A New Effective Approach for Modelling and Verification of Security Protocols

Mirosław Kurkowski, Olga Siedlecka-Lamch, Henryk Piech

Institute of Computer and Information Sciences
Czestochowa University of Technology

September 28, 2012

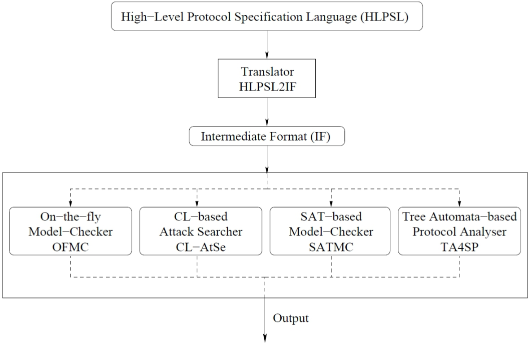
The importance of security protocols

Mirosław Kurkowski, Olga Siedlecka-Lamch, Henryk Piech

- 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

Few words about AVISPA I

- 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:
  Kurkowski, M., Penczek, W.: Verifying Security Protocols Modeled by Networks of Automata, Fund. Inform., Vol. 79 (3-4), pp. 453-471, IOS Press 2007

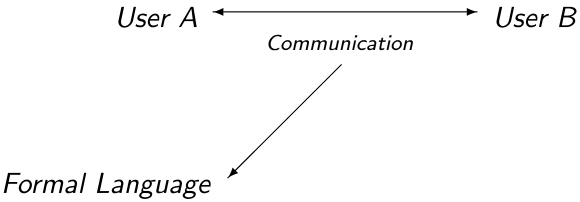
Motivation

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

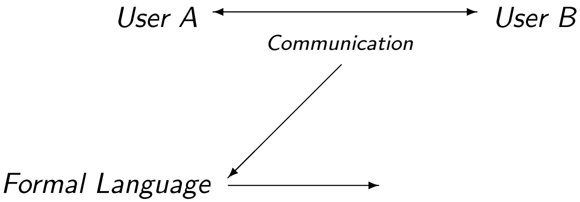
Our approach

$$\textit{User A} \xleftrightarrow[\textit{Communication}]{\hspace{4cm}} \textit{User B}$$

Our approach

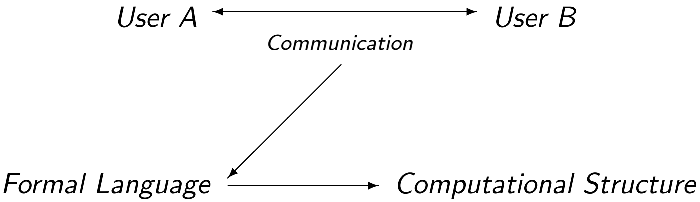


Our approach

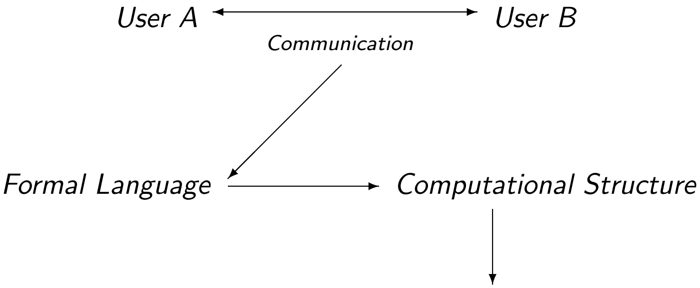


Our approach

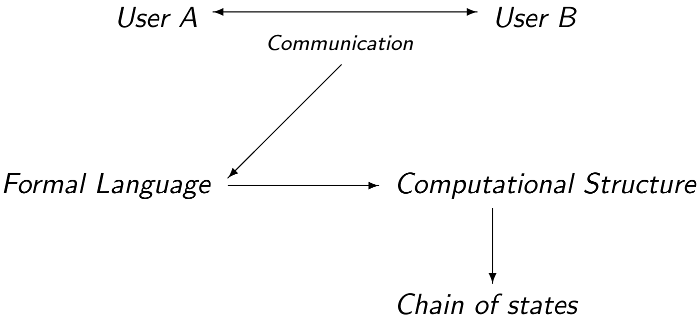
Mirosław Kurkowski, Olga Siedlecka-Lamch, Henryk Piech



