Analyzing Security Protocols with Rewriting Techniques

Michael Rusinowitch

LORIA

July 2006
Security Problems

Communication in Open Networks

Alice and Bob exchange Messages
An Intruder controls the communication channel

Security Objectives for: A sends message M to B

Confidentiality (only A and B know M)
Integrity of datas (M is not altered)
Authenticity (B knows that A has sent M)

Solution: Security Protocols

SSL: browsers
PGP: mail
SET: E-commerce
Kerberos: remote login ....
Building Blocks for Security Protocols

Cryptographic Procedures: Encryption or signature of messages

- public keys \( \{M\}_{KB}, K_B^{-1} \Rightarrow M \)
- secret keys \( \{M\}_K, K \Rightarrow M \)

(Pseudo-)Random Number Generators: to generate “Nonces”, e.g. for “Challenge-Response”

Protocols: recipe for exchanging messages

Steps like: A sends B his name together with the message M. The pair \( \{A, M\} \) is encrypted with B’s key.

\[ A \rightarrow B : \{A, M\}_{KB} \]
Alice \rightarrow \text{Banque} : \{ \text{transfer 1000 EUROS to Charlie} \}_{K^{-1}_{Alice}}
Alice $\rightarrow$ Banque : $\{ \text{transfer 1000 } \text{EUROS} \text{ to Charlie} \}^{K^{-1}_{Alice}}$

Charlie $\rightarrow$ Banque : $\{ \text{transfer 1000 } \text{EUROS} \text{ to Charlie} \}^{K^{-1}_{Alice}}$

... 

Replay attacks
An Authentication Protocol: Needham-Schroeder

1. $A \rightarrow B : \quad \{A, N_A\}_{K_B}$
2. $B \rightarrow A : \quad \{N_A, N_B\}_{K_A}$
3. $A \rightarrow B : \quad \{N_B\}_{K_B}$

Translation:

- $\{A, N_A\}_{K_B}$
  - “I am Alice and here is a Nonce $N_A$.”

- $\{N_A, N_B\}_{K_A}$
  - “Here is your Nonce $N_A$ and I also have one for you.”

- $\{N_B\}_{K_B}$
  - “I got it! It is $N_B$.”

Protocols are typically small and convincing …
... but also often wrong!
Attack on Needham-Schroeder

NSPK #1

NSPK #2
Attack on Needham-Schroeder

\[ \{A, N_A\}_{K_C} \]
Attack on Needham-Schroeder

\{A, N_A\}_{K_C} \quad \text{NSPK #1} \quad \{A, N_A\}_{K_B} \quad \text{NSPK #2}
Attack on Needham-Schroeder

\[ \{A, N_A\}_{K_C} \]

\[ \{N_A, N_B\}_{K_A} \]

\[ \{A, N_A\}_{K_B} \]

\[ \{N_A, N_B\}_{K_A} \]
Attack on Needham-Schroeder

\[\{A, N_A\}_{K_C}\]
\[\{N_A, N_B\}_{K_A}\]
\[\{N_B\}_{K_C}\]
\[\{A, N_A\}_{K_B}\]
\[\{N_A, N_B\}_{K_A}\]
Attack on Needham-Schroeder

\[ \{A, N_A\}_{K_C} \]

\[ \{N_A, N_B\}_{K_A} \]

\[ \{N_B\}_{K_C} \]

\[ \{A, N_A\}_{K_B} \]

\[ \{N_A, N_B\}_{K_A} \]

\[ \{N_B\}_{K_B} \]

\(B\) believes he is speaking with \(A\)!
What went wrong?

• Problem in step 2  \( B \rightarrow A : \{N_A, N_B\}_{K_A} \)
  Agent \( B \) should also give his name:  \( \{N_A, N_B, B\}_{K_A} \)

• Is the improved version now correct?
Type Confusion (Otway-Rees)

Alice → Bob : $\{ NA, Alice, Bob \}_{KAS}$

Bob → Server : $\{ NA, Alice, Bob \}_{KAS}, \{ NB, Alice, Bob \}_{KBS}$

Server → Bob : $\{ NA, key \}_{KAS}, \{ NB, key \}_{KBS}$

Bob → Alice : $\{ NA, key \}_{KAS}$
Intruder Model (Dolev Yao 1982)

He can
spy, record, modify, reply
masquerade by changing source address
initiate parallel sessions
(create type confusions)
Intruder Model (Dolev Yao 1982)

He can
spy, record, modify, reply
masquerade by changing source address
initiate parallel sessions
(create type confusions)

He cannot
decrypt without the key
create cipher without both plaintext and encryption key

\[ \{ M \}_{K}, \{ M' \}_{K} \not\Rightarrow \{ M \cdot M' \}_{K} \]

black-box cryptography or perfect encryption hypothesis
... messages are first-order terms
Formal Analysis of Security Protocols

• Challenging as general problem is **undecidable**.

