

**Security Mechanism Names for Media**  
**draft-dawes-sipcore-mediasec-parameter-10.txt**

Abstract

Negotiating the security mechanisms used between a Session Initiation Protocol (SIP) user agent and its next-hop SIP entity is described in [2]. This document adds the capability to distinguish security mechanisms that apply to the media plane by defining a new Session Initiation Protocol (SIP) header field parameter to label such security mechanisms.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 29, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Problem Statement . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Access Network Protection . . . . .	<a href="#">4</a>
<a href="#">4.</a>	Solution . . . . .	<a href="#">4</a>
<a href="#">4.1.</a>	Signaling security negotiation . . . . .	<a href="#">4</a>
<a href="#">4.2.</a>	Header fields for signaling security negotiation . . . . .	<a href="#">5</a>
<a href="#">4.3.</a>	Syntax . . . . .	<a href="#">5</a>
<a href="#">4.4.</a>	Protocol Operation . . . . .	<a href="#">5</a>
<a href="#">4.4.1.</a>	The "mediasec" Header Field Parameter . . . . .	<a href="#">5</a>
<a href="#">4.4.2.</a>	Client Initiated . . . . .	<a href="#">5</a>
<a href="#">4.5.</a>	Security Mechanism Initiation . . . . .	<a href="#">7</a>
<a href="#">4.6.</a>	Duration of Security Associations . . . . .	<a href="#">7</a>
<a href="#">4.7.</a>	Summary of Header Field Use . . . . .	<a href="#">7</a>
<a href="#">5.</a>	Backwards Compatibility . . . . .	<a href="#">7</a>
<a href="#">6.</a>	Examples . . . . .	<a href="#">7</a>
<a href="#">6.1.</a>	Initial Registration 3GPP . . . . .	<a href="#">7</a>
<a href="#">6.2.</a>	Re-Registration 3GPP . . . . .	<a href="#">12</a>
<a href="#">6.3.</a>	Client Initiated as per <a href="#">RFC 3329</a> . . . . .	<a href="#">14</a>
<a href="#">6.4.</a>	Server Initiated as per <a href="#">RFC 3329</a> . . . . .	<a href="#">16</a>
<a href="#">6.5.</a>	Using Media Plane Security . . . . .	<a href="#">18</a>
<a href="#">7.</a>	Formal Syntax . . . . .	<a href="#">20</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">20</a>
<a href="#">9.</a>	IANA Considerations . . . . .	<a href="#">20</a>
<a href="#">9.1.</a>	Registry for Media Plane Security Mechanisms . . . . .	<a href="#">20</a>
<a href="#">9.2.</a>	Registration Template . . . . .	<a href="#">21</a>
<a href="#">9.3.</a>	Header Field Names . . . . .	<a href="#">21</a>
<a href="#">9.4.</a>	Response Codes . . . . .	<a href="#">21</a>
<a href="#">10.</a>	Security Considerations . . . . .	<a href="#">21</a>
<a href="#">11.</a>	References . . . . .	<a href="#">21</a>
<a href="#">11.1.</a>	Normative References . . . . .	<a href="#">21</a>
<a href="#">11.2.</a>	Informative References . . . . .	<a href="#">22</a>
<a href="#">Appendix A.</a>	Additional stuff . . . . .	<a href="#">22</a>
	Author's Address . . . . .	<a href="#">22</a>



## **1. Problem Statement**

In the 3GPP defined architecture and SIP profile for packet-domain communication, SIP signaling is security protected at the network layer but media-plane traffic is not (it is protected by the cellular wireless access). The SIP signaling security used by 3GPP runs from the user device to the first hop proxy and negotiation of security mechanism and the start of security protection is described in [2]. Because the 3GPP architecture also allows access technologies that do not protect media, e.g. WiFi, this document extends the negotiation of security mechanism to the media plane. During previous discussion of the topic of media plane security it was suggested that DTLS-SRTP should be used, but 3GPP considered this impractical to implement in the 3GPP-defined architecture and also limited in terms of meeting all 3GPP requirements which include protection of non-RTP media such as MSRP.

