

Secure VoIP: call establishment and media protection

Johan Bilien Erik Eliasson Joachim Orrblad Jon-Olov Vatn
Royal Institute of Technology (KTH)
Stockholm, Sweden

Abstract

In this paper we study the possibility of establishing a secure VoIP telephone call using SIP. Different security services relevant for VoIP are presented and we argue that end-to-end authentication and encryption should be provided by default. For media protection we evaluate the possibility of using either SRTP or IPSec, and we examine several alternatives of how a secure VoIP call can be established. The solution we suggest is based on SRTP for media protection, S/MIME and MIKEY for end-to-end authentication and keying, and TLS for hop-by-hop protection of SIP messages.

We also present measurements of secure call establishment for MIKEY, SRTP and IPSec using our own SIP user agent (minisip). Our conclusion is that the call establishment delay will not be significantly affected by introducing these security protocols.

1 Introduction

When using SIP based IP telephony over the Internet today few users pay attention to whether their call is secure or not. However, in order for IP telephony to be widely adopted we strongly believe that security facilities must be provided. In particular we argue that end-to-end authentication between is not only possible, but that this initial authentication handshake should establish session keys, which can be used to protect the subsequent (voice) data stream.

To describe the VoIP security issues we anticipate we first consider a SIP call setup[1]. Fig. 1 shows the messages exchanged when a user (Alice@minisip.com) calls her friend (Bob@ssvl.kth.se). We have assumed that Alice sends the `INVITE` message via her SIP proxy, which in turn uses DNS to locate the SIP proxy for ssvl.kth.se.

We believe that a user (Alice) will associate the term *secure VoIP call* with properties such as:

1. *A call will only be established with the callee she expects.* Securing the SIP Registration messages will defeat some of the simple redirection attacks. To ensure Alice that it is really Bob's user agent she is communicating with, end-to-end authentication is needed.
2. *Charging is done correctly.* If charging is desired its correctness is vital, however, we assume that flat rate will be used for Internet calls (fixed

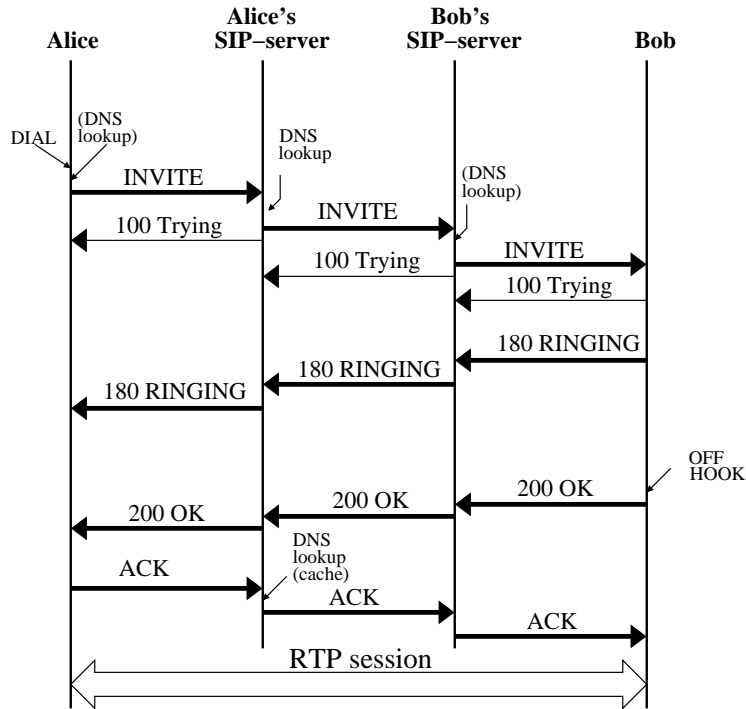


Figure 1: Establishing a VoIP session using SIP. (The ACK may alternatively be sent directly from Alice to Bob.)

monthly cost or free), thus we will not consider this requirement further in this paper.

3. *Unwanted calls will be efficiently blocked to avoid VoIP spamming.* If VoIP calls are flat rate (or even a very low rate) one can expect a similar situation for spam phone calls as is currently the case for email spam. However, an authentication handshake at call establishment makes it possible to reject a call automatically based on user preferences.
4. *The voice data will be protected against eavesdropping.* If Alice initiates a secure call she expects to be able to speak in private with the callee (Bob). The need for this service is probably greater for VoIP than for regular telephony (PSTN), because the possibility of eavesdropping of an IP call is greater, in particular since many (commodity) tools to do this are readily available. Fortunately, session keys to encrypt and integrity protect the (RTP) audio streams can be generated as a side-effect of the authentication handshake.
5. *Information about who Alice is calling (or who is calling Alice) should not be revealed by eavesdropping.* This security requires that one must encrypt the SIP call setup messages in Fig. 1, e.g., by using TLS transport. While an attacker may still be able to guess who Alice is calling, e.g., by inspecting the DNS traffic (if Bob has his own domain) or watching the RTP traffic

(if Bob has a fixed IP address), this will not directly indicate whom she has called.

6. *Alice's identity should not be revealed by eavesdropping.* This requirement is hard to meet and we do not believe that many users find this property crucial. Even if we are able to protect all SIP signaling from revealing her identity there may be many other ways for a persistent attacker to acquire this information, perhaps by observing other traffic that the user sends and receives.
7. *Alice's identity should be hidden from the callee.* A system should allow a caller to be anonymous. By introducing an initial authentication handshake we do not exclude the possibility for callers to be anonymous; however, the callee can reject such calls.

We address items 1 and 3 by enabling Alice and Bob to authenticate each other during call establishment. Authentication can be seen as part of a *keying protocol* used to establish a security association between Alice and Bob, i.e., to negotiate what cipher suite to use, create a (master) session key, etc. This security context can then be used by a security protocol, to encrypt and integrity protect user data, thereby addressing item 4. In section 2 we describe the use of IPSec/ESP[14] and the secure real-time transport protocol (SRTP[7]) to protect real-time audio data. In section 3 we examine alternative approaches of how keying protocols can be used with SIP to establish a VoIP call protected by SRTP or IPSec.

To address the security concerns raised in items 5 and 6 we recommend that SIP messages be secured 'hop-by-hop' using TLS tunnels between the respective SIP entities, at least on the path between a user agent and its SIP server. The use of TLS to protect the SIP signaling follows the specification of a secure SIP URI (SIPS[29]), although the use of a SIPS URI implies that **all** hops between SIP entities (except perhaps for the last hop between Bob's proxy and Bob) must be protected by TLS. It is worth noting that the use of the SIPS URI does **not** imply end-to-end authentication or encryption of voice *data* – it only specifies hop-by-hop protection of the SIP *signaling*. We believe that users will find this *neither* satisfactory *nor* intuitive.

