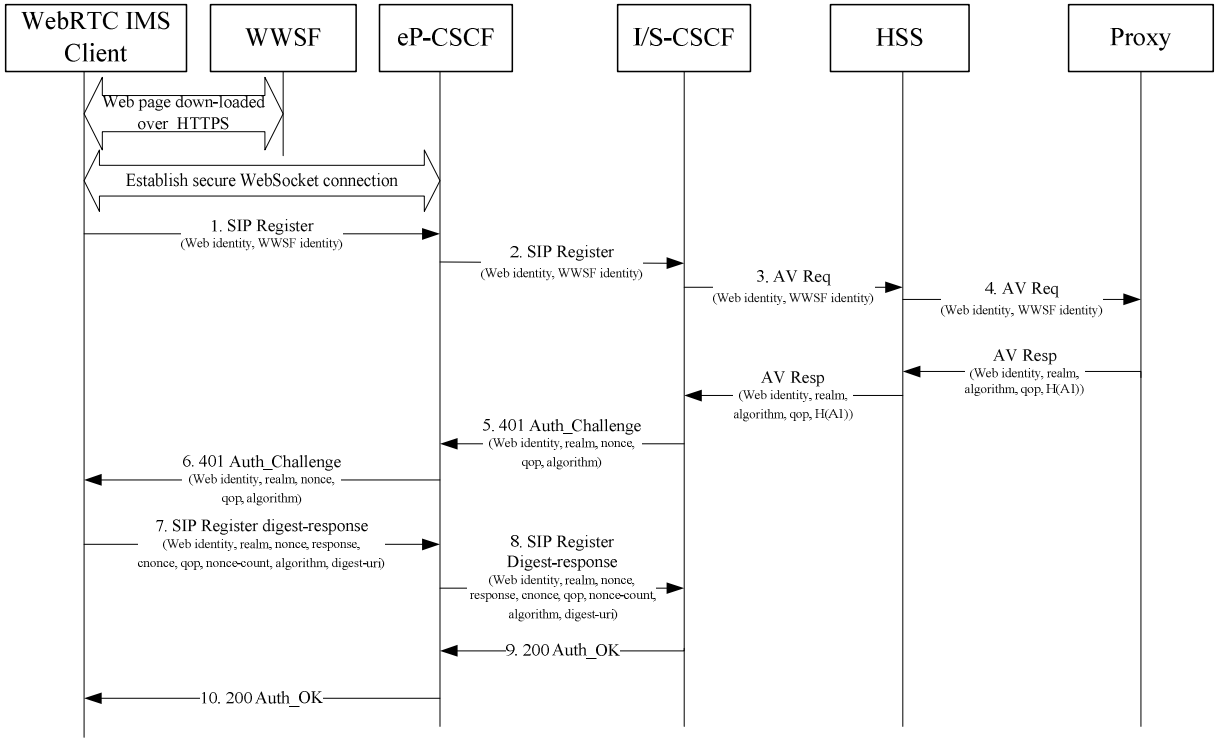

Figure 6.1.2.4-1: WIC direct authentication using web credentials

6.1.2.5 Trusted Node Authentication using OAuth 2.0 Implicit Grant

In the OAuth 2.0 framework**,** Implicit grant based authorization is used when the client application is not capable of keeping the client"s own credentials secret. The application runs in a less-trusted environment which makes it vulnerable for attacks that could leak the access token. In scenarios like this, the client application just maintains its

Client ID and doesn"t need to know or store its Client secret (a.k.a password). JavaScript based application running in the context of browser is an example of an application that can use Implicit grant to authenticate the resource owner.

In 3GPP IMS WebRTC architecture, WIC is a JavaScript based application running in the browser. It is therefore possible to use Implicit grant based authorization to authenticate and authorize WIC to use IMS services on behalf of the IMS subscriber.

The procedure in this clause lists all the steps till the point TNA as specified for IMS in Annex U of TS 33.203 [5] is applied by the trusted eP-CSCF. The signalling between eP-CSCF and the rest of the IMS core is unchanged from the signalling flow in Annex X of TS 33.203.

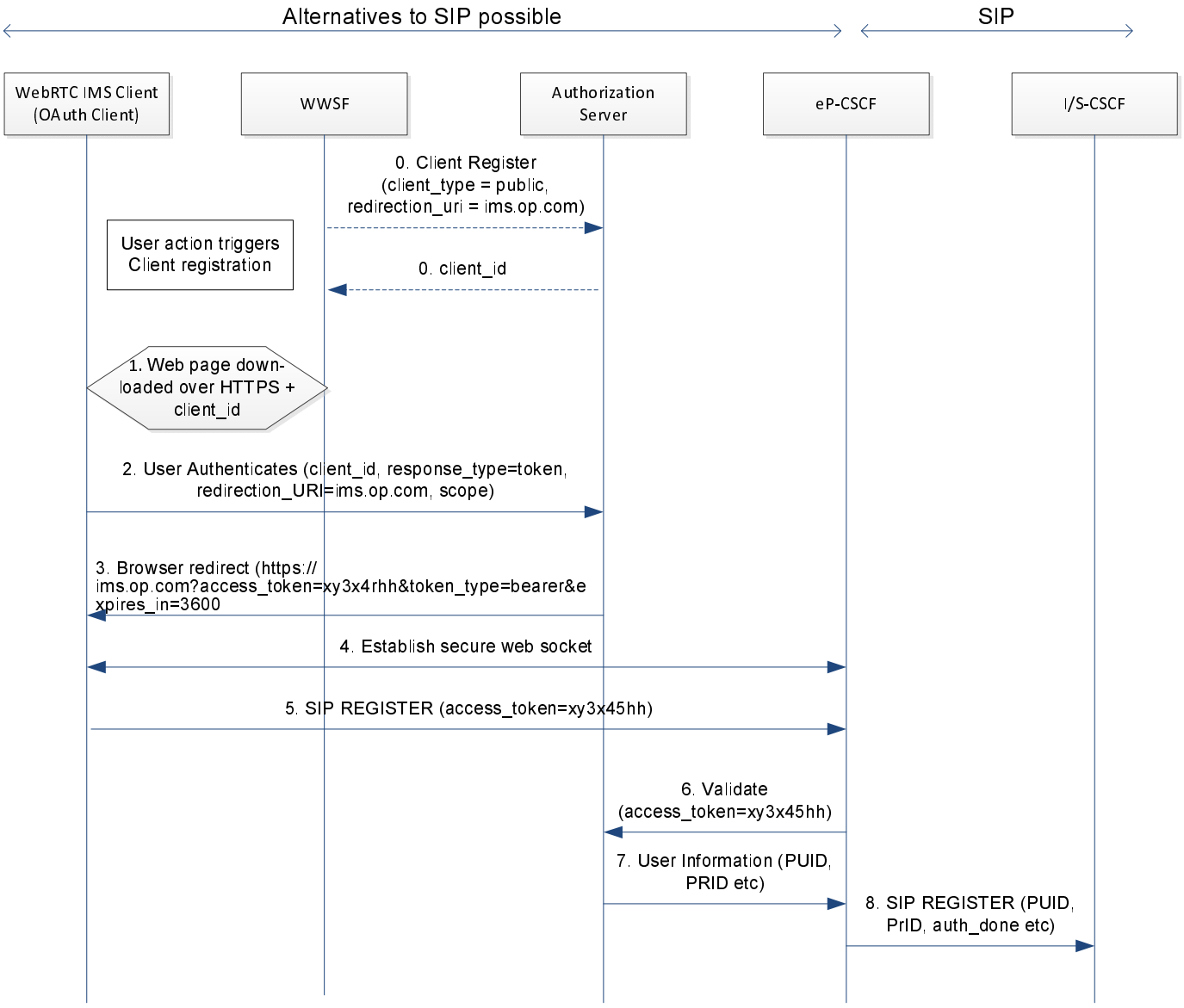

Figure 6.1.2.5-1: Trusted Node Authentication using OAuth 2.0 Implicit Grant

The details of the signalling flows are as follows :

0. Client is registered with the Authorization server

Before a client application can request access to the protected resources of IMS subscriber, the client application must first register with the Authorization server associated with the resource server (IMS operator). The field

client_type is set as public. A redirect URI is also registered with the server. This is the client"s redirection point used by the Authorization server to redirect the browser (user-agent) once the IMS subscriber is authenticated successfully.

