

Lecture Note 08

Date: 12.10.2009

Contents

1	Active Intruder: Security Problem	2
1.1	Active Intruder Security Problem	2
1.2	Model: actions, roles and protocol	2
1.2.1	Action (Definition)	2
1.2.2	Role (Definition)	2
1.2.3	Protocol (Definition)	3
1.2.4	Semantic	5
1.2.5	Preservation of the secrecy	6
1.2.6	Interleaving	6
1.2.7	Secrecy over $<_E$	7
2	Bounded Number of Sessions	8
2.1	Constraints System	8
2.1.1	From Protocols to Constraints system	8
2.2	Resolution of Constraints systems	10
2.2.1	Properties of simplification rules	11
2.3	Decidability	11
3	Preservation of secrecy: a NP-hard problem	14

1 Active Intruder: Security Problem

1.1 Active Intruder Security Problem

An active intruder has the ability to

- Intercept messages adding them to his knowledge.
- Play messages from his knowledge
- Start new sessions

The Execution tree has:

- Infinite branching (size of messages is not bounded)
- Infinite depth (number of sessions is not bounded)

An active intruder with bounded number of sessions, usually attacks are done over a few sessions. This is theoretically decidable.

1.2 Model: actions, roles and protocol

1.2.1 Action (Definition)

An Action is a couple $(recv(u), send(v))$ such that $u \in T(F, X) \cup \{init\}$, $v \in T(F, X) \cup \{stop\}$. Denoted $(u \rightarrow v)$.

This means whenever (u) is received (v) is sent back. An action formally has an initialization and a stop.

1.2.2 Role (Definition)

A role is a finite sequence of actions:

$$(u_1 \rightarrow v_1), \dots, (u_n \rightarrow v_n) \\ \text{such that } vars(v_i) \subseteq \bigcup_{1 \leq j \leq i} vars(u_j).$$

It should be noted that any sent response during the role should contain only items that were previously received.

1.2.3 Protocol (Definition)

A Protocol P is a finite set of roles: $P = \{R_1, \dots, R_k\}$

In conclusion: P is a protocol containing a finite set of roles, each role containing a finite set of actions, each action is a couple of $(recv(u), send(v))$.

We look at the Needham-schroeder protocol as an example to illustrate this model:

1. $A \rightarrow B : N_a, A_{pk(B)}$
2. $B \rightarrow A : N_a, N_b, A_{pk(A)}$
3. $A \rightarrow B : N_b, A_{pk(B)}$

Write down each agent's role description, this A talks with anybody.

Action 1. The following describes the role of A who first initializes the protocol with the public key of B , sends his identity and a nonce to B .

$$R_a = (\langle init, X_b \rangle \rightarrow \{N_a, A\}_{pk(X_b)}),$$

Action 3. A receives the sent nonce and the new nonce back from B , uses his private key to get them and sends back the nonce generated by B .

$$(\{x_{N_a}, x_A\}_{pk(A)} \rightarrow \{x_{N_b}\}_{pk(x_b)}),$$

Action 2. The following describes the role of B who first receives A 's identity along with the nonce and uses his private key to get them. Then sends back the nonce sent by A and a new generated nonce encrypted by A 's public key.

$$R_b = (\{x_{N_a}, x_A\}_{pk(B)} \rightarrow \{x_{N_a}, N_b\}_{pk(x_A)}),$$

Action 4. Finally B receives the nonce he sent encrypted with his public key and the session is established and the protocol terminates.

$$(\{N_b\}_{pk(B)} \rightarrow stop),$$

It could be noticed that A has to know B before hand to be able to use X_b and use it along with $init$. It also should be noted that this protocol is vulnerable to a man-in-the-middle attack.