2 Securing VoIP media - at what layer?

The major question when supporting end-to-end security is what layer should provide this; i.e., the network layer or some higher layer? For reliable data transfers the major alternatives are either IPSec (network layer) or TLS (transport/application layer). For real-time UDP traffic the main alternative is SRTP (i.e., transport/application layer). Although operating at different layers, SRTP and IPSec provide similar services:

- **Encapsulation format:** Fig. 2 illustrates both SRTP and IPSec/ESP. SRTP specifies the encapsulation format for the protected RTP packet, as well as what parts of the RTP packet are covered by the encryption and authentication algorithms respectively. Only *two* fields are added: the authentication tag (recommended) and the master key index (MKI) field (optional).

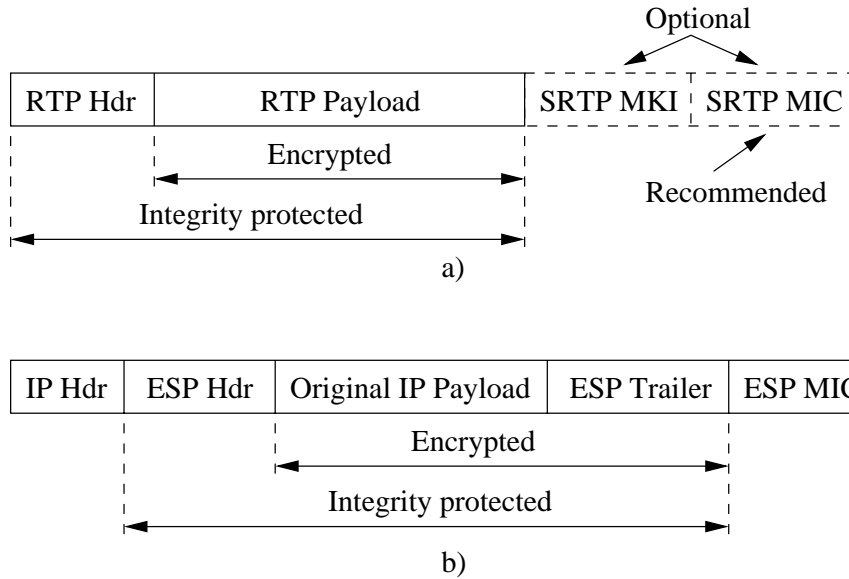


Figure 2: Packet formats for (a) SRTP and (b) IPSec/ESP (tunneling mode).

IPSec[15] defines two IP headers, the *authentication header* (AH[13]) and the *encapsulation security payload* (ESP[14]). Here we only consider IPSec with ESP in *transport mode*. As opposed to SRTP, use of IPSec/ESP will require additional encapsulation for NAT traversal[16]. However, IPSec aware NATs exist.

- **Cryptographic transforms:** SRTP defines the protocols to use for encryption (the default is the advanced encryption standard (AES) in counter mode (AES-CM[18]) with a 128 bit session key). AES-CM enables the receiver to process the packets in random order, a feature which is desirable for real-time applications where packets are **not** delivered reliably and thus may certainly be out of order. SRTP also defines protocols to use for packet authentication/integrity protection. The default is HMAC-SHA1[17, 22] using a 128 session key and a 32 bit authentication tag.

IPSec does not target any specific application, thus it does not use the same default encryption mechanism as SRTP. However, AES-CM and HMAC-SHA1 are available for ESP[11, 19], thus IPSec should be able to handle real-time voice traffic as effectively as SRTP.

- **Session key generation mechanism:** SRTP requires a *master* key to be provided, e.g., using a keying protocol such as MIKEY (see section 3.1.1). Based on this master key SRTP can derive the session keys for its security transforms (encryption key, authentication key, and salt).

Just as SRTP, ESP relies on a keying mechanism to provide the security association for the communication between Alice and Bob as well as performing user authentication, negotiation of ciphers and establishment of session keys. For this purpose IETF standardized the *Internet key exchange*

protocol (IKE)[10].

Use of SRTP is signalled as a parameter in the SDP message (RTP/SAVP) carried in the SIP `INVITE`. Using SRTP with the default cryptographic transforms does **not** affect the end-to-end delay or the end-node processing load significantly as compared to sending regular (non-protected) RTP traffic. Abad presents measurement results showing there is an additional roughly $80\ \mu\text{s}$ end-to-end delay, resulting in a data throughput of 20 Mbit/s when using SRTP on a 700 MHz Pentium III[8]. Although cryptographic performance is implementation dependent, we see no reason why performing the corresponding cryptographic operations with IPSec/ESP (in kernel space) should not perform as well as SRTP.

3 Secure call establishment

The previous section described how confidentiality and integrity could be added to the media stream using SRTP or IPSec/ESP provided that the required security association, keys, etc. have already been established. This requires a key agreement protocol, which in turn relies on the existence of a long-term secret between Alice and Bob in order to do mutual authentication. This long-term secret could be either manually configured into Alice and Bob or involve one or more *trusted third parties* for enhanced scalability.

When selecting a keying protocol to use for secure VoIP there are three main issues: (1) *which* keying protocol to use, (2) if it should be run *natively* or *carried* within the SIP signaling packets, and (3) if so, *how* SIP should carry these keying protocol messages. We have grouped the alternative approaches according to the second issue: in section 3.1 we examine solutions where keying messages are carried by SIP (integrated keying) and section 3.2 describes approaches where keying protocols are run natively (separate keying).

3.1 Integrated keying

When Alice calls Bob she generally does not know his current location beforehand. “Piggybacking” the keying messages in the SIP call establishment signaling messages is therefore a natural approach. SIP is a text-based protocol sharing several similarities with SMTP (and HTTP) and the SIP message contains a MIME content type line, specifying the kind of message(s) carried in the body part of the SIP message. We now have the choice of encoding the keying messages (or keying parameters) as SDP attributes, or to use a separate MIME type. We also have the choice between using S/MIME[27] (or other protocols designed for SMTP or HTTP) to protect the keying messages, or use some protocol designed more specifically for VoIP such as the *multimedia Internet keying* protocol (MIKEY[4]). We will examine MIKEY and how it can be used as a keying protocol for SRTP and IPSec/ESP in sections 3.1.1 and 3.1.2, while approaches relying on S/MIME are described in section 3.1.3.

