SIPPING
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2008

J. Fischl
CounterPath Solutions, Inc.
H. Tschofenig

E. Rescorla
Network Resonance
July 9, 2007

Datagram Transport Layer Security (DTLS) Protocol for Protection of Media Traffic Established with the Session Initiation Protocol
draft-fischl-sipping-media-dtls-03.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies how to use the Session Initiation Protocol (SIP) to establish secure media sessions using or over the Datagram Transport Layer Security (DTLS) protocol. It describes a mechanism of transporting a fingerprint attribute in the Session Description

Protocol (SDP) that identifies the key that will be presented during the DTLS handshake. It relies on the SIP identity mechanism to ensure the integrity of the fingerprint attribute. This allows the establishment of media security along the media path.

Table of Contents

1. Introduction 3
2. Overview 4
3. Motivation 5
4. Terminology 6
5. Verifying Certificate Integrity 6
6. Miscellaneous Considerations 7
6.1. Anonymous Calls 8
6.2. Early Media 8
6.3. Forking 8
6.4. Delayed Offer Calls 9
6.5. Session Modification 9
6.6. UDP Payload De-multiplex 9
6.7. Rekeying 9
6.8. Conference Servers and Shared Encryptions Contexts 10
6.9. Media over SRTP 10
7. Example Message Flow 10
8. Security Considerations 15
8.1. UPDATE 16
8.2. SIPS 16
8.3. S/MIME 17
8.4. Single-sided Verification 17
8.5. Continuity of Authentication 17
8.6. Short Authentication String 18
8.7. Perfect Forward Secrecy 18
9. Requirements Analysis 18
9.1. Forking and retargeting (R1, R2, R3) 18
9.2. Clipping (R5) 19
9.3. Passive Attacks (R6) 19
9.4. Perfect Forward Secrecy (R7) 19
9.5. Algorithm Negotiation (R8, R9) 19
9.6. Endpoint Idenfification When Forking (R10) 19
9.7. 3rd Party Certificates (R11) 19
9.8. FIPS 140-2 (R12) 20
10. IANA Considerations 20
11. Acknowledgments 20
12. References 20
12.1. Normative References 20
12.2. Informational References 21
Authors' Addresses 23
Intellectual Property and Copyright Statements 25

1. Introduction

The Session Initiation Protocol (SIP) [RFC3261] and the Session Description Protocol (SDP) [I-D.ietf-mmusic-sdp-new] are used to set up multimedia sessions or calls. SDP is also used to set up TCP [I-D.ietf-mmusic-sdp-comedia] and additionally TCP/TLS connections for usage with media sessions [I-D.ietf-mmusic-comedia-tls]. The Real-Time Protocol (RTP) [RFC3550] is used to transmit real time media on top of UDP, TCP [I-D.ietf-avt-rtp-framing-contrans], and TLS [I-D.ietf-mmusic-comedia-tls]. Datagram TLS [RFC4347] was introduced to allow TLS functionality to be applied to datagram transport protocols, such as UDP and DCCP. This draft provides guidelines on how to use and to support for (a) transmission of media over DTLS and (b) to establish SRTP security using extensions to DTLS (see [I-D.ietf-avt-dtls-srtp]).

The goal of this work is to provide a key negotiation technique that allows encrypted communication between devices with no prior relationships. It also does not require the devices to trust every call signaling element that was involved in routing or session setup. This approach does not require any extra effort by end users and does not require deployment of certificates to all devices that are signed by a well-known certificate authority.

The media is transported over a mutually authenticated DTLS session where both sides have certificates. The certificate fingerprints are sent in SDP over SIP as part of the offer/answer exchange. The SIP Identity mechanism [I-D.ietf-sip-identity] is used to provide integrity for the fingerprints. It is very important to note that certificates are being used purely as a carrier for the public keys of the peers. This is required because DTLS does not have a mode for carrying bare keys, but it is purely an issue of formatting. The certificates can be self-signed and completely self-generated. All major TLS stacks have the capability to generate such certificates on demand. However, third party certificates MAY also be used for extra security.

This approach differs from previous attempts to secure media traffic where the authentication and key exchange protocol (e.g., MIKEY [RFC3830]) is piggybacked in the signaling message exchange. With this approach, establishing the protection of the media traffic between the endpoints is done by the media endpoints without involving the SIP/SDP communication. It allows RTP and SIP to be used in the usual manner when there is no encrypted media.

In SIP, typically the caller sends an offer and the callee may subsequently send one-way media back to the caller before a SIP answer is received by the caller. The approach in this

specification, where the media key negotiation is decoupled from the SIP signaling, allows the early media to be set up before the SIP answer is received while preserving the important security property of allowing the media sender to choose some of the keying material for the media. This also allows the media sessions to be changed, re-keyed, and otherwise modified after the initial SIP signaling without any additional SIP signaling.

Design decisions that influence the applicability of this specification are discussed in Section 3.

