Key Management for Multimedia Sessions (and SRTP)

Fredrik Lindholm,
Ericsson Research

Design Criteria for Multimedia Session Key Management in Heterogeneous Networks
<draft-blom-mm-kmgt-00.txt>

Rolf Blom
Elisabetta Carrara
Fredrik Lindholm
Jari Arkko

Key Management for Multimedia Sessions
<draft-carrara-mm-kmgt-sol-00.txt>

Elisabetta Carrara
Fredrik Lindholm
Mats Näslund
Karl Norrman
Jari Arkko

Outline

- Background and Problem
- Requirements
- Proposed solution
- Questions

Scenarios and Environment

- Multimedia sessions
- Peer-to-peer communication
- Multicast
- Small groups

- Heterogeneous environment

The origin of the key management work

The Secure Real-time Transport Protocol (SRTP)

- Work item in the AVT WG
- Encryption of the RTP payload
- Optional authentication of the RTP packet

Some of the properties:

- a low computational cost
- limited packet expansion
- the preservation of RTP header compression efficiency

Problem

- Need a key management protocol for SRTP and other possible multimedia security protocols
- Must be able to work in the stated scenarios and environments

- Any existing solutions?
  - Kerberos, IKE, GDOI, TLS etc.

General Requirements

- It MUST give e2e security
- It MUST support multi-party and multicast sessions
- It SHOULD be possible for a single party to generate a key and transmit it securely to another party (i.e. a key transport mechanism SHOULD be supported)

General Requirements (cont.)

- It SHOULD be adaptable to also support future multimedia security protocols
- Both unidirectional and mutual authentication SHOULD be possible
- SAs SHOULD NOT be associated/indexed with IP addresses

Admin Domain Requirements

- It MUST be possible to manage the users in different administrative domains.
- It MUST allow users from different administrative domains to communicate securely.
- It MUST at least be scalable to a size close to current Public Switched Telephone Network (PSTN)

Key Maintenance Requirements

It MUST specify mechanisms for:

- re-keying (new master key),
- session key derivation (from the master key), and
- key refresh (new session key)

- convenient keying and re-keying mechanisms suitable for small groups MUST be provided

DoS Protection Requirements

- It SHOULD NOT create any state at the responder side until a positive authentication has been performed
- It SHOULD NOT devote substantial computation resources to peers whose claimed source has not been verified to be operational

Efficiency Requirements

- Only the minimal set of necessary functions SHOULD be mandatory to implement
- It MUST support a selection of efficient security protocols, algorithms, and parameters

Efficiency Requirements (cont.)

Strive to reduce:

- the number of round trips (MUST)
- the size of the messages (SHOULD)
- the number of expensive cryptographic operations (SHOULD)

To Conclude

1. End-to-end protection requirement
2. Key transport mechanism requirement
3. Small groups/multicast
4. Efficiency requirements

	1	2	3	4
Kerberos	No			
IKE		No	No	No
GDOI		?		No
TLS		No	No	

Basic ideas

- Using key transport mechanisms
- Integrate (if possible) with session control protocols

