

Advanced Cryptography

1st Semester 2007-2008

Passive Intruder

Pascal Lafourcade

Université Joseph Fourier, Verimag

Master: November 5th 2007

Last Time (I)

Symmetric encryption and Protocols

- ECB, CBC, FBC, OFB
- Attack on ECB
- Hybrid Encryption
- OAEP
- Logical Attacks
- Needham Schroeder

Remarks, questions, comments ?

Last Time (II)

Exercises done

- 1) ElGamal
- 2) CBC Attacks

Outline of Today:

- 1 Needham Schroeder
- 2 Dolev Yao's Intruder
- 3 Indecidability for unbounded number of sessions
- 4 Notion of Locality
- 5 Passive Intruder: Intruder Deduction Problem
- 6 Diffie-Hellman
- 7 Conclusion

Outline

- 1 Needham Schroeder
- 2 Dolev Yao's Intruder
- 3 Indecidability for unbounded number of sessions
- 4 Notion of Locality
- 5 Passive Intruder: Intruder Deduction Problem
- 6 Diffie-Hellman
- 7 Conclusion

Example: Needham-Schroeder Protocol 1978



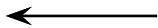
$\{N_A, A\}_{K_B}$



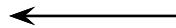
Example: Needham-Schroeder Protocol 1978



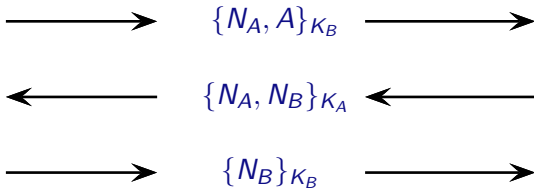
$\{N_A, A\}_{K_B}$



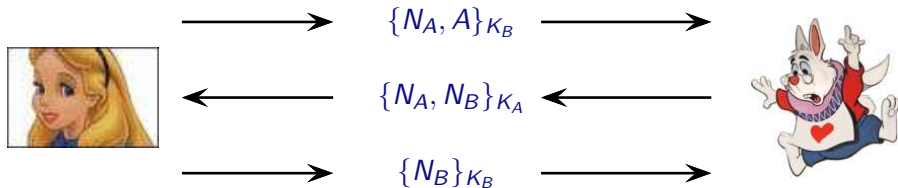
$\{N_A, N_B\}_{K_A}$



Example: Needham-Schroeder Protocol 1978



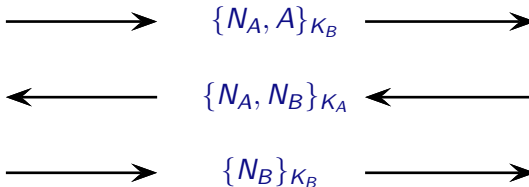
Example: Needham-Schroeder Protocol 1978



Question

- Is N_B a shared secret between A et B ?

Example: Needham-Schroeder Protocol 1978



Question

- Is N_B a shared secret between A et B ?

Answer

- In 1995, G.Lowe find an attack **17 years** after its publication!

Low Attack on the Needham-Schroeder

so-called “Man in the middle attack”



Agent A



Intruder I



Agent B

$$\begin{aligned} A &\longrightarrow B : \{A, N_a\}_{K_B} \\ B &\longrightarrow A : \{N_a, N_b\}_{K_A} \\ A &\longrightarrow B : \{N_b\}_{K_B} \end{aligned}$$

Low Attack on the Needham-Schroeder

so-called "Man in the middle attack"



Agent A

$\{A, N_a\}_{K_I}$ →



Intruder I

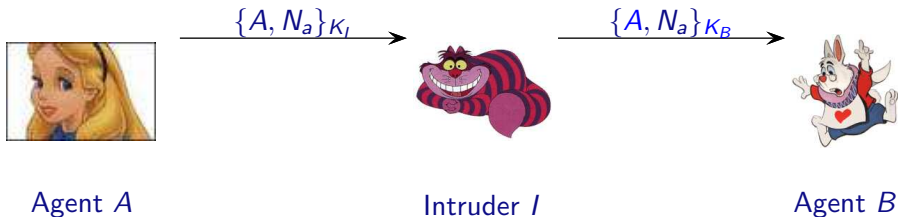


Agent B

- $A \longrightarrow B : \{A, N_a\}_{K_B}$
 $B \longrightarrow A : \{N_a, N_b\}_{K_A}$
 $A \longrightarrow B : \{N_b\}_{K_B}$

Low Attack on the Needham-Schroeder

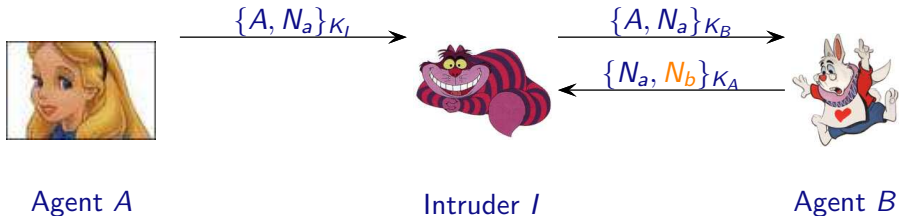
so-called “Man in the middle attack”



- $A \longrightarrow B : \{A, N_a\}_{K_B}$
 $B \longrightarrow A : \{N_a, N_b\}_{K_A}$
 $A \longrightarrow B : \{N_b\}_{K_B}$

Low Attack on the Needham-Schroeder

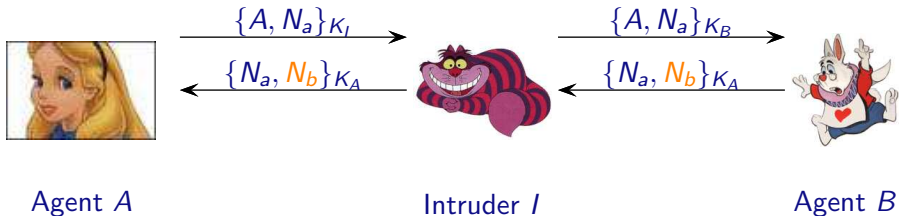
so-called “Man in the middle attack”



- $A \longrightarrow B : \{A, N_a\}_{K_B}$
- $B \longrightarrow A : \{N_a, N_b\}_{K_A}$
- $A \longrightarrow B : \{N_b\}_{K_B}$

