Security Models
Lecture 3
Passive Intruder

Pascal Lafourcade

2020-2021

Outline of Today:

1. Logical Attacks

Outline of Today:

1. Logical Attacks
2. Diffie-Hellman

Outline of Today:

1. Logical Attacks
2. Diffie-Hellman
3. Needham Schroeder

Outline of Today:

1. Logical Attacks
2. Diffie-Hellman
3. Needham Schroeder
4. Dolev Yao's Intruder

Outline of Today:

1. Logical Attacks
2. Diffie-Hellman
3. Needham Schroeder
4. Dolev Yao's Intruder
5. Undecidability for unbounded number of sessions

Outline of Today:

1. Logical Attacks
2. Diffie-Hellman
3. Needham Schroeder
4. Dolev Yao's Intruder
5. Undecidability for unbounded number of sessions
6. Notion of Locality

Outline of Today:

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

Outline

Attacks

Logical Attacks
Perfect cryptography
Computational vs symbolic

Simple Example

Replay message

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).

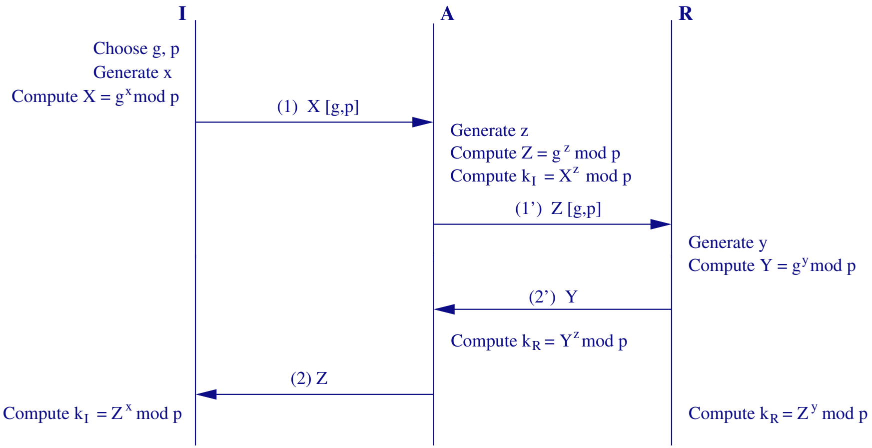
Outline

The Diffie-Hellman protocol

g, p are public parameters.

$$(g^y)^x \bmod p = k = g^{xy} \bmod p = (g^x)^y \bmod p$$

Man-in-the-middle attack



Outline

Messages Abstraction

- Names: A, B or Alice, Bob, ...
- Nonces: N_A. Fresh data.
- Keys: K and **inverse keys** K^{-1}
- Asymmetric Encryption: : $\{M\}_{K_A}$
- Symmetric Encryption: $\{M\}_{K_{AB}}$.
- Message concatenation: $\langle M_1, M_2 \rangle$.

Example: $\{\langle A \oplus N_B, K_{AB} \rangle\}_{K_B}$.

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?

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?

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!

Lowe Attack on the Needham-Schroeder

so-called “Man in the middle attack”

Needham-Schroeder corrected by Lowe 1995

Question

- This time the protocol is secure?

Type flaw attacks

- A message consists of a sequence of sub-messages.
  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

Another Type Flaw Attack: Otway-Rees Protocol

Otway-Rees

1 $A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
2 $B \rightarrow S : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
3 $S \rightarrow B : (M, (N_A, Kab)_{Kas}, (N_B, Kab)_{Kbs})$
4 $B \rightarrow A : (M, (N_A, Kab)_{Kas})$

where M is the session-identifier.

Another Type Flaw Attack: Otway-Rees Protocol

Otway-Rees

1 $A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
2 $B \rightarrow S : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
3 $S \rightarrow B : (M, (N_A, Kab)_{Kas}, (N_B, Kab)_{Kbs})$
4 $B \rightarrow A : (M, (N_A, Kab)_{Kas})$

where M is the session-identifier.