• Several **sources of infinity** in protocol analysis:
  – Unbounded **number of possible intruder messages**.
  – Unbounded **message depth**.
  – Unbounded **number of agents**.
  – Unbounded **number of sessions** or protocol steps.
  – Unbounded **number of possible values** for nonces.

• Possible approaches:
  – **Falsification** identifies attack traces but does not guarantee correctness.
    ⇒ **constraint solving**
  – **Verification** proves correctness but is difficult to automate (requires induction and often restrictions).
    ⇒ **tree automata approximation** or **resolution theorem proving**

July 2006
Formal Methods

- **Modal Logic**
  - Burrows Abadi, Needham (BAN)
    “if you believe that only you and Bob know $K$ then you ought to believe that anything you receive encrypted by $K$ comes from Bob”

- **Model-Checking**
  - Lowe (CSP/FDR), ...
  efficient, approximations for finiteness (random numbers = constants) $\rightarrow$ incompleteness

- **Infinite Traces Analysis**
  - Meadows (prolog and rewriting, narrowing), Paulson (Isabelle), ...
  invariant proofs by induction $\rightarrow$ correction
The AVISPA Tool

- verify that a protocol can be implemented
- translate a protocol to a set of transition rules
- to detect attacks, or validate the protocol

example of a non implementable protocol:

1. $A \rightarrow B : \{X\}Ka$
2. $B \rightarrow A : X$

IST Project AVISPA:
ETH Zurich (Basin), INRIA, U. Genova (Armando), Siemens (Cuellar)
www.avispa-project.org

related project: RNTL PROUVE (Treinen), . . .
High-Level Protocol Specification Language (HLPSL)

Translator
HLPSL2IF

Intermediate Format (IF)

On-the-fly Model-Checker
OFMC

CL-based Attack Searcher
AtSe

SAT-based Model-Checker
SATMC

Tree Automata-based Protocol Analyser
TA4SP

avispa script file

Output Format (OF)

July 2006
The AVISPA Tool
Roles in HLPSL

role bob(A,B: agent, Kb,Ks: public_key, 
    KeyRing: (agent.public_key) set, 
    SND,RCV: channel(dy)) played_by B def= 
local State: nat, Na,Nb: text, Ka: public_key 
init State:=0 
transition 
    1a. State=0 \ RCV({Na’.A}_{Kb}) \ in(A.Ka’,KeyRing) 
        =|> State’:=2 \ Nb’:=new() \ SND({Na’.Nb’}_{Ka’}) 
...
end role

role session(A,B: agent, Ka,Kb,Ks: public_key, 
    KeySet: agent -> (agent.public_key) set) def= 
local SND,RCV: channel(dy) 
composition 
    alice(A,B,Ka,Ks,KeySet(A),SND,RCV) 
    /\ bob(A,B,Kb,Ks,KeySet(B),SND,RCV) 
end role
role environment() def=
    local KeySet: agent -> (agent.public_key) set,
        KeyRing: (agent.public_key) set,
        Snd,Rcv: channel(dy)
    const a,b,s,i: agent, ka,kb,ks,ki: public_key
    init KeySet:= { a.{}, b.{a.ka}, i.{a.ka,b.kb} }
        KeyRing:= {a.ka,b.kb,s.ks,i.ki}
    intruder_knowledge={i,a,b,ks,ki,inv(ki)}
    composition
        server(s,ks,KeyRing,Snd,Rcv)
        session(a,b,ka,kb,ks,KeySet)
        session(a,i,ka,ki,ks,KeySet)

end role

goal
    authentication_on nb
    weak_authentication_on na
    secrecy_of na, nb
    [](<-> has_seen(A,B,M) => ((has_seen(B,A,M) and iknows(M)) or B=i))

end goal
Intermediate Format: Transition Rules

State = multiset of terms (built with “.” )

\[\text{state\_Server}(S, \text{Ks}, \text{KeyMap}, \_, \_, \_, \_, \_, \_, \_, \text{SID}).\]
\[\text{iknows}(\text{pair}(A, B)).\]
\[\text{contains}(\text{pair}(B, \text{Kb}), \text{KeyMap})\]

⇒

\[\text{state\_Server}(S, \text{Ks}, \text{KeyMap}, A, B, \text{Kb}, \_, \_, \_, \_, \_, \_, \_, \text{SID}).\]
\[\text{iknows}(\text{crypt}(\text{inv}(\text{Ks}), \text{pair}(B, \text{Kb}))).\]
\[\text{contains}(\text{pair}(B, \text{Kb}), \text{KeyMap})\]
Pattern-matching model

