



---

**TECHNICAL UNIVERSITY**  
**OF CLUJ-NAPOCA**  
**AUTOMATION AND COMPUTER SCIENCE FACULTY**

**eng. Béla GENGE**

# **PHD THESIS**

**(abstract)**

**CONTRIBUTIONS TO THE COMPOSITION AND  
IMPLEMENTATION OF SECURITY PROTOCOLS**

**ADVISOR:**  
**Prof. dr. eng. Iosif IGNAT**

---

**2009**

---

# Table of Contents

<b>1. Introduction .....</b>	<b>5</b>
1.1 Security protocols .....	5
1.2 Research trends .....	9
1.3 Thesis objectives and contributions .....	12
1.4 Thesis structure .....	16
1.5 Conclusions.....	18
<b>2. Related work .....</b>	<b>20</b>
2.1 Introduction.....	20
2.2 Security protocol composition .....	20
2.3 Performance evaluation of security protocols .....	25
2.4 Security protocol implementation.....	28
2.5 Conclusions.....	32
<b>3. Composition of security protocols.....</b>	<b>34</b>
3.1 Introduction.....	34
3.2 Protocol models .....	37
3.2.1 Constructing protocol models .....	37
3.2.2 Constructing intruder protocol models .....	41
3.3 Composition of preconditions and effects .....	42
3.3.1 Composition of participant models .....	42
3.3.2 Composition of protocol models.....	44
3.4 Composition of terms.....	44
3.4.1 Canonical model .....	45
3.4.2 Intruder canonical model .....	47
3.4.3 Mapping functions .....	48
3.4.4 Independence of protocol models .....	50
3.4.5 Term composition of protocol models.....	52
3.5 Composition of protocol model sequences .....	52
3.6 Experimental results .....	55
3.6.1 Composition of Yahalom-Lowe and Kao-Chow protocols .....	55
3.6.2 Composition of Lowe-Needham-Schroeder and ISO9798 protocols .....	62
3.6.3 Composition of protocols from existing libraries .....	68
3.7 Conclusions.....	70
<b>4. Performance evaluation of security protocols .....</b>	<b>71</b>
4.1 Introduction.....	71
4.2 Extending the canonical model.....	72
4.3 Evaluating the performance of cryptographic algorithms .....	75
4.3.1 Evaluating the performance of symmetric algorithms.....	76
4.3.2 Evaluating the performance of hash algorithms .....	77
4.3.3 Evaluating the performance of asymmetrical algorithms .....	78
4.4 Modeling the performance of cryptographic algorithms .....	79
4.4.1 Mathematical model.....	79
4.4.2 Model validation .....	81
4.5 Experimental results .....	85
4.5.1 Comparative performance evaluation of CCITT X.509 v1 and v1c.....	85
4.5.2 Comparative performance evaluation of protocols from existing libraries .....	88
4.6 Conclusions.....	90
<b>5. Constructing specifications for implementing security protocols.....</b>	<b>91</b>

5.1 Introduction.....	91
5.2 Requirements .....	91
5.2.1 Requirements for modeling preconditions and effects .....	92
5.2.2 Requirements for modeling transmitted and received terms .....	92
5.2.3 Requirements for knowledge modeling .....	93
5.3 Choosing technologies for specifications .....	93
5.4 Sequential specification (S-SEC).....	94
5.4.1 Structure of S-SEC.....	95
5.4.2 S-SEC model.....	96
5.5 Semantic specification (S-SEM).....	97
5.5.1 Structure of S-SEM.....	97
5.5.2 S-SEM model.....	105
5.6 Generating SEC-SEM (S-SEC-SEM) specifications.....	107
5.6.1 Modeling preconditions, effects and message sequences .....	107
5.6.2 Modeling participant knowledge .....	109
5.6.3 Modeling connections between message sequences and knowledge.....	115
5.6.4 Algorithms for generating S-SEC-SEM .....	118
5.6.5 Maintaining security properties in S-SEC-SEM.....	124
5.7 Experimental results .....	126
5.7.1 Constructing S-SEC-SEM for the Lowe-BAN protocol.....	127
5.7.2 Executing the generated specifications .....	135
5.8 Conclusions.....	137
<b>6. Middleware for the composition and implementation of security protocols .....</b>	<b>139</b>
6.1 Introduction.....	139
6.2 Requirements .....	140
6.3 Service oriented architecture .....	140
6.3.1 Name service architecture.....	142
6.3.2 Specification service architecture .....	143
6.3.3 Authorization and composition service architecture.....	144
6.3.4 Resource service architecture.....	147
6.4 Software architecture .....	150
6.4.1 Communication and XML messaging layer .....	151
6.4.2 Security protocol layer.....	151
6.4.3 Service protocol layer .....	153
6.5 Accessing resources.....	157
6.6 Experimental results .....	158
6.6.1 Data transfer.....	160
6.6.2 Accessing simple services.....	161
6.6.3 Accessing composed services .....	163
6.6.4 Transfer of composed data.....	165
6.7 Conclusions.....	167
<b>7. Final conclusions.....</b>	<b>169</b>
<b>References.....</b>	<b>175</b>
Appendices.....	187

# 1. Introduction

Security protocols are „communication protocols dedicated to achieving security goals” (C.J.F. Cremers and S. Mauw) [CM05] such as confidentiality, integrity or availability. Achieving such security goals is made through the use of cryptography. The explosive development of today's Internet and the technological advances made it possible to implement and use security protocols in a wide range of applications such as sensor networks, electronic commerce or routing environments.

Designing new protocols is a challenging task if we look at the number of attacks that have been discovered over the years [Cla96] after the protocols have been published. However, in the last few years the use of protocol composition [Cho06, Cre06b] has been successfully applied to create new protocols based on existing [DDMR07, ACG+08] or predefined protocols [Cho06].

In order to implement the resulting composed protocols, the security community developed two main approaches: manual and automated. Manual implementation [DA99, OAS05a, Ylo06] makes use of the human operator to manually write the code corresponding to the new security protocols. As opposed to this, automated implementation [SPP01, MBJ+02, AM03, BLG+08] ensures the localization, processing, execution and message implementation of security protocol specifications without the need of human intervention.