3.1.1 Multimedia Internet Keying (MIKEY)

MIKEY[4] supports three different authentication mechanisms: *shared key*, *public key*, and *signed Diffie-Hellman* authentication. The negotiation of MIKEY parameters (including authentication mechanism) follows the SDP Offer/Answer model and is integrity protected by the authentication mechanism offered, i.e., a message integrity code (MIC) for the *shared key*, or a digital signature for the other two mechanisms.

Shared key authentication is probably going to be used in early deployments as people can manually exchange secret keys with their friends. To enable secure IP telephony on a larger scale a PKI should be introduced. A likely scenario is that user agents will store a small set of root CAs, just as web browsers do today. SIP providers will probably have certificates signed by one of these root CAs, and these providers will in turn issue user certificates to their customers. Of the certificate based authentication mechanisms we only consider *signed Diffie-Hellman*. The reasons for excluding the *public key* mechanism are that we expect it to have a longer delay (Alice will somehow have to retrieve Bob's public key before sending the `INVITE`), and we have not yet implemented the *public key* mechanism into our user agent (minisip). Furthermore, with signed Diffie-Hellman one has *perfect forward secrecy*¹.

When including certificates in the MIKEY messages the SIP messages become too large for UDP transport (SIP messages larger than 1300 bytes must be sent using congestion controlled transport[29]). For signed Diffie-Hellman we generally have to use TCP or TLS, however, using TLS would be the default choice for users who wish to avoid unnecessary exposure of who they are calling or receiving calls from.

MIKEY is an extensible keying protocol capable of exchanging parameters for various security protocols, however, SRTP is currently the only protocol which MIKEY has been standardized to support. Adding an "IPSec/ESP profile" to MIKEY would be a straightforward task as shown by Orrblad[24].

MIKEY messages can be *carried* in SIP messages as part of the SIP call establishment signaling, but as mentioned early in section 3 one needs to decide *how* this should be accomplished. There are two design issues related to this: *how* should MIKEY messages be *encoded/encapsulated*, and *which SIP messages* should be used to carry these encoded MIKEY messages. We start with the issue of how MIKEY messages should be encapsulated/encoded within a SIP packet. As Orrblad points out the approach will differ if MIKEY is used to setup a SRTP or IPSec connection.

Encapsulation of MIKEY messages Current work within IETF suggests the use of SDP key management extensions to carry MIKEY messages (as well as other keying mechanisms) within a new SDP attribute, the "key-mgmt" attribute[6]. This works fine when using MIKEY as keying protocol for SRTP, however, when MIKEY is used with IPSec/ESP Orrblad[24] argues that a SDP attribute is not the right location for a MIKEY message. To use MIKEY as keying protocol for IPSec/ESP Orrblad suggests that the MIKEY message is en-

¹With perfect forward secrecy an attacker (Trudy) would be unable to decipher a recorded conversation between Alice and Bob even if she at a later stage would gain access to their private keys.

coded as a multi-part MIME message in the SIP body. We will describe both these alternatives below:

- **MIKEY message as SDP attribute:** Within the MMUSIC IETF WG work is in progress to define a SDP key management attribute (`key-mgmt`) attribute enabling Alice to offer Bob one or more keying mechanisms during the connection establishment[6]. Although more than one keying protocol could be offered in parallel, most effort has focused on MIKEY. Fig. 3 shows how the “key management attribute” could be used to offer three keying protocols (MIKEY and two others) in SDP. Each attribute carries the data for the key management protocol being offered (encoded in base64).

The “key-mgmt” attribute works well when MIKEY is used to establish SRTP sessions, however, as described by Orrblad[24] the “key-mgmt” attribute as specified is tightly connected to the media transport and therefore unsuitable for IPsec/ESP. If SDP is to be used to carry MIKEY messages with IPsec/ESP parameters Orrblad suggests the introduction of yet another SDP attribute, which should be independent of the media transport, but otherwise similar to the “key-mgmt” attribute.

- **MIKEY message as MIME payload:** Instead of carrying the MIKEY message in an SDP attribute, Orrblad suggests that the MIKEY message is carried as multi-part MIME body as the preferred solution when establishing IPsec/ESP connections. To have MIKEY messages carried as MIME payload a corresponding MIME type has to be registered. The feasibility of this approach has demonstrated by implementing support for this in *minisip*[24].

3.1.2 Mapping MIKEY messages to SIP messages

Whether we carry MIKEY messages as SDP attributes or a multi-part MIME body we need to decide which SIP messages should be used to carry the MIKEY messages. Alice will put the *MIKEY Init* in her `INVITE` message to Bob, but

```
v=0
o=alice 2891092738 2891092738 IN IP4 lost.example.com
s=Secret discussion
t=0 0
c=IN IP4 lost.example.com
a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAAAAAAAAsAyO...
a=key-mgmt:keyp1 727gkdOshsuiSDF9sdhsdKnD/dhsoSJokdo7eWD...
a=key-mgmt:keyp2 DfsnuiSDSh9sdh Kksd/dhsoddo7eOok727gWsJD...
m=audio 39000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 42000 RTP/SAVP 31
a=rtpmap:31 H261/90000
```

Figure 3: SDP key management extension example (from [6]).

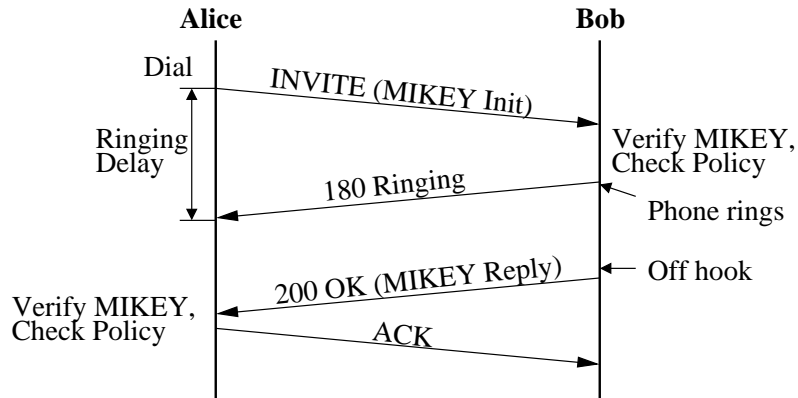


Figure 4: MIKEY messages in SIP INVITE and 200 OK

in which SIP message should Bob put the *MIKEY Reply*? Below we will describe three different alternatives, where the last two alternatives assume Alice and Bob have implemented support for reliable provisional acknowledgments (PRACKs[28]).