1. $A \rightarrow B : \{N_A, A\}_{KB}$
2. $B \rightarrow A : \{N_A, N_B\}_{KA}$
3. $A \rightarrow B : \{N_B\}_{KB}$

is translated to:

\[
\begin{align*}
\text{step } i, \quad \text{expected Answer}_i & \quad \Rightarrow \quad \text{composed Message}_i \\
(A,1), \quad \text{Init} & \quad \Rightarrow \quad \{N_A, A\}_{KB} \\
(B,1), \quad \{x, y\}_{KB} & \quad \Rightarrow \quad \{x, N_B\}_{Ky} \\
(A,2), \quad \{N_A, z\}_{KA} & \quad \Rightarrow \quad \{z\}_{KB} \\
(B,2), \quad \{N_B\}_{KB} & \quad \Rightarrow \quad \text{End}
\end{align*}
\]
**Intruder Rules**

<table>
<thead>
<tr>
<th>Decomposition:</th>
<th>Composition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>{a, b} \rightarrow {a, b}, a, b</td>
<td>a, b \rightarrow a, b, {a, b}</td>
</tr>
<tr>
<td>(K^{-1}, {a}_K) \rightarrow (K^{-1}, {a}_K, a)</td>
<td>a, K \rightarrow a, K, {a}_K</td>
</tr>
<tr>
<td>S, {a}_S \rightarrow S, {a}_S, a</td>
<td>a, S \rightarrow a, S, {a}_S</td>
</tr>
</tbody>
</table>

\(t \in forge(E) \text{ iff } E \rightarrow^* t, \ldots\)
Passive Intruder

\[\{\text{secret}\}_{k_1}^{k_2}, \{k_1, k_2\}_{k_3}, k_3\]
Passive Intruder

\[
\{secret\}_{k_2} \{k_1\}_{k_2}, \ {k_1, k_2}_{k_3}, \ k_3
\]

→

\[
\{secret\}_{k_2} \{k_1\}_{k_2}, \ k_1, \ k_2, \ k_3 \ldots
\]
Passive Intruder

\[
\{\text{secret}\}\{k_1\}_{k_2}, \ \{k_1, k_2\}_{k_3}, \ k_3
\]

\[\rightarrow\]

\[
\{\text{secret}\}\{k_1\}_{k_2}, \ k_1, k_2, k_3 \ldots
\]

\[\rightarrow\]

\[
\{\text{secret}\}\{k_1\}_{k_2}, \ \{k_1\}_{k_2}, \ k_3 \ldots
\]
Passive Intruder

\[
\{\text{secret}\}\{k_1\}_{k_2}, \ \{k_1, k_2\}_{k_3}, \ k_3
\]
\[
\rightarrow
\]

\[
\{\text{secret}\}\{k_1\}_{k_2}, k_1, k_2, k_3 \ldots
\]
\[
\rightarrow
\]

\[
\{\text{secret}\}\{k_1\}_{k_2}, \ \{k_1\}_{k_2}, k_3 \ldots
\]
\[
\rightarrow
\]

\text{secret \ldots}

Active Intruder, Protocol Insecurity Problem

the protocol is insecure (i.e. there exists an attack) iff there exists a solution to:

\[
\text{Answer}_i \in forge(\text{Message}_0, \text{Message}_1, \ldots, \text{Message}_{i-1}) \quad \text{for } i = 1, \ldots, k
\]

and \( \text{Secret} \in forge(\text{Message}_0, \text{Message}_1, \ldots, \text{Message}_k) \)

(Message\(_0\) contains the initial knowledge of intruder)
Example

\[ \{x, y\}_{KB} \in forge(Message_0, \{NA, A\}_{KB}) \]

\[ \land \{NA, z\}_{KA} \in forge(Message_0, \{NA, A\}_{KB}, \{x, NB\}_{Ky}) \]

\[ \land NB \in forge(Message_0, \{NA, A\}_{KB}, \{x, NB\}_{Ky}, \{z\}_{KB}) \]

with initial intruder knowledge = Message_0 = \{KB, KA, A, B, I, Init\}

and protocol:

1. \( A \rightarrow B : \{NA, A\}_{KB} \)
2. \( B \rightarrow A : \{NA, NB\}_{KA} \)
3. \( A \rightarrow B : \{NB\}_{KB} \)
Upper bound

1. $t \in \text{forge}(E_0)$ can be checked in polynomial time: if the intruder can forge message $t$ from initial knowledge $E_0$, then he can forge $t$ with a number of elementary operations $\leq |t| + |E_0|$

2. if protocol $P$ is insecure then there exists an attack where every message sent has size $\leq |P|$
Upper bound

1. $t \in \text{forge}(E_0)$ can be checked in polynomial time: if the intruder can forge message $t$ from initial knowledge $E_0$, then he can forge $t$ with a number of elementary operations $\leq |t| + |E_0|$

