Lecture 1. Introduction to cryptographic protocols

S P Suresh, Chennai Mathematical Institute
spsuresh@cmi.ac.in

Topics in Security
August – November 2017
August 16, 2017

Administrative details

- Aim: Learn about formal modelling and verification of cryptographic protocols
- Instructors: S P Suresh and Vaishnavi Sundararajan
- Evaluation: Assignments (3×15), take-home exam (30), programming project (25)
- Moodle page: to be set up

Outline

1. What are security protocols?

2. A key establishment protocol

Outline

1. What are security protocols?

2. A key establishment protocol

What are security protocols?

- Systematic sequence of message exchanges to achieve a goal.
- Based on cryptographic tools.
- Distinct from cryptography schemes.

The domain of cryptography

- Transform a plain text to cipher text in such a way that it is computationally very difficult to compute the inverse without the key.
- With public key cryptography and digital signatures, one has a reasonable assurance of secrecy and authenticity.
- Cryptanalysis: attacks on cryptographic schemes, involve sophisticated mathematical techniques and computing power.
- Works at the level of messages.
- The connection between different messages is not the concern of cryptography.

An example security protocol

Msg 1. $A \rightarrow B : \{x\}_{k_b}$

Msg 2. $B \rightarrow A : \{x\}_{k_a}$

An example security protocol

Msg 1. $A \rightarrow B : \{x\}_{k_b}$
Msg 2. $B \rightarrow A : \{x\}_{k_a}$

- To be distinguished from a mere sequence of (signed and) encrypted communcations.
- The two messages are part of one logical entity: with the x providing the connection.
- Authentication protocols are typically run prior to a secure session, to establish identities, keys, etc.

Outline

1 What are security protocols?

2 A key establishment protocol

The basic setting

- *A* and *B* wish to share a session key,

The basic setting

- *A* and *B* wish to share a session key, despite the machinations of *I* (the malicious intruder).

The basic setting

- A and B wish to share a session key, despite the machinations of I (the malicious intruder).
- They are aided in this by the trusted server S.

The basic setting

- A and B wish to share a session key, despite the machinations of I (the malicious intruder).
- They are aided in this by the trusted server S.
- A, B, and S do not engage in activity that deliberately compromises the security of the key.

The basic setting

- A and B wish to share a session key, despite the machinations of I (the malicious intruder).
- They are aided in this by the trusted server S.
- A, B, and S do not engage in activity that deliberately compromises the security of the key.
- S is trusted to generate a random, unguessable key.

The goals

- At the end of the protocol, the value of the session key should be known to both A and B, but to none else save S.

The goals

- At the end of the protocol, the value of the session key should be known to both A and B, but to none else save S.
- A and B should have some assurance that the key is newly generated.

A first attempt

Msg 1. $A \rightarrow S : A, B$
Msg 2. $S \rightarrow A : k_{ab}$
Msg 3. $A \rightarrow B : k_{ab}, A$

The hidden assumptions

- Only the messages passed in a successful run of the protocol are specified.

The hidden assumptions

- Only the messages passed in a successful run of the protocol are specified.
- There is no description of what happens if a message of the wrong format is received, or if no message is received at all.

The hidden assumptions

- Only the messages passed in a successful run of the protocol are specified.
- There is no description of what happens if a message of the wrong format is received, or if no message is received at all.
- There is no specification of the internal actions of the different principals.

Defects in our first protocol

- Obvious breach of secrecy.

Defects in our first protocol

- Obvious breach of secrecy.
- Security Assumption: I is able to eavesdrop on all messages sent in a cryptographic protocol.

Defects in our first protocol

- Obvious breach of secrecy.
- Security Assumption: I is able to eavesdrop on all messages sent in a cryptographic protocol.
- Need to use cryptography to provide secrecy.

A second attempt

- Use the shared keys k_{as} and k_{bs}:

$$
\begin{array}{ll}
\text{Msg 1.} & A \rightarrow S : A, B \\
\text{Msg 2.} & S \rightarrow A : \{k_{ab}\}_{k_{as}}, \{k_{ab}\}_{k_{bs}} \\
\text{Msg 3.} & A \rightarrow B : \{k_{ab}\}_{k_{bs}}, A
\end{array}
$$

A second attempt

- Use the shared keys k_{as} and k_{bs}:

$$\begin{array}{ll} \text{Msg 1.} & A \rightarrow S : A, B \\ \text{Msg 2.} & S \rightarrow A : \{k_{ab}\}_{k_{as}}, \{k_{ab}\}_{k_{bs}} \\ \text{Msg 3.} & A \rightarrow B : \{k_{ab}\}_{k_{bs}}, A \end{array}$$

- k_{as} is assumed to be known only to A and S. Similarly with k_{bs}.

