Probabilistic Model Checking of Security Protocols without Perfect Cryptography Assumption

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

Czestochowa University of Technology
Cardinal Stefan Wyszynski University

CN2016

Importance of Security Protocols

- Key point of security systems
- Used in many areas
- Errors in: structure, operations, security
- Specification and verification importance
- Need for the complete formal model
- IT market development sets new requirements

New Challenges

- Large number of keys
- Is a perfect cryptography assumption today fully justified?
- "Tailor-made" security
- Probabilistic approach

World Leaders

- AVISPA
- UPPAAL
- PRISM

AVISPA

AVISPA

(Automated Validation of Internet Security Protocols and Applications)project aims at developing a push-button, industrial-strength technology for the analysis of large-scale Internet security-sensitive protocols and applications.

Avispa provide:

- specification language - HLPSL
- four modules looking for attack in four different ways

A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigano, L. Vigneron, The Avispa Tool for the automated validation of internet security protocols and applications, in proceedings of CAV 2005, Computer Aided Verification, LNCS 3576, Springer Verlag, 2005

UPPAAL

UPPAAL

is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.). It is appropriate for systems that can be modeled as a collection of non-deterministic processes with finite control structure and real-valued clocks, communicating through channels or shared variables.

Uppaal consists of three main parts:

- specification language
- simulator
- model-checker

Behrmann G., David A., Larsen K. G., A Tutorial on Uppaal, In proceedings of the 4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM-RT'04). LNCS 3185

PRISM

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

PRISM

PRISM is a probabilistic model checker, a tool for formal modelling and analysis of systems that exhibit random or probabilistic behaviour.

PRISM can build and analyse several types of probabilistic models:

- discrete-time Markov chains (DTMCs)
- continuous-time Markov chains (CTMCs)
- Markov decision processes (MDPs)
- probabilistic automata (PAs)
- probabilistic timed automata (PTAs)

Models are described using the PRISM language.

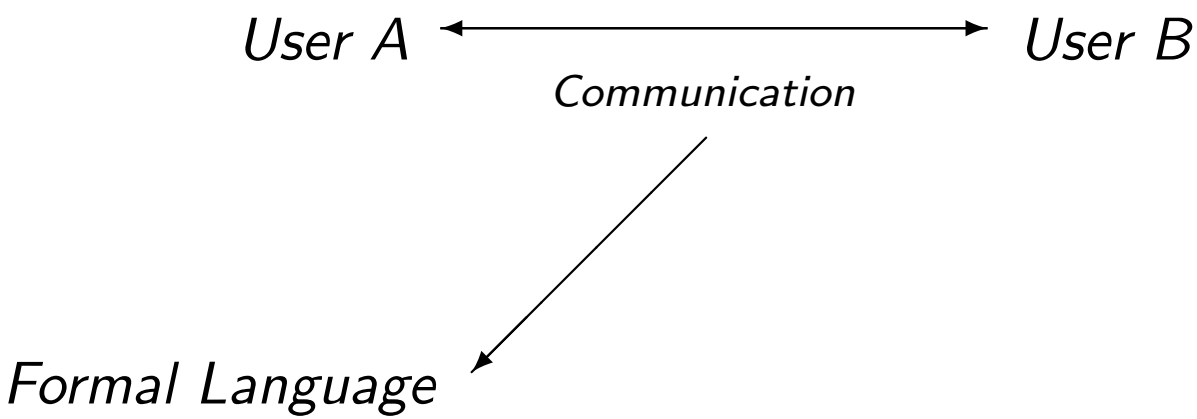
Kwiatkowska M., Norman G. and Parker D.. PRISM 4.0: Verification of Probabilistic Real-time Systems. In Proc. 23rd International Conference on Computer Aided Verification (CAV'11), volume 6806 of LNCS, pages 585-591, Springer, 2011