2. if protocol $P$ is insecure then there exists an attack where every message sent has size $\leq |P|$

($k.|P|$ for $k$ sessions )
Upper bound

1. \( t \in forge(E_0) \) can be checked in polynomial time: if the intruder can forge message \( t \) from initial knowledge \( E_0 \), then he can forge \( t \) with a number of elementary operations \( \leq |t| + |E_0| \)

2. if protocol \( P \) is insecure then there exists an attack where every message sent has size \( \leq |P| \)

\((k.|P| \text{ for } k \text{ sessions })\)

\( \Rightarrow \text{protocol insecurity for bounded number of sessions is of complexity } \leq \text{NP.} \)
1. Forgeability in polynomial time

Assume $E_0 \rightarrow E_1 \rightarrow \ldots \rightarrow E_m$ is a minimal length derivation to forge $t$ from $E_0$ then all terms in all $E_i$ are subterms of $t$ or $E_0$. 
1. Forgeability in polynomial time

Assume $E_0 \rightarrow E_1 \rightarrow \ldots \rightarrow E_m$ is a minimal length derivation to forge $t$ from $E_0$ then all terms in all $E_i$ are subterms of $t$ or $E_0$.

**Proof**: in a minimal derivation

- same term is not composed and decomposed
- only (sub)terms from $E_0$ are decomposed
- only (sub)terms from $E_0$ or $t$ are composed
2. Messages sizes

in a minimal attack $\sigma$, for all variables $x$ there exists a non variable subterm $t$ of the protocol such that $\sigma(x) = t\sigma$
NP Algorithm

Given a protocol $P$ with $n$ steps

1. guess a substitution (polynomial size in $|P|$)
   \[ \text{Var}(P) \rightarrow \{ s \mid s \text{ subterm of } P \} \]

2. check for $i = 1..n$ if $\text{Answer}_i \in \text{forge}(\text{Message}_0, \text{Message}_1, ..., \text{Message}_{i-1})$

3. check if $\text{Secret} \in \text{forge}(\text{Message}_0, \text{Message}_1, ..., \text{Message}_n)$

4. if all tests positive then YES

For $k$ sessions we first guess an interleaving of the sessions.
NP-hardness of Insecurity for Finite Sessions

3-SAT formula: \( \phi = (x_1 \lor x_2 \lor \neg x_3) \land (\ldots) \land \ldots \)

Protocol: A builds an instance of \( \phi \):
\[ A : \{x_1, x_2, \ldots, x_n\} \Rightarrow \{\{x_1, x_2, \{x_3\}_K\}, \{\ldots\}, \ldots, end\}\}_P \]

Some principals check the truth value of literals, for instance
\[ P : \{\{True, y, z,\}, w\}_P \Rightarrow \{w\}_P \]
\[ N : \{\{False\}_K, y, z\}, w\}_P \Rightarrow \{w\}_P \]
\[ Z : \{end\}_P \Rightarrow \text{Secret} \]

The protocol is subject to an attack iff \( \phi \) is satisfied by some assignment (Intruder initially knows only True and False)
Extending Dolev-Yao Intruder by New Rules

\[ \{a, b\}_K \rightarrow \{a\}_K \]

for cipher block chaining attacks (CBC)
Extending Dolev-Yao Intruder by New Rules

\[ \{a, b\}_K \rightarrow \{a\}_K \]

for cipher block chaining attacks (CBC)

\[ a \rightarrow \{\{a\}_K\}_K \]

if plain and ciphered texts cannot be distinguished
Extending Dolev-Yao Intruder by New Rules

\[
\{a, b\}_K \rightarrow \{a\}_K
\]

for cipher block chaining attacks (CBC)

\[
a \rightarrow \{\{a\}_K\}_K
\]

if plain and ciphered texts cannot be distinguished

\[
a, b \rightarrow a \oplus b
\]

where \(\oplus\) is ACUN and represents the bitwise XOR
CBC attack on Needham Schroeder with Symmetric Key

1. \( A \rightarrow S : \ A, B, N_A \)
2. \( S \rightarrow A : \ \{N_A, B, K_{AB}, \{K_{AB}, A\}K_{BS}\}K_{AS} \)
3. \( A \rightarrow B : \ \{K_{AB}, A\}K_{BS} \)
4. \( B \rightarrow A : \ \{N_B\}K_{AB} \)
5. \( A \rightarrow B : \ \{N_B - 1\}K_{AB} \)
6. \( B \rightarrow A : \ \{\text{Secret}\}K_{AB} \)

The intruder forges \( \{N_A, B\}K_{AS} \) from 2. and sends it to \( A \) in another session where \( B \) is the initiator. The key \( N_A \) gets accepted by \( A \) and the intruder can derive the Secret.