Attack

Another Type Flaw Attack: Otway-Rees Protocol

Otway-Rees

1 $A \to B : (M, A, B, (N_A, M, A, B)_{Kas})$
2 $B \to S : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
3 $S \to B : (M, (N_A, Kab)_{Kas}, (N_B, Kab)_{Kbs})$
4 $B \to A : (M, (N_A, Kab)_{Kas})$

where M is the session-identifier.

Attack

1 $A \to B : (M, A, B, (N_A, M, A, B)_{Kas})$

Another Type Flaw Attack: Otway-Rees Protocol

Otway-Rees

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow S : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
$3\ S \rightarrow B : (M, (N_A, Kab)_{Kas}, (N_B, Kab)_{Kbs})$
$4\ B \rightarrow A : (M, (N_A, Kab)_{Kas})$

where M is the session-identifier.

Attack

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow I(S) : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$

Another Type Flaw Attack: Otway-Rees Protocol

Otway-Rees

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow S : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
$3\ S \rightarrow B : (M, (N_A, Kab)_{Kas}, (N_B, Kab)_{Kbs})$
$4\ B \rightarrow A : (M, (N_A, Kab)_{Kas})$

where M is the session-identifier.

Attack

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow I(S) : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
$3\ I(S) \rightarrow B : (M, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
Kab = (M,A,B)

Another Type Flaw Attack: Otway-Rees Protocol

Otway-Rees

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow S : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
$3\ S \rightarrow B : (M, (N_A, Kab)_{Kas}, (N_B, Kab)_{Kbs})$
$4\ B \rightarrow A : (M, (N_A, Kab)_{Kas})$

where M is the session-identifier.

Attack

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow I(S) : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
$3\ I(S) \rightarrow B : (M, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
Kab = (M,A,B)$4\ B \rightarrow A : (M, (N_A, M, A, B)_{Kas})$

Another Type Flaw Attack: Otway-Rees Protocol

Otway-Rees

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow S : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
$3\ S \rightarrow B : (M, (N_A, Kab)_{Kas}, (N_B, Kab)_{Kbs})$
$4\ B \rightarrow A : (M, (N_A, Kab)_{Kas})$

where M is the session-identifier.

Attack

$1\ A \rightarrow B : (M, A, B, (N_A, M, A, B)_{Kas})$
$2\ B \rightarrow I(S) : (M, A, B, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
$3\ I(S) \rightarrow B : (M, (N_A, M, A, B)_{Kas}, (N_B, M, A, B)_{Kbs})$
Kab = (M,A,B)$4\ B \rightarrow A : (M, (N_A, M, A, B)_{Kas})$

Another Type Flaw Attack: Yahalom Protocol

Yahalom

1 $A \to B : (A, N_A)$
2 $B \to S : (B, (A, N_A, N_B)_{Kbs})$
3 $S \to A : ((B, Kab, N_A, N_B)_{Kas}, (A, Kab, N_B)_{Kbs})$
4 $A \to B : ((A, Kab, N_B)_{Kbs}, (N_B)_{Kab})$

Another Type Flaw Attack: Yahalom Protocol

Yahalom

1 $A \rightarrow B : (A, N_A)$
2 $B \rightarrow S : (B, (A, N_A, N_B)_{Kbs})$
3 $S \rightarrow A : ((B, Kab, N_A, N_B)_{Kas}, (A, Kab, N_B)_{Kbs})$
4 $A \rightarrow B : ((A, Kab, N_B)_{Kbs}, (N_B)_{Kab})$

Attack

1 $I(A) \rightarrow B : (A, N_A)$
2 $B \rightarrow I(S) : (B, (A, N_A, N_B)_{Kbs})$
4 $I(A) \rightarrow B : ((A, N_A, N_B)_{Kbs}, (N_B)_{N_A})$

Another Type Flaw Attack: Woo Lam Protocol

Woo Lam

1 $A \rightarrow B : (A)$
2 $B \rightarrow A : (N_B)$
3 $A \rightarrow B : (A, B, N_B)_{Kas}$
4 $B \rightarrow S : (A, B, (A, B, N_B)_{Kas})_{Kbs}$
5 $S \rightarrow B : (A, B, N_B)_{Kbs}$

