Conversational IP Multimedia Security

R. Blom, E. Carrara, F. Lindholm, K. Norrman, and M. Näslund

Ericsson Research, Ericsson AB, Torshamsgatan 23, SE-16480 Stockholm, Sweden
{rolf.blom, elisabetta.carrara, fredrik.lindholm, karl.norrman, mats.naslund}@era.ericsson.se

Abstract **- With the introduction of 3G systems, multimedia applications over wireless will become widely available to the general public. One such application will be peer-to-peer Conversational Multimedia communication in which voice, video, still pictures and other media can be used simultaneously and in an interactive way. However, in an all IP environment it is important to have high quality and efficient security services to protect the traffic against eavesdropping and manipulations. In particular, end-to-end security is considered attractive. This paper investigates the security requirements that emerge from Conversational IP Multimedia applications in heterogeneous environments, with special emphasis on the requirements stemming from the wireless access. The design and the design goals of both SRTP, a security protocol for protection of media traffic, and MIKEY, a key management protocol specially developed for those environments, are also described.**

Keywords: conversational multimedia, security, 3G, heterogeneous network, SRTP, MIKEY.

1. Introduction

The third generation (3G) mobile telecommunication networks have great potential to bring increased value in wireless services to the mass market. One of the driving forces is the combination of mobile and Internet based communication offering real-time IP-based services, which utilize the advanced bearer capabilities of 3G radio systems such as higher bandwidth and Quality of Service.

One of the main benefits that 3G systems bring is the multimedia enhancements of existing mobile services and the creation of completely new services based on the new network capabilities. The technologies that enable efficient, cost-acceptable IP Multimedia services in 3G are under intensive development. In particular, it is crucial to design these services in such a way that they are suitable not only for the wired part of the network but also for the wireless links. The wireless links are generally the most constrained links in a heterogeneous environment, e.g. in bandwidth and delay.

Security is an essential aspect of the challenge of implementing end-to-end services on IP-based mobile transport networks. In particular, it is appealing to have end-to-end (e2e) protection for media traffic. Restricting the media access to the sender and intended recipients would guarantee confidentiality, protection of valuable content, company and government interest, etc. Such e2e security is especially important once the media is carried over the public Internet. A security framework is needed to protect IP Multimedia, and has to be designed taking into account the requirements posed by the emerging environments and applications.

This paper discusses a security solution to provide end-to-end protection for the Conversational Multimedia applications running over heterogeneous networks.

2. Scenarios

Some of the scenarios addressed by the security solution described in this paper are:

- simple Voice over IP (VoIP) calls, where a client initiates a VoIP session using the Session Initiation Protocol (SIP) [1] and the embedded Session Description Protocol (SDP) [2];
- server-to-client streaming applications, where a client uses the Real Time Streaming Protocol (RTSP) [3] to setup and control a streaming session;
- a small group communication, where three or more participants are involved in e.g., a SIP/VoIP session.

These applications typically use a control protocol to setup/control the media and the Real-time Transport Protocol (RTP) [4] as actual transport protocol for the media.

3. Design Goals and Network Impacts

There are several security protocols, which may be used to protect the data traffic. IPsec [5] is a well-known security protocol working at the network layer; TLS [6] is a transport-layer protocol. These protocols are not the best choice for all types of traffic. To accommodate the requirements from the heterogeneous environment and the real-time applications, security protocols adapted to the application and working at the application layer appear more suitable.

In particular, four factors must be taken into account in the mentioned scenarios: bandwidth, delay, computational power of mobile terminals, and transmission-error sensitivity.

Existing solutions in the IP world are usually not designed with heterogeneous networks in mind. Typically they show little

0-7803-7605-6/02/$17.00 ©2002 IEEE.

concern about bandwidth consumption and number of round-trips. Thus, since the radio spectrum in wireless networks is a limited and expensive resource, the use of standard security protocols would give high system costs, as these protocols require guaranteed high quality connections.