1'. \( B \rightarrow S : .. \)
2'. \( S \rightarrow B : .. \)
3'. \( I(B) \rightarrow A : \ \{N_A, B\}K_{AS} \)
4'. \( ... : .. \)
5'. \( ... : .. \)
6'. \( A \rightarrow I(B) : \ \{\text{Secret}\}N_A \)
XOR attack

Protocol:

1. \( A \rightarrow B : \{Na, A\}_{Kb} \)
2. \( B \rightarrow A : \{Nb, Na \oplus B\}_{Ka} \)
3. \( A \rightarrow B : \{Nb\}_{Kb} \)

Attack:

1. \( A \rightarrow I : \{Na, A\}_{Ki} \)
1'. \( I(A) \rightarrow B : \{Na \oplus B \oplus I, A\}_{Kb} \)
2'. \( B \rightarrow I(A) : \{Nb, Na \oplus B \oplus I \oplus B\}_{Ka} \)
2. \( I \rightarrow A : \{Nb, Na \oplus I\}_{Ka} \)
3. \( A \rightarrow I : \{Nb\}_{Ki} \)
3'. \( I(A) \rightarrow B : \{Nb\}_{Kb} \)
Diffie Hellman attack

1. $A \rightarrow B : g^{Na}$
2. $B \rightarrow A : g^{Nb}$
3. $A \rightarrow B : \{Secret\}_{g^{Na \times Nb}}$

**shared key:** $g^{Na \times Nb}$

**an attack:** the intruder sends $g$ in step 2, receives back $\{Secret\}_{g^{Na}}$ and deduces $Secret$

**exp theory:** $\times$ abelian group operator and $(g^a)^b = g^{a \times b}$, $g^1 = g$.

$$a, b, c \ldots \rightarrow_{exp} a^{bnb} \times c^{nc} \times \ldots$$
Exp intruder

• adapted notion of subterms (\( a \times b \) not subterm of \( a \times b \times c \))

• if \( t \in Forge(E) \) there is a derivation s.t. all intermediate terms are in \( Subterms(E,t) \).

• \( U \rightarrow_{exp} v \) can be checked in polynomial time in \( U,v \).

• By closure, we can compute \( Forge(E) \cap Subterms(E,t) \) in polynomial time.

\[ \Rightarrow t \in Forge(E) \text{ can be checked in polynomial time w.r.t. } E,t \]

**Theorem 1.** \( t \in Forge(E) \) can be checked in polynomial time.

**Theorem 2.** protocol insecurity for Exp theory \( \in NP \).
E-Unification can be reduced to Insecurity

\[ S \rightarrow A : \bar{x} \]

\[ A \rightarrow B : \{ s(\bar{x}), t(\bar{x}) \}_{KAB} \]

\[ B \rightarrow A : \text{ok} \]

\[ A \rightarrow B : \{ NA, NA \}_{KAB} \]

\[ B \rightarrow A : \{ Secret \}_{NA} \]

the protocol is insecure iff \( s \) and \( t \) are unifiable.
Deciding Security (finite sessions - active intruder)

Amadio, Lugiez 2000: atomic keys, decidable
Chevalier, Vigneron 2001; Millen Shmatikov 2001: composed keys, decidable
Turuani R. 2001: composed keys, NP
Comon Shmatikov 2003: constraint solving for XOR
Millen Shmatikov 2004: Diffie-Hellman with less restrictions:
Delaune, Jacquemard 2004: Dolev Yao with explicit decryption
Cortier Zalinescu 2006: with an ordering
Goal-Oriented Constraint Resolution

Rules

\[ K, m \in \text{forge}(E) \]
\[ \{m\}_K \in \text{forge}(E) \]

\[ l\sigma \in \text{forge}((E \cup m')\sigma) \]
\[ m, l \in \text{forge}(E \cup m') \]
where \[ m\sigma = m'\sigma \]