This thesis proposes as its main goal to develop methods for automated composition and implementation of security protocols. To achieve the proposed goal, the thesis analyzes the following main aspects: modeling security protocols, verifying the independence of security protocols, performance evaluation, semantic specifications, service-oriented middleware.

The main contributions in this chapter were:

1. Presentation of the importance of the domain;
2. Presentation of the main forms of protocol specifications;
3. Indication of open research directions.

## 2. Related work

In this chapter I briefly present and analyze the main papers from the literature dedicated to the three main research directions on which the thesis is focused:

- Automated composition of security protocols;
- Performance evaluation of security protocols;
- Automated implementation of security protocols.

In the field of security protocol composition, there are two directions to be considered: composition of predefined protocols and composition of existing protocols. The first direction [Gut01, Gut02, GF02, Cho06] makes use of predefined protocols, called primitives, to construct more complex, composed protocols. The second direction [CR03, DDMP03, Cre06b, DDMR07, CDPW07, ACG+08] uses existing protocols and verifies if the implied protocols maintain their security properties.

As previously identified, the literature abounds in methods proposed to compose security protocols. However, the number of papers drops significantly when we look for automated composition methods. A recently published method [ACG+08] proposes manual composition of protocol messages and automated verification of the correctness of the resulted security protocol through the use of the Scyther [Cre06a] tool.

In order to ensure the composition of protocols with identical properties, we used the performance criteria to choose a protocol for the final sequence. In the field of performance evaluation, papers are divided into two categories: papers dealing with the evaluation through

direct measurements [Das00, VW01, HM02, Hir03, Gut03, SRW05, Zha05, CDW06] and papers dealing with the evaluation through parametric models [BBC+05, Kir05].

In the composition phase, performance evaluation through implementation and measurement is not possible. Applying the parametric models is also not possible because of the missing information related to the environment where protocols are executed.

In the field of protocol implementation we encounter a large number of papers both for manual implementation [DA99, Ylo06, OAS05a, OAS05b, WWWC06, OAS07a, OAS07b, IBM07] and automated implementation [SPP01, MBJ+02, AM03, DKF+03, KLM05, TH05, BLG+08].

In the Web services field, papers dealing with automated implementation of security protocols are not so abundant [DKF+03, KLM05, BLG+08] and provide a semi-automatic method, in the sense that they ensure automated protocol classification selection rather than an automated implementation.

Main own contributions in this chapter:

1. Identification of most important papers from the three main research directions;
2. Comparative analysis of the most important existing methods.

### 3. Composition of security protocols

In this chapter I present a new method for the sequential composition of security protocols. Because of the similarities between parallel and sequential composition, the proposed method can also be applied in the parallel composition process using a subset of the operations used in the sequential composition process. The proposed composition method involves composing in a first stage the protocol preconditions and effects followed by the composition of participant chains.

The composition uses a protocol model constructed from participant models. Protocol participants communicate by exchanging *terms* constructed from elements belonging to the following basic sets: P, denoting the set of role names; N, denoting the set of random numbers or *nonces* (i.e. "number once used"); K, denoting the set of cryptographic keys; C, denoting the set of certificates and M, denoting the set of user-defined message components.

To denote the encryption type used to create cryptographic terms, we define the following *function names*:

$$\begin{aligned} \text{FuncName} ::= & \text{sk} && (\text{symmetric encryption function}) \\ & | \text{pk} && (\text{asymmetric encryption function}) \\ & | h && (\text{hash function}) \\ & | \text{hmac} && (\text{hmac function}) \end{aligned}$$

The above-defined basic sets and function names are used in the definition of *terms*, where we also introduce constructors for pairing and encryption:

$$T ::= . | P | N | K | C | M | (T, T) | \{T\}_{\text{FuncName}(T)},$$

where the `.' symbol is used to denote an empty term.

To capture the sending and receiving of terms, the definition of nodes uses *signed terms*. The occurrence of a term with a positive sign denotes transmission, while the occurrence of a term with a negative sign denotes reception.

**Definition 1.** A participant model (M-PART) is a tuple  $\langle \text{prec}, \text{eff}, \text{type}, \text{gen}, \text{part}, \text{chain} \rangle$ , where  $\text{prec}, \text{eff} \in \text{PR\_CC}^*$  denote protocol precondition and effect sets,  $\text{type} \in \text{PR\_TIP}^*$  denotes the type of each component,  $\text{gen} \in T^*$  denotes generated terms,  $\text{part} \in R$  denotes the

protocol participant name and  $chain \in (\pm T)^*$  denotes a sequence of nodes. A protocol model (*M-PROT*) is a set of participant models.

In the composition process of two security protocols we first need to compose the preconditions and effects. In other words, we need to establish if the knowledge needed by protocol participants to run a given protocol, expressed through the form of precondition predicates, is available and if the set of precondition and effect predicates is not destructive.

In order to establish if the set of preconditions corresponding to a protocol can be satisfied based on the effects corresponding to another protocol we use the predicate  $PART\_PREC : T^* \times PR\_CC^* \times PR\_CC^*$ . For two participant models,  $\zeta_1 = \langle prec_1, eff_1, type_1, gen_1, part_1, chain_1 \rangle$ ,  $\zeta_2 = \langle prec_2, eff_2, type_2, gen_2, part_2, chain_2 \rangle$ , and a context model containing initial participant knowledge,  $ci$ , the  $PART\_PREC$  predicate is defined as follows:

$$PART\_PREC(ci, eff_1, prec_2) = \begin{cases} True, & \text{if } eff_1 \subseteq prec_2 \cup \{\cup\{CON\_TERM(t) \mid t \in ci\}\}, \\ False, & \text{otherwise.} \end{cases}$$

The second property that must be ensured for the composed protocol, is the non-destructivity of the confidentiality property. In order to establish if the preconditions and effects of two participant models are destructive we use the predicate  $PART\_NONDISTR : PR\_CC^* \times PR\_CC^* \times PR\_CC^*$  which holds only if all confidential terms from one participant model maintain their confidentiality property in the second participant model also. Thus, the predicate is defined as:

$$PART\_NONDISTR(eff_1, prec_2, eff_2) = \begin{cases} True, & \text{if for } \forall EF_1(t_1) \in eff_1 \wedge \forall PR_2(t_2) \in prec_2 \\ & EF_1 \neq CON\_CONF \vee \text{if } EF_1 = CON\_CONF \wedge t_1 = t_2 \text{ then} \\ & \exists EF_2(t_2) \in eff_2 : EF_2 = CON\_CONF, \\ False, & \text{otherwise.} \end{cases}$$

In the second phase, we compose protocol terms. This process uses a canonical model that focuses on terms that can be verified by protocol participants. For each term from the protocol model, the canonical model provides a corresponding syntactical representation through the use of *basic types*. These denote the terms that can be verified by protocol participants also including a representation for terms that can not be verified because of limited participant knowledge.