The purpose of this specification is to define a new header field parameter for the Session Initiation Protocol (SIP) that distinguishes security mechanisms that apply to the media plane and to create an IANA registry for these mechanisms. This header field parameter may be used with the Security-Client, Security-Server, and Security-Verify header fields defined by [2].

The header field parameter introduced by this draft originates from 3GPP specifications and related procedures and header field values are also described in 3GPP specifications, primarily in 3GPP TS 33.328 [2]. The purpose of this draft is to name the header field parameter, give several illustrative examples to make it clear how it is used, and set up an IANA registry for existing and future values. The draft does not propose that IETF defines any new security setup procedures, ciphering, integrity protection etc.

## **2. Introduction**

[2] describes negotiation of a security mechanism for SIP signaling between a UAC and its first hop proxy and allows a client or network to ensure that protection of SIP signaling is turned on when the client registers with the network. SIP signaling is then protected as it traverses the access network. To enable similar protection for media, this document enables client and network to exchange their security capabilities for the media plane combined with the negotiation described in [2]. Similar to the signaling plane, the evolution of security mechanisms for media often introduces new algorithms, or uncovers problems in existing ones, making capability exchange of such mechanisms a necessity.



### **3. Access Network Protection**

Some access technologies, such as many cellular wireless accesses, protect the data passed over them by default but some, such as WLAN, do not. For accesses with no inherent protection, it is useful for the media controlled by SIP signaling to be protected by default because of vulnerability to eavesdropping. It is currently possible for a UA to request protection of the media plane end-to-end by including the crypto attribute in SDP at session setup. This does not guarantee protection however, because it relies on support of encryption by the called UA, or by another entity in the path taken by the media. In some cases, the session will originate in an access that protects the media and terminate in one that does not, meaning that media is protected in all but some hops of its path. In cases where the same provider supplies the user equipment and provides the IP access, the IP access technology that the UA will use is predictable and the media is vulnerable only as far as the core network. In such cases, the user equipment it is possible to protect the media plane by encrypting at the UA and decrypting at the edge of the core network, and for the user agent that originates or terminates the session to expect the edge of the core network to be capable of encrypting and decrypting media. The header field parameter described in this document enables this case of first-hop protection, which is typically provided by default to a user agent.

### **4. Solution**

#### **4.1. Signaling security negotiation**

A specification already exists for setting up security for SIP signaling between a client and its first-hop proxy, as defined in [2] which gives an overview of the mechanism as follows:

```
1. Client -----client list-----> Server
2. Client <-----server list----- Server
3. Client -----(turn on security)----- Server
4. Client -----server list-----> Server
5. Client <-----ok or error----- Server
```

Figure 1: Security agreement message flow from [RFC 3329](#)

The security mechanism above ensures that SIP signaling is protected between a client and its first hop entity but the media plane is still unprotected. This document proposes that client and server additionally exchange their media plane security capabilities at steps 1 and 2. Media plane security needs to be applied on a per-



media basis at the time that media is initiated. Therefore the client and server need not turn on media plane security immediately.

This document defines the "mediasec" header field parameter that labels any of the Security-Client:, Security-Server:, or Security-Verify: header fields as applicable to the media plane and not the signaling plane.

#### **4.2. Header fields for signaling security negotiation**

The "mediasec" header field parameter defined in this document is used with procedures defined in [2] to distinguish media plane security, with the difference that media plane security need not be started immediately and can be applied and removed on-the-fly as media are added and removed within a session. The SIP responses that can contain the Security-Client, Security-Server, and Security-Verify header fields are SIP responses 421 (Extension Required) and 494 (Security Agreement Required) as defined in [2].

#### **4.3. Syntax**

This document does not define any new SIP header fields, it defines a header field parameter for header fields Security-Client, Security-Server and Security-Verify defined in [2].

#### **4.4. Protocol Operation**

##### **4.4.1. The "mediasec" Header Field Parameter**

The "mediasec" header field parameter may be used in the Security-Client, Security-Server, or Security-Verify header fields defined in [2] to indicate that a header field applies to the media plane. Any one of the media plane security mechanisms supported by both client and server, if any, may be applied when a media stream is started. Or a media stream may be set up without security.