Use of standard IP transport for voice is very inefficient in terms of bandwidth, due to protocol headers introduced at each layer of the stack. A typical RTP voice payload in a 3G application is around 33 bytes, while the IP/UDP/RTP headers add 40 bytes in IPv4 (60 in IPv6). An efficient header compression mechanism such as ROHC [7] can compress the IPv4/UDP/RTP headers into 2 bytes (on average). For short voice packets over RTP, this yields major savings in bandwidth.

IP security protocols generally add their own headers. Some encryption algorithms also require message padding, and message authentication adds several bytes per-packet. Use of encryption may obstruct efficient header compression when the former is applied over the headers prior the header compression.

Delay represents a critical factor for real-time applications. Already the presence of the air link introduces delay. When security is enabled, extra processing time is needed, hence the efficiency of protocols and algorithms is essential. Moreover, the endpoints will typically be thin clients with limited computational resources. Thus the footprint (code size) and the computational cost of the protocols must be small. In general, public key operations are expensive, and they should be used as little as possible in favor of symmetric key operations.

The radio medium introduces an environment where bit errors are common and have to be handled. Certain types of applications, e.g. audio, have means to handle bit errors in a reasonable way. But if message authentication is applied, a single bit error will cause the packet to be dropped before reaching the application, which may degrade the perceived quality of the received data. However, if message authentication is not enabled, bit errors are still a concern in the decryption phase: it is important not to propagate errors during the processing on the receiver side. Error propagation and positions of erroneous bits are dependent on the choice of the cipher. For this reason, stream ciphers may be to prefer, as these do not propagate errors as block ciphers do.

4. Securing the Media Traffic

The Secure Real-time Transport Protocol (SRTP) [8] is a security protocol for RTP, under development in IETF.

SRTP is specifically designed for RTP, which is the typical transport protocol for carrying voice, audio and video. As such, RTP applications span from best-effort (where time is not a critical factor, e.g. download) to real-time applications (where time is very critical, e.g. streaming and conversational multimedia). A security protocol implies computational work, and especially real-time traffic, as motivated previously, is demanding. SRTP is specifically designed to provide high throughput and to fulfill the requirements (Section 3) arising from the heterogeneous networks.

SRTP is accompanied by SRTCP, a security protocol for the RTP control protocol (RTCP). SRTCP was designed using the same design criteria as for SRTP.

4.1. *Bandwidth Preservation*

Bandwidth preservation is a central concern in SRTP. It is common practice to use a packet sequence number to synchronize security algorithms. RTP has a sequence number in the header, monotonically incremented for each packet sent in a given stream. SRTP uses the RTP sequence number, and thus has no need to add an extra field for synchronization. However, the RTP sequence number is only 16 bits. If the sequence number recycles, the key in use may need to be changed (depending on the cipher). A new key may require a new run of the key management protocol (re-keying) which consumes resources and might therefore not be desirable. To allow for long sessions without the need of re-keying, SRTP maintains locally a 32-bit "rollover counter", to expand the space of the RTP sequence number, counting wraps of the shorter counter ("implicit sequencing").

SRTP in its basic configuration does not add any bits to the RTP packet, except the authentication tag when integrity protection is required. An optional, variable length field (the Master Key Identifier, MKI) may be added, to support re-keying. Re-keying without message expansion can be based on the sequence number. However, this method assumes simple scenarios and is less flexible than the use of the MKI.

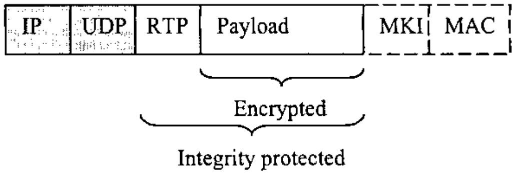

Fig 1: Overview of an SRTP packet.

4.2. *Security Functions*

It is important to be able to choose the appropriate security protection for each type of traffic. In SRTP, it is possible to independently enable encryption of the RTP payload and integrity protection of the RTP header and payload, together with replay protection (Fig.1). The RTP header is not encrypted to allow header compression, according to the bandwidth preservation design goal. SRTCP follows the same principles, however integrity protection is mandatory, as a control protocol can involve actions on the data traffic like termination of the connection.

4.3. *SRTP Framework*

SRTP is designed in accordance with good practice to allow for future extensions. In the past, it has for example happened that algorithms that were thought secure showed weaknesses and needed to be replaced, and that new better and more efficient algorithms were developed. Certain applications may also need new types of protection not yet supported by SRTP, e.g. data origin authentication for group communication, or new crypto algorithms. SRTP is designed as a framework, to allow such extensions.

The definition inside SRTP of predefined transforms provides cryptographic suites fulfilling the identified requirements for heterogeneous environments and real-time applications, and they improve interoperability. SRTP defines as encryption transforms Counter Mode [9], as default, and f8-mode [10]. They are both stream ciphers to avoid error propagation and both are based on the block cipher AES [11]. The predefined authentication transform is HMAC/SHA1 [12], a well-tested keyed hash based function. For bandwidth preservation, the authentication tag is by default truncated to 4 bytes, introducing a tradeoff with security, which however is believed to be acceptable for the kind of real-time traffic considered here.

SRTP is designed independently from the actual key management protocol to be used. The number of keys that SRTP needs depends on the enabled security functions: authentication and encryption keys for SRTP, authentication and encryption keys for SRTCP, SRTP and SRTCP salting keys (to defer certain off-line key-collision attacks, c.f. [13]). To limit the number of keys to be exchanged via key management, SRTP uses the concept of "master key" and "session key":

- a master key is exchanged via key management,
- session keys are securely derived from the exchanged master key. A session key is the actual key used to e.g. encrypt or authenticate.

An optional "key refresh" can be enabled: it allows to periodically derive new session keys (without further calls to the key management), improving security (limiting the amount of ciphertext for a certain key).

One master key can be shared between multiple SRTP streams, in order to reduce the number of keys exchanged and stored in a multimedia session. However, to allow such sharing, it is necessary to avoid re-use of the keystream ("two-time pad"), likely to disclose the plaintexts. SRTP can ensure that such reuse doesn't take place by making sure of the uniqueness of the Initialization Vector (IV) for the stream cipher. Otherwise different master keys have to be used. In a typical scenario, a sender will use one master key to secure its outgoing streams, and another sender will use a different master key.

Multicast scenarios exhibit very special properties. For example, it may be expensive to provide data origin authentication (proof of who is the actual source). An authentication mechanism like HMAC, based on a shared secret key, can only assure that the message comes from one of the group members. Public key operations (signatures) would be necessary and they are far too expensive for the real-time type of traffic, both in term of time and bandwidth. Efficient schemes to provide data origin authentication are under development, e.g. [14], however they are not yet standardized. For the time being, SRTP chooses not to offer data origin authentication support for groups. Another problem that multicast potentially shows is overload due to RTCP Receiver Reports sent back to the sender. Processing a high number of SRTCP packets might quite overload the SRTP sender.

5. Key Management

A security protocol needs a key management protocol to securely exchange keys and security parameters between the involved parties. Therefore, to complete the SRTP work, a key management protocol satisfying the same type of requirements is also under development, the Multimedia Internet KEYing (MIKEY) [15]. This section describes the design decisions, tradeoffs done and some of the most important basic cryptographic functions which MIKEY is currently based upon.

5.1. *Background to security functionality choices*

In existing peer-to-peer security protocols it is often possible to negotiate keys and other cryptographic parameters. Of course, the negotiation adds complexity to the setup phase. If the key management protocol also has to handle groups, negotiation with all the involved parties would add even more complexity. To limit the complexity, the key management protocol should give the possibility to push keys and security parameters to a set of other parties without any need for negotiation.

As described previously, the scenarios considered range from peer-to-peer applications to interactive groups. In the former case, a key negotiation method (mutual agreement on the key) is possible to use, while in the latter case a key transport method (pushing of the key) is preferred. This leads to the wish to provide both a key transport mechanism and a mechanism for key negotiation such as the Diffie-Hellman key exchange [16].

For the key transport method, a question that may arise is whether this should be based on public keys or on symmetric keys. Basing the key transport on symmetric keys gives a big advantage from efficiency point-of-view, while basing it on public keys gives a more scalable solution. An important aspect in 3G networks, is the existence of a symmetric key infrastructure. A streaming server in a 3G network, could benefit much from using a symmetric key transport, and re-use the symmetric key infrastructure in place. However, in a peer-to-peer conversation between end users, public keys might be preferred for scalability reasons and inter-domain communication.

The scenarios considered uses media-setup protocols such as SIP and RTSP. Integrating/tunneling the key management into those protocols reduces the number of roundtrips required in the total setup phase. As the media stream information is carried in SIP and RTSP as well, a tight coupling with the key management can be achieved by the integration. This may be used to setup security for more than one media stream at the same time (instead of setting up the security for each media stream separately). Also note that integrating the key management protocol within the SIP and RTSP implies at most one roundtrip to setup the security.

5.2. *Cryptographic protocol*

The basic cryptographic primitives must at least allow authentication of the corresponding peer, integrity of the message, replay protection, and confidentiality of the distributed keys. Authentication of a peer is often done in a challenge-response fashion. However, if the protocol should only use one roundtrip, a direct authentication method must be used, such as using a Message Authentication Code (MAC) or by applying a signature.

Two key transport methods and one key exchange method have been proposed to satisfy the different scenarios. The two key transport methods, one based on pre-shared symmetric keys and one based on public keys, use common functionality for the key transport. We will here concentrate on basics of the key transport methods and not discuss the key exchange.

In the pre-shared key method (Fig. 2), the initiator of the protocol creates a message consisting of a session identifier (ID_S), a timestamp (T), its own identity (ID_I), a random number (R), and one or more cryptographic keys that should be transported (K_S). The transported keys are encrypted using a pre-shared key for encryption (K_E) and the message is authenticated using a MAC with a pre-shared authentication key (K_A).

The responder of the protocol verifies the received message and then (if accepted) may return a verification message, included in the protocol to obtain explicit mutual authentication.

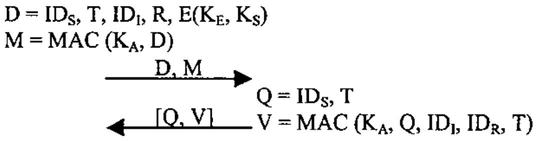
$D = ID_S, T, ID_I, R, E(K_E, K_S)$
$M = MAC(K_A, D)$
D, M
$Q = ID_S, T$
[Q, V]
$V = MAC(K_A, Q, ID_I, ID_R, T)$

Fig. 2. Key transport method based on pre-shared keys.

The same approach is used for the public key based method. However, instead of using a pre-shared key, a temporary key is encrypted with the public key of the corresponding party. This temporary key is then used to encrypt the transported keys (K_S), i.e., an envelope approach is used. In the first message, a signature is also applied instead of a MAC on the message. However, in the response message, no changes at all are made except that the pre-shared authentication key is replaced by the temporary key obtained from the initiating party, i.e. only symmetric cryptography is used, which then improves the overall performance.

Timestamps are used to provide replay protection. This will of course require the entities' clocks to be relatively well synchronized within a certain time window. Messages received within this time window need to be matched against and cached in a replay cache, while messages not within the window are simply dropped. Hence, the size of the time window will depend on the amount of data that can be stored in the replay cache. A method to dimension the time window and the replay cache can be based on an estimation of the maximum number of expected messages per second, and the storage size required for each message (or hash of the message).

A system may sometimes be the target of DoS attacks, such as flooding of messages to overflow the replay cache. Therefore, using a dynamic size of the time window could prevent replay cache overflows. The dynamic reduction of the time window can be triggered when the replay cache has reached a certain limit, whereby the time windows is decreased proportionally against the number of received messages compared to the expected maximum number of messages.

6. Conclusions

Security is important when deploying Multimedia applications. It is especially crucial to offer end-to-end security, in a way that only the final, intended recipients can access the content. When the networks are heterogeneous and the applications are real-time, several factors need to be taken into account in the design of a security solution. SRTP and MIKEY are two protocols under development in IETF that specifically address those types of scenarios. When creating a system for secure conversational multimedia (see Fig 3), SRTP and MIKEY can be used as building blocks to add security for the media traffic between two or more entities. However, a complete solution may need other security mechanisms as well, e.g., to protect the control signaling between different nodes.

The paper has shown some of the design choices and tradeoffs to be faced while designing components of a security solution for Conversation Multimedia applications in heterogeneous networks. The solution described here for secure conversational multimedia is characterized by:

- no message expansion (using basic configuration),
- no error propagation (use of stream ciphers),
- header compression possible (only payload encryption applied),
- one roundtrip key management that can be integrated with the media session setup,
- no security protocol parameter negotiation (instead distribution),
- key transport mechanism, which allows key distribution for groups,
- security setup for more than one SRTP stream at once.

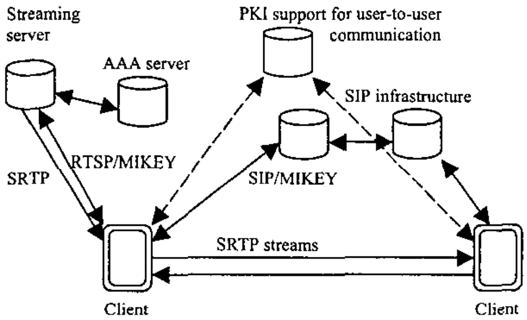

Fig. 3. Example of a possible conversational multimedia system including security functionality.

Acknowledgments

We would like to thank all the people involved in the design and standardization process of SRTP and MIKEY, in particular Jari Arkko and our co-authors of the SRTP draft, Mark Baugher, David McGrew, and David Oran.

References

[1] M. Handley et al. "SIP: Session Initiation Protocol", IETF, RFC 2326, March 1999.

[2] M. Handley and V. Jacobson, "SDP: Session Description Protocol", IETF, RFC 2327, April 1998.

[3] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)", IETF, RFC 2543, April 1998.

[4] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications", IETF, RFC 1889, January 1996.

[5] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol", IETF, RFC 2401, November 1998.

[6] T. Dierks and C. Allen, "The TLS Protocol Version 1.0", IETF, RFC 2246, January 1999.

[7] C. Bormann et al., "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", IETF, RFC 3095, July 2001.

[8] Baugher et al. "The Secure Real Time Transport Protocol (SRTP)", IETF Draft, Work in progress, June 2002.

[9] H. Lipmaa, P. Rogaway, and D. Wagner, "CTR-Mode Encryption", NIST, http://csrc.nist.gov/encryption/modes/workshop1/papers/lipmaa-ctr.pdf

[10] "Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specification (Release 4)", 3GPP TS 35.201 V4.1.0, December 2001.

[11] Advanced Encryption Standard (AES), Federal Information Processing Standard Publications (FIPS PUBS) 197, November 2001.

[12] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", IETF, RFC 2104, February 1997.

[13] D. McGrew and S. Fluhrer, "Attacks on Encryption of Redundant Plaintext and Implications on Internet Security", Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptography (SAC 2000), Springer-Verlag.

[14] A. Perrig, R. Canetti, D. Tygar, and D. Song, "Efficient and Secure Source Authentication for Multicast", Proc. of Network and Distributed System Security Symposium NDSS 2001, pp. 35-46, 2001.

[15] J. Arkko et al., "MIKEY: Multimedia Internet KEYing", IETF Draft, Work in progress, July 2002.

[16] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp. 644-654, 1976.