Logic for Computer Security Protocols

John Mitchell
Stanford University

Outline
- Example
  - Floyd-Hoare logic of programs
- BAN logic
- Current Protocol Logic

Part I
Logic of programs

Before-after assertions
- Main idea
  - $F \implies P \implies G$
    - If $F$ is true before executing $P$, then $G$ after
- Two variants
  - Total correctness $F [P] G$
    - If $F$ before, then $P$ will halt with $G$
  - Partial correctness $F (P) G$
    - If $F$ before, and if $P$ halts, then $G$

While programs
- Programs
  $P ::= x := e \mid P ; P \mid \text{if } B \text{ then } P \text{ else } P \mid \text{while } B \text{ do } P$
where $x$ is any variable
$e$ is any integer expression
$B$ is a Boolean expression (true or false)

Assertion about assignment
- Assignment axiom
  $F(t) (x := t) F(x)$
- Examples
  - $7 = 7 \{ x := 7 \} x = 7$
  - $(y+1) > 0 \{ x := y+1 \} x > 0$
  - $x + 1 = 2 \{ x := x + 1 \} x = 2$

This is not most general case.
Need to assume no aliasing...
Rule of consequence

◆ If
  • $F \{ P \} G$
◆ And
  • $F' \rightarrow F$ and $G \rightarrow G'$
◆ Then
  • $F \{ P \} G'$

Conditional

\[
\begin{align*}
F \land B & \{ P_1 \} G \\
F \land \neg B & \{ P_2 \} G
\end{align*}
\]

\[
F \{ \text{if } B \text{ then } P_1 \text{ else } P_2 \} G
\]

◆ Example
  
  \[\text{true (if } y \geq 0 \text{ then } x \leftarrow y \text{ else } x \leftarrow -y) \quad x \geq 0\]

Sequence

\[
\begin{align*}
F & \{ P_1 \} G \\
G & \{ P_2 \} H
\end{align*}
\]

\[
F \{ P_1 ; P_2 \} H
\]

◆ Example
  
  \[x=0 \{ x := x+1 ; x := x+1 \} \quad x=2\]

Loop Invariant

\[
\begin{align*}
F \land B & \{ P \} F \\
F \{ \text{while } B \text{ do } P \} & F \land \neg B
\end{align*}
\]

◆ Example
  
  \[\text{true \{ while } x \neq 0 \text{ do } x \leftarrow x-1\} \quad x=0\]

Example: Compute $d=x-y$

◆ Assertion
  
  \[y \leq x \quad (d=0, \text{ while } (y+d) \land x \text{ do } d := d+1) \quad y+d=x\]
◆ Main ideas in proof
  
  \[\begin{align*}
\text{Choose loop invariant } y+d & \leq x \\
\text{y+d} & \leq x \{ P_1 \} y+d & \leq x \\
y+d & \leq x (\text{while } B \text{ do } P_2) y+d & \leq x \land \neg B
\end{align*}\]
  
  Use assignment axiom and sequence rule to complete the proof of property of $P_1$
Facts about Hoare logic

- Compositional
  - Proof follows structure of program
- Sound
- "Relative completeness"
  - Properties of computation over N provable from properties of N
  - Some technical issues ...
- Important concept: Loop invariant !!!
  - Common practice beyond Hoare logic

Part II

BAN Logic

There is something called BAN

- Needham
  - "The main contribution of BAN logic was to make the study of 3-line protocols intellectually respectable."

Paper,

Using BAN Logic

- Protocol expressed in "idealized" form
- Identify initial assumptions in the language of BAN logic
- Use postulates and rules of BAN logic to deduce new predicate

Notation

\[ P \models X : \quad P \text{ believes } X \]
- P would be entitled to believe X.
- The principal P may act as though X is true.

\[ P \models X : \quad P \text{ sees } X \]
- P can read the contents of X (possibly after decryption, assuming P has the needed keys)
- P can include X in messages to other principals

BAN Logic

\[ P \models X \]
- P once said X
  - P sent a message including the statement X.
  - Possibly in the past or in the current run of the protocol
  - P believed that X was true when it sent the message

\[ P \models \Rightarrow X \]
- P controls X
  - P has jurisdiction over X
  - P is a trusted authority on the truth of X

\[ #(X) \]
- X is fresh
  - The present begins with the start of the current execution of the current protocol
  - X is fresh if it is not contained in any message in the past
BAN Logic

\[ K \]
\[ P \leftrightarrow Q : \quad \text{K is a shared key for P and Q.} \]
\[ \quad \text{• K is a secure key for communication between P and Q.} \]
\[ \quad \text{• K will never be discovered by any principal except for P or Q, or a principal trusted by either P or Q.} \]

\[ K \]
\[ |\rightarrow P : \quad \text{K is a public key for P.} \]
\[ \quad \text{• The matching secret key (the inverse of K, denoted by K^{-1}) will never be discovered by any principal except P, or principals trusted by P.} \]