Values in the Security-Client, Security-Server, or Security-Verify header fields labelled with the "mediasec" header field parameter are specific to the media plane and specific to the secure media transport protocol used on the media plane.

##### **4.4.2. Client Initiated**

A client wishing to use the security capability exchange of this specification MUST add a Security-Client header field to a request addressed to its first-hop proxy (i.e., the destination of the request is the first-hop proxy). This header field contains a list of all the media plane security mechanisms that the client supports.





The client SHOULD NOT add preference parameters to this list. The client MUST add a "mediasec" header field parameter to the Security-Client header field.

The contents of the Security-Client header field may be used by the server to include any necessary information in its response.

As described in [2], the response will be 494 if the client includes "sec-agree" in the Require and Proxy-Require header fields, or a 2xx response if the Require and Proxy-Require header fields do not contain "sec-agree". The server MUST add its list to the response even if there are no common security mechanisms in the client's and server's lists. The server's list MUST NOT depend on the contents of the client's list.

Any subsequent SIP requests sent by the client to that server MAY make use of the media security capabilities exchanged in the previous step by including media plane security parameters in SDP in the session or the media description. These requests MUST contain a Security-Verify header field that mirrors the server's list received previously in the Security-Server header field.

The server MUST check that the security mechanisms listed in the Security-Verify header field of incoming requests correspond to its static list of supported security mechanisms.

Note that, following the standard SIP header field comparison rules defined in [3], both lists have to contain the same security mechanisms in the same order to be considered equivalent. In addition, for each particular security mechanism, its parameters in both lists need to have the same values.

The server can proceed processing a particular request if, and only if, the list was not modified. If modification of the list is detected, the server MUST respond to the client with a 494 (Security Agreement Required) response. This response MUST include the server's unmodified list of supported security mechanisms.

Once security capabilities have been exchanged between two SIP entities, the same SIP entities MAY use the same security when communicating with each other in different SIP roles. For example, if a UAC and its outbound proxy exchange some media-plane security mechanisms, they may try to use the same security for incoming requests (i.e., the UA will be acting as a UAS).

The user of a UA SHOULD be informed about the results of the security mechanism agreement. The user MAY decline to accept a particular



security mechanism, and abort further SIP communications with the peer.

#### **4.5. Security Mechanism Initiation**

Once the client chooses a security mechanism from the list received in the Security-Server header field from the server, it MAY initiate that mechanism on a session level, or on a media level when it initiates new media in an existing session.

#### **4.6. Duration of Security Associations**

Once media-plane security capabilities have been exchanged, both the server and the client need to know until when they can be used. The media plane security mechanism setup is valid for as long as the UA has a SIP signaling relationship with its first-hop proxy or until new keys are exchanged in SDP. The SDP used to set up media plane security will be protected by a security association used to protect SIP signaling and the media plane security mechanism can be used until the signaling plane security association expires.

#### **4.7. Summary of Header Field Use**

The header fields defined in this document may be used to exchange supported media plane security mechanisms between a UAC and other SIP entities including UAS, proxy, and registrar. Information about the use of headers in relation to SIP methods and proxy processing is given in [2] Table 1.

### **5. Backwards Compatibility**

Security mechanisms that apply to the media plane only MUST NOT have the same name as any signaling plane mechanism. If a signaling plane security mechanism name is re-used for the media plane and distinguished only by the "mediasec" parameter, then implementations that do not recognize the "mediasec" parameter may incorrectly use that security mechanism for the signaling plane.

### **6. Examples**

The following examples illustrate the use of the mediasec header field parameter defined above.

#### **6.1. Initial Registration 3GPP**

At initial registration, the client includes its supported media plane security mechanisms in the SIP REGISTER request. The first-hop



proxy returns its supported media plane security mechanisms in the SIP 401 (Unauthorized) response.

As per [2], a UA negotiates the security mechanism for the media plane to be used with its outbound proxy without knowing beforehand which mechanisms the proxy supports, as shown in Figure 2 below.