- **MIKEY Reply in 200 OK:** The most straightforward alternative is to put the *MIKEY Reply* in the 200 OK sent when Bob picks up his phone. This approach is shown in Fig. 4 and the SIP messages exchanged are the same as in the regular SIP call establishment in Fig. 1. The reason the 200 OK (and not the 180 Ringing) is used to carry the *MIKEY Reply* is that the 200 OK is sent reliably as opposed to the 180 Ringing provisional response. This mechanism is currently used by *minisip* and can be used if either (or both) Alice or Bob does not support PRACKs.

The advantage of this solution is its simplicity. Note that Bob is able to filter VoIP spam calls, since he authenticates the callee and checks the call against his policy configuration *before* his phone rings. The drawbacks are (1) that some MIKEY and SRTP/IPSec setup processing takes place after Bob has picked up his phone, which can lead to *clipping* effects at the beginning of the call², and (2) that Bob may suffer from *ghost ringing* if Alice for some reason rejects his *MIKEY Reply*, and thus the call. Although we expect the latter case to occur only rarely, it may happen if Alice's policy check does not accept the credentials which Bob presents³. (If Alice wants to reject a call before it has been established we suggest that she does this by sending a CANCEL message to Bob.)

More details on *clipping effects* and *Ringing delays* (the time from Alice has dialed Bob's "number" until she hears a local ring tone) is provided in section 4.

²Some of this MIKEY and SRTP/IPSec processing, which *minisip* currently performs after Bob picks up his phone, would be possible to do while his phone is ringing, thereby reducing the clipping effects.

³Examples of situations when Alice can reject Bob's credentials include: his certificate has expired, she is unable to find a valid trust path for his certificate, or there is a mismatch in MIKEY time-stamps.

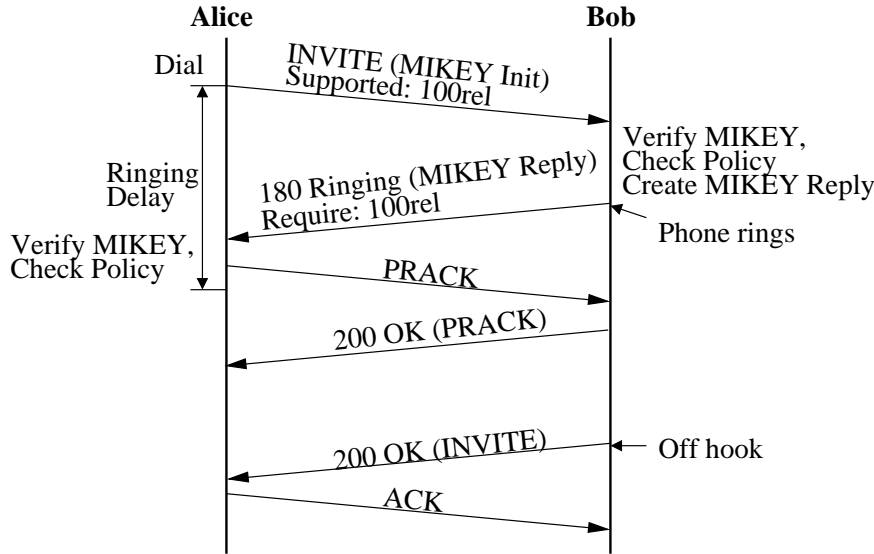


Figure 5: MIKEY messages in SIP INVITE and 180 Ringing

- **MIKEY Reply in a SIP Provisional Response:** To avoid clipping effects due to cryptographic processing at the start of the media session and to eliminate the risk of *ghost ringing* we suggest that SIP phones implement support for reliable provisional acknowledgments (PRACKs[28]). We present two approaches (note only the second eliminates the risk of *ghost ringing*):

- **MIKEY Reply in 180 Ringing:** If Alice announces PRACK support in her INVITE Bob could send the *MIKEY Reply* in the 180 Ringing message, see Fig. 5. Alice would be able to send her PRACK immediately; since as opposed to the next example she has no reason to wait for the outcome of the MIKEY verify and policy processing. As compared to the prior case Alice will experience a somewhat longer *Ringing delay*, since Bob has to construct his *MIKEY Reply* before sending the 180 Ringing message. This should not constitute any problem since we believe Alice is less sensitive to delays before she gets a local ringing tone as opposed to clipping effects at the beginning of the call.

As in the previous case we suggest that Alice sends a CANCEL message to Bob if she rejects the *MIKEY Reply* (this would happen after she sends the PRACK). As Bob's phone starts to ring at the time he sends the *MIKEY Reply* this solution still suffers from *ghost ringing*.

- **MIKEY Reply in 183 Session in Progress:** To eliminate the risk for *ghost ringing* we suggest that the *MIKEY Reply* is sent in a 183 Session in Progress message and that Bob's phone rings first when the corresponding PRACK arrives, see Fig. 6. Although this will increase the *Ringing delay* even more, we believe this is the most appropriate way to send the *MIKEY Reply*.

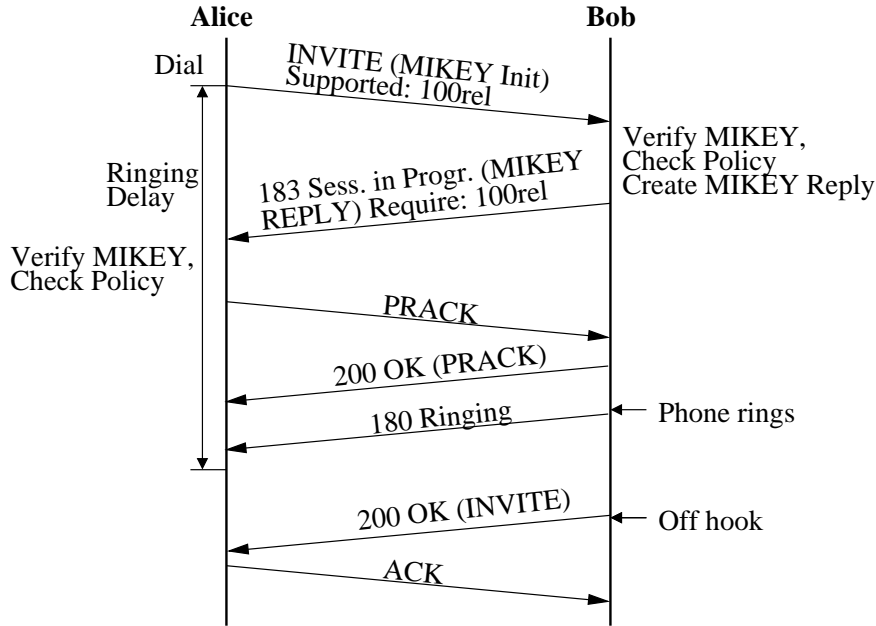


Figure 6: MIKEY messages in SIP INVITE and 183 Session in progress

In order for this approach to work Alice has to wait until she has accepted Bob's reply before she sends the PRACK. (If she rejects the message she should as usual send a CANCEL).

