

Information Security

Holger Schlingloff

Jan 30th, 2002



Key exchange and -management

Potential attack point for message exchange with asymmetric cryptosystems is, as we have seen in the spoofing example, the authenticity of the communication partners: How do communication partners know their mutual identity?

(The same problem appears in ordinary surface mail).

Solution with trust centers and smart cards: a trustworthy third party. User inserts card into machine, types in his PIN, the rest is automated.

Which protocols/algorithms are used?

Definitions

Definition 1 (protocol)

A protocol is a distributed algorithm involving several parties, which is defined by a sequence of steps which fix the actions and messages between the parties to achieve the desired goal.

Definition 2 (Key establishment)

Key establishment (Schlüsselselfestlegung) is the process or protocol to establish a common secret between two or more parties for later cryptographic use.

Two variants of key establishment:

- **Key transport (Schlüsselaustausch):** One party creates the key and sends it to the other(s)
- **Key agreement (Schlüsselvereinbarung):** The key is calculated by all involved parties from information contributed by all parties.

Keys by itself can be symmetric or asymmetric, and dynamic (for one session only) or static (a priori, for several sessions).

This yields a matrix of key establishment protocols:

	KEY ESTABLISHMENT	
	key transport	key agreement
symmetric		
asymmetric		
	dynamic	static

Prerequisites for key establishment protocols:

- **Trusted third party** (Trusted server, authentication server) S
stepwise building of trust
- various assumptions on A and S , e.g., each communication partner A received from S a key which is only known to A and S
 - A must identify him/herself personally with S ; e.g., identity card, passport etc.
 - S must keep the key secret (prevention against housebraking, burglary, fraud, ...)
 - A must keep the key secret (e.g. by SmartCard+PIN; legal consequences for dissemination)

Authentication problem: person / data / key

- Assumptions on attackers; e.g., no possibility for cryptanalysis
 - Recording, modification, deletion, detour, or replay of packets
 - Initiation of the protocol or interference with it
 - Known-key-attack: does a breaking of the key for one session lead to the possibility of calculating subsequent keys?

Key hierarchy: Master key, server key, session key, ...

Protocols for key exchange

Notation: $(i) A \rightarrow B : M$ In step number i , person A sends to B message M

$\{M\}^k$ Message M , encoded with key k .

Definition 3 (nonce)

A nonce is a unique message identifier which has not been used before in the protocol.

Subsequently, R is a nonce consisting of a random number, T is a nonce involving a time stamp, and N is a nonce built from a sequence number.

Exchange of symmetric keys with authentication server

Assumption: Both partners have common knowledge of a long term key k , they need a short term transaction key.

$$(1) \quad A \rightarrow B \quad : \quad \{R_A\}^k$$

$$(2) \quad A \leftrightarrow B \quad : \quad \{M\}^{R_A}$$

A possible attack is the replay of messages. Revised protocol:

$$(1) \quad A \rightarrow B \quad : \quad \{R_A, T_A\}^k$$

$$(2) \quad A \leftrightarrow B \quad : \quad \{M\}^{R_A}$$

A variant of this is the challenge-response protocol:

- (1) $A \leftarrow B : N_B$
- (2) $A \rightarrow B : \{R_A, N_B\}^k$
- (3) $A \leftrightarrow B : \{M\}^{R_A}$

Weakness: the key is generated solely by A

Modified version:

- (1) $A \leftarrow B : N_B$
- (2) $A \rightarrow B : \{R_A, N_A, N_B\}^k$
- (3) $A \leftarrow B : \{(R_A, R_B), N_A, N_B\}^k$
- (4) $A \leftrightarrow B : \{M\}^{(R_A, R_B)}$

Agreement of symmetric keys with one-way function:

- (1) $A \rightarrow B : R_A$
- (2) $A \leftrightarrow B : \{M\}^{f(k, R_A)}$

Symmetric keys with authentication server

Needham-Schröder shared-key protocol

Prerequisites: parties A , B , trusted server S ; A and B each have a common secret with S , viz k_{AS} and k_{BS} ; need a common secret between A and B , viz k_{AB} , to exchange messages.

- (1) $A \rightarrow S$: (A, B, N_A)
- (2) $A \leftarrow S$: $\{N_A, B, K_{AB}, \{k_{AB}, A\}^{k_{BS}}\}^{k_{AS}}$
- (3) $A \rightarrow B$: $\{K_{AB}, A\}^{k_{BS}}$
- (4) $A \leftarrow B$: $\{N_B\}^{k_{AB}}$
- (5) $A \rightarrow B$: $\{N_B - 1\}^{k_{AB}}$ “handshake”
- (6) $A \leftrightarrow B$: $\{M\}^{k_{AB}}$

Kerberos key distribution protocol

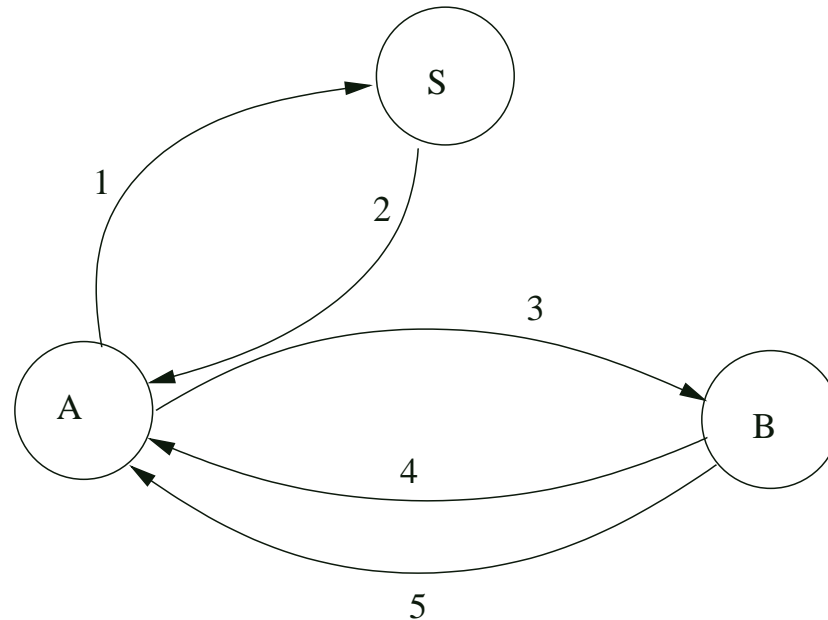
refinement of the above; L is the lifespan (duration of validity) for the key

- (1) $A \rightarrow S : (A, B, N_A)$
- (2) $A \leftarrow S : (\{k_{AB}, A, L\}^{k_{BS}}, \{k_{AB}, N_A, L, B\}^{k_{AS}})$
- (3) $A \rightarrow B : (\{k_{AB}, A, L\}^{k_{BS}}, \{A, T_A\})$
- (4) $A \leftarrow B : \{T_A\}^{k_{AB}}$
- (5) $A \leftrightarrow B : \{M\}^{k_{AB}}$

In step (4), B can check whether A is authentic and T_A is within the lifespan of the key.

Remarks:

- Time stamps requires additional clock synchronization
- the protocol is secure, if modifications are excluded (i.e., if any step fails, it is aborted)



A protocol with asymmetric keys

Assumption: each participant has a private key k_A^d , public key k_A^e .

This must hold also for the server (k_S^d, k_S^e) !

All public keys are registered with the server.

- | | | | | |
|-----|-------------------|---|------------------------|----------------------------------|
| (1) | $A \rightarrow S$ | : | (A, B) | “I want to talk to B ” |
| (2) | $A \leftarrow S$ | : | $\{k_B^e, B\}^{k_S^d}$ | B 's public key, signed by S |
| (3) | $A \rightarrow B$ | : | $\{N_A, A\}^{k_B^e}$ | communication req. w. nonce |
| (4) | $B \rightarrow S$ | : | (B, A) | “ A wants to talk to me” |
| (5) | $B \leftarrow S$ | : | $\{k_A^e, A\}^{k_S^d}$ | A 's public key, signed by S |
| (6) | $B \rightarrow A$ | : | $\{N_A, N_B\}^{k_A^e}$ | “double handshake” |

$$(7) \quad A \rightarrow B : \{N_B, \{k_{AB}^{k_A^d}, A\}^{k_B^e}$$

only A knows N_B

$$(8) \quad A \leftrightarrow B : \{M\}^{k_{AB}}$$

