Epistemic Logic

And Applications to Formal Protocol Verification
Outline

- Epistemic Logic
  - Syntax
  - Semantics
  - Axioms

- Burrow-Abadi-Schroeder (BAN) Logic
  - Syntax & Semantics
  - Verification of Needham-Schroeder Protocol
Epistemic Logic

Syntax Samples:

- $\varphi$
- $K_a \varphi$
- $K_a (\neg K_b \varphi)$
- $K_a \varphi \rightarrow \varphi$
- $B_a \varphi$
Epistemic Logic (2)

- Semantics: Kripke Models
  - \( M = \langle W, \varepsilon, (R_a)_{a \in A} \rangle \)
  - \( W \): Worlds – \( P \): Proposition – \( A \): Subjects
  - \( \varepsilon : W \rightarrow P(P) \)
  - \( R_a \subseteq W \times W \)

- Semantics of \( K_a \varphi \)
  - \( (M, w) \models K_a \varphi \) iff \( \forall w' \in R_a (w) : (M, w) \models \varphi \)
An Example

- Three children: a, b, c.
- $\varphi_a$ : Forehead of the child “a” is dirty.
- Kripke Model:

\[
(M,110) \models K_a \varphi_b
\]
\[
(M,110) \models K_a \neg \varphi_c
\]
\[
(M,110) \models \neg K_a \varphi_a
\]
\[
(M,110) \models \neg K_a \neg \varphi_a
\]

[Ramezanian]
Axioms

- \( \varphi \) s.t. \( \varphi \) is tautology.
- (T) \( K_i \varphi \rightarrow \varphi \)
- (D) \( K_i \varphi \rightarrow \neg K_i \neg \varphi \)
- (K) \( K_i (\varphi \rightarrow \psi) \rightarrow (K_i \varphi \rightarrow K_i \psi) \)
  \[ K_i \varphi \land K_i (\varphi \rightarrow \psi) \rightarrow K_i \psi \]
- (4) \( K_i \varphi \rightarrow K_i K_i \varphi \)
- (5) \( \neg K_i \varphi \rightarrow K_i \neg K_i \varphi \)
Axioms (2)

- Implication Rules:
  - (MP) \[ \varphi \rightarrow (\varphi \rightarrow \psi) \]
  - \[ \psi \]
  - \[ \varphi \]
  - \[ K_i \varphi \]

- Knowledge System: KT45 = S5
- Belief System: KD45
S5 is Sound & Complete, if:

- (T) $R_i$ is Reflexive.
- (4) $R_i$ is Transitive.
- (5) $R_i$ is Euclidean.

Omniscience Problem!
BAN Logic


Aims:
- What does this protocol achieve?
- Does this protocol need more assumptions than another one?
- Does this protocol do anything unnecessary that could be left out without weakening the protocol?
- Does this protocol encrypt something that could be sent in clear without weakening the protocol?
Syntax Overview

- Syntax consists of:
  - Agent Identifiers (A, B, ...)
  - Key Identifiers (k_a, k_b, k_{ab}, ...)
  - Identifiers for atomic formulas (n_a, n_b, ...)
  - A set of formulas (X, Y, ...)

- Formulas:
  - Boolean Expressions: \( A \text{ believes } A \leftarrow^{k} B \)
  - Idealized Messages: \( \{n_a, A \leftarrow^{k} B\}_{k_as} \)
Basic Formulas

- Time epochs:
  - Past
  - Present

Formulas

- \( X_1, \ldots, X_n \) : concatenation
- \( \{X\}_k \) : \( X \) encrypted with \( k \)
Formulas – Idealized Messages

**Formulas.**

- \( A \xrightarrow{k} B \) : \( k \) is a key shared by \( A \) and \( B \).
- fresh(\( X \)) : \( X \) is a valid belief in the present epoch.

In fresh(\( X \)), the argument \( X \) could for instance be a nonce \( n \) or a formula of the form \( A \xleftarrow{k} B \):

- Intuitively, in these cases fresh(\( X \)) holds iff \( n \) (resp. \( k \)) has been generated in the present epoch.

The original BAN papers:

- write \( \#(X) \) instead of fresh(\( X \)).
Formulas (cont.)

- \( A \text{ receives } X \) : \( A \) receives idealized message \( X \)
- \( A \text{ said } X \) : \( A \) once sent idealized message \( X \)

- We pronounce:
  - \( A \text{ said } X \) as “\( A \text{ once said } X \)”.  
  - This emphasizes that \( A \) may have sent \( X \) long ago.