Our approach



Our approach



Needham-Schroeder protocol:

$$\begin{array}{llll} \alpha_1 & A & \rightarrow B & : \langle N_A \cdot i(A) \rangle_{K_B}, \\ \alpha_2 & B & \rightarrow A & : \langle N_A \cdot N_B \rangle_{K_A}, \\ \alpha_3 & A & \rightarrow B & : \langle N_B \rangle_{K_B}. \end{array} \tag{1}$$

Lowe's atack:

$$\begin{array}{llll} \alpha_1^1 & A & \rightarrow \iota & : \langle N_A \cdot i(A) \rangle_{K_\iota}, \\ \quad \alpha_1^2 & \iota(A) & \rightarrow B & : \langle N_A \cdot i(A) \rangle_{K_B}, \\ \quad \alpha_2^2 & B & \rightarrow \iota(A) & : \langle N_A \cdot N_B \rangle_{K_A}, \\ \alpha_2^1 & \iota & \rightarrow A & : \langle N_A \cdot N_B \rangle_{K_A}, \\ \alpha_3^1 & A & \rightarrow \iota & : \langle N_B \rangle_{K_\iota}, \\ \quad \alpha_3^2 & \iota(A) & \rightarrow B & : \langle N_B \rangle_{K_B}. \end{array} \tag{2}$$

Basic notations

Definition

- $\mathbf{P} = \{P_1, P_2, \ldots, P_{n_P}\}$ - a set of the honest participants in the network,
- $\mathbf{P}_\iota = \{\iota, \iota(P_1), \iota(P_2), \ldots, \iota(P_{n_P})\}$ - a set of the dishonest participants containing the Intruder and the Intruder impersonating the participant P_i for $1 \leqslant i \leqslant n_P$,
- $\mathbf{I} = \{i(P_1), \ldots, i(P_{n_P}), i_\iota\}$ - a set of the identifiers of the participants in the network,
- $\mathbf{K} = \bigcup_{i=1}^{n_P}\{K_{P_i}, K_{P_i}^{-1}\} \cup \{K_\iota, K_\iota^{-1}\}$ - a set of the public and private cryptographic keys (already existing or possible to be generated) of the participants,
- $\mathbf{N} = \bigcup_{i=1}^{N_P}\{N_{P_i}^1, \ldots, N_{P_i}^{k_N}\} \cup \{N_\iota^1, \ldots, N_\iota^{k_N}\}$ - a set of the *nonces*

A set of letters

Definition

By *a set of letters* $\mathbf{L}$ we mean the smallest set satisfying the following conditions:

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

The protocol

Definition

The protocol Π 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 α 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

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

$$\alpha_1^2 = (A, C, \langle N_A \cdot i(A) \rangle_{K_C}, \{N_A\}, \{i(A), N_A, K_C\}),$$
$$\alpha_2^2 = (C, A, \langle N_A \cdot N_C \rangle_{K_A}, \{N_C\}, \{N_A, N_C, K_A\}),$$
$$\alpha_3^2 = (A, C, \langle N_C \rangle_{K_C}, \emptyset, \{N_C, K_C\}).$$

Sequence of executions

Consider the following finite sequence of the execution of the protocol's steps: $\mathcal{R} = \alpha_{k_1}^{i_1}, \alpha_{k_2}^{i_2}, \alpha_{k_3}^{i_3}, \ldots, \alpha_{k_s}^{i_s}$, where in denoting a step α 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:

$$\mathcal{R} = \alpha_1^1, \alpha_2^1, \alpha_1^2, \alpha_3^1, \alpha_2^2, \alpha_3^2.$$

The knowledge of users

Definition

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

$$Know^{j+1}_p = \begin{cases} Know^j_p \ \cup \ Gen(\alpha^{i_{j+1}}_{k_{j+1}}) & \text{if} \quad p = Send(\alpha^{i_{j+1}}_{k_{j+1}}), \\ \kappa(Know^j_p \cup \{Mess(\alpha^{i_{j+1}}_{k_{j+1}}))\}) & \text{if} \quad p = Resp(\alpha^{i_{j+1}}_{k_{j+1}}), \\ Know^j_p & \text{othervise.} \end{cases}$$

The run

Definition

By the run we call any finite sequence of the steps of protocol's executions $\mathfrak{r} = \alpha_{k_1}^{i_1}, \alpha_{k_2}^{i_2}, \alpha_{k_3}^{i_3}, \ldots, \alpha_{k_s}^{i_s}$ that meets the following conditions:

1. $\forall_{j=1,\ldots,s}[k_j = 1 \vee \exists_{t<j}(i_t = i_j \ \wedge \ k_t = k_j - 1)]$,
2. $\forall_{j=2,\ldots,s}[Know(\alpha_{k_j}^{i_j}) \subseteq Know^{j-1}_{Send(\alpha_{k_j}^{i_j})} \cup Gen(\alpha_{k_j}^{i_j}))]$.