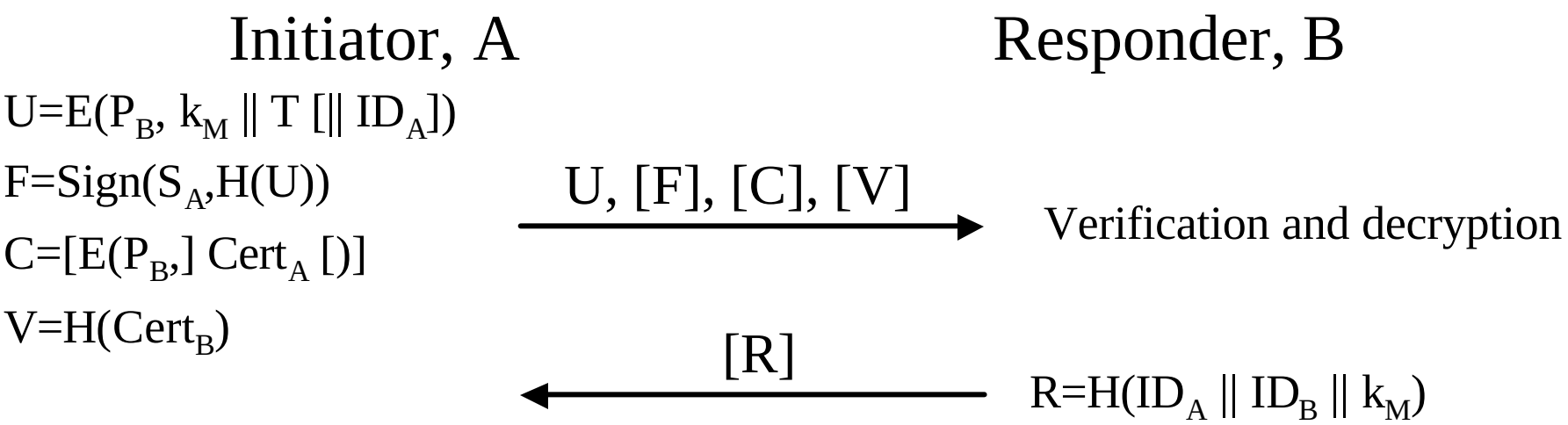
Key transport mechanisms

- Shared secret based
- Public key based

Key negotiation mechanism

- Diffie-Hellman based exchange

Public-key Based Mechanism

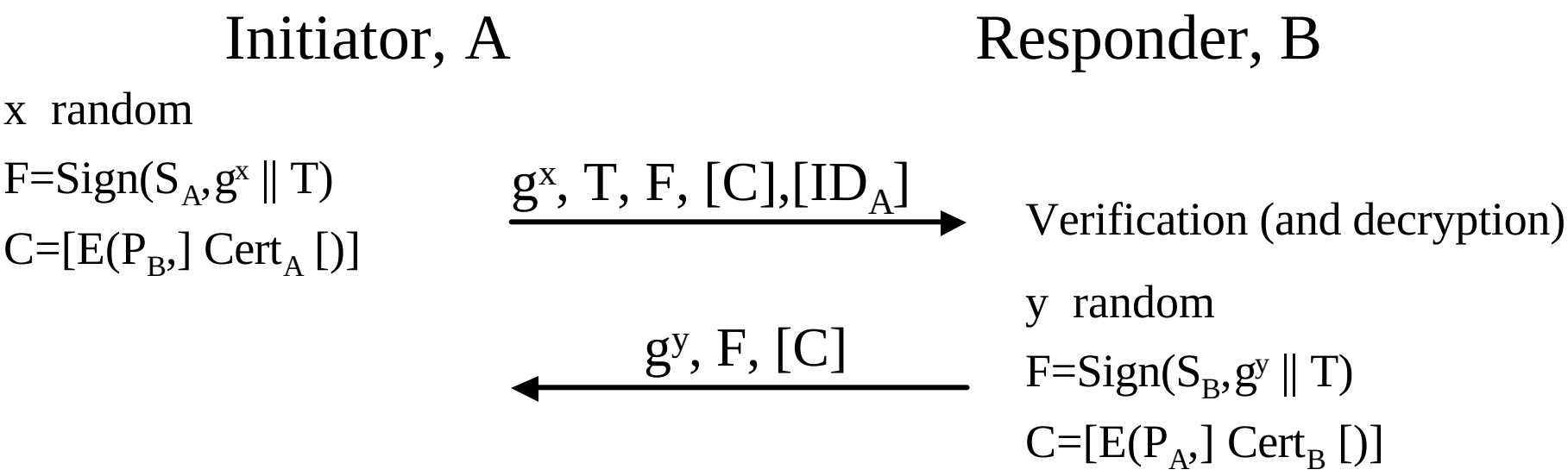


Public/Private key:	P_X/S_X
Master key:	k_M
Timestamp:	T
Certificate:	$Cert_X$

Shared Secret Based Mechanism

- Similar to the public-key based
- Exchange
  - public-key encryption with symmetric
  - signature with MAC

Diffie-Hellman Based Exchange



Public/Private key: P_X/S_X Certificate: $Cert_X$

DH group generator: g Timestamp: T

Master key: $k_M=g^{xy}$

Other Issues

Negotiation of cryptographic parameters

Initiator: sends one proposal

Responder: Possible error message with the supported parameters

- Session Key derivation and key refresh
  - AES in Output Feed-Back Mode
- Re-keying
  - Executing the protocol again (without previously exchanged static information)

Integration with Session Control Protocols

Two proposed protocols:

- SIP (Session Initiation Protocol)
- RTSP (Real Time Streaming Protocol)

- Use the attribute field (a=) in SDP (Session Description Protocol)

Time for Comments and Questions