Key distribution protocols and deployed protocols

Giampaolo Bella

Overview

- Classical key distribution protocols
  - TMN
  - symmetric Needham-Schroeder
  - Denning-Sacco
- Deployed protocols
  - Kerberos IV
  - SSL/TLS

Features of the TMN protocol, 1990

(Tatebayashi-Matsuzaki-Newman)

- Symmetric. Trusted server.
- Aim: Key distribution.
- Agents don't have long-term keys.
- Randomly chosen keys: $K_A, K_B, \ldots$
- Standard encryption function $e(\cdot)$, invertible only by the server.
- Vernam encryption function $v(\cdot, \cdot)$

$$v(m_1, v(m_1, m_2)) = m_2$$

The TMN protocol

$$\begin{array}{lllll}
1. & A & \to & \mathsf{S} & : \; A, \mathsf{S}, B, e(K_A) \\
2. & \mathsf{S} & \to & B & : \; \mathsf{S}, B, A \\
3. & B & \to & \mathsf{S} & : \; B, \mathsf{S}, A, e(K_B) \\
4. & \mathsf{S} & \to & A & : \; \mathsf{S}, A, B, v(K_A, K_B)
\end{array}$$

A extracts K_B from message 4.

The peers should agree on the session key chosen by B.

The protocol suffers a numbers of attacks — Lowe-Roscoe, 1997.

Attack 1 on TMN

1.	C_A	$\rightarrow$	S	:	$A, \mathsf{S}, B, e(K_C)$
2.	S	$\rightarrow$	B	:	S, B, A
3.	B	$\rightarrow$	S	:	$B, \mathsf{S}, A, e(K_B)$
4.	S	$\rightarrow$	C_A	:	$\mathsf{S}, A, B, v(K_C, K_B)$

C impersonating A extracts K_B from message 4.

Failures of both authentication (B believes A is alive) and confidentiality (of K_B).

Attack 2 on TMN

1.	A	$\rightarrow$	S	:	$A, \mathsf{S}, B, e(K_A)$
2.	S	$\rightarrow$	C_B	:	S, B, A
3.	C_B	$\rightarrow$	S	:	$B, \mathsf{S}, A, e(K_C)$
4.	S	$\rightarrow$	A	:	$\mathsf{S}, A, B, v(K_A, K_C)$

Failures of both authentication (A believes B is alive) and confidentiality (C chooses the session key!).

Attack 3 on TMN

$$
\begin{array}{lllll}
1. & C_A & \to & \mathsf{S} & : A, \mathsf{S}, B, e(K_C) \\
2. & \mathsf{S} & \to & B & : \mathsf{S}, B, A \\
3. & B & \to & \mathsf{S} & : B, \mathsf{S}, A, e(K_B) \\
4. & \mathsf{S} & \to & C_A & : \mathsf{S}, A, B, v(K_C, K_B) \\
1'. & A & \to & \mathsf{S} & : A, \mathsf{S}, B, e(K_A) \\
2'. & \mathsf{S} & \to & C_B & : \mathsf{S}, B, A \\
3'. & C_B & \to & \mathsf{S} & : B, \mathsf{S}, A, e(K_B) \\
4'. & \mathsf{S} & \to & A & : \mathsf{S}, A, B, v(K_A, K_B)
\end{array}
$$

Upgrading the protocol

Each agent A has a key Ka shared with the server — it's not K_A!
Spy shouldn't be able to forge $Ka.K_A$ or alike.

$$
\begin{array}{lllll}
1. & A & \to & \mathsf{S} & : A, \mathsf{S}, B, e(Ka.K_A) \\
2. & \mathsf{S} & \to & B & : \mathsf{S}, B, A \\
3. & B & \to & \mathsf{S} & : B, \mathsf{S}, A, e(Kb.K_B) \\
4. & \mathsf{S} & \to & A & : \mathsf{S}, A, B, v(K_A, K_B)
\end{array}
$$

Attack 1 on new TMN

1.	C	$\rightarrow$	S	:	$C, \mathsf{S}, B, e(Kc.K_C)$
2.	S	$\rightarrow$	C_B	:	S, B, C (intercepted)
2′.	C_S	$\rightarrow$	B	:	S, B, A
3.	B	$\rightarrow$	C_S	:	$B, \mathsf{S}, A, e(Kb.K_B)$ (intercepted)
3′.	C_B	$\rightarrow$	S	:	$B, \mathsf{S}, C, e(Kb.K_B)$
4.	S	$\rightarrow$	C	:	$\mathsf{S}, C, B, v(K_C, K_B)$

Interpreting the findings

How serious are these attacks?