Scyther Notation

```
A: const Na: Nonce;
   var Nb: Nonce;
   send(A,B, {Na,A}pk(B));
   recv(B,A, {Na,Nb}pk(A));
   send(A,B, {Nb}pk(B));
B: const Nb: Nonce;
   var Na: Nonce;
   recv(A,B,{Na,A}pk(B));
   send(B,A,{Na,Nb}pk(A));
   recv(A,B,{Nb}pk(B));
```

Exercise: Denning-Sacco Protocol

1. $A \rightarrow S : \langle A, B \rangle$
2. $S \rightarrow A : \{\{\langle B, N_{AB} \rangle, \langle N_s, \{\langle N_{AB}, \langle A, N_s \rangle \} \}_{K_{BS}} \rangle\} \}_{K_{AS}}$
3. $A \rightarrow B : \langle N_{AB}, \langle A, N_s \rangle \rangle_{K_{BS}}$
4. $B \rightarrow A : \{S_{AB}\}_{N_{AB}}$

$P_{DS} = \{R_A, R_B, R_S\}$ models one session of A , B and S .

Action 1. This is sent to S requesting to start a session between A and B which are both known to S .

$$R_A = (\langle \text{init}, X_B \rangle \rightarrow \langle A, X_B \rangle),$$

Action 3. This is received back from S , so A decrypts the part encrypted with the shared key between A and S and sends the rest to B .

$$(\{\{\langle X_B, x_{N_{AB}} \rangle, \langle x_{N_s}, z_A \rangle\} \}_{K_{AS}} \rightarrow z_A),$$

Action 5. The response from B containing the identity of S , thus terminating the protocol.

$$(\{w_A\}_{x_{N_{AB}}} \rightarrow \text{stop})$$

Action 4. B receives from A the encrypted message and decrypts it and sending back the identity of S to A .

$$R_B = (\{\langle y_{N_{AB}}, \langle X_A, y_{N_s} \rangle \rangle_{K_{BS}} \rightarrow \{S_{AB}\}_{y_{N_{AB}}})$$

Action 2. S receives from A the request and generates nonces and encrypts parts of them with shared keys between B and S , and parts with shared keys between A and S and sends them back to A .

$$R_S = (\langle X_A, X_B \rangle \rightarrow \{\{\langle X_B, N_{AB}, \langle N_s, \{\langle N_{AB}, \langle X_A, N_s \rangle \} \}_{K_{BS}} \rangle\} \}_{K_{AS}}$$

Note: K_{AS} means that it is shared between A and S .

1.2.4 Semantic

States and Transitions (Definition) A state is a couple (T, P) where T is a set of ground terms (intruder knowledge) and P a protocol.

The transition relation between states (T, P) and (T', P') noted $(T, P) \rightarrow (T', P')$ means this:

- $R_i \in P, R_i = (u \rightarrow v)$

This means: let's take action $(u \rightarrow v)$ in any role R_i of protocol P

- $T \vdash u\sigma (dom(\sigma) = vars(u))$

Meaning: let's take $u\sigma$ such that I can deduce $u\sigma$ using the knowledge T

- $T' = T \cup \{v\sigma\}$

Meaning I can now compute a new $v\sigma$ instead of v and add it to the previous knowledge

- $R'_i \in P', R'_i = (P \setminus \{R_i\}) \cup R_i\sigma$

All items sent and deduced are added to the intruder's knowledge. Thus, adding new role to the protocol in the new state after the transition.

Example

Let $T = \{a, b, k_l\}$ **and** $p = \{R\}$ **where** $R = (\langle x, y \rangle \rightarrow \langle \{y\}_K, x \rangle), (z \rightarrow \langle x, \langle y, z \rangle \rangle)$.

Can we define a transition relation for this cases ?

- $(T, P) \rightarrow (T \cup \{\langle \{b\}_K, a \rangle\}, \{(z \rightarrow \langle a, \langle b, z \rangle \rangle)\})$

Yes because I already has a, b in T , then receiving z , I can compute the result

- $(T, P) \rightarrow (T \cup \{\langle \{a\}_{k_l}, a \rangle\}, \{(z \rightarrow \langle a, \langle \{a\}_{k_l}, z \rangle \rangle)\})$

Yes because of knowledge T , he has a, k_l , then receiving z , he can compute

- $(T, P) \rightarrow (T \cup \{\langle \{a\}_k, a \rangle\}, \{(z \rightarrow \langle a, \langle \{a\}_k, z \rangle \rangle)\})$

No because of he never has k on his knowledge.

Each branch has a *finite depth*, but *possible an infinite branching*.

1.2.5 Preservation of the secrecy

Let T_1 be a ground set of terms (Initial knowledge of the intruder).

A protocol P does not preserve the secrecy of a ground term s for T_1 if there exists a state (T', P') , such that (This is a correction to the notes)

- $T' \vdash s$
- $(T_1, P) \rightarrow^*(T', P')$

where \rightarrow^* is the reflexive and transitive closure of \rightarrow .

If there does not exist such a state (T', P') we say that P *preserves the secrecy* of s for the initial intruder knowledge T_1 .

A protocol simply preserves secrecy if, at the end, the intruder can't deduce S (the secret). We will discover now another approach of the preservation of secrecy

1.2.6 Interleaving

- Definition (Partial Order $<_P$)

A protocol P define a partial order $<_P$ on actions of P , such that

- $(u_i \rightarrow v_i) <_P (u_j \rightarrow v_j)$

– if $R \in P, R = (u_1 \rightarrow v_1) \dots (u_i \rightarrow v_i) \dots (u_j \rightarrow v_j) \dots (u_n \rightarrow v_n)$ ($1 \leq i \leq j \leq n$).

- Definition (Execution Order $<_E$)

An execution order $<_E$ of P is a total order on the subset A of actions of P , compatible with $<_P$ and stable by predecessor, i.e.

if $b \in A$ and $a <_P b$ then $a \in A$ and $a <_E b$

It corresponds to an interleaving of roles.

1.2.7 Secrecy over $<_E$

Let an execution order $<_E$ of P . We assume that

$$(u_1 \rightarrow v_1) <_E \cdots <_E (u_n \rightarrow v_n)$$

$<_E$ does not preserve the secrecy of s , given T_1 if there exists $\sigma_1, \dots, \sigma_n$ such that

$$(P_1, T_1) \rightarrow (P_1, T_1 \cup v_1\sigma_1) \rightarrow \cdots \rightarrow (P_n, T_1 \cup v_1\sigma_1, \dots, v_n\sigma_n)$$

and

$$T_1 \cup v_1\sigma_1, \dots, v_n\sigma_n \vdash s.$$

2 Bounded Number of Sessions

2.1 Constraints System

Definitions

A **constraint** is an expression $T \Vdash u$ where T is a set of terms and u a term.

The set of terms (T) represents the set of messages known to the attacker. And the expression $T \Vdash u$, means that the attacker must be able to synthesize u from the set of terms T .

A **constraints system** C is a finite set of constraints $\bigcup_{1 \leq i \leq n} T_i \Vdash u_i$ such that:

- $T_i \subseteq T_{i+1}$ ($1 \leq i \leq n$).
- if $T_i \Vdash u_i \in C$ and $x \in vars(T_i)$ then $T_j = \min \{T' \mid T' \Vdash v \in C, x \in vars(v)\}$ exists and $j < i$.

A substitution σ is a solution of C if $T\sigma \vdash u\sigma$ for all $T \Vdash u \in C$. We denote by \perp a constraints system unsatisfiable.

2.1.1 From Protocols to Constraints system

let P a protocol, $<_E$ an execution order of P and s a secret term.

$$(u_1 \rightarrow v_1) <_E (u_2 \rightarrow v_2) <_E \dots <_E (u_n \rightarrow v_n)$$

We associate C :

$$T_1 \Vdash u_1$$

$$T_2 = T_1 \cup \{v_1\} \Vdash u_2$$

.

.

.

$$T_n = T_{n-1} \cup \{V_{n-1}\} \Vdash u_n$$

$$T_{n+1} = T_n \cup \{v_n\} \Vdash s$$

We show that C has a solution iff $<_E$ does not preserve the secret of the term s .

Exercise 1

$$A \rightarrow B : \langle A, K \rangle$$

$$B \rightarrow A : \{\langle N_A, N_B \rangle\}_{K_{ab}}$$

$$A \rightarrow B : N_B$$

$$B \rightarrow A : \{\langle K, N_B \rangle\}_{K_{ab}}$$

$$A \rightarrow B : \{s\}_K$$

Intruder knows only identities of A and B .

- Give role specification of this protocol of an instance of execution between A and B .
- Give a constraints system associated to this protocol between A and B .

Solution:

$$T_1 = \{A, B, \langle A, N_A \rangle, \{\langle N_A, N_B \rangle\}_{K_{ab}}, N_B, \{\langle K, N_B \rangle\}_{K_{ab}}, \{s\}, \text{init}, \text{stop}\}$$

Roles:

$$R_A = (\text{init} \rightarrow \langle A, N_A \rangle), \left(\{\langle N_A, X_{N_B} \rangle\}_{K_{(A, X_B)}} \rightarrow X_{N_B} \right), \left(\{\langle X_K, X_{N_B} \rangle\}_{K_{(A, X_B)}} \rightarrow \{s\}_{X_K} \right)$$

As we know a **Role** is a finite sequence of actions, and an **Action** is a couple ($\text{recv}(u), \text{send}(v)$) denoted as $(u \rightarrow v)$.

$$(\text{init} \rightarrow \langle A, N_A \rangle)$$

So first A receives the initiation of the **Protocol** and sends both his identity and nonce to B .

$$\left(\{\langle N_A, X_{N_B} \rangle\}_{K_{(A, X_B)}} \rightarrow X_{N_B} \right)$$

Then A receives his nonce and B 's nonce encrypted with the symmetric key between A and B and sends back B 's nonce.

$$\left(\{ \langle X_K, X_{N_B} \rangle \}_{K_{(A,B)}} \rightarrow \{s\}_{X_K} \right)$$

After that A receives the B 's nonce again with a key K , both encrypted with the symmetric key between them. Finally A sends S encrypted with the key K to B , $\{s\}_{X_K}$.

As you can see all the messages that are not know to A are denoted by X , for example N_B is denoted as X_{N_B} . The same goes for the role of B .

$$R_B = \left(\langle X_A, X_{N_A} \rangle \rightarrow \{ \langle X_{N_A}, N_B \rangle \}_{K_{(A,B)}} \right), \left(N_B \rightarrow \{ \langle K, N_B \rangle \}_{K_{(A,B)}} \right), (\{X_s\}_K \rightarrow stop)$$

Constraint System:

$$\begin{array}{ll} T_1 & \Vdash init \\ T_2 = T_1 \cup \{ \langle A, N_A \rangle \} & \Vdash \langle X_A, X_{N_A} \rangle \\ T_3 = T_2 \cup \left\{ \{ \langle X_{N_A}, N_B \rangle \}_{K_{(A,B)}} \right\} & \Vdash \{ \langle N_A, X_{N_B} \rangle \}_{K_{(A,B)}} \\ T_4 = T_3 \cup \{ X_{N_B} \} & \Vdash N_B \\ T_5 = T_4 \cup \left\{ \{ \langle K, N_B \rangle \}_{K_{(A,B)}} \right\} & \Vdash \{ \langle X_K, X_{N_B} \rangle \}_{K_{(A,B)}} \\ T_6 = T_5 \cup \{ \{s\}_{X_K} \} & \Vdash \{X_s\}_K \\ T_7 = T_6 \cup \{ stop \} & \Vdash s \end{array}$$

2.2 Resolution of Constraints systems

Rules of simplification: $C \rightsquigarrow_\sigma C'$

$$\begin{array}{ll} R_1 & C \cup \{ T \Vdash u \} \rightsquigarrow C \quad \text{if } T \cup \{ x \mid T' \Vdash x \in C, T' \subset T \} \vdash u \\ R_2 & C \cup \{ T \Vdash u \} \rightsquigarrow_\sigma C_\sigma \cup \{ T\sigma \Vdash u\sigma \} \quad \sigma = mgu(t, u), t \in st(T), t, u \text{ no variables} \\ R_3 & C \cup \{ T \Vdash u \} \rightsquigarrow_\sigma C_\sigma \cup \{ T\sigma \Vdash u\sigma \} \quad \sigma = mgu(t_1, t_2), t_1, t_2 \in st(T), t_1, t_2 \text{ no variables} \\ R_4 & C \cup \{ T \Vdash \{u\}_v \} \rightsquigarrow C \cup \{ T \Vdash u, T \Vdash v \} \\ R_5 & C \cup \{ T \Vdash \} \rightsquigarrow C \cup \{ \langle u, v \rangle T \Vdash u, T \Vdash v \} \\ R_6 & C \cup \{ T \Vdash u \} \rightsquigarrow \perp \quad \text{if } T = \emptyset \text{ or } var(T) = var(u) = \emptyset \text{ and } T \forall u \end{array}$$

2.2.1 Properties of simplification rules

- **Preservation**
Simplification rules transform a constraints system into a constraints system.
- **Correctness**
if $C \rightsquigarrow_{\sigma} C'$ then if θ is a solution of C' , $\sigma\theta$ is also a solution of C .
- **Termination**
Simplification rules always terminate: There does not exist infinite chain $C \rightsquigarrow_{\sigma_1} C_1 \rightsquigarrow_{\sigma_2} C_2 \rightsquigarrow_{\sigma_3} \dots$
- **Solved Form**
A constraints system C is in solved form if $C = \perp$ or if each constraint is of the following form $T \Vdash x$ where x is a variable $T \neq \emptyset$.

All constraints system is solved form different of \perp has at least one solution.

- **Completeness**
if C is a constraints system not in solved form and if σ is a solution of C then there exists θ, τ such that $C \rightsquigarrow_{\theta} C', \sigma = \theta\tau$ and τ is a solution of C' .

2.3 Decidability

Theorem: Preservation of the secrecy for protocol with bounded number of sessions is decidable.

- Guess an interleaving and build constraints system associated to the protocol.
- Using previous lemma C has a solution iff there exists C' in solved form such that $C' = \perp$ (there is not attack) and $C' \rightsquigarrow \tau C'$.
- Using termination lemma to conclude.

We also can show that the protocol is co-NP.

Here are two exercises to illustrate.

Exercise 1:

$$A \rightarrow B : \{\langle A, K \rangle\}_{K_{ab}}$$

$$B \rightarrow A : \{\langle N_A, N_B \rangle\}_{K_{ab}}$$

$$A \rightarrow B : N_B$$

$$B \rightarrow A : \{\langle K, N_B \rangle\}_{K_{ab}}$$

$$A \rightarrow B : \{s\}_K$$

Intruder knows only identities of A and B.

- Use simplification rules to transform the system in solved form.
- There exists an easy attack, can you find it ?

Solution:

From the initial intruder knowledge (T_1).

$$T_1 = \{A, B, \langle A, B \rangle, \{\langle N_A, N_B \rangle\}_{K_{ab}}, N_B, \{\langle K, N_B \rangle\}_{K_{ab}}, \{s\}_K, \text{init}, \text{stop}\}$$

Then, lets illustrate the constraint system that associated with the protocol between A and B .

$$\begin{array}{lll}
C_1 & T_1 & \Vdash \text{init} \\
C_2 & T_2 = T_1 \cup \{\langle A, N_A \rangle\} & \Vdash \langle X_A, X_{N_A} \rangle \\
C_3 & T_3 = T_2 \cup \{\{\langle X_{N_A}, N_B \rangle\}_{K_{(A,B)}}\} & \Vdash \{\langle N_A, X_{N_B} \rangle\}_{K_{(A,X_B)}} \\
C_4 & T_4 = T_3 \cup \{X_{N_B}\} & \Vdash N_B \\
C_5 & T_5 = T_4 \cup \{\{\langle K, N_B \rangle\}_{K_{(A,B)}}\} & \Vdash \{\langle X_k, X_{N_B} \rangle\}_{K_{(A,X_B)}} \\
C_6 & T_6 = T_5 \cup \{\{s\}_{X_K}\} & \Vdash \{X_s\}_K \\
C_7 & T_7 = T_6 \cup \{\text{stop}\} & s
\end{array}$$

As we know there are such a rules of simplification that is applied to transform the system in solved form. So starting on C_1 that does not require any rule since it is already in resolved form. Next, Lets look at rule simplification R_2 :

$$R_2 \ C \cup \{T \Vdash u\} \rightsquigarrow C_\sigma \cup \{T_\sigma \Vdash u\sigma\} \ \sigma = \text{mgu}(t, u), \ t \in \text{st}(T), \ t, \ u \text{ no variables}$$

Next, by applying R_2 on C_2 which gives $\sigma_1 = \{X_{N_A} \leftarrow N_A, X_A \leftarrow A\}$ and R_1 , leads the system like:

$$\begin{array}{lll}
C_3\sigma_1 & T_3 = T_2 \cup \{\{\langle X_{N_A}, N_B \rangle\}_{K_{(A,B)}}\} & \Vdash \{\langle N_A, X_{N_B} \rangle\}_{K_{(A,X_B)}} \\
C_4\sigma_1 & T_4 = T_3 \cup \{X_{N_B}\} & \Vdash N_B \\
C_5\sigma_1 & T_5 = T_4 \cup \{\{\langle K, N_B \rangle\}_{K_{(A,B)}}\} & \Vdash \{\langle X_k, X_{N_B} \rangle\}_{K_{(A,X_B)}} \\
C_6\sigma_1 & T_6 = T_5 \cup \{\{s\}_{X_K}\} & \Vdash \{X_s\}_K \\
C_7\sigma_1 & T_7 = T_6 \cup \{\text{stop}\} & s
\end{array}$$

Next, by applying R_2 on C_3 which gives $\sigma_2 = \{X_{N_B} \leftarrow N_B, X_B \leftarrow B\}$ (or N_A) and R_1 , leads the system like:

$$\begin{array}{lll}
C_5\sigma_1\sigma_2 & T_5 = T_4 \cup \{\{\langle K, N_B \rangle\}_{K_{(A,B)}}\} & \Vdash \{\langle X_k, X_B \rangle\}_{K_{(A,B)}} \\
C_6\sigma_1\sigma_2 & T_6 = T_5 \cup \{\{s\}_{X_K}\} & \Vdash \{X_s\}_K \\
C_7\sigma_1\sigma_2 & T_7 = T_6 \cup \{\text{stop}\} & s
\end{array}$$

Last two steps are:

- Apply R_2 on $C_5\sigma_1\sigma_2$ give $\sigma_3 = \{X_K \leftarrow N_A\}$
- Apply R_2 on $C_6\sigma_1\sigma_2\sigma_3$ give $\sigma_4 = \{X_S \leftarrow s\}$

Finally, the resolution of constraint system gives the following attack: Send second message $\{\langle N_A, N_B \rangle\}_{K_{ab}}$ instead of the fourth message $\{\langle K, N_B \rangle\}_{K_{ab}}$. Hence A will reply $\{s\}_{N_A}$ because intruder knows N_A .

Exercise 2:

$$A \rightarrow B : \{\langle A, K \rangle\}_{K_{ab}}$$

$$B \rightarrow A : \{s\}_{K_{ab}}$$

Intruder knows only identities of A and B. Show that the secret data s is preserved by one single session between A and B.

Solution:

From the initial intruder knowledge (T_1).

$$T_1 = \{A, B, \{\langle A, K \rangle\}, \{s\}_{K_{ab}}\}$$

Then, let's illustrate the constraint system that associated with the protocol between A and B.

$$\begin{array}{lll} C_1 & T_1 & \Vdash \{\langle A, X_K \rangle\}_{K_{ab}} \\ C_2 & T_2 = T_1 \cup \{\langle A, X_K \rangle\}_{K_{ab}} & \Vdash \{s\}_{X_{K_{ab}}} \\ C_3 & T_3 = T_2 \cup \{s\}_{X_{K_{ab}}} & \Vdash s \end{array}$$

There are such a rules of simplification that is applied to transform the system in solved form. In this exercise, we can simply apply nothing or R_4 or R_5 and R_2 on C_1 that give $\sigma_0 = \{X_K \leftarrow K, X_{K_{ab}} \leftarrow K_{ab}\}$. Next (and last) step is to apply R_5 or nothing and R_2 on $C_2\sigma_0$ that give $\sigma_1 = \{X_{N_B} \leftarrow N_B\}$ (or N_A).

Each time you meet a solved form of the form \perp with R_6 .

3 Preservation of secrecy: a NP-hard problem

- theorem

Decide if a protocol P does not preserve the secrecy term s from an initial knowledge T_1 is NP-difficult.

We will prove it by reduction, according to the article

The proof consists on finding a solution to this problem is equivalent of finding a solution for 3-SAT

- 3-SAT Recall problem

Input: set of propositional variables $\{x_1, \dots, x_n\}$ and a conjunction of clauses with 3 literals.

$$f(\vec{x}) = \bigwedge_{1 \leq i \leq I} (x_{i,1}^{\epsilon_{i,1}} \vee x_{i,2}^{\epsilon_{i,2}} \vee x_{i,3}^{\epsilon_{i,3}})$$

where $\epsilon_{i,j} \in \{+, -\}$ and $x^+ = x$, $x^- = \neg x$

Question : Does exist a valuation V of x_1, \dots, x_n , such that $V(f(\vec{x})) = \top$

We know that this problem is NP-Complete

- The reduction proof

With the same notations than previously, let us define g

$g(x_{i,j}^{\epsilon_{i,j}}) = x_{i,j}$ if $\epsilon_{i,j} = +$, $g(x_{i,j}^{\epsilon_{i,j}}) = \{x_{i,j}\}_K$ else

So $\forall 1 < i < I : f_i(\vec{x}) = \langle g(x_{i,1}^{\epsilon_{i,1}}), \langle g(x_{i,2}^{\epsilon_{i,2}}), g(x_{i,3}^{\epsilon_{i,3}}) \rangle \rangle$

Let us introduce now the following protocol designed such that an intruder can deduce s iff $f(\vec{x})$ is satisfaisable.

$$A : \langle x_1 \langle \dots, x_n \rangle \rangle \rightarrow \{ \langle f_1(\vec{x}), \langle f_2(\vec{x}), \langle \dots, \langle f_n(\vec{x}), end \rangle \dots \rangle \rangle \}_p$$

$$\forall 1 < i < I :$$

$$B_i : \{ \langle \langle \top, \langle x, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p$$

$$\bar{B}_i : \{ \langle \langle \{ \perp \}_K, \langle x, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p$$

$$C_i : \{ \langle \langle x, \langle \top, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p$$

$$\bar{C}_i : \{ \langle \langle x, \langle \{ \perp \}_K, y \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p$$

$$D_i : \{ \langle \langle x, \langle y, \top \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p$$

$$\bar{D}_i : \{ \langle \langle x, \langle y, \{ \perp \}_K \rangle \rangle, z \rangle \}_p \rightarrow \{z\}_p$$

$$E : \{ end \}_p \rightarrow s$$

We take $\{\top, \perp\}$ for the initial intruder knowledge.

Hence there is an attack on this protocol iff the message sent by principal A can be reduced to $\{end\}_P$

i.e. for all i , there exists j such that $g(x_{i,j}^{\varepsilon_{i,j}}) = x_{i,j} \in \{\top, \{\neg\top\}_P\}$

i.e. the intruder has given to A a term representing a solution of 3-SAT ($\{\neg\top\}_P$ represents \top).

Hence the protocol admits an attack iff the corresponding 3-SAT problem has a solution.