UAC	Proxy	Registrar
user1_public1@home1.net	pcscf1.home1.net	registrar.home1.net
------(1) REGISTER---->		
Security-Client: sdes-srtp; mediasec		
	---(2) REGISTER--->	
	<----(3) 401-----	
<------(4) 401-----		
Security-Server: sdes-srtp; mediasec		
------(5) REGISTER----->		
Security-Client: sdes-srtp; mediasec		
Security-Verify: sdes-srtp; mediasec		
	---(6) REGISTER--->	
	<----(7) 200 OK----	
<------(8) 200 OK-----		
------(9) INVITE----->		
Security-Verify: sdes-srtp; mediasec		
Content-Type: application/sdp		
a=3ge2ae		
a=crypto:1 AES_CM_128_HMAC_SHA1_80		
inline:WVNfX19zZW1jdGwgKCKgewkyMjA7fQp9CnVubGVz 2^20 1:4		
FEC_ORDER=FEC_SRTP		

Figure 2: Exchange of Media Security Mechanisms at Initial Registration

The UAC sends a REGISTER request (1) to its outbound proxy indicating the security mechanisms for the media plane and that it supports in a



Security-Client: header field. Indication of media security mechanisms is identified by the "mediasec" header field parameter.

The outbound proxy forwards the REGISTER request (2) to the registrar with the Security-Client: header field removed as described in [2].

The registrar responds with a 401 (Unauthorized) response (3) to the REGISTER request.

The outbound proxy responds forwards the 401 (Unauthorized) response (4) to the UAC with its own list of security mechanisms for the media plane in the Security-Server: header field. Security mechanisms for the media plane are distinguished by the "mediasec" header field parameter.

The UAC sends a second REGISTER request (5) using the security credentials it received in the 401 (Unauthorized) response. The UAC includes the security mechanisms for the media plane and that it supports in a Security-Client: header field. The UAC also echos the list of security mechanisms it received from the outbound proxy in the Security-Server: header field. Media security mechanisms are distinguished by the "mediasec" header field parameter.

The REGISTER request is forwarded to the registrar (6) and the registrar responds with 200 OK (7), which is forwarded to the UAC (8).

When the connection is successfully established, the UAC sends an INVITE request(9) including an SDP description of the media plane security to be used (a="e2ae" and a crypto attribute). This INVITE contains a copy of the server's security list in a Security-Verify header field. The server verifies it, and since it matches its static list, it processes the INVITE and forwards it to the next hop.

If this example was run without the Security-Server header field in Step (2), the UAC would not know what kind of security the other one supports, and would be forced to make error-prone trials.

More seriously, if the Security-Verify header field was omitted in Step (3), the whole process would be prone to MitM attacks. An attacker could remove the media plane security description from the header in Step (1), therefore preventing protection of the media plane.

(1) REGISTER sip:registrar.home1.net SIP/2.0

Via: SIP/2.0/UDP

[5555::aaa:bbb:ccc:ddd];comp=sigcomp;branch=z9hG4bKnashds7





Max-Forwards: 70  
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11  
From: <sip:user1\_public1@home1.net>;tag=4fa3  
To: <sip:user1\_public1@home1.net>  
Contact: <sip:[5555::aaa:bbb:ccc:ddd];comp=sigcomp>;expires=600000  
Call-ID: apb03a0s09dkjdfglkj49111  
Authorization: Digest username="user1\_private@home1.net",  
realm="registrar.home1.net", nonce="", uri="sip:registrar.home1.net",  
response=""  
Security-Client: ipsec-3gpp; alg=hmac-sha-1-96; spi-c=23456789; spi-  
s=12345678; port-c=2468; port-s=1357  
Security-Client: sdes-srtp; mediasec \*\*\*new\*\*\*  
Require: sec-agree  
Proxy-Require: sec-agree  
CSeq: 1 REGISTER  
Supported: path  
Content-Length: 0

(2) REGISTER sip:registrar.home1.net SIP/2.0  
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP  
[5555::aaa:bbb:ccc:ddd];comp=sigcomp;branch=z9hG4bKnashds7  
Max-Forwards: 69  
P-Access-Network-Info:  
Path:  
Require:  
P-Visited-Network-ID:  
P-Charging-Vector:  
From:  
To:  
Contact:  
Call-ID:  
Authorization:  
CSeq:  
Supported:  
Content-Length:

