Innovations in Key Management Protocols for Secure Wireless Multimedia Communication

Mamoona Asghar¹, Mohammed Ghanbari¹, and Martin Fleury¹,*

¹School of Computer Sci. and Electronic Eng., University of Essex, Colchester CO4 3SQ, United Kingdom

Abstract: With the growth of multimedia services over wireless systems, the security of communication over cellular mobile phone networks, wireless access networks, and unreliable mobile ad-hoc networks is becoming a challenging issue. Secure communication implies data confidentiality and integrity, along with authentication, bringing with it commercial viability and protection of intellectual property. Innovatory key management protocols, specialized for multimedia exchange, are now under active development. Whereas encryption algorithms are, for reasons of open scrutiny and validation, not a subject of confidentiality, a shared secret, the session key, must be confidential and that session key relies on the existence of a secret permanent key. If a server can successfully distribute keys to remote clients without interception, the greater becomes the chance of secure communication, even though no side channel and no physical exchange of keys are required. Therefore, the focus of this review is to highlight: the key generation mechanisms, interaction with associated key management protocols; and the strength of key-management protocols against security attacks. In addition, the paper reviews recent patents on multimedia key management and also examines counter-measures present in key management protocols against known key and data attacks.

Keywords: Authentication, confidentiality, key derivation, key establishment, key management, message authentication, mobile networks, multimedia, security protocols

* Address correspondence to this author at tel. +44 1260 872684, fax.+44 1260 872900, fleury.martin55@gmail.com
1. INTRODUCTION

Multimedia services in mobile communication systems are proliferating due to the promise of communication [1] anywhere and at any time. Because transmission takes place over a shared wireless medium, external access is easier to accomplish, even though some measure of protection, for example see [2] and the associated special issue papers, is possible at the physical protocol layer. Multimedia communication is also vulnerable within a wired environment, due to packet interception, but the popularity of mobile systems implies that the problem of secure communication needs to be urgently addressed.

Multimedia data should remain confidential and its integrity guaranteed at higher layers of the protocol stack for data protection and privacy reasons [3] and if multimedia providers are to be confident that their intellectual property will be protected. Cryptography is the conventional method of providing confidentiality to data [4], although the cryptographic algorithms are never a secret, as their steps should be open to every one. The latter allows the steps to be openly analysed [5] and prevents the originators of a weak system from relying on obfuscation. What should be secret and needs to be hidden from public and unauthorised access is the cryptographic key that is used by cryptographic algorithms. This is established historically by Kerckhoff’s principle [6] which states that attackers can know the chosen cipher algorithm but not the key. Keys are also employed in message authentication and in certificated authentication. In addition, keys act to jointly guarantee message integrity against tampering, while at the same time authenticating the source of a message.

The five main security services are, in fact, authentication, confidentiality, integrity of data, non-repudiation, and availability [7], though others exist such as reliability and accountability. All of these security services cannot be effectively used without first establishing a proper key management solution, despite the availability of authentication mechanisms such as the Public Key Infrastructure (PKI). The main purpose of this paper is to take an overview of existing (standardized and proposed) key management protocols for multimedia communication. In doing so, the paper identifies key management protocols for multimedia communication and reviews innovations in multimedia key management, including recent patents. These protocols differ from existing methods due to the real-time nature of multimedia communication, which implies that lengthy key negotiation procedures will add additional latency to the connection start-up time.

Key management consists of the distribution of permanent and session keys over an insecure electronic medium. In fact, key management is a general function to secure the multimedia message exchange channel, which is handled in the upper protocol layers and which encapsulates the mechanisms of key generation, distribution, storage, protection and re-keying for the cryptographic algorithms.

There needs to be a mechanism to supply keying material to the security protocols. The keying material can comprise of asymmetric key pairs, secret symmetric keys, and initialization parameters for a session. Most key management protocols use hybrid key establishment schemes in which permanent asymmetric keys serve to establish symmetric session keys. Unfortunately, the process of encryption by asymmetric keys is computational intensive, due to the large-number of arithmetic modulo operations involved. In contrast, symmetric key encryption algorithms are commonly designed for hardware implementation, involving bitwise operations. Therefore, key wrapping can occur in which a symmetric key is transported to the recipient, encrypted by an authenticated public key.

Thus, key establishment [8] can be divided into key exchange and key transport. Key exchange allows two parties to agree upon the shared key, while key transport is the transportation of a key to one party, or a group if the key is created by any one party. As a result, for transportation, the most important step towards secure communication is authentication of the receiver or sender or both. If the authentication is not robust then secure communication cannot proceed.
The remainder of the paper is divided into four Sections: Section 2 describes the context and reviews prior research on the security of mobile networks. Section 3 describes the main security protocol (SRTP) for multimedia transport. Section 4 describes key management protocols associated with SRTP, while Section 5 describes various patents in this area. Finally, Section 6 draws some conclusions.

2. CONTEXT

As they become aware of multimedia systems, people become no longer satisfied with only speaking to each other, they want other services that include multimedia elements such as text, images, animations, high fidelity audio, and video. With the extensive use of the Internet, the emerging multimedia services have moved towards [9] providing customized user-centered services varying from entertainment plus lifestyle applications including mobile TV, and mobile payments (m-commerce) for services such as real-time audio and video data exchange. From 3rd generation mobile services, the mobile phone was promoted as a ‘multimedia computer’ due to: a multi-megapixel camera capable of still and moving images; integrated 3D sound stereo speakers; and real-time operating systems. The known multimedia services provided by 3rd generation mobile phones are: Multimedia Messaging Service (MMS); video calling and conferences; Internet access; radio; push-to-talk services; mobile TV; 3D gaming and multi-gaming; together with the previous generation’s instant messaging and voice calls.