Another Type Flaw Attack: Woo Lam Protocol

Woo Lam

1 $A \rightarrow B : (A)$
2 $B \rightarrow A : (N_B)$
3 $A \rightarrow B : (A, B, N_B)_{Kas}$
4 $B \rightarrow S : (A, B, (A, B, N_B)_{Kas})_{Kbs}$
5 $S \rightarrow B : (A, B, N_B)_{Kbs}$

Attack

1 $I(A) \rightarrow B : (A)$

Another Type Flaw Attack: Woo Lam Protocol

Woo Lam

1 $A \rightarrow B : (A)$
2 $B \rightarrow A : (N_B)$
3 $A \rightarrow B : (A, B, N_B)_{Kas}$
4 $B \rightarrow S : (A, B, (A, B, N_B)_{Kas})_{Kbs}$
5 $S \rightarrow B : (A, B, N_B)_{Kbs}$

Attack

1 $I(A) \rightarrow B : (A)$
2 $B \rightarrow I(A) : (N_B)$

Another Type Flaw Attack: Woo Lam Protocol

Woo Lam

1 $A \rightarrow B : (A)$
2 $B \rightarrow A : (N_B)$
3 $A \rightarrow B : (A, B, N_B)_{Kas}$
4 $B \rightarrow S : (A, B, (A, B, N_B)_{Kas})_{Kbs}$
5 $S \rightarrow B : (A, B, N_B)_{Kbs}$

Attack

1 $I(A) \rightarrow B : (A)$
2 $B \rightarrow I(A) : (N_B)$
3 $I(A) \rightarrow B : (N_B)$
instead of $(A, B, N_B)_{Kas}$

Another Type Flaw Attack: Woo Lam Protocol

Woo Lam

1 $A \rightarrow B : (A)$
2 $B \rightarrow A : (N_B)$
3 $A \rightarrow B : (A, B, N_B)_{Kas}$
4 $B \rightarrow S : (A, B, (A, B, N_B)_{Kas})_{Kbs}$
5 $S \rightarrow B : (A, B, N_B)_{Kbs}$

Attack

1 $I(A) \rightarrow B : (A)$
2 $B \rightarrow I(A) : (N_B)$
3 $I(A) \rightarrow B : (N_B)$
instead of $(A, B, N_B)_{Kas}$4 $B \rightarrow I(S) : (A, B, N_B)_{Kbs}$

Another Type Flaw Attack: Woo Lam Protocol

Woo Lam

1 $A \rightarrow B : (A)$
2 $B \rightarrow A : (N_B)$
3 $A \rightarrow B : (A, B, N_B)_{Kas}$
4 $B \rightarrow S : (A, B, (A, B, N_B)_{Kas})_{Kbs}$
5 $S \rightarrow B : (A, B, N_B)_{Kbs}$

Attack

1 $I(A) \rightarrow B : (A)$
2 $B \rightarrow I(A) : (N_B)$
3 $I(A) \rightarrow B : (N_B)$
instead of $(A, B, N_B)_{Kas}$4 $B \rightarrow I(S) : (A, B, N_B)_{Kbs}$
5 $I(S) \rightarrow B : (A, B, N_B)_{Kbs}$

Questions?

How can we find such attacks?

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

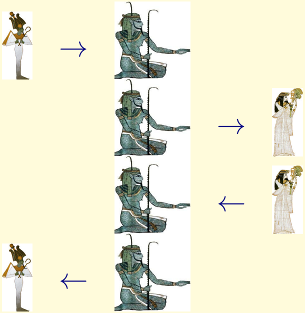
Why is it difficult to verify such protocols?