In response, the Authorization server will assign a unique client ID to the registered client.

Step 0 is completed independently in advance of the following steps.

Note: It is assumed that the user has a web-based account with the Authorization Server.

1. Web page downloaded from WWSF

The user accesses a URI to the WWSF to initiate an HTTPS connection to the WWSF. The browser downloads and initializes WIC from the WWSF. WIC learns its client ID at this point.

Note: There may be a login step required in this step. This authenticates the resource owner with the WWSF.

2. User selects an IMS action from the web page

This redirects the IMS subscriber to the Authorization server. The client includes its previously registered client ID and the redirect_uri in this request. It also includes response_type along with the scope (optional) and local state (optional) in this request. The fields response_type is set to "token" as an indicator to the Authorization server that it should respond directly with the access token.

The user authenticates with the Authorization server and authorizes the WIC to access IMS communication services.

3. WIC gets the access token

The Authorization server now redirects the user-agent back to the client web application using the redirection URI provided earlier. The redirection URI includes the access token in the URI fragment. The user-agent (browser) will follow the redirect. The resulting HTML page will have the JavaScript code to parse the access token from the redirect URI. The user-agent passes the access token to the client.

The access token is associated with the authenticated user and client, and has a certain lifetime and scope. The associated authorization information such as the user IMPU/IMPI, WAF Id etc. can either be encoded into the token itself and verified through a signature or MAC (so called self-contained token), or retrieved as part of the validation response if the validation is performed against the WAF (Step 7 below).

4. Establishment of secure connection between WIC and eP-CSCF

The WIC opens a wss (secure web socket) connection to the eP-CSCF.

5. REGISTER request

The WIC sends a REGISTER request. This includes the access token which the WIC has previously obtained in step 3. The access token is included in the Authorization header in the REGISTER request.

The access token is of the bearer token type [RFC 6750].

6. Validation of access token at eP-CSCF

The eP-CSCF extracts the access token from the incoming W2 message and validates it over the W5 reference point with the Authorization server.

7. IMS subscriber information retrieved from the Authorization server

If the access token is valid, the eP-CSCF obtains the associated IMS subscriber information including the IMPU and IMPI, the token expiry time token scope etc.

8. REGISTER request (eP-CSCF to IMS core)

Step 5 and beyond of the existing solutions for IMS WebRTC Scenarios 2 and 3 as specified in Annex X of TS 33.203 apply from this point onwards.

6.1.3 Assignment of IMS identities to WebRTC IMS Client from pool of IMS subscriptions held by WWSF

6.1.3.1 General

In this scenario it is assumed that the "WWSF is provided with a pool of subscriptions with IMS and can assign individual Public User Identities within this pool" (quoted from TS 23.228 [3]). This assignment is temporary and the same IMPU (and IMPI) may be re-assigned to a different user after some period.

The user's web identity may be authenticated by a third party authentication service (WWSF or authorization server), but "the WWSF may decide not to authenticate the user. Unauthenticated users are anonymous to the third party but may still be authorized for IMS service" (quoted from TS 23.228 [3]).

NOTE 1: The difference to the scenario addressed in clause 6.1.2 is that, in the present scenario, the IMS subscriber is the WWSF, not the user. There is no linkage between the user's web identity that may be authenticated by a third party authentication service and the assigned IMS identities.

NOTE 2: Considerations on Lawful Interception, e.g. when the user is anonymous to the third party, are outside the scope of the present document.

The authentication service issues authorization information to the WebRTC IMS Client (WIC) that the WIC presents to the eP-CSCF. The eP-CSCF validates the authorization information. Provided the validation of the authorization information is successful, the eP-CSCF performs the IMS registration on behalf of the user.

6.1.3.2 Use of Trusted Node Authentication (TNA)

The scenario applies Trusted Node Authentication (TNA) specified for IMS in Annex U of TS 33.203 [5].
While TNA was specified mainly for interworking with the CS access domain, the technology is access and protocol independent. In TNA, the trusted node (I.e. eP-CSCF) can authorize the user for IMS access by means of authorization information received from the third party authentication services, that the trusted node can provide interworking between the IMS domain and the other domain, in which the WWSF resides, if necessary, and that the operator trusts the authorization information provided by the third party authentication service. It is clear that the operator trusts the eP-CSCF, performing the role of trusted node in TNA, as the eP-CSCF resides in the operator network, according to TS 23.228 [3].

The authorization information is sent to the WebRTC IMS Client which includes it in the initial registration request to the eP-CSCF. Provided the validation of the authorization information is successful, the e-PCSCF will proceed with the IMS registration of the user using TNA.

An example of a signalling flow for when the Trusted Node performs registration on behalf of the WebRTC IMS Client is shown in Figure 6.1.3.2-1. In this figure SIP over secure WebSocket is used between the WebRTC IMS Client and the eP-CSCF. Other protocols (e.g. HTTP RESTful or JSON over WebSocket) can also be used.
The signalling between the Trusted Node and the rest of the IMS core is unchanged from the signalling flow in Annex U of TS 33.203 [5] in Figure 6.1.2.2-1.

In the example authentication protocol between the WebRTC IMS Client and the eP-CSCF the WWSF first obtains an access token from an Authorization server which contains information authorizing access to one of the WWSF's IMS subscriptions from the pool. The token is then sent to the WebRTC IMS Client which includes it in the initial registration request to the eP-CSCF. Provided the token validation is successful, the e-PCSCF proceeds with the IMS registration of the user using TNA.

In the example, the access token is associated with one of the WWSF's IMS subscriptions from the pool and has a certain lifetime and scope. This authorization information can either be encoded into the token itself and validated through a signature or MAC (so called self-contained token) or retrieved as part of the response if the validation is performed against the WAF.

NOTE 1: In this release it is only the W2 interface that is partially described by way of example (but not fully specified); how the WWSF obtains the token and how it is made available to the WebRTC IMS Client is left out of scope.

NOTE 2: In the present 3GPP release the format of the authorization information and the related validation procedure are left out of scope. It is assumed that the eP-CSCF can check the validity of the token and obtain the IMPU, IMPI, WWSF identity, lifetime, and scope parameters.

NOTE 3: W1 and W2 interfaces are out of scope of the present 3GPP release. Nevertheless, consideration needs to be given to the risk of token disclosure. Integrity and confidentiality protected using TLS is a mandatory requirement in the OAuth bearer token specification RFC 6750 [14].