The verification process makes use of these types to decide if attacks can be constructed on each protocol model by using terms extracted from the other considered protocol models. In order to verify this I used an intruder model based on the Dolev-Yao [DY83,Cer01] model to capture the powers that can be used by an intruder. I proved that if certain conditions are met, the intruder can not use its powers to construct attacks on protocols based on messages extracted from other protocols.

Canonical terms are defined as follows:

$$\mathcal{T} ::= . \mid BasicType \mid (\mathcal{T}, \mathcal{T}) \mid \{\mathcal{T}\}_{FuncName(\mathcal{T})}$$

**Definition 2.** A canonical participant model (*M-PART-C*), is a pair  $\langle part, ml_{cc} \rangle$ , where  $part \in \mathcal{R}$  corresponds to the participant name and  $ml_{cc} \in (Classifier \times (\pm T)^*)^*$  is an LCC set. A canonical protocol model (*M-PROT-C*) is a collection of canonical participant models.

Based on the defined models, the composition of protocol terms involves verifying the *instance-independence* and *canonical-independence* properties, defined as follows.

**Definition 3.** Two participant models,  $\zeta_1$  and  $\zeta_2$ , are instance-independent if their properties are non-destructive, i.e. the predicates  $PART\_NONDISTR(\text{eff}_1, \text{prec}_2, \text{eff}_2)$  and  $PART\_NONDISTR(\text{eff}_2, \text{prec}_1, \text{eff}_1)$  hold. Two protocol models are instance-independent if all their participant models are instance-independent.

**Definition 4.** Let  $c\_txEnc : MPART-C \rightarrow \mathcal{T}^*$  be a function mapping the set of transmitted encryption terms for a given participant model and  $c\_rxEnc : MPART-C \rightarrow \mathcal{T}^*$  a function mapping the set of received terms for a given participant model. Then two canonical participant models  $\zeta_1, \zeta_2 \in MPART-C$  are canonical-independent if for all  $t_1 \in c\_txEnc(\zeta_1)$  and all  $t_2 \in c\_rxEnc(\zeta_2)$ , the predicate  $TCANACC(t_1, t_2)$  does not hold, where

$$TCANACC(t, t') = \begin{cases} \text{True}, & \text{if } t = t' \vee (t' = \mathbf{u} \wedge t \in \text{BasicType}) \vee \\ & (t = \mathbf{u} \wedge t' \in \text{BasicType}), \\ TCANACC(t_1, t'_1) \wedge & \text{if } (t = (t_1, t_2) \wedge t' = (t'_1, t'_2)) \vee \\ TCANACC(t_2, t'_2), & (t = \{t_1\}_{f(t_2)} \wedge t' = \{t'_1\}_{f(t'_2)} \wedge (t_2 = t'_2 \vee t'_2 = \mathbf{u})), \\ \text{False}, & \text{otherwise.} \end{cases}$$

In order to prove that if the instance-independence and canonical-independence conditions are met, the intruder can not construct attacks on the involved protocols, I used the following proposition, which was proved to hold:

**Proposition 1.** Let  $\xi_1, \xi_2 \in MPROT$  and  $\xi'_1, \xi'_2 \in MPROT-C$  their canonical representations, such that  $\xi'_1 = c\_mprot(\xi_1)$  and  $\xi'_2 = c\_mprot(\xi_2)$ . If  $\xi_1$  and  $\xi_2$  are instance-independent and  $\xi'_1$  and  $\xi'_2$  are canonical-independent, then  $\xi_1$  and  $\xi_2$  are independent.

**Table 1.** Results of the composition and validation process

Nr.	Protocol 1 (P1)	Protocol 2 (P2)	Comp. PE (P1-P2/ P2-P1)	Comp. T (P1-P2/ P2-P1)	Verif. Scyther (P1-P2/ P2-P1)
1.	Lowe-BAN	ISO9798	NO/YES	YES/YES	YES/YES
2.	Lowe-BAN	CCITTX.509 v1	NO/NO	YES/YES	YES/YES
3.	ISO9798	CCITTX.509 v1	YES/YES	YES/YES	YES/YES
4.	ISO9798	CCITTX.509 v1c	YES/YES	YES/YES	YES/YES
5.	CCITTX.509 v1	CCITTX.509 v1c	YES/YES	YES/YES	YES/YES
6.	BAN Concrete RPC	Lowe-BAN	YES/YES	NO/NO	NO/NO
7.	Lowe-Denning-Sacco	Kao-Chow v1	YES/YES	NO/NO	NO/NO
8.	Kao-Chow v1	Kao-Chow v2	YES/YES	YES/YES	YES/YES
9.	Lowe-Denning-Sacco	Kerberos v5	YES/YES	NO/NO	NO/NO
10.	Lowe-Kerberos v5	Neuman-Stubblebine	YES/YES	NO/NO	NO/NO
11.	Hwang-Neuman-Stubblebine	Needham-Schroeder	YES/YES	YES/YES	YES/YES
12.	Needham-Schroeder	CCITTX.509 v1	YES/NO	YES/YES	YES/YES
13.	Lowe-Needham-Schroeder	ISO9798	YES/NO	YES/YES	YES/YES
14.	Otway-Rees	Lowe-BAN	YES/NO	YES/YES	YES/YES
15.	SPLICE/AS	Needham-Schroeder	YES/YES	YES/YES	YES/YES
16.	TMN	Andrew RPC	YES/NO	YES/YES	YES/YES
17.	Yahalom-Lowe	Kao-Chow v1	YES/YES	NO/NO	NO/NO

For the validation of the proposed composition method, we composed 17 protocol pairs from the SPOR [LSV08] and John Clark [CJ97] libraries. The validation of the protocol independence proposition was made using the state-of-the-art security protocol verification tool

Scyther [Cre06a], by verifying the correctness of the composed protocols. The composed protocols and the result of the composition process, are given in Table 1.