2. Overview

Endpoints wishing to set up an RTP media session do so by exchanging offers and answers in SDP messages over SIP. In a typical use case, two endpoints would negotiate to transmit audio data over RTP using the UDP protocol.

Figure 1 shows a typical message exchange in the SIP Trapezoid.

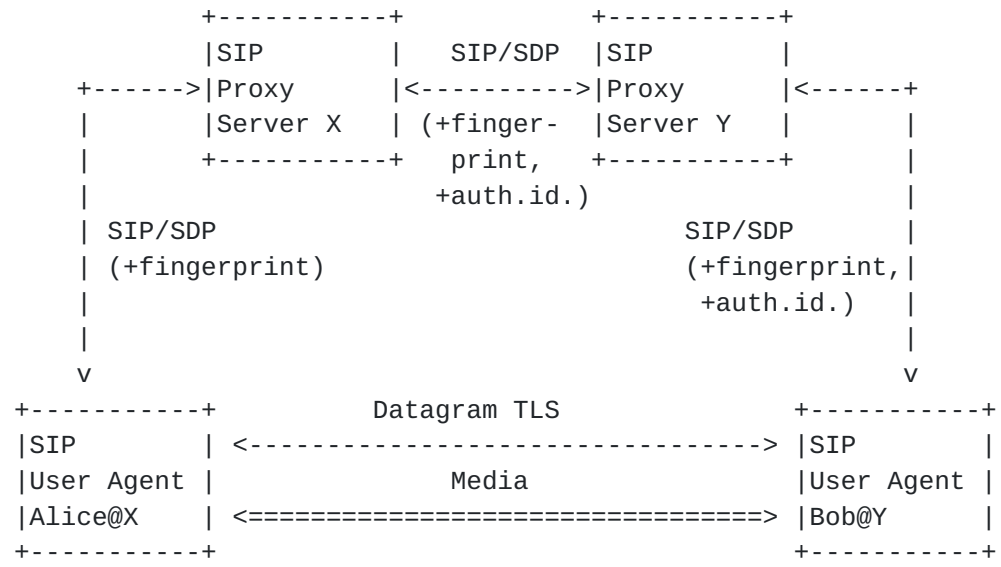


Figure 1: DTLS Usage in the SIP Trapezoid

Consider Alice wanting to set up an encrypted audio session with Bob. Both Bob and Alice could use public-key based authentication in order to establish a confidentiality protected channel using DTLS.

Since providing mutual authentication between two arbitrary end points on the Internet using public key based cryptography tends to be problematic, we consider more deployment friendly alternatives. This document uses one approach and several others are discussed in Section 8.

Alice sends an SDP offer to Bob over SIP. If Alice uses only self-signed certificates for the communication with Bob, a fingerprint is included in the SDP offer/answer exchange. This fingerprint is integrity protected using the identity mechanism defined in Enhancements for Authenticated Identity Management in SIP [I-D.ietf-sip-identity]. When Bob receives the offer, Bob establishes a mutually authenticated DTLS connection with Alice. At this point Bob can begin sending media to Alice. Once Bob accepts Alice's offer and sends an SDP answer to Alice, Alice can begin sending confidential media to Bob.

3. Motivation

Although there is already prior work in this area (e.g., Secure Descriptions for SDP [I-D.ietf-mmusic-sdescriptions], Key Management Extensions [I-D.ietf-mmusic-kmgmt-ext] combined with MIKEY [RFC3830] for authentication and key exchange), this specification is motivated as follows:

- TLS will be used to offer security for connection-oriented media. The design of TLS is well-known and implementations are widely available.
- This approach deals with forking and early media without requiring support for PRACK [RFC3262] while preserving the important security property of allowing the offerer to choose keying material for encrypting the media.
- The establishment of security protection for the media path is also provided along the media path and not over the signaling path. In many deployment scenarios, the signaling and media traffic travel along a different path through the network.
- This solution works even when the SIP proxies downstream of the identity service are not trusted. There is no need to reveal keys in the SIP signaling or in the SDP message exchange. In order for SDES and MIKEY to provide this security property, they require distribution of certificates to the endpoints that are signed by well known certificate authorities. SDES further requires that the endpoints employ S/MIME to encrypt the keying material.
- In this method, SSRC collisions do not result in any extra SIP signaling.

- Many SIP endpoints already implement TLS. The changes to existing SIP and RTP usage are minimal even when DTLS-SRTP [I-D.ietf-avt-dtls-srtp] is used.

4. Terminology

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 [RFC2119].

DTLS/TLS uses the term "session" to refer to a long-lived set of keying material that spans associations. In this document, consistent with SIP/SDP usage, we use it to refer to a multimedia session and use the term "TLS session" to refer to the TLS construct. We use the term "association" to refer to a particular DTLS ciphersuite and keying material set. For consistency with other SIP/SDP usage, we use the term "connection" when what's being referred to is a multimedia stream that is not specifically DTLS/TLS.

