6.857 Computer and Network Security **Fall Term, 1997**

Lecture 16 : October 28, 1997

Lecturer: Ron Rivest *Scribe: Chris Sepulveda*

convention

1 Handouts

- Midterm Quiz
- *A Logic for Authentification* by Burrows, Abadi, and Needham

2 Today's Topics

- Kerberos
- BAN Logic – for proving the security of protocols
- BAN Logic proof of Kerberos

3 Kerberos

3.1 The Kerberos protocol

Assumptions

- A and B have no prior knowledge of each other (i.e., no public keys)
- A and B trust S
- A and B have established secret keys with S (k_{as}, k_{bs})

The protocol is defined as follows:

1. A sends a request for an authenticator for B to S.

2. S replies with a message only decodable by A which contains a timestamp and an authenticator A can send to B to demonstate he is real.

3. sends an initial message to B telling him he is there, has a valid authenticator, and wants to know if he knows how to decode it and reply.

4. Finally, B demonstrates that he can parse the authenticator and sends back a reply to A's challenge.

4 BAN Logic

In the Kerberos protocol, for example, how can we be sure that it is secure? We ideally would like some formal method for asserting that we trust or don't trust the Kerberos protocol. BAN logic is an attempt to provide a formal method for analyzing protocols.

The basic problems that BAN logic tries to address is:

- there is some goal we want to achieve
- we have some idea of what we want, need
- we want to satisfy ourselves that our solution to the the above goal works
- we don't want to depend on trial by fire

There are a lot of problems involved when trying to prove security. As we go over BAN logic, we want to scrutinize it, considering how much do we trust its validity? Are the rules of BAN logic sound and reasonable? When are assumptions being made and what are they?.

Even if we prove something like Kerberos works theoretically, we still can have problems. Using somthing like BAN logic, we are proving the ideal. Unfortunately, implementations aren't always(ever?) ideal. This is just another concern we need to be aware of.

4.1 The BAN terms (principals)

The following terms are described in the BAN paper.

Principals are those agents involved in the protocol (usually people or programs). P, Q, R are all possible symbols for the principals. (S,A,B are principals in Kerberos).

Keys are used to encrypt messages symmetrically. K_{ab} and K_{as} are possible keys.

Public Keys are similar to Keys except that they are used in pairs, one of which is known to all. (K_b, K_b^{-1}) is a possible pair.

Nonces are message parts that are not meant to be repeated. N_a, N_b are nonces.

Timestamps are similar to nonces in that they are unlikely to be repeated. However, they reflect the current time of the person creating them. T_a, T_b are timestamps.

4.2 The *BAN* Statements

Once we have a formal definition of terms, we need a formal definition of the statements we may make (the constructs)

P **believes** X : P *believes* X, or P would be entitled to believe X. In particular, P can take X as true.

P **sees** X : P *sees* X. P has received some message X and is capable of reading and repating it.

P **said** X : P *once said* X. P at some time sent a message including the statement X. It is not knownwhether this is a replay, though it is known that P believed X when he sent it.

P **controls** X : P has *jurisdiction over* X. The principal P is an authority on X and should be trusted on this matter. (In Kerberos, S controls K_{ab})

fresh(X) : The message X is *fresh*; that is, X has not been sent in a message at any time before the current run of the protocol. This is usually true for *nonces*.

$P \overset{K}{\leftrightarrow} Q$: P and Q may use the *shared key* K to communicate. The key K is good in that it will be known only by P and Q.

$\overset{K}{\mapsto} P$: P has K as a *public key*. The matching *secret key* (denoted K^{-1}) will never be discovered by any principal excpet P or a principal trusted by P.

$P \overset{X}{\rightleftharpoons} Q$: The formula X is a *secret* known only to P and to Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another. An example is a secret password.

$\{X\}_K$ **or** $[X]_K$: This represents the formula X encrypted under the key K. This is short for $[X]_K from P$.