Main own contributions in this chapter:

1. A new protocol model that includes explicit definition of protocol preconditions, effects and explicit term construction;
2. A new canonical protocol model that emphasizes the terms that can be verified by protocol participants based on their local knowledge;
3. A new protocol independence verification method;
4. A new syntactical composition method, that eliminates the need for human intervention;
5. Composition of 17 pairs of security protocols.

## 4. Performance evaluation of security protocols

In this chapter I present a new performance evaluation method used in the composition process of protocols with equal security properties. By using this method, the most performant protocol can be chosen in the final composed protocol sequence.

As opposed to existing methods [BBC+05, Kir05], where the developed performance evaluation models include parameters for the running physical environment, in this thesis, I propose a comparative performance evaluation. The proposed method enriches the canonical model introduced in the previous section with classifiers denoting the construction and processing operations. A special attention is given to the performance evaluation of cryptographic algorithms, because of their role in the performance of security protocols [Gut03, CDW06].

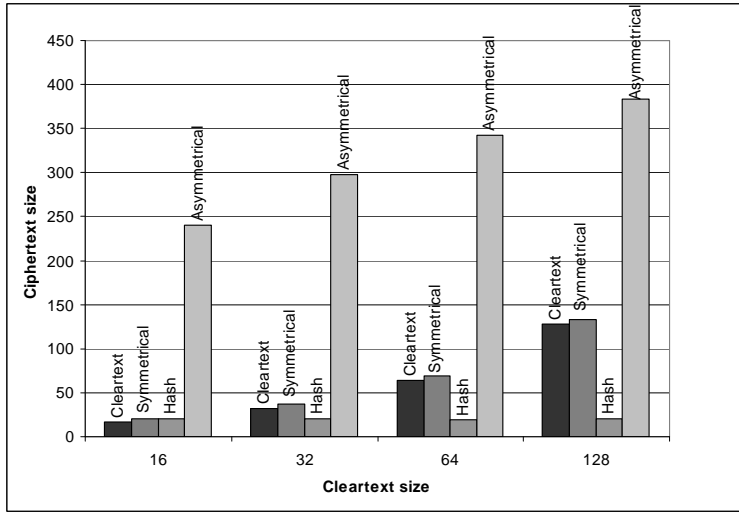
The cost of each operation is mapped through the use of the following functions:

$$f_{sk-c}, f_{sk-d}, f_{pk-c}, f_{pk-d}, f_{pk-s}, f_{pk-vs}, f_h, f_{hm}, f_{kg}, f_{ng}, f_c, f_i : \mathbb{R}^+ \rightarrow \mathbb{R}^+,$$

denoting the cost of symmetric encryption, symmetric decryption, asymmetric encryption, asymmetric decryption, digital signature, digital signature verification, hash, keyed hash, key generation, random number generation, concatenation and split, respectively.

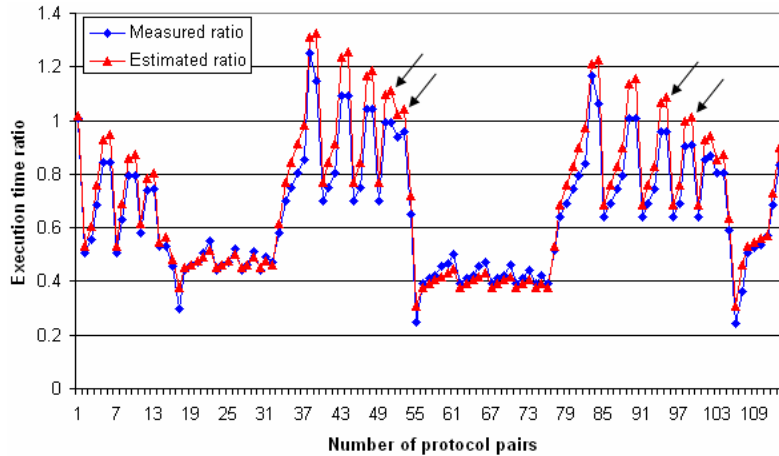
Term size plays an important role in the performance of security protocols. Because of this, we define several functions to map the size of terms to symbolic values. The size of each basic typed term is mapped to a symbolic value using the  $|\_| : BasicType \rightarrow \mathbb{R}^+$  operator. By applying cryptographic operations on terms, the size of the result depends on the type of the algorithm that is used [Kir05] (i.e. symmetrical, asymmetrical or hash). In order to model the resulting size, we conducted an exhaustive measurement of the ciphertext size resulted by applying cryptographic algorithms on cleartext of various size. The implementations were chosen from three well-known cryptographic libraries: Cryptlib [Gut08], OpenSSL [OSS08] and Crypto++ [Cry08].

From the results shown in Figure 1 we can clearly state that the size of ciphertext resulting from hash operations is the same for any cleartext while the size of ciphertext resulting from symmetrical cryptographic operations follows the size of cleartext. In contrast, the size of ciphertext resulting from asymmetrical operations is much greater than the original cleartext and strongly depends on the length of the key used in the process. We introduce the  $\lambda_s, \lambda_A, \lambda_{SM}, \lambda_H, \lambda_{SM-H} : \mathcal{T} \rightarrow \mathbb{R}^+$  functions to map the size of ciphertext resulted by applying symmetrical, asymmetrical, signature, hash, and and signature with hash operations respectively, on t-terms. We also introduce the  $\Delta : \mathcal{T} \rightarrow \mathbb{R}^+$  function to map the size of concatenated canonical terms.



