Inductive Analysis of the Internet Protocol TLS

Lawrence C. Paulson

Computer Laboratory

University of Cambridge

TLS: An Internet Protocol

- to protect data between Web browsers and servers
- RSA and symmetric-key encryption (among others)
- random-number generator for negotiating secrets
- resumption of old sessions with new keys
- also known as “SSL 3.1”

Hello Messages

client hello $A \to B : A, Na, Sid, Pa$

server hello $B \to A : Nb, Sid, Pb$

resumption? go straight to Finished messages

server certificate $B \to A :$ certificate(B, Kb)

$Sid =$ session Id (for resumption) $Pa, Pb =$ crypto preferences

Client Key Exchange Messages

client certificate*	$A \to B$: certificate(A, Ka)
client key exchange	$A \to B$: $\{\!\|PMS\|\!\}_{Kb}$
certificate verify*	$A \to B$: $\{\!\|\text{Hash} \ldots\|\!\}_{Ka^{-1}}$

* omit for anonymous session

PMS = pre-master-secret

Diffie-Hellman exchange also possible

Finished Messages

$M = PRF(PMS, Na, Nb)$ master-secret

$Finished =$ hash of previous messages

client finished $A \to B : \{|Finished|\}_{\mathsf{clientK}(Na,Nb,M)}$

server finished $B \to A : \{|Finished|\}_{\mathsf{serverK}(Na,Nb,M)}$

clientK, serverK make fresh session keys

Each party checks the other's

An Inductive Approach to Proving Protocols

- Work in higher-order logic
- Inductively model traces of agent actions
- Include an active attacker, compromised & careless agents
- No finite-state assumptions
- Prove results by induction
- Mechanized using Isabelle/HOL

Message Types

$$
\begin{array}{llcll}
msg & X, \ldots & = & \textsf{Agent } A & \\
 & & \mid & \textsf{Nonce } N & \text{non-guessable number} \\
 & & \mid & \textsf{Number } N & \text{guessable number} \\
 & & \mid & \textsf{Key } K & \\
 & & \mid & \textsf{Hash } X & \\
 & & \mid & \{\!|X_1, \ldots, X_n|\!\} & \text{concatenation} \\
 & & \mid & \textsf{Crypt } K\, X & \text{strong encryption}
\end{array}
$$

Inductively Defining the Protocol: Hello

client hello. If Na is fresh in the trace, may add

$$\text{Says}\ A\ B\ \{\!|A, Na, Sid, Pa|\!\}$$

server hello. If the trace has $\text{Says}\ A'\ B\ \{\!|A, Na, Sid, Pa|\!\}$ and Nb is fresh, may add

$$\text{Says}\ B\ A\ \{\!|Nb, Sid, Pb|\!\}$$

Defining the Protocol: Client Key Exchange

certificate. May add Says $B\ A\ (\mathsf{certificate}(B, \mathsf{pubK}\ B))$ to a trace

client key exchange. If the trace contains the events

$$\mathsf{Says}\ B'\ A\ \{\!|Nb, Sid, Pb|\!\}$$

$$\mathsf{Says}\ B''\ A\ (\mathsf{certificate}(B, Kb))$$

and PMS is fresh, may add

$$\mathsf{Says}\ A\ B\ (\mathsf{Crypt}\ Kb\ PMS)$$

Modelling Attacks and Accidents

Fake. If X can be forged in the trace, may add Says Spy B X

SpyKeys. If the spy has $\{\!|Na, Nb, M|\!\}$ then he has

$$PRF(M, Na, Nb) \quad \text{and} \quad \mathsf{sessionK}(Na, Nb, M)$$

Oops. Anybody who uses a session key may give it to the spy.

Security Goals Proved

- The pre-master-secret remains secret (assuming honest peers)
- The master-secret remains secret
- Certificate verify guarantees that the client is present
- session keys remain secret (unless given away)
- A message encrypted with peer's session key came from him

Lemmas Proved Along the Way

- Protocol steps don’t reveal private keys
- All certificates are valid (too perfect?)
- A fresh PMS yields fresh session keys
- Compromise of a session key doesn’t compromise any PMS (hard to prove)

Related Work

Wagner and Schneier's analysis of SSL 3.0:

- weaknesses in abstract protocol (fixed in TLS)
- discussion of cryptanalysis

Dietrich's thesis:

- investigated anonymous connections against an eavesdropper using NCP belief logic

Mitchell et al.: simple model-checking experiments

Comments on TLS

- Strengthen client key exchange to

$$A \to B : \{\!|A, PMS|\!\}_{Kb} \ ?$$

- Explicitness: beware of hashing everything but the kitchen sink
- Make the abstract message exchange part of every protocol spec

Conclusions

- 6 weeks effort; 8 minutes cpu time (model-checking: 8 hours)
- mundane proofs but interesting model
- Can model key negotiation
- Non-determinism is no obstacle
- Realistic protocols can now be analyzed—abstractly, at least