Example

\[ K, a, x \in \text{forge}(E) \]
\[ \{a, x\}_K \in \text{forge}(E) \]
Experimentations
<table>
<thead>
<tr>
<th>Problem</th>
<th>Roles</th>
<th>Steps</th>
<th>Attack</th>
<th>Time</th>
<th>Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAMobileIP - secrecy</td>
<td>5</td>
<td>9</td>
<td>NO</td>
<td>614.97</td>
<td>2</td>
</tr>
<tr>
<td>AAAMobileIP - w.auth</td>
<td>5</td>
<td>9</td>
<td>NO</td>
<td>2031.9</td>
<td>2</td>
</tr>
<tr>
<td>ChapV2 - secrecy</td>
<td>2</td>
<td>4</td>
<td>NO</td>
<td>0.04</td>
<td>4</td>
</tr>
<tr>
<td>ChapV2 - s.auth.</td>
<td>2</td>
<td>4</td>
<td>NO</td>
<td>0.02</td>
<td>4</td>
</tr>
<tr>
<td>EKE - secrecy</td>
<td>2</td>
<td>5</td>
<td>NO</td>
<td>0.15</td>
<td>3</td>
</tr>
<tr>
<td>EKE - s.auth.</td>
<td>2</td>
<td>5</td>
<td>YES</td>
<td>0.07</td>
<td>3</td>
</tr>
<tr>
<td>ISO-PK1 - s.auth.</td>
<td>2</td>
<td>1</td>
<td>YES</td>
<td>0.02</td>
<td>3</td>
</tr>
<tr>
<td>ISO-PK2 - s.auth.</td>
<td>2</td>
<td>2</td>
<td>NO</td>
<td>0.02</td>
<td>4</td>
</tr>
<tr>
<td>ISO-PK3 - w.auth.</td>
<td>2</td>
<td>2</td>
<td>YESNEW</td>
<td>0.02</td>
<td>4</td>
</tr>
<tr>
<td>ISO-PK4 - s.auth.</td>
<td>2</td>
<td>3</td>
<td>NO</td>
<td>0.39</td>
<td>5</td>
</tr>
<tr>
<td>KerberosV - secrecy</td>
<td>4</td>
<td>6</td>
<td>NO</td>
<td>115.35</td>
<td>3</td>
</tr>
<tr>
<td>KerberosV - w.auth.</td>
<td>4</td>
<td>6</td>
<td>NO</td>
<td>178.29</td>
<td>3</td>
</tr>
<tr>
<td>SHARE - secrecy</td>
<td>2</td>
<td>4</td>
<td>NO</td>
<td>0.05</td>
<td>3</td>
</tr>
<tr>
<td>SHARE - w.auth.</td>
<td>2</td>
<td>4</td>
<td>YES</td>
<td>0.03</td>
<td>3</td>
</tr>
<tr>
<td>TLS - secrecy</td>
<td>2</td>
<td>4</td>
<td>NO</td>
<td>41.47</td>
<td>4</td>
</tr>
<tr>
<td>TLS - s.auth.</td>
<td>2</td>
<td>4</td>
<td>NO</td>
<td>36.82</td>
<td>4</td>
</tr>
<tr>
<td>UMTS-AKA - secrecy</td>
<td>2</td>
<td>3</td>
<td>NO</td>
<td>0.08</td>
<td>5</td>
</tr>
<tr>
<td>UMTS-AKA - w.auth.</td>
<td>2</td>
<td>3</td>
<td>NO</td>
<td>0.40</td>
<td>5</td>
</tr>
</tbody>
</table>

**Legend:**

- **YES**: a known attack was found;
- **NO**: no attacks were found;
- **YESNEW**: a new attack in the typed model was found;

July 2006
Correction of H.530 Protocol of the ITU by AVISPA/ETHZ

H.323 MT

compute DH: $g^x \mod p$

1.) $GRQ(EP_B, GK_B, 0, CH_i, T_1, g^x, HMAC_{ZZ}(GRQ))$

compute DH: $g^y \mod p$

W: $g^x \oplus g^y$

2.) RIP(...)

K: $g^y \mod p$

13.) $GCF(GK_B, EP_B, CH_i, CH_1(T_{11}), g^y, HMAC_{ZZ}(W), HMAC_{ZZ}(GK_B), HMAC(GCF))$

K: $g^y \mod p$

W: $g^x \oplus g^y$

14.) $RRQ(EP_B, GK_B, CH_2, CH_3, (T_{12}), HMAC(RRQ))$

15.) $RCF(GK_B, EP_B, CH_3, CH_i, (T_{15}), HMAC(RCF))$

AuthenticationRequest ($GRQ(...), GK_B, W, HMAC$)

AuthenticationConfirmation ($HMAC_{ZZ}(W), HMAC_{ZZ}(GK_B), HMAC$)

July 2006
Correction of H.530 Protocol of the ITU by AVISPA/ETHZ

1. MT → VGK : MT, VGK, NIL, CH1, {G}DHX, 
   F(ZZ, MT, VGK, NIL, CH1, {G}DHX)
2. VGK → AuF : MT, VGK, NIL, CH1, {G}DHX, 
   F(ZZ, MT, VGK, NIL, CH1, {G}DHX), 
   VGK, {G}DHX XOR {G}DHY, 
   F(ZZ, MT, VGK, NIL, CH1, {G}DHX), 
   VGK, {G}DHX XOR {G}DHY)
3. AuF → VGK : VGK, MT, F(ZZ, VGK), 
   F(ZZ, {G}DHX XOR {G}DHY), 
   F(ZZ, VGK, MT, F(ZZ, VGK), 
   F(ZZ, {G}DHX XOR {G}DHY))
4. VGK → MT : VGK, MT, CH1, CH2, {G}DHY, 
   F(ZZ, {G}DHX XOR {G}DHY), 
   F(ZZ, VGK), 
   F({{G}DHX}DHY, VGK, MT, CH1, CH2, {G}DHY, 
   F(ZZ, {G}DHX XOR {G}DHY), F(ZZ, VGK))
5. MT → VGK : MT, VGK, CH2, CH3, 
   F({{G}DHX}DHY, MT, VGK, CH2, CH3)
6. VGK → MT : VGK, MT, CH3, CH4, 
   F({{G}DHX}DHY, VGK, MT, CH3, CH4)
Implementation of Needham-Schroeder Protocol with
El-Gamal encryption