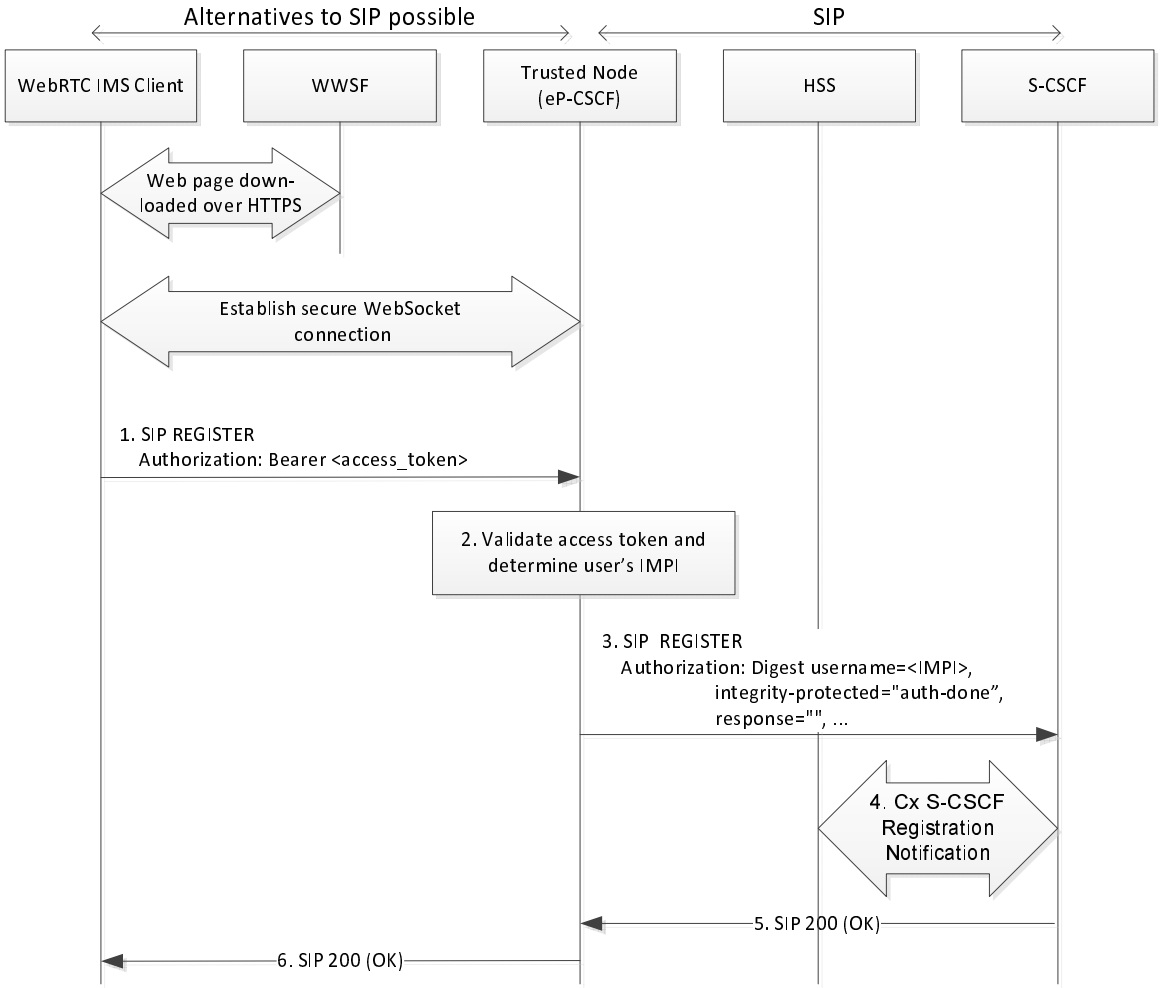

Figure 6.1.3.2-1: Trusted Node performs registration on behalf of the WebRTC client (example flow)

The details of the example signalling flows are as follows:

NOTE 4: If the WAF resides in the operator domain, we can use the WAF control to authorize the WWSF. If WAF is not in the operator domain, then we can use the authorization of the WWSF by S-CSCF and HSS.

0. Token request (WWSF to WAF)

If the WAF resides in the operator domain, when the WAF receives a token request, it can either authenticate the user itself as part of the token issuance process, or it trusts the user identity supplied by the WWSF which is assumed to have authenticated the user prior to sending the token request. Only if this check is successful will the WAF return the authorization token; otherwise, the WAF rejects the request of the WWSF. And if a WWSF is suspected of a security breach the WAF will block all token requests originating from that WWSF.

1) REGISTER request (WebRTC IMS Client to Trusted Node)

The WebRTC IMS Client establishes a secure WebSocket connection with the eP-CSCF and sends a REGISTER request. The Authorization header includes authorization information, e.g. an OAuth 2.0 access token, which the WebRTC IMS Client has previously obtained. A so called "bearer" token type may be used; see RFC 6750 [14].

NOTE 5: OAuth bearer tokens can be used with signalling protocols that supports the Authorization header defined in RFC 2617, for example SIP and HTTP.

2) Validation of security token at eP-CSCF

The eP-CSCF extracts the authorization information, e.g. the access token, and validates it in some unspecified manner ensuring that only an authorized source can have generated the authorization information.
If the authorization information is valid the eP-CSCF obtains the associated authorization information, including the IMPI and IMPU assigned to the user, the WWSF identity, and the authorization information scope. The eP-CSCF verifies that the scope includes the value "webrtc-ims-client-access-to-ims".

Under certain assumptions, the eP-CSCF can also verify that the IMPI, if it exists at all in the IMS, belongs to an IMS subscription in the pool of IMS subscriptions assigned to the WWSF.

NOTE 6: Such an assumption would be e.g. that the IMPIs from the pool of IMS subscriptions assigned to the WWSF have a special form, and the IMS provider does not assign IMPIs of this form to any other WWSF. E.g., for the WWSF "socialnet.com', the IMPIs could all be of the form "xyz@socialnet.com'. (This would not imply the converse that all IMPIs of this form actually do belong to an IMS subscription in the pool of IMS subscriptions assigned to "socialnet.com' as the present scenario is not using wildcard IMPIs.) Then, if an IMPI of the form "xyz@socialnet.com' is presented to the eP-CSCF for registration of a WIC, the eP-CSCF knows that the authorization information for using this IMPI needs to be provided by socialnet.com. So, the eP-CSCF retrieves the cryptographic verification key of socialnet.com to verify the received token, or the eP-CSCF contacts socialnet.com to obtain confirmation. No other WWSF, even if it was compromised, could issue false authorization information about this IMPI. However, the IMPU would not have to follow the same special format as the IMPI.

The eP-CSCF further verifies other verifiable information, such as a time stamp and a validity period.

If the validation fails in some respect, the eP-CSCF declines the register request, closes the web socket and aborts the procedure.

3) REGISTER request (Trusted Node to S-CSCF)

The eP-CSCF proceeds if the previous step has provided it with IMPI, IMPU(s) of the user requesting registration, an assurance that the user is authorised to use this IMPI and IMPU, and an identity of the entity that provided this assurance (authorization entity). Then, the eP-CSCF generates a TNA Authorization header and forwards the request to the S-CSCF (via the I-CSCF). The format of the TNA Authorization header is specified in TS 24.292 [15], clause 6.2, and contains, among others, the IMPI assigned to the user, an integrity-protected directive set to auth-done, and an empty response directive. Furthermore, the eP-CSCF includes the identity of the WWSF if the WAF belongs to the third party network.

4) Cx: S-CSCF Registration Notification

Based on the presence of the "integrity-protected" directive set to indicate that authentication has already been performed, the S-CSCF knows that the user's authorization has already been validated by the Trusted Node. The S-CSCF informs the HSS that the user has been registered. Upon being requested by the S-CSCF, the HSS will also include the user profile in the response sent to the S-CSCF. For detailed message flows see TS 29.228 [16].

If the WAF is not inside of the operator domain, the HSS further includes a list of identities of WWSFs outside the IMS provider"s domain allowed for this IMS subscription. When the S-CSCF received an identity of the WWSF from the eP-CSCF, it checks whether the WWSF identity received from the eP-CSCF and the HSS respectively match. If it is, the S-CSCF proceeds with the next step; otherwise, it rejects the registration. And if a WWSF is suspected of a security breach the S-CSCF will block all registration attempts involving assertions from that WWSF.

5) 200 (OK) response (S-CSCF to eP-CSCF)

The S-CSCF sends a 200 (OK) response to the eP-CSCF (via I-CSCF) indicating that registration was successful.

When TLS is used between WIC and eP-CSCF, then, similar to the registration procedure for SIP Digest with TLS, the eP-CSCF associates the IMPI and all successfully registered IMPUs with the TLS Session ID when the 200 (OK) is received.

6) 200 (OK) response (eP-CSCF to WebRTC IMS Client)

The eP-CSCF forwards the 200 (OK) response to the WebRTC IMS Client indicating that Registration was successful.

NOTE 7: The eP-CSCF can verify that the web-page establishing the signalling connection comes from a trusted domain by inspecting the value of Origin header. This header is inserted by the browser in the WebSocket handshake and in every HTTP request (requires the use of CORS, http://www.w3.org/TR/cors/). The protection mechanism works under the assumption that the browser is not under the attacker's control, which means that the contents of the Origin header can be trusted.

Editor's Note: It is desirable for 3GPP to provide a security mechanism for the interface between WIC and eP CSCF in Rel-12, but it is FFS whether this goal can be achieved in Rel-12. Furthermore, it is FFS, which authentication mechanism to specify. It is also FFS whether this security mechanism should be mandatory to implement, but not mandatory to use, or whether it should just be an example security mechanism. It is agreed that, if SA2 does not provide a full specification of the signalling interface as mandatory to implement, then it only makes sense to have an example security mechanism in SA3. It is not intended to make it mandatory to use. The advantages of such a 3GPP-defined security mechanism for the interface between WIC and eP-CSCF would include ensuring interoperability between WICs and eP CSCFs from a security point of view and ensuring a minimum level of security.

Example countermeasures to satisfy REQ 3.1 from clause 5 are:

Step 2 states that "Under certain assumptions, the eP-CSCF can also verify that the IMPI, if it exists at all in the IMS, belongs to an IMS subscription in the pool of IMS subscriptions assigned to the WWSF." If this assumption cannot be made then similar countermeasures to the ones provided for the second registration scenario would be required.

