1 Introduction
2 Example of the security protocol
3 Formal model
4 Chains of states
5 The method
6 Experimental results
7 Conclusions
8 References
The importance of security protocols

- Security protocols - a key point of safety
- Used in many areas of computer science
- Errors in the protocol’s design can be found
- The need of specification and verification of SP
- The need of full and formal description of the protocol
- Deductive and algorithmic methods of verification
The verification system AVISPA (Automated Validation of Internet Security Protocols and Applications) was designed and implemented as the result of the EU research project.

The AVISPA is composed of four modules:
Few words about AVISPA II

- AVISPA detected a number of previously unknown attacks on some of the protocols analysed, eg on some protocols of the ISO-PK family, on the IKEv2-DS protocol, and on the H.530 protocol.
- In several cases other verification methods can be more effective:
Motivation

- For convenience
- To be faster
- To build a tool base for further research
Our approach

User A

Communication

User B
Our approach

User A <-> User B

Communication

Formal Language
Our approach

User A ← Communication → User B

Formal Language
Our approach

User A ← Communication → User B

Formal Language ← Communication → Computational Structure
Our approach

User A <-> User B

Communication

Formal Language -> Computational Structure

Chain of states
Needham-Schroeder protocol and Lowe’s attack

Needham-Schroeder protocol:

\[ \alpha_1 \ A \rightarrow B : \langle N_A \cdot i(A) \rangle_{K_B}, \]
\[ \alpha_2 \ B \rightarrow A : \langle N_A \cdot N_B \rangle_{K_A}, \]
\[ \alpha_3 \ A \rightarrow B : \langle N_B \rangle_{K_B}. \]  
(1)

Lowe’s attack:

\[ \alpha_1^1 \ A \rightarrow \iota : \langle N_A \cdot i(A) \rangle_{K_\iota}, \]
\[ \alpha_1^2 \ i(A) \rightarrow B : \langle N_A \cdot i(A) \rangle_{K_B}, \]
\[ \alpha_2^2 \ B \rightarrow i(A) : \langle N_A \cdot N_B \rangle_{K_A}, \]
\[ \alpha_2^1 \ i \rightarrow A : \langle N_A \cdot N_B \rangle_{K_A}, \]
\[ \alpha_3^1 \ A \rightarrow \iota : \langle N_B \rangle_{K_\iota}, \]
\[ \alpha_3^2 \ i(A) \rightarrow B : \langle N_B \rangle_{K_B}. \]  
(2)
Basic notations

Definition

- \( P = \{P_1, P_2, \ldots, P_{nP}\} \) - a set of the honest participants in the network,
- \( P_\ell = \{\ell, \ell(P_1), \ell(P_2), \ldots, \ell(P_{nP})\} \) - a set of the dishonest participants containing the Intruder and the Intruder impersonating the participant \( P_i \) for \( 1 \leq i \leq n_P \),
- \( I = \{i(P_1), \ldots, i(P_{nP}), \ell\} \) - a set of the identifiers of the participants in the network,
- \( K = \bigcup_{i=1}^{nP} \{K_{P_i}, K_{P_i}^{-1}\} \cup \{K_\ell, K_\ell^{-1}\} \) - a set of the public and private cryptographic keys (already existing or possible to be generated) of the participants,
- \( N = \bigcup_{i=1}^{nP} \{N_{P_i}^1, \ldots, N_{P_i}^{k_N}\} \cup \{N_1^\ell, \ldots, N_{k_N}^\ell\} \) - a set of the nonces
A set of letters

Definition

By a set of letters $L$ we mean the smallest set satisfying the following conditions:

1. $P \cup P_i \cup I \cup K \cup N \subseteq L$,
2. If $X, Y \in L$, then the concatenation $X \cdot Y \in L$,
3. If $X \in L$ and $K \in K$, then $\langle X \rangle_K \in L$, $\langle X \rangle_K$ is a ciphertext consisting of the letter $X$ encrypted with the key $K$. 
The protocol

Definition

The protocol $\Pi$ is a sequence of steps defined as ordered five-tuples:

$$\alpha = (P, Q, M, G, K).$$

In such step $P$ is the step initiator (sending part), $Q$ - a message recipient, $M$ - a sent message, $G$ - a set of information required in order to be generated by $P$ for the execution of the step $\alpha$ and $K$ is a set of information required for $P$ in order to send $M$.