Low Attack on the Needham-Schroeder

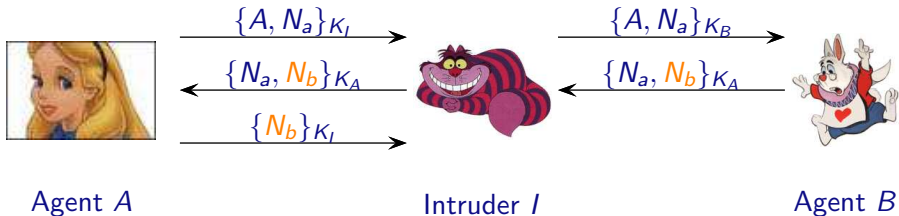
so-called “Man in the middle attack”



- $A \longrightarrow B : \{A, N_a\}_{K_B}$
- $B \longrightarrow A : \{N_a, N_b\}_{K_A}$
- $A \longrightarrow B : \{N_b\}_{K_B}$

Low Attack on the Needham-Schroeder

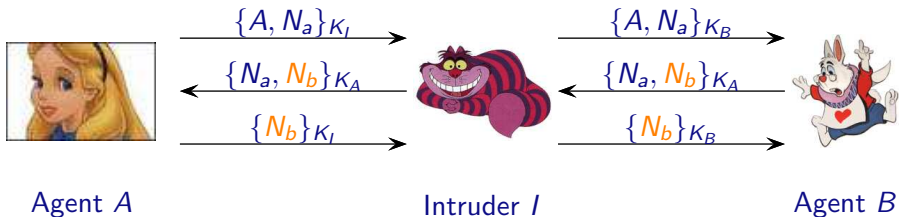
so-called “Man in the middle attack”



- $A \longrightarrow B : \{A, N_a\}_{K_B}$
- $B \longrightarrow A : \{N_a, N_b\}_{K_A}$
- $A \longrightarrow B : \{N_b\}_{K_B}$

Low Attack on the Needham-Schroeder

so-called "Man in the middle attack"



- $A \longrightarrow B : \{A, N_a\}_{K_B}$
- $B \longrightarrow A : \{N_a, N_b\}_{K_A}$
- $A \longrightarrow B : \{N_b\}_{K_B}$

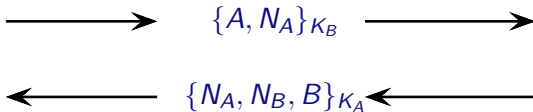
Needham-Schroeder corrected by Lowe 1995



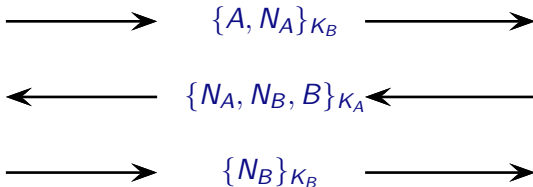
$\{A, N_A\}_{K_B}$



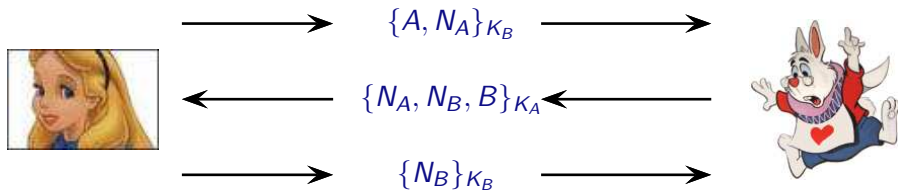
Needham-Schroeder corrected by Lowe 1995



Needham-Schroeder corrected by Lowe 1995



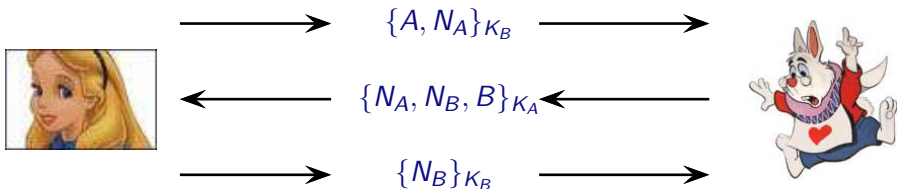
Needham-Schroeder corrected by Lowe 1995



Question

- This time the protocol is secure?

Needham-Schroeder corrected by Lowe 1995



Question

- This time the protocol is secure?

Answer

- There exists a type flaw attack.

Type flaw attacks

- A message consists of a sequence of submessages.
Examples: a principal's name, a nonce, a key, ...

- Messages sent as bit strings. No type information.

1011 0110 0010 1110 0011 0111 1010 0000

- **Type flaw** is when $A \rightarrow B : M$ and B accepts M as valid but parses it differently. I.e., B interprets the bits differently than A .
- **Example:** two 16-bit nonces $\{N_A, N_B\}$ could be mistaken as a 32-bit shared key.
Let's consider several examples from actual protocols.

Type Flaw Attack on the Needham-Schroeder-Lowe



Agent A



Intruder I



Agent B

$$\begin{aligned} A &\longrightarrow B : \{A, N_a\}_{K_B} \\ B &\longrightarrow A : \{N_a, N_b, B\}_{K_A} \\ A &\longrightarrow B : \{N_b\}_{K_B} \end{aligned}$$

Type Flaw Attack on the Needham-Schroeder-Lowe



Agent A



Intruder I

$\{A, I\}_{K_R}$



Agent B

- $A \longrightarrow B : \{A, N_a\}_{K_B}$
 $B \longrightarrow A : \{N_a, N_b, B\}_{K_A}$
 $A \longrightarrow B : \{N_b\}_{K_B}$

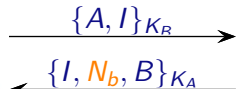
Type Flaw Attack on the Needham-Schroeder-Lowe