The countermeasures differ in the enforcement points:

- Control by eP-CSCFs: TR 23.701 [4], Annex A.1.3.3, states: "The eP-CSCF verifies that the WWSF is authorized to allocate IMS identities that it assigns to a WIC." This text suggests control by eP-CSCFs. In order to enable this verification all eP-CSCFs that may receive assertions (in the form of authorization tokens) issued by a certain third party authentication service have to be provided with the list of the IMS identities belonging to the pool of IMS subscriptions of WWSFs. But, considering that several eP-CSCFs can receive assertions issued by one third party authentication service, one eP-CSCF can receive assertions issued by several third party authentication services operated by different third parties, and that these lists would have to be updated dynamically, this solution may be difficult to manage and not scale well (unless the above assumption, e.g. about the form of IMS identities can be made, in which case the countermeasure would be easy to implement in the eP-CSCF). In view of these disadvantages one may want to look at using a different enforcement point, cf. next paragraph.

- Control by S-CSCF and HSS: For each IMS subscription, an HSS entry indicates, which third party authentication service or WWSF owns the subscription relating to a given IMS identity. The HSS is the natural repository for subscription-related information. This information is sent to the S-CSCF over Cx during registration. The eP-CSCF sends the identity of the third party authentication service or WWSF (whichever acted as the authorization entity) to the S-CSCF with the REGISTER message. The S-CSCF can then check whether the third party authentication service identities or WWSF received from the eP-CSCF and the HSS respectively match.

The details of the countermeasures are selected as follows:

If WAF is deployed inside of the operator domain, then we can use the countermeasure control by WAF maintained by operator. If WAF belongs to the 3rd party domain, then we can use the countermeasure control by S-CSCF and HSS.

Figure 6.1.3.2-2 shows an example registration flow illustrating the case when the control is enforced by S-CSCF and HSS. The new parameters are shown in figure.

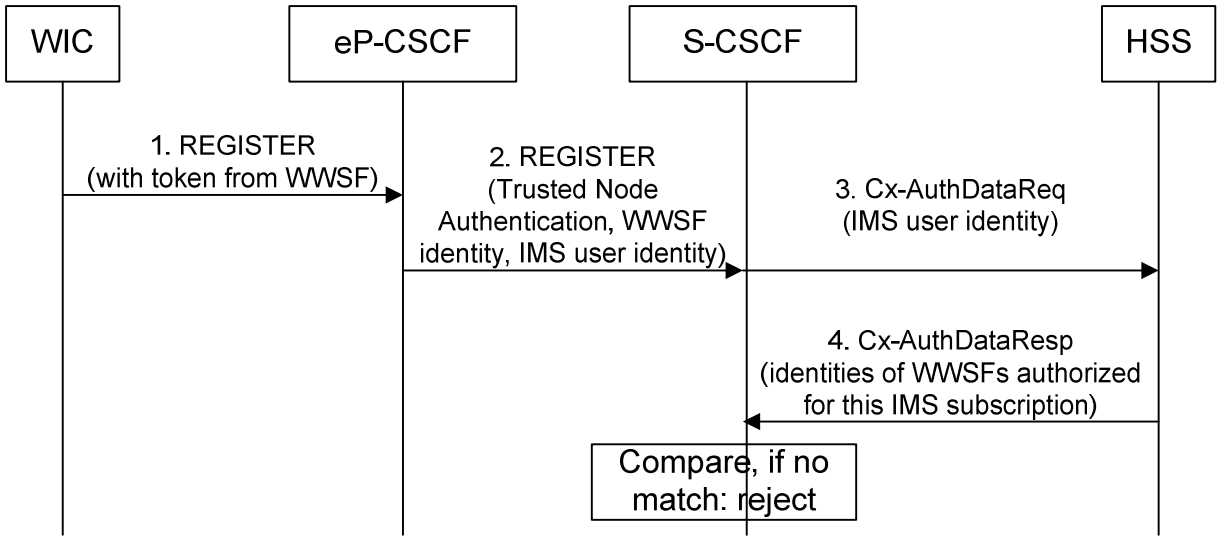

Figure 6.1.3.2-2: Example registration flow satisfying REQ 2.1

Example countermeasures to satisfy REQ 3.2 from clause 5 are:

- Control by eP-CSCFs: When a third party authentication service or WWSF is under suspicion of a security breach an eP-CSCF can block all registration attempts involving assertions from that third party authentication service. All eP-CSCFs that can receive assertions from the third party authentication service under suspicion would have to be provided with the information, which third party authentication service to block.

- Control by S-CSCF and HSS: The eP-CSCF has to explicitly send the identity of the third party authentication or WWSF service to the S-CSCF with the REGISTER message. (The mechanism from the countermeasures to satisfy REQ1 could be re-used.) Then the S-CSCF can block all registration attempts involving assertions from that third party authentication service. All involved S CSCFs would have to be provided with the information, which third party authentication service to block, either by OAM or from the HSS.

The details of the countermeasures are selected as follows:
If WAF is deployed inside of the operator domain, then we can use the countermeasure control by WAF maintained by operator. If WAF belongs to the 3rd party domain, then we can use the countermeasure control by S-CSCF and HSS.

6.2 Enhancements to IMS media plane security

Editor's Note: This clause contains the needed Enhancements to IMS media plane security to support WebRTC IMS Clients, I.e. support of DTLS-SRTP.

6.2.1 Media security for RTP

6.2.1.1 General

According to [10], all RTP traffic generated or received by a WebRTC client shall be protected with SRTP, using DTLS-SRTP as the key management protocol. This means that if a WebRTC IMS Client is supposed to be able to communicate with existing IMS endpoints (e.g. IMS UE or PSTN GW), DTLS-SRTP and SRTP shall be terminated at an intermediate node.

This clause describes the additional procedures and interface extensions required to support end-to-access-edge (e2ae) security for RTP using DTLS-SRTP and SRTP.

Editor's Note: The solution for e2ae security outlined in this clause only applies to network centric approach for WebRTC access to IMS. Whether SA3 should study the device centric approach as well (where transcoding and encryption/decryption is handled in the UE) depends on the outcome of the SA2 discussions.

6.2.1.2 e2ae security for RTP using DTLS-SRTP

E2ae protection of RTP using DTLS-SRTP is similar to e2ae protection of MSRP using TLS and the session establishment procedures are therefore largely the same. In both cases certificate fingerprints need to be exchanged over SDP and the media has to be anchored in IMS by inserting a gateway on the media path. Similarly as for e2ae protection using SDES and TLS, the signalling path between the WebRTC IMS Client and the eP-CSCF needs to be secured.

Figure 6.2.1.2-1 shows the originating procedure for e2ae protection of RTP using DTLS-SRTP. The terminating procedure is similar and is not shown here.

Note that no assumption is made on the interface between the WebRTC IMS client and the eP-CSCF except that it is SDP based and integrity protected.

Since only e2ae security is supported at the moment, the WebRTC IMS Client is required to include the indication "e2ae-security requested by UE" in every offer it creates.

It is assumed that the eP-CSCF is aware of the fact the IMS UE is a WebRTC IMS Client and automatically applies e2ae security for terminating calls. Therefore, unlike the existing e2ae security for RTP and MSRP, there is no need for the IMS UE to explicitly indicate support of e2ae security during registration.

NOTE 1: Two DTLS-SRTP handshakes are needed if RTP and RTCP are sent over separate transport flows. If RTP/RTCP multiplexing is used, only a single DTLS-SRTP handshake is needed.

NOTE 2: In this release, DTLS-SRTP is only intended to be used by WebRTC IMS Clients. Use of DTLS-SRTP by other types of IMS UEs may be studied in future releases.

The DTLS-SRTP profile to use is described in Annex B of this document.

