Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 6, 2012

P. Dawes
Vodafone Group

February 3, 2012

Capability Exchange for Media Plane Security
draft-dawes-dispatch-mediasec-parameter-05.txt

Abstract

Negotiating the security mechanisms used between a Session Initiation Protocol (SIP) user agent and its next-hop SIP entity is already described in an RFC. This document extends negotiation of a security mechanism to the media plane by defining a new Session Initiation Protocol (SIP) header field parameter to label security mechanisms that apply to the media plane.

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 RFC 2119 [3].

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 http://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 August 6, 2012.

Copyright Notice

Table of Contents

1. Introduction

RFC 3329 [4] describes negotiation of a security mechanism for SIP signalling between a UAC and its first hop proxy. This document extends the concept of security negotiation by adding exchange of security capability for the media plane. Similar to the signalling plane, the evolution of security mechanisms for media often introduces new algorithms, or uncovers problems in existing ones, making negotiation of mechanisms a necessity.

The purpose of this specification is to define negotiation functionality for the Session Initiation Protocol (SIP) [1]. This negotiation is intended to work only between a UA and its first-hop SIP entity.

1.1. Motivations

1.1.1. Access Network Protection

Some access technologies protect the data passed over them by default, for example many cellular wireless accesses, but some do not, for example WLAN. For accesses with no protection, it is useful for the media controlled by SIP signalling 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. This document describes this case of first-hop protection, which is typically provided by default to a user agent. Both media and signalling must pass through the entity at the edge of the core network, which must therefore be a back-to-back user agent (B2BUA).

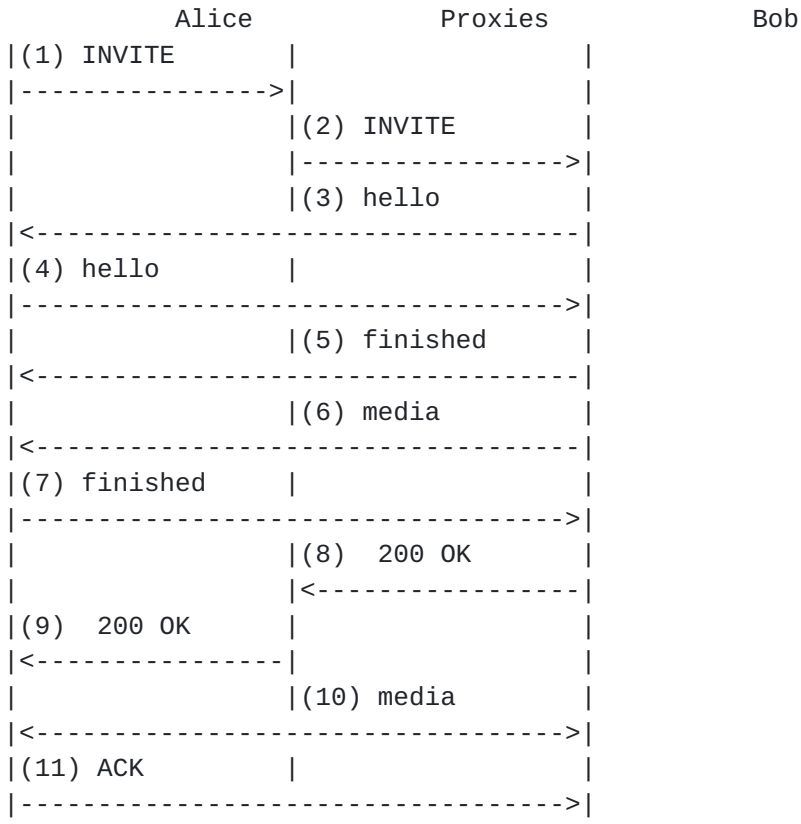
End to access edge media protection described in this document is not a substitute for end-to-end media protection. A user agent requests end-to-access-edge media protection by including a "a=3ge2ae" SDP attribute at session setup. If this attribute is not included, then

end-to-end protection is expected by the user agent and protection MUST NOT fall back to end-to-access-edge protection.