In this document, the term "Mutual DTLS" indicates that both the DTLS client and server present certificates even if one or both certificates are self-signed.

5. Verifying Certificate Integrity

The offer/answer model, defined in [RFC3264], is used by protocols like the Session Initiation Protocol (SIP) [RFC3261] to set up multimedia sessions. In addition to the usual contents of an SDP [I-D.ietf-mmusic-sdp-new] message, each 'm' line will also contain several attributes as specified in [I-D.fischl-mmusic-sdp-dtls], [RFC4145] and [I-D.ietf-mmusic-comedia-tls].

The endpoint MUST use the setup and connection attributes defined in [RFC4145]. A setup:active endpoint will act as a DTLS client and a setup:passive endpoint will act as a DTLS server. The connection attribute indicates whether or not to reuse an existing DTLS association.

The endpoint MUST use the certificate fingerprint attribute as specified in [I-D.ietf-mmusic-comedia-tls].

The setup:active endpoint establishes a DTLS association with the setup:passive endpoint [RFC4145]. Typically, the receiver of the SIP INVITE request containing an offer will take the setup:active role.

The certificate presented during the DTLS handshake MUST match the

fingerprint exchanged via the signaling path in the SDP. The security properties of this mechanism are described in Section 8.

If the fingerprint does not match the hashed certificate then the endpoint MUST tear down the media session immediately.

When an endpoint wishes to set up a secure media session with another endpoint it sends an offer in a SIP message to the other endpoint. This offer includes, as part of the SDP payload, the fingerprint of the certificate that the endpoint wants to use. The SIP message containing the offer is sent to the offerer's sip proxy over an integrity protected channel which will add an identity header according to the procedures outlined in [I-D.ietf-sip-identity]. When the far endpoint receives the SIP message it can verify the identity of the sender using the identity header. Since the identity header is a digital signature across several SIP headers, in addition to the bodies of the SIP message, the receiver can also be certain that the message has not been tampered with after the digital signature was applied and added to the SIP message.

The far endpoint (answerer) may now establish a mutually authenticated DTLS association to the offerer. After completing the DTLS handshake, information about the authenticated identities, including the certificates, are made available to the endpoint application. The answerer is then able to verify that the offerer's certificate used for authentication in the DTLS handshake can be associated to the certificate fingerprint contained in the offer in the SDP. At this point the answerer may indicate to the end user that the media is secured. The offerer may only tentatively accept the answerer's certificate since it may not yet have the answerer's certificate fingerprint.

When the answerer accepts the offer, it provides an answer back to the offerer containing the answerer's certificate fingerprint. At this point the offerer can definitively accept or reject the peer's certificate and the offerer can indicate to the end user that the media is secured.

Note that the entire authentication and key exchange for securing the media traffic is handled in the media path through DTLS. The signaling path is only used to verify the peers' certificate fingerprints.

6. Miscellaneous Considerations

6.1. Anonymous Calls

When making anonymous calls, a new self-signed certificate SHOULD be used for each call so that the calls can not be correlated as to being from the same caller. In situations where some degree of correlation is acceptable, the same certificate SHOULD be used for a number of calls in order to enable continuity of authentication Section 8.5.

Additionally, it MUST be ensured that the Privacy header [RFC3325] is used in conjunction with the SIP identity mechanism to ensure that the identity of the user is not asserted when enabling anonymous calls. Furthermore, the content of the subjectAltName attribute inside the certificate MUST NOT contain information that either allows correlation or identification of the user that wishes to place an anonymous call.

6.2. Early Media

If an offer is received by an endpoint that wishes to provide early media, it MUST take the setup:active role and can immediately establish a DTLS association with the other endpoint and begin sending media. The setup:passive endpoint may not yet have validated the fingerprint of the active endpoint's certificate. The security aspects of media handling in this situation are discussed in Section 8.

6.3. Forking

In SIP, it is possible for a request to fork to multiple endpoints. Each forked request can result in a different answer. Assuming that the requester provided an offer, each of the answerers' will provide a unique answer. Each answerer will create a DTLS association with the offerer. The offerer can then correlate the SDP answer received in the SIP message by comparing the fingerprint in the answer to the hashed certificate for each DTLS association.

Note that in the situation where a request forks to multiple endpoints that all share the same certificate, there is no way for the caller to correlate the DTLS associations with the SIP dialogs. Practically, this is not a problem, since the callees will terminate the unused associations. No new security problem is introduced here since endpoints which share the same certificate are assumed to represent the same user.

6.4. Delayed Offer Calls

An endpoint may send a SIP INVITE request with no offer in it. When this occurs, the receiver(s) of the INVITE will provide the offer in the response and the originator will provide the answer in the subsequent ACK request or in the PRACK request [RFC3262] if both endpoints support reliable provisional responses. In any event, the active endpoint still establishes the DTLS association with the passive endpoint as negotiated in the offer/answer exchange.