(3) SIP/2.0 401 Unauthorized  
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP  
[5555::aaa:bbb:ccc:ddd];comp=sigcomp;branch=z9hG4bKnashds7  
From: <sip:user1\_public1@home1.net>;tag=4fa3  
To: <sip:user1\_public1@home1.net>; tag=5ef4  
Call-ID: apb03a0s09dkjdfglkj49111  
WWW-Authenticate: Digest realm="registrar.home1.net", nonce=base64(RAND +  
AUTN + server specific data), algorithm=AKAv1-MD5,  
ik="00112233445566778899aabbccddeeff", ck="ffeeddcbbbaa11223344556677889900"  
CSeq: 1 REGISTER  
Content-Length: 0

(4) SIP/2.0 401 Unauthorized  
Via: SIP/2.0/UDP  
[5555::aaa:bbb:ccc:ddd];comp=sigcomp;branch=z9hG4bKnashds7  
From:  
To:  
Call-ID:  
WWW-Authenticate: Digest realm="registrar.home1.net", nonce=base64(RAND +  
AUTN + server specific data), algorithm=AKAv1-MD5  
Security-Server: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432;  
spi-s=87654321; port-c=8642; port-s=7531

Security-Server: sdes-srtp; mediasec \*\*\*new\*\*\*  
CSeq:  
Content-Length:

(5) REGISTER sip:registrar.home1.net SIP/2.0  
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:  
1357;comp=sigcomp;branch=z9hG4bKnashds7  
Max-Forwards: 70  
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11  
From: <sip:user1\_public1@home1.net>;tag=4fa3  
To: <sip:user1\_public1@home1.net>  
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>;expires=600000  
Call-ID: apb03a0s09dkjdfglkj49111  
Authorization: Digest username="user1\_private@home1.net",  
realm="registrar.home1.net", nonce=base64(RAND + AUTN + server specific data),  
algorithm=AKAV1-MD5, uri="sip:registrar.home1.net",  
response="6629fae49393a05397450978507c4ef1"  
Security-Client: ipsec-3gpp; alg=hmac-sha-1-96; spi-c=23456789; spi-  
s=12345678; port-c=2468; port-s=1357  
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432;  
spi-s=87654321; port-c=8642; port-s=7531  
Security-Client: sdes-srtp; mediasec \*\*\*new\*\*\*  
Security-Verify: sdes-srtp; mediasec \*\*\*new\*\*\*  
Require: sec-agree  
Proxy-Require: sec-agree  
CSeq: 2 REGISTER  
Supported: path  
Content-Length: 0

(6) REGISTER sip:registrar.home1.net SIP/2.0  
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP  
[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7  
Max-Forwards: 69  
P-Access-Network-Info:  
Path:  
Require:  
P-Visited-Network-ID:  
P-Charging-Vector:  
From:  
To:  
Contact:  
Call-ID:  
Authorization:  
CSeq:  
Supported:  
Content-Length:

(7) SIP/2.0 200 OK

Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP  
[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7  
Path: <sip:term@pcscf1.visited1.net;lr>  
From:  
To:  
Call-ID:  
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>;expires=600000  
CSeq:

The UAC sends a REGISTER request (1) and includes the security mechanisms for the media plane and that it supports in a Security-

Client: header field. The UAC also echos the list of security

Dawes

Expires November 29, 2019

[Page 12]

mechanisms it received from the outbound proxy in the Security-Server: header field. Media security mechanisms are distinguished by the "mediasec" header field parameter. In the example below, the Security-Verify: header field is included as required by [5] clause 5.1.1.4.2 when setting up ipsec-3gpp signaling plane security.