3.1.3 Using S/MIME to protect keying:

In the previous sections we examined the use of MIKEY as a keying protocol, as well as various methods of how MIKEY messages should be integrated with the SIP call establish signaling. In this section we will briefly present two approaches where S/MIME is used to protect the keying messages. The second approach suggests using "unauthenticated" MIKEY as keying protocol by relying on S/MIME for protection.

Using S/MIME with the SDP "crypto" attribute Within the MMUSIC IETF WG work is in progress to define a SDP "crypto" attribute to hold security parameters for SRTP[2], relying on some external security protocol such as S/MIME or TLS to protect these parameters. The main drawback we see with this approach compared to MIKEY is that it lacks support for *perfect forward secrecy* as can be achieved with MIKEY with signed Diffie-Hellman.

Using S/MIME with MIKEY Bilien has suggested the use of S/MIME and MIKEY⁴, where MIKEY/Diffie-Hellman is used for keying and security parameter exchange (SRTP or IPSec), and S/MIME (instead of MIKEY) is used "externally" to protect the message.

⁴Discussion on the IETF MSEC WG mailing list, March 2005 (subject "MIKEY D-H + SIP layer signature").

With this approach we achieve *perfect forward secrecy* for the protected media session **and** we have a uniform way to achieve end-to-end security of MIKEY as well as other parts of the SIP message. For example S/MIME could protect the SDP message and relevant fields of the SIP header[26] in addition to MIKEY. The SIP messages would be smaller, since a single certificate⁵ is used to protect all the desired parts of the SIP message. It would also avoid interoperability issues occurring when both S/MIME and MIKEY each carry certificates, there are questions as whether the certificate identifiers must match or not.

Unfortunately MIKEY currently only allows “external” protection of the MIKEY message when using pre-shared key authentication (“null” key). If “null authentication” becomes available for MIKEY/Diffie-Hellman the combination of S/MIME and MIKEY/Diffie-Hellman is our preferred choice of keying mechanism for VoIP.

3.2 Separate keying

An alternative to using SIP as a carrier of keying messages, Alice and Bob could use an existing keying protocol and run it natively between each other. We believe the main purpose for selecting this alternative would be if IPsec is preferred over SRTP and if, e.g., IKE or KINK[30] is preferred over MIKEY as a keying protocol for IPsec. However, even if Alice prefers to run IKE (natively) she still relies on SIP to find the location of Bob, since she only knows his SIP URI (Bob@ssvl.kth.se). Below we will describe three approaches for how IKE could be used together with SIP to establish a secure call:

- **Regular call establishment:** Alice starts by establishing an insecure call to Bob (as shown in Fig. 1), and runs IKE once the call is established. This is a simple approach, but has several drawbacks. For example, neither Alice nor Bob will know in advance if the other side supports IPsec, and the start of the voice session will either be unprotected or delayed while the IPsec tunnel is being established.
- **Key agreement using SIP OPTIONS message:** RFC 3329[5] specifies a mechanism where Alice could use a SIP OPTIONS message to agree on a what keying protocol to use with her SIP proxy. One could consider a similar approach, where negotiation of keying protocol was done between Alice and Bob instead of between Alice and her SIP proxy. In short this would mean that an initial SIP OPTIONS/200 OK exchange would take place between Alice and Bob via the SIP proxies (similar to Fig. 1). Alice would learn Bob’s address from the SIP Contact header and is thus able to setup an IPsec connection to Bob. Further SIP signaling (e.g. INVITE) as well as media streams can be sent directly between Alice and Bob.
- **Using SDP Security Preconditions:** RFC 3312[9] describes a generic framework for preconditions with SIP and is designed to address situations where Alice and Bob need to run a separate protocol end-to-end (e.g.

⁵The number of certificates actually carried in the SIP message would vary if a single or a chain of certificates is transmitted. It is also possible to send URLs to certificates instead of the certificate itself.

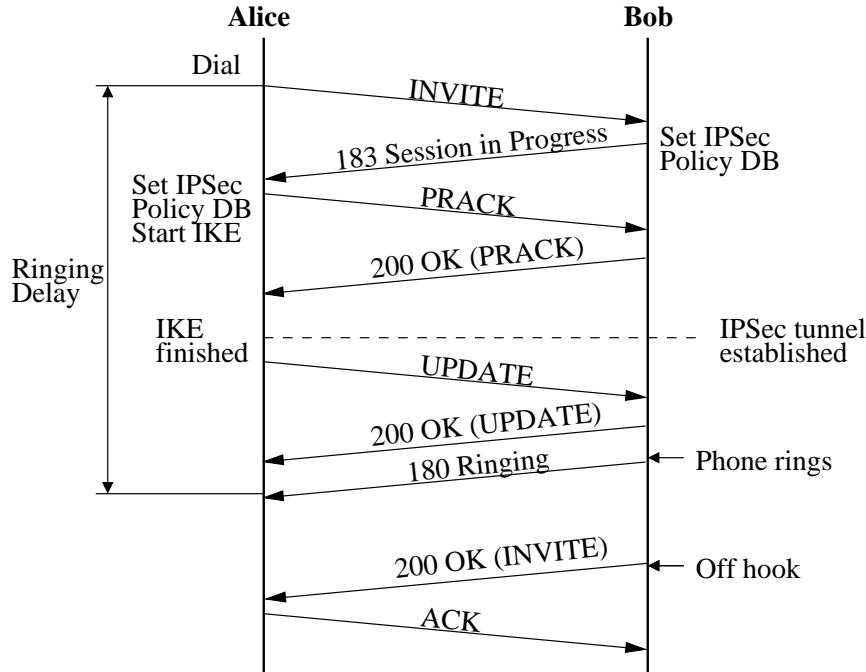


Figure 7: Using security preconditions with IKE/IPSec.

RSVP) before the session can be established. Within the IETF MMUSIC WG work is in progress to define *security preconditions* in-line with this framework[3]. However, this current proposal only describes how to signal the use of MIKEY and SRTP (which we prefer to do as shown in Fig. 6), and does not provide the primitives for Alice and Bob to specify the use of IKE. If the security preconditions framework would support establishment of IPsec connections, then we believe the call setup would be as shown in Fig. 7.