$\langle X \rangle_Y$: This represents X *combined with* the formula Y. Y is intended to be secret and that its presence proves the identity of whoever utters $\langle X \rangle_Y$. In implementations, X can simply be concatenated with the password Y. We assume that this is encrypted so that replay cannot be used.

4.3 Ban Rules of Inference

1. *Message meaning rules* concern the interpretation of messages. They all derive beliefs about the origin of messages.

   For shared keys, we postulate

   $$\frac{P \text{ believes } Q \overset{K}{\leftrightarrow} P, P \text{ sees } [X]_K}{P \text{ believes } Q \text{ said } X}$$

   That is, if P believes that the key K is shared with Q and sees X encrypted under K, then P believes that Q once said X.

   For public keys, we postulate

   $$\frac{P \text{ believes } \overset{K}{\mapsto} Q, P \text{ sees } [X]_{K^{-1}}}{P \text{ believes } Q \text{ said } X}$$

   That is, if P believes that K is Q's public key, and P receives a message encoded with Q's secret key, then P believes Q once said X.

   For shared secrets, we postulate

   $$\frac{P \text{ believes } Q \overset{Y}{\rightleftharpoons} P, P \text{ sees } \langle X \rangle_Y}{P \text{ believes } Q \text{ said } X}$$

   That is, if P believes that the secret Y is shared with Q and sees $\langle X \rangle_Y$, then P believes that Q once said X.

2. The *nonce-verification* rule expresses the check that a message is recent, and hence, that the sender still believes in it:

$$\frac{P \text{ believes fresh } (X),\ P \text{ believes } Q \text{ said } X}{P \text{ believes } Q \text{ believes } X}$$

   That is, if P believes that X could have been uttered only recently and that Q once said X, then P believes that Q believes X. For simplicity, X should be "cleartext".

3. The *jurisdiction* rule states that if P believes that Q has jurisdiction over X, then P trusts Q on the trueth of X:

$$\frac{P \text{ believes } Q \text{ controls } X,\ P \text{ believes } Q \text{ believes } X}{P \text{ believes } X}$$

4. If a principal sees a formula, the he also sees its components, provided he knows the necessary keys:

$$\frac{P \text{ sees } (X,Y)}{P \text{ sees } X}, \quad \frac{P \text{ sees } \langle X \rangle_Y}{P \text{ sees } X}, \quad \frac{P \text{ believes } Q \overset{K}{\leftrightarrow} P \text{(,) } P \text{ sees } [X]_K}{P \text{ sees } X},$$
$$\frac{P \text{ believes } \overset{K}{\mapsto} P,\ P \text{ sees } [X]_K}{P \text{ sees } X}, \quad \frac{P \text{ believes } \overset{K}{\mapsto} P,\ P \text{ sees } [X]_{K^{-1}}}{P \text{ sees } X}.$$

   Note that if P sees X and P sees Y it does NOT follow that P sees (X, Y) since that means that X and Y were uttered at the same time.

5. If one part of the formula is fresh, then the entire formula must be fresh:

$$\frac{P \text{ believes fresh}(X)}{P \text{ believes fresh}(X, Y)}.$$

5 Proof of Kerberos

Now that we have a system for proving a protocol works, lets test the Kerberos protocol.

5.1 Kerberos, in BAN logic

Let's re write the protocol in the language of BAN logic.

1. $A \rightarrow S : (A, B)$
2. $S \rightarrow A : [T_s, A \overset{K_{ab}}{\leftrightarrow} B, [T_s, A \overset{K_{ab}}{\leftrightarrow} B]_{K_{bs}}]_{K_{as}}$
3. $A \rightarrow B : [T_s, A \overset{K_{ab}}{\leftrightarrow} B]_{K_{bs}}, [T_s, A \overset{K_{ab}}{\leftrightarrow} B]_{K_{ab}}$
4. $B \rightarrow A : [T_s, A \overset{K_{ab}}{\leftrightarrow} B]_{K_{ab}}$

5.2 Goals

We want to prove the following:

$$A \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B \ \& \ B \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B$$

and ideally

$$A \text{ believes } B \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B \ \& \ B \text{ believes } A \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B$$

Basically, we want A and B to establish their key, K_{ab} and believe that each has the key and that is is valid. Also, we would like each to believe that the other believes that they have established the same, valid key.