1.1.2. DTLS-SRTP

1.1.2.1. Overview of Operation

DTLS-SRTP is described in RFC 5763 [10], which shows the basic message flow in Figure 1. The DTLS handshake takes place when Bob receives the initial INVITE request. Alice uses the source IP address of the DTLS "hello" message (3) as the destination address for "hello" message (4) even though Bob does not provide a contact address until the 200 OK message (8).



Message (1): INVITE Alice -> Proxy

Figure 1: Security capability exchange message flow

1.1.2.2. Limitations

Reasons why DTLS-SRTP is not suitable for some networks are detailed in 3GPP TR 33.828 [14] and summarized below.

In some networks, media packets are blocked between Alice and Bob until Alice receives the 200 (OK) message 9, which blocks the DTLS handshake until after Alice receives message (9). Media will not be successfully exchanged unless the DTLS handshake is re-attempted after message (9) 200 OK. Even if the handshake is re-attempted, the media will be clipped until the handshake is complete.

Some networks are required by regulation to provide lawful intercept, and no method compatible with DTLS-SRTP is available when a UA is outside its own network (i.e., roaming). Also, lawful intercept would mandate all users to disclose all their keys all the time, which might delay communication setup as networks need the keys prior allowing the media.

1.1.3. SDP Capability Negotiation

SDP capability negotiation is described in RFC 5939 [11] and describes sending multiple potential SDP combinations as an offer, such that a user agent can offer a choice of media security alternatives in the body of an initial INVITE request. However, the caller UA has no prior knowledge of whether media plane security setup will succeed and in many cases it will fail or cause a lengthy delay while the user agent re-attempts, for example using a different IP access network.

1.1.4. Motivations for RFC 3329

RFC3329 describes why security is needed to protect SIP signalling from man-in-the-middle attacks, and to accomodate the expected wide variation in security mechanism support by SIP entities. The media plane requires similar protection and exchange of security capabilities, for example to prevent eavesdropping in environments such as public wireless access networks that have no inherent security. For the media plane security mechanism defined by this document, the cryptographic key is in plain text in SDP, therefore signalling SHOULD be protected e.g. using the security mechanism negotiation described by RFC 3329 [4]

1.2. Design Goals

Security on the media plane differs from security for signalling, because it can be applied per media stream and also because multiple media streams can be started and stopped within a single SIP session.

For a single media stream, any one of the media plane security mechanisms supported by client and server may be applied, or no media plane security may be applied at all. Therefore, this specification defines secure capability exchange and use of security mechanisms for media, but with no obligation to use the indicated security mechanisms.

1. The entities involved in the security agreement process need to find out exactly which security mechanisms to apply, preferably without excessive additional roundtrips.

2. The selection of security mechanisms itself needs to be secure. Traditionally, all security protocols use a secure form of negotiation. For instance, after establishing mutual keys through Diffie-Hellman, IKE sends hashes of the previously sent data including the offered crypto mechanisms [9]. This allows the peers to detect if the initial, unprotected offers were tampered with.

3. The security agreement process should not introduce any additional state to be maintained by the involved entities.

2. Solution

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 signalling plane and the "mediasec" option tag used to indicate or require support of the mechanism described in this document. Any one of the mechanisms labelled with the "mediasec" header field parameter can be applied on-the-fly as a media stream is started, unlike mechanisms for signalling one of which is chosen and then applied throughout a session.