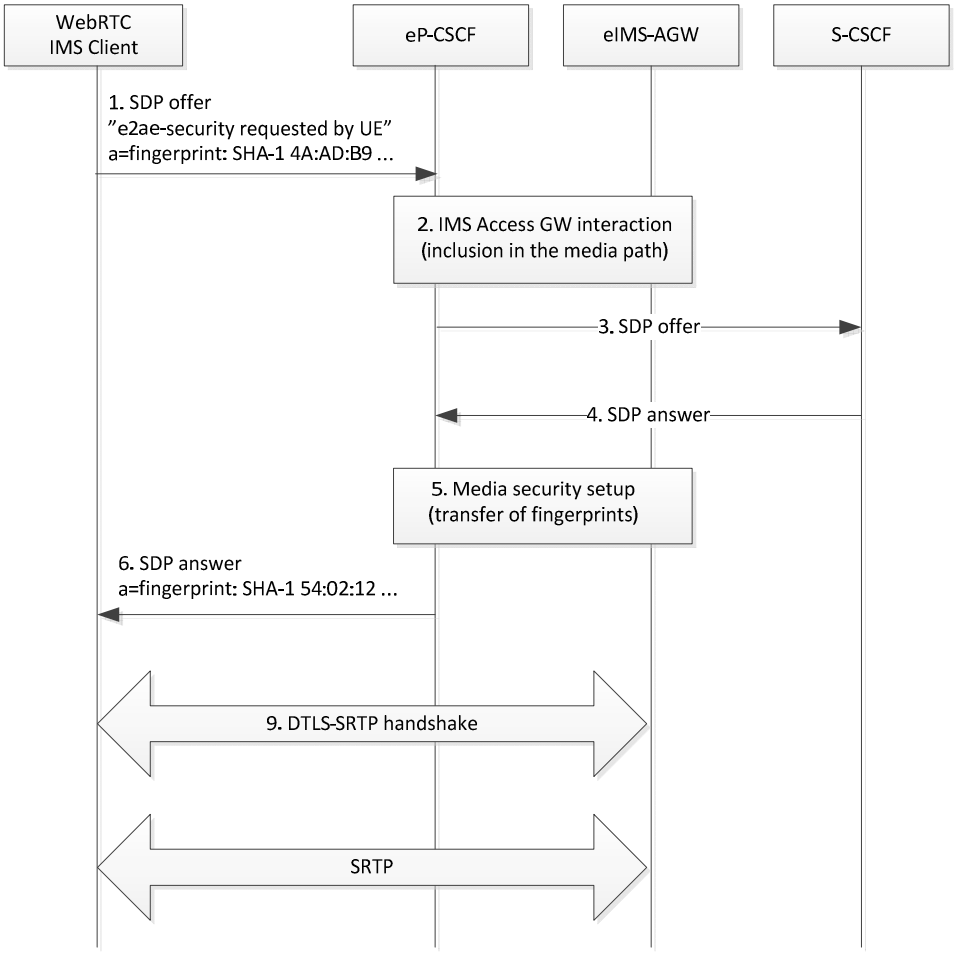

Figure 6.2.1.2-1: E2ae protection of RTP based on DTLS-SRTP

6.2.2 Media security for WebRTC Data Channels

6.2.2.1 General

This clause describes how end-to-access-edge (e2ae) security is achieved for WebRTC Data Channels.

WebRTC-compatible browsers use SCTP over DTLS as transport protocol for peer-to-peer data. A WebRTC Data Channel is defined as two unidirectional SCTP streams, one in each direction, which are managed together as a single entity (see draft-ietf-rtcweb-data-protocol [18]). The application protocol which runs on top of the WebRTC Data Channel is not specified and the JavaScript is free to implement any protocol it requires.

The application protocols that a WebRTC IMS Client may need to support are MSRP, BFCP, T.140, and T.38. Figure 6.2.2.1-1 shows the common protocol stack and the required protocol translation. The transport protocol that the IMS-AGW applies on the remote side (marked X in the figure) depends on the application protocol.
For MSRP and BFCP X=TCP, for T.140 X=RTP/UDP, and for T.38 X=UDPTL/UDP. In general the IMS-AGW forwards the application protocol messages transparently. The only exception is MSRP messages which contain IP address information and therefore needs to re-written by the IMS-AGW. This can however be avoided if both endpoint support the MSRP CEMA extension [20].

T.140 (real-time text) and T.38 (fax) are included here for sake of completeness. These are legacy protocols and are not expected to be commonly used.

Editor's Note: The final list of supported application protocols (e.g. MSRP, BFCP, T.140, and T.38) is to be decided by CT groups.

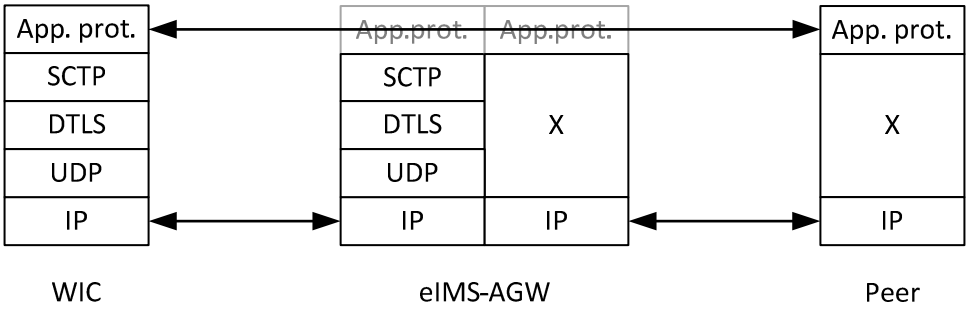

Figure 6.2.2.1-1: Protocol stack for WebRTC Data Channels

6.2.2.2 e2ae security for WebRTC Data Channels

E2ae security for WebRTC Data Channels is achieved in the same way as e2ae security for MSRP over TLS/TCP. In both cases certificate fingerprints need to be exchanged over SDP and the media has to be anchored in IMS by inserting a gateway on the media path. To ensure the integrity of the certificate fingerprint the signalling path is assumed to be protected.

Figure 6.2.2.2-1 shows the originating procedure for e2ae protection of WebRTC Data Channels. The terminating procedure is similar and is not shown here. Note that no assumptions are made on the interface between the WebRTC IMS Client and eP-CSCF except that it SDP based and integrity protected.

Since only e2ae security is supported at the moment, the WebRTC IMS Client is required to include the indication "e2ae-security requested by UE" in every offer it creates.

It is assumed that the eP-CSCF is aware of the fact the IMS UE is a WebRTC IMS Client and automatically applies e2ae security for terminating calls. Therefore, unlike the existing e2ae security for MSRP over TLS/TCP, there is no need for the IMS UE to indicate support of e2ae security during registration.