Some modelling assumptions

- Messages are not treated as bit strings, but as symbolic terms.

Some modelling assumptions

- Messages are not treated as bit strings, but as symbolic terms.
- Perfect encryption assumed: I does not engage in cryptanalysis.

Some modelling assumptions

- Messages are not treated as bit strings, but as symbolic terms.
- Perfect encryption assumed: I does not engage in cryptanalysis.
- She learns m from $\{m\}_k$ only if she has the inverse of k.

Some modelling assumptions

- Messages are not treated as bit strings, but as symbolic terms.
- Perfect encryption assumed: I does not engage in cryptanalysis.
- She learns m from $\{m\}_k$ only if she has the inverse of k.
- Moreover, it is usually assumed that $\{m\}_k = \{m'\}_{k'}$ only if $m = m'$ and $k = k'$. So I cannot learn secrets by accident.

Some modelling assumptions

- Messages are not treated as bit strings, but as symbolic terms.
- Perfect encryption assumed: I does not engage in cryptanalysis.
- She learns m from $\{m\}_k$ only if she has the inverse of k.
- Moreover, it is usually assumed that $\{m\}_k = \{m'\}_{k'}$ only if $m = m'$ and $k = k'$. So I cannot learn secrets by accident.
- But still, I might exploit protocol loopholes to learn secrets.

Breach of authentication

- Security Assumption: I is able to alter all the messages sent in the public channel. She can also re-route any message to any principal. She can also generate and insert completely new messages.

Breach of authentication

- Security Assumption: I is able to alter all the messages sent in the public channel. She can also re-route any message to any principal. She can also generate and insert completely new messages.
- Breach of authentication ...

Msg 1.	$A \longrightarrow S : A, B$
Msg 2.	$S \longrightarrow A : \{k_{ab}\}_{k_{as}}, \{k_{ab}\}_{k_{bs}}$
Msg 3.	$A \longrightarrow (I)B : \{k_{ab}\}_{k_{bs}}, A$
Msg 3.	$I \longrightarrow B : \{k_{ab}\}_{k_{bs}}, I$

Breach of authentication

- Security Assumption: I is able to alter all the messages sent in the public channel. She can also re-route any message to any principal. She can also generate and insert completely new messages.
- Breach of authentication ...

$$
\begin{array}{ll}
\text{Msg 1.} & A \rightarrow S : A, B \\
\text{Msg 2.} & S \rightarrow A : \{k_{ab}\}_{k_{as}}, \{k_{ab}\}_{k_{bs}} \\
\text{Msg 3.} & A \rightarrow (I)B : \{k_{ab}\}_{k_{bs}}, A \\
\text{Msg 3.} & I \rightarrow B : \{k_{ab}\}_{k_{bs}}, I
\end{array}
$$

- B is led to the mistaken belief that she shares the key with I.

Another attack!!

- Security Assumption: I might be a legitimate member of the community, for all we know.

Another attack!!

- Security Assumption: I might be a legitimate member of the community, for all we know.
- No principal knows that she is indeed a hacker.

Another attack!!

- Security Assumption: I might be a legitimate member of the community, for all we know.
- No principal knows that she is indeed a hacker.
- A leak ...

Msg 1. $A \rightarrow (I)S : A, B$
Msg 1. $I \rightarrow S : A, I$
Msg 2. $S \rightarrow I : \{k_{ai}\}_{k_{as}}, \{k_{ai}\}_{k_{is}}$
Msg 2. $(I)S \rightarrow A : \{k_{ai}\}_{k_{as}}, \{k_{ai}\}_{k_{is}}$
Msg 3. $A \rightarrow (I)B : \{k_{ai}\}_{k_{is}}, A$

Fixing the leak

- Include the identity of the partner in the messages.

Msg 1. $A \rightarrow S: A, B$

Msg 2. $S \rightarrow A: \{k_{ab}, B\}_{k_{as}}, \{k_{ab}, A\}_{k_{bs}}$

Msg 3. $A \rightarrow B: \{k_{ab}, A\}_{k_{bs}}$

Fixing the leak

- Include the identity of the partner in the messages.

Msg 1.	$A \rightarrow S: A, B$
Msg 2.	$S \rightarrow A: \{k_{ab}, B\}_{k_{as}}, \{k_{ab}, A\}_{k_{bs}}$
Msg 3.	$A \rightarrow B: \{k_{ab}, A\}_{k_{bs}}$

- Are I's previous attempts quelled by the new design?

Replays ...