Agent A



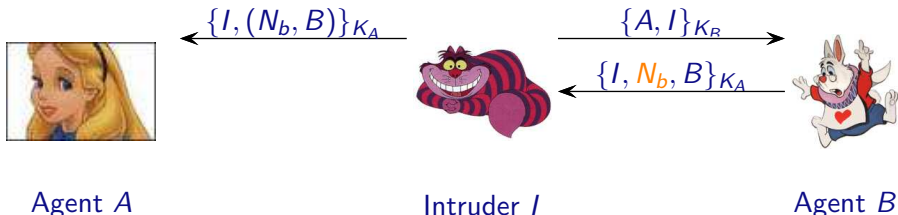
Intruder I



Agent B

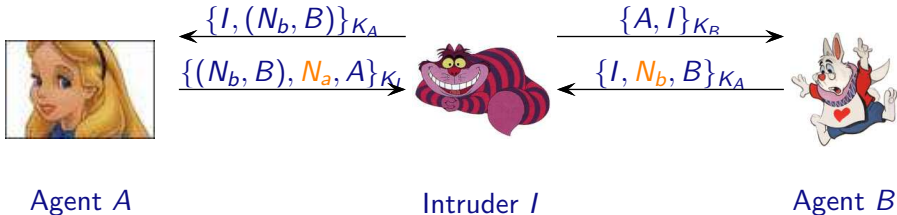
- $A \longrightarrow B : \{A, N_a\}_{K_B}$
- $B \longrightarrow A : \{N_a, N_b, B\}_{K_A}$
- $A \longrightarrow B : \{N_b\}_{K_B}$

Type Flaw Attack on the Needham-Schroeder-Lowe



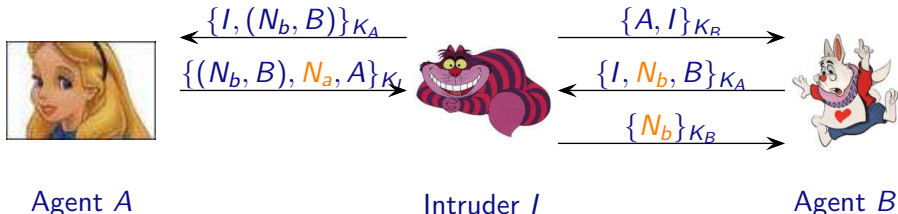
- $A \longrightarrow B : \{A, N_a\}_{K_B}$
 $B \longrightarrow A : \{N_a, N_b, B\}_{K_A}$
 $A \longrightarrow B : \{N_b\}_{K_B}$

Type Flaw Attack on the Needham-Schroeder-Lowe



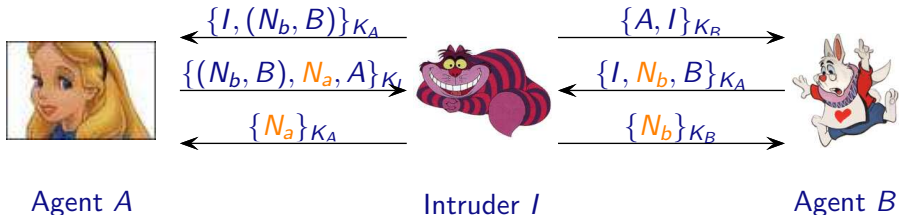
- $A \longrightarrow B : \{A, N_a\}_{K_B}$
 $B \longrightarrow A : \{N_a, N_b, B\}_{K_A}$
 $A \longrightarrow B : \{N_b\}_{K_B}$

Type Flaw Attack on the Needham-Schroeder-Lowe



- $A \longrightarrow B : \{A, N_a\}_{K_B}$
- $B \longrightarrow A : \{N_a, N_b, B\}_{K_A}$
- $A \longrightarrow B : \{N_b\}_{K_B}$

Type Flaw Attack on the Needham-Schroeder-Lowe



- $A \longrightarrow B : \{A, N_a\}_{K_B}$
- $B \longrightarrow A : \{N_a, N_b, B\}_{K_A}$
- $A \longrightarrow B : \{N_b\}_{K_B}$

Examples of kinds of attack

- **Man-in-the-middle (or parallel sessions) attack:** pass messages through to another session $A \leftrightarrow I \leftrightarrow B$.
- **Replay (or freshness) attack:** record and later re-introduce a message or part.
- **Reflection attack:** send transmitted information back to originator.
- **Oracle attack:** take advantage of normal protocol responses as encryption and decryption “services”.
- **Type flaw (confusion) attack:** substitute a different type of message field (e.g. a key vs. a name).

Questions?

How can we find such attacks?

- Models for Protocols
- Models for Properties
- Theories
- Dedicated Techniques
- Tools
 - Automatic
 - Semi-automatic

Why is it difficult to verify such protocols?

- Messages: Size not bounded
- Nonces: Arbitrary number
- Channel: Unsecure
- Intruder: Unlimited capabilities
- Instances: Unbounded numbers of principals
- Interleaving: Unlimited applications of the protocol.

Outline

- 1 Needham Schroeder
- 2 Dolev Yao's Intruder**
- 3 Indecidability for unbounded number of sessions
- 4 Notion of Locality
- 5 Passive Intruder: Intruder Deduction Problem
- 6 Diffie-Hellman
- 7 Conclusion

The Intruder is the Network (Worst Case)



The Intruder is the Network (Worst Case)



Listen

Passive: Intruder deduction problem

The Intruder is the Network (Worst Case)



Passive: Intruder deduction problem

Active: Security problem

Listen

Intercept message

(Re)play message

Delete message

The Intruder is the Network (Worst Case)



Listen

Intercept message

(Re)play message

Delete message

Passive: Intruder deduction problem

Active: Security problem

Intruder Capabilities (Dolev-Yao Model 80's)

- Encryption, Decryption with a key
- Pairing, Projection.

Proof System

A **sequent** is an expression of the form $T \vdash u$.

Definition

A **proof** of a sequent $T \vdash u$ is a tree whose nodes are labeled by either sequents or expressions of the form " $v \in T$ ", such that:

- Each leaf is labeled by an expression of the form $v \in T$, and each non-leaf node is labeled by an sequent.
- Each node labeled by a sequent $T \vdash v$ has n children labeled by $T \vdash s_1, \dots, T \vdash s_n$ such that there is an instance of an inference rule with conclusion $T \vdash_E v$ and **hypotheses** $T \vdash s_1, \dots, T \vdash s_n$.
- The **root** of the tree is labeled by $T \vdash u$.

A **subproof** of a proof P is a subtree of P .

Notions for Proof System

Definition

- **Size of a proof** P of $T \vdash u$ is denoted by $|P|$, is the number of nodes in the proof.
- A proof P of $T \vdash u$ is **minimal** if there does not exist a proof P' of $T \vdash u$ such that $|P'| < |P|$.
- A **simple proof** is a proof where each node $T \vdash v$ occurs at most once on each branch.