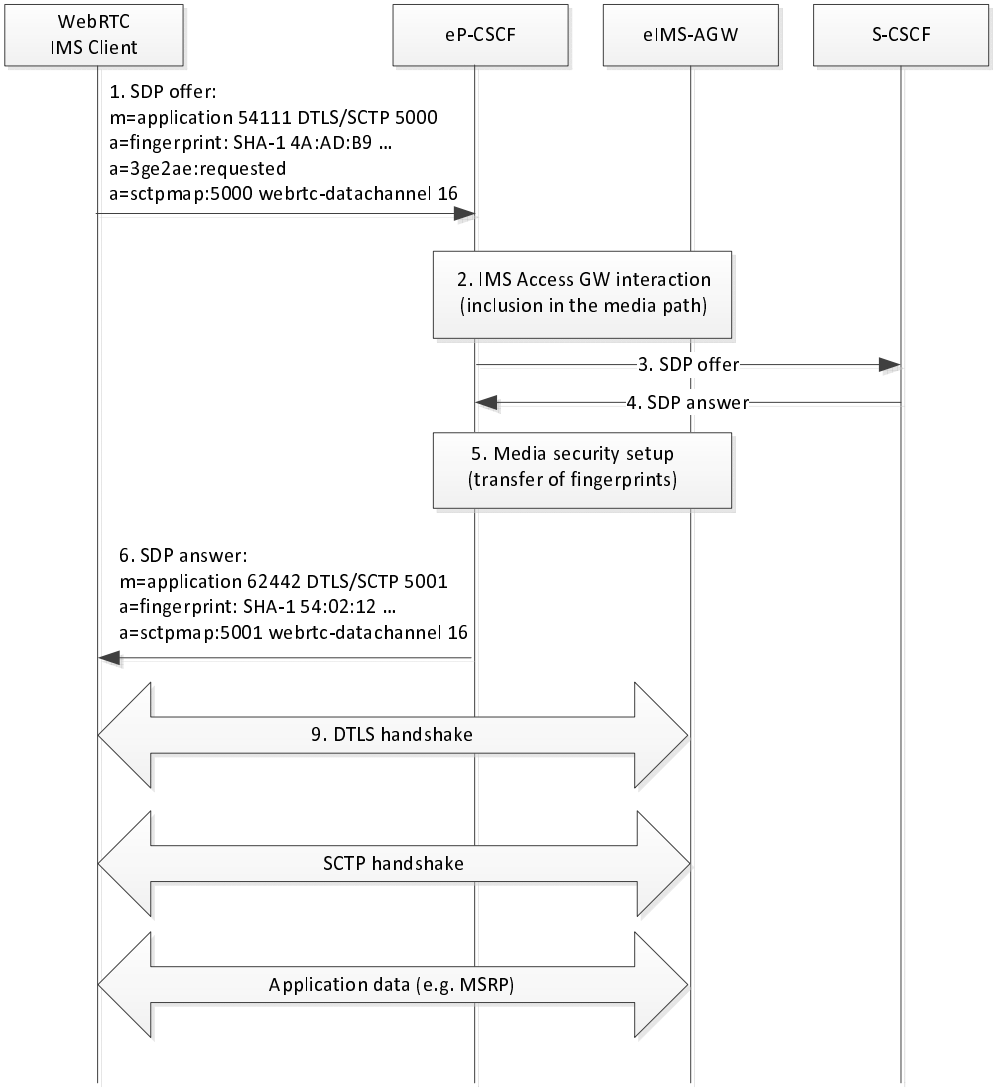

Figure 6.2.2.2-1: E2ae protection of WebRTC Data Channels

NOTE 1: How the application protocol (e.g. MSRP) and the WebRTC Data Channel configuration (e.g. stream identifiers, reliable or unreliable transmission, etc) are communicated to the remote endpoint is out-of-scope of this document and is left for the corresponding stage 3 specification. Whether this is done via SDP offer/answer as in draft-ejzak-dispatch-webrtc-data-channel-sdpneg [19], or using the in-band RTCWeb Data Channel Protocol defined in draft-ietf-rtcweb-data-protocol [18], has no relevance for security purposes.

NOTE 2: Whether multiple WebRTC Data Channels are allowed to share the same SCTP association and DTLS connection is out-of-scope of this document and is left for the corresponding stage 3 specifications. The decision to this question is not considered to have any security impact.

6.3 Other security aspects

Editor's Note: If needed, this clause contains study of other security aspects such as privacy, NAT/firewall traversal, control plane security aspects, etc.

6.3.1 Firewall traversal

A WebRTC IMS Client (WIC) may face the same firewall traversal scenario where a restrictive firewall blocks UDP and only allow TLS/443 (HTTPS) and/or TCP/80 (HTTP) to pass, as described in TR 33.830 [22].
For signalling, because WIC always sends signalling over secure WebSocket or HTTPS, a restrictive firewall does not block signalling messages and there is no need for a firewall traversal solution. However, a restrictive firewall may block WebRTC media which is sent over UDP. Therefore a firewall traversal solution is needed for WebRTC media.

To be able to traverse restrictive firewalls a WIC may support the ICE/STUN/TURN based method described in Annex W.3 of TS 33.203 [5]. In order to apply this method in the WebRTC access to IMS scenario some minor modifications will be required:

- Instead of using SIP over TLS for signalling, the WIC and eP-CSCF uses the proprietary W2 interface which runs on top of secure WebSocket or HTTPS.

- Support of ICE TCP is not required as TCP based media transport is not used by WICs

NOTE 1: Additional changes than the ones listed above may be identified in the future release.

Similar as described in Annex W.3 of TS 33.203 [5], if the browser executing the WIC is configured with an HTTP proxy, the HTTP CONNECT method is used to establish the TLS tunnels to the eP-CSCF and the TURN server.

NOTE 2: In order for the firewall traversal solution in Annex W.3 of TS 33.203 to work in the WebRTC case, the browser executing the WIC shall support TURN over TLS on the non-standard port 80 and 443. It also needs to support the use of HTTP CONNECT for the establishment of the TLS connection to the TURN server.

NOTE 3: At the time of writing, the IETF RTCWEB WG is still working on a firewall traversal solution for WebRTC and the specification is currently in the early draft stage. The TURN based solution is one of the most likely candidates but the details of the solution may still change. Furthermore, in order for the TURN server to authenticate and authorize the WIC, TURN credentials have to be provisioned in the WIC and TURN server. Further analysis is required before such a provisioning mechanism can be specified. Due to these reasons specification of a firewall traversal solution for WebRTC is deferred to future releases.

7 Assessment of solutions

Editor's Note: If needed, this clause will contain assessments of the various solutions.

8 Conclusions and recommendations

Security solutions for the interfaces W1, W2, W4 are out of scope of Rel-12. But example solutions are provided in the present document.

In particular, the following security aspects are not specified in Rel-12:

- The security mechanism for the interface between WIC and eP-CSCF (only example authentication solutions have been provided).
- Whether to use OAuth in conjunction with the Trusted Node Authentication for Scenario 2 and 3; if OAuth is used, how to define the token format and verification procedure, how the WWSF obtains the token and how the token is made available to the WebRTC IMS Client.
- Details on message parameters of IMS AKA procedure.
  These will be described in 3GPP CT1 and CT4 specifications for WebRTC access to IMS.

Annex A:
Secure usage of GBA with UE browser

This clause describes a sequence flow for secure usage of GBA with UE browser as described in normative Annex D of 3GPP TS 33.222 [11].

In this message flow the following architecture is assumed:

- GBA Function: The GBA Function handles establishment of GBA-specific keys. In particular, the establishment of the key Ks can use any of the methods defined by TS 33.220 [12] (e.g. based on AKA or GBA_Digest). The GBA Function is not part of the web browser.

NOTE: In the case of GBA_Digest, the GBA Function treats SIP Digest credentials as specified in Annex N of TS 33.203 [5].

- Web Browser: The web browser is either native or downloaded and contains some functions which support usage of GBA. In particular we have in the architecture:

- GBA_API: Part of the browser that communicates with the GBA Function and receives GBA authentication token material requests from the Javascript code.

- Javascript: Downloaded Javascript code.

- Engine: Sets up communication with the NAF.

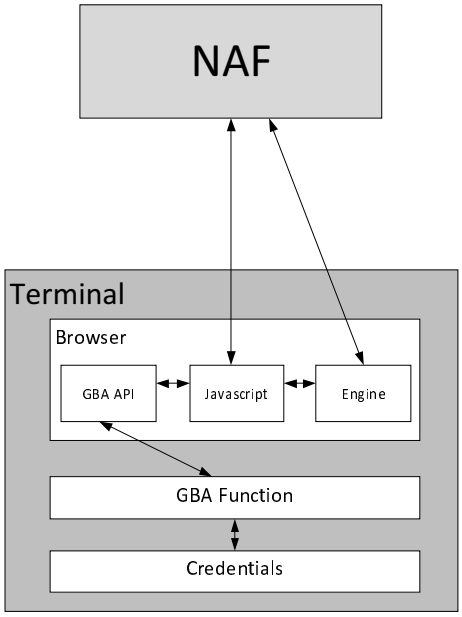

Figure A-1: Example Architecture

Below is a sequence flow diagram of GBA usage in Web context, I.e. within Javascript.

