Red Cryptography (Formal Analysis of Cryptographic Protoc

Jonathan Herzog

9 March 2001

Introduction

- Rogoway described two "worlds" of cryptographic 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 a
- 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 o
  cipals
  - No conditionals (except to abort)
- Abstract cryptographic primitives (encryption, sig
- Achieve authentication and/or key transmission

+

Needham–Schroeder Public Key Protoco

1. $A \longrightarrow B$**:** $\{\!|N_a\, A|\!\}_{K_B}$

2. $B \longrightarrow A$**:** $\{\!|N_a\, N_b|\!\}_{K_A}$

3. $A \longrightarrow B$**:** $\{\!|N_b|\!\}_{K_B}$

- **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 s** 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:**
  - **enc** : $\mathcal{K} \times \mathcal{A} \rightarrow \mathcal{A}$ (**Range:** $\mathcal{E}$)
  - **concat** : $\mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$ (**Range:** $\mathcal{C}$)

Message Algebra (cont.)

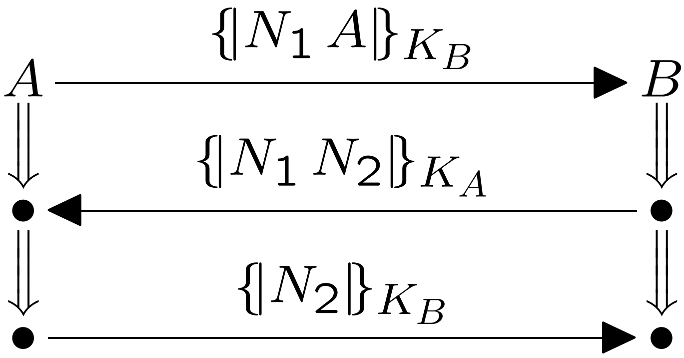
- Message algebra is "free"
  - Unique representation of terms
  - *Exactly* one way to build elements from aton ations
- $\mathcal{K}$, $\mathcal{T}$, $\mathcal{E}$, $\mathcal{C}$ mutually disjoint
- For all M_1, M_2, M_3, $M_4 \in \mathcal{A}$, k_1, $k_2 \in \mathcal{K}$, $T \in \mathcal{T}$
  - $M_1 M_2 \neq M_3 M_4$, 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 erties of encryption scheme

Adversary

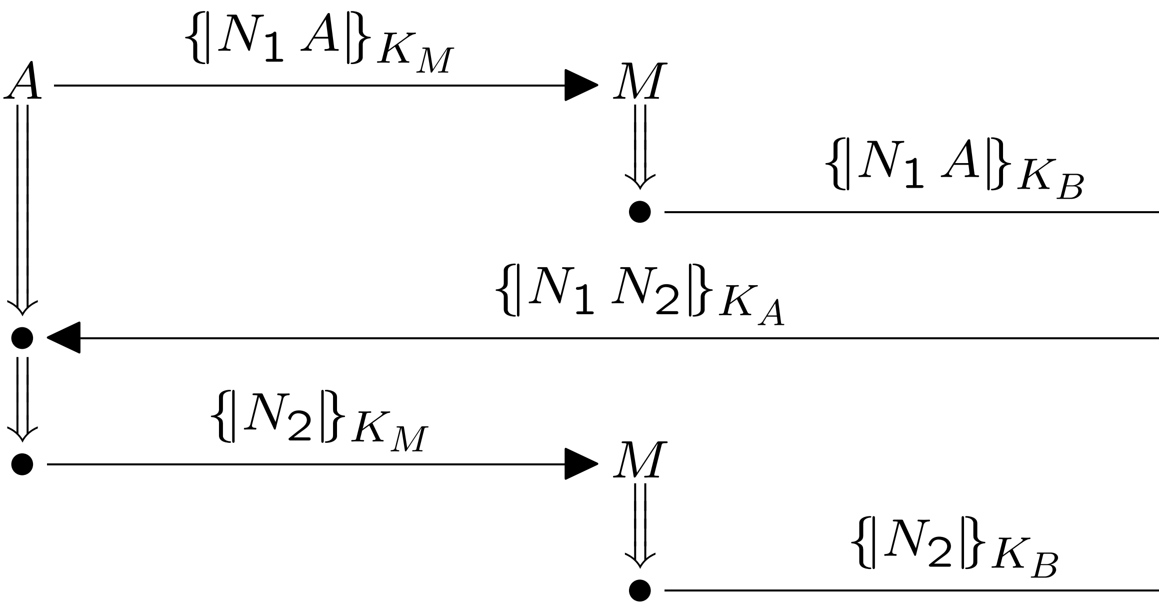
- Adversary has complete control over the network
  - Can intercept, delete, delay, replay messages
- Unbounded time, but limited in available cryptog erations
  - Separate, concatenate known messages
  - Decrypt with known key
  - Encrypt with known key
  - Sign with known key
  - Create fresh values, 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 her
  and B
- For every *Initiator*, there should be a correspor
  sponder that agrees on the values in question
- For every *Responder*, there should be a corre
  Initiator that agrees on the values in question

