Prace Naukowe Akademii im. Jana Długosz w Częstochowie
Matematyka X, Częstochowa 2005

A State Transition Model for Honest Executions of Authentication Protocols*

Miroslaw Kurkowski

Institute of Mathematics and Computer Science,
Jan Długosz University of Częstochowa,
al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland
E-mail address: m.kurkowski@ajd.czest.pl

Abstract
Cryptographic protocols are very good tools to achieve authentication in large distributed computer network. These protocols are precisely defined sequences of action (communication and computation steps) which use some cryptographic mechanism such as encryption and decryption. It is well known that the design of authentication cryptographic protocols is error prone. Several protocols have been shown flawed in computer security literature. Due to this it is necessary to have some methods of analysis and properties verification of these protocols. In investigations of these properties a suitable formal model is needed. This model should express all important properties and ideas of protocols. In this paper we propose a new formal model of "honest" executions of cryptographic authentication protocol. We hope that this model is a good startpoint for further investigations and will be usefull in verification of real executions of cryptographic protocols.

Keywords: Cryptographic authentication protocols, verification, model checking.

*Extended version of a talk presented at the IX Conference "Applications of Algebra", Zakopane, March 7–13, 2005.

1 Introduction

Authentication is the process by which participants in a computer network prove their identity. Usually, principals share a special random number (secret) with some trusted machine, called an authentication server. By proving possession of this number, a principal can establish trust in its identity. Authentication in a large, distributed system is challenging because participants communicate over a network that is vulnerable to many Intruders attacks. A passive Intruder can be on line and obtain sensitive information. An active Intruder can obtain and modify messages and insert his own data to the net. Encryption can unable the attacks of an active intruder. Many encryption schemes preserve the integrity property, where any modification to some part of the data causes the decryption to fail. Thus, without knowledge of the key, an active, malicious Intruder's ability is limited to blocking data from reaching its destination. Such an Intruder can impersonate some participant in the computer network and intercept his rights and privileges.

The problem of looking for methods of correctness verification of the cryptographic authentication protocols is still important. In the last decade many methods and results are introduced and published. These methods allowed to discover many kinds of attacks upon the authentication protocols (see for example [1, 7–10]).

Catherine Meadows in [13] defines four approaches to the analysis of cryptographic protocols*:

[1] To model and verify the protocol using specification languages and verification tools not specifically developed for the analysis of cryptographic protocols.

[2] To develop expert systems that a protocol designer can use to develop and investigate different scenarios.

[3] To model the requirements of a protocol family using logics developed for the analysis of knowledge and belief.

*Another interesting paper with rather complete information on this topic is [5].

[4] To develop a formal model based on the algebraic term-rewriting properties of cryptographic systems.

The most effective method of verification of the authentication crytographic protocols is model checking of the specially defined partially ordered spaces which express executions of the protocols in the real network.

The applied by many researchers techniques, which uses model checking of the specially defined spaces which express executions of the protocols in the real net, enabled to find many attacks upon the protocols. The best examples are paper by G. Lowe [7–10], C. Meadows [12–16] and E. Clarkes group [3, 4, 11] and a few others [2, 17]. In these papers interesting definitions of computer network models were given.

Usually model checking techniques provide some problems in verification. Main of these problems is explosion of states of constructed spaces.

In this paper we present a new abstraction definition of crytographic authentication protocols and new definition of protocols executions. We define also a space of all executions of these protocols. Obviously this space is very huge. We show how introduce some relations which allows considering different executions without carrying another ones.*

2 Syntax

Due to the difference of cryptographic authentication protocols it is difficult to give the model of protocol executions space which has do with every kinds of these protocols.

We may also notice that attacks upon the protocols are due to the weakness of the structure of sending messages during the protocol executions. In the below consideration we assume that every user of the net (the Intruder too) may execute every actions. The only conditions

*This investigations are due to authors PhD dissertation [6].

are possesing a suitable cryptographic key to decrypting/encrypting messages and possesing secrets used in messages.

Below we give the basic syntactic definitions of our model. Let:

$\mathcal{Z}_{\mathbf{P}} = \{\mathcal{P}_{\mathbf{1}}, \mathcal{P}_{\mathbf{2}}, \mathcal{P}_{\mathbf{3}} \dots\}$ - be the countable set of symbols representing the users of the computer network.

$\mathcal{Z}_{\mathbf{A}} = \{\mathcal{A}_{\mathbf{1}}, \mathcal{A}_{\mathbf{2}}, \mathcal{A}_{\mathbf{3}}, \dots\}$ - be the countable set of symbols representing the symbols of an alphabet.

$\mathcal{Z}_{\mathbf{K}} = \{\mathcal{K}_{\mathbf{1}}, \mathcal{K}_{\mathbf{2}}, \mathcal{K}_{\mathbf{3}} \dots\}$ - be the countable set of symbols representing the cryptographic keys.

$\mathcal{Z}_{\mathbf{S}} = \{\mathcal{S}_{\mathbf{1}}, \mathcal{S}_{\mathbf{2}}, \mathcal{S}_{\mathbf{3}} \dots\}$ - be the countable set of symbols representing the confidential informations (ex. *nonces*).

Auxiliary symbols: , ( ) { }.

Definition of the set of *messages therms* $\mathcal{T}$. Let $\mathcal{T}$ be a smallest set that fulfils:

[1] $\mathcal{Z}_P \cup \mathcal{Z}_A \cup \mathcal{Z}_K \cup \mathcal{Z}_S \ \subseteq \mathcal{T}$.

[2] If $X \in \mathcal{T}$ and $Y \in \mathcal{T}$, then the sequence $(X, Y) \in \mathcal{T}$,

[3] If $X \in \mathcal{T}$ and $K \in \mathcal{Z}_K$, then $\{X\}_K \in \mathcal{T}$ ($\{X\}_K$ is the therm that is interpreted as the ciphertext containing the message X encrypted under the key K).

Definition. *The step of the pseudoprotocol* we call any kind of element of cartesian product $\mathcal{Z}_P \otimes \mathcal{Z}_P \times \mathcal{T}$ *.

We denote steps of pseudoprotocols by α, β etc.

Definition. By *the pseudoprotocol* we mean any finite sequence of steps $(\alpha_1, \alpha_2, \dots, \alpha_n)$.

We denote pseudoprotocols by Σ, Δ etc.

For simplify, we will accept the following notations:
If $\Sigma = (\alpha_1, \alpha_2, \dots, \alpha_n)$, then we denote all steps of the protocol by: α_i^{Σ}, dla $i = 1, 2 \dots, n$.

*$A \otimes A$ denote the cartesian product without pairs like (a, a).

3 Computational structure

We will now present the construction of the partial order structure which express the reality of sending messages while autherntication protocols are executed in the real computer network. Now we assume that all users behave properly. Let:

- $\mathbf{P} = \{P_1, P_2, \ldots, P_{k_P}\}$ be a set of cooperating with eachother users of the computer network,
- $\mathbf{A} = \{a_1, a_2, \ldots, a_{k_A}\}$ be a set of symbols of any finite alphabet,
- $\mathbf{K} = \{K_1, K_2, \ldots, K_{k_K}\}$ be a set of cryptographic keys (already existing or possible to be generated) of the users of the net,
- $\mathbf{S} = \{S_1, S_2, \ldots, S_{k_S}\}$ be a set of confidential messages (ex. *nonces*).

Definition. By *the set of messages* $\mathbf{L}$ we mean the smallest set satisfying follows:

[1] $\mathbf{P} \cup \mathbf{A} \cup \mathbf{K} \cup \mathbf{S} \subseteq \mathbf{L}$,

[2] If $X \in \mathbf{L}$ i $Y \in \mathbf{L}$, then a sequence $(X, Y) \in \mathbf{L}$,

[3] If $X \in \mathbf{L}$ i $K \in \mathbf{K}$, then $\{X\}_K \in \mathbf{L}$ ($\{X\}_K$ is the ciphertext containing the message X encrypted under the key K).

Lets consider all partial iniection f which map $f : dom(f) \to \mathbf{L}$, where $dom(f) \subseteq \mathcal{T}$, satisfying the following conditions:

[1] $f(\mathcal{Z}_P) \subseteq \mathbf{P}$,

[2] $f(\mathcal{Z}_A) \subseteq \mathbf{A}$,

[3] $f(\mathcal{Z}_K) \subseteq \mathbf{K}$,

[4] $f(\mathcal{Z}_S) \subseteq \mathbf{S}$,

[5] $\forall_{X,Y \in \mathcal{T}} \; (\; g(X, Y)) = g(X), g(Y) \;)$ (homomorphism);

[6] $\forall_{X \in \mathcal{T}} \; \forall_{K \in \mathcal{Z}_K} (\; g(\{X\}_K)) = \{g(X)\}_{g(K)})$.

Definition. We call function f by *partial interpretations* of the set $\mathcal{T}$.

We denote by $\mathcal{F}$ the set of all partial interpretations of the set $\mathcal{T}$.

Definition. By *the essential space* we mean a pair $(P, \mathcal{F})$.

Definition. If $\mathcal{P}, \mathcal{Q} \in \mathcal{Z}_P \cap dom(f)$ and $\mathcal{L} \in dom(f)$, then the triple $(f(\mathcal{P}), f(\mathcal{Q}), f(\mathcal{L}))$ we call *the step pseudoprotocol interpretation* $\alpha = (\mathcal{P}, \mathcal{Q}, \mathcal{L})$. If a step α is i - this step of the protocol Σ, then its interpretation we will denote by $\alpha(f, \Sigma, i)$.

Definition. By *the execution of the pseudoprotocol*

$$\Sigma = (\alpha_1^\Sigma, \alpha_2^\Sigma, \dots, \alpha_n^\Sigma)$$

with interpretation f we mean the sequence:

$$(\alpha(f, \Sigma, 1), \alpha(f, \Sigma, 2), \dots, \alpha(f, \Sigma, n)).$$

Because verification can be lead across for only one protocol, we will consider only one protocol Σ. This doesn't mean that there is a limit for further consideration.

Definition. By *the run* we mean any finite or not sequence of the protocol steps interpretations Σ:

$$\nabla = (\alpha_1(f_1, \Sigma, i_1), \alpha_2(f_2, \Sigma, i_2), \dots, \alpha_n(f_n, , \Sigma, i_n), \dots)$$

which fulfils the following conditions:

[1] $\forall_{n \in \mathbf{N}^+} \ [i_n > 1 \ \Rightarrow \ \exists_{k<n} \ (\ f_k = f_n \wedge \ i_k = i_n - 1)]$,

[2] $\forall_{i,j \in \mathbf{N}^+} (f_i \neq f_j \ \Rightarrow \ f_i(\mathbf{S}) \cap f_j(\mathbf{S}) = \emptyset)$.

Definition. By *the space of protocol runs* $\mathcal{R}$ we mean the set of all runs.

Definition. *The prefix* of the run ∇ is an arbitrary (ex. empty) its beginning sector.

Definition. By Ω we mean the set of all prefixes of runs from the set $\mathcal{R}$.

We now introduce a binary relation between elements of the set Ω. Let:

$$\nabla^1 = (\alpha_1^1(f_1^1, \Sigma, i_1^1), \alpha_2^1(f_2^1, \Sigma, i_2^1), \ldots, \alpha_n^1(f_n^1, \Sigma, i_n^1))$$

$$\nabla^2 = (\alpha_1^2(f_1^2, \Sigma, i_1^2), \alpha_2^2(f_2^2, \Sigma, i_2^2), \ldots, \alpha_m^2(f_m^2, \Sigma, i_m^2))$$

be arbitrary runs prefixes from Ω.

Definition. Let $\mathcal{R} \subseteq \Omega \times \Omega$ be a binary relation such that:

$$\nabla^1 \mathcal{R} \nabla^2 \equiv^{df} (f_n^1 = f_m^2 \wedge i_n^1 = i_m^2).$$

Lemma. The relation $\mathcal{R}$ is reflexive, symmetric and transitive, i.e. it is an equivalence relation.

Consider the set $\Omega_{\mathcal{R}}$ of all equivalence classes of the relation $\mathcal{R}$.

Observe that every equivalence class $[\nabla]_{\mathcal{R}}$ from $\Omega_{\mathcal{R}}$ is represented by a certain steps interpretation of the protocol $\alpha(f, \Sigma, i)$ which is a least element of any prefix from the set $[\nabla]_{\mathcal{R}}$.

To simplify our consideration we denote all equivalence classes $[\nabla]_{\mathcal{R}}$ by $[\alpha(f, \Sigma, i)]$ (where $\alpha(f, \Sigma, i)$ is the least element of every prefix belonging to the class $[\nabla]_{\mathcal{R}}$).

In such constructed the set $\Omega_{\mathcal{R}}$ we introduce the following binary re-
lation:

Let $[\alpha(f_{n_f}, \Sigma, i_{n_i})]$ and $[\alpha(f_{m_f}, \Sigma, i_{m_i})]$ be arbitrary elements from $\Omega_{\mathcal{R}}$.

Definition. Let $\preceq \subseteq \Omega_{\mathcal{R}} \times \Omega_{\mathcal{R}}$, be a binary relation such that:

$$[\alpha(f_{n_f}, \Sigma, i_{n_i})] \preceq [\alpha(f_{m_f}, \Sigma, i_{m_i})] \equiv^{df}$$

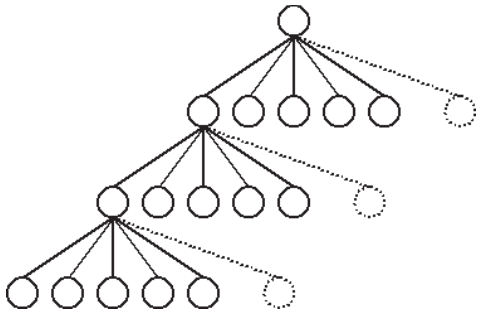
$$\equiv^{df} (f_{n_f} = f_{m_p} \wedge i_{n_i} \leq i_{m_i}).$$

Lemma. The relation $\preceq$ is reflexive, antysymmetric and transitive, i.e. it is a partial order relation.

The structure constructed above enables describing the process of sending information during the actual executions of authentications

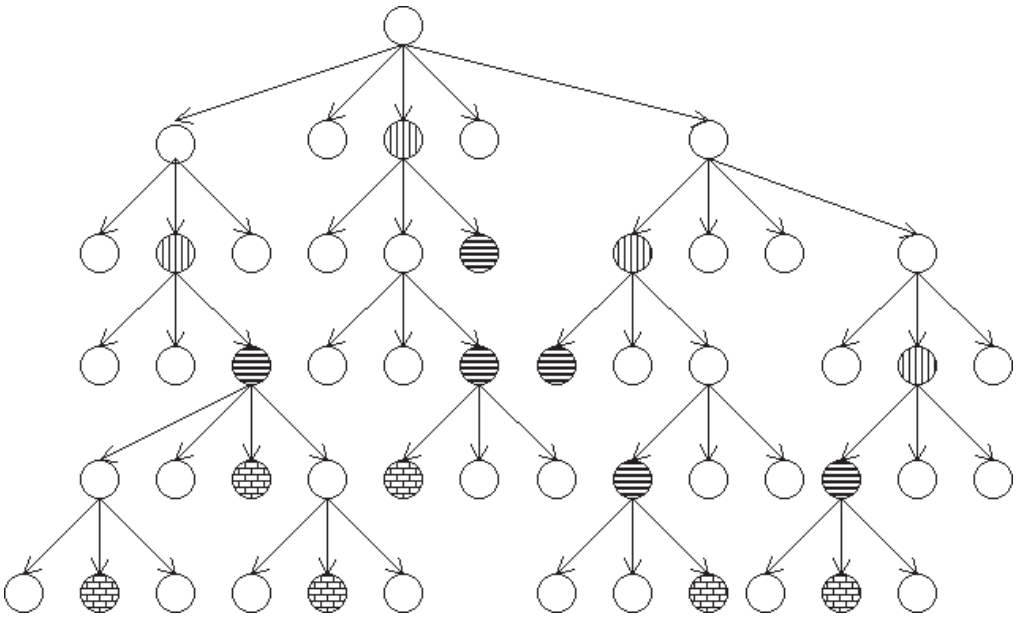
protocols. Now we present the advantages comming from the structure defined above.

The metods based on model checking often face the problem of states explosion in the space which describe a given problem. It is not hard to notice that the case of building space for authentication protocols is similar.

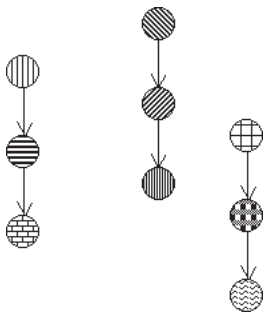


Lets consider the following example of a protocol, in which exange with three messages takes place.

We may identify the space of all possible runs with a tree of a common root, whose maximum paths are runs and the root is the empty prefix. During the automatic search of this space, we consider the same steps of a given execution. We show this type of situation in the drawing below.



After applicating the reduction above, the partial order structure (the space) of the executions of the protocols is presented as follows:



Each execution of a protocol may be considered separately. However, lets notice, that after applying the reduction individual steps in the structure are really equivalence classes containing all prefixes of the runs ending on a given step. Therefore, examining individual executions, we don't lose the general idea.

Recall that in the beginning of our reflections we have established that every user of the network is honest. Therefore, this construction does not enable examination of space of the execution of protocols with an Intruder taking part.

4 Conclusion and future work

In this paper we have presented a new definition of crytographic authentication protocols and new definition of protocols executions. We also defined a space of all executions of these protocols. To decrease number of states in our model we have proposed some partial order reduction.

Investigated structures allows expresing of authentication protocols executions without Intruder. Obviously in real executions particips which communicate to each other in a computer network are vulnerable to Intruders attacks. We hope that this model is a good startpoint for further investigations and will be usefull in verification of real executions of cryptographic protocols with some model of Intruder.

References

[1] M. Abadi. Explicit communication revisited: two new attacks on authentication protocols. *IEEE Transactions on Software Engineering*, **23**, 3, pp. 185–186, 1997.

[2] U. Carlsen. Generating formal cryptographic protocol specifications. *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, IEEE Computer Society Press, pp. 137–146, 1994.

[3] E. Clarke, S. Jha, W. Marrero. Partial order reductions for security protocols verification. *Proceedings of TACAS/ETAPS*, Springer-Verlag. pp. 503–518, 2000.

[4] E. Clarke, W. Marrero, S. Jha. A machine checkable logic of knowledge for specifying security properties of electronic commerce protocols, *Proc. IFIP Working Conference on Programming Concepts and Methods (PROCOMET)*, June 1998.

[5] S. Gritzalis, D. Spinellis, P. Georgiadis. Security protocols over open networks and distributed systems: formal methods for their analysis, design and verification, *ACM Computer Communications*, **22**, 8, pp. 695–707, 1999.

[6] M. Kurkowski. *Dedukcyjne metody weryfikacji poprawnosci protokolow uwierzytelniania, Deduction methods of verification of authentication protocols.* PhD dissertation, Institute of Computer Science of The Polish Academy of Sciences, Warsaw, Poland 2003.

[7] G. Lowe. An attack on the Needham-Schroeder public-key authentication protocol. *Information Processing Letters*, **56**, No 3, pp. 131–133, 1995.

[8] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. *Proceedings of TACAS*, Springer-Verlag, pp. 147–166, 1996.

[9] G. Lowe. *A Family of Attacks upon Authentication Protocols.* Technical report 1997/5, Department of Mathematics and Computer Science, University of Leicester.

[10] G. Lowe. Some new attacks upon security protocols. *Proceedings of the IEEE Computer Security Foundations Workshop IX*, IEEE Computer Society Press, 1996.

[11] W. Marrero, E. Clarke, S. Jha. *Model Checking for Security Protocols.* CMU Report No. CMU-CS-97-139, 1997.

[12] C. Meadows. Applying formal methods to the analysis of a key-management protocol. *Journal of Computer Security*, **. 1**, pp. 5–35, 1992.

[13] C. Meadows. Formal verification of cryptographic protocols: a survey. *Advances in Cryptology, Proceedings of ASIACRYPT '94*, Springer-Verlag, pp. 133–150, 1995.

[14] C. Meadows. Language generation and verification in the NRL protocol analyzer. *Proceedings of the 1996 IEEE Computer Security Foundation Workshop IX*, IEEE Computer Society Press, pp. 48-61, 1996.

[15] C. Meadows. The NRL protocol analyzer: an overview. *Journal of Logic Programming*, **26**, No. 2, pp. 113–131, 1996.

[16] C. Meadows. Using the NRL protocol analyzer to examine protocol suites. *Proceedings of the 1998 LICS Workshop on Formal Methods and Security Protocols*, 1998. http://www.cs.bell-labs.com/who/nch/fmsp/program.html

[17] V. Varadharajan. Verification of network security protocols. *Computers and Security. Elsevier Advanced Technology.*, **8**, pp. 693–708, 1989.