P Minimal $\Rightarrow P$ Simple

If P is a minimal proof of $T \vdash u$ then P is a simple proof of $T \vdash u$.

Dolev-Yao Deduction System

Deduction System : $T_0 \vdash^? s$

$$(A) \quad \frac{u \in T_0}{T_0 \vdash u}$$

$$(UL) \quad \frac{T_0 \vdash \langle u, v \rangle}{T_0 \vdash u}$$

$$(P) \quad \frac{T_0 \vdash u \quad T_0 \vdash v}{T_0 \vdash \langle u, v \rangle}$$

$$(UR) \quad \frac{T_0 \vdash \langle u, v \rangle}{T_0 \vdash v}$$

$$(C) \quad \frac{T_0 \vdash u \quad T_0 \vdash v}{T_0 \vdash \{u\}_v}$$

$$(D) \quad \frac{T_0 \vdash \{u\}_v \quad T_0 \vdash v}{T_0 \vdash u}$$

Example: $T_0 \vdash^? s$

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = b$

Example: $T_0 \vdash? s$

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = b$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k} \\
 (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array} \\
 \hline
 (D) \frac{\begin{array}{c}
 (A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \\
 (D) \frac{}{T_0 \vdash c}
 \end{array}}{T_0 \vdash b}
 \end{array}$$

Exercise: $T_0 \vdash? s$

Is it possible from T_0 to deduce s

- $T_0 = \{a, k\}$ and $s = \langle a, \{a\}_k \rangle$
- $T_0 = \{a, k\}$ and $s = \langle b, \{k\}_a \rangle$
- $T_0 = \{\{k\}_a, b\}$ and $s = \langle \{b\}_{\{k\}_a}, \{k\}_a \rangle$
- $T_0 = \{\langle a, \{k\}_a \rangle\}$ and $s = \{\langle a, \{k\}_a \rangle\}_k$

Dolev-Yao 1982

- Intruder controls the network and can:
 - intercept messages
 - modify messages
 - block messages
 - generate new messages
 - insert new messages
- Perfect cryptography:
 - Abstraction with terms algebra
 - Decryption only if inverse key is known
- Protocol has
 - Arbitrary number of principals
 - Arbitrary number of parallel sessions
 - Messages with arbitrary size

Outline

- 1 Needham Schroeder
- 2 Dolev Yao's Intruder
- 3 Indecidability for unbounded number of sessions**
- 4 Notion of Locality
- 5 Passive Intruder: Intruder Deduction Problem
- 6 Diffie-Hellman
- 7 Conclusion

Main Results

In general security problem **undecidable** [DLMS'99, AC'01]

Bounded number of session \Rightarrow **Decidability** [AL'00, RT'01]

Undecidability

Definition (Post Correspondance Problem (PCP))

Let Σ be a finite alphabet.

Input : Sequence of pairs $\langle u_i, v_i \rangle_{1 \leq i \leq n}$ $u_i, v_i \in \Sigma^*$, $n \in \mathbb{N}$

Question : Existence of $k, i_1, \dots, i_k \in \mathbb{N}$ such that

$$u_{i_1} \dots u_{i_k} = v_{i_1} \dots v_{i_k}?$$

Undecidability

Definition (Post Correspondance Problem (PCP))

Let Σ be a finite alphabet.

Input : Sequence of pairs $\langle u_i, v_i \rangle_{1 \leq i \leq n}$ $u_i, v_i \in \Sigma^*$, $n \in \mathbb{N}$

Question : Existence of $k, i_1, \dots, i_k \in \mathbb{N}$ such that

$$u_{i_1} \dots u_{i_k} = v_{i_1} \dots v_{i_k}?$$

Example

u_1	u_2	u_3	u_4	v_1	v_2	v_3	v_4
<i>aba</i>	<i>bbb</i>	<i>aab</i>	<i>bb</i>	<i>a</i>	<i>aaa</i>	<i>abab</i>	<i>babba</i>

Solution: **1431**

$$u_1 \cdot u_4 \cdot u_3 \cdot u_1 = aba \cdot bb \cdot aab \cdot aba = a \cdot babba \cdot abab \cdot a = v_1 \cdot v_4 \cdot v_3 \cdot v_1$$

But no solution for $\langle \mathbf{u}_1, \mathbf{v}_1 \rangle, \langle \mathbf{u}_2, \mathbf{v}_2 \rangle, \langle \mathbf{u}_3, \mathbf{v}_3 \rangle$

Undecidability

Definition (Post Correspondance Problem (PCP))

Let Σ be a finite alphabet.

Input : Sequence of pairs $\langle u_i, v_i \rangle_{1 \leq i \leq n}$ $u_i, v_i \in \Sigma^*$, $n \in \mathbb{N}$

Question : Existence of $k, i_1, \dots, i_k \in \mathbb{N}$ such that

$$u_{i_1} \dots u_{i_k} = v_{i_1} \dots v_{i_k}?$$

Example

u_1	u_2	u_3	u_4	v_1	v_2	v_3	v_4
<i>aba</i>	<i>bbb</i>	<i>aab</i>	<i>bb</i>	<i>a</i>	<i>aaa</i>	<i>abab</i>	<i>babba</i>

Solution: **1431**

$$u_1 \cdot u_4 \cdot u_3 \cdot u_1 = aba \cdot bb \cdot aab \cdot aba = a \cdot babba \cdot abab \cdot a = v_1 \cdot v_4 \cdot v_3 \cdot v_1$$

But no solution for $\langle \mathbf{u}_1, \mathbf{v}_1 \rangle, \langle \mathbf{u}_2, \mathbf{v}_2 \rangle, \langle \mathbf{u}_3, \mathbf{v}_3 \rangle$

PCP is undecidable

Undecidability for Protocols

We construct a protocol such that decidability of secret implies decidability of PCP.

$$A : \text{send}(\{\langle u_i, v_i \rangle\}_{K_{ab}}) \quad (1 \leq i \leq n)$$

$$B : \text{receive}(\{\langle x, y \rangle\}_{K_{ab}}) \\ \text{send}(\{\langle \langle x \cdot u_i, y \cdot v_i \rangle \rangle_{K_{ab}}, \{s\}_{\langle \langle x \cdot u_i, x \cdot u_i \rangle \rangle_{K_{ab}}}\}) \quad (1 \leq i \leq n)$$

We assume that \mathbf{K}_{AB} is a shared key between **A** and **B**.