Alice sends an `INVITE` (via her SIP proxies) to Bob requiring the use of IPsec. Bob responds using a `183 Session in progress` message. Both sides will then modify their IPsec policy databases according to the media streams specified by SDP. Alice would start IKE and once the IPsec connection is up she will send a SIP `UPDATE` to resume the call establishment.

If IKE is to be used to establish secure VoIP calls we believe the last of these three alternatives has the greatest chance to gain acceptance.

4 Implementation and measurements

4.1 Testbed

Fig. 8 shows the testbed used in this study. It contains three interconnected networks of which two represent the two domains involved (`minisip.com` and

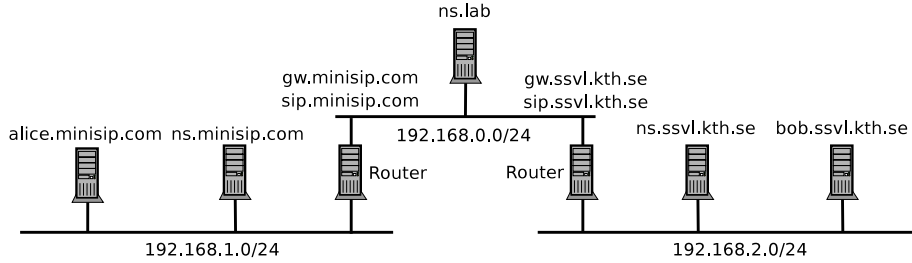


Figure 8: Testbed setup.

`ssvl.kth.se`), and the third represents the Internet. In each domain there is a name-server (BIND 9, 700 MHz Pentium 3 laptop). The root name-server `ns.lab` manages the delegation of `minisip.com` and `ssvl.kth.se` to their respective name server. The two routers (1.1 GHz Celeron desktops) perform static routing, and each router also runs a SIP server, SIP Express Router[12] (version 0.9 with Peter Griffith’s patch to support TLS).

Alice and Bob use an open source Linux SIP user agent, `minisip`[21], running on 500 MHz Pentium 3 laptops with a Linux 2.6 kernel. For cryptographic operations, `minisip` uses functions in `libcrypto`, from the `OpenSSL`[23] project.

The measurements were done by inserting hooks (saving time-stamps) in the `minisip` code. The additional processing in `minisip` adds an error of less than $5 \mu\text{s}$ per time-stamp.

4.2 Call setup in minisip

`Minisip` is able to set up VoIP calls with media protected by either SRTP or IPSec/ESP. MIKEY is used to exchange keys and security parameters in both cases. Pre-shared keys and Diffie-Hellman key exchange are supported. In the latter case, `minisip` pre-computes a couple of private and public Diffie-Hellman keys that can later be used when either receiving or making a call. The time it takes to do this is shown as `a0` and `b0` in Fig. 9. All measurements have been made using Diffie-Hellman key exchange.

When MIKEY is using Diffie-Hellman key exchange `minisip` sends the user’s personal certificate and the certificate of his provider in the MIKEY messages. The two certificates makes the SIP messages too large to be transported by UDP. Thus `minisip` transports these messages by TCP or TLS. Root certificates are stored in the terminal and any received certificate is verified against them.

- a10 and b21** Creation of the MIKEY initiation and response message. This includes their digital signature.
- a1** *Ringling delay*: The time from when Alice has dialed Bob’s number until she is notified that Bob’s user agent is ringing.
- a12, b10 and a20** Processing of incoming SIP message.
- a21 and b11** Verification of the MIKEY digital signature.

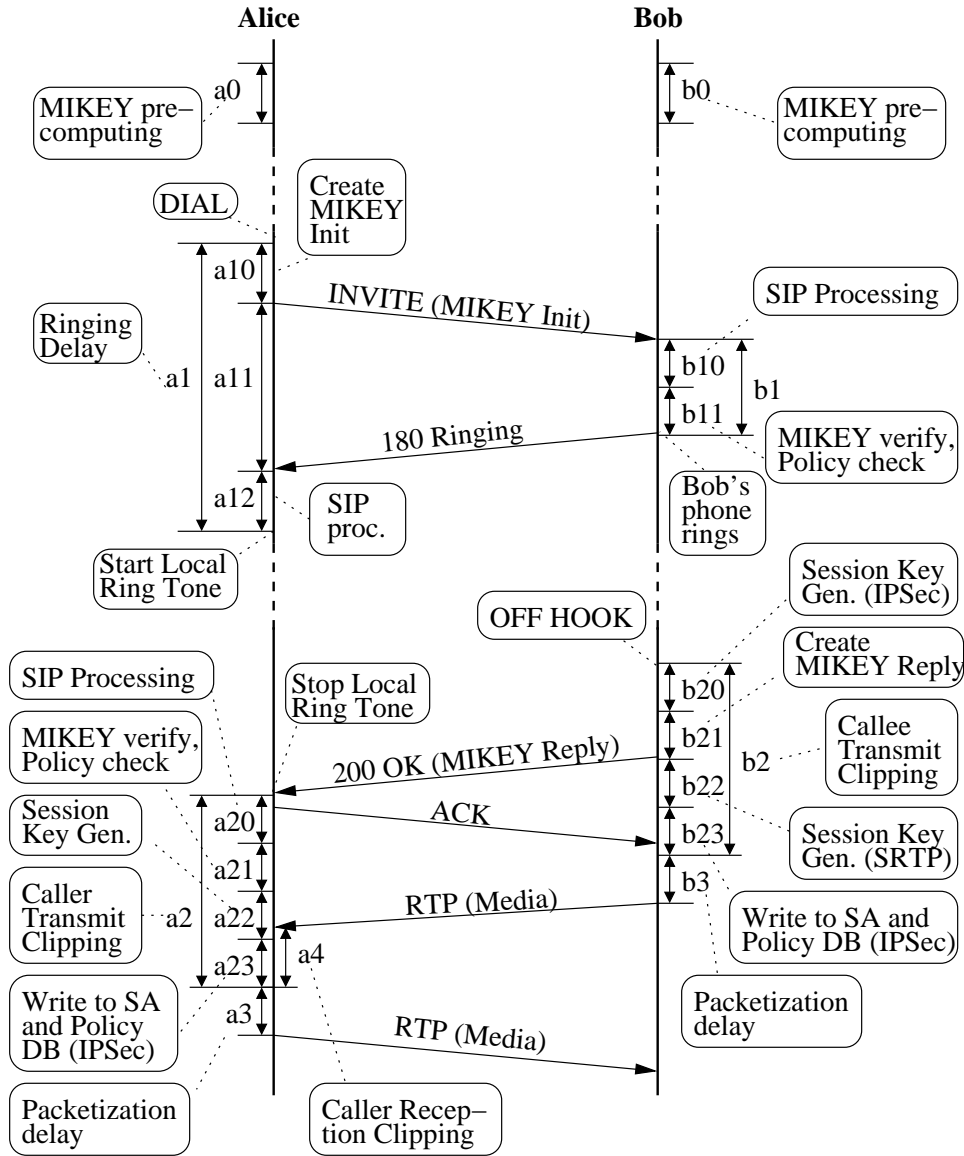


Figure 9: Secure call establishment in minisip.

