Formal Analysis of Security Protocols

Dr. Changyu Dong

cd04@doc.ic.ac.uk
Security Protocols

• Are protocols concerning with security properties, e.g. integrity and confidentiality.
• For example
  – SSL, Kerberos, WPA, CHAP, Contract signing, E-voting, E-money.
• They have to be right, but they often aren’t.
Formal Analysis

• There are now several methods for analysing security protocols, based on:
  – Modal and temporal logics
  – Process algebras (CSP, pi-calculus)
  – Theorem-proving tools

• They have resulted in:
  – increased confidence in some protocols,
  – discovery of many protocol flaws,
  – a better understanding of how to design robust protocols.
BAN Logic

- Named after its authors Burrows, Abadi and Needham.
- A modal logic for analysing authentication protocols.
- First formal analysis technique for Security Protocols.
- Easy to use.
- Finds many attacks on known protocols.
Authentication, an Intuitive Example

• Imaging you received this email from your best friend’s address. Would you click the link?

Hi,
Here are some wonderful pictures. Click the link below: 

flickr.com

Tom
Authentication, an Intuitive Example

• Now imaging the email is written in this way

Hi,
Remember the guy who drank beer through his nose last night in the pub? I’ve put some photos online. Click the link below:

flickr.com

Tom

• Would you click the link?
Authentication, an Intuitive Example

• Now imaging you received this email and your friend Bob told you he received the same message and the link was genuine.

    Hi,
    Here are some wonderful pictures. Click the link below: flickr.com

    Tom

• Would you click the link?
The Key Factors in Authentication

- Some secrets
- Or some trust
- And ...
Authentication, an Intuitive Example

• Imaging you received this email again one week later
  
  Hi,  
  Remember the guy who drank beer through his nose last night in the pub? I’ve put some photos online. Click the link below:  
  flickr.com
  
  Tom

• Would you click the link?
The Key Factors in Authentication

- Some secrets
- Or some trust
- And Freshness
BAN Logic: Basic Notations

- **P believes X**: Principle P believes X. P acts as if X is true. P knows X is indeed true or the truth of X is justified by some evidence.

- **P sees X**: The principal P receives a message containing X. Seeing is NOT believing. P does not necessarily believes X even if P sees X.

- **P said X**: At some point in the past, P is known to have sent a message including X.

- **P controls X**: P has jurisdiction over X, or P is trusted as an authority on X.
BAN Logic: Basic Notations

• *(Fresh)*: *X* is recent or has not been used before. It is a fresh value (nonce, timestamp).

• $P \leftrightarrow^K Q$: *P* and *Q* may use the shared key *K* to communicate. *K* is only known to *P* and *Q* or a principal trusted by *P* and *Q* (such as an authentication server).

• $\rightarrow^K P$: *P* has *K* as a public key. The private key ($K^{-1}$) is known only to *P*.

• $P \leftrightarrow^X Q$: *X* is a shared secret between *P* and *Q*.
BAN Logic: Basic Notations

- \{X\}_K: X is encrypted under the key K.
- \langle X \rangle_Y: X is combined with a secret Y.
- X,Y means X and Y
Authentication goals: formalised

• Mutually Authenticated key establishment between A and B:

\[ A \text{ believes } A \xleftrightarrow{K} B, \ B \text{ believes } A \xleftrightarrow{K} B \]

• Mutually Authenticated & confirmed key establishment between A and B:

\[ A \text{ believes } B \text{ believes } A \xleftrightarrow{K} B, \ B \text{ believes } A \text{ believes } A \xleftrightarrow{K} B \]

• Data Origin Authentication

\[ A \text{ believes } B \text{ believes } X \]
Inference Rules

• Message-meaning

\[ P \text{ believes } Q \leftrightarrow P, \ P \text{ sees } \{X\}_K \]

\[ P \text{ believes } Q \text{ said } X \]

• K is a key shared between Q and P, so if P sees a message encrypted by K, it must come from Q (P ignores its own messages). Similarly:

\[ P \text{ believes } P \leftrightarrow Q, \ P \text{ sees } \langle X \rangle_Y \]

\[ P \text{ believes } Q \text{ said } X \]

\[ P \text{ believes } Q, \ P \text{ sees } \{X\}_{K^{-1}} \]

\[ P \text{ believes } Q \text{ said } X \]
Inference Rules

• Nonce-verification

\[
P \text{ believes fresh}(X), \ P \text{ believes } Q \text{ said } X
\]

\[
P \text{ believes } Q \text{ believes } X
\]

• If P believes that Q once said X, then P believes that Q once believed X. If X is fresh, then Q should still hold this belief.
Inference Rules

• Jurisdiction rule

\[
P \text{ believes } \underbrace{Q \text{ controls } X, P \text{ believes } Q \text{ believes } X}_{P \text{ believes } X}
\]

• If P trusts Q as an authority on X, then P should believe X if Q does so.
Inference Rules