Intruder can find \mathbf{s} iff he can solve PCP.

Outline

- 1 Needham Schroeder
- 2 Dolev Yao's Intruder
- 3 Indecidability for unbounded number of sessions
- 4 Notion of Locality**
- 5 Passive Intruder: Intruder Deduction Problem
- 6 Diffie-Hellman
- 7 Conclusion

Syntactic Subterms

Equivalent definition for Dolev Yao model

$S(t)$ is the smallest set such that:

- $t \in S(t)$
- $\langle u, v \rangle \in S(t) \Rightarrow u, v \in S(t)$
- $\{u\}_v \in S(t) \Rightarrow u, v \in S(t)$

Exercise:

- Let $t = \{\langle a, \{b\}_{k_2} \rangle\}_{k_1}$

Syntactic Subterms

Equivalent definition for Dolev Yao model

$S(t)$ is the smallest set such that:

- $t \in S(t)$
- $\langle u, v \rangle \in S(t) \Rightarrow u, v \in S(t)$
- $\{u\}_v \in S(t) \Rightarrow u, v \in S(t)$

Exercise:

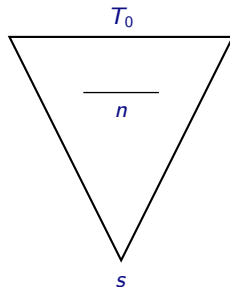
- Let $t = \{\langle a, \{b\}_{k_2} \rangle\}_{k_1}$

$$S(t) = \{t, a, b, k_1, k_2, \{b\}_{k_2}, \langle a, \{b\}_{k_2} \rangle\}$$

Definition of S-Locality

- A proof P of $T_0 \vdash s$ is S-local :

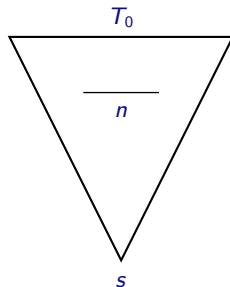
$$\forall n \in P, n \in S(T_0 \cup \{s\})$$



Definition of S-Locality

- A proof P of $T_0 \vdash s$ is S-local :

$$\forall n \in P, n \in S(T_0 \cup \{s\})$$



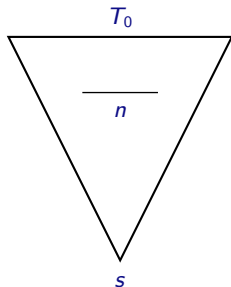
S-Local Proof:

A proof P of $T \vdash w$ is **S-local** if all nodes are in $S(T \cup \{w\})$.

Definition of S-Locality

- A proof P of $T_0 \vdash s$ is S-local :

$$\forall n \in P, n \in S(T_0 \cup \{s\})$$



S-Local Proof:

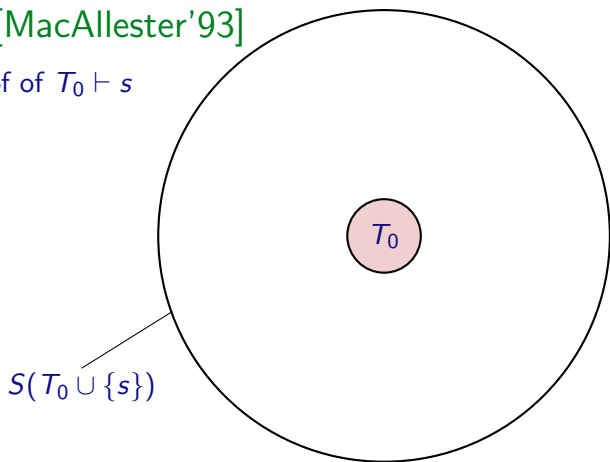
A proof P of $T \vdash w$ is **S-local** if all nodes are in $S(T \cup \{w\})$.

S-Locality :

A proof system is **S-local** if whenever there is a proof of $T \vdash w$ then there is also a S-local proof of $T \vdash w$.

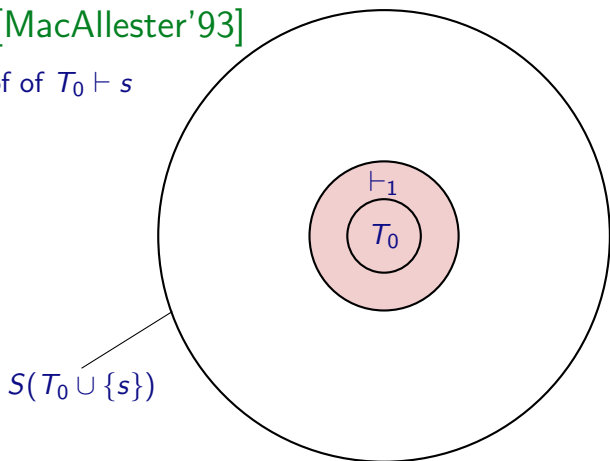
Locality Idea [MacAllester'93]

P a S -local proof of $T_0 \vdash s$



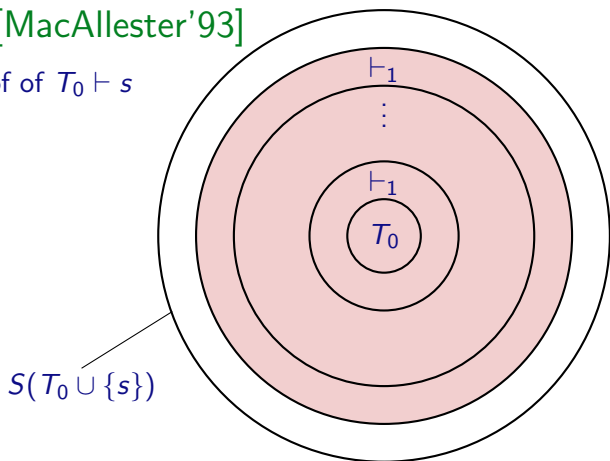
Locality Idea [MacAllester'93]

P a S-local proof of $T_0 \vdash s$



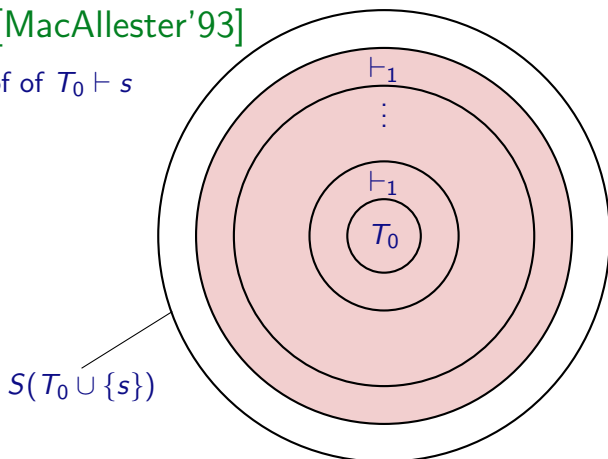
Locality Idea [MacAllester'93]

P a S-local proof of $T_0 \vdash s$



Locality Idea [MacAllester'93]

P a S-local proof of $T_0 \vdash s$



Intruder Deduction Problem : $T_0 \vdash^? s$

- S-locality
- One-step deductibility

Example: $T_0 \vdash s$ is it a local proof?