Needham-Schroeder: Flawed!



- 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) cha
  - High level of abstraction
  - Goals expressible as safety properties
- Standard formal methods problem
- Attacked using standard formal methods tools

Model Checking

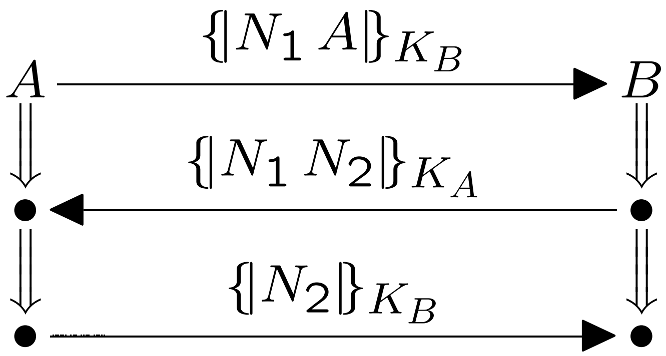
- Describe system as state machine
  - Security properties can be described as stateme
    executions
  - Algorithms, tools exist that exhaustively searc
    executions to verify properties.
- Regular participants simple to describe as state m
- Modeling the adversary more complex

Model Checking: Adversary

- State of adversary described by set of “known” te
- Presumed to start with some initial set
- If M is sent by regular participant, can move into st
  $I' = I \cup \{M\}$
- 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 $I' =$
- If $M, k \in I$, then can move into state where $I' = I$
- Can send any message in set of known terms to a
  participant

Model Checking: Security Conditions

- Security conditions can be expressed as safety pro



- System should never reach state where N_1, N_2 in
  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 tac
- Cons
  - State space explosion
    - Infinite number of adversary states
    - Some attacks use multiple initiators, respo
  - Impossible (in general) to catch all possible a

Needham–Schroeder Lowe Protocol

$$
\begin{tikzcd}
A \arrow[r, "\{\!|N_1\,A|\!\}_{K_B}"] \arrow[d, Rightarrow] & B \arrow[d, Rightarrow] \\
\bullet \arrow[d, Rightarrow] & \bullet \arrow[l, "\{\!|N_1\,N_2\,B|\!\}_{K_A}"'] \arrow[d, Rightarrow] \\
\bullet \arrow[r, "\{\!|N_2|\!\}_{K_B}"] & \bullet
\end{tikzcd}
$$

- **Proven correct**
  1. **If an attack exists on any system, an attack system with one initiator, one responder (pene per)**
  2. **No attacks exist on that system (model chec**
- **Statement (1) shown for restricted class of proto**
- **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

$P \models X$	P **believes** X
$P \triangleleft X$	P **sees** X
$P \overset{K}{\longleftrightarrow} Q$	P**,** Q **can use shared key** K **to commu**
$P \Rightarrow X$	P **has jurisdiction over** X
$P \leadsto X$	P **once said** X
$\sharp(X)$	X **is fresh**

BAN Logic: Deductions

$$\frac{P \models Q \Rightarrow X \qquad P \models Q \models X}{P \models X}$$

$$\frac{P \models \sharp(X) \qquad P \models Q \leadsto X}{P \models Q \models X}$$

$$\frac{P \models Q \xleftrightarrow{K} P \qquad P \triangleleft \{\!|X|\!\}_K}{P \models Q \leadsto X}$$