**Figure 1.** Cleartext and ciphertext size for symmetrical, hash and asymmetrical algorithms

Next, I exhaustively evaluated the performance of cryptographic algorithms with all possible key sizes and encryption modes from three cryptographic libraries: Cryptlib, OpenSSL and Crypto++. Based on the measurement, I constructed a polynomial model that approximates the performance of encryption and decryption algorithms:  $f(x) = \alpha_4 x^3 + \alpha_3 x^2 + \alpha_2 x + \alpha_1$ . For each class of algorithm, I determined  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  using the least squares fitting method.



**Figure 2.** Graphical representation of the execution time ratio

In order to validate the constructed model, I first generated 1000 security protocols, and comparatively evaluated the performance of each protocol pair. From figure 2, we can observe that we have estimation errors near 1, because of protocols with similar performances. The proposed method has also been applied on 18 pairs of existing protocols taken from the SPORE [LSV08] and John Clark's [CJ97] libraries. From the total number of 306 combinations, I measured a total of 20 estimation errors, i.e. 6.53%.

Main own contributions in this chapter:

1. Explicit modeling of message construction and processing operations;
2. Development of performance evaluation criteria;
3. Exhaustive performance evaluation of all supported cryptographic algorithms, key sizes and encryption modes;
4. Construction of a new polynomial model corresponding to the performance of cryptographic algorithm classes;

- Comparative performance evaluation of 18 security protocols from the SPORE and John Clark's security library.

## 5. Constructing specifications for implementing security protocols

In this chapter I propose a new specification model for automated implementation of security protocols. The novelty of the proposed specification model lies in the explicit specification of construction and processing operations, based on which participants can automatically execute them.

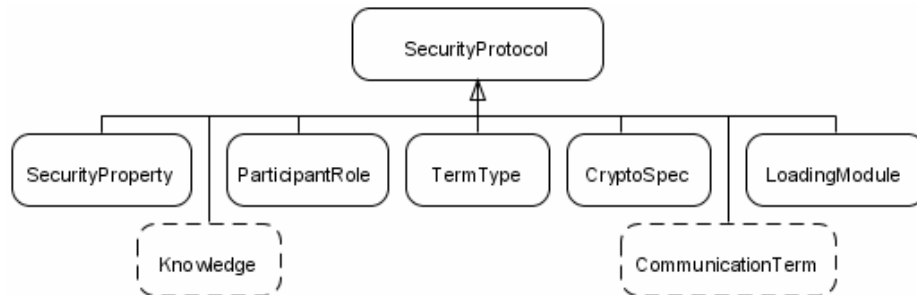
The specifications are constructed using WSDL-S [AFM+05, WWWC05] and OWL [WWWC04] languages. By using these, the specifications can be easily integrated in Web service specific environments. The WSDL-S component is modeled as a sequential specification (S-SEC), while the OWL component is modeled as a semantical specification (S-SEM). The role of the first component is to identify protocol preconditions, effects and message sequences. The role of the second component is to describe the construction and processing operations of protocol messages. The two components are defined as follows.

**Definition 5.** A sequential model is a tuple  $\langle PREC, EFECT, MSGSEQ, tiptr \rangle$ , where  $PREC \in ADNOT^*$  is an ordered sequence of adnotations, denoting protocol preconditions,  $EFECT \in ADNOT^*$  is an ordered sequence of adnotations, denoting protocol effects,  $MSGSEQ \in MSG^*$  is an ordered sequence of messages, and  $tiptr \in TIPTR$  is the transport type used to send and receive messages.

**Definition 6.** A semantic model is a triplet  $\langle CONC, PROPR, INST \rangle$ , where  $CONC \in CONC^*$  is a sequence of concepts,  $PROPR \in PROPR^*$  is a sequence of properties and  $INST \in INST^*$  is a sequence of instances.

Specifications are constructed from a core ontology, shown in figure 3. For each protocol, the concepts illustrated with interrupted lines, are extended with concepts and properties. The remaining concepts denote sub-ontologies with concepts used to model security properties, participant roles, term types, cryptographic properties, and modules.

Because of the complexity of these specifications, they must be carefully generated. We constructed several generating rules and algorithms for transforming regular specifications into semantic ones. The correctness of the constructed specifications follows from the one-to-one mapping and from the information provided by the user, considered to be correct.



**Figure 3.** Core ontology for constructing security protocol specifications

In order to validate the constructed specifications, we generated 10 specifications for protocols ranging from Lowe-BAN, ISO9798, Kerberos symmetric key to CCITT X.509. These cover a wide range of protocols that use asymmetric and symmetric key-based cryptography. The

execution timings of these specifications are given in table 2. These results have shown that the specifications contain sufficient information to allow an automated execution of security protocols. They also provide a proof that protocols can be executed in real time, and can be used in a wide variety of systems that require the use of security protocols.

**Table 2.** Part of the execution timings of the generated specifications

Protocol participant	Spec. proc. (ms)	Msg. constr. (ms)	Msg. proc. (ms)	Total (ms)
BAN Init.	14.58	11.81	3.68	30.08
BAN Resp.	14.03	2.86	1.62	18.52
ISO9798 Init.	13.07	35.784	23.30	72.16
ISO9798 Resp.	13.51	6.876	12.24	32.63
Kerb. Init. 1	22.63	0.83	0	23.47
Kerb. Init. 2	12.61	0.55	1.58	14.76
Kerb. Init. 3	2.23	3.34	0.94	6.52
Kerb. Resp. 1	19.28	0	0.41	19.69
Kerb. Resp. 2	10.81	3.379	1.67	15.87
Kerb. Resp. 3	5.25	11.41	3.59	20.26

Main own contributions in this chapter:

1. Identification of the requirements for constructing a new semantic specifications that can be easily integrated in Web services;
2. A new formal model of WSDL-S and OWL;
3. A new security protocol description ontology, that can be extended with concepts and properties in order to generate protocol specifications;
4. A set of rules and algorithms for generating security protocol specifications;
5. 10 specifications for two and three party security protocols.

