Red Cryptography
(Formal Analysis of Cryptographic Protocols)

Jonathan Herzog
9 March 2001
Introduction

- Rogoway described two “worlds” of cryptographic protocol analysis
  - Blue: computational view
  - Red: formal methods view
- Blue world is probably well-known in CIS
- Red world may be less so
- In this talk: introduction to the formal methods approach

- Goals:
  - No new material
  - Give background on class of problems
  - Stimulate interest
Overview of Talk

- Scope of problem: abstracted authentication and mission protocols
- Formal methods approaches (at least one)
  - Model checkers
  - Specialized logics
  - Theorem provers
- Open problems
Protocols

- More limited definition than usually used
- Sequence of messages between small number (2 or 3) principals
  - No conditionals (except to abort)
- Abstract cryptographic primitives (encryption, signatures)
- Achieve authentication and/or key transmission
Needham–Schroeder Public Key Protocol

1. $A \rightarrow B$: $\{N_a \ A\}^{K_B}$
2. $B \rightarrow A$: $\{N_a \ N_b\}^{K_A}$
3. $A \rightarrow B$: $\{N_b\}^{K_B}$

- First published in 1978
- $A$, $B$ assumed to know each other’s public
- $N_a$, $N_b$ are “fresh” nonces
- $K_A$, $K_B$: public keys
- Designed to provide mutual authentication and secrecy of $N_a$, $N_b$
Message Algebra

- Messages are elements of an “algebra” $\mathcal{A}$
- 2 disjoint sets of atomic messages:
  - Texts ($\mathcal{T}$)
  - Keys ($\mathcal{K}$)
- 2 operators:
  - $\text{enc} : \mathcal{K} \times \mathcal{A} \rightarrow \mathcal{A}$ (Range: $\mathcal{E}$)
  - $\text{concat} : \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$ (Range: $\mathcal{C}$)
Message Algebra (cont.)

- Message algebra is “free”
  - Unique representation of terms
  - *Exactly* one way to build elements from atomics, operations
- \( K, T, E, C \) mutually disjoint
- For all \( M_1, M_2, M_3, M_4 \in A, k_1, k_2 \in K, T \in T \)
  - \( M_1 M_2 \neq M_3 M_4 \), unless \( M_1 = M_3, M_2 = M_4 \)
  - \( \{M_1\}_{k_1} \neq \{M_2\}_{k_2} \) unless \( M_1 = M_2, k_1 = k_2 \)

- Justification: looking for flaws that do not depend on properties of encryption scheme
Adversary

- Adversary has complete control over the network
  - Can intercept, delete, delay, replay messages
- Unbounded time, but limited in available cryptographic operations
  - Separate, concatenate known messages
  - Decrypt with known key
  - Encrypt with known key
  - Sign with known key
  - Create fresh values, keys
  - Use public values, keys
- May be regular participant, also
  - Presumed to start knowing some set of keys
Needham-Schroeder Goals

• *Initiator*, *Responder* are *roles* instantiated here as *A* and *B*

• For every *Initiator*, there should be a corresponding *Responder* that agrees on the values in question

• For every *Responder*, there should be a corresponding *Initiator* that agrees on the values in question
Needham-Schroeder: Flawed!

\[ A \rightarrow \{N_1 A\}_{K_M} \rightarrow M \]

\[ \{N_2\}_{K_M} \rightarrow M \rightarrow \{N_2\}_{K_B} \]

\[ \{N_1 N_2\}_{K_A} \]

- Due to Gavin Lowe (1995)
- Note that flaw exists independently of underlying
Formal Methods

- One view of problem:
  - Communicating sequential processes
  - Communicating through malicious (noisy) channels
  - High level of abstraction
  - Goals expressible as safety properties
- Standard formal methods problem
- Attacked using standard formal methods tools
Model Checking

- Describe system as state machine
  - Security properties can be described as statements about executions
  - Algorithms, tools exist that exhaustively search executions to verify properties.
- Regular participants simple to describe as state machine
- Modeling the adversary more complex
Model Checking: Adversary