Assume the following notation: if $\alpha = (P, Q, M, G, K)$, then by $Send(\alpha)$, $Rec(\alpha)$, $Mess(\alpha)$, $Gen(\alpha)$, $Know(\alpha)$ we mean the following elements: $P$, $Q$, $M$, $G$, $K$. 
Many executions

Examples

\[ \alpha_1^1 = (A, B, \langle N_A \cdot i(A) \rangle K_B, \{N_A\}, \{i(A), N_A, K_B\}), \]
\[ \alpha_2^1 = (B, A, \langle N_A \cdot N_B \rangle K_A, \{N_B\}, \{N_A, N_B, K_A\}), \]
\[ \alpha_3^1 = (A, B, \langle N_B \rangle K_B, \emptyset, \{N_B, K_B\}). \]

\[ \alpha_1^2 = (A, C, \langle N_A \cdot i(A) \rangle K_C, \{N_A\}, \{i(A), N_A, K_C\}), \]
\[ \alpha_2^2 = (C, A, \langle N_A \cdot N_C \rangle K_A, \{N_C\}, \{N_A, N_C, K_A\}), \]
\[ \alpha_3^2 = (A, C, \langle N_C \rangle K_C, \emptyset, \{N_C, K_C\}). \]
Consider the following finite sequence of the execution of the protocol’s steps: \( R = \alpha_{k_1}^{i_1}, \alpha_{k_2}^{i_2}, \alpha_{k_3}^{i_3}, \ldots, \alpha_{k_s}^{i_s} \), where in denoting a step \( \alpha \) the superscript indicates the number of execution, and the subscript indicates the number of the step in the given execution.

If we consider the two different executions of the same protocol, which consist of three steps, \( \alpha_1^1, \alpha_2^1, \alpha_3^1 \) and \( \alpha_1^2, \alpha_2^2, \alpha_3^2 \), the possible sequence is:

\[
R = \alpha_1^1, \alpha_2^1, \alpha_1^2, \alpha_3^1, \alpha_2^2, \alpha_3^2.
\]
The knowledge of users

Definition

Consider any $j = 1, \ldots, s - 1$ for any sequence $R = \alpha_{k_1}^{i_1}, \alpha_{k_2}^{i_2}, \alpha_{k_3}^{i_3}, \ldots, \alpha_{k_s}^{i_s}$. For every user $p \in P$ we have:

$$
\text{Know}_{p}^{j+1} = \begin{cases} 
\text{Know}_{p}^{j} \cup \text{Gen}(\alpha_{k_j}^{i_{j+1}}) & \text{if } p = \text{Send}(\alpha_{k_j}^{i_{j+1}}), \\
\kappa(\text{Know}_{p}^{j} \cup \{\text{Mess}(\alpha_{k_j}^{i_{j+1}})\}) & \text{if } p = \text{Resp}(\alpha_{k_j}^{i_{j+1}}), \\
\text{Know}_{p}^{j} & \text{otherwise.}
\end{cases}
$$
The run

Definition

By the run we call any finite sequence of the steps of protocol’s executions \( t = \alpha^1_{k_1}, \alpha^2_{k_2}, \alpha^3_{k_3}, \ldots, \alpha^s_{k_s} \) that meets the following conditions:

1. \( \forall j=1,\ldots,s \left[ k_j = 1 \lor \exists t<j \left( i_t = i_j \land k_t = k_j - 1 \right) \right], \)

2. \( \forall j=2,\ldots,s \left[ \text{Know}(\alpha^i_{k_j}) \subseteq \text{Know}^j \left( \text{Send}(\alpha^i_{k_j}) \cup \text{Gen}(\alpha^i_{k_j}) \right) \right]. \)
Types of states

1. $S^i_j$ - the execution of $i$-th step in the $j$-th execution
2. $G^N_A$ - the nonce/key $N_A$ generated by user $A$
3. $K^X_A$ - user $A$ acquired message $X$
4. $P^X_A$ - user $A$ has to possess knowledge of element $X$ in order to carry out a given step
Chains for NSPK

\[
\alpha_1^1 = (G^NA, S_1^1, K^NA_B),
\]

\[
\alpha_2^1 = (P^NA_B, G^NB_B, S_2^1, K^NB_A),
\]

\[
\alpha_3^1 = (P^NB_A, S_3^1).
\]

The set of the states preceding the state corresponding with the execution of the steps \( S \) will be marked hereinafter by \( \text{PreCond}(S) \). Accordingly, by using \( \text{PostCond}(S) \) we will mark the set of the states found in the sequence after the state \( S \).
Intruder - Dolev-Yao model

- if Intruder’s knowledge is enough it can execute protocol steps as an another participant
- if Intruder has a right key it can decrypt received ciphers
- Intruder can use nonces and timestamps many times