As an example, of a multimedia service, the MBMS (Multimedia Broadcast Multicast Services) [10] are unidirectional, point to multipoint services introduced for GSM-based wireless networks within the 3rd generation of mobile phones. Third Generation Project Partnership (3GPP) has established MBMS [10] with the goals of 1) Less network resource utilization in both access and core networks and 2) To build a scalable, reliable and efficient service platform. 3GPP in release 6 [11] launched MBMS services to meet the increasing demands of multimedia download and streaming applications [12] in mobile networks.

With the growth of such services the security of communication over cellular mobile phone networks, broadband wireless access networks, and unreliable mobile ad-hoc networks is becoming a challenging issue. Data confidentiality is associated with cryptography. However, it is also important to ensure that the data have not been tampered with; that is, their integrity has not been compromised. Cryptographic hashes are the basis of integrity checks. A message should be checked for its integrity even if it is not encrypted. Keys function as a means of authentication, as possession of a key is proof that an owner is who they say they are. Thus, message authentication codes (MACs) are a development of the cryptographic hash in that a key is involved in their generation. In the Hashed MAC or HMAC a key is added to a message before a hash is taken. In fact, in the HMAC algorithm the process is repeated twice.

Multimedia security protocols have three options to manage or distribute keys i.e. 1) manually, 2) automatically and 3) through a trusted third-party. Automated key management derives the short-term and long-term session keys (needed to protect the payload), while manual key management deals with the distribution of such keys. Automated key management ensures that the fresh short-term key is applied in every session, which is not guaranteed by manual systems, because they mostly avail themselves of long-term session keys. If automated key management system is deployed, security is improved, while manual key management [13] is very helpful for debugging a security system.

Much research has been carried out in respect of securing mobile communication sessions in a variety of ways. To take some examples for lightweight, infrastructure-less ad hoc networks, developments have occurred in: implementing security over routers by using secure routing protocols for end-to-end network layer security [14][15][16][17][18]; establishing a trust infrastructure [19] [20] [21][22]; and intrusion detection [23][24][25] [26]. Other research is needed in the area of preventing security attacks on the mobile communication channels such as active and passive attacks,
denial of service (DOS) attacks, and man-in-the-middle attacks.

End user identification is also important for mobile commerce applications. The end user authentication/identification issue in the mobile communication system is addressed by using end-to-end security protocols [27] and in [28] research was carried to detect a previous scheme’s security flaws by performing an impersonation attack on the authentication system before presenting a modified solution to authentication.

The authors of the research in [29] applied the idea of a distributed public-key management service. Servers will collectively sign the certificate of every new node entering in the network. A third party known as the dealer is used to initialize the system. After the initialization, the dealer is ignored and the trust relationship is managed between the server and a node.

The key distribution issues in respect to mobile ad-hoc networks multimedia services are addressed in some papers. In a study [30], the authors worked on a key management scheme called (HFS) Hash Function Scheme for UMTS MBMS services in contrast to the 3GPP 33.246 proposed key management mechanisms (KMM). Their work modifies the previous Key-Tree Scheme (KTS) and resolves IP multicast network securities issues for UMTS MBMS. The authors then compare KTS with their HFS Scheme. The research suggests that HFS reduces the storage and communication overhead for MBMS. However, the research does not clarify the issue of whether the HFS is secure enough for the distribution of keys over a network.

A survey paper [31] has presented the security threats and their countermeasures for the mobile ad hoc networks across the multiple layers of the TCP/IP protocol stack. The survey suggests the use of the IPsec or Transport Layer Security (TLS) [33] protocols for the creation of session keys. However, the work also points out the weaknesses of IPsec for a mobile ad hoc network or MANET [20].

A number of researchers have also investigated the use of the Secure Real-Time Transport Protocol (SRTP) and its associated key management protocols in wired and wireless communications [34]. Ericson and Nokia (mobile telecommunication companies) presented many documents, for example see [35][36], to show that the SRTP is a good option for the protection of streamed MBMS services. They also suggest that Multimedia Internet Keying Protocol (MIKEY), which is an associated SRTP key management protocol, is good enough for the security of MBMS services. Notice that MIKEY is considered in more detail later in this paper.

Research in [37] presented a security architecture based on standard mechanisms to avoid intrusion detection and the flooding attacks on Session Initiation Protocol (SIP) [38], Real-Time Transport Protocol (RTP) [39] and Internet Protocol (IP) protocols for media contents.

3. SECURITY PROTOCOLS

In the past couple of decades, many security protocols have been standardized. The security protocol is the mechanism to secure the data in conjunction with the key management protocol. A security protocol typically comprises [40] of features such as key agreement or establishment; entity authentication; symmetric encryption and message authentication; and reliable application-level data transport.

The security issues regarding authentication start from the session initiation phase known as the Signal plane and terminate when the session is established for data transmission. After the Signal plane, when both parties (peers or groups) have been authenticated, the Media plane starts working. The Media plane deals with the transport of multimedia [41]. Thus, there is a need for security protocols on both the signal and the media planes. However, this paper deals only with the media plane.