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = b$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k} \\
 (D) \frac{}{T_0 \vdash c}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c}
 \end{array} \\
 (D) \frac{}{T_0 \vdash b}
 \end{array}$$

Example: $T_0 \vdash s$ is it a local proof?

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = b$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k} \\
 (D) \frac{}{T_0 \vdash c}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{k \in T_0}{T_0 \vdash k} \\
 (A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c}
 \end{array}
 \end{array}
 \quad
 \frac{}{T_0 \vdash b}$$

$$S(T_0 \cup \{s\}) = T_0 \cup \{a, b, c, \{c\}_k\}$$

Locality Theorem

Theorem of Locality [McAllester 93]

If a proof system P is SyntacticSubterm-local then there is a P -time procedure to decide the deductibility in P .

Locality Theorem

Theorem of Locality [McAllester 93]

If a proof system P is SyntacticSubterm-local then there is a P -time procedure to decide the deductibility in P .

Restrictions:

- Deduction system must be finite
- Use just syntactic subterms

Adapted McAllester Results

McAllester's Algorithm

Input : T_0, w

$T \leftarrow T_0;$

while $(\exists s \in S(T_0, w)$ such that $T \vdash^{\leq 1} s$ and $s \notin T)$

$T \leftarrow T \cup \{s\};$

Output : $w \in T$

Theorem

Let be P a proof system, if:

- the size of $S(T)$ is polynomial in the size of T ,
- P is S-local,
- one-step deducibility is P-time decidable,

then provability in the proof system P is P-time decidable.

Outline

- 1 Needham Schroeder
- 2 Dolev Yao's Intruder
- 3 Indecidability for unbounded number of sessions
- 4 Notion of Locality
- 5 Passive Intruder: Intruder Deduction Problem**
- 6 Diffie-Hellman
- 7 Conclusion

Locality Theorem

Theorem of Locality [McAllester 93]

If a proof system P is SyntacticSubterm-local then there is a P -time procedure to decide the deductibility in P .

Locality Theorem

Theorem of Locality [McAllester 93]

If a proof system P is SyntacticSubterm-local then there is a P -time procedure to decide the deductibility in P .

Result:

Dolev Yao deduction system is S-local.

Example I

GOAL: Find a good S !

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = b$

Example I

GOAL: Find a good S !

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = b$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \\
 (D) \frac{}{T_0 \vdash b}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k} \\
 (D) \frac{}{T_0 \vdash c}
 \end{array}
 \quad
 \begin{array}{c}
 (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array}
 \end{array}
 \quad
 \frac{}{T_0 \vdash b}$$

Example I

GOAL: Find a good S !

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = b$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k}
 \end{array} \\
 \begin{array}{c}
 (A) \frac{k \in T_0}{T_0 \vdash k} \\
 (D) \frac{}{T_0 \vdash c}
 \end{array} \\
 \hline
 (D) \frac{\begin{array}{c} (A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \\ (D) \frac{}{T_0 \vdash b} \end{array}}{T_0 \vdash b}
 \end{array}$$

$S(T_0) = T_0 \cup \{s, a, b, c, d, k, \{a\}_k, \{b\}_c, \{c\}_k\}$

Example II

GOAL: Find a good S .

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = \langle b, k \rangle$

Example II

GOAL: Find a good S .

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = \langle b, k \rangle$

$$\begin{array}{c}
 \begin{array}{c}
 (A) \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 (UR) \frac{}{T_0 \vdash \{c\}_k} \\
 (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array} \\
 \hline
 (D) \frac{(A) \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \quad (D) \frac{}{T_0 \vdash c}}{T_0 \vdash b} \\
 \hline
 (P) \frac{}{T_0 \vdash \langle b, k \rangle} \quad (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array}$$

Example II

GOAL: Find a good S .

Example

$T_0 = \{k, \{b\}_c, \langle a, \{c\}_k \rangle\}$ and $s = \langle b, k \rangle$

$$\begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \text{(A)} \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle} \\
 \text{(UR)} \frac{\text{(A)} \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle}}{T_0 \vdash \{c\}_k} \\
 \text{(A)} \frac{k \in T_0}{T_0 \vdash k} \\
 \text{(D)} \frac{\text{(A)} \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \quad \text{(D)} \frac{\text{(UR)} \frac{\text{(A)} \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle}}{T_0 \vdash \{c\}_k} \quad \text{(A)} \frac{k \in T_0}{T_0 \vdash k}}{T_0 \vdash c}}{T_0 \vdash b} \\
 \text{(P)} \frac{\text{(D)} \frac{\text{(A)} \frac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \quad \text{(D)} \frac{\text{(UR)} \frac{\text{(A)} \frac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle}}{T_0 \vdash \{c\}_k} \quad \text{(A)} \frac{k \in T_0}{T_0 \vdash k}}{T_0 \vdash c}}{T_0 \vdash b}}{T_0 \vdash \langle b, k \rangle} \\
 \text{(A)} \frac{k \in T_0}{T_0 \vdash k}
 \end{array}
 \end{array}
 \end{array}
 \end{array}$$

$S(T_0) = T_0 \cup \{s, a, b, c, k, \{b\}_k, \{c\}_k\}$ but $\langle b, k \rangle \notin S(T_0)$

It is Not enough

Notice that $\langle b, k \rangle \in S(T_0 \cup \{s\})$

Example III

GOAL: Find a good S .

Example

$T_0 = \{k, \{c\}_k\}$ and $s = c$

Example III

GOAL: Find a good S .

