Security for Real-time Multimedia in Heterogeneous Environments

Mats Näslund
Communications Security Lab
Ericsson Research
mats.naslund@era.ericsson.se

Based on joint work with
M. Baugher, D. McGrew, D. Oran @ Cisco
J. Arkko, R. Blom, E. Carrara, F. Lindholm, and K. Norrman @ Ericsson
Outline

• Background
  – Environment and assumptions
  – Scenarios
  – General design goals and requirements
• Security Protocol: SRTP
• Key management: MIKEY
• Comparison to existing protocols (IPsec, TLS, ….)
• Summary and conclusions
Physical Environment

Heterogeneous Networks

3G Operator

Internet

WLAN Hotspot

Fixed

Heterogeneous Networks

Internet

3G Operator

WLAN Hotspot

Fixed
Basic assumptions

Conversational Multimedia or Streaming Applications

The Real-time Transport Protocol (RTP) [RFC 1889] used for media transport

Control protocol for media set up

E.g. Session Initiation Protocol (SIP) [RFC 3261] or Real-time Streaming Protocol (RTSP) [RFC 2326]
Scenarios

peer-to-peer communication (e.g. SIP call)

one-to-many (multicast) (e.g. web TV using RTSP)

small-size groups (e.g. SIP conference call)
Characteristics to Consider

- IP based, packet switched communication
- Conv. MM Applications (real-time)
- Wireless links in the path
- Low power devices

What are the requirements implied?
Wireless Links

can have

- Low bandwidth
- Bit-errors
- Unequal Error Protection (UEP)

⇒ Minimize # roundtrips and data for key-exchange

Security processing should not expand messages too much

⇒ Security processing should not increase bit error-rates or packet loss rates, nor “move errors around”
Typical Voice/Video Applications

Most CODECS built to tolerate a few bit errors, but packet loss degrades speech quality

Again:

⇒ Security processing should not increase BER
⇒ Security processing should not increase packet loss
Typical VoIP Application

```
| IP(v4) | UDP | RTP | Encoded Speech |
```

- 40 bytes
- ≈ 30 bytes

Header Compression (ROHC, RFC 3095) needed for economy:

```
ROHC  Encoded Speech
```

- 1-2 bytes

⇒ Security processing must allow header compression
IP + Real-time

IP: Packets may be re-ordered and/or lost

Real-time ⇒ no retransmission (typically UDP)

⇒ The security protocol must be able to fast-forward/rewind in the “decryption stream” with no extra delay
Low-power Devices

⇒ Security Processing must
- be efficient
- have small footprint
- avoid public key operations
Summary and Implications of Requirements:
Security Protocol Implications

Bandwidth economy \( \Rightarrow \) stream cipher (no padding)
short(optional) auth tags

Speech quality preservation \( \Rightarrow \) stream-cipher (error-robust)
(in presence of errors) optional auth

Header Compr. preservation \( \Rightarrow \) headers in the clear

Packet loss/reorder \( \Rightarrow \) per packet synch (without IV)

Security and efficiency \( \Rightarrow \) state of the art transforms
Summary and Implications of Requirements: Key-Exchange Implications

- **Efficiency**: ➞ pre-shared key possibility
- **Wireless links**: ➞ one roundtrip
- **Group and multicast scenarios**: ➞ key “push”
SRTP
The Secure Real-time Transport Protocol
The SRTP packet

- confidentiality of the RTP payload
- integrity protection of the entire RTP packet & replay protection (optional)
- MasterKey Identifier (optional), signals which key to use

RTP Header | RTP Payload | MKI | Auth Tag

encrypted

authenticated
Default Encryption: AES Counter Mode

ind: index/seq. no of RTP packet (48 bits)
S: “salt”/randomizer (up to 112 bits)

Max: $2^{48}$ packets
$2^{16}$ blocks

Key (128 bits) → AES → AES → ... → AES → plaintext → ciphertext
Default Message Authentication: HMAC_SHA1