Apart from other well-known security protocols such as TLS and IPsec [42], which are already well-known, the most commonly employed multimedia security protocol for wired and wireless networks is Secure Real-time Transport Protocol (SRTP), which is now discussed.
Key Management for Secure Multimedia Communication

SRTP [43] provides message authentication, integrity, confidentiality, and replay protection to RTP traffic. SRTP defines a set of default cryptographic transforms with appropriate key management. SRTP encapsulates the RTP packet in its own packet format and transmits. The same process has been performed with secure RTCP (SRTCP) for the Real-time Transport Control Protocol (RTCP). (RTCP is the session information protocol associated with RTP [39].) SRTP is regarded as secure for unicast and multicast RTP applications.

SRTP provides additional features to reduce the load on key management and to further enhance the security. They include:

- A single key known as the master key can provide keying material for confidentiality, and integrity protection, both for the SRTP stream and the corresponding SRTCP stream. This is achieved with a key derivation function, which derives other keys from the master key (as described shortly).
- In addition, the key derivation can be built up to period wise refresh of the master keys, which limits the amount of cipher text generated by a fixed key.
- Salt keys are used to provide protection against pre-computation and time-memory tradeoff attacks. Salt keys are meant to substitute some bytes of the session key to make the session key more secure.

The cryptographic contexts are maintained by the sender and receiver for the transmission of SRTP streams. The master key is used for the derivation of the session keys, while the session keys are used for encryption and authentication purposes. The key management protocol provides the master key and the other security parameters in the cryptographic context. SRTP key derivation is summarized in Fig. 1, with the packet index used for the sequencing of packets.

SRTP depends on external key management protocols to set up the Master key. These protocols are summarized in Table 1 and discussed in Section 4.

![Fig. (1). SRTP key derivation structure.](image)

### Table 1. SRTP with associated key management protocols.

<table>
<thead>
<tr>
<th>Security Protocol</th>
<th>Associated Key Management Protocols/Schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRTP (Application Layer)</td>
<td>SDES [55], MIKEY [49], ZRTP [54], DTLS-SRTP [58], DTLS-SRTP Key Transport (KTR) [60], GDOL-SRTP [64]</td>
</tr>
</tbody>
</table>

4. KEY MANAGEMENT

To achieve security, key management is a mechanism to supply keying material to the security protocols. The keying material can comprise of symmetric/asymmetric key pairs, secret keys and initialization parameters for a session [44]. For transmission, the most important step to secure communication is authentication. If the authentication is not robust then the remaining security functions cannot be attained.

The basic function of key management is the establishment of keying material. Key establishment can be bifurcated into key exchange and key transport [4]. Key exchange allows two parties to agree upon the shared key, while key transport is the transportation of key to one party or a group if it is created by any one party. Most key management protocols use this type of hybrid key establishment scheme.
The major challenge in designing the key management protocols is complexity. The key management protocols have been standardized by keeping in view the following fundamental features.

1) **Entity Authentication**
   The unilateral (one peer) or mutual (both peers) authentication assures the identity of the communicating parties; that both are valid to start communication.

2) **Confidentiality**
   The key management protocol must assure the key secrecy so that unauthorized parties will not be able to lay their hands on the key.

3) **Fresh session key**
   The key management protocol must assure keys are refreshed at the start of every session.

4) **Perfect Forward Secrecy (PFS)**
   It is a fundamental assurance of any key management protocol that compromising the Master key may not compromise the generated session keys [44][45][46].

5) **Strength against known key attacks**
   The key management protocol is strong against known exhaustive key search attacks.

6) **Forward Secrecy**
   The key management protocol must ensure that if the current session key is compromised then it will not affect the security of the next sessions [47][48].

7) **Backward Secrecy**
   The key management protocol must ensure that if the current session key is compromised then it will not affect the security of the previous sessions [47][48]. Forward and backward secrecy also imply that multiple session keys are independent of each other.

8) **Robustness**
   The key management protocol must tolerate hardware and software failures due to limited connectivity [45].

9) **Efficiency**
   The efficiency of key management protocols can be measured by the computational, memory, communication and energy consumptions costs.

10) **Scalability**
   The key management protocol must be scalable enough; the hierarchical or parallel structure of key derivation must be maintained during the process of key generation and transportation.

According to their communication modes, some key management protocols work for unicast or multicast communication and some are hybrid. The next part of this paper comprises of a concise overview of the key management protocols.

### 4.1 Multimedia Internet Keying Protocol

The Multimedia Internet Keying Protocol (MIKEY) [49] is an important addition to the security of multimedia, as it is specifically designed to tackle the key exchange problem in real-time networks. It is used for the key management of one-to-one, one-to-many and many-to-many small-size group communication. The key management protocol is devised to enable end-to-end security i.e. only the participants involved in the communication have authorized access to the generated key(s) and hence to the content. Key generation, Fig. 2, is efficient.

MIKEY uses a total of eight keys. The keys will be generated on either sender side or both sides (sender and receiver) and comprise of:

a) Traffic Generation Key (TGK)

b) Traffic Encryption Key (TEK)

c) Encryption Keys (total of two, one each for sender and receiver)

d) Authentication Keys (total of two, one each for sender and receiver)

e) Salting Keys (total of two, one each for sender and receiver)

MIKEY supports five methods for transporting/establishing a TGK (item 1 above) or to setup a common secret, for the all communication scenarios by using either:

1) pre-shared key
2) public-key encryption
3) Diffie-Hellman (DH) key exchange.
Key Management for Secure Multimedia Communication

4) HMAC-Authenticated Diffie-Hellman DH-HMAC [50]
5) Reverse RSA (RSA-R) [51]

MIKEY produces a data security association (SA) with the TEK (generated by the TGK) and security parameters/policies as an input to the security protocol. MIKEY also has the capability of establishing keys and parameters for more than one security protocol (or for multiple instances of the same security protocol) at the same time. The crypto sessions (CS), which are a uni- or bi-directional data streams, are secured by the single instance of the security protocol. The Crypto Session Bundle (CSB) is the collection of multiple CS with common TGK and security parameters. Both the CS and SB have their own unique identifiers (CS ID and CSB ID).