- State of adversary described by set of “known” terms $I$
- Presumed to start with some initial set $I$
- If $M$ is sent by regular participant, can move into state where $I' = I \cup \{M\}$
- If $(M_1, M_2) \in I$, then can move into state where $\{M_1\}, I' = I \cup \{M_2\}$
- If $\{M\}_k, k^{-1} \in I$, can move into state where $I' = I \cup \{M\}$
- If $M, k \in I$, then can move into state where $I' = I \cup \{\{M\}_k\}$
- Can send any message in set of known terms to any regular participant
Model Checking: Security Conditions

- Security conditions can be expressed as safety properties

$$\{N_1 A\}_B$$

$$\{N_1 N_2\}_A$$

$$\{N_2\}_B$$

- System should never reach state where \(N_1, N_2\) in adversary set
- \(Init[A, B, N_1, N_2].3 \Rightarrow Respond[B, A, N_1, N_2].2\)
- \(Respond[B, A, N_1, N_2].3 \Rightarrow Init[A, B, N_1, N_2].3\)
Model Checking: Pros and Cons

- **Pros**
  - Conceptually simple
  - Exhaustive search of all possible adversary tactics

- **Cons**
  - State space explosion
    - Infinite number of adversary states
    - Some attacks use multiple initiators, responders
  - Impossible (in general) to catch all possible attacks
Needham–Schroeder Lowe Protocol

- Proven correct
  1. If an attack exists on any system, an attack exists on a system with one initiator, one responder (pencil and paper)
  2. No attacks exist on that system (model checker)
- Statement (1) shown for restricted class of protocols
- Open problem: similar result for larger class?
BAN Logic (1989)

- Named after Burrows, Abadi, Needham
- "Many sorted modal logic" of belief
- Turn protocol messages into logical statements
- Apply inference rules
- Arrive at desired goals
BAN Logic: Operators

- $P \models X$  \hspace{1cm} P \text{ believes } X
- $P \mathrel{\triangleleft} X$  \hspace{1cm} P \text{ sees } X
- $P \xrightarrow{K} Q$  \hspace{1cm} P, Q \text{ can use shared key } K \text{ to communicate}
- $P \Rightarrow X$  \hspace{1cm} P \text{ has jurisdiction over } X
- $P \sim X$  \hspace{1cm} P \text{ once said } X
- $\#(X)$  \hspace{1cm} X \text{ is fresh}
BAN Logic: Deductions

\[
\begin{align*}
P \models Q \Rightarrow X & \quad P \models Q \models X \\
\hline
P \models X
\end{align*}
\]

\[
\begin{align*}
P \models \#(X) & \quad P \models Q \leadsto X \\
\hline
P \models Q \models X
\end{align*}
\]

\[
\begin{align*}
P \models Q \xleftarrow{K} P & \quad P \triangleleft \{X\}_K \\
\hline
P \models Q \leadsto X
\end{align*}
\]
Otway-Rees Protocol


1. $A \rightarrow B$: $M A B \{N_a M A B\} K_{as}$
2. $B \rightarrow S$: $M A B \{N_a M A B\} K_{as} N_b \{M A B\} K_{bs}$
3. $S \rightarrow B$: $M \{N_a K_{ab}\} K_{as} \{N_b K_{ab}\} K_{bs}$
4. $B \rightarrow A$: $M \{N_a K_{ab}\} K_{as}$

- $S$: Distinguished session key server
- $K_{as}, K_{bs}$: Long term shared, symmetric keys
- $M$: Public session identifier
BAN Logic: Idealization

- This:
  
  \[ A \rightarrow B : \quad M \ A \ B \ \{N_a \ M \ A \ B\}\_K_{as} \]
  
  \[ B \rightarrow S : \quad M \ A \ B \ \{N_a \ M \ A \ B\}\_K_{as} \ N_b \ \{M \ A \ B\}\_K_{bs} \]
  
  \[ S \rightarrow B : \quad M \ \{N_a \ K_{ab}\}\_K_{as} \ \{N_b \ K_{ab}\}\_K_{bs} \]
  
  \[ B \rightarrow A : \quad M \ \{N_a \ K_{ab}\}\_K_{as} \]