2.1. Header Fields Defined in RFC3329

As stated earlier, defines security mechanism agreement for signalling, including the "sec-agree" option tag that can appear in Supported:, Require:, and Proxy-Require: header fields. The "mediasec" header field parameter and the "mediasec" option tag defined in this document extend the procedures in RFC 3329 [4] to 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. Media plane security can be supported independently of any signalling plane security defined in RFC 3329 [4], but in order to protect any cryptographic key carried in SDP signalling plane security as defined

in RFC 3329 [4] SHOULD be used. A user agent or proxy that implements RFC 3329 [4] but does not implement this document and receives the Require; and Proxy-Require; header fields containing only the "mediasec" option tag will return a 420 (Bad extension) response, thereby informing the entity that sent them that this document is not supported. This document requires the first reliable response to include the media plane security capabilities, and therefore adds the 2xx response to the SIP responses that can contain the Security-Client, Security-Server, and Security-Verfiy header fields. RFC 3329 [4] allows only the Security-Server header field in SIP responses 421 (Extension Required) and 494 (Security Agreement Required).

2.2. Overview of Operation

The message flow is identical to the flow in RFC 3329 [4], but it is not mandatory for the user agent to apply media plane security immediately after it receives the list of supported media plane mechanisms from the server, or any timer after that, nor will the lack of a mutually supported media plane security mechanism prevent SIP session setup. In the message flow below, only Step 3 differs from RFC 3329 [4].

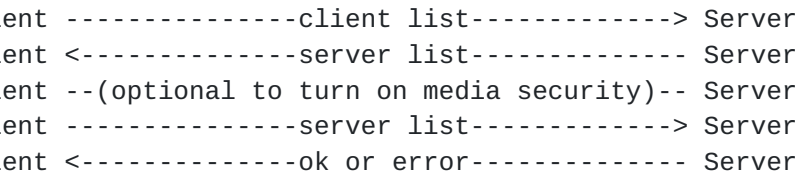


Figure 2: Security capability exchange message flow

Step 1: Clients wishing to use this specification can send a list of their supported security mechanisms along with the first request to the server.

Step 2: Servers wishing to use this specification can challenge the client to perform the security agreement procedure. The security mechanisms and parameters supported by the server are sent along in this challenge.

Step 3: The client may then proceed to select any media security mechanism they have in common and to turn on the selected security.

Step 4: The client contacts the server again, now using the selected security mechanism. The server's list of supported security

mechanisms is returned as a response to the challenge.

Step 5: The server verifies its own list of security mechanisms in order to ensure that the original list has not been modified.

2.3. Syntax

This document does not define any new SIP header fields, it reuses Security-Client, Security-Server and Security-Verify defined in RFC 3329 [4]. However, this document defines the mechanism-name "sdes-srtp". The description of Mechanism-name from RFC 3329 [4] is repeated below.

Mechanism-name

This token identifies the security mechanism supported by the client, when it appears in a Security-Client header field; or by the server, when it appears in a Security-Server or in a Security-Verify header field. The mechanism-name tokens are registered with the IANA. This specification defines one value:

o "sdes-srtp" for using SDES with SRTP [8].

2.4. Protocol Operation

2.4.1. The "mediasec" Header Field Parameter

The "mediasec" header field parameter may be used in the Security-Client, Security-Server, or Security-Verfiy header fields defined in RFC 3329 [4] 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-Verfiy header fields labelled with the "mediasec" header field parameter are specfic to the media plane and specific to the secure media transport protocol used on the media plane. This document defines the following value:

o sdes-srtp: SDES security mechanism for SRTP applied end to access edge

2.4.2. 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 client MUST add both Require and Proxy-Require header fields with the value "mediasec" to its request.

The contents of the Security-Client header field may be used by the server to include any necessary information in its response. However, for the purpose of the media plane security mechanism used in this document no such information is necessary.

A server receiving an unprotected request that contains a Require or Proxy-Require header field with the value "mediasec" MUST add a Security-Server header field to this response listing the security mechanisms that the server supports to its first reliable response to the client. Because this document is an extension of RFC 3329 [4], this 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.

All the subsequent SIP requests sent by the client to that server MAY make use of the security mechanism initiated 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. These requests MUST also have both a Require and Proxy-Require header fields with the value "mediasec".