1.	A	$\rightarrow$	S	:	$A, \mathsf{S}, B, e(K_A)$
2.	S	$\rightarrow$	B	:	S, B, A
3.	B	$\rightarrow$	S	:	$B, \mathsf{S}, A, e(K_B)$
4.	S	$\rightarrow$	A	:	$\mathsf{S}, A, B, v(K_A, K_B)$

1.	$A \rightarrow B$	: $\{\|Na, A\|\}_{Kb}$
2.	$B \rightarrow A$	: $\{\|Na, Nb\|\}_{Ka}$
3.	$A \rightarrow B$	: $\{\|Nb\|\}_{Kb}$

Fairly easy to spot...

Not designed for active attacker!

Symmetric Needham-Schroeder, 1978

$$
\begin{array}{lll}
1. & A \to \mathsf{S} & : \quad A, B, Na \\
2. & \mathsf{S} \to A & : \quad \{\!|Na, B, Kab, \underbrace{\{\!|Kab, A|\!\}_{Kb}}_{ticket}|\!\}_{Ka} \\
3. & A \to B & : \quad \underbrace{\{\!|Kab, A|\!\}_{Kb}}_{ticket} \\
4. & B \to A & : \quad \{\!|Nb|\!\}_{Kab} \\
5. & A \to B & : \quad \{\!|Nb - 1|\!\}_{Kab}
\end{array}
$$

Authentication OK. Key distribution OK. Accidents?? Cryptanalysis??

Cheating on B

Suppose C gets hold of an old Kab.

$$
\begin{array}{lll}
 & \vdots & \\
3. & C_A \to B & : \quad \underbrace{\{\!|Kab, A|\!\}_{Kb}}_{ticket} \\
4. & B \to C_A & : \quad \{\!|Nb|\!\}_{Kab} \\
5. & C_A \to B & : \quad \{\!|Nb - 1|\!\}_{Kab}
\end{array}
$$

B would believe A is alive and, so, would use Kab.

BAN Kerberos, 1989

(symmetric crypto)

1. $A \rightarrow \mathsf{S} : A, B$
2. $\mathsf{S} \rightarrow A : \{\!|Ts, B, Kab, \{\!|Ts, A, Kab|\!\}_{Kb}|\!\}_{Ka}$
3. $A \rightarrow B : \underbrace{\{\!|Ts, A, Kab|\!\}_{Kb}}_{ticket}, \{\!|A, Ta|\!\}_{Kab}$
4. $B \rightarrow A : \{\!|Ta + 1|\!\}_{Kab}$

The ticket expires!

Denning-Sacco, 1981

They introduce timestamps. (asymmetric crypto)

1. $A \rightarrow \mathsf{S} : A, B$
2. $\mathsf{S} \rightarrow A : C_A, C_B$
3. $A \rightarrow B : C_A, C_B, \{\!|\{\!|K, T|\!\}_{Ka^{-1}}|\!\}_{Kb}$

Does step 3 authenticate A with B?.

An attack — Abadi-Needham, 1996

No! The entire lifetime for T can be exploited (by 2 sessions).

$$
\begin{array}{ll}
1. & A \to \mathsf{S} : A, C \\
1'. & C \to \mathsf{S} : C, B \\
2. & \mathsf{S} \to A : C_A, C_C \\
2'. & \mathsf{S} \to C : C_C, C_B \\
3. & A \to C : C_A, C_C, \{\!|\{\!|K, T|\!\}_{Ka^{-1}}|\!\}_{Kc} \\
3'. & C \to B : C_A, C_B, \{\!|\{\!|K, T|\!\}_{Ka^{-1}}|\!\}_{Kb}
\end{array}
$$

The cipher $\{\!|K, T|\!\}_{Ka^{-1}}$ doesn't state the identity of its intended recipient, which is, instead, inferred.

Fixing the flaw — Abadi-Needham,

Step 3 must be explicit.

$$
\begin{array}{ll}
1. & A \to \mathsf{S} : A, B \\
2. & \mathsf{S} \to A : C_A, C_B \\
3. & A \to B : C_A, C_B, \{\!|\{\!|K, T, B|\!\}_{Ka^{-1}}|\!\}_{Kb}
\end{array}
$$

Checking . . .

$$
\begin{array}{ll}
 & \vdots \\
3. & A \to C : C_A, C_C, \{\!|\{\!|K, T, C|\!\}_{Ka^{-1}}|\!\}_{Kc} \\
3'. & C \to B : C_A, C_B, \{\!|\{\!|K, T, C|\!\}_{Ka^{-1}}|\!\}_{Kb}
\end{array}
$$

Timestamps vs. nonces

Issue of the session key in

– symmetric Needham-Schroeder

$$2. \quad \mathsf{S} \to A \quad : \quad \{\!|Na, B, Kab, \{\!|Kab, A|\!\}_{Kb}|\!\}_{Ka}$$