5.3 Assumptions

A believes $A \overset{K_{as}}{\leftrightarrow} S$,	B believes $B \overset{K_{bs}}{\leftrightarrow} S$,
S believes $A \overset{K_{as}}{\leftrightarrow} S$,	S believes $B \overset{K_{bs}}{\leftrightarrow} S$,
S believes $A \overset{K_{ab}}{\leftrightarrow} B$,	B believes (S controls $A \overset{K}{\leftrightarrow} B$),
A believes (S controls $A \overset{K}{\leftrightarrow} B$),	B believes fresh(T_s),
A believes fresh(T_s),	B believes fresh(T_a),

5.4 the proof

Starting with message 2 (see protocol),

$$A \text{ sees } [T_s, (A \overset{K_{ab}}{\leftrightarrow} B), [T_s, A \overset{K_{ab}}{\leftrightarrow} B]_{K_{bs}}]_{K_{as}}$$

and using the assumption

$$A \text{ believes } A \overset{K_{as}}{\leftrightarrow} S$$

apply a *message meaning rule* for shared keys we get

$$A \text{ believes } S \text{ said } (T_s,\ (A \overset{K_{ab}}{\leftrightarrow} B),\ [T_s,\ A \overset{K_{ab}}{\leftrightarrow} B]_{K_{bs}})$$

which can be broken up into

$$A \text{ believes } S \text{ said } (T_s,\ (A \overset{K_{ab}}{\leftrightarrow} B))$$

Using the assumption

$$A \text{ believes } \text{fresh}(T_s)$$

and the *nonce verification rule*, we get

$$A \text{ believes } S \text{ believes } (T_s, A \overset{K_{ab}}{\leftrightarrow} B)$$

Breaking this up into

$$A \text{ believes } S \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B$$

Now use the assumption

$$A \text{ believes } S \text{ controls } A \overset{K_{ab}}{\leftrightarrow} B$$

with the *juristiction rule* we get

$$A \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B$$

This completes the analysis of messaage 2.

You can repeat the same analysis to get

$$B \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B$$

Now message 3 can be decrypted and the following can be proved

$$B \text{ believes } A \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B$$

The last message assures A that B believes the key. We can use the *nonve verification* and *message meaning* rules to prove

$$\begin{array}{cc} A \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B & B \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B \\ A \text{ believes } B \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B & B \text{ believes } A \text{ believes } A \overset{K_{ab}}{\leftrightarrow} B \end{array}$$

We have proved our goals, and can verify that the idealization of Kerberos works, given our assumptions.

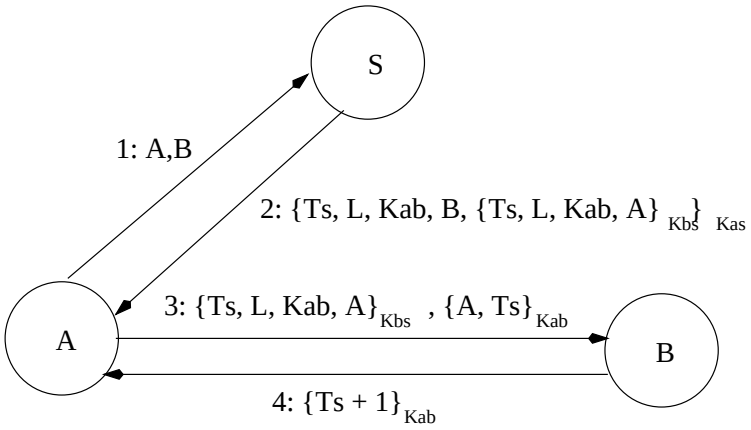

Figure 1: The Kerberos Protocol.