6.5. Session Modification

Once an answer is provided to the offerer, either endpoint MAY request a session modification which MAY include an updated offer. This session modification can be carried in either an INVITE or UPDATE request. In this case, it is RECOMMENDED that the offerer indicate a request to reuse the existing association (using the connection attribute) as described in Connection-Oriented Media [RFC4145]. Once the answer is received, the active endpoint will either reuse the existing association or establish a new one, tearing down the existing association as soon as the offer/answer exchange is completed. The exact association/connection reuse behavior is specified in RFC 4145 [RFC4145].

6.6. UDP Payload De-multiplex

Interactive Connectivity Establishment (ICE), as specified in [I-D.ietf-mmusic-ice], provides a methodology of allowing participants in multi-media sessions to verify mutual connectivity. In order to make ICE work with this specification the endpoints MUST be able to demultiplex STUN packets from DTLS packets. STUN [RFC3489] packets MUST NOT be sent over DTLS.

The first byte of a STUN message is 0 or 1 and it is reasonable to expect it to remain 0 or 1 for the near future. The first byte of a DTLS packet is "Type" which can currently have values of 20, 21, 22, and 23 as defined in ContentType declaration in [I-D.ietf-tls-rfc2246-bis]. It is reasonable to expect the first byte to remain under 64 and greater than 1. For RTP the first byte has a value that is 196 or above. A viable demultiplexing strategy would be to look at the first byte of the UDP payload and if the value is less than 2, assume STUN, if greater or equal to 196 assume RTP, otherwise assume DTLS.

6.7. Rekeying

As with TLS, DTLS endpoints can rekey at any time by redoing the DTLS handshake. While the rekey is under way, the endpoints continue to

use the previously established keying material for usage with DTLS. Once the new session keys are established the session can switch to using these and abandon the old keys. This ensures that latency is not introduced during the rekeying process.

Further considerations regarding rekeying in case the SRTP security context is established with DTLS can be found in Section 3.7 of [I-D.ietf-avt-dtls-srtp].

6.8. Conference Servers and Shared Encryptions Contexts

It has been proposed that conference servers might use the same encryption context for all of the participants in a conference. The advantage of this approach is that the conference server only needs to encrypt the output for all speakers instead of once per participant.

This shared encryption context approach is not possible under this specification. However, it is argued that the effort to encrypt each RTP packet is small compared to the other tasks performed by the conference server such as the codec processing.

Future extensions such as [I-D.mcgrew-srtp-ekt] could be used to provide this functionality in concert with the mechanisms described in this specification.

6.9. Media over SRTP

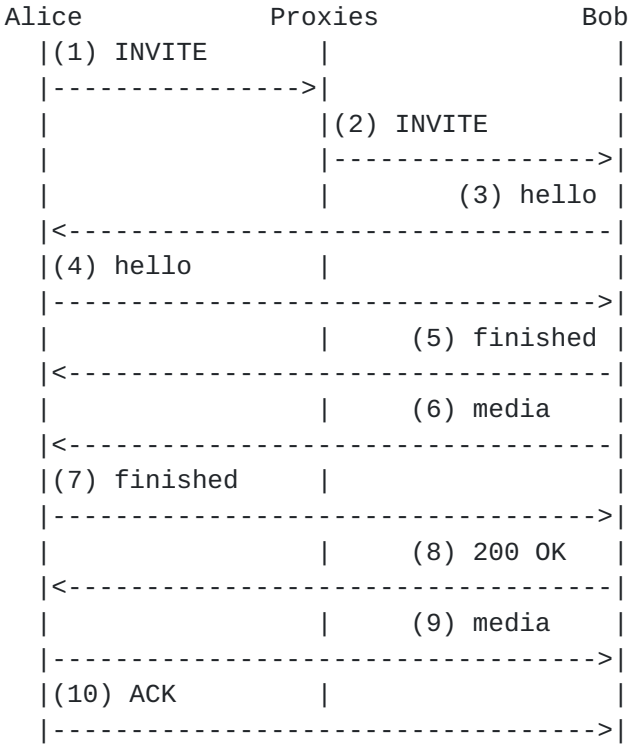
Because DTLS's data transfer protocol is generic, it is less highly optimized for use with RTP than is SRTP [RFC3711], which has been specifically tuned for that purpose. DTLS-SRTP [I-D.ietf-avt-dtls-srtp], has been defined to provide for the negotiation of SRTP transport using a DTLS connection, thus allowing the performance benefits of SRTP with the easy key management of DTLS. The ability to reuse existing SRTP software and hardware implementations may in some environments another important motivation for using DTLS-SRTP instead of RTP over DTLS. Implementations of this specification SHOULD support DTLS-SRTP [I-D.ietf-avt-dtls-srtp].