Example

$T_0 = \{k, \{c\}_k\}$ and $s = c$

$$\begin{array}{c}
 \frac{(A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k} (A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k}}{(P) \frac{}{T_0 \vdash \langle \{c\}_k, \{c\}_k \rangle}} \\
 \frac{(UL) \frac{}{T_0 \vdash \{c\}_k}}{(D) \frac{}{c}} \quad (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array}$$

Example III

GOAL: Find a good S .

Example

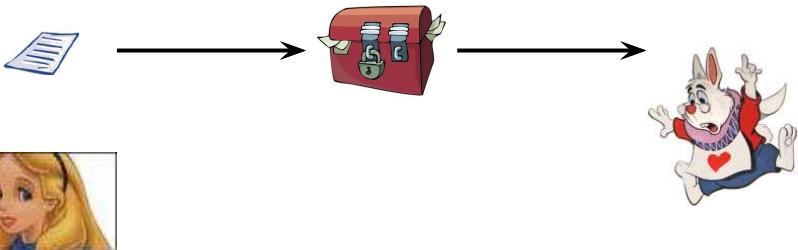
$T_0 = \{k, \{c\}_k\}$ and $s = c$

$$\begin{array}{c}
 \frac{(A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k} (A) \frac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k}}{(P) \frac{}{T_0 \vdash \langle \{c\}_k, \{c\}_k \rangle}} \\
 \frac{(UL) \frac{}{T_0 \vdash \{c\}_k}}{(D) \frac{}{c}} \quad (A) \frac{k \in T_0}{T_0 \vdash k}
 \end{array}$$

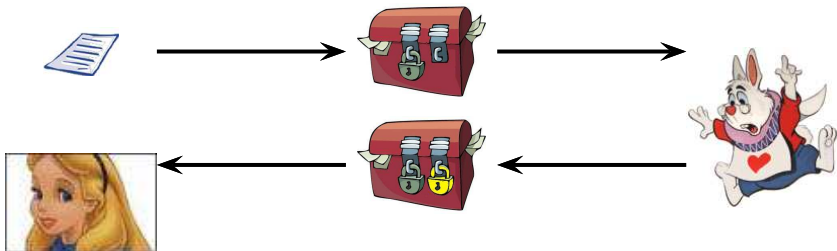
$S(T_0) = T_0 \cup \{c\}$ but $\langle \{c\}_k, \{c\}_k \rangle$

It is Not in $S(T_0)$

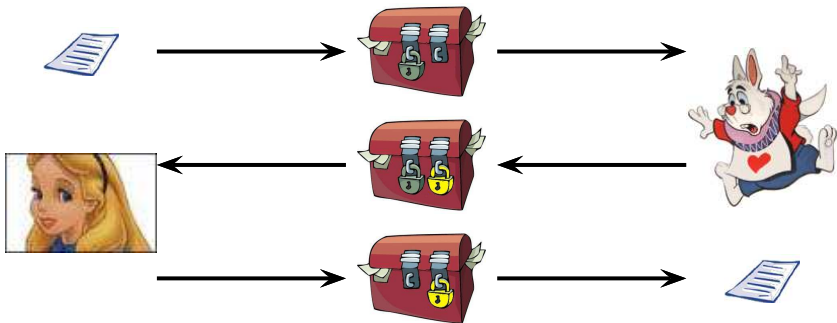
Example :



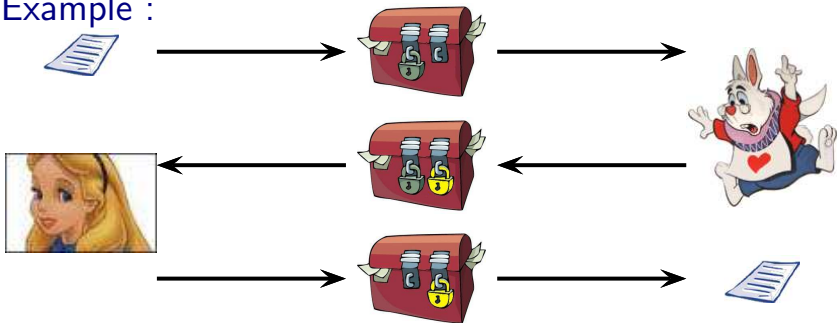
Example :



Example :



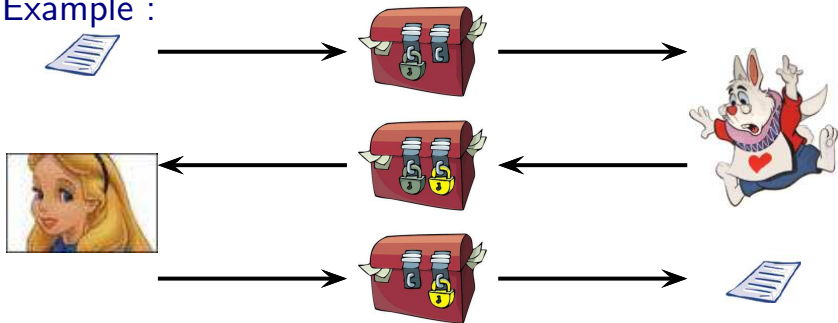
Example :



Shamir 3-Pass Protocol

$$1 \quad A \rightarrow B : \{m\}_{K_A}$$

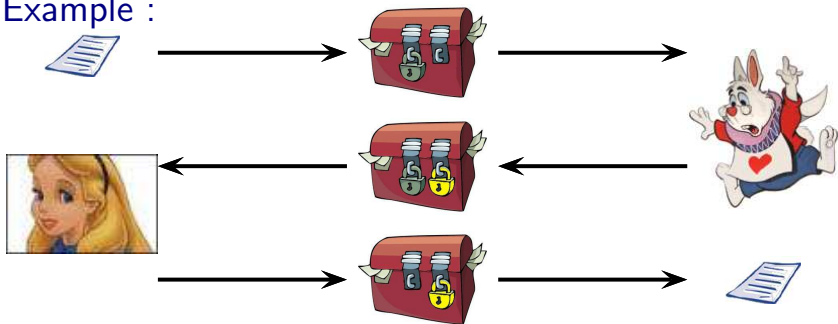
Example :



Shamir 3-Pass Protocol

- 1 $A \rightarrow B : \{m\}_{K_A}$
- 2 $B \rightarrow A : \{\{m\}_{K_A}\}_{K_B}$