The TEK can be used directly by the security protocol or it can be used to derive further master keys from the TEK. It is, however, up to the security protocol to define how the TEK is used.

MIKEY can be used for the updating of the TEKs and the CSs in the current CSB. This is done by executing the transport/exchange phase once again to obtain a new TGK (and consequently derive new TEKs) or to update some other specific CS parameters. MIKEY uses AES-CM, AES-F8 for the encryption algorithms [52] and a 160-bit authentication tag, generated by the Hash-based Message Authentication Code (HMAC) [53] with the SHA-1 [4] [5] cryptographic hash algorithm.

4.2 Zimmerman Real-Time Transport Protocol

Zimmerman Real-Time Transport Protocol (ZRTP) [54] is a cryptographic key agreement protocol that is used to negotiate the keys for encryption between two end points. It provides authentication, and confidentiality to the media streams (specifically voice messages). It uses ephemeral Diffie-Hellman key exchange [42] with hash commitment during the call setup in transport over the same RTP session using the UDP port after session initialization by SIP. ZRTP generates parameters from a shared secret. This is used to generate keys and a salt for SRTP and provides Perfect Forward Secrecy (PFS), as the keys are destroyed after each call session.

ZRTP does all key negotiation in RTP media streams and consequently it is independent of any signalling layer protocol. It employs a Short Authentication String (SAS), which is a cryptographic hash of two Diffie-Hellman values, for the users to compare with each other and detect a man-in-the-middle attack. It executes its key agreements and management in a solely peer-to-peer manner over the RTP packet stream.

ZRTP works in three major key agreement modes, which are:

1. **Diffie-Hellman Mode**
   The Diffie-Hellman (DH) parameters are exchanged.

2. **Pre-shared Mode**
   The DH calculation is not required if the shared secret is available from the previous ZRTP session. This mode is indicated in the commit message.

3. **Multi-stream Mode**
   This is an alternative key agreement method to the pre-shared mode. The main difference is that the pre-shared mode uses a previously cached shared secret, while multi-stream mode requires an active ZRTP session key as the initial keying material.
The working of the DH mode is summarized in Fig. 3. The ZRTP endpoint EP1 initiates the exchange by sending a HELLO message to the other endpoint to confirm the existence of the other endpoint and discover the common encryption algorithm between them. The HELLO message has the SRTP configuration options and the ZRTP ID (ZID), which is a 96-bit random number generated once at installation time. HELLOACK is the response acknowledgement message of HELLO. After the initial HELLO/HELLOACK discovery handshaking, commit messages are sent by an endpoint; the one who will send a commit message will be the initiator of a ZRTP conversation session and will be responsible for driving the key agreement exchange; herein EP1 is the initiator.

After the COMMIT, the Diffie-Hellman shared secret generation process starts. The pvr (representing the public value for responder) with five HMACs parameters (shared secret hashes) are transmitted to EP2. These are used to generate the ZRTP shared secret. EP2 sends the DHPART2 message which contains the same parameters as DHPART1, the pvi (representing the public value for initiator) and the HMAC parameters. The SAS is calculated as the hash of the ZRTP messages (responder’s HELLO, COMMIT, DHPART1, and DHPART2). The freshly generated random number is used as a keys generation material, especially for the DH secret exponents and nonces (unpredictable numbers used only once to identify a message).

After the Diffie-Hellman key exchange method, both EP1 and EP2 generate their SRTP master key and master salt key for the two RTP streams for EP1 and EP2 respectively. Both RTP streams will get the different master key and salt key, so that each end uses the two initiator and responder master key and salt key to encrypt and decrypt the corresponding RTP stream.

After the SRTP keys generation, the CONFIRM1, CONFIRM2 and CONFIRM2 ACK messages are exchanged between end points having parameters of flags and sig, which is the signature algorithm to sign the SAS. The CONFIRM1 and CONFIRM2 messages contain information about the secret keys’ life expectancy and are exchanged in response to the successful completion of the key negotiation process. The CONFIRM2 ACK is the response to stop the further retransmission of CONFIRM2 messages. For the termination of encrypting media, the GOCLEAR message is used. The message does not terminate the session but changes the state of the RTP stream from being encrypted to unencrypted.

As a summary, Table 2 lists ZRTP’s supported algorithms.
Table 2. ZRTP supported algorithms and parameter specifications.

<table>
<thead>
<tr>
<th>Cipher algorithms</th>
<th>AES-128, AES-192 and AES-256 in counter mode and CFB mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TwoFish block cipher with 128, 192 and 256 bit keys</td>
</tr>
<tr>
<td></td>
<td>Camellia block cipher with 128, 192 and 256 bit keys</td>
</tr>
<tr>
<td>Hash algorithms</td>
<td>SHA-256</td>
</tr>
<tr>
<td></td>
<td>SHA-384</td>
</tr>
<tr>
<td>Authentication tag</td>
<td>HMAC-SHA1 with 32 bit and 80 bit tag</td>
</tr>
<tr>
<td></td>
<td>Skein MAC key(^1) with 32 bit and 64 bit tag</td>
</tr>
<tr>
<td>Key agreement algorithms</td>
<td>DH mode with 3072-bit prime and with 2048-bit prime value</td>
</tr>
<tr>
<td>in Diffie-Hellman mode</td>
<td>Elliptic Curve DH with (p=256, 384) and 521)</td>
</tr>
<tr>
<td>SAS type</td>
<td>Base32 encoding and base256 encoding</td>
</tr>
<tr>
<td>Signature type</td>
<td>OpenPGP Signature</td>
</tr>
<tr>
<td></td>
<td>X.509v3 certificate</td>
</tr>
</tbody>
</table>

\(^1\) The Skein hash function is a candidate function to become SHA-3.