- Decomposition rules

\[
\frac{P \text{ sees } (X,Y)}{P \text{ sees } X} \quad \frac{P \text{ sees } <X>_Y}{P \text{ sees } X} \quad \frac{P \text{ believes } (X,Y)}{P \text{ believes } X}
\]

- Decryption rules

\[
\frac{P \text{ believes } Q}{P, P \text{ sees } \{X\}_K} \quad \frac{P \text{ believes } \{X\}_K}{P \text{ sees } X} \quad \frac{P \text{ believes } Q, P \text{ sees } \{X\}_{K^{-1}}}{P \text{ sees } X}
\]
Inference Rules

• Freshness rule

\[
P \text{ believes } \text{fresh}(X) \\
\frac{}{P \text{ believes } \text{fresh}(X, Y)}
\]

• If part of the message is fresh, then the whole message is also fresh.
Protocol Analysis with BAN Logic

- Idealise the protocol by replacing concrete messages by idealised messages.
- State the initial assumptions
- Apply the inference rules to each message and interpret the meaning of derived formulas.
Idealise the Protocol

• Drop unencrypted messages, because they do not contribute to the growth of principle’s beliefs.
• Drop all other data that does not contribute to the growth of beliefs. (Usually nonces and timestamps need to be left in.)
• Understand the real meaning of messages and represent the meaning in the idealised messages using correct formulas.
Andrew Secure RPC

- This protocol is intended to distribute a new session key between two principals A and B who share a long-term key.

\[
\begin{align*}
A & \to B: A, \{ N_a \} K_{ab} \\
B & \to A: \{ N_a + 1, N_b \} K_{ab} \\
A & \to B: \{ N_b + 1 \} K_{ab} \\
B & \to A: \{ K_{ab}', N_b \} K_{ab}
\end{align*}
\]

- Idealise

\[
\begin{align*}
A & \to B: \{ N_a \} K_{ab} \\
B & \to A: \{ N_a, N_b \} K_{ab} \\
A & \to B: \{ N_b \} K_{ab} \\
B & \to A: \{ A \to B, N_b \} K_{ab}
\end{align*}
\]
Initial Assumptions

\[ A \text{ believes } A \leftarrow K_{ab} \rightarrow B, \ B \text{ believes } A \leftarrow K_{ab} \rightarrow B \]

\[ B \text{ believes } A \leftarrow K'_{ab} \rightarrow B \]

\[ A \text{ believes (B controls } A \leftarrow K_{ab} \rightarrow B), \ A \text{ believes fresh}(N_a), \ B \text{ believes fresh}(N_b), \ B \text{ believes fresh}(N'_b) \]

- \( K_{ab} \) is pre-shared
- \( K'_{ab} \) is generated by B
- \( N_a \) is generated by A
- \( N_b \) is generated by B
Analyse the Protocol

• The goal of the protocol is to distribute a new shared keys between A and B.
• That means, at the end of the protocol, A should believe that it shares $K_{ab}'$ with B and B should believe that it shares $K_{ab}'$ with A.

\[ B \text{ believes } A \leftrightarrow B, \quad A \text{ believes } A \leftrightarrow B \]

• The first goal is in the initial assumptions, how about the second?
Analyse the Protocol

• Let’s go backwards:

\[
\text{A believes } \overrightarrow{A \rightarrow B} \quad \text{Jurisdiction rule}
\]

\[
\text{A believes } \overrightarrow{B \rightarrow A}, \quad \text{A believes } \overrightarrow{B \rightarrow A} \quad \text{Nonce - verification rule}
\]

\[
\text{Initial Assumptions: } \overrightarrow{A \rightarrow B}, \quad \text{A believes } \text{fresh}(\overrightarrow{A \rightarrow B}), \quad \text{A believes } \text{B said } \overrightarrow{A \rightarrow B}
\]
Analyse the Protocol

A believes fresh(A → B), A believes B said A → B

↑ message meaning

A believes A → B, A sees \{ A → B \}_{K_{ab}}

↑ Decomposition Rule

Initial Assumptions

A sees \{ A → B, N_{b} \}_{K_{ab}}

↑

message 4
Analyse the Protocol

• How to prove $A \text{ believes fresh}(A \leftrightarrow B)$?

• Unfortunately, we cannot.

• That means the protocol has a flaw.
  – A cannot verify the new key is fresh.
  – An attacker can mount a replay attack.
Modified Protocol

\[
\begin{align*}
A \rightarrow B &: A, N_a \\
B \rightarrow A &: \{N_a, K'_{ab}\} K_{ab} \\
A \rightarrow B &: \{N_a\} K'_{ab} \\
B \rightarrow A &: N_b
\end{align*}
\]
A Few Words at the End

• BAN logic can help you find flaws
  – But not all the flaws
    • The Needham-Schroeder public-key protocol was proved to be correct in the BAN logic paper, but later was found flawed.

• The original paper is worth reading
  – A Logic of Authentication, M Burrows, M Abadi, R Needham, ACM Transactions on Computer Systems 1990