\[ A \rightarrow B : (\alpha_B^{x_1} , \alpha_B^{k_B \cdot x_1} \oplus (A, N_a)) \]

\[ B \rightarrow A : (\alpha_B^{x_2} , \alpha_A^{k_A \cdot x_2} \oplus (N_a, N_b)) \]

\[ A \rightarrow B : (\alpha_B^{x_3} , \alpha_B^{k_B \cdot x_3} \oplus (N_b)) \]

Need of three disjoint theories

- $\oplus$

- pairing

- abelian group and exponentiation
Examples of Decidable Theories of Interest

- Encryption/decryption with explicit destructors
- Pair
- Exclusive or
- Exclusive or with homomorphism
- Abelian group
- Exponentiation with abelian group

*Remark:* these results are ad hoc and cannot be combined as is.
An Alternative Equational Intruder Model

A set of rules $\mathcal{L}$ of type

$x_1, \ldots, x_n \rightarrow t(x_1, \ldots, x_n)$

where $t$ is a term

and

An equational theory $\mathcal{E}$

Example: Exclusive or

$x, y \rightarrow x \oplus y$  
$\rightarrow 0$

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

Example: Dolev Yao with explicit decryption [DJ04]

$x, y \rightarrow \text{se}(x, y)$
$x, y \rightarrow \text{sd}(x, y)$

and

$\text{sd}(\text{se}(x, y), y) = x$
Intruder Deductions and Constraints

We are interested in derivations: $S \ (\rightarrow_L \cup =_\varepsilon)^* \ T$

Example: $a, a \oplus b \rightarrow_L a, a \oplus b, (a \oplus b) \oplus a =_\varepsilon a, a \oplus b, b$

and constraints: Given $S$ and $T$, find $\sigma$ such that

$$S\sigma \ (\rightarrow_L \cup =_\varepsilon)^* \ T\sigma$$

Remark: Generalizes equational unification

Notation: $E \triangleright m$ means: the intruder has to find a substitution $\sigma$ s.t. there exists a derivation from $E\sigma$ to a set containing $m\sigma$
Application to Protocol Analysis

Protocol:

\[
\begin{align*}
A \rightarrow B & : N_A \\
B \rightarrow A & : N_A \oplus k
\end{align*}
\]

Intruder initial knowledge: A,B

He can get the secret key \( k \) in one run if:

\[
\begin{align*}
A, B, N_A & \triangleright x \\
A, B, N_A, x \oplus k & \triangleright k
\end{align*}
\]

can be solved in \( x \)
1. A signature $\mathcal{G}$ defining operations.

2. A set of terms $\mathcal{S}$ representing the possible deductions of the intruder. 
   \textit{Deductions rules:} Given $t \in \mathcal{S}$ the intruder may apply any ground instance of the rule: $\text{Var}(t) \rightarrow t$

3. An equational theory $\mathcal{E}$ e.g. for “evaluating” some operators (destructors).

\textbf{Example:}

$$\pi_1(\text{pair}(x, y)) = x$$
$$\text{dec}(\{x\}_y, y) = x$$

\textit{Notation:} Intruder systems are denoted $\langle \mathcal{G}, \mathcal{S}, \mathcal{E} \rangle$. 

Protocol Model

The constraint $E \triangleright m$ means $m$ has to be composed from $E$ by the intruder.

Unification constraints represent the checks the actors of the protocol can do on messages.

Example: Message $m$ is a pair

$$m \equiv \text{pair}(x, y)$$
Example

Protocol:
\[
\begin{align*}
A &\rightarrow B : \text{pair}(N_A, A) \\
B &\rightarrow A : N_A \oplus k
\end{align*}
\]

Intruder initial knowledge: A, B

Constraints to solve:
\[
\begin{align*}
A, B, N_A &\triangleright v_1 \\
A, B, N_A, \pi_1(v_1) \oplus k &\triangleright v_2 \\
v_1 &\overset{?}{=} \text{pair}(v_0, A) \\
v_2 &\overset{?}{=} k
\end{align*}
\]

In general: An execution is represented by a constraint system
\[
((E_i \triangleright v_i)_{i:1..n}, \bigwedge_{\alpha} r_\alpha \overset{?}{=} t_\alpha)
\]

July 2006
Deterministic Constraint Systems

- The variables appearing in sent messages correspond to previously received messages (unlike: $A, B \triangleright x; \ A, B, x \oplus y \triangleright z$)

- This leads to a special type of constraint systems called *deterministic*.