Otway-Rees Protocol

- **Otway–Rees protocol (1987) (Adapted in BAN pa**

1.$A \longrightarrow B$**:** $M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{as}}$
2.$B \longrightarrow S$**:** $M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{as}}\ N_b\ \{\!|M\ A\ B|\!\}_{K_{bs}}$
3.$S \longrightarrow B$**:** $M\ \{\!|N_a\ K_{ab}|\!\}_{K_{as}}\ \{\!|N_b\ K_{ab}|\!\}_{K_{bs}}$
4.$B \longrightarrow A$**:** $M\ \{\!|N_a\ K_{ab}|\!\}_{K_{as}}$

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

BAN Logic: Idealization

- **This:**

$$
\begin{array}{ll}
A \rightarrow B: & M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{as}} \\
B \rightarrow S: & M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{as}}\ N_b\ \{\!|M\ A\ B|\!\}_{K_{bs}} \\
S \rightarrow B: & M\ \{\!|N_a\ K_{ab}|\!\}_{K_{as}}\ \{\!|N_b\ K_{ab}|\!\}_{K_{bs}} \\
B \rightarrow A: & M\ \{\!|N_a\ K_{ab}|\!\}_{K_{as}}
\end{array}
$$

- **Becomes:**

$$
\begin{array}{ll}
A \rightarrow B: & \{\!|M\ A\ B\ N_a|\!\}_{K_{as}} \\
B \rightarrow S: & \{\!|M\ A\ B\ N_a|\!\}_{K_{as}}\ N_b\ \{\!|M\ A\ B|\!\}_{K_{bs}} \\
S \rightarrow B: & \{\!|N_a, (A \overset{K_{ab}}{\longleftrightarrow} B), (B \leadsto M\ A\ B)|\!\}_{K_{as}} \\
 & \{\!|N_b, (A \overset{K_{ab}}{\longleftrightarrow} B), (A \leadsto M\ A\ B)|\!\}_{K_{bs}} \\
B \rightarrow A: & \{\!|N_a, (A \overset{K_{ab}}{\longleftrightarrow} B), (B \leadsto M\ A\ B)|\!\}_{K_{as}}
\end{array}
$$

BAN Logic: Starting Assumptions

$A \mid\equiv A \xleftrightarrow{K_{as}} S$ $\quad$ $B \mid\equiv B \xleftrightarrow{K_{bs}} S$ $\quad$ $S \mid\equiv$

$A \mid\equiv (S \Rightarrow A \xleftrightarrow{K_{ab}} B)$ $\quad$ $B \mid\equiv (S \Rightarrow A \xleftrightarrow{K_{ab}} B)$ $\quad$ $S \mid\equiv$

$A \mid\equiv (S \Rightarrow (B \leadsto X))$ $\quad$ $B \mid\equiv (S \Rightarrow (A \leadsto X))$ $\quad$ $S \mid\equiv$

$A \mid\equiv \sharp(N_a)$ $\quad$ $B \mid\equiv \sharp(N_b)$

$A \mid\equiv \sharp(N_b)$

BAN Logic: Conclusions

$$A \mid\equiv A \stackrel{K_{ab}}{\longleftrightarrow} B \qquad B \mid\equiv A \stackrel{K_{ab}}{\longleftrightarrow} B$$

$$A \mid\equiv B \mid\equiv (M\; A\; B) \qquad B \mid\equiv A \leadsto (M\; A$$

BAN Logic: Flawed!

- **Assume C has $\{\!|M'\,C\,B|\!\}_{K_{bs}}$ from previous run**

$$
\begin{array}{rl}
C(A) \longrightarrow B: & M\;A\;B\;\{\!|N_c\;M'\;C\;B|\!\}_{K_{cs}} \\
B \longrightarrow C(S): & M\;A\;B\;\{\!|N_c\;M\;A\;B|\!\}_{K_{cs}}\;N_b\;\{\!|M\;A \\
C \longrightarrow S: & M'\;C\;B\;\{\!|N_c\;M'\;C\;B|\!\}_{K_{bs}}\;N_b\;\{\!|M' \\
S \longrightarrow C(B): & M'\;\{\!|N_c\;K_{cb}|\!\}_{K_{as}}\;\{\!|N_b\;K_{cb}|\!\}_{K_{bs}} \\
C(S) \longrightarrow B: & M\;\{\!|N_c\;K_{cb}|\!\}_{K_{cs}}\;\{\!|N_b\;K_{cb}|\!\}_{K_{bs}} \\
B \longrightarrow C(A): & M\;\{\!|N_c\;K_{cb}|\!\}_{K_{cs}}
\end{array}
$$