- The original BAN papers:
  - write \( A \triangleleft X \) instead of \( A \text{ receives } X \),
  - pronounce “\( A \text{ sees } X \)” instead of “\( A \text{ receives } X \)”,
  - write \( A \parallel X \) instead of \( A \text{ said } X \).
Formulas (cont.)

- A believes $X$ : A believes $X$
- A controls $X$ : A has jurisdiction over $X$

The next sentence is provable in BAN logic:

- If A believes B controls $X$ and A believes B believes $X$, then A believes $X$.
- B could, for instance, have jurisdiction to generate shared keys for A and B. (Then $X$ would be $A \xrightarrow{kab} B$.)

The original BAN papers:

- write $A \models X$ instead of A believes $X$,
- write $A \models X$ instead of A controls $X$. 
Rules – Receives

(Receives)

\[ A \text{ receives } \{X\}_k \quad A \text{ believes } A \xleftarrow{k} B \]

\[ A \text{ believes } B \text{ said } X \]

- This rule should be pronounced as:
  - If \( A \text{ receives } \{X\}_k \) and \( A \text{ believes } A \xleftarrow{k} B \)
  then \( A \text{ believes } B \text{ said } X \).

- The other proof rules are all of this format, too.

- This rule is sound under the assumption that each agent recognizes and ignores his own messages.
Rules – Fresh

(Fresh)

\[
\begin{align*}
A \text{ believes } B \text{ said } X & \quad A \text{ believes } \text{fresh}(X) \\
\hline
A \text{ believes } B \text{ believes } X
\end{align*}
\]
Rules – Fresh Inject

(Fresh Inject)

\[
\frac{A \text{ believes } \text{fresh}(X_i)}{A \text{ believes } \text{fresh}(X_1, \ldots, X_i, \ldots, X_n)}
\]

This rule says:

- If \(A\) believes that some part \(X_i\) of a formula is fresh, then she believes that the whole formula is fresh.
- Typically, \(X_i\) is a nonce \(n\), a freshly generated session key \(k\), or a basic formula of the form \(A \xleftarrow{k} B\) where \(k\) is freshly generated.
(Jurisdiction)

\[
A \text{ believes } B \text{ believes } X \quad A \text{ believes } B \text{ controls } X
\]

\[
A \text{ believes } X
\]
Rules – Selection

(R-Select)
\[ A \text{ receives } (X_1, \ldots, X_i, \ldots, X_n) \]
\[ A \text{ receives } X_i \]

(BB-Select)
\[ A \text{ believes } B \text{ believes } (X_1, \ldots, X_i, \ldots, X_n) \]
\[ A \text{ believes } B \text{ believes } X_i \]

(BS-Select)
\[ A \text{ believes } B \text{ said } (X_1, \ldots, X_i, \ldots, X_n) \]
\[ A \text{ believes } B \text{ said } X_i \]
Goal: A and B want to establish a short-term session key $k_{ab}$.

They trust the server S to generate session keys.

They share long-term keys $k_{as}$ and $k_{bs}$ with S.

1: A, B, $N_a$

2: $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{ba}}\}_{K_{as}}$

3: $\{K_{ab}, A\}_{K_{ba}}$

4: $\{N_b\}_{K_{ab}}$

5: $\{N_b - 1\}_{K_{ab}}$
Idealized Messages

2. \( S \rightarrow A : \{ na, A \xleftarrow{kab} B, \text{fresh}(A \xrightarrow{kab} B), T \}_{kas} \)
   where \( T = \{ A \xleftrightarrow{kab} B, \text{fresh}(A \xrightarrow{kab} B) \}_{kbs} \)

3. \( A \rightarrow B : \{ A \xleftrightarrow{kab} B, \text{fresh}(A \xrightarrow{kab} B) \}_{kbs} \)

4. \( B \rightarrow A : \{ A \xleftrightarrow{kab} B \}_{kab} \)

5. \( A \rightarrow B : \{ A \xleftarrow{kab} B \}_{kab} \)
Assumptions

\[ S \text{ believes } A \xleftrightarrow{kab} B \quad \text{(I-S1)} \]
\[ S \text{ believes fresh}(A \xleftrightarrow{kab} B) \quad \text{(I-S2)} \]
\[ A \text{ believes } A \xleftrightarrow{kas} S \quad \text{(I-A1)} \]
\[ A \text{ believes } S \text{ controls } A \xleftrightarrow{kab} B \quad \text{(I-A2)} \]
\[ A \text{ believes } S \text{ controls fresh}(A \xleftrightarrow{kab} B) \quad \text{(I-A3)} \]
\[ A \text{ believes fresh}(na) \quad \text{(I-A4)} \]
\[ B \text{ believes } B \xleftrightarrow{kbs} S \quad \text{(I-B1)} \]
\[ B \text{ believes } S \text{ controls } A \xleftrightarrow{kab} B \quad \text{(I-B2)} \]
\[ B \text{ believes } S \text{ controls fresh}(A \xleftrightarrow{kab} B) \quad \text{(I-B3)} \]
Assume:

\* Init \*

A receives \{na, \(A \leftrightarrow_{kab} B\), fresh(\(A \leftrightarrow_{kab} B\)), T\}_kas \hspace{1cm} (R2)

Show:

A believes \(A \leftrightarrow_{kab} B\)
A believes fresh(\(A \leftrightarrow_{kab} B\))

A believes \(S\) said (\(na, A \leftrightarrow_{kab} B\), fresh(\(A \leftrightarrow_{kab} B\)), T) \hspace{1cm} (R2-1)
by (R2), (I-A1) and rule (Receives)

A believes fresh(\(na, A \leftrightarrow_{kab} B\), fresh(\(A \leftrightarrow_{kab} B\)), T) \hspace{1cm} (R2-2)
by (I-A4) and rule (Fresh Inject)

A believes \(S\) believes (\(na, A \leftrightarrow_{kab} B\), fresh(\(A \leftrightarrow_{kab} B\)), T) \hspace{1cm} (R2-3)
by (R2-1), (R2-2) and rule (Fresh)
A believes S believes $A \leftrightarrow_{kab} B$  
by (R2-3) and rule (BB-Select)  \hfill (R2-4)

A believes S believes fresh($A \leftrightarrow_{kab} B$)  
by (R2-3) and rule (BB-Select)  \hfill (R2-5)

A believes $A \leftrightarrow_{kab} B$  
by (R2-4), (I-A2) and rule (Jurisdiction)  \hfill (R2-6)

A believes fresh($A \leftrightarrow_{kab} B$)  
by (R2-5), (I-A3) and rule (Jurisdiction)  \hfill (R2-7)
After Message 3

Assume:

\[ \text{Init} \]

\[ B \text{ receives } \{ A \overset{kab}{\leftrightarrow} B, \text{fresh}(A \overset{kab}{\leftrightarrow} B) \}_{kbs} \]  

(R3)

Show:

\[ B \text{ believes } A \overset{kab}{\leftrightarrow} B \]

\[ B \text{ believes fresh}(A \overset{kab}{\leftrightarrow} B) \]

\[ B \text{ believes } S \text{ said } (A \overset{kab}{\leftrightarrow} B, \text{fresh}(A \overset{kab}{\leftrightarrow} B)) \]  

(R3-1)

by (R3), (I-B1) and rule (Receives)

- We’d like to promote (R3-1) to:
  
  \[ B \text{ believes } S \text{ believes } (A \overset{kab}{\leftrightarrow} B, \text{fresh}(A \overset{kab}{\leftrightarrow} B)) \]

- But we can’t, because the message does not contain a proof of freshness.

- We are stuck!
Realized Attack

// A and S exchange messages 1 and 2 normally.
A → I : \{msg_3, A, kab\}_{kbs} // I saves message 3.
I → B : \{msg_3, A, kab\}_{kbs}
// B and A exchange messages 4 and 5 normally.

// Time passes and I manages to crack kab.
I → B : \{msg_3, A, kab\}_{kbs} // I sends cracked key to B.
B generates nonce nb'
B → I : \{msg_4, nb'\}_{kab} // I completes run with B.
I → B : \{msg_5, nb'\}_{kab}

Now B thinks he shares kab with A.
But he really shares it with I. Ooops!
Questions?


- رسول رمضانیان، مدل سازی منطقی سیستم های اجتماعی، پیشنویس 1388، قابل دریافت از http://sina.sharif.edu/~ramezanian/Soicalsystems2.pdf
BAN Logic – Limitations

- BAN logic assumes that agents never publish secrets, but BAN logic does not verify this.
- BAN logic assumes that agents can recognize type flaws, but BAN logic does not verify the absence of type flaws.
  - In particular, BAN logic assumes that agents always recognize and ignore messages that they have sent themselves.
- BAN logic assumes that all protocol participants are honest. No compromised agents are considered. Attackers do not have valid keys.
- Like always in this class, BAN logic assumes perfect cryptography.