Our Approach

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

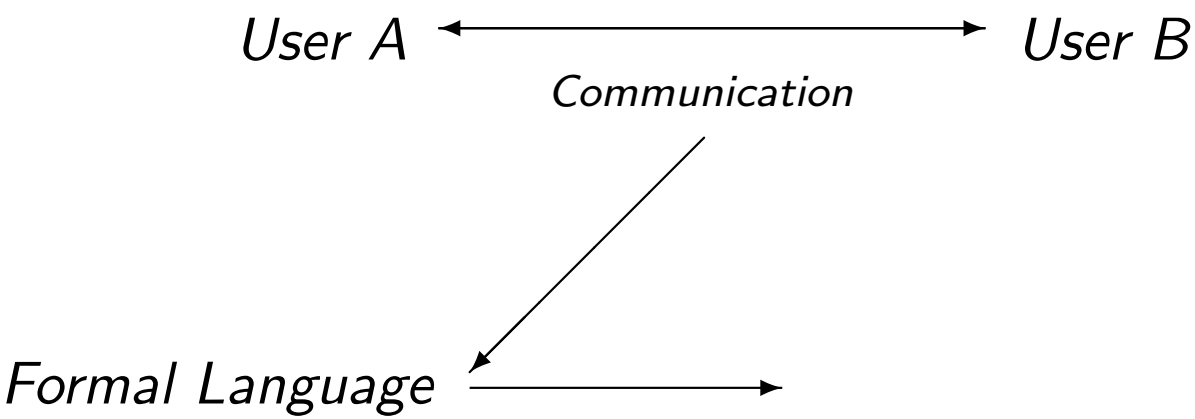
Our Approach

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski



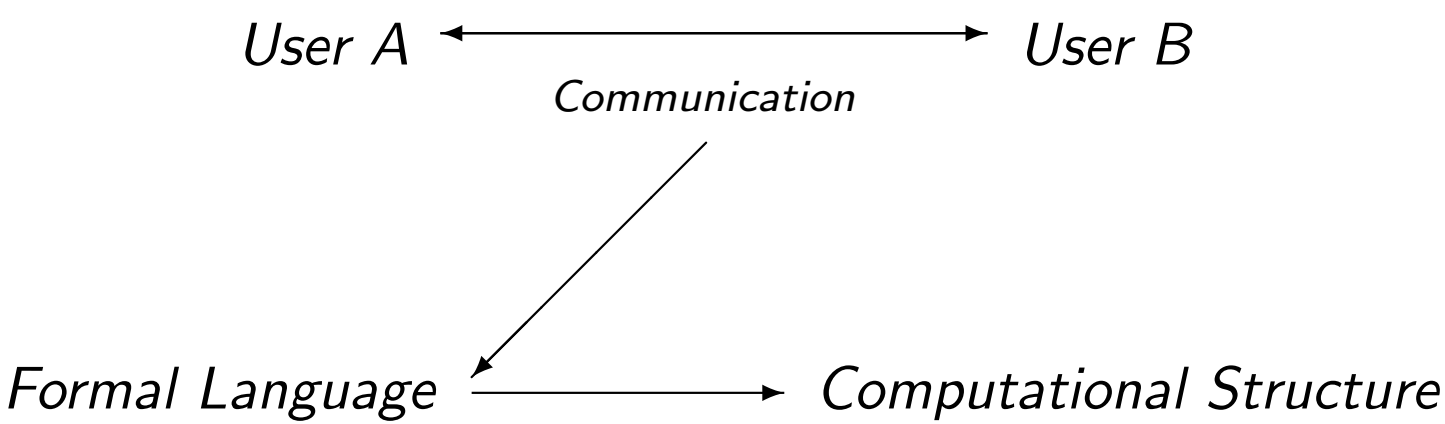
Our Approach

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski



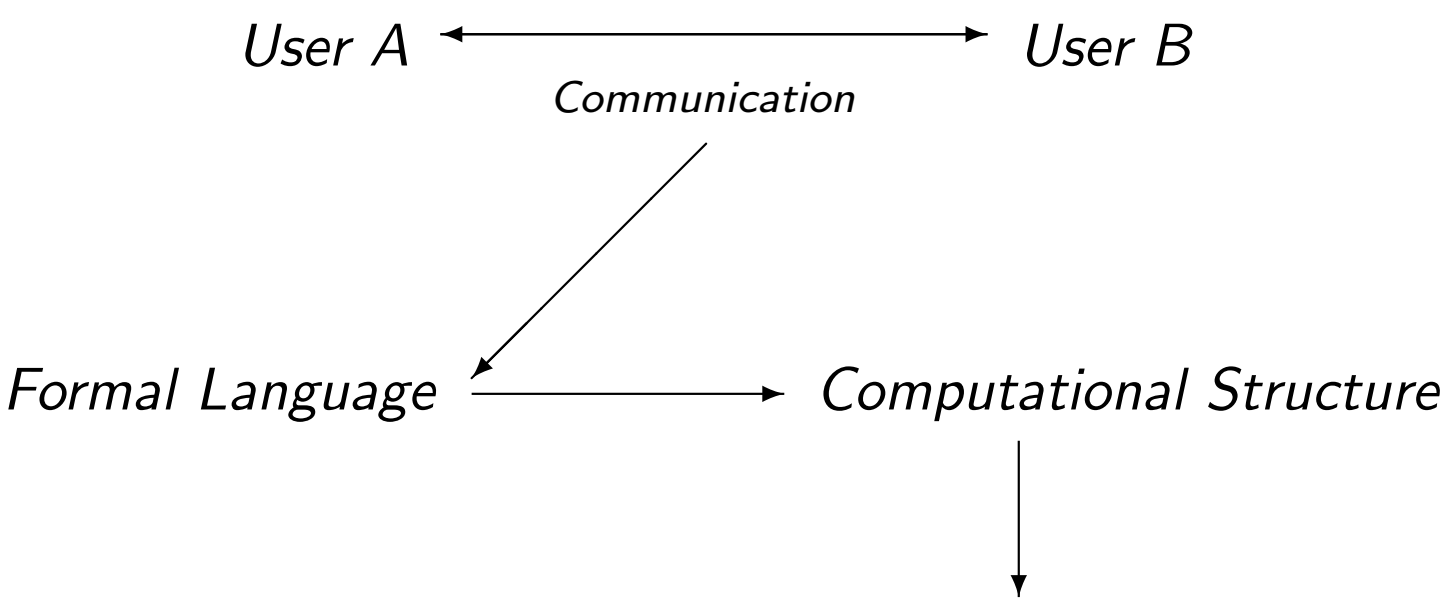
Our Approach

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski



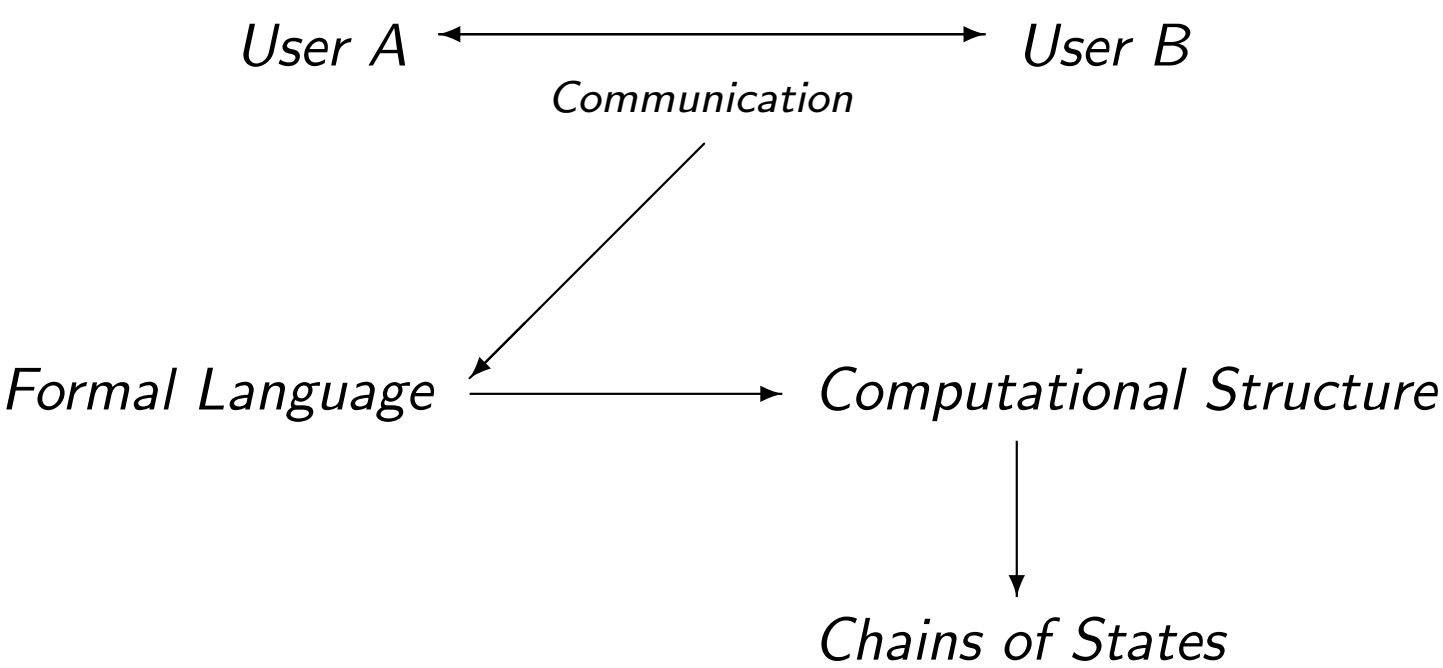
Our Approach

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski



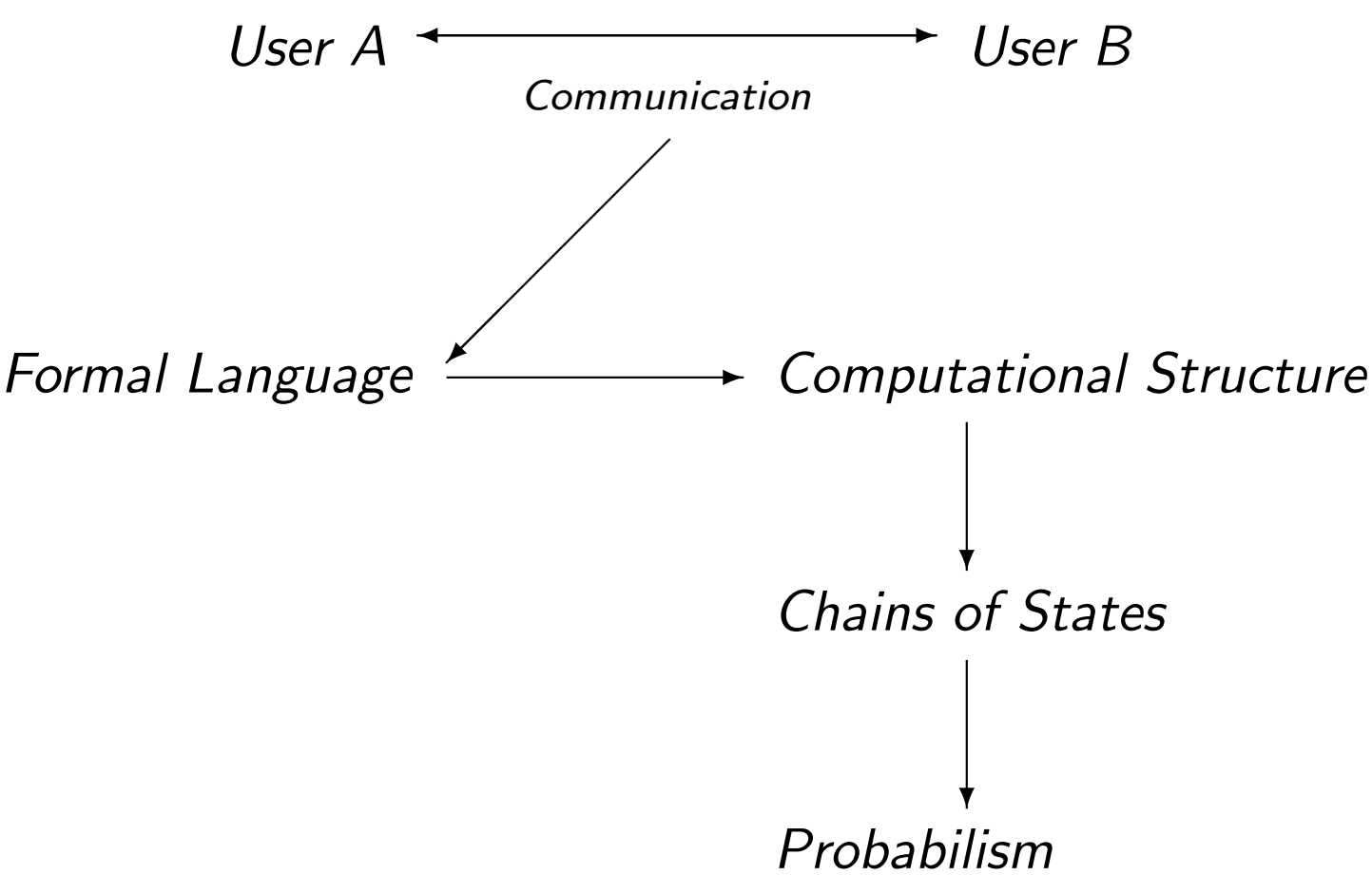
Our Approach

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski



Our Approach

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski



Example Needham-Schroeder Protocol and Lowe's attack

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

Needham-Schroeder Protocol:

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

Lowe's attack:

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

Chains of States - Types of States

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

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

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

Execution:

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

can be encoded by:

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

Example with the Intruder

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

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

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

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 $\exists_{t<i}(s_t = G_U^X \ \vee \ s_t = K_U^X)$.