BAN Logic: Source of Flaws

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

(In specification) $\{\!|N_b\, K_{ab}|\!\}_{K_{bs}}$

(Idealized as) $\{\!|N_b, (A \xleftrightarrow{K_{ab}} B), (A \leadsto M\; A\; B$

(Should be) $\{\!|N_b, (A \xleftrightarrow{K_{ab}} B), (A \leadsto M'\; A\; B$

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 spelle verified
    - Can build up lemmas for use in bigger pro
  - 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 (**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 $e\sharp x$ trace"

Honest Participants: Otway–Rees

$$A \rightarrow B: \quad M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{as}}$$

- **If ev is a trace, N_a a fresh nonce, $A \neq B$ and $B \neq$**
  $$(\text{Says A B } \langle M\ A\ B \{\!|N_a\ A\ B|\!\}_{K_{as}} \rangle)\ \sharp\ ev$$
  is also a valid trace

$$B \rightarrow S: \quad M\ A\ B\ \{\!|N_a\ M\ A\ B|\!\}_{K_{as}}\ N_b\ \{\!|M\ A\ B|\!\}_{K_{bs}}$$

- **If ev is a trace containing (Says A' B $\langle M\ A\ B\ X$ fresh, and $B \neq S$, then**
  $$\text{Says B S } \langle M\ A\ B\ X\ N_b\ \{\!|M\ A\ B|\!\}_{K_{bs}} \rangle\ \sharp\ e$$
  is also a valid trace

Modeling the Adversary

- **Need some additional operators**
  - analz H **is the set of terms the adversary c**
    from H**:**

$$\mathsf{H} \subseteq \mathsf{analz\ H}$$

$$\langle X, Y \rangle \in \mathsf{analz\ H} \Rightarrow \mathsf{X} \in \mathsf{analz\ H} \wedge \mathsf{Y} \in \mathsf{ar}$$

$$\{\!|\mathsf{X}|\!\}_K \in \mathsf{analz\ H} \wedge K^{-1} \in \mathsf{analz\ H} \Rightarrow \mathsf{X} \in \mathsf{a}$$

  - synth H **is the set of what the adversary can r**
    H**:**

$$\mathsf{X} \in \mathsf{synth\ H} \wedge \mathsf{Y} \in \mathsf{synth\ H} \Rightarrow \langle X, Y \rangle \in \mathsf{sy}$$

$$X \in \mathsf{synth\ H} \wedge \mathsf{K} \in \mathsf{synth\ H} \Rightarrow \{\!|\mathsf{X}|\!\}_{\mathsf{K}} \in \mathsf{sy}$$

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 X **events in** ev**, w**
    bad
- **Then if** X $\in$ synth(analz (spies ev))**, then**

  $$\text{Says Spy B X} \sharp\ ev$$

  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 operation
- Get from theorem prover
  - Environment in which to prove security prope
  - Assistance in doing so

Security Goals

- **Secrecy of session keys: For every valid trace ev,**

  $$\text{Says S B } \langle \text{M A B} \{\!|N_a\,\text{K}|\!\}_{K_{as}}\,\{\!|N_b\,\text{K}|\!\}_{K_{bs}} \in$$

  then $K \notin \text{analz (spies } ev)$
- **Authentication condition: For every valid trace ev**

  $$\text{Says A B } \langle M \text{ A B} \{\!|N_a\,\text{A B}|\!\}_{K_{as}}\rangle \in ev$$

  and

  $$\text{Says B' A } \langle M\ \{\!|N_a\,K|\!\}_{K_{as}}\rangle \in ev$$

  then

  $$\text{Says S B'' } \langle M\ \{\!|N_a\,K|\!\}_{K_{as}}\,\{\!|N_b'\,K|\!\}_{K_{bs}}\rangle \in$$

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 underly
    tives
  - Incorporating probabilistic reasoning
- Minimal systems that contain attacks
- Denial of service