- a22, b20 and b22** Session key generation from the Diffie-Hellman key exchange. In the case of SRTP, Bob does this after sending the 200 OK (b22) whereas for IPSec the current version of minisip makes Bob send this before (b20).
- a23 and b23** Writing the security association and IPSec policy to the Linux kernel (IPSec only).

- a3 and b3** *Packetization delay*: Waiting for and processing the first 20 ms audio data using a G.711 CODEC once the security association is complete. *a3* and *b3* may also include a “one-time” initiation delay of the sound device.
- b2** Time when Bob has picked up the phone, but is unable to transmit due to ongoing cryptographic processing. We refer to this as the *callee transmit clipping*.
- a2** Time when Alice’s user agent has received the 200 OK SIP response, but is unable to transmit due to ongoing cryptographic processing. We refer to this as the *caller transmit clipping*, as Alice may try to talk as soon as the local ring tone is stopped.
- a4** Time when Alice receives data from Bob, but is unable to process it due to ongoing cryptographic processing. We refer to this as the *caller reception clipping*.

4.3 Measurements

Both SRTP and IPSec/ESP have been measured with both TCP and TLS. For each of the four test cases ten calls were made between Alice and Bob. Tables 1, 2, and 3 summarize the measurements of the delays indicated in Fig. 9. The network delays in the testbed are insignificant, since the distance between the SIP entities are only one hop, i.e., in a “real situation” larger network delays will be added when any message is transmitted. It should also be noted that all calls were done when Alice’s and Bob’s user agents already had an existing TLS connection to their respective proxies and these connections were reused. If no TLS connection exists additional delay will be introduced and this is likely to happen when Bob’s proxy contacts Bob’s user agent[20] for the first time.

The *ringing delay* (“a1”) was of the order of 70-80 ms for all tested cases. This can be considered as insignificant to a human user, used to ringing delays higher than a second on traditional telephony networks. However, when the *MIKEY Reply* is carried in a provisional response as we recommend (see Fig. 6), then we can expect an increased *ringing delay*.

The time it takes to perform the cryptographic processing in TLS is not measured, but it results in an increased value of “a11” compared with using TCP. However the measurements show an interesting side effect of using TLS: the digital signatures of MIKEY messages and their verification take shorter time (“b10”, “a21” and “b21”) when TLS is used. An explanation could be that the TLS connection setup involved the same cryptographic operations, thus the libcrypto library has been initialized and part of the data (certificates and keys) has been cached.

The measured clipping delays at both the caller and the callee are much higher when using IPSec than when using SRTP. System calls setting the security association and the IPSec policy (shown as “a23” and “b23” in Fig. 9) are responsible for the main part of this delay. We do not know the exact reason for these relatively large delays.

The positive values for “a4” when using IPSec indicate that encrypted packets arrive at Alice *before* the computation of the keys needed to decrypt them has been finished. In a real network one could expect larger *caller reception clipping*, since the 200 OK is sent via the SIP proxies and therefore would be

delayed even more than the media packets from Bob.

Table 1: Ringing phase: Average delays [ms] and standard deviation.

		a1				b1		
		a10	a11	a12		b10	b11	
IPSec	TCP	40.2 (0.0)	31.3 (1.8)	2.1 (0.0)	73.6 (1.8)	6.8 (0.0)	5.1 (0.0)	18.6 (0.0)
	TLS	41.2 (3.3)	32.7 (0.2)	2.1 (0.0)	76.1 (3.4)	6.8 (0.0)	3.4 (0.0)	16.9 (0.1)
SRTP	TCP	39.6 (0.0)	30.6 (0.2)	2.1 (0.0)	72.2 (0.2)	7.2 (0.0)	5.1 (0.0)	19.0 (0.0)
	TLS	39.7 (0.2)	32.8 (0.2)	2.1 (0.0)	74.7 (0.3)	7.3 (0.0)	3.4 (0.0)	17.5 (0.1)

Table 2: Answering phase, caller side: Average delays [ms] and standard deviation.

		a2				a3	a4	
		a20	a21	a22	a23			
IPSec	TCP	5.9 (0.0)	4.8 (0.0)	132.0 (0.6)	659.8 (0.6)	804.4 (1.0)	30.1 (0.2)	83.3 (0.1)
	TLS	6.1 (0.0)	3.1 (0.0)	132.1 (0.2)	660.0 (0.8)	803.3 (0.8)	30.2 (2.6)	83.5 (0.3)
SRTP	TCP	6.6 (0.0)	4.8 (0.0)	134.2 (0.7)	N/A	146.2 (0.7)	28.1 (0.0)	-14.5 (0.6)
	TLS	6.9 (0.0)	3.2 (0.0)	134.9 (1.2)	N/A	145.5 (1.2)	28.2 (0.1)	-13.4 (1.3)

Table 3: Answering phase, callee side. Average delays [ms] and standard deviation.

		b2				b3	
		b20	b21	b22	b23		
IPSec	TCP	132.2 (0.3)	32.4 (0.1)	N/A	665.7 (0.2)	835.9 (0.3)	28.2 (0.0)
	TLS	132.3 (0.3)	28.7 (0.2)	N/A	667.0 (1.1)	834.8 (1.1)	28.2 (0.0)
SRTP	TCP	N/A	32.9 (0.1)	136.5 (0.4)	N/A	172.8 (0.4)	28.2 (0.0)
	TLS	N/A	29.2 (0.0)	136.5 (1.6)	N/A	170.3 (1.6)	28.1 (0.4)

5 Conclusions and future work

In this paper we have described design alternatives to establish a secure VoIP call. The solution we prefer would consist of the following components:

- **SRTP:** SRTP would be used to protect the media streams. Using SRTP it is easy to write portable implementations, which can be independent of the IPSec support provided by the end-host system.