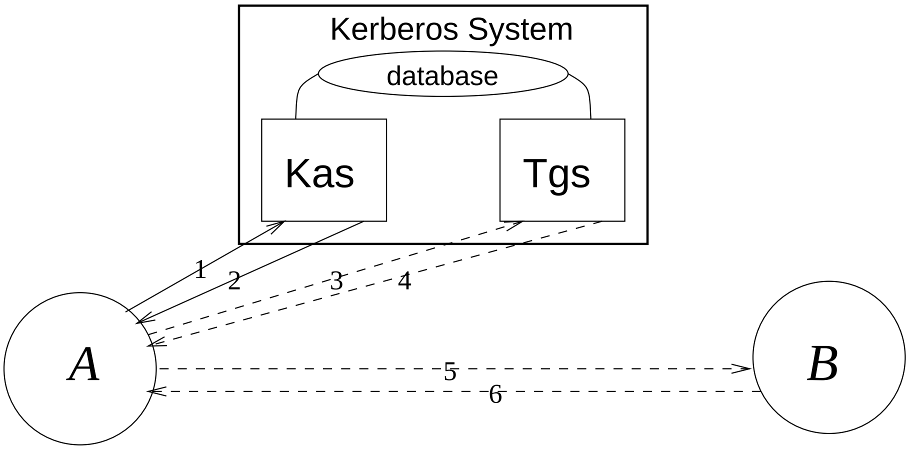
– BAN Kerberos

$$2. \quad \mathsf{S} \to A \quad : \quad \{\!|Ts, B, Kab, \{\!|Ts, A, Kab|\!\}_{Kb}|\!\}_{Ka}$$

- Freshness is analogous.
- Design complexity is higher with nonces.
- Temporal validity not conveyed by nonces.

Deployed Protocols

The Kerberos IV layout, 1989



AUTHENTICATION phase	:	steps 1, 2.
AUTHORISATION phase	:	steps 3, 4.
SERVICE phase	:	steps 5, 6.

Features of Kerberos IV

- Designed for LANs. Second and third phases are optional.
- They are transparent to the user.
- Each phase provides the initiator with credentials for the subsequent phase.

  AUTHENTICATION provides authKey and authTicket.

  AUTHORISATION provides servKey and servTicket.
- Different session keys have different lifetimes.
- An authKey may encrypt several servKeys.

Kerberos IV

AUTHENTICATION

1. $A \rightarrow \mathsf{Kas} \ : \ A, \mathsf{Tgs}, T1$

2. $\mathsf{Kas} \rightarrow A \ : \ \{\!|authK, \mathsf{Tgs}, Ta, \underbrace{\{\!|A, \mathsf{Tgs}, authK, Ta|\!\}_{Ktgs}}_{authTicket}|\!\}_{Ka}$

AUTHORISATION

3. $A \rightarrow \mathsf{Tgs} \ : \ \underbrace{\{\!|A, \mathsf{Tgs}, authK, Ta|\!\}_{Ktgs}}_{authTicket}, \underbrace{\{\!|A, T2|\!\}_{authK}}_{authenticator}, B$

4. $\mathsf{Tgs} \rightarrow A \ : \ \{\!|servK, B, Ts, \underbrace{\{\!|A, B, servK, Ts|\!\}_{Kb}}_{servTicket}|\!\}_{authK}$

SERVICE

5. $A \rightarrow B \ : \ \underbrace{\{\!|A, B, servK, Ts|\!\}_{Kb}}_{servTicket}, \underbrace{\{\!|A, T3|\!\}_{servK}}_{authenticator}$

6. $B \rightarrow A \ : \ \{\!|T3 + 1|\!\}_{servK}$

Goals achieved by Kerberos IV

Regularity. Long-term keys are never sent on the network.

Unicity.

1. If Kas works correctly, each authKey enjoys unicity.
2. If Tgs works correctly, each servKey enjoys unicity.

Confidentiality.

1. AuthKeys or servKeys are confidential if issued for agents whose shared keys are not compromised.
2. ServKeys are subject to an attack from a realistic accident.

All proven formally!

Goals achieved by Kerberos IV

Authentication. Mutual non-injective agreement on the session key holds.

Key distribution. The peers agree on the session key.

The protocol conforms to the principle of goal availability in respect to all goals but *one* . . .

All proofs mechanised on a theorem prover.

The attack on servKeys

Suppose the spy gets hold of an authKey that has expired.
While A is killed, the spy learns $servK$ from

$$4.\quad \text{Tgs} \rightarrow A \;:\; \{\!|servK, B, Ts, \underbrace{\{\!|A, B, servK, Ts|\!\}_{Kb}}_{servTicket}|\!\}_{authK}$$

and executes . . .