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

TMN Protocol: Distribution of a fresh symmetric key

[Tatebayashi, Matsuzuki, Newmann 89]:

O S I



$O \rightarrow S$: $O, I, \{N_O\}_{Pub_S}$

$S \rightarrow I$: S, O

$S \leftarrow I$: $I, O, \{N_I\}_{Pub_S}$

$O \leftarrow S$: $S, I, N_O \oplus N_I$

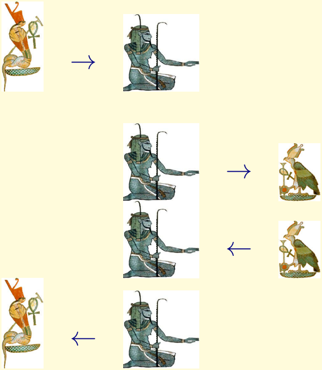
Osiris retrieves N_I:

Using $x \oplus x \oplus y = y$, knowing N_O

Attack on TMN Protocol [Simmons'94]

With homomorphic encryption $\{a\}_k \oplus \{b\}_k = \{a \oplus b\}_k$

$B \quad S \quad C$



$B \rightarrow S$: $B, C, \underbrace{\{N_I\}_{Pub_S} \oplus \{N_B\}_{Pub_S}}_{\{N_I \oplus N_B\}_{Pub_S}}$

$S \rightarrow C$: S, B

$S \leftarrow C$: $C, B, \{N_C\}_{Pub_S}$

$B \leftarrow S$: $S, (N_I \oplus N_B) \oplus N_C$

Buto Learns:

Using $x \oplus x \oplus y = y$, knowing N_I and N_C he deduces N_B

Outline

The Intruder is the Network (Worst Case)

Intruder Capabilities (Dolev-Yao Model 80's)

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

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

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 $T \vdash s_1, \ldots, T \vdash s_n$ such that there is an instance of an inference rule with conclusion $T \vdash_E v$ and **hypotheses** $T \vdash s_1, \ldots, T \vdash s_n$.
- 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 $|P'| < |P|$.

Dolev-Yao Deduction System

Deduction System : $T_0 \vdash^? s$

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

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

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

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

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

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

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 $s = b$

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

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 $s = \{\langle a, \{k\}_a \rangle\}_k$

Outline

Main Results

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

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

Undecidability

Definition (Post Correspondence 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 $k, i_1, \ldots, i_k \in \mathbb{N}$ such that $u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}$?

Undecidability

Definition (Post Correspondence 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 $k, i_1, \ldots, i_k \in \mathbb{N}$ such that $u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}$?

Example

u_1	u_2	u_3	u_4	v_1	v_2	v_3	v_4
aba	bbb	aab	bb	a	aaa	$abab$	$babba$

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 $\langle \mathrm{u}_1, \mathrm{v}_1 \rangle, \langle \mathrm{u}_2, \mathrm{v}_2 \rangle, \langle \mathrm{u}_3, \mathrm{v}_3 \rangle$

Undecidability

Definition (Post Correspondence 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 $k, i_1, \ldots, i_k \in \mathbb{N}$ such that
$u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}$?

Example

u_1	u_2	u_3	u_4	v_1	v_2	v_3	v_4
aba	bbb	aab	bb	a	aaa	$abab$	$babba$

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 $\langle \mathrm{u}_1, \mathrm{v}_1 \rangle, \langle \mathrm{u}_2, \mathrm{v}_2 \rangle, \langle \mathrm{u}_3, \mathrm{v}_3 \rangle$

PCP is undecidable

Undecidability for Protocols

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

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

$$\begin{array}{ll} B: & receive(\{\langle x, y \rangle\}_{K_{ab}}) \\ & send(\langle \{\langle x \cdot u_i, y \cdot v_i \rangle\}_{K_{ab}}, \{s\}_{\langle \{\langle x \cdot u_i, x \cdot u_i \rangle\}_{K_{ab}} \rangle}) \qquad (1 \leq i \leq n) \end{array}$$

We assume that K_{AB} is a shared key between A and B.

Intruder can find s iff he can solve PCP.