- Becomes:
  
  \[ A \rightarrow B : \quad \{M \ A \ B \ N_a\}\_K_{as} \]
  
  \[ B \rightarrow S : \quad \{M \ A \ B \ N_a\}\_K_{as} \ N_b \ \{M \ A \ B\}\_K_{bs} \]
  
  \[ S \rightarrow B : \quad \{N_a, (A \overset{K_{ab}}{\leftrightarrow} B), (B \rightsquigarrow M \ A \ B)\}\_K_{as} \]
  
  \[ \quad \{N_b, (A \overset{K_{ab}}{\leftrightarrow} B), (A \rightsquigarrow M \ A \ B)\}\_K_{bs} \]
  
  \[ B \rightarrow A : \quad \{N_a, (A \overset{K_{ab}}{\leftrightarrow} B), (B \rightsquigarrow M \ A \ B)\}\_K_{as} \]
BAN Logic: Starting Assumptions

\[
\begin{align*}
A & \models A \xleftarrow{K_{as}} S \\
A & \models (S \Rightarrow A \xleftarrow{K_{ab}} B) \\
A & \models (S \Rightarrow (B \rightsquigarrow X)) \\
A & \models \#(N_a) \\
A & \models \#(N_b) \\
B & \models B \xleftarrow{K_{bs}} S \\
B & \models (S \Rightarrow A \xleftarrow{K_{ab}} B) \\
B & \models (S \Rightarrow (A \rightsquigarrow X)) \\
B & \models \#(N_b)
\end{align*}
\]
BAN Logic: Conclusions

\[ A \models A \xleftrightarrow{K_{ab}} B \]
\[ A \models B \models (M A B) \]
\[ B \models A \xrightarrow{K_{ab}} B \]
\[ B \models A \leadsto (M A B) \]
BAN Logic: Flawed!

- Assume $C$ has $\{M'\ C\ B\}_K_{bs}$ from previous run

\[
\begin{align*}
C(A) & \rightarrow B : \quad M\ A\ B \ \{N_c \ M' \ C \ B\}_K_{cs} \\
B & \rightarrow C(S) : \quad M\ A\ B \ \{N_c \ M\ A\ B\}_K_{cs} \ N_b \ \{M\ A\ B\} \\
C & \rightarrow S : \quad M'\ C\ B \ \{N_c \ M' \ C \ B\}_K_{bs} \ N_b \ \{M'\ C\ B\} \\
S & \rightarrow C(B) : \quad M' \ \{N_c \ K_{cb}\}_K_{as} \ \{N_b \ K_{cb}\}_K_{bs} \\
C(S) & \rightarrow B : \quad M \ \{N_c \ K_{cb}\}_K_{cs} \ \{N_b \ K_{cb}\}_K_{bs} \\
B & \rightarrow C(A) : \quad M \ \{N_c \ K_{cb}\}_K_{cs}
\end{align*}
\]
BAN Logic: Source of Flaws

- Idealization process translates informal to formal
  - Cannot easily be done formally
  - Informal idealization as fallible as human judgment

(In specification) \( \{ N_b, K_{ab} \}^{K_{bs}} \)

(Idealized as) \( \{ N_b, (A \xleftarrow{K_{ab}} B), (A \leadsto M \, A \, B) \}^{K_{bs}} \)

(Should be) \( \{ N_b, (A \xleftarrow{K_{ab}} B), (A \leadsto M' \, A \, B) \}^{K_{bs}} \)
BAN Logic: Pros and Cons

● Pros
  – Relatively simple
  – Catches *most* errors
  – Usually decidable
    ● Can often be automated efficiently
    ● Seconds to generate proof

● Cons
  – Idealization process a source of errors
  – Semantics difficult
  – No concept of confidentiality
  – Assumes replay protection
Theorem Provers

- For this talk: Paulson (1998)
- Heavy use of theorem prover (Isabelle)
  - Proof checker
    - Requires every step of a proof to be spelled out and verified
    - Can build up lemmas for use in bigger proofs
  - Proof automator
    - Can automatically perform some proofs
    - Can automate large parts of others
    - Often requires some human guidance
Specifying the Protocol

- **Create** (disjoint) sets of abstract data types
  - Agents, Nonces, Numbers
  - Keys
  - Encryptions \((\text{Crypt } K \ X)\)
  - Concatenations \(\langle X, X' \rangle\)

- **Create** events
  - Says A B X
  - Notes A X
Specifying the Protocol (cont.)