## 6. Middleware for the composition and implementation of security protocols

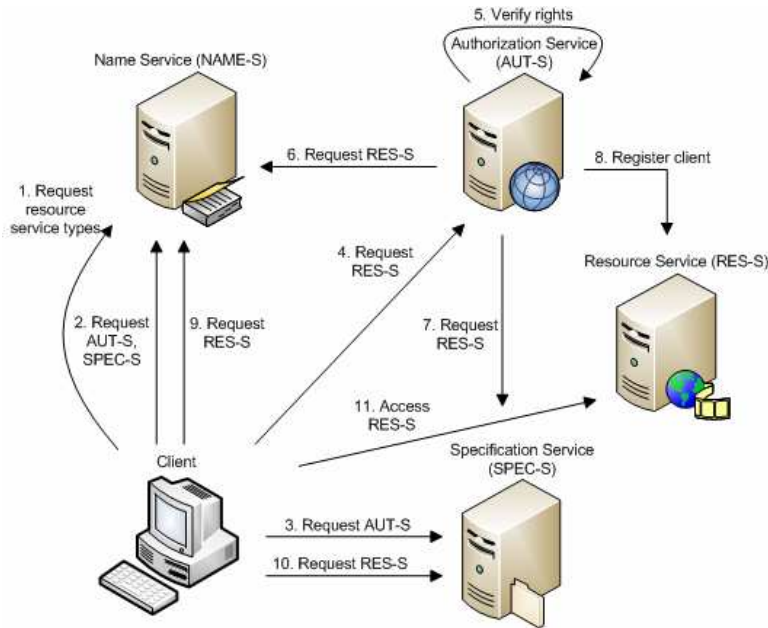
In this chapter I design a new middleware for the automated composition and implementation of security protocols. The proposed middleware defines two software layers: communication layer and service layer. The communication layer provides the implementation of message transfer and the implementation of operations for executing security protocol specifications. The service layer provides the implementation of service capabilities.

The proposed middleware uses four types of services: authorization and composition service, name service, specification service and resource service. In order for client applications to access these services, they must follow several steps, from downloading specifications, to composing the service and receiving the access token. These steps are shown in figure 4.

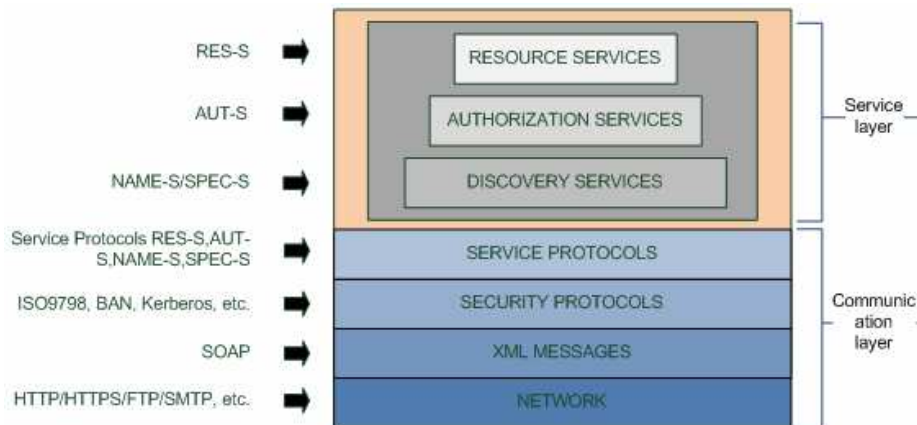
The software stack is constructed using the existing SOAP XML-based messaging layer from Web services. On top of this layer I constructed a security protocol layer that automatically executes the provided specifications. Messages exchanged by users are constructed by another layer, denoted by service protocols.

In order to evaluate the performance of the proposed middleware, I developed a new video surveillance system. The services have been implemented using the NSPR [Moz08a] (i.e.

Netscape Portable Runtime) API from the Mozilla platform. The implementation of the cryptographic operations was made using OpenSSL [OSSLO8].



**Figure 4.** Steps required for accessing resources

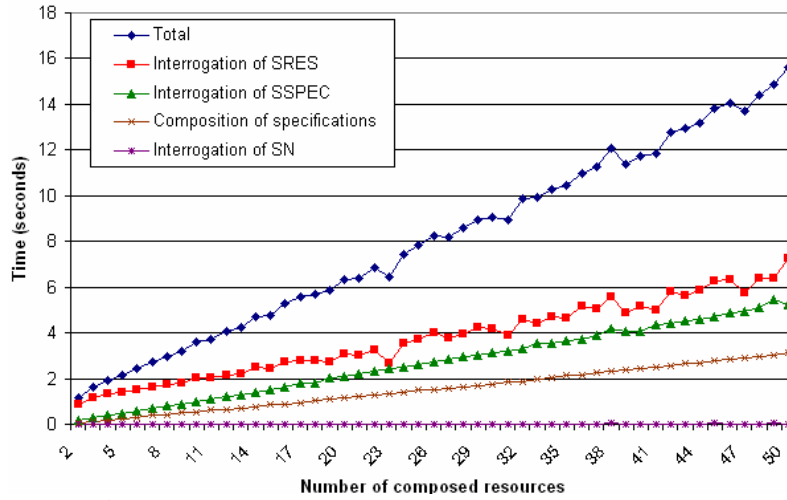


**Figure 5.** Software stack

The implemented resources provide capturing of video frames from physical cameras, the saving of video frames and the replay of saved frames. In figure 6, we can see the time needed for accessing composed resources. Most of the time is occupied by composing the resource services, followed by the interrogation of the specification service. The composition process has a linear evolution corresponding to the number of resources composed. Because of the small messages sizes, the fastest operation is the interrogation of the name service.

Main own contributions in this chapter:

1. Identification of the requirements for developing a new middleware;
2. Design of a new service-oriented architecture and of a software-oriented architecture based on Web services;
3. Development of several extension for the existing WS-Security standard for implementing new composed security protocols;
4. Implementation of a new distributed surveillance system, where security protocols are composed and executed automatically.



**Figure 6.** Accessing time for composed resources

## 7. Final conclusions

The main goal of this thesis was to develop and implement methods for the automated composition and implementation of security protocols. In order to achieve the proposed goal, I unified different research directions proposed in this thesis: security protocol composition, performance evaluation, security protocol implementation. The theoretical results from each direction have been validated with experimental results in each chapter. The proposed middleware from the last chapter ensures the unification of the developed methods and illustrates their applicability, through the development and implementation of a video surveillance system.