The REGISTER request is forwarded to the registrar (2) and the registrar responds with 200 OK (3), which is forwarded to the UAC (4).

```
(1) REGISTER sip:registrar.home1.net SIP/2.0
    Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:
1357;comp=sigcomp;branch=z9hG4bKnashds7
    Max-Forwards: 70
    P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
    From: <sip:user1_public1@home1.net>;tag=4fa3
    To: <sip:user1_public1@home1.net>
    Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>;expires=600000
    Call-ID: apb03a0s09dkjdfgklkj49111
    Authorization: Digest username="user1_private@home1.net",
realm="registrar.home1.net", nonce=base64(RAND + AUTN + server specific
data), algorithm=AKAv1-MD5, uri="sip:registrar.home1.net",
response="6629fae49393a05397450978507c4ef1", integrity-protected="yes"
    Security-Client: ipsec-3gpp; alg=hmac-sha-1-96; spi-c=23456789; spi-
s=12345678; port-c=2468; port-s=1357
    Security-Client: sdes-srtp; mediasec ***new***
    Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432;
spi-s=87654321; port-c=8642; port-s=7531
    Security-Verify: sdes-srtp; mediasec ***new***
    Require: sec-agree
    Proxy-Require: sec-agree
    CSeq: 3 REGISTER
    Supported: path
    Content-Length: 0
```

```
(2) REGISTER sip:registrar.home1.net SIP/2.0
    Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
    P-Access-Network-Info:
    Max-Forwards: 69
    Path:
    Require:
    P-Visited-Network-ID:
    P-Charging-Vector:
    From:
```



To:  
Contact:  
Call-ID:  
Authorization:  
CSeq:  
Supported:  
Content-Length:

Dawes

Expires November 29, 2019

[Page 13]

```
(3) SIP/2.0 200 OK
    Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
    Path:
    From:
    To:
    Call-ID:
    Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>;expires=600000
    CSeq:
    Date: Wed, 11 July 2001 08:49:37 GMT
    P-Associated-URI: <sip:user1_public2@home1.net>,
<sip:user1_public3@home1.net>, <sip:+1-212-555-1111@home1.net;user=phone>
    Content-Length:

(4) SIP/2.0 200 OK
    Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:
1357;comp=sigcomp;branch=z9hG4bKnashds7
    Path:
    From:
    To:
    Call-ID:
    Contact:
    CSeq:
    Date:
    P-Associated-URI:
    Content-Length:
```

Figure 5: Use of mediasec parameter

### **6.3. Client Initiated as per [RFC 3329](#)**

Media plane security mechanisms are also exchanged at client initiated security negotiation described in [\[2\]](#).

Dawes

Expires November 29, 2019

[Page 14]

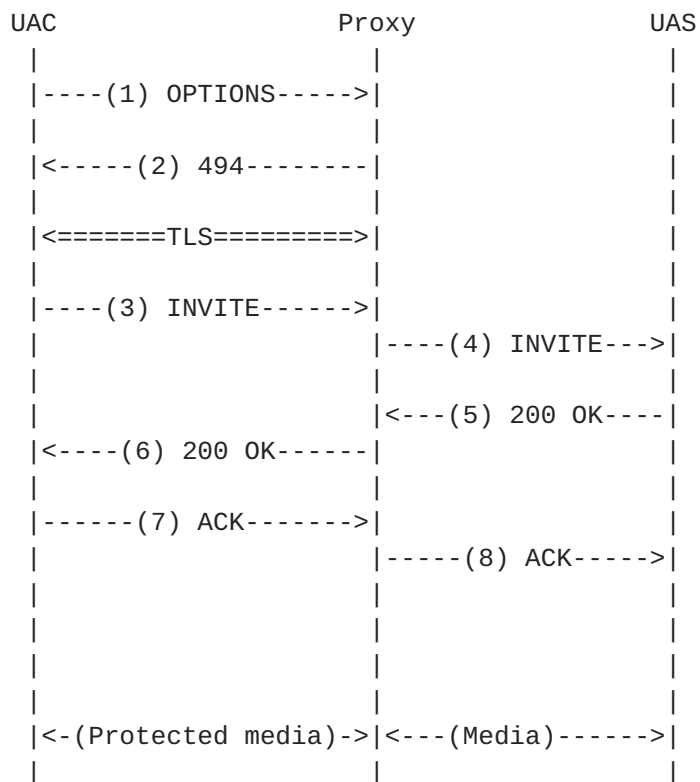


Figure 6: Negotiation Initiated by the Client.

After exchange of security capabilities, the UAC sends an INVITE request(3) including an SDP description of the media plane security to be used (`a="e2ae"` and a crypto attribute). This INVITE contains a copy of the server's security list in a Security-Verify header field. The server verifies it, and since it matches its static list, it processes the INVITE and forwards it to the next hop.



- (1) OPTIONS sip:proxy.example.com SIP/2.0  
Security-Client: tls  
Security-Client: digest  
Security-Client: sdes-srtp; mediasec  
Require: sec-agree  
Proxy-Require: sec-agree
- (2) SIP/2.0 494 Security Agreement Required  
Security-Server: ipsec-ike;q=0.1  
Security-Server: tls;q=0.2  
Security-Server: sdes-srtp; mediasec
- (3) INVITE sip:proxy.example.com SIP/2.0  
Security-Verify: ipsec-ike;q=0.1  
Security-Verify: tls;q=0.2  
Security-Verify: sdes-srtp; mediasec  
Route: sip:callee@domain.com  
Require: sec-agree  
Proxy-Require: sec-agree

#### **6.4. Server Initiated as per [RFC 3329](#)**

Media plane security mechanisms are also exchanged at server initiated security negotiation described in [\[2\]](#).



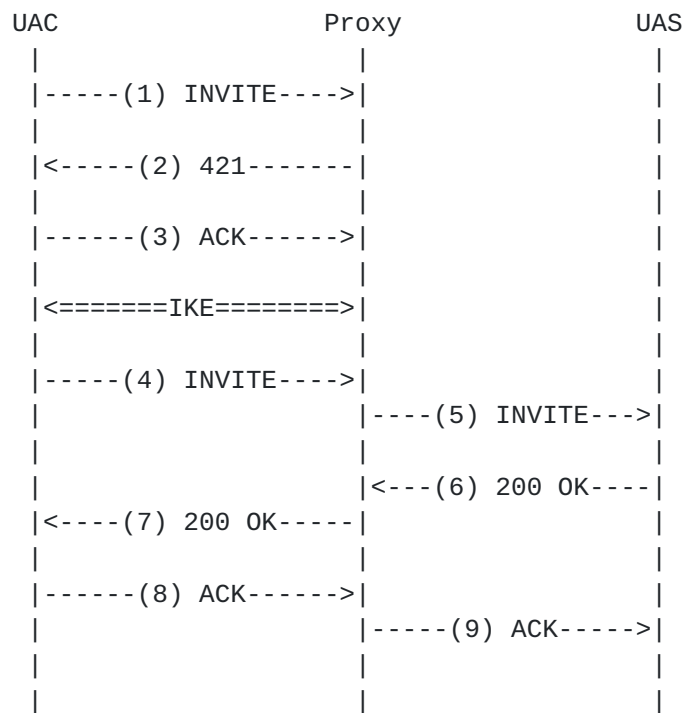


Figure 7: Negotiation Initiated by the Server.

Media security mechanisms are included in Security-Server: and Security-Client: header fields in the same way as signaling security mechanisms.

- (1) INVITE sip:uas.example.com SIP/2.0
- (2) SIP/2.0 421 Extension Required  
 Security-Server: ipsec-ike;q=0.1  
 Security-Server: tls;q=0.2  
 Security-Server: mechanism; mediasec
- (4) INVITE sip:uas.example.com SIP/2.0  
 Security-Verify: ipsec-ike;q=0.1  
 Security-Verify: tls;q=0.2  
 Security-Verify: mechanism; mediasec

Figure 8: Negotiation Initiated by the Server.





### **[6.5.](#) Using Media Plane Security**

To request end to access edge media security either on a session or media level the UE sends, for example, an SDP Offer for an SRTP stream containing one or more SDES crypto attributes, each with a key and other security context parameters required according to [\[4\]](#), together with the attribute "a=3ge2ae".

```
(3) INVITE sip:bob@ua2.example.com SIP/2.0
Security-Verify: ipsec-ike;q=0.1
Security-Verify: tls;q=0.2
Security-Verify: sdes-srtp;mediasec
Route: proxy.example.com
Require: sec-agree
Proxy-Require: sec-agree

Via: SIP/2.0/TCP proxy.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@ua1.example.com>;tag=9fxced76s1
To: Bob <sip:bob@ua2.example.com>

Call-ID: 3848276298220188511@ua1.example.com
CSeq: 1 INVITE
Contact: <sip:alice@ua1.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 285

v=0
o=alice 2890844526 2890844526 IN IP4 ua1.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/SAVP 0
a=3ge2ae
a=crypto:1 AES_CM_128_HMAC_SHA1_80
      inline:WVNfX19zZW1jdGwgKCKgewkyMjA7fQp9CnVubGVz|2^20|1:4
      FEC_ORDER=FEC_SRTTP
a=rtpmap:0 PCMU/8000

(4) INVITE sip:bob@ua2.example.com SIP/2.0
    Route: sip:proxy.example.com

(5) SIP/2.0 200 OK

(6) SIP/2.0 200 OK
    Security-Server: tls;q=0.2
    Security-Server: sdes-srtp;mediasec
    a=3ge2ae
    a=crypto:1 AES_CM_128_HMAC_SHA1_80
    a=crypto:1 AES_CM_128_HMAC_SHA1_80
    inline:PS1uQCvEECFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:4
```

Figure 9: Using media security



## 7. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in [RFC 5234](#) [[RFC5234](#)].

"mediasec" is a "header field parameter", as defined by [[RFC3968](#)].

Header Field Name in which the parameter can appear.

Security-Client

Security-Server

Security-Verify

Header Fields	Parameter Name	Values	Reference
-----	-----	-----	-----
Security-Client	mediasec	No	[this document]
Security-Server	mediasec	No	[this document]
Security-Verify	mediasec	No	[this document]

Name of the Header Field Parameter being registered.

"mediasec"

## 8. Acknowledgements

Remember, it's important to acknowledge people who have contributed to the work.

This template was extended from an initial version written by Pekka Savola and contributed by him to the xml2rfc project.

## 9. IANA Considerations

This specification creates a new registry for media plane security mechanisms.

### 9.1. Registry for Media Plane Security Mechanisms

The IANA has created a subregistry for media plane security mechanism token values to be used with the 'mediasec' header field parameter under the Session Initiation Protocol (SIP) Parameters registry.



Security Mechanism Name for Media	Reference
-----	-----

As per the terminology in [[RFC5226](#)], the registration policy for new media plane security mechanism token values shall be 'Specification Required'.

## **[9.2.](#) Registration Template**

To: ietf-sip-sec-agree-mechanism-name@iana.org Subject: Registration of a new SIP Security Agreement mechanism

Mechanism Name:

(Token value conforming to the syntax described in [Section 4.3.](#))

Published Specification(s):

(Descriptions of new SIP media plane security agreement mechanisms require a published specification.)

## **[9.3.](#) Header Field Names**

This specification registers no new header fields.

## **[9.4.](#) Response Codes**

This specification registers no new response codes.

## **[10.](#) Security Considerations**

This specification is an extension of [[2](#)] and as such shares the same security considerations.

A further consideration of this specification is protection of the cryptographic key to be used for SRTP and carried in SDP. In order to protect this key, one of the security mechanisms defined in [[2](#)] SHOULD be used in parallel with this specification.

## **[11.](#) References**

### **[11.1.](#) Normative References**

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997, <<http://xml.resource.org/public/rfc/html/rfc2119.html>>.





- [2] Arkko, J., Torvinen, V., Camarillo, G., Niemi, A., and T. Haukka, "Security Mechanism Agreement for the Session Initiation Protocol (SIP)", [RFC 3329](#), DOI 10.17487/RFC3329, January 2003, <<https://www.rfc-editor.org/info/rfc3329>>.
- [3] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [4] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", [RFC 4568](#), DOI 10.17487/RFC4568, July 2006, <<https://www.rfc-editor.org/info/rfc4568>>.

## **11.2. Informative References**

- [5] 3GPP, "IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3", 3GPP TS 24.229 10.13.0, September 2013.
- [6] 3GPP, "IP Multimedia Subsystem (IMS) media plane security", 3GPP TS 33.328, September 2014.

## **Appendix A. Additional stuff**

You can add appendices just as regular sections, the only difference is that they go within the "back" element, and not within the "middle" element. And they follow the "reference" elements.

### **Author's Address**

Peter Dawes  
Vodafone Group Services Ltd.  
Newbury  
UK

Email: [peter.dawes@vodafone.com](mailto:peter.dawes@vodafone.com)