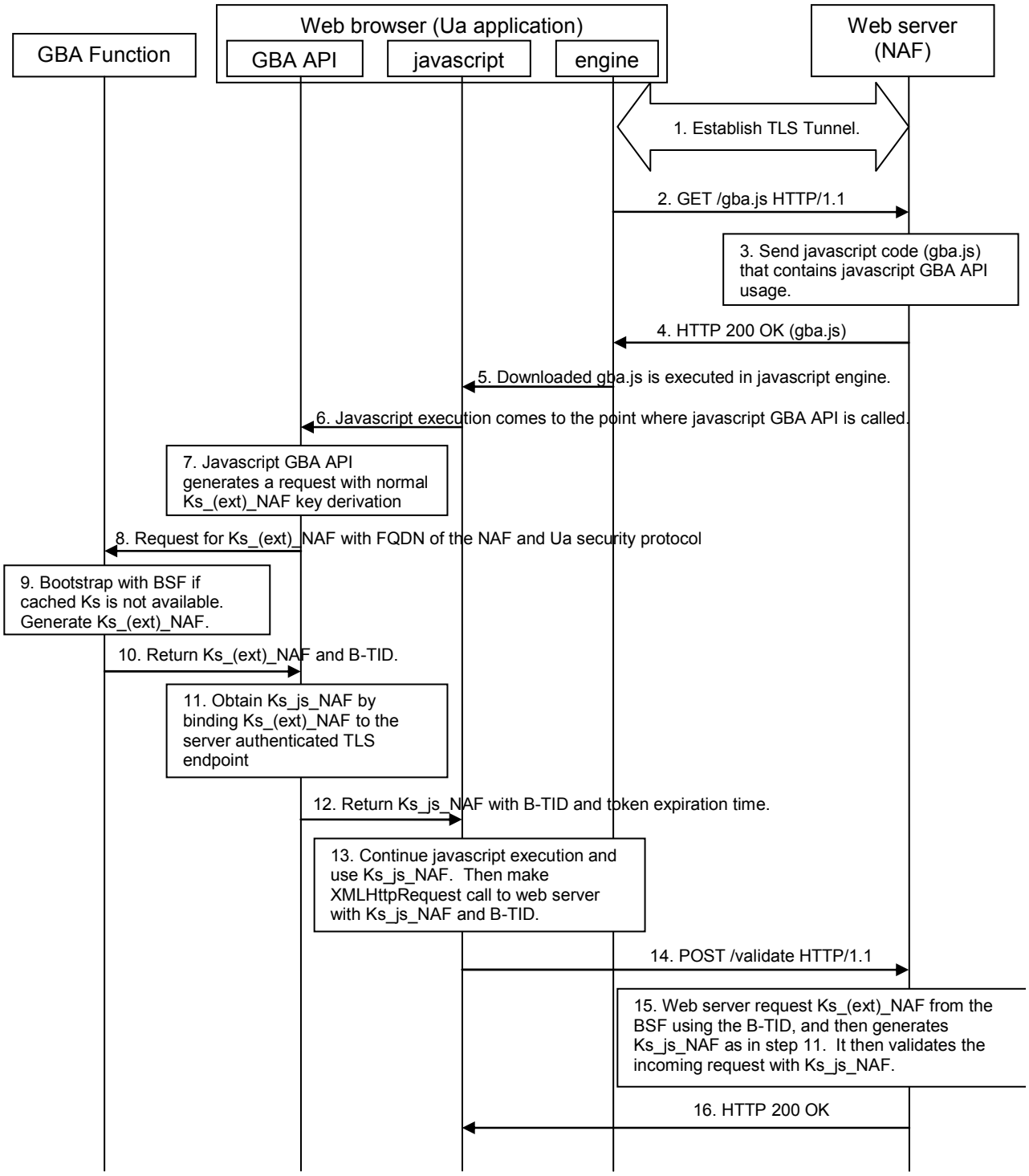

Figure A-2: Example sequence flow

The web browser is considered to be a trusted application in the sense that the user trusts it to handle security related functions properly, I.e. setting TLS sessions with servers, sandboxing the Javascript code that is downloaded from the web servers, and not leaking sensitive information like a password to third parties. In the sequence flow diagram, the web browser is divided into three functional blocks:

- engine module handles the basic functionalities for the web browser like setting up TLS with web servers, downloading web resources from network, and providing the user interface with the end user.

- GBA API module offers the API towards any Javascript executing in the web browser. As Javascript should not be explicitly trusted, the web browser and the GBA API should not reveal any sensitive information to the Javascript, nor should they accept any sensitive information from the Javascript more than necessary.

- Javascript module executes the downloaded Javascript. Any Javascript executing in web browser should be considered not trusted and should not be granted access to sensitive resources or the access to those resources should be controlled.

The communication between web browser and web server in the depicted sequence flow diagram is executed inside a server authenticated TLS tunnel. Also, the web browser is in the process of downloading a html page where one of the linked Javascript resources is "gba.js".

1) The web browser and the web server establish a server authenticated TLS session. The use of TLS message integrity is mandatory, while the use of TLS encryption is optional. All further messages between the web server and UE shall be sent through this tunnel.

2) The web browser engine makes a HTTP GET request to the server to download gba.js resource from the server.

3) The web server sends the gba.js file that contains the Javascript GBA API call on the browser. The gba.js can also contain additional logical elements that make use of the Javascript specific authentication token Ks_js_NAF.

Example on how a GBA API call could look like:

```
document.gba.getGBAToken(successCallback, errorCallback);
```

4) As a HTTP response to the HTTP request made in step 2, the web server returns the gba.js to the web browser.

5) The engine in the web browser starts to execute the Javascript in gba.js in Javascript sandbox.

6) The Javascript comes to a point where a call to GBA API is made.

7) Browser's Javascript GBA API locates the relevant information about the Javascript, I.e. in what html page it is executing, from what url was the html page downloaded from, and which TLS ciphersuite is used in the TLS tunnel. The FQDN of the NAF can be extracted from the url of the web page, and the Ua security protocol identifier can be derived from the used TLS ciphersuite. FQDN of the NAF and the Ua security protocol identifier form the NAF_ID.

8) Browser's Javascript GBA API makes a call to ME's GBA Function with the NAF_ID derived in step 7.

9) The GBA Function bootstraps with the BSF if there is no valid GBA master key Ks. From the Ks, Ks_(ext)_NAF NAF specific key is derived using the NAF_ID.

10) The GBA Function returns the Ks_(ext)_NAF key to browser's Javascript GBA API with the bootstrapping transaction identifier (B-TID).

11) Upon receiving the Ks_(ext)_NAF key, browser's javascript GBA API will derive the Javascript specific authentication token Ks_js_NAF that is bound to the server authenticated TLS session. The values of the bindingType in GBAOptions are "tls-key-extractor" (I.e. RFC 5705 [4] is used with the label " TLS_MK_Extr ") and "tls-server-endpoint" (I.e. RFC 5929 [7] is used), then Ks_js_NAF is derived as:

Ks_js_NAF = KDF (Ks_(ext)_NAF, TLS_MK_Extr, tls-server-endpoint)

The tls-server-endpoint, tls-unique value and TLS_MK_Extr are all related to the TLS connection that established the TLS session in step 1.

12) Browser's Javascript GBA API returns Javascript specific Ks_js_NAF authentication token, B-TID and authentication token lifetime to the executing javascript.

13) The Javascript continues to execute and it uses the Ks_js_NAF authentication token the way the web server has instructed (via Javascript).

Example on how Javascript can extract parameters from result object in Javascript (continued from step 2).

```
function successCallback(result) {
   var token = result.token;
   var btid = result.btid;
   var lifetime = result.expiryTime;
}
```

14) After executing the client side logic, the Javascript makes an XMLHttpRequest (ajax call, HTTP request) to the web server. This request contains at least Ks_js_NAF or hash of it, and B-TID.

15) The web server fetches the Ks_(ext)_NAF key from the BSF, and it then derives the Ks_js_NAF the same way it was done in step 11. The web server will then compare the received Ks_js_NAF with the locally derived one and validate that the TLS session is the same as was used for the request that established the TLS session in step 1.

16) If the received Ks_js_NAF is valid, the web server will continue to process the request made in step 14 and return the result to the web browser (to the Javascript).

Annex B:
Profiling of DTLS-SRTP

The present Annex contains a list of parameters that may be contained in the use_srtp extension in the DTLS extended client hello, according to RFC 5764 [17]. The rest of the DTLS profile is as defined in Annex M of TS 33.328 [6].

SRTP Protection Profiles:

The SRTP protection profile "AES_CM_128_HMAC_SHA1_80", as defined in RFC 5763, is mandatory to support. Support of other protection profiles is optional.

SRTP Master Key Identifier (MKI):

It is optional to use and support. Since a DTLS-SRTP handshake results in single SRTP master key, an endpoint has at most one active master key at any point in time. MKI signalling is therefore typically not required (the major exception would be if the peers perform frequent re-keying) and is not recommended.

Annex C:
Linking IMS identities and web identities - Example security mechanisms

These mechanisms realising REQ 2.5 from clause 5.3 are out of scope of 3GPP. Nevertheless, examples of such mechanisms may be useful as guidance by implementers. The examples given here do not imply that they would be the only possible realisations, others are certainly permitted. But they all have in common that the IMS subscriber or IMS user as well as the WWSF and the IMS provider need to be involved.