New Challenges - Probabilistic Approach

- Large number of keys
- Is a perfect cryptography assumption today fully justified?
- "Tailor-made" security

Automata Model

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

Consider a **probabilistic automaton** of protocol runs: $\mathcal{A} =< \mathcal{Q}, \Sigma, \delta, q_0, F >$, where:

- $\mathcal{Q}$ is the finite set of states,
- Σ is the input alphabet (a power set of the set of all keys that Intruder needs to break in order to gain access to secret information),
- $\delta \subseteq \mathcal{Q} \times \Sigma \times < 0, 1 > \times \mathcal{Q}$ is the transition relation (with distinguished probability),
- q_0 is the initial state,
- $F \subseteq \mathcal{Q}$ is the set of finite (accepting) states.

Needham-Schroeder-Lowe Protocol

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

NSPKL:

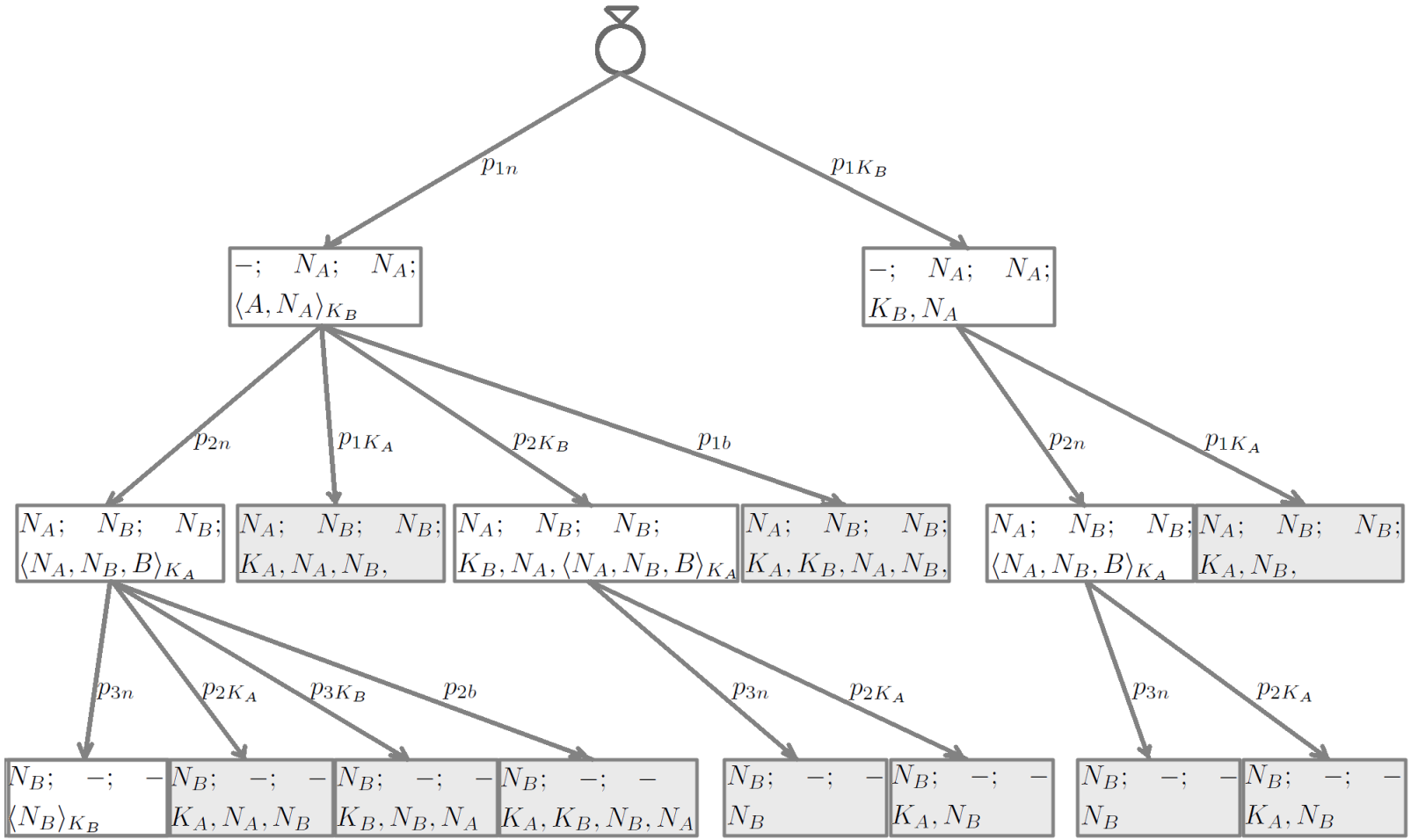
$$\begin{aligned} \alpha_1 \quad & A \rightarrow B : \langle N_A \cdot i(A) \rangle_{K_B}, \\ \alpha_2 \quad & B \rightarrow A : \langle N_A \cdot N_B \cdot i(B) \rangle_{K_A}, \\ \alpha_3 \quad & A \rightarrow B : \langle N_B \rangle_{K_B}. \end{aligned} \tag{3}$$