SERVICE

$$5.\quad C_A \rightarrow B \;:\; \underbrace{\{\!|A, B, servK, Ts|\!\}_{Kb}}_{servTicket}, \underbrace{\{\!|A, T3'|\!\}_{servK}}_{authenticator}$$

$$6.\quad B \rightarrow C_A \;:\; \{\!|T3' + 1|\!\}_{servK}$$

The spy can cheat on B for the lifetime of Ts!

Fixing the attack

The moment that an authKey expires, then all servKeys associated with it must expire too.
Step 4 creates such associations.

$$4. \quad \text{Tgs} \rightarrow \; A \; : \; \{\!|servK, B, Ts, \underbrace{\{\!|A, B, servK, Ts|\!\}_{Kb}}_{servTicket}|\!\}_{authK}$$

Refine Tgs's functioning by the check

$$currentTime + \text{servKlife} \leq Ta + \text{authKlife}$$

Features of TLS, 1999

(Transport Layer Security)

- Descendant of SSL 3.0, which was subject to a *cipher-suite rollback attack*.
- Widespread use, considered the highest security standard by most URLs.
- Available protocol descriptions are vague about the sought goals but exaggerate the discussion of details (common problem).
- Presenting TLS is difficult!

The TLS message components

A	*client*
B	*server*
Sid	*session identifier*
Na, Nb	*client or server's nonce*
P_a, P_b	*client or server's crypto preferences*
cert(A, Ka), *cert*(B, Kb)	*client or server's certificates (sealed by trusted authority's private key)*
PMS	*pre-master-secret*
M	*master secret*

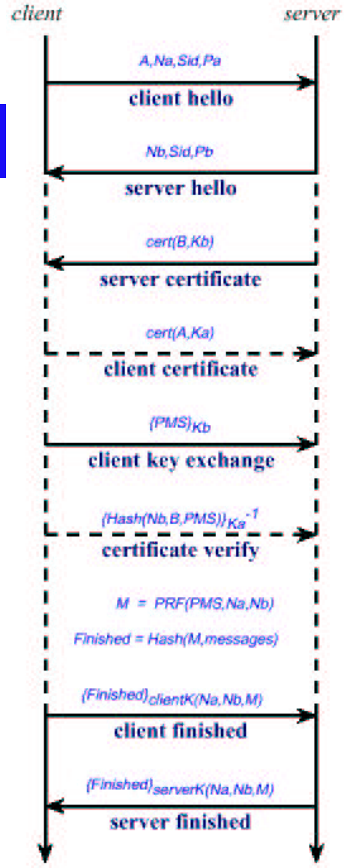
The TLS handshake protocol (Paulson's version)

A	$\rightarrow$	B	:	A, Na, S_{id}, P_a	**client hello**
B	$\rightarrow$	A	:	Nb, S_{id}, P_b	**server hello**
B	$\rightarrow$	A	:	$cert(B, Kb)$	**server certificate**
A	$\rightarrow$	B	:	$cert(A, Ka)$	**client certificate**
A	$\rightarrow$	B	:	$\{\!\|PMS\|\!\}_{Kb}$	**client key exchange**
A	$\rightarrow$	B	:	$\{\!\|Hash(Nb, B, PMS)\|\!\}_{Ka^{-1}}$	**certificate verify**
A	$\rightarrow$	B	:	$\{\!\|Finished\|\!\}_{clientK(Na,Nb,M)}$	**client finished**
B	$\rightarrow$	A	:	$\{\!\|Finished\|\!\}_{serverK(Na,Nb,M)}$	**client finished**

$$(M = PRF(PMS, Na, Nb), \quad Finished = Hash(M, all\_messages))$$

Some portions of TLS are optional

You may think of A as a *client* and of B as a *server*.



What is missing to Paulson's version

1. Field widths, choice of the cryptographic algorithms, failure messages.
2. Various certification authorities: various certificate forms.
3. B's **certificate request**.
4. Computing *PMS* via a Diffie-Hellman exchange.
5. All previous handshake messages hashed in **certificate verify**.
6. All previous handshake messages hashed in *Finished* **certificate verify** can be intercepted.
7. MACs (because encryption is perfect).

The goals of TLS

“Early” authentication (of the client with the server). If **certificate verify** is in the traffic, then it originated with the client.

Confidentiality. If client and server’s long-term keys are confidential, then so are client and server’s session keys, *PMS* and M.

Authentication.

1. If client receives **server finished**, then this originated with the server.
2. If server receives **client finished**, then this originated with the client *provided that there was early authentication*.

The client may remain unauthenticated. Formal proofs are difficult!

Conclusions

- Protocol specifications are difficult to fully understand.
  1. Spot underlying assumptions (e.g. on spy, on session keys).
  2. Skip implementative details.
  3. Bring goals to a focus.
- Goal failures (*design errors*) even with perfect cryptography.
- Investigating goal availability may pinpoint goal failures (e.g. Kerberos IV).
- Internet transactions under TLS are reasonably secure but we can do better . . .