4.3 Session Description Protocol

The Session Description Protocol (SDP) \([55]\) with security descriptions (SDES) \([56]\) provides a way to negotiate the cryptographic keys and security parameters for SRTP either in a single message or a roundtrip exchange for unicast streams. SDES itself does not provide authenticated key establishment services (AKE) as other key management protocols provide to the security protocol, which is why the SDES is only recommended to be used with those data security protocols (e.g. IPSec \([57]\) or TLS \([33]\)) that can protect the SDP messages.

Three different crypto-suites (identification of encryption and authentication transforms) are defined by SDES, by using distinct variations of AES and SHA-1 to provide encryption and authentication. These three crypto-suites are:

1. AES_CM_128_HMAC_SHA1_80
2. AES_CM_128_HMAC_SHA1_32
3. F8_128_HMAC_SHA1_32

All encryption methods set the length of the master key as 128 bits, the master salt key as 112 bits, the encryption key as 128 bits, and SRTP & SRTCP authentication keys are of length 160 bits. Even if the same crypto-suite is used by both sender and receiver, the same keys and salts are not used by each side, so that each side will generate and pass the parameters using an SDP message. This SDP message requires the other security protocols for its secure transmission.

The “inline” key method is used by the SRTP security description. The inline key holds the keying material (master key and salt) and key related policies such as lifetime and association of Master Key Identifier (MKI) with incoming SRTP packet and a specific master key, such that:

```
"inline" <key|salt> ["|" lifetime] ["|" MKI :|" length]
```

key|salt: master key and salt concatenation
lifetime: master key lifetime (max number of SRTP or SRTCP packets using this master key)(optional field)
MKI:length MKI and its field length in SRTP packets

4.4 Datagram Transport Layer Security

Datagram Transport Layer Security (DTLS) \([58]\) provides secure data transmission with parameter negotiation and key management mechanism on a media plane.

The DTLS-SRTP is the SRTP extension of DTLS which offers SRTP encryption with DTLS key management. In DTLS-SRTP, the data is protected by SRTP, while the keying material, algorithms and SRTP parameters are established by the DTLS handshake.

A DTLS-SRTP session is a point-to-point session (unicast RTP session) with exactly two participants. Each session contains the single DTLS association and either two SRTP contexts (bi-directional flow of data) or one SRTP context (unidirectional flow of data). A single
DTLS-SRTP session only protects data carried over a single UDP source and destination port pair.

Two framing formats are used, i.e. SRTP packets encapsulating the RTP packets and the DTLS record packets having the handshake messages. The RTP and RTCP communication between the client and server is initiated after the DTLS handshake process.

There are different SRTP keys for client and server RTP packets. All the RTP packets generated by clients use the same keys and all RTP packets on servers use the same key for the same channel. The SRTP implementation must ensure that the SSRC (Synchronization Source) values for all the RTP sources over same channel must be different to avoid the “two-time pad” SRTP problem [40].

Multiple SRTP Security Profiles are used in DTLS-SRTP with multiple crypto algorithms such as

- SRTP_AES128_CM_HMAC_SHA1_80,
- SRTP_AES128_CM_HMAC_SHA1_32,
- SRTP_NULL_HMAC_SHA1_80,
- SRTP_NULL_HMAC_SHA1_32.

All algorithms use different key lengths and sizes for their tags.

DTLS-SRTP provides the following options:

- **TLS PRF (Pseudo Random Function)** is used to generate initial keys to input into the SRTP Key Derivation Function (KDF).
- The **Key Derivation Rate (KDR)** is zero, means the keys will not re-generate by using the SRTP sequence number.
- The key derivation methods and the other security parameters are the same, as mentioned in the SRTP RFC [43].

The master keys used for the protection of the DTLS record and SRTP packets are generated by the TLS exporter [59] by the following formula and then the resulting keying material is assigned to the SRTP Client and Server to make their Master and Master salt keys.

2 * (SRTP.master_key_len + SRTP.master_salt_len) bytes of data

The four keying material values (master key and master salt key for each direction) are used by the SRTP Key Derivation Function (KDF) to derive the further SRTP and SRTCP keys to encrypt and authenticate the packets sent by the clients or servers. This process is summarized in Fig. 4.

![Fig. (4). DTLS-SRTP key generation structure.](image)

After the expiry of keys, a new DTLS session will be established for the key replacements with the new handshake over the existing DTLS sessions. The use of multiple data protection negotiations in a single handshake adds some complexities to the DTLS-SRTP protocol, resulting in problems such as bid-down attacks, DTLS record and SRTP packets confusion, confusion of two frames sequence numbers, renegotiation process and extra decryption cost.

### 4.5 DTLS-SRTP Key Transport Protocol

To overcome the problems of SRTP key distribution for multiple recipients in DTLS-SRTP, the DTLS-SRTP key transport (KTR) [60] has been proposed as an extension of DTLS-SRTP to deploy the DTLS-SRTP in multicast scenarios (small groups) as well.

The same SRTP keying material is distributed to the multiple DTLS-SRTP
Key Management for Secure Multimedia Communication

recipients. The key-transport extension is also negotiated in the TLS handshake procedure during the call session with the normal parameters of DTLS-SRTP. The difference between DTLS-SRTP and DTLS-SRTP KTR is that in DTLS-SRTP KTR the keying material is not generated by the TLS handshake. The SRTP keys are sent to multiple recipients within the DTLS session as a new TLS content type which has four types of message. The DTLS-SRTP KTR is basically for the secure key transmission to the multipoint users, so the key generation mechanism is the same as the DTLS-SRTP.

Every listener establishes a unicast DTLS-SRTP session with the speaker and the speaker sends the SRTP keys to each listener by using DTLS-SRTP KTR. The DTLS-SRTP KTR is based on multiple Point-to-Multipoint models i.e.

- Point-to-Multipoint model based on the RTP mixer model [39].
- Point-to-Multipoint model based on multicasting
- Point-to-Multipoint model based on video switching Multi Control Units (MCUs)

The new keys are derived on the sender and receiver sides whenever the request is initiated. DTLS-SRTP KTR uses subset-difference based key management [61] which is by the Logical Key Hierarchy (LKH) [62]. Every new listener requires the new SRTP key when it joins or leaves the communication. The DTLS-SRTP requires encryption of the new SRTP key $N$ times for $N$ active listeners, which requires more CPU time and cycles, while LKH allows new SRTP key to be encrypted once for $N$ listeners. The keys are distributed by the key servers which have the database of all hierarchical keys.

The proposed draft also compares the CPU utilization time in two scenarios, 1) Security descriptions [56] working with DTLS-SRTP and 2) the performance of DTLS-SRTP key transport in the case of interworking with other key management systems.

Two-time pad and the group communication joining/leaving notification remains a problematic issue in DTLS-SRTP key transport protocol.

4.6 GDOI for SRTP

Group Domain of Interpretation (GDOI) [63] is a group key management cryptographic protocol. It is used for IP security protocols and can be applied at the application layer.

The GDOI is flexible enough to provide services to many data security protocols. In fact, an Internet draft [64] has been purposed for using the GDOI with SRTP. GDOI is the authenticated unicast and multicast key establishment protocol for groups and provides support for the member revocation algorithms i.e. by means of LKH [62]. This section now describes the usage of GDOI for SRTP.

In addition to signalling for SRTP, the GDOI-SRTP optionally uses Encrypted Key Transport (EKT) protocol for signalling as well. The GDOI defines two new payloads for SRTP; the SA-TEK, Security Association Traffic Encrypting Key payloads for SRTP (SRTP SA-TEK) and EKT (EKT SA-TEK).

GDOI works above the transport services. It communicates with SRTP using its API. GDOI GCKS provides the cryptographic keys, together with cryptographic and session parameters to SRTP via the API to a GDOI member according to a pre-configured group policy. The group controller/key server (GCKS) automatically generates some information about keying material.

In centralized configurations, where the GCKS is remote to the SRTP sender, the EKT is used to transport the SSRC, the Rollover Counter (ROC), and current SRTP sequence number (SEQ), to the SRTP receiver. The EKT correctly initializes these parameters for late joiners of the multicast group or following RTP SSRC collision repair [65]. But in distributed GCKS scenarios, the GCKS itself initializes these parameters by getting the data through an API with the member. The decision of either using EKT or not can be set in configuration options of GDOI-SRTP. EKT is required in a
case when the GCKS cannot reliably initialize the SA-TEK with SSRC, ROC and SEQ fields. Two SA-EKT payloads are used with EKT:

1) SRTP SA-TEK payload, which is mandatory, and defines the SRTP master key and salt; and
2) EKT SA-TEK payload which defines the EKT key.

One key download payload is used in either case to send in a GDOI-SRTP exchange, as when EKT is signalled, GDOI does not download the SRTP key as a TEK.

**4.7 Security attacks**

Table 3 shows the SRTP associated key management protocols on the basis of their existing countermeasures against security attacks. As such Table 3 is self-explanatory.
Table 3. Key management Protocols Summaries with their existing countermeasures against security attacks

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
<td>Dependent on linked security protocol</td>
<td>Encryption schemes</td>
<td>Encryption methods</td>
<td>Encryption methods</td>
<td>Public and shared key cryptography</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>Provided with pre-shared key, Public key encryption, D-H algorithm, authentication tag</td>
<td>Provided by Short Authentication String (SAS) and D-H Algorithm</td>
<td>Provided by authentication tag, fingerprints, PKIX certificates</td>
<td>Provided by authentication tag, fingerprints, PKIX certificates (5763)</td>
<td>Provided by Pre-shared keys/ Public Key encryption/ Digital signatures</td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td>Dependent on linked security protocol</td>
<td>Envelope key, Authentication</td>
<td>CRC, MAC, Short Authentication String (SAS)</td>
<td>Certificates</td>
<td>Certificates</td>
<td>Symmetric key cryptography, Digital signatures</td>
</tr>
<tr>
<td>Non-repudiation</td>
<td>Dependent on linked security protocols</td>
<td>Envelope key, Authentication</td>
<td>Short Authentication String (SAS)</td>
<td>Certificates</td>
<td>Certificates</td>
<td>Symmetric key cryptography, Digital signatures</td>
</tr>
<tr>
<td>Access Control</td>
<td>Authentication of linked security protocol</td>
<td>Authentication</td>
<td>Short Authentication String (SAS)</td>
<td>Authentication</td>
<td>Authentication</td>
<td>Authentication</td>
</tr>
<tr>
<td>Replay attack</td>
<td>Dependent on linked security protocols</td>
<td>Timestamps</td>
<td>Digital signatures over SAS</td>
<td>Certificates</td>
<td>Certificates</td>
<td>Nonce, Digital Signatures</td>
</tr>
<tr>
<td>DOS attack</td>
<td>Dependent on linked security protocols</td>
<td>Initiator Identity in message, MAC</td>
<td>Hash chains computed with nonce</td>
<td>DTLS handshake</td>
<td>DTLS handshake</td>
<td>Digital signatures</td>
</tr>
<tr>
<td>Man-in-the-middle Attack</td>
<td>Dependent on linked security protocols</td>
<td>Authentication</td>
<td>Short Authentication String (SAS)</td>
<td>DTLS handshake</td>
<td>DTLS handshake</td>
<td>Authentication, MAC</td>
</tr>
<tr>
<td><strong>Connection Hijacking Attack</strong></td>
<td>Authentication of linked security protocol</td>
<td>Authentication</td>
<td>Short Authentication String (SAS)</td>
<td>Authentication</td>
<td>Authentication</td>
<td>Authentication, MAC</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------</td>
<td>--------------</td>
<td>----------------------------------</td>
<td>--------------</td>
<td>--------------</td>
<td>--------------------</td>
</tr>
<tr>
<td><strong>Impersonation</strong></td>
<td>Authentication of linked security protocol</td>
<td>Authentication</td>
<td>Short Authentication String (SAS)</td>
<td>Authentication</td>
<td>Authentication</td>
<td>Authentication, MAC</td>
</tr>
<tr>
<td><strong>Communication Mode</strong></td>
<td>Supports unicast communication for SRTP streams and multicasting with SIP</td>
<td>Supports hybrid, unicast and multicast</td>
<td>Supports hybrid, unicast and multicast with dual ZRTP sessions</td>
<td>Supports bidirectional SRTP unicast communication</td>
<td>Supports hybrid, unicast and multicast</td>
<td>Supports Group Communication</td>
</tr>
<tr>
<td><strong>Transport Protocol</strong></td>
<td>UDP</td>
<td>UDP</td>
<td>UDP</td>
<td>UDP</td>
<td>UDP</td>
<td>IP, UDP, TCP</td>
</tr>
</tbody>
</table>
Key Management for Secure Multimedia Communication

5. MULTIMEDIA KEY MANAGEMENT PATENTS

Various patents related to SRTP and its associated key management protocols used for multimedia transmission security are now reviewed. The patents show the suitability of key management protocols for multimedia key management in different inventions.

The inventors in [66] presented methods for bandwidth efficient cryptographic synchronization of data packets by periodically appending the roll-over counter (ROC) value with the data packet on predetermined sequence number values. SRTP is used for the data transmission from sender to receiver. The sender first determines the sequence number of a transmitted packet; either it is divisible by \( R \) or not. The \( R \) is the mutually agreed integer value from sender and receiver. The selection of the \( R \) value can be performed using protocols such as SIP, SRTP and MIKEY. If the sequence number is evenly divisible by \( R \) then the ROC value is appended to the data packet and sent to the receiver or vice versa. The receiver side performs an integrity check across the received data packet.

The invention in [67] is the securing of the IP-based Multimedia Subsystem (IMS – a recent 3GPP defined concept) communication channel with end-to-end encryption of data transmission in the IMS media plane. RSA encryption and Diffie-Hellman encryption will be used for generation of the keys and to secure the communication path. A secure communication channel is established between sender and receiver devices, at least one of the devices being associated with the network core, which includes interpretation data and means for using the interpretation data to interpret the messages communicated between sender and receiver devices.

The authors of [68] invent a mechanism to secure the applied key management protocol, i.e. MIKEY, by supplying an additional symmetric key. The first symmetric key will be used for both sender and a receiver device to start the secure communication and the time-variable parameter is transmitted from the sender to the receiver. The second symmetric key will be calculated by taking the first symmetric key and the first time-variable parameter as an input of pre-defined function. The second symmetric key is used for the protection of the key management protocol.

In [69] is an invention of a method, apparatus, product and storage medium for securing communication and data connections by using cryptographic keys. The invention provides a flexible light-weight exchange mechanism for cryptographic keys, which provides good security for signalling paths without an expensive computational cost. The invention also provides a variety of possibilities for establishing keys between communication endpoints, which supports appropriate selection of good solutions to problems like forking, retargeting, multicasting and media key management problems. Different modes provide better performance than earlier key management standards like SDES, in terms of forking and retargeting and some are better than MIKEY-RSA-R, in terms of computational cost. The invention replaced some key-info related semantics of SDES to enhance the performance.

The patent in [70] is an invention of a system for the transmission of security policies in multicast sessions to improve the key exchange in previous schemes, which cannot realize the safe negotiation of the security policy of an SRTP multicast session. The method is beneficial for in-band security policy negotiation as it avoids the early arrival of content and the security policy transmission to the receiver avoids indefinite trusted terminals. The DTLS (Datagram Transport Layer Security) is used in the first step to maintain the secure session for transmitting the security policy response and the SRTP multicast session data, and sending the multiplexed data to the receiver. The method of receiving the security policy of the multicast sessions includes the generation of security policy requests after the DTLS session setup. The multicast receiver has a key pulling unit that is configured to set up the DTLS session, generate a security policy request, and send the multiplexed SRTP session data plus security policy to the sender. While the sender receives the security policy, constructs a response and then sends the multiplexed SRTP session data.
Key Management for Secure Multimedia Communication

Recent Patents on Computer Science

plus constructed security policy response to the receiver. The security policy includes a content key (CK), key encryption key (KEK) and SRTP session information. The CK is used to provide the primary key to the SRTP stack, and the KEK protects the privacy and integrity of the security policy updating data. The KEK is optional in the case when the security policy does not need to be updated.