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 RFC 3261 [7], 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. If the list was not modified, and the server is a proxy, it MUST remove the

"mediasec" value from both the Require and Proxy-Require header fields, and then remove the header fields if no values remain.

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.

2.4.3. Server Initiated

A server decides to use the security agreement described in this document based on local policy. If a server receives a request from the network interface that is configured to use this mechanism, it must check that the request has only one Via entry. If there are several Via entries the server is not the first-hop SIP entity and it MUST NOT use this mechanism. For such a request, the server must return a 502 (Bad Gateway) response.

A server that decides to use this agreement mechanism MUST challenge unprotected requests with one Via entry regardless of the presence or the absence of any Require, Proxy-Require or Supported header fields in incoming requests.

A server that by policy requires the use of this specification and receives a request that does not have the mediasec option tag in a Require, Proxy-Require or Supported header field MUST return a 421 (Extension Required) response. If the request had the "mediasec" option tag in a Supported header field, it MUST return a 494 (Security Agreement Required) response. In both situations the server MUST also include in the response a Security-Server header field listing its media-plane security capabilities and a Require header field with an option-tag "mediasec" in it.

Clients that support the extension defined in this document SHOULD add a Supported header field with a value of "mediasec".

2.5. 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. For the mechanism defined in this document, the UA sends 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 RFC 4568 [8], together with the attribute "a=3ge2ae", to the next hop proxy.

2.6. Duration of Security Assocations

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 signalling relationship with its first-hop proxy or until new keys are exchanged in SDP. In many cases, the SDP used to set up media plane security will be protected by a security association used to protect SIP signalling. If SIP signalling is protected by a security association, then the media plane security mechanism can be used until the signalling plane security association expires.

2.7. 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 summarized in Table 1.

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Security-Client	R	ard	-	o	-	o	o	o
Security-Server	2xx, 421, 494	ard	-	o	-	o	o	o
Security-Verify	R	ard	-	o	-	o	o	o
			SUB	NOT	PRK	IFO	UPD	MSG
Security-Client	R	ard	o	o	-	o	o	o
Security-Server	2xx, 421, 494	ard	o	o	-	o	o	o
Security-Verify	R	ard	o	o	-	o	o	o

Table 1: Summary of Header Field Usage

The "where" column describes the request and response types in which the header field may be used. The header may not appear in other types of SIP messages. Values in the where column are:

R: Header field may appear in requests.

2xx, 421, 494: A numerical value indicates response codes with which the header field can be used.

a: A proxy can add or concatenate the header field if not present.

r: A proxy must be able to read the header field, and thus this header field cannot be encrypted.

d: A proxy can delete a header field value.

The next six columns relate to the presence of a header field in a method:

o: The header field is optional.

3. Backwards Compatibility

Security mechanisms that apply to the media plane only MUST NOT have the same name as any signalling plane mechanism. If a signalling 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 signalling plane.

4. Examples

The following examples illustrate the use of the mechanism defined above.

4.1. Client Initiated

Typically, media plane security capabilities will be exchanged in parallel with security negotiation. However, it is also possible that media plane security capabilities are exchanged independently.

4.1.1. In Parallel with Security Negotiation

As per RFC 3329 [4], a UA negotiates the security mechanism for signalling to be used with its outbound proxy without knowing beforehand which mechanisms the proxy supports as shown in Figure 3 below.