Types of states

Mirosław Kurkowski, Olga Siedlecka-Lamch, Henryk Piech

1. S_j^i - the execution of i-th step in the j-th execution
2. $G_A^{N_A}$ - the nonce/key N_A generated by user A
3. K_A^X - user A acquired message X
4. P_A^X - user A has to possess knowledge of element X in order to carry out a given step

Chains for NSPK

$$
\begin{aligned}
\alpha_1^1 &= (G_A^{N_A}, S_1^1, K_B^{N_A}),\\
\alpha_2^1 &= (P_B^{N_A}, G_B^{N_B}, S_2^1, K_A^{N_B}),\\
\alpha_3^1 &= (P_A^{N_B}, S_3^1).
\end{aligned}
$$

The set of the states preceding the state corresponding with the execution of the steps S will be marked hereinafter by $PreCond(S)$. Accordingly, by using $PostCond(S)$ we will mark the set of the states found in the sequence after the state S.

Intruder

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

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 $X \vdash M$.

Example with intruder

Corresponding chains for the Lowe's attack:

$$
\begin{aligned}
\alpha_1^1 &= (G_A^{N_A}, S_1^1, K_\iota^{N_A}),\\
\alpha_2^1 &= (P_\iota^{\langle N_A \cdot N_B \rangle_{K_A}}, S_2^1, K_A^{N_B})\\
\alpha_3^1 &= (P_A^{N_B}, S_3^1, K_\iota^{N_B}),
\end{aligned}
$$

$$
\begin{aligned}
\alpha_1^2 &= (P_\iota^{N_A}, S_1^2, K_B^{N_A}),\\
\alpha_2^2 &= (P_B^{N_A}, G_B^{N_B}, S_2^2, K_\iota^{\langle N_A \cdot N_B \rangle_{K_A}}),\\
\alpha_3^2 &= (P_\iota^{N_B}, S_3^2).
\end{aligned}
$$

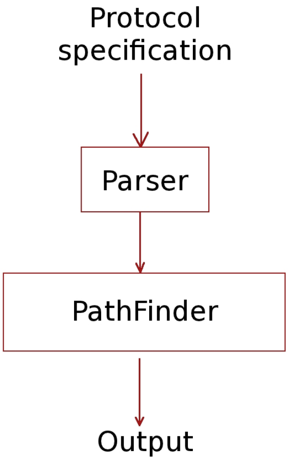
A correct chain of states

Definition

We call the sequence of the protocol's states: $\mathfrak{s} = s_1, s_2, \ldots$ **a correct chain of states** iff the following conditions holds:

1. if $s_i = S_j^k$ for some j, k then $j = 1 \vee \exists_{t<i}(s_t = S_{j-1}^k)$ and $PreCond(S_j^k) \subseteq \{s_1, \ldots, s_{i-1}\} \ \wedge \ PostCond(S_j^k) \subseteq \{s_{i+1}, \ldots\}$,
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 $\exists_{t<i}(s_t = G_U^X \ \vee \ s_t = K_U^X)$.

The method



Experimental results

Protocol	AVISPA		VerICS		Chains	
	EnT (ms)	SolT (ms)	EnT (ms)	SolT (ms)	EnT (ms)	SolT (ms)
NSPK	90	< 10	< 1	36	< 1	< 1
$NSPK_{Lowe}$	90	< 10	< 1	960	< 1	< 1
UnT_WMF	NA	NA	< 1	32	< 1	< 1

Experimental results

Protocol	VerICS				Chains	
	automata	variables	clauses	time (ms)	chains	time (ms)
NSPK	32	5226	15124	36	36	< 1
NSPK_{Lowe}	32	9723	28171	960	36	< 1
UnT_WMF	20	4331	11432	32	22	< 1
DY	35	6153	17735	41	24	< 1
TMN	24	4821	12533	32	40	< 1