The authors of [71] invented a method to encrypt an advertisement and have an associated first decryption key multiplexed into a key distribution system. The method identifies the gap in an entertainment stream where an advertisement may be inserted or substituted (spliced). The advertisement and entertainment stream will be decrypted by the decryption key from the key distribution system and rendered in chronological order. The entertainment stream must be encrypted at the real-time protocol layer. The invented method introduces a key distribution system that supports at least GDOI or Internet Streaming Media Alliance (ISMA) authentication and encryption. The cryptographic key source must be established to receive and supply the keys for encryption and decryption based on GDOI-SRTP.

Some patents are also found which discourage the use of SRTP associated key management protocols and invent their own methods. For example, [72] is an invention of an efficient key exchange system in SIP. The patent is a cost-effective secure communication system, which establishes a communication session by using bi-directional secure key exchanges between two parties (sender and receiver) with at least one proxy server between sender and receiver. By these bi-directional key exchanges the session keys and the master keys with relevant parameters will be exchanged. The invention discourages the use of known key management protocols like ZRTP and DTLS-SRTP. It is pointed out that these protocols are expensive to implement with SIP sessions, and have subtle vulnerabilities.

6. CONCLUSION

In this paper we have surveyed many possible SRTP associated key management protocols for multimedia transmission security and a number of promising patents. From the literature survey, it is revealed that few key management protocols for multimedia streams have been implemented for the key establishment/distribution of cellular and mobile ad hoc network, these protocols being ZRTP and MIKEY. However, the paper has highlighted the key generation and maintenance mechanisms of a variety of other key management protocols to cover the associated key management protocols of SRTP. Some analysis of SRTP and key management protocols has taken place and this is discussed briefly here.

It is established that SRTP can attain high throughput and low packet expansion and prove to be a suitable protection for heterogeneous environments (i.e. a mix of wired and wireless networks). It can also provide support to ad hoc mobile networks. However, all the session keys are derived from the single master key. Thus, there could be a serious security compromise if the same set of keys is generated by the master key in subsequent sessions. Therefore, it is recommended that automatic key management should be used for the SRTP and SRTCP keying material.

The synchronization source SSRC is unique between all the RTP streams in the same RTP session, so the master key must use the SSRC for the derivation of session keys. Even with different SSRC, the extensive use of the same master key results in chances of collision and time-memory trade-off attacks, which can be solved by substituting salt session keys.

The master key can be of 128, 192 and 256 bits. SRTP uses session keys with different key sizes. The session encryption key is 128 bits. The session authentication key is 160 bits, while the session salt key is 112 bits. SRTP has a “seekable” stream cipher that prevents denial-of-service attacks.

SRTP also provides effective confidentiality to the RTP payload and header information. SRTP is robust to denial-of-service and replay
Key Management for Secure Multimedia Communication

Recent Patents on Computer Science

Key management attacks because of strong authentication. Thus, SRTP is a reliable choice for unicast and group communication multimedia systems.

Almost all the mention key management protocols use the Diffie-Hellman (DH) algorithm. Only an exhaustive key search can be used to find the key. In order to prevent that, the minimum key length should be more than 90 bits, the size of the large prime number used should be 180 bits, and the size of the modulus must be 1400 bits. The use of the DH algorithm is desirable as it provides perfect forward secrecy and may be used without a PKI (Public Key Infrastructure). However, it has higher computational and bandwidth resource consumption. It should be recalled that without an authentication process the DH algorithm is more vulnerable to the man-in-the-middle attack. The advantage of using certified DH public keys reduces the overhead of prior session establishment communication such as any handshake procedure or initial dialog opening packets and establishment of pseudo-sessions from source to destination parties. There is no additional overhead of sending encryption and authentication keys to the destination.

ZRTP has some unique features that do not exist in other key management protocols i.e. it uses a public key but does not rely on PKI and it is uses DH but not in a conventional way. It is specifically designed for unicast media sessions. However, for multiparty secure conferencing, separate ZRTP sessions may be negotiated between each party and the conference bridge. ZRTP is used for VoIP and MIKEY for the multimedia security, both over mobile networks. DTLS-SRTP is also used for VoIP.

Upon completing the review, we realize that key generation and distribution mechanisms are a very important facet of any security scheme. We also envision that there is still plenty of work left in the research of multimedia key management protocols and there is a need for more inventions, to work in more challenging scenarios.

This paper has explained the use of various systems and has also brought out the shortcomings and advantages of these. It is hoped that it will serve as a good reference point for future researchers of key management and related patents.

CONFLICT OF INTEREST

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, “Innovations in Key Management for Wireless Multimedia Communication”.

REFERENCES

Key Management for Secure Multimedia Communication


[29] Hanka, O., Eichert, M., Pfannenstein, M., Eberspächer, J., Steinbach, E. A distributed public key infrastructure based on threshold cryptography for

the HiiMap next generation Internet architecture. Future Internet 2011; 3, 14-30.


[36] 3GPP TSG SA WG3 Ad hoc (3 - 4 September 2003)- Introducing SRTP in TS 33.246


Key Management for Secure Multimedia Communication


Recent Patents on Computer Science