Attacks

- *attacks for authentication* – an attack exists if an execution of the protocol in which Intruder uses identifiers of another user (impersonating) is possible
- *attacks for secrecy*
- *reflexion attacks*
Intruder - Dolev-Yao model

- if Intruder’s knowledge is enough it can execute protocol steps as an another participant
- if Intruder has a right key it can decrypt received ciphers
- Intruder can use nonces and timestamps many times

Attacks

- **attacks for authentication** – an attack exists if an execution of the protocol in which Intruder uses identifiers of another user (impersonating) is possible
- **attacks for secrecy**
- **refflexion attacks**
Example with intruder

Example

If the sent message is ciphertext $\langle N_A \rangle K_A \cdot \langle N_B \rangle K_B$, the message can be composed in five ways. In each case the Intruder can compose and use during the execution of the protocol the message:

- $X_1 = \{\langle N_A \rangle K_A \cdot \langle N_B \rangle K_B\}$,
- $X_2 = \{\langle N_A \rangle K_A, \langle N_B \rangle K_B\}$,
- $X_3 = \{N_A, K_A, \langle N_B \rangle K_B\}$,
- $X_4 = \{\langle N_A \rangle K_A, N_B, K_B\}$,
- $X_5 = \{N_A, K_A, N_B, K_B\}$.

In each case the Intruder can compose and use during the execution of the protocol the message $\langle N_A \rangle K_A, \langle N_B \rangle K_B$. The fact, that from a given set $X$ a message $M$ can be composed, is denoted by $X \vdash M$. 
Corresponding chains for the Lowe’s attack:

\[ \alpha_1^1 = (G_A^{N_A}, S_1^1, K_l^{N_A}), \]
\[ \alpha_1^2 = (P_l^{\langle N_A \cdot N_B \rangle_{K_A}}, S_2^1, K_A^{N_B}), \]
\[ \alpha_1^3 = (P_A^{N_B}, S_3^1, K_l^{N_B}), \]
\[ \alpha_2^1 = (P_l^{N_A}, S_1^2, K_B^{N_A}), \]
\[ \alpha_2^2 = (P_B^{N_A}, G_B^{N_B}, S_2^2, K_l^{\langle N_A \cdot N_B \rangle_{K_A}}), \]
\[ \alpha_2^3 = (P_l^{N_B}, S_3^2). \]
A correct chain of states

Definition

We call the sequence of the protocol’s states: \( s = s_1, s_2, \ldots \) a **correct chain of states** iff the following conditions holds:

1. if \( s_i = S_j^k \) for some \( j, k \) then \( j = 1 \lor \exists t < i (s_t = S_{j-1}^k) \) and 
   \( \text{PreCond}(S_j^k) \subseteq \{s_1, \ldots, s_{i-1}\} \land \text{PostCond}(S_j^k) \subseteq \{s_{i+1}, \ldots\} \),
2. if \( s_i = G_U^X \), then \( \forall t \neq i (s_t \neq G_U^X) \),
3. if \( s_i = P_U^X \), then \( \exists t < i (s_t = G_U^X \lor s_t = K_U^X) \).
The method
## Experimental results

<table>
<thead>
<tr>
<th>Protocol</th>
<th>AVISPA</th>
<th>VerICS</th>
<th>Chains</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EnT (ms)</td>
<td>SolT (ms)</td>
<td>EnT (ms)</td>
</tr>
<tr>
<td>NSPK</td>
<td>90</td>
<td>&lt; 10</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>NSPK(_{Lowe})</td>
<td>90</td>
<td>&lt; 10</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>UnT_WMF</td>
<td>NA</td>
<td>NA</td>
<td>&lt; 1</td>
</tr>
</tbody>
</table>
Experimental results

<table>
<thead>
<tr>
<th>Protocol</th>
<th>VerICS</th>
<th>Chains</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>automata</td>
<td>variables</td>
</tr>
<tr>
<td>NSPK</td>
<td>32</td>
<td>5226</td>
</tr>
<tr>
<td>NSPK_Lowe</td>
<td>32</td>
<td>9723</td>
</tr>
<tr>
<td>UnT_WMF</td>
<td>20</td>
<td>4331</td>
</tr>
<tr>
<td>DY</td>
<td>35</td>
<td>6153</td>
</tr>
<tr>
<td>TMN</td>
<td>24</td>
<td>4821</td>
</tr>
</tbody>
</table>
Conclusions

- Our approach is simple and convenient
- The method has been implemented
- The obtained results are very promising
- A research on further optimization of the method and its implementation, as well as its application for other protocols is in progress
- The adaptation of the method for time dependent protocols has been planned