7. Example Message Flow

Prior to establishing the session, both Alice and Bob generate self-signed certificates which are used for a single session or, more likely, reused for multiple sessions. In this example, Alice calls Bob. In this example we assume that Alice and Bob share the same proxy.

The example shows the SIP message flows where Alice acts as the passive endpoint and Bob acts as the active endpoint meaning that as soon as Bob receives the INVITE from Alice, with DTLS specified in the 'm' line of the offer, Bob will begin to negotiate a DTLS association with Alice for both RTP and RTCP streams. Early media (RTP and RTCP) starts to flow from Bob to Alice as soon as Bob sends the DTLS finished message to Alice. Bi-directional media (RTP and RTCP) can flow after Bob sends the SIP 200 response and once Alice has sent the DTLS finished message.

The SIP signaling from Alice to her proxy is transported over TLS to ensure an integrity protected channel between Alice and her identity service. Note that all other signaling is transported over TCP in this example although it could be done over any supported transport.



Message (1): INVITE Alice -> Proxy

This shows the initial INVITE from Alice to Bob carried over the TLS transport protocol to ensure an integrity protected channel between Alice and her proxy which acts as Alice's identity service. Note that Alice has requested to be the passive endpoint which means that it will act as the DTLS server and Bob will initiate the session. Also note that there is a fingerprint

attribute on the 'c' line of the SDP. This is computed from Bob's self-signed certificate.

[[NOTE: This example is not completely correct because the exact syntax of the SDP is not yet determined. The MMUSIC working group is currently working on standardizing mechanisms for SDP capability negotiation which will enable this sort of best-effort encryption. When that work is finished, this draft will be harmonized with it.]]

```
INVITE sip:bob@example.com SIP/2.0
Via: SIP/2.0/TLS 192.168.1.101:5060;branch=z9hG4bK-0e53sadfkasldkfj
Max-Forwards: 70
Contact: <sip:alice@192.168.1.103:6937;transport=TLS>
To: <sip:bob@example.com>
From: "Alice"<sip:alice@example.com>;tag=843c7b0b
Call-ID: 6076913b1c39c212@REVMTEpG
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Content-Type: application/sdp
Content-Length: xxxx

v=0
o=- 1181923068 1181923196 IN IP4 192.168.1.103
s=example1
c=IN IP4 192.168.1.103
a=setup:passive
a=connection:new
a=fingerprint: \
  SHA-1 4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
t=0 0
m=audio 6056 UDP/TLS/RTP/AVP 0
a=sendrecv
```

Message (2): INVITE Proxy -> Bob

This shows the INVITE being relayed to Bob from Alice (and Bob's) proxy. Note that Alice's proxy has inserted an Identity and Identity-Info header. This example only shows one element for both proxies for the purposes of simplification. Bob verifies the identity provided with the INVITE.

```
INVITE sip:bob@example.com SIP/2.0
Via: SIP/2.0/TLS 192.168.1.101:5060;branch=z9hG4bK-0e53sadfkasldkfj
Via: SIP/2.0/TCP 192.168.1.100:5060;branch=z9hG4bK-0e53244234324234
Via: SIP/2.0/TCP 192.168.1.103:6937;branch=z9hG4bK-0e5b7d3edb2add32
Max-Forwards: 70
Contact: <sip:alice@192.168.1.103:6937;transport=TLS>
To: <sip:bob@example.com>
From: "Alice"<sip:alice@example.com>;tag=843c7b0b
Call-ID: 6076913b1c39c212@REVMTEpG
CSeq: 1 INVITE
Identity: CyI4+nAkHrH3ntmaxgr01TMxTmtjP7MASwliNRdupRI1vpkXRvZXx1ja9k
          3W+v1PDsy32MaqZi0M5WfEkXxbgTnPYW0jIoK8HMyY1VT7egt0kk4XrKFC
          HYWGCl0nB2sNsM9CG4hq+YJZTMaSROoMUBhikVIjnQ8ykeD6UXNOyfI=
Identity-Info: https://example.com/cert
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Content-Type: application/sdp
Content-Length: xxxx

v=0
o=- 1181923068 1181923196 IN IP4 192.168.1.103
s=example1
c=IN IP4 192.168.1.103
a=setup:passive
a=connection:new
a=fingerprint: \
  SHA-1 4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
t=0 0
m=audio 6056 UDP/TLS/RTP/AVP 0
a=sendrecv
```

Message (3): ClientHello Bob -> Alice

Assuming that Alice's identity is valid, Message 3 shows Bob sending a DTLS ClientHello directly to Alice for each 'm' line in the SDP. In this case two DTLS ClientHello messages are sent to Alice. Bob sends a DTLS ClientHello to 192.168.1.103:6056 for RTP and another to port 6057 for RTCP.