Outline

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 :

Definition of S-Locality

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

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 :

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]

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

- S-locality
- One-step deductibility

Example: a local proof of $T_0 \vdash s$

Example

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

$$
(D)\dfrac{(D)\dfrac{(UR)\dfrac{(A)\dfrac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle}}{T_0 \vdash \{c\}_k} \qquad (A)\dfrac{k \in T_0}{T_0 \vdash k}}{T_0 \vdash c} \qquad (A)\dfrac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c}}{T_0 \vdash b}
$$

Example: a local proof of $T_0 \vdash s$

Example

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

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

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

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 P be 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

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 of necessity of $S(T \cup \{s\})$

Example

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

Example of necessity of $S(T \cup \{s\})$

Example

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

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

Example of necessity of $S(T \cup \{s\})$

Example

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

$$(P)\dfrac{(D)\dfrac{(A)\dfrac{\{b\}_c \in T_0}{T_0 \vdash \{b\}_c} \quad (D)\dfrac{(UR)\dfrac{(A)\dfrac{\langle a, \{c\}_k \rangle \in T_0}{T_0 \vdash \langle a, \{c\}_k \rangle}}{T_0 \vdash \{c\}_k} \quad (A)\dfrac{k \in T_0}{T_0 \vdash k}}{T_0 \vdash c}}{T_0 \vdash b} \quad (A)\dfrac{k \in T_0}{T_0 \vdash k}}{T_0 \vdash \langle b, k \rangle}$$

$S(T_0) = T_0 \cup \{a, b, c, k, \{b\}_k, \{c\}_k\}$ 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 non minimal proof is not S-local

GOAL: Find a good S.

Example

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

Example non minimal proof is not S-local

GOAL: Find a good S.

Example

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

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

Example non minimal proof is not S-local

GOAL: Find a good S.

Example

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

$$(D)\dfrac{(UL)\dfrac{(P)\dfrac{(A)\dfrac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k}\quad(A)\dfrac{\{c\}_k \in T_0}{T_0 \vdash \{c\}_k}}{T_0 \vdash \langle\{c\}_k, \{c\}_k\rangle}}{T_0 \vdash \{c\}_k}\qquad (A)\dfrac{k \in T_0}{T_0 \vdash k}}{c}$$

$S(T_0) = T_0 \cup \{c\}$ but $\langle\{c\}_k, \{c\}_k\rangle$
It is Not in $S(T_0 \cup \{s\})$

Example :

Shamir 3-Pass Protocol

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

Example :

Shamir 3-Pass Protocol

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

Example :

Shamir 3-Pass Protocol

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

Example :

Shamir 3-Pass Protocol

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

Example :

Shamir 3-Pass Protocol

$$
\begin{array}{llllll}
1 & A & \rightarrow & B & : & \{m\}_{K_A} \\
2 & B & \rightarrow & A & : & \{\{m\}_{K_A}\}_{K_B}
\end{array}
$$

Example :

Shamir 3-Pass Protocol

1	A	$\rightarrow$	B	:	$\{m\}_{K_A}$	Commutative
2	B	$\rightarrow$	A	:	$\{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A}$	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)

- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ — **A**ssociativity
- $x \oplus y = y \oplus x$ — **C**ommutativity
- $x \oplus 0 = x$ — **U**nity
- $x \oplus x = 0$ — **N**ilpotency

Logical Attack on Shamir 3-Pass Protocol (I)

Perfect encryption one-time pad (Vernam Encryption)

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

XOR Properties (ACUN)

- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ — **A**ssociativity
- $x \oplus y = y \oplus x$ — **C**ommutativity
- $x \oplus 0 = x$ — **U**nity
- $x \oplus x = 0$ — **N**ilpotency

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

$$\begin{array}{lllll} 1 & A & \rightarrow & B: & m \oplus K_A \\ 2 & B & \rightarrow & A: & (m \oplus K_A) \oplus K_B \\ 3 & A & \rightarrow & B: & m \oplus K_B \end{array}$$

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

$$\begin{array}{lllll} 1 & A & \rightarrow & B: & m \oplus K_A \\ 2 & B & \rightarrow & A: & (m \oplus K_A) \oplus K_B \\ 3 & A & \rightarrow & B: & m \oplus K_B \end{array}$$

Passive attacker :

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

Thank you for your attention.

Questions ?