Example security mechanism M1:

The IMS subscriber agrees by some unspecified means to linking the two types of identities, the IMPI and the web identity, or to linking the two accounts. The subscription and user data in the IMS and at the WWSF contain a common piece of information linking the two types of identities. The IMS provider and the WWSF exchange this common piece of information for each subscriber/user during a set-up phase, and the WWSF then includes it in the authentication information (token) sent via the WIC to the eP-CSCF during IMS registration. It needs to be determined what this common piece of information could be (e.g. a string pointing to name or email address), and how it could be dynamically kept up-to-date during the lifetime of the system.

Example security mechanism M2:

This mechanism is more detailed than mechanism M1 in that it describes some steps to link the two types of identities. It starts from the WWSF side. It proceeds in the following steps:

- The web user tells the WWSF to link her account with a particular (IMPU).
- The WWSF informs the IMS provider about this request through some out-of-band means.
- The IMS provider derives the IMPI from the IMPU sending from WWSF, or uses the IMPU to query database for the IMPI.
- The IMS provider asks the IMS subscriber relating to that IMPI in a sufficiently secure way (by e.g. web interface, SMS, email, letter), whether the IMS subscriber really meant to link his IMS subscription to this particular web identity.
- The IMS subscriber sends a sufficiently secured confirmation to the IMS provider.
- The IMS provider sends confirmation information containing the IMPI to the WWSF through the out-of-band means.
- The WWSF stores the association between the IMS identity (IMPU, IMPI) and web identity.
  The IMS provider need not store this association.
- The WWSF may send a confirmation to the user.

Annex D:
Mapping OAuth 2.0 to IMS WebRTC

The OAuth 2.0 framework defines the following roles (taken from RFC 6749)

- **Resource owner (a.k.a. the User)** - An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.
- **Resource server (a.k.a. the API server)** - The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.
- **Authorization server** - The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorization.
- **Client** - An application making protected resource requests on behalf of the resource owner and with its authorization. The term 'client' does not imply any particular implementation characteristics (e.g. whether the application executes on a server, a desktop, or other devices).
  The OAuth 2.0 **Client** role is subdivided into a set of client types and profiles.

The Rel-12 architecture for WebRTC IMS Client access to IMS is as follows:

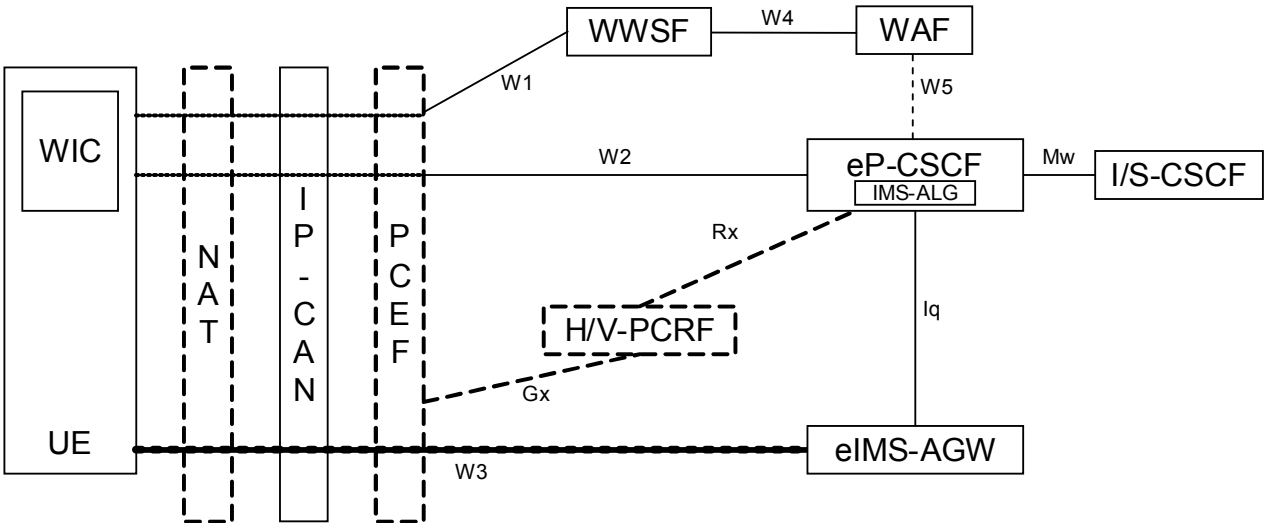

Figure D-1: Architecture for WebRTC IMS Client access to IMS

Various entities and the role they play in the above architecture are as follows:

- A WebRTC IMC Client (WIC) is a WebRTC JavaScript application that provides access to the communication services of the IMS. It is downloaded from the WWSF and executes on any device (UE) that supports a WebRTC Compliant browser.

- A WebRTC Web Server Function (WWSF) is the initial point of contact for the user to access IMS communications services using WebRTC. It provides the WIC for downloading to the browser on the UE.

- The WebRTC Authorization Function (WAF) issues the authorization token after authenticating the user itself as part of the token issuance process, or it trusts the user identity provided by the WWSF.

- The WIC receives the token and constructs the required W2 Register message to be sent to the eP-CSCF embedding the received token as one of the parameters in the message.

- The eP-CSCF can chose to validate the received token by sending it to the WAF over the W5 interface. Once it receives a successful validation message from the WAF, it uses the existing TNA method to trigger IMS Registration with the IMS Core.

One can deduce the following from the above discussion:

1) WAF plays the role of the Authorization server
2) eP-CSCF plays the role of the Resource server
3) User behind the browser is the Resource owner

Mapping OAuth 2.0 Client

The JavaScript based application providing access to the IMS communication services is centrally hosted on the WWSF and available to a user for download to the UE. Once it gets downloaded to the UE (now a WIC), it runs completely in the context of the browser providing authorized access to the IMS services.

This leads to the following models:

a) Model A - WWSF as the OAuth Client

In this model, WWSF is solely responsible to register and interface with the Authorization Server for the purpose of fetching an access token.

- WWSF registers with the OAuth 2.0 compliant WAF. WAF assigns a unique Client ID and Client secret to WWSF. These credentials are stored in WWSF and not exposed to the WIC.
- WIC provides a front end to the OAuth Client and does not directly interact with the Authorization Server.
- Fits well with the following Authorization grant types:
  - Authorization code
  - Client Credentials
- When WIC requests an access token from the Authorization Server, WWSF intercepts this request and embeds Client credentials (including the secret) into the request.
  - In Authorization code grant, WAF authenticates both the user (username/pwd) and WWSF (client ID and client secret)
  - In Client Credentials code grant, WAF ONLY authenticates the WWSF. WAF assumes the user is authenticated independently outside the OAuth 2.0 authorization framework.

b) Model B - WIC as the OAuth Client

In this model, downloaded WIC play the role of OAuth client.

- Before OAuth is used, the client application is first registered with the Authorization server through offline means (not defined by OAuth spec). In response, WAF assigns a unique Client ID to the Client.
- WWSF is in possession of the Client ID and makes it available to WIC when it is downloaded to the UE.
- WIC uses the Client ID in its request to the WAF for an access token.

This model fits well with the following Authorization grant type:

- Implicit grant –

- The User Agent (browser) loads the page from WWSF and executes the WIC JavaScript code
- WIC (OAuth Client) sees that it doesn"t have an Access Token to register with the IMS network and redirects the UA to the Authorization Server with a Client ID and a redirect URL (optional)**.**
- The Authorization Server (WAF) authenticates the user and redirects back to the UA with the access token in the URL Fragment.
- WIC extracts this access token and uses it to register with the IMS network.

Editor"s Note: It is ffs whether registration scenario in TS 33.203 Annex X.4 fits into any of the models described above or needs a new mapping.

Annex E:
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2013-11	SA3#73				Initial TR version	-	0.1.0
2014-01	SA3#74				Updated according to SA3#74 decisions	0.1.0	0.2.0
2014-04	SA3#74 bis				Updated according to SA3#74bis decisions	0.2.0	0.3.0
2014-05	SA3#75	S3-140970			Updated according to SA3#75 decisions	0.3.0	0.4.0
2014-06	-				MCC clean-up for submission to SA#64 for Information. Changed "must" into "shall", by indicating that this is an informative document.	0.4.0	1.0.0
2014-08	SA3#76				Updated according to SA3#76 decisions: S3-142284, S3-142285, S3-142287.	1.0.0	1.1.0
2014-09	SA#65	SP-140582			Presented for approval	1.1.0	2.0.0
					Upgrade to Rel-12	2.0.0	12.0.0

History

Document history		
V12.0.0	October 2014	Publication