- Model protocol runs as *traces*
  - Finite sequences of events
- Valid traces defined inductively
  - $[]$ is a trace
  - Multiple rules of the form:
    
    “If $x$ is a valid trace satisfying $P(x)$, then $e\#x$ is a valid trace”
Honest Participants: Otway–Rees

\[ A \rightarrow B : \ M A B \{\{N_a M A B\}\} K_{as} \]

- If \( ev \) is a trace, \( N_a \) a fresh nonce, \( A \neq B \) and \( B \neq S \), then
  \[
  (\text{Says} \ A \ B \langle M A B \{\{N_a A B\}\} K_{as}\rangle) \parallel ev
  \]
  is also a valid trace

\[ B \rightarrow S : \ M A B \{\{N_a M A B\}\} K_{as} N_b \{\{M A B\}\} K_{bs} \]

- If \( ev \) is a trace containing \( (\text{Says} \ A' \ B \langle M A B X \rangle) \) fresh, and \( B \neq S \), then
  \[
  \text{Says} \ B \ S \langle M A B X N_b \{\{M A B\}\} K_{bs}\rangle \parallel ev
  \]
  is also a valid trace
Modeling the Adversary

- Need some additional operators
  - analz $H$ is the set of terms the adversary can learn from $H$:
    $H \subseteq \text{analz } H$
    $\langle X, Y \rangle \in \text{analz } H \Rightarrow X \in \text{analz } H \land Y \in \text{analz } H$
    $\{X\}_K \in \text{analz } H \land K^{-1} \in \text{analz } H \Rightarrow X \in \text{analz } H$
  - synth $H$ is the set of what the adversary can make from $H$:
    $X \in \text{synth } H \land Y \in \text{synth } H \Rightarrow \langle X, Y \rangle \in \text{synth } H$
    $X \in \text{synth } H \land K \in \text{synth } H \Rightarrow \{X\}_K \in \text{synth } H$
Modeling the Adversary (cont).

- Let $ev$ be a valid trace. Let spies $ev$ contain
  - All messages from all Says events in $ev$
  - Adversary’s initial state ($advInit$)
  - Long term keys of agents in $bad$
  - Any messages in Notes $A \times X$ events in $ev$, where $A \in bad$

- Then if $X \in \text{synth(analz (spies ev))}$, then
  
  Says Spy B $X \# ev$

  is also a valid trace.
Theorem Prover Use

- Give to theorem prover:
  - Data types,
  - Operations (definitions, laws)
  - Trace extension rules for honest participants
  - Trace extension rules for adversary
  - 110 intermediate lemmas regarding operations

- Get from theorem prover
  - Environment in which to prove security properties
  - Assistance in doing so
Security Goals

- **Secrecy of session keys**: For every valid trace $ev$, if
  \[ \text{Says } S \ B \ \langle M \ A \ B \ \{ \{ N_a \ K \} \}_{K_{as}} \ \{ \{ N_b \ K \} \}_{K_{bs}} \rangle \in ev \]
  then $K \not\in \text{analz (spies } ev)$

- **Authentication condition**: For every valid trace $ev$,
  \[ \text{Says } A \ B \ \langle M \ A \ B \ \{ \{ N_a \ \} \}_{K_{as}} \ \{ \{ N_{b} \ \} \}_{K_{bs}} \rangle \in ev \]
  and
  \[ \text{Says } B' \ A \ \langle M \ \{ \{ N_a \ \} \}_{K_{as}} \rangle \in ev \]
  then
  \[ \text{Says } S \ B'' \ \langle M \ \{ \{ N_a \ \} \}_{K_{as}} \ \{ \{ N^{'}_b \ K \} \}_{K_{bs}} \rangle \in ev \]
Theorem Provers: Pros and Cons

- **Pros:**
  - Finds all errors
  - High degree of certainty

- **Cons:**
  - Difficult!
    - Theorem provers hard to use
    - Weeks to write/debug specification
    - Hours to verify proofs
  - Proofs very often give no intuition
  - Better than pencil and paper?

- Next time: Strand Space method
Open Problems

- Non-free algebras
  - Exclusive-or
  - Exponentiation (Diffie–Hellman)
- Unifying with blue world
  - Specifying/weakening assumptions on underlying primitives
  - Incorporating probabilistic reasoning
- Minimal systems that contain attacks
- Denial of service