Message (4): ServerHello+Certificate Alice -> Bob

Alice sends back a ServerHello, Certificate, ServerHelloDone for both RTP and RTCP associations. Note that the same certificate is used for both the RTP and RTCP associations. If RTP/RTCP multiplexing [I-D.ietf-avt-rtp-and-rtcp-mux] were being used only a single association would be required.

Message (5): Certificate Bob -> Alice

Bob sends a Certificate, ClientKeyExchange, CertificateVerify, change_cipher_spec and Finished for both RTP and RTCP associations. Again note that Bob uses the same server certificate for both associations.

Message (6): Early Media Bob -> Alice

At this point, Bob can begin sending early media (RTP and RTCP) to Alice. Note that Alice can't yet trust the media since the fingerprint has not yet been received. This lack of trusted, secure media is indicated to Alice.

Message (7): Finished Alice -> Bob

After Message 5 is received by Bob, Alice sends change_cipher_spec and Finished.

Message (8): 200 OK Bob -> Alice

When Bob answers the call, Bob sends a 200 OK SIP message which contains the fingerprint for Bob's certificate. When Alice receives the message and validates the certificate presented in Message 5. The endpoint now shows Alice that the call as secured.

```
SIP/2.0 200 OK

To: <sip:bob@example.com>;tag=6418913922105372816
From: "Alice" <sip:alice@example.com>;tag=843c7b0b
Via: SIP/2.0/TCP 192.168.1.103:6937;branch=z9hG4bK-0e5b7d3edb2add32
Call-ID: 6076913b1c39c212@REVMTEpG
CSeq: 1 INVITE
Contact: <sip:192.168.1.104:5060;transport=TCP>
Content-Type: application/sdp
Content-Length: xxxx

v=0
o=- 6418913922105372816 2105372818 IN IP4 192.168.1.104
s=example2
c=IN IP4 192.168.1.104
a=setup:active
a=connection:new
a=fingerprint:\
  SHA-1 FF:FF:FF:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
t=0 0
m=audio 12000 UDP/TLS/RTP/AVP 0
a=rtpmap:0 PCMU/8000/1


Message (9):  RTP+RTCP Alice -> Bob


   At this point, Alice can also start sending RTP and RTCP to Bob

Message 10:  ACK Alice -> Bob


   Finally, Alice sends the SIP ACK to Bob.
```

8. Security Considerations

DTLS or TLS media signalled with SIP requires a way to ensure that the communicating peers' certificates are correct.

The standard TLS/DTLS strategy for authenticating the communicating parties is to give the server (and optionally the client) a PKIX [RFC3280] certificate. The client then verifies the certificate and checks that the name in the certificate matches the server's domain name. This works because there are a relatively small number of servers with well-defined names; a situation which does not usually occur in the VoIP context.

The design described in this document is intended to leverage the authenticity of the signaling channel (while not requiring confidentiality). As long as each side of the connection can verify the integrity of the SDP INVITE then the DTLS handshake cannot be hijacked via a man-in-the-middle attack. This integrity protection is easily provided by the caller to the callee (see Alice to Bob in Section 7) via the SIP Identity [I-D.ietf-sip-identity] mechanism. However, it is less straightforward for the responder.

Ideally Alice would want to know that Bob's SDP had not been tampered with and who it was from so that Alice's User Agent could indicate to Alice that there was a secure phone call to Bob. This is known as the SIP connected party problem and is still a topic of ongoing work in the SIP community. In the meantime, there are several approaches that can be used to mitigate this problem: Use UPDATE, Use SIPS, Use S/MIME, Single Sided Verification, or use human-read Short Authentication String (SAS) to validate the certificates. Each one is discussed here followed by the security implications of that approach.

8.1. UPDATE

[I-D.ietf-sip-connected-identity] defines an approach for a UA to supply its identity to its peer UA and for this identity to be signed by an authentication service. For example, using this approach, Bob sends an answer, then immediately follows up with an UPDATE that includes the fingerprint and uses the SIP Identity mechanism to assert that the message is from Bob@example.com. The downside of this approach is that it requires the extra round trip of the UPDATE. However, it is simple and secure even when not all of the proxies are trusted. In this example, Bob only needs to trust his proxy.

[[OPEN ISSUE: Note that there is a window of vulnerability during the early media phase of this operation before Alice receives the UPDATE (which immediately follows the SDP answer). During this window, Alice cannot be sure of Bob's identity. This risk might be mitigated by including a secret in the offer which must be used to establish the DTLS association, for instance via TLS PSK [RFC4279]. We are still studying this issue. Obviously, this is more attractive is SIPS is used.]]

8.2. SIPS

In this approach, the signaling is protected by TLS from hop to hop. As long as all proxies are trusted, this provides integrity for the fingerprint. It does not provide a strong assertion of who Alice is communicating with. However, as much as the target domain can be trusted to correctly populate the From header field value, Alice can

use that. The security issue with this approach is that if one of the Proxies wished to mount a man-in-the-middle attack, it could convince Alice that she was talking to Bob when really the media was flowing through a man in the middle media relay. However, this attack could not convince Bob that he was taking to Alice.

8.3. S/MIME

[RFC3261] defines a S/MIME security mechanism for SIP that could be used to sign that the fingerprint was from Bob. This would be secure. However, so far there have been no deployments of S/MIME for SIP.

8.4. Single-sided Verification

In this approach, no integrity is provided for the fingerprint from Bob to Alice. In this approach, an attacker that was on the signaling path could tamper with the fingerprint and insert themselves as a man-in-the-middle on the media. Alice would know that she had a secure call with someone but would not know if it was with Bob or a man-in-the-middle. Bob would know that an attack was happening. The fact that one side can detect this attack means that in most cases where Alice and Bob both wish the communications to be encrypted there is not a problem. Keep in mind that in any of the possible approaches Bob could always reveal the media that was received to anyone. We are making the assumption that Bob also wants secure communications. In this do nothing case, Bob knows the media has not been tampered with or intercepted by a third party and that it is from Alice@example.com. Alice knows that she is talking to someone and that whoever that is has probably checked that the media is not being intercepted or tampered with. This approach is certainly less than ideal but very usable for many situations.

[TODO]

8.5. Continuity of Authentication

One desirable property of a secure media system is to provide continuity of authentication: being able to ensure cryptographically that you are talking to the same person as before. With DTLS, continuity of authentication is achieved by having each side use the same public key/self-signed certificate for each connection (at least with a given peer entity). It then becomes possible to cache the credential (or its hash) and verify that it is unchanged. Thus, once a single secure connection has been established, an implementation can establish a future secure channel even in the face of future insecure signalling.

In order to enable continuity of authentication, implementations

SHOULD attempt to keep a constant long-term key. Verifying implementations SHOULD maintain a cache of the key used for each peer identity and alert the user if that key changes.

8.6. Short Authentication String

An alternative available to Alice and Bob is to use human speech to verify each others' identity and then to verify each others' fingerprints also using human speech. Assuming that it is difficult to impersonate another's speech and seamlessly modify the audio contents of a call, this approach is relatively safe. It would not be effective if other forms of communication were being used such as video or instant messaging. DTLS supports this mode of operation. The minimal secure fingerprint length is around 64 bits.

ZRTP [I-D.zimmermann-avt-zrtp] includes Short Authentication String mode in which a unique per-connection bitstring is generated as part of the cryptographic handshake. The SAS can be as short as 25 bits and so is somewhat easier to read. DTLS does not natively support this mode, however it would be straightforward to add one as a TLS extension [RFC3546].

8.7. Perfect Forward Secrecy

One concern about the use of a long-term key is that compromise of that key may lead to compromise of past communications. In order to prevent this attack, DTLS supports modes with Perfect Forward Secrecy using Diffie-Hellman and Elliptic-Curve Diffie-Hellman cipher suites. When these modes are in use, the system is secure against such attacks. Note that compromise of a long-term key may still lead to future active attacks. If this is a concern, a backup authentication channel such as manual fingerprint establishment or a short authentication string should be used.

9. Requirements Analysis

[I-D.wing-media-security-requirements] describes security requirements for media keying. This section evaluates this proposal with respect to each requirement.

9.1. Forking and retargeting (R1, R2, R3)

In this draft, the SDP offer (in the INVITE) is simply an advertisement of the capability to do security. This advertisement does not depend on the identity of the communicating peer, so forking and retargeting work work when all the endpoints will do SRTP (R1). When a mix of SRTP and non-SRTP endpoints are present, we expect to

use the SDP capabilities mechanism currently being defined [I-D.ietf-mmusic-sdp-capability-negotiation] to transparently negotiate security where possible (R2). Because DTLS establishes a new key for each session, only the entity with which the call is finally established gets the media encryption keys (R3).

9.2. Clipping (R5)

Because the key establishment occurs in the media plane, media need not be clipped before the receipt of the SDP answer (R5).

9.3. Passive Attacks (R6)

The public key algorithms used by DTLS (RSA, Diffie-Hellman, and Elliptic Curve Diffie-Hellman) are secure against passive attacks (R6).

9.4. Perfect Forward Secrecy (R7)

DTLS supports Diffie-Hellman and Elliptic Curve Diffie-Hellman cipher suites which provide PFS (R7).

9.5. Algorithm Negotiation (R8, R9)

DTLS negotiates cipher suites before performing significant cryptographic computation and therefore supports algorithm negotiation (R8) and multiple cipher suites (R9) without additional computational expense.

9.6. Endpoint Idenfification When Forking (R10)

Once the SDP response is received, the implementation can match the fingerprint against the offered client Certificate message (R10). Note, however, that if the server is using ephemeral DH or ECDH, it still must compute a fresh DH share and sign it in the ServerKeyExchange. This could be optimized away by having a DTLS ClientHello extension in which the client provide a copy of its fingerprint in advance.

9.7. 3rd Party Certificates (R11)

Third party certificates are not required. However, if the parties share an authentication infrastructure that is compatible with TLS (3rd party certificates or shared keys) it can be used (R11).

9.8. FIPS 140-2 (R12)

TLS implementations already may be FIPS 140-2 approved and the algorithms used here are consistent with the approval of DTLS and DTLS-SRTP (R12).

10. IANA Considerations

This specification does not require any IANA actions.

11. Acknowledgments

Cullen Jennings contributed substantial text and comments to this document. This document benefited from discussions with Francois Audet, Nagendra Modadugu, and Dan Wing. Thanks also for useful comments by Flemming Andreasen, Rohan Mahy, David McGrew, and David Oran.

12. References

12.1. Normative References

[I-D.ietf-mmusic-comedia-tls]
Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", draft-ietf-mmusic-comedia-tls-06 (work in progress), March 2006.

[I-D.ietf-mmusic-sdp-new]
Handley, M., "SDP: Session Description Protocol", draft-ietf-mmusic-sdp-new-26 (work in progress), January 2006.

[I-D.ietf-sip-identity]
Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

[I-D.fischl-mmusic-sdp-dtls]
Fischl, J. and H. Tschofenig, "Session Description Protocol (SDP) Indicators for Datagram Transport Layer Security (DTLS)", draft-fischl-mmusic-sdp-dtls-02 (work in progress), March 2007.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261] 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.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[RFC3489] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

12.2. Informational References

[I-D.ietf-avt-dtls-srtp]
McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)", draft-ietf-avt-dtls-srtp-00 (work in progress), July 2007.

[I-D.ietf-avt-rtp-and-rtcp-mux]
Perkins, C. and M. Westerlund, "Multiplexing RTP Data and

Control Packets on a Single Port",
draft-ietf-avt-rtp-and-rtcp-mux-05 (work in progress),
May 2007.

[I-D.ietf-avt-rtp-framing-contrans]
Lazzaro, J., "Framing RTP and RTCP Packets over
Connection-Oriented Transport",
draft-ietf-avt-rtp-framing-contrans-06 (work in progress),
September 2005.

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment
(ICE): A Protocol for Network Address Translator (NAT)
Traversal for Offer/Answer Protocols",
draft-ietf-mmusic-ice-16 (work in progress), June 2007.

[I-D.ietf-mmusic-kmgmt-ext]
Arkko, J., "Key Management Extensions for Session
Description Protocol (SDP) and Real Time Streaming
Protocol (RTSP)", draft-ietf-mmusic-kmgmt-ext-15 (work in
progress), June 2005.

[I-D.ietf-mmusic-sdescriptions]
Andreasen, F., "Session Description Protocol Security
Descriptions for Media Streams",
draft-ietf-mmusic-sdescriptions-12 (work in progress),
September 2005.

[I-D.ietf-mmusic-sdp-capability-negotiation]
Andreasen, F., "SDP Capability Negotiation",
draft-ietf-mmusic-sdp-capability-negotiation-05 (work in
progress), March 2007.

[I-D.ietf-mmusic-sdp-comedia]
Yon, D., "Connection-Oriented Media Transport in the
Session Description Protocol (SDP)",
draft-ietf-mmusic-sdp-comedia-10 (work in progress),
November 2004.

[I-D.ietf-sip-connected-identity]
Elwell, J., "Connected Identity in the Session Initiation
Protocol (SIP)", draft-ietf-sip-connected-identity-05
(work in progress), February 2007.

[I-D.ietf-tls-rfc2246-bis]
Dierks, T. and E. Rescorla, "The TLS Protocol Version
1.1", draft-ietf-tls-rfc2246-bis-13 (work in progress),
June 2005.

[I-D.mcgrew-srtp-ekt]
McGrew, D., "Encrypted Key Transport for Secure RTP", draft-mcgrew-srtp-ekt-03 (work in progress), July 2007.

[I-D.wing-media-security-requirements]
Wing, D., "Requirements for a Media Security Key Management Protocol", draft-wing-media-security-requirements-04 (work in progress), June 2007.

[I-D.zimmermann-avt-zrtp]
Zimmermann, P., "ZRTP: Media Path Key Agreement for Secure RTP", draft-zimmermann-avt-zrtp-03 (work in progress), March 2007.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[RFC3546] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 3546, June 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

Authors' Addresses

Jason Fischl
CounterPath Solutions, Inc.
Suite 300, One Bentall Centre, 505 Burrard Street
Vancouver, BC V7X 1M3
Canada

Phone: +1 604 320-3340
Email: jason@counterpath.com

Hannes Tschofenig

Email: Hannes.Tschofenig@gmx.net

Eric Rescorla
Network Resonance
2483 E. Bayshore #212
Palo Alto, CA 94303
USA

Email: ekr@networkresonance.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).