A constraint system

$((E_i \triangleright v_i)_{i:1..n}, \bigwedge_{\alpha} r_{\alpha} \ ? \ t_{\alpha})$

is *deterministic* if for all $i$ one has $\text{Var}(E_i) \subseteq \{v_1, \ldots, v_{i-1}\}$

*Remark*: no restriction on unification problems
Ordered satisfiability

• Input: A constraint system $\mathcal{C}$ and a total ordering $\prec$ on constants and variables of $\mathcal{C}$

• A substitution $\sigma$ satisfies $\mathcal{C}$ w.r.t order $\prec$ if $\sigma$ satisfies $\mathcal{C}$ and for all variable $x$ and for all constant $c$ one has:

$$x \prec c \implies c \notin \text{Sub}(x\sigma)$$

Remark: Permits to avoid cyclic solutions

July 2006
**Ordered satisfiability - Example**

Case of an abelian group operator $+$:

$$\left(a, b \triangleright x, a, b, x - 3c \triangleright z, y \overset{?}{=} 2x, a \prec x \prec b \prec y \prec c \prec z\right)$$

The constraint is equivalent to:

$$\left(a, b \triangleright x, a, b, -3c \triangleright z, y \overset{?}{=} 2x, a \prec x \prec b \prec y \prec c \prec z\right)$$

which is equivalent to a linear system:

$$\begin{align*}
x_b &= x_c = y_b = y_c = 0, \\
y_a &= 2x_a, \\
z_c &= -3\lambda
\end{align*}$$

where $x_i$ is the coefficient of $i$ in $x$. 

July 2006
Combining Intruders/ Well-formed derivations

$\langle G_1, S_1, E_1 \rangle$ and $\langle G_2, S_2, E_2 \rangle$ are two intruders over disjoint signatures $G_1$ and $G_2$

A subterm value of $t$ is either $t$ or a strict maximal alien subterm of a subterm value of $t$.

Example: $F_1 = \{ \oplus, a, b, c \}$ and $F_2 = \{ f \}$

$$\begin{align*}
\text{Sub}(f(f(a)) \oplus (b \oplus c)) &= \{ f(f(a)) \oplus (b \oplus c), f(f(a)), a, b, c \} \\
\text{Sub}(f(f(f(b) \oplus c))) &= \{ f(f(f(b) \oplus c)), f(b) \oplus c, f(b), b, c \} \\
\text{Sub}(0) &= \{ 0 \}
\end{align*}$$

A derivation $E_0 \rightarrow_\mathcal{U} E_0, t_1 \rightarrow_\mathcal{U} \cdots \rightarrow_\mathcal{U} E_n$ of intruder system $\mathcal{U} = \langle G_1, S_1, E_1 \rangle \cup \langle G_2, S_2, E_2 \rangle$ is well-formed if every message generated by an intermediate step is a subterm value of the goal, the initial set of messages or a constant.

Lemma: A derivation of minimal length starting from $E$ of goal $t$ is well-formed.
Combining Intruders (Chevalier R. 05)

If:

• \( \langle G_1, S_1, E_1 \rangle \) and \( \langle G_2, S_2, E_2 \rangle \) are two intruders over two disjoint signatures \( G_1 \) and \( G_2 \)

• the ordered satisfiability problems are decidable for deterministic constraints systems over signatures \( G_1 \) and \( G_2 \)

Then:

satisfiability problems are decidable for deterministic constraint systems over the signature \( G_1 \cup G_2 \) w.r.t. the intruder \( \langle G_1 \cup G_2, S_1 \cup S_2, E_1 \cup E_2 \rangle \)

cf: Baader-Schulz and Schmidt-Schauß combination of unifiers results.
Conclusion

- protocol modelling with (set of) terms rewriting
- insecurity decidable for finite sessions for many equational theories in NP for XOR, Diffie-Hellman exponentiation, commuting public keys
- practical procedures already used by industry (e.g. Siemens)
Some other directions

• group protocols - open-ended datastructures (transactions lists, security associations, messages transducers . . . )

• contract-signing protocols - complex properties such as fairness or abuse-freeness (no party can prove to a third party that it has the power to both enforce and cancel the contract).

• strong secrecy: two execution of the protocol with distinct instances for secret are indistinguishable to adversary

• relation between the symbolic and computational models: when does inability of deriving the secret in the formal world entails indistinguishability of adversary’s view in the computational world?
Questions ?