- But I has a very old trick to counter this.

$$
\begin{array}{ll}
\text{Msg 1.} & A \rightarrow S : A, B \\
\text{Msg 2.} & (I)S \rightarrow A : \{k'_{ab}, B\}_{k_{as}}, \{k'_{ab}, A\}_{k_{bs}} \\
\text{Msg 3.} & A \rightarrow B : \{k'_{ab}, A\}_{k_{bs}}
\end{array}
$$

Replays ...

- But I has a very old trick to counter this.

$$\begin{array}{ll} \text{Msg 1.} & A \rightarrow S : A, B \\ \text{Msg 2.} & (I)S \rightarrow A : \{k'_{ab}, B\}_{k_{as}}, \{k'_{ab}, A\}_{k_{bs}} \\ \text{Msg 3.} & A \rightarrow B : \{k'_{ab}, A\}_{k_{bs}} \end{array}$$

- Security Assumption: I is able to obtain the value of the session key k_{ab} used in any sufficiently old previous run of the protocol.

Why is the replay attack bad?

- If I manages to break k'_{ab} in the time between the two sessions, A and B are in for a lot of trouble!

Why is the replay attack bad?

- If I manages to break k'_{ab} in the time between the two sessions, A and B are in for a lot of trouble!
- Even if the key is not broken, what if the next message in the original session was $\{\text{“Deposit Rs. 10000 from my account into } I\text{'s”}\}_{k'_{ab}}$?

Why is the replay attack bad?

- If I manages to break k'_{ab} in the time between the two sessions, A and B are in for a lot of trouble!
- Even if the key is not broken, what if the next message in the original session was $\{\text{“Deposit Rs. 10000 from my account into } I\text{'s”}\}_{k'_{ab}}$?
- Enables I to replay it in the later session, with obvious effects.

Challenge-response with nonces

- A nonce is a random value generated by one party and returned to that party to show that a message is newly generated.

Challenge-response with nonces

- A nonce is a random value generated by one party and returned to that party to show that a message is newly generated.
- The Needham-Schröder shared-key protocol

Msg 1. $A \rightarrow S : A, B, m$

Msg 2. $S \rightarrow A : \{k_{ab}, B, m, \{k_{ab}, A\}_{k_{bs}}\}_{k_{as}}$

Msg 3. $A \rightarrow B : \{k_{ab}, A\}_{k_{bs}}, A$

Msg 4. $B \rightarrow A : \{n\}_{k_{ab}}$

Msg 5. $A \rightarrow B : \{n - 1\}_{k_{ab}}$

Denning-Sacco

- B does not have direct contact with S …

Denning-Sacco

- B does not have direct contact with S ...
- and therefore ...

$$
\begin{array}{ll}
\text{Msg 3.} & (I)A \rightarrow B : \{k'_{ab}, A\}_{k_{bs}} \\
\text{Msg 4.} & B \rightarrow (I)A : \{n\}_{k'_{ab}} \\
\text{Msg 5.} & (I)A \rightarrow B : \{n - 1\}_{k'_{ab}}
\end{array}
$$

A fifth attempt

- Let B get a freshness assurance directly from S.

$$\begin{array}{ll}
\text{Msg 1.} & B \rightarrow A : B, n \\
\text{Msg 2.} & A \rightarrow S : A, B, m, n \\
\text{Msg 3.} & S \rightarrow A : \{k_{ab}, B, m\}_{k_{as}}, \{k_{ab}, A, n\}_{k_{bs}} \\
\text{Msg 4.} & A \rightarrow B : \{k_{ab}, A, n\}_{k_{bs}}
\end{array}$$

A fifth attempt

- Let B get a freshness assurance directly from S.

$$
\begin{array}{ll}
\text{Msg 1.} & B \rightarrow A : B, n \\
\text{Msg 2.} & A \rightarrow S : A, B, m, n \\
\text{Msg 3.} & S \rightarrow A : \{k_{ab}, B, m\}_{k_{as}}, \{k_{ab}, A, n\}_{k_{bs}} \\
\text{Msg 4.} & A \rightarrow B : \{k_{ab}, A, n\}_{k_{bs}}
\end{array}
$$

- Is this protocol “correct”?

A fifth attempt

- Let B get a freshness assurance directly from S.

$$
\begin{array}{ll}
\text{Msg 1.} & B \rightarrow A : B, n \\
\text{Msg 2.} & A \rightarrow S : A, B, m, n \\
\text{Msg 3.} & S \rightarrow A : \{k_{ab}, B, m\}_{k_{as}}, \{k_{ab}, A, n\}_{k_{bs}} \\
\text{Msg 4.} & A \rightarrow B : \{k_{ab}, A, n\}_{k_{bs}}
\end{array}
$$

- Is this protocol "correct"?
- How do we prove correctness of protocols in general?