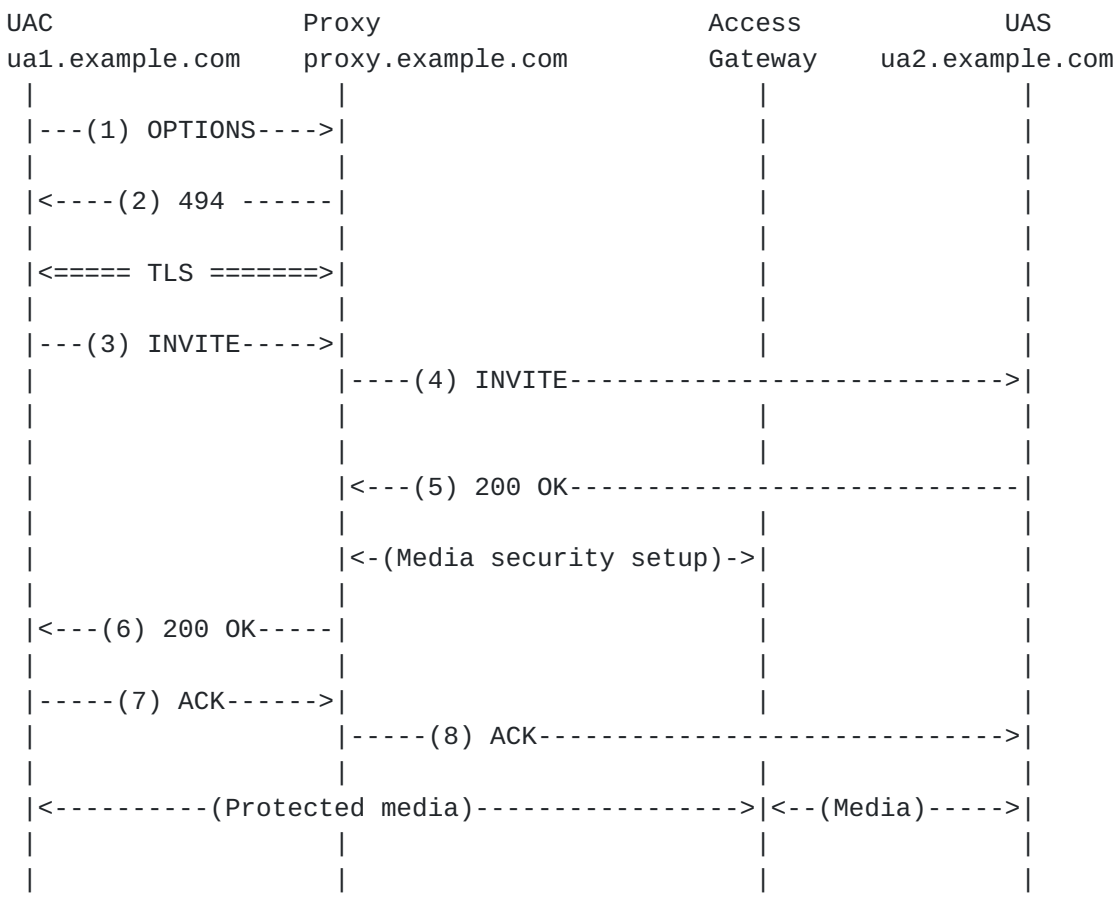


Figure 3: Negotiation Initiated by the Client

The UAC sends an OPTIONS request to its outbound proxy indicating security mechanisms for security negotiation and indicating at the same time that it is able to exchange capability of security mechanisms for the media plane and that it supports SDES for SRTP to the next hop.

The outbound proxy responds to the UAC with its own list of security mechanisms, also including SDES for the media plane. Indication of media security mechanisms is identified by the "mediasec" header field parameter. Media security mechanisms are returned by the client to the server in the Security-Verify: header field in the same way as for signalling security mechanisms.

When the connection is successfully established, the UAC sends an INVITE request including an SDP description of the media plane security to be used (a="e2ae" and a crypto attribute). This INVITE

contains the server's security lists for both media and signalling planes 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) OPTIONS sip:proxy.example.com SIP/2.0
   Security-Client: tls
   Security-Client: sdes-srtp;mediasec
   Require: sec-agree, mediasec
   Proxy-Require: sec-agree, mediasec

(2) SIP/2.0 494 Security Agreeement Required
   Security-Server: ipsec-ike;q=0.1
   Security-Server: tls;q=0.2
   Security-Server: sdes-srtp;mediasec

(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, mediasec
   Proxy-Require: sec-agree, mediasec

   Via: SIP/2.0/TCP proxy.example.com:5060;branch=z9hG4bK74bf9
   Max-Forwards: 70
   From: Alice <sip:alice@ua1.example.com>;tag=9fxced76sl
   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_SRTP
   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 4: Use of mediasec parameter

4.1.2. Independent of Security Negotiation

Typically, media plane security capabilities will be exchanged in parallel with security negotiation. However, it is also possible that media plane security capabilities are exchanged independently.

