Security Protocols and Their Correctness

Lawrence C. Paulson

Computer Laboratory

University of Cambridge

Can Cryptography Make Networks Secure?

Goals:

- Authenticity: who sent it?
- Secrecy: who can receive it?

Threats:

- Active attacker
- Careless & compromised agents ...NO code-breaking

Some Notation

A, B	agent names (Alice, Bob)
Na	nonce chosen by Alice (a random number)
Ka	Alice's public key
$\{X\}_{Ka}$	message encrypted using Ka

- anybody can encrypt
- only Alice can recover X

The Needham-Schroeder Protocol

$$1. \quad A \to B : \{\!|Na, A|\!\}_{Kb}$$

Alice sends Bob an encrypted nonce

$$2. \quad B \to A : \{\!|Na, Nb|\!\}_{Ka}$$

Bob returns Na with a nonce of his own

$$3. \quad A \to B : \{\!|Nb|\!\}_{Kb}$$

Alice returns Bob's nonce

What Does Needham-Schroeder Accomplish?

Only Bob could recover Na

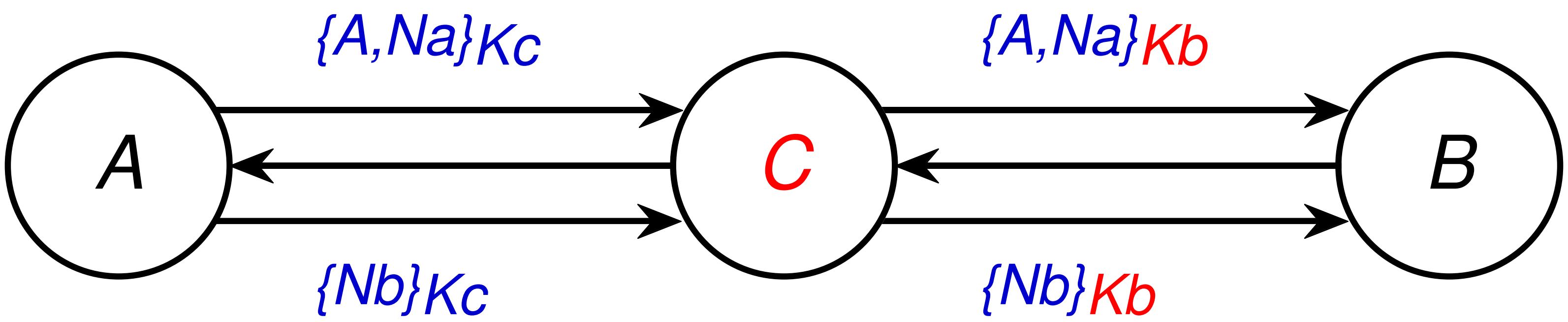
Only Alice could recover Nb

- Therefore Alice and Bob are present now

But are the nonces secret?

A Middle-Person Attack

Villain Charlie can masquerade as Alice to Bob



Lowe's Attack in Detail

$$
\begin{array}{ll}
1. & A \to C : \{\!|Na, A|\!\}_{Kc} \\
1'. & C(A) \to B : \{\!|Na, A|\!\}_{Kb} \\
2'. & B \to C(A) : \{\!|Na, Nb|\!\}_{Ka} \\
2. & C \to A : \{\!|Na, Nb|\!\}_{Ka} \\
3. & A \to C : \{\!|Nb|\!\}_{Kc} \\
3'. & C(A) \to B : \{\!|Nb|\!\}_{Kb}
\end{array}
$$

Can protocols be verified?

Verification Method I: Authentication Logics

BAN logic: Burrows, Abadi, Needham (1989)

Models agent beliefs:

Nonce N is fresh   Key Kab is good

Agent S can be trusted

- Allows short, abstract proofs but misses many flaws

Verification Method II: State Enumeration

Specialized tools (Meadows, Millen)

General model-checkers (Lowe)

Model protocol as a finite-state system

- Automatically finds attacks but requires strong assumptions

Can we use formal proof?

Inductive Protocol Verification

- Traces of events: A sends X to B
- Operational model of agents
- Algebraic theory of messages (derived)
- A general attacker
- Proofs mechanized using Isabelle/HOL

Sets of Messages

parts H: the components of H

$$\text{Crypt}\,K\,X \mapsto X$$

analz H: the accessible components of H

$$\text{Crypt}\,K\,X,\; K^{-1} \mapsto X$$

synth H: messages that can be made from H

$$X,\; K \mapsto \text{Crypt}\,K\,X$$

Defined inductively

Some Algebraic Laws

$$\text{parts}(\text{parts}\ H) = \text{parts}\ H$$

$$\text{parts}(\text{analz}\ H) = \text{parts}\ H$$

$$\text{analz}(\text{synth}\ H) = \text{analz}\ H \cup \text{synth}\ H$$

$$\text{synth}(\text{analz}\ H) = ??$$

Keep the 3 notions separate

Model as set transformers

Part of a Protocol Specification

If a trace has the event

$$\text{Says } A' \; B \; (\text{Crypt}(\text{pubK } B) \{\!|Na, A|\!\})$$

and Nb is fresh, then may add the event

$$\text{Says } B \; A \; (\text{Crypt}(\text{pubK } A) \{\!|Na, Nb|\!\})$$

B doesn't know the true sender (shown as A')

Modelling Attacks and Accidents

Fake. If $X \in \mathsf{synth}(\mathsf{analz}(\mathsf{spies}\, evs))$

may add the event

$\mathsf{Says\ Spy}\ B\ X$

Can also model accidents: giving secrets away

Does one compromise lead to others?

Facts that Can be Proved

- Secret keys are never lost
- Nonces uniquely identify their message of origin
- Nonces stay secret (under certain conditions!)

Proved by induction, simplification & classical reasoning

Simplification of analz: case analysis, big formulas

Final Remarks

- A dozen protocols analyzed:

  (Otway-Rees, Yahalom, Needham-Schroeder, . . .)
- TLS: an Internet protocol
- 2–9 minutes CPU time per protocol
- few hours or days human time per protocol
- a good complement to model-checking