Well known

Default 128-bit key and 32-bit auth. tag size

SRTP packet

32 bits
Key Derivation/Refresh

- Need session keys for auth./encryption
- First key derivation generates these from master key
- Subsequent (optional) rounds of key derivation ‘refreshes’ the session keys every r:th packet
- Key derivation also based on AES-CM

Key mgmt

index/counter
r (refresh rate)

key derivation

master key

encr session key

auth session key
SRTP Processing

- Roll-over count
- Master key (from key mgmt)
- Seq nr
- RTP Packet
- Key derivation/refresh function
- Encryption key
- Encryption
- Authentication key
- Authentication

Unauthenticated SRTP Packet
Authenticated SRTP Packet
MIKEY
Multimedia Internet KEYing
MIKEY Properties

- Can create keys and parameters for multiple secure sessions
- Limited possibility for negotiation
- Uses one roundtrip
- Possible to integrate into session control protocols (e.g., SIP or RTSP)
General Notation

Crypto Session Bundle 1

| Audio stream 1 (SRTP) | Crypto Session A
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Video stream 1 (SRTP)</td>
<td>Crypto Session B</td>
</tr>
</tbody>
</table>

Crypto Session Bundle 2

<table>
<thead>
<tr>
<th>Audio stream 2 (SRTP)</th>
<th>Crypto Session C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video stream 2 (SRTP)</td>
<td>Crypto Session D</td>
</tr>
</tbody>
</table>
General Overview

Key transport/exchange

TEK generation key (TGK)

Crypto Session ID

TEK derivation

TEK(s) (+ security parameters)

(Crypto Session Bundle)

Crypto Session

(Security Protocol)
Key Transport/Exchange

*Always included*: Timestamp (anti-replay), crypto info

**Key transport** *(pre-shared or public key)*

\[ E(\text{key}, \text{TGK}) \]  
\[ + \text{ MAC/Signature} \]

\[ \text{[Verification MAC]} \]

**Key exchange** *(Diffie-Hellman type)*

\[ g^x + \text{ Signature}(g^x) \]

\[ g^y + \text{ Signature}(g^x || g^y) \]

\[ \text{TGK} = g^{xy} \]
Pseudo Random Function (TEK Derivation)

**Input:** inkey of length inkey_len, Label  
**Output:** outkey of desired length, outkey_len

Similar to TLS...
The P-function

Label

HMAC

HMAC

HMAC

inkey

Output:

HMAC

HMAC

HMAC

HMAC

outkey_len

Mats Näslund, Ericsson Research

Security for Real-time Multimedia in Heterogeneous Environments
Combined Example
Secure Conversational Multimedia Example

v=0
o=alice 2891092738 2891092738 IN IP4 lost.somewhere.com
s=Cool stuff
e=alice@w-land.org
t=0 0
c=IN IP4 lost.somewhere.com
a=key-mgmt:MIKEY uiSDF9sdhs727ghsd/dhsoKkdOokdo7eWsnDSJD...
m=audio 49000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 52230 RTP/SAVP 31
a=rtpmap:31 H261/90000

v=0
o=bob 2891092897 2891092897 IN IP4 found.somehere.com
s=Cool stuff
e=bob@null.org
t=0 0
c=IN IP4 found.somehere.com
a=key-mgmt:MIKEY skaoqDeMkdwRW2781
m=audio 49030 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 52230 RTP/SAVP 31
a=rtpmap:31 H261/90000
Yet Another Protocol?

IPsec and IKE:
- Too much overhead
- Not header compression friendly
- Lack of suitable transforms
- Too many roundtrips for key exchange
- Not possible to “push” keys

(W)TLS:
- Only TCP support
- Key mgmt tied to security protocol
- Not possible to “push” keys
Summary and Status

• SRTP
  – Protocol to secure the Real-time Transport Protocol
  – Joint IETF draft with CISCO
  – Passed Audio Video Transport WG Last Call. Now with IESG.

• MIKEY
  – Key management protocol to distribute keys for SRTP
    (and similar protocols)
  – Ericsson IETF drafts, WG items in:
    • Multicast Security (MSEC) group, and
    • Multiparty Multimedia Session Control (MMUSIC) group

For more info, see http://standards.ericsson.net/fli/
The End