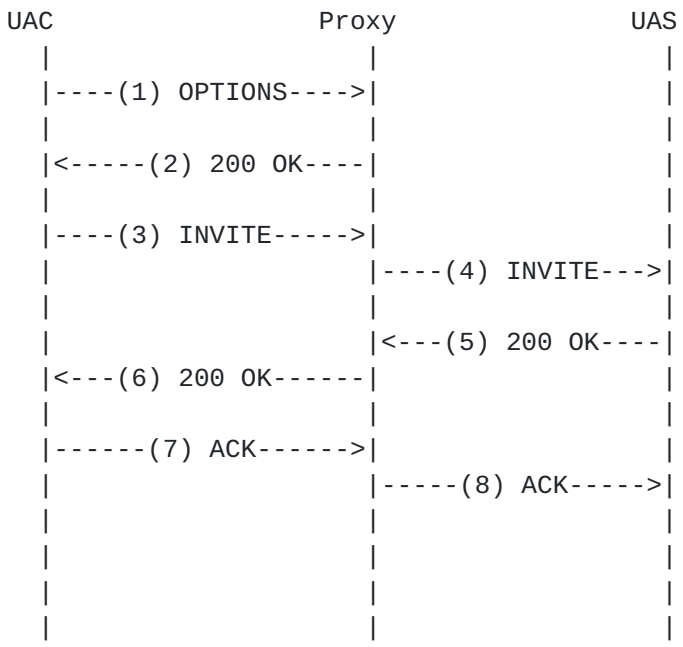


Figure 5: Negotiation Initiated by the Client

The UAC sends an OPTIONS request to its outbound proxy, indicating at the same time that it is able to exchange capability of security mechanisms for the media plane and that it supports SDES for SRTP to the next hop.

The outbound proxy responds to the UAC with its own list of security mechanisms, also including SDES for the media plane. Indication of media security mechanisms is identified by the "mediasec" header field parameter.

When the connection is successfully established, the UAC sends an INVITE request 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) OPTIONS sip:proxy.example.com SIP/2.0
   Security-Client: sdes-srtp;mediasec
   Require: mediasec
   Proxy-Require: mediasec

(2) SIP/2.0 200 OK
   Security-Server: sdes-srtp;mediasec

(3) INVITE sip:proxy.example.com SIP/2.0
   Security-Verify: sdes-srtp;mediasec
   Route: sip:callee@domain.com
   Require: mediasec
   Proxy-Require: mediasec