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

References I

- Amnell, T. et all: UPPAAL - Now, Next, and Future, Proc. of the 4th Summer School 'Modelling and Verification of Parallel Processes' (MOVEP'00), LNCS, 2067, 99–124, Springer-Verlag, 2001.
- Armando, A. et all: The AVISPA tool for the automated validation of internet security protocols and applications. In Proc. of 17th Int. Conf. on Computer Aided Verification (CAV'05), vol. 3576 of LNCS, pp. 281–285. Springer, 2005.
- Armando, A., Compagna, L.: Sat-based model-checking for security protocols analysis. International Journal of Information Security, 7(1):3–32, 2008.
- Basin, D. A., Wolff, B. (editors): Theorem Proving in Higher Order Logics, 16th International Conference, TPHOLs 2003, Roma, Italy, September 8-12, 2003, Proceedings, volume 2758 of Lecture Notes in Computer Science. Springer, 2003.
- Bella, G., Massacci, F., and Paulson, L. C.: Verifying the set registration protocols. IEEE Journal on Selected Areas in Communications, 20(1):77–87, 2003.
- Bella G, Paulson L.C.: Using Isabelle to prove properties of the kerberos authentication system. In H. Orman and C. Meadows, editors, *Proc. of the DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.
- Burrows, M., Abadi, M., and Needham, R. M.: A logic of authentication. ACM Trans. Comput. Syst., 8(1):18–36, 1990.
- E. Cohen. Taps: A first-order verifier for cryptographic protocols. In *CSFW '00: Proceedings of the 13th IEEE Computer Security Foundations Workshop (CSFW'00)*, page 144, Washington, DC, USA, 2000. IEEE Computer Society.

References II

- Corin, R., Etalle, S., Hartel, P. H., and Mader, A.: Timed model checking of security protocols. In Proc. of the 2004 ACM Workshop on Formal Methods in Security Engineering (FMSE'04), pages 23–32. ACM, 2004.
- Delzanno, G., and Ganty, P.: Automatic verification of time sensitive cryptographic protocols.In Proc. of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04), volume 2988 of LNCS, pages 342–356. Springer, 2004.
- Dolev, D. and Yao, A.: On the security of public key protocols.IEEE Transactions on Information Theory, 29(2):198–207, 1983.
- Evans, N., and Schneider, S.: Analysing time dependent security properties in CSP using PVS. In Proc. of the 6th European Symposium on Research in Computer Security (ESORICS'00), vol. 1895 of LNCS, pp. 222–237. Springer, 2000.
- Gorrieri, R., Locatelli, E., and Martinelli, F.: A simple language for real-time cryptographic protocol analysis. In Proc. of the 12th European Symposium on Programming (ESOP'03), volume 2618 of LNCS, pages 114–128. Springer, 2003.
- Jakubowska, G., and Penczek, W.: Is your security protocol on time? In Proc. of FSEN'07, volume 4767 of LNCS, pages 65–80. Springer-Verlag, 2007.
- Jakubowska, G., and Penczek, W., and Srebrny, M.: Verifying security protocols with timestamps via translation to timed automata. In Proc. of the International Workshop on Concurrency, Specification and Programming (CS&P'05), pages 100–115. Warsaw University, 2005.
- Kacprzak, M., et. all : Verics 2007 - a model checker for knowledge and real-time. Fundam. Inform., 85(1-4):313–328, 2008

Chains for SP

Mirosław Kurkowski, Olga Siedlecka-Lamch, Henryk Piech

References III

- Kurkowski, M., Srebrny, M.: A Quantifier-free First-order Knowledge Logic of Authentication, Fund. Inform., vol. 72, pp. 263-282, IOS Press 2006.
- Kurkowski, M., Penczek, W.: Verifying Security Protocols Modeled by Networks of Automata, Fund. Inform., Vol. 79 (3-4), pp. 453-471, IOS Press 2007.
- Kurkowski, M., Penczek, W.: Verifying Timed Security Protocols via Translation to Timed Automata, Fund. Inform., vol. 93 (1-3), pp. 245-259, IOS Press 2009.
- Kurkowski M., Penczek W.: Applying Timed Automata to Model Checking of Security Protocols, in ed. J. Wang, Handbook of Finite State Based Models and Applications, pp. 223–254, CRC Press, Boca Raton, USA, 2012.
- Lowe, G.: Breaking and Fixing the Needham-Schroeder Public-key Protocol Using fdr., In TACAS, LNCS, Springer, 147-166, 1996
- Meadows C.: The NRL protocol analyzer: An overview. *Journal of Logic Programming*, 26(2):13–131, 1996.
- Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. Commun. ACM, 21(12), 993-999, 1978.
- Paulson, L. C.: Inductive analysis of the internet protocol tls. ACM Trans. Inf. Syst. Secur., 2(3):332–351, 1999.