- **MIKEY/Diffie-Hellman with S/MIME:** We suggest that MIKEY with Diffie-Hellman is used to for end-to-end keying and exchange of SRTP security parameters, but that S/MIME rather than MIKEY itself is used to protect the key exchange. With this approach the same certificate can be used to secure MIKEY as well as other parts of the SIP message. As this alternative is currently not allowed in MIKEY we recommend use of *regular* MIKEY with signed Diffie-Hellman as “fall-back”.
- **TLS:** To avoid exposure of Alice’s identity as well as who she calls and receives calls from, we suggest that TLS is used to secure the SIP messages hop-by-hop between the SIP entities.

To avoid cryptographic processing leading to *clipping effects* at call startup and to avoid *ghost ringing* we suggest that SIP user agents implement support for PRACKs and transmit the *MIKEY Reply* as in an 183 Session in Progress message as described in Fig. 6. We intend to add this support to *minisip* and evaluate its impact on the call setup performance.

In this paper we have shown some of the possibilities to protect media streams using IPSec as an alternative to SRTP. Our implementation is based on MIKEY as a keying protocol, where the MIKEY messages are carried as a multi-part MIME body in the SIP message rather than as an SDP attribute.

We have presented practical measurements on call setup performance in our open source SIP user agent *minisip*, using MIKEY for SRTP and ESP with both TCP and TLS transport. To simplify certificate verification in our “web-like” trust model, Alice and Bob each send their own as well as their CAs certificate along with the MIKEY messages. This implies that MIKEY with Diffie-Hellman is too large to send using UDP transport, thus TCP or TLS is used.

Setting up IPSec/ESP takes longer time than SRTP, thus in our setup use of IPSec leads to increased risks of clipping. As these results are implementation dependent we can **not** draw the conclusion that IPSec in general gives longer call setup delays than SRTP.

Currently we are implementing support for alternative PKI trust models in *minisip* (based on ideas presented by Perlman[25]) to evaluate their suitability for secure VoIP. We are also implementing support to manage caller and callee preferences into *minisip*, the main purpose being to block VoIP spam. In addition to these security related projects we and others are actively developing different parts of *minisip*. Join us at <http://www.minisip.org>!

References

- [1] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*, August 2001. RFC 3161.
- [2] Flemming Andreasen, Mark Baugher, and Dan Wing. Session Description Protocol Security Descriptions for Media Streams. IETF draft <draft-ietf-mmusic-sdescriptions-09.txt>, February 2005. Work in progress.

- [3] Flemming Andreasen and Dan Wing. Security Preconditions for Session Description Protocol Media Streams. IETF draft <draft-ietf-mmusic-securityprecondition-00.txt>, February 2005.
- [4] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. *MIKEY: Multimedia Internet KEYing*, August 2004. RFC 3830.
- [5] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, and T. Haukka. *Security Mechanism Agreement for the Session Initiation Protocol (SIP)*, January 2003. RFC 3329.
- [6] Jari Arkko, Elisabetta Carrara, Fredrik Lindholm, Mats Naslund, and Karl Norrman. Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP). IETF draft <draft-ietf-mmusic-kmgmt-ext-14.txt>, March 2005. Work in progress.
- [7] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. *The Secure Real-time Transport Protocol (SRTP)*, March 2004. RFC 3711.
- [8] Israel Abad Caballero. Secure Mobile VoIP. Master's thesis, Department of Microelectronics and Information Technology, Royal Institute of Technology, June 2003. Available online at ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/030626-Israel_Abad_Caballero-final-report.pdf.
- [9] G. Camarillo, Ed., W. Marshall, Ed., and J. Rosenberg. *Integration of Resource Management and Session Initiation Protocol (SIP)*, October 2002. RFC 3312.
- [10] D. Harkins and D. Carrel. *The Internet Key Exchange (IKE)*, November 1998. RFC 2409.
- [11] R. Housley. *Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)*, January 2004. RFC 3686.
- [12] SIP Express Router. <http://www.iptel.org/ser/> Last visited november 2003.
- [13] S. Kent and R. Atkinson. *IP Authentication Header*, November 1998. RFC 2402.
- [14] S. Kent and R. Atkinson. *IP Encapsulating Security Payload (ESP)*, November 1998. RFC 2406.
- [15] S. Kent and R. Atkinson. *Security Architecture for the Internet Protocol*, November 1998. RFC 2401.
- [16] T. Kivinen, B. Swander, A. Huttunen, and V. Volpe. *Negotiation of NAT-Traversal in the IKE*, January 2005. RFC 3947.
- [17] H. Krawczyk, M. Bellare, and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication*, February 1997. RFC 2104.

- [18] Helger Lipmaa, Phillippe Rogaway, and David Wagner. CTR-Mode Encryption. NIST. Available online at <http://csrc.nist.gov/encryption/modes/workshop1/papers/lipmaa-ctr.pdf>.
- [19] C. Madson and R. Glenn. *The Use of HMAC-MD5-96 within ESP and AH*, November 1998. RFC 2403.
- [20] R. Mahy. Connection Reuse in the Session Initiation Protocol (SIP). IETF draft <draft-ietf-sip-connect-reuse-03.txt>, October 2004. Work in progress.
- [21] Minisip Project. <http://www.minisip.org/>.
- [22] The Secure Hash Algorithm (SHA-1). National Institute of Standards and Technology, NIST FIPS PUB 180-1, "Secure Hash Standard," U.S. Department of Commerce, April 1995.
- [23] OpenSSL Project. <http://www.openssl.org/> Last visited november 2003.
- [24] Joachim Orrblad. Alternatives to MIKEY/SRTP to secure VoIP. Master's thesis, Royal Institute of Technology (KTH), Sweden, March 2005. Available online at <ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/050330-Joachim-Orrblad.pdf>.
- [25] Radia Perlman. An Overview of PKI Trust Models. *IEEE Network Magazine*, Nov/Dec 1999.
- [26] J. Peterson. *Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format*, September 2004. RFC 3893.
- [27] B. Ramsdell and Ed. *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification*, July 2004. RFC 3851.
- [28] J. Rosenberg and H. Schulzrinne. *Reliability of Provisional Responses in Session Initiation Protocol (SIP)*, June 2002. RFC 3262.
- [29] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. *SIP: Session Initiation Protocol*, June 2002. RFC 3261.
- [30] M. Thomas. *Requirements for Kerberized Internet Negotiation of Keys*, June 2001. RFC 3129.