(4) INVITE sip:proxy.example.com SIP/2.0
     Route: sip:callee@domain.com

(5) SIP/2.0  200 OK

(6) SIP/2.0  200 OK
        Security-Server: sdes-srtp;mediasec
```

Figure 6: Use of mediasec parameter

4.2. Server Initiated

4.2.1. In Parallel with Security Negotiation

In the example in Figure 7 the client sends an INVITE towards the callee using an outbound proxy. This INVITE does not contain a Require header field.

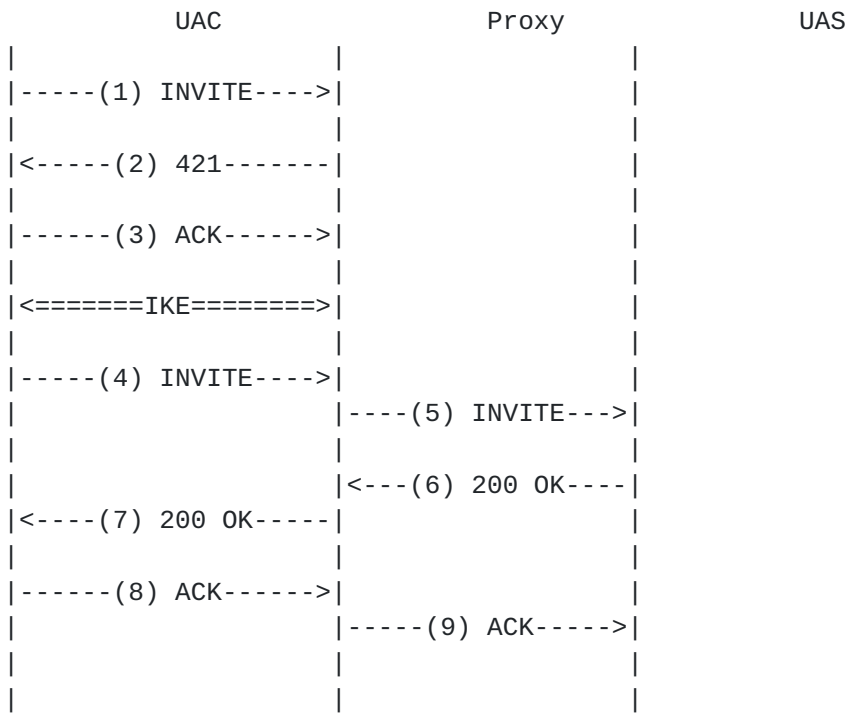


Figure 7: Server initiated media security

The proxy, following its local policy, does not accept the INVITE. It returns a 421 (Extension Required) with a Security-Server header field that lists SDES for SRTP for the media plane, as well as TLS and ipsec-ike for the signalling plane.

The server includes both sec-agree and mediasec option tags in a Require header field.

Since the UAC supports SDES for SRTP, the second INVITE (4) contains a Security-Verify header field that mirrors the Security-Server header field received in the 421. A description of the security to be used for the media plane is OPTIONAL in INVITE (4) and will be present if security is to be applied to the media in the session.

```
(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: sdes-srtp;mediasec
        Require: sec-agree, 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: sdes-srtp;mediasec
```

Figure 8: Server initiated media security

4.2.2. Independent of Security Negotiation

In the example in Figure 9 the client sends an INVITE towards the callee using an outbound proxy. This INVITE does not contain a Require header field.

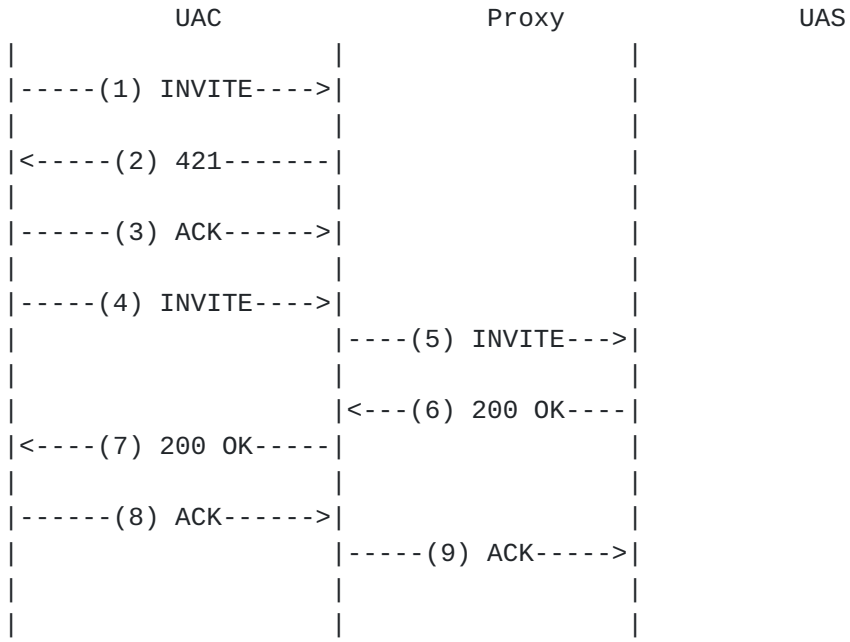


Figure 9: Server initiated media security

The proxy, following its local policy, does not accept the INVITE. It returns a 421 (Extension Required) with a Security-Server header

field that lists SDES for SRTP for the media plane, but no security mechanisms for the signalling plane.

Since the UAC supports SDES for SRTP, the second INVITE (4) contains a Security-Verify header field that mirrors the Security-Server header field received in the 421. A description of the security to be used for the media plane is OPTIONAL in INVITE (4) and will be present if security is to be applied to the media in the session.