Example :



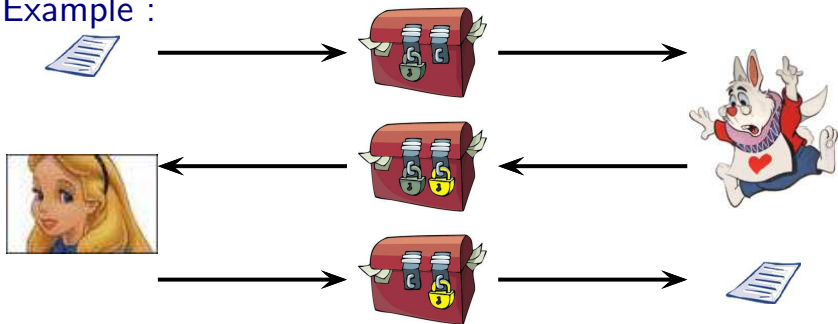
Shamir 3-Pass Protocol

$$1 \quad A \rightarrow B : \{m\}_{K_A}$$

$$2 \quad B \rightarrow A : \{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A}$$

Commutative
Encryption

Example :



Shamir 3-Pass Protocol

- 1 $A \rightarrow B : \{m\}_{K_A}$
- 2 $B \rightarrow A : \{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A}$
- 3 $A \rightarrow B : \{m\}_{K_B}$

Commutative
Encryption

Logical Attack on Shamir 3-Pass Protocol (I)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

XOR Properties (ACUN)

- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$
- $x \oplus y = y \oplus x$
- $x \oplus 0 = x$
- $x \oplus x = 0$

Associativity

Commutativity

Unity

Nilpotency

Logical Attack on Shamir 3-Pass Protocol (I)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

XOR Properties (ACUN)

- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

Associativity

- $x \oplus y = y \oplus x$

Commutativity

- $x \oplus 0 = x$

Unity

- $x \oplus x = 0$

Nilpotency

Vernam encryption is a **commutative encryption** :

$$\{\{m\}_{K_A}\}_{K_I} = (m \oplus K_A) \oplus K_I = (m \oplus K_I) \oplus K_A = \{\{m\}_{K_I}\}_{K_A}$$

Logical Attack on Shamir 3-Pass Protocol (II)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

Shamir 3-Pass Protocol



- 1 $A \rightarrow B : m \oplus K_A$
- 2 $B \rightarrow A : (m \oplus K_A) \oplus K_B$
- 3 $A \rightarrow B : m \oplus K_B$



Passive attacker :

$$m \oplus K_A \quad m \oplus K_B \oplus K_A \quad m \oplus K_B$$



Logical Attack on Shamir 3-Pass Protocol (II)

Perfect encryption one-time pad (Vernam Encryption)

$$\{m\}_k = m \oplus k$$

Shamir 3-Pass Protocol



- 1 $A \rightarrow B : m \oplus K_A$
- 2 $B \rightarrow A : (m \oplus K_A) \oplus K_B$
- 3 $A \rightarrow B : m \oplus K_B$



Passive attacker :

$$m \oplus K_A \oplus m \oplus K_B \oplus K_A \oplus m \oplus K_B = m$$

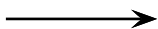


Outline

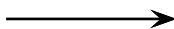
- 1 Needham Schroeder
- 2 Dolev Yao's Intruder
- 3 Indecidability for unbounded number of sessions
- 4 Notion of Locality
- 5 Passive Intruder: Intruder Deduction Problem
- 6 Diffie-Hellman**
- 7 Conclusion

The Diffie-Hellman protocol

g, p are public parameters.



$$g^x \bmod p$$



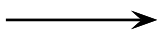
Diffie chooses x and computes $g^x \bmod p$.

Hellman chooses y and computes $g^y \bmod p$.

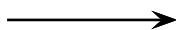
Basic Diffie-Hellman key-exchange: initiator I and responder R exchange public “half-keys” to arrive at mutual session key $k = g^{xy} \bmod p$.

The Diffie-Hellman protocol

g, p are public parameters.



$$g^x \bmod p$$



$$g^y \bmod p$$

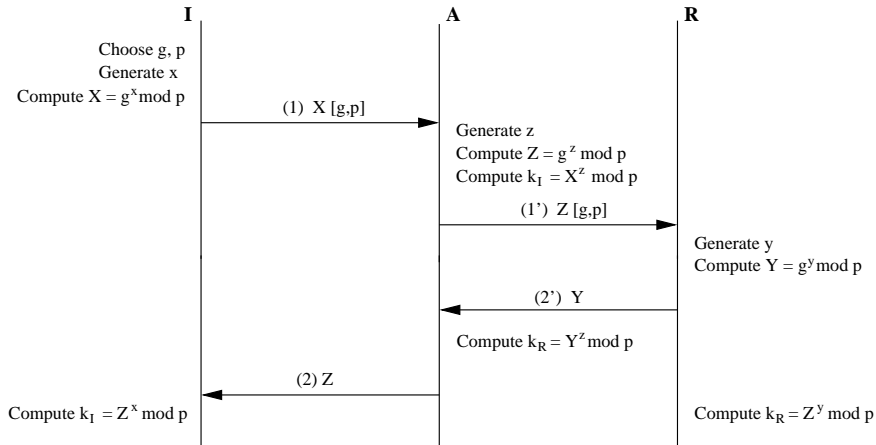


Diffie chooses x and computes $g^x \bmod p$.

Hellman chooses y and computes $g^y \bmod p$.

Basic Diffie-Hellman key-exchange: initiator I and responder R exchange public “half-keys” to arrive at mutual session key $k = g^{xy} \bmod p$.

Man-in-the-middle attack



Outline

- 1 Needham Schroeder
- 2 Dolev Yao's Intruder
- 3 Indecidability for unbounded number of sessions
- 4 Notion of Locality
- 5 Passive Intruder: Intruder Deduction Problem
- 6 Diffie-Hellman
- 7 Conclusion**

Summary

Today

- Continuous Control
- Needham Schroeder
- Dolev Yao Intruder
- Undecidability Result
- Locality
- Passive Intruder
- Diffie Hellman

Next Time

Playing with Tools

- Scyther
- Avispa: OFMC, CI-Atse, SATMC, TA4SP
- Proverif

Thank you for your attention.

Questions ?