Let's assume that:

- α_1' - α_1 and breaking/gaining K_B^{-1} with probability p_{1K_B}
- α_2' - α_2 and breaking/gaining K_A^{-1} with probability p_{1K_A} or K_B^{-1} with probability p_{2K_B}, or both keys with probability p_{1b}
- α_3' - α_3 and breaking/gaining K_B^{-1} with probability p_{3K_B} and so on...

Probabilistic Model

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski



Automatic Tool

- On the input: protocol specification (in ProToc) communication parameters
- Parsing
- Executions building
- Chains of states creation
- Adding probabilistic analysis
- On the output: probability of breaking each key or combination of keys in each step

Experiments

Olga Siedlecka-Lamch, Miroslaw Kurkowski, Jacek Piatkowski

Protocol	Number of nodes	Computing time [s]
NSPKL	1112	0,011
NSPKL_server2	17004	0,55
NSPKL_server1	58442	2,53
NSPKL_non2	2691920	183

Summary

Shown method allows us to determine for the input protocol which keys are most important (breaking/gaining them guarantees the easiest interception of confidential information), and hence the correct choice of encryption strength or security. It also highlights the keys that are not so important and their cryptographic power can be reduced - thus relieving the server.

References I


Armando, A., et. al.: 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)


Burrows M., Abadi M., Needham R.: A Logic of Authentication, In: Proceedings of the Royal Society of London A, vol. 426, pp. 233–271, (1989)


Cremers, C.: The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols, In: Proceedings of the 20th International Conference on Computer Aided Verification, Princeton, USA, pp 414–418 (2008)


Dolev, D. and Yao, A.: On the security of public key protocols. In: IEEE Transactions on Information Theory, 29(2), pp. 198–207 (1983)


Hyla, T., El Fray, I., Kurkowski, M., Mackow, W., Pejas, J.: Practical Authentication Protocols for Protecting and Sharing Sensitive Information on Mobile Devices, In: Cryptography and Security Systems, vol. 448 of CCIS, pp. 153–165, Springer Verlag (2014)


Kurkowski, M., Grosser, A., Piatkowski, J., Szymoniak, S.: ProToc - an universal language for security protocols specification, In: Advances in Intelligent Systems and Computing, vol. 342, pp 237-248, Springer Verlag (2015)


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


Kurkowski, M., Siedlecka-Lamch, O., Dudek, P.: Using Backward Induction Techniques in (Timed) Security Protocols Verification, In: Proceedings of 12th International Conference CISIM 2013, Krakow, Poland, Lecture Notes in Computer Science vol. 8104, pp. 265–276 (2013)

References II


Lowe, G.: Breaking and Fixing the Needham-Schroeder Public-key Protocol Using fdr., In:TACAS, LNCS, Springer, pp. 147–166 (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.: Inductive Analysis of the Internet Protocol TLS, TR440, University of Cambridge, Computer Laboratory (1998)


Pnueli, A., Zuck, L.D.: Probabilistic Verification, Information and Computation, Vol. 103, Issue 1, pp. 1–29 (1993)


Siedlecka-Lamch O., Kurkowski M., Szymoniak S.,Piech H.: Parallel Bounded Model Checking of Security Protocols, In: Proc. of PPAM'13, vol. 8384 of LNCS, pp. 224–234, Springer Verlag (2014)