```
(1) INVITE sip:uas.example.com SIP/2.0

(2) SIP/2.0 421 Extension Required
  Security-Server: sdes-srtp;mediasec

(4) INVITE sip:uas.example.com SIP/2.0
  Security-Verify: sdes-srtp;mediasec
```

Figure 10: Server initiated media security

5. 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"

6. 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.

7. IANA Considerations

The "mediasec" parameter and any new security mechanisms for the media plane must be IANA registered. This specification defines a new mechanism-name "sdes-srtp" in Section 2.3 which requires a central coordinating body. The body responsible for this coordination is the Internet Assigned Numbers Authority (IANA).

This document defines one mechanism-name to be initially registered, namely "sdes-srtp". Following the policies outlined in [10], further mechanism-names are allocated based on IETF Consensus.

Registrations with the IANA MUST include the mechanism-name token being registered, and a pointer to a published RFC describing the details of the corresponding security mechanism.

7.1. Registration Information

IANA registers new mechanism-names at http://www.iana.org/assignments/sip-parameters under "Security Mechanism Names". As this document specifies a mechanism-name, the initial IANA registration for mechanism-names will contain the information shown in Table 2. It also demonstrates the type of information maintained by the IANA.

Mechanism Name	Reference
sdes-srtp	this document

Table 2: Initial IANA registration

7.2. 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 2.3.)

Published Specification(s):

(Descriptions of new SIP Security Agreement mechanisms require a published RFC.)

7.3. Header Field Names

This specification registers no new header fields.

7.4. Response Codes

This specification registers no new response codes.

7.5. Option Tags

This specification defines a new option tag, namely mediasec. The option tag is defined by the following information, which has been included in the sub-registry for option tags under http://www.iana.org/assignments/sip-parameters.

Name: mediasec

Description:

This option tag indicates support for the Capability Exchange for Media Plane Security mechanism. When used in the Require, or Proxy-Require headers, it indicates that proxy servers are required to use the Capability Exchange for Media Plane Security mechanism. When used in the Supported header, it indicates that the User Agent Client supports the Capability Exchange for Media Plane Security mechanism. When used in the Require header in the 494 (Security Agreement Required) or 421 (Extension Required) responses, it indicates that the User Agent Client must use the Capability Exchange for Media Plane Security mechanism.

8. Security Considerations

This specification is an extension of RFC 3329 [4] 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 RFC 3329 [4] SHOULD be used in parallel with this specification.

9. References

9.1. Normative References

[1] authSurName, authInitials., "example1", year.

[2] authSurName, authInitials., "example2", year.

[3] 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>.

[4] Arkko, J., Torvinen, V., Camarillo, G., Niemi, A., and T. Haukka, "Security Mechanism Agreement for the Session Initiation Protocol (SIP)", RFC 3329, January 2003.

[5] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[6] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[7] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[8] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[9] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[10] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security

(DTLS)", RFC 5763, May 2010.

[11] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", RFC 5939, September 2010.

9.2. Informative References

[12] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[13] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[14] 3GPP, "IP Multimedia Subsystem (IMS) media plane security", 3GPP TR 33.828 9.1.0, June 2010.

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