The most important own contributions in this thesis are:

1. A new security protocol model that allows the automated composition of security protocols, containing explicit preconditions, effects, term construction and type definitions;
2. Composition of 17 protocol pairs from the SPORE and John Clark's security protocol library;
3. A new comparative performance evaluation method of security protocols, used in the process of composition of protocols with identical properties;
4. Development of a new specification model, based on WSDL-S and OWL;
5. A new security protocol core ontology that models cryptographic properties, term types, modules, security properties, participant knowledge and message sequences;
6. Construction of the specifications for 10 representative security protocols, that use symmetric, asymmetric and hash-based cryptography;
7. Design and implementation of a middleware for the automated composition and execution of services that use security protocols;
8. Implementation of a new video surveillance system based on the developed middleware.

Possible future research could be focused on the following directions: composition of protocols with partially different security properties, development of tools for generating security protocol specifications, based on the proposed models, integration of existing service capability composition methods with the composition of security protocols proposed in this thesis.

## Selected references

- [ACG+08] S. Andova, Cas J.F. Cremers, K. Gjosteen, S. Mauw, S. Mjolsnes, and S. Radomirovic, A framework for compositional verification of security protocols, *Information and Computation, Special issue on Computer Security: Foundations and Automated Reasoning, Volume 206, Issues 2-4*, pages 425-459, Elsevier, 2008.
- [AFM+05] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M. Schmidt, A. Sheth, K. Verma, *Web Service Semantics - WSDL-S*, A joint UGA-IBM Technical Note, version 1.0, April 18, 2005.
- [AM03] I. Abdullah and D. Menascé, Protocol specification and automatic implementation using xml and cbse, *IASTED conference on Communications, Internet and Information Technology*, November 2003.
- [BBC+05] Chiara Bodei, Mikael Buchholtz, Michele Curti, Pierpaolo Degano, Flemming Nielson, Hanne Riis Nielson, Corrado Priami, *On Evaluating the Performance of Security Protocols*, Lecture Notes in Computer Science, Springer, Berlin, 2005.
- [BLG+08] C. Blanco, J. Lasheras, R. Valencia-Garcia, E. Fernandez-Medina, A. Toval, and M. Piattini, A systematic review and comparison of security ontologies, *Proc. of the Third International Conference on Availability, Reliability and Security*, pages 813-820, 2008.
- [CDW06] Cristian Coarfa, Peter Druschel and Dan S. Wallach, Performance Analysis of TLS Web Servers, *ACM Transactions on Computer Systems*, 24 (1), pages 39-69, 2006.
- [Cer01] Ilario Cervantes, *The Dolev-Yao Intruder is the Most Powerful Attacker*, 16th Annual Symposium on Logic in Computer Science, LICS'01, IEEE Computer Society Press, Boston, MA, 2001.
- [CDPW07] Ran Canetti, Yevgeniy Dodis, Rafael Pass, Shabsi Walfish, *Universally Composable Security with Global Setup*, Theory of Cryptography Conference (TCC), February 2007.
- [Cho06] Hyun-Jin Choi, *Security protocol design by composition*, Cambridge University, UK, Technical report Nr. 657, ISSN 1476-2986, 2006.
- [CJ97] John Clark and Jeremy Jacob, *A Survey of Authentication Protocol Literature: Version 1.0*, York University, 17 November 1997.
- [Cla96] John Clark, *Attacking Authentication Protocols*, [www.cs.york.ac.uk/~jac/papers/newHISJ.ps](http://www.cs.york.ac.uk/~jac/papers/newHISJ.ps), March 1996.
- [CM05a] C. Cremers, S. Mauw, Checking secrecy by means of partial order reduction, In S. Leue and T. Systs, editors, Germany, september 7-12, 2003, revised selected papers LNCS, Springer, Vol. 3466, 2005.
- [CR03] Ran Canetti, Tal Rabin, *Universal Composition with Joint State*, In *Proceedings of CRYPTO 2003*, Lecture Notes in Computer Science, vol. 2729. Springer Verlag, New York, pages 265-281, 2003.
- [Cre06a] Cas Cremers, *Scyther - Semantics and Verification of Security Protocols*, Thesis, University Press Eindhoven, 2006.
- [Cre06b] Cas J. F. Cremers, *Compositionality of Security Protocols: A Research Agenda*, *Electr. Notes Theor. Comput. Sci.*, 142, pages 99-110, 2006.
- [Cri01] Valentin Cristea, *A Collaborative Environment for High Performance Computing*, *IWCC 2001*, pages 47-59, 2001.
- [Cry08] *Crypto++ Software Distribution, Version 5.5.2*, <http://www.cryptopages.com/>, 2008.
- [DA99] Dierks T., Allen C., *The TLS Protocol, Version 1.0, Request for Comments: 2246*, Network Working Group, January 1999.
- [Das00] Neil Daswani, *Cryptographic Execution Time for WTLS Handshakes on Palm OS Devices*, Certicom Public Key Solutions, September 2000.
- [DDMP03] Anupam Datta, Ante Derek, John C. Mitchell, Dusko Pavlovic, *Secure Protocol Composition*, *Proceedings of the 2003 ACM workshop on Formal methods in security engineering*, pages 11-23, 2003.
- [DDMR07] A. Datta, A. Derek, J. C. Mitchell, A. Roy, *Protocol Composition Logic (PCL)*, *Electronic Notes in Theoretical Computer Science*, Vol. 172, pages 311-358, 2007.
- [DKF+03] Grit Denker, Lalana Kagal, Tim Finin, Massimo Paolucci and Katia Sycara, *Security for DAML Web Services: Annotation and Matchmaking*, LNCS 2870, pages 335-350, 2003.
- [DY83] D. Dolev and A.C. Yao. *On the security of public key protocols*. *IEEE Transactions on Information Theory*, IT-29(2), pages 198-208, 1983.
- [GF02] Joshua D. Guttman, F. Javier Thayer Fabrega, *Authentication tests and the structure of bundles*, *Theoretical Computer Science*, Vol. 283, No. 2, pages 333-380, June 2002.
- [GH07b] Genge Bela, Haller Piroaska, *Extending the Strand Space Model for Security Protocol Composition*, *International Scientific Conference "Interdisciplinarity in Engineering"*, Inter-Ing 2007, Târgu Mures, Romania, pages 1-7, November 2007.
- [GH08a] Genge Bela, Haller Piroaska, *A Modeling Framework for Generating Security Protocol Specifications*, *10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'08)*, Workshop on Global Computing Models and Technologies, Timisoara, Romania, IEEE Computer Society Press, pages 362-365, 2008.
- [GH09a] Genge Bela, Haller Piroaska, *Towards Automated Secure Web Service Execution*, *Networking 2009*, Aachen, Germany, May 11-15, *Lecture Notes in Computer Science (LNCS 5550)*, Springer-Verlag, pp. 943-954, 2009.
- [GH09b] Genge Bela, Haller Piroaska, *Middleware for Automated Implementation of Security Protocols*, *6th European Semantic Web Conference*, Heraklion, Greece, *Lecture Notes in Computer Science (LNCS 5554)*, Springer-Verlag, pp. 476-490, 2009.
- [GHO08] Genge Bela, Haller Piroaska, Ovidiu Ratoi, *Constructing Security Protocol Specifications for Web Services*, C. Badica et al. (Eds.), *Intelligent Distributed Computing*, Italy, Appears in *Studies In Computational Intelligence*, Springer-Verlag Berlin Heidelberg, pages 245-250, Sept. 2008.
- [GHOI08] Genge Bela, Haller Piroaska, Ovidiu Ratoi, Iosif Ignat, *Term-based composition of security protocols*, 2008 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, pages 233-238, May 2008.

- [GHIO08] Genge Bela, Haller Piroska, Iosif Ignat, Ovidiu Ratoi, Informal specification-based performance evaluation of security protocols, 4th IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, pages 193-200, August 2008.
- [GI06] Genge Bela, Iosif Ignat, A typed specification for security protocols, In the Proceedings of the 5th WSEAS Int. Conf. on Data Networks, Communications & Computers, Bucharest, Romania, pages 113-118, October 2006.
- [GI07a] Genge Bela, Iosif Ignat, An Abstract Model for Security Protocol Analysis, WSEAS Transactions on Computers, Issue 2, Volume 6, pages 207-215, February 2007.
- [GI07b] Genge Bela, Iosif Ignat, Verifying the Independence of Security Protocols, 3rd IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, pages 155-163, September 2007.
- [GI08] Genge Bela, Iosif Ignat, Syntactic Sequential Composition of Security Protocols, Journal of Automation Computers Applied Mathematics (ACAM), Volume 17, No. 2, pages 169-178, 2008.
- [GM05] Gorgan D., Melenti C., Parallel and distributed graphical processing on Grid structure of geographic and environment data, Ed Mediamira, 2005.
- [Gut01] J. D. Guttman, Key compromise and the authentication tests, Electronic Notes in Theoretical Computer Science, 2001.
- [Gut02] Joshua D. Guttman, Security protocol design via authentication tests, In Proceedings of the 15th IEEE Computer Security Foundations Workshop, IEEE CS Press, June, 2002.
- [Gut03] Peter Gutmann, Performance Characteristics of Application-level Security Protocols, 2003, available at [http://www.cs.auckland.ac.nz/~pgut001/pubs/app\\_sec.pdf](http://www.cs.auckland.ac.nz/~pgut001/pubs/app_sec.pdf).
- [Gut08] Peter Gutmann, Cryptlib Encryption Toolkit, <http://www.cs.auckland.ac.nz/~pgut001/cryptlib/index.html>, 2008.
- [Hir03] S. Hirani, Energy consumption of encryption schemes in wireless devices, Master's thesis, Telecommunications Program, University of Pittsburgh, Pittsburgh, Pennsylvania, 2003.
- [HM02] Alan Harbitter, Daniel A. Menasce, A methodology for measuring the performance of Authentication Protocols, ACM Transactions on Information and System Security, Vol. 5, No. 4, pages 458-491, November 2002.
- [IBM07] IBM Corporation, Web Services Federation Language (WS-Federation), available at <http://www.ibm.com/developerworks/library/specification/ws-fed/>, May 2007.
- [Kir05] Phongsak Kiratiwintakorn, Energy efficient security framework for wireless Local Area Networks, PhD Thesis, University of Pittsburgh, 2005.
- [KLM05] Anya Kim, Jim Luo, and Myong Kang, Security Ontology for Annotating Resources, R. Meersman and Z. Tari (Eds.): CoopIS/DOA/ODBASE 2005, LNCS 3761, Springer-Verlag Berlin Heidelberg, pages 1483-1499, 2005.
- [LSV08] Laboratoire Specification et Verification, Security Protocol Open Repository (2008), <http://www.lsv.ens-cachan.fr/spore/>.
- [MBJ+02] L. Mengual, N. Barcia, E. Jimnez, E. Menasalvas, J. Setin and J. Ygez, Automatic implementation system of security protocols based on formal description techniques, Proceedings of the Seventh International Symposium on Computers and Communications, pages 355-401, 2002.
- [Moz08a] Mozilla Corporation, NSPR, Netscape Portable Runtime, <http://www.mozilla.org/projects/nspr/>, 2008.
- [OAS05a] Organization for the Advancement of Structured Information Standards, eXtensible Access Control Markup Language (XACML) TC version 2.0, available at [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=xacml](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml), February 2005.
- [OAS05b] Organization for the Advancement of Structured Information Standards, Security Assertion Markup Language (SAML), version 2.0, available at <http://saml.xml.org/saml-specifications>, March 2005.
- [OAS07a] Organization for the Advancement of Structured Information Standards, Web Service Trust (WS-Trust) version 1.3, available at <http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html>, March 2007.
- [OAS07b] Organization for the Advancement of Structured Information Standards, Web Service Secure Conversation, version 1.3, available at <http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html>, March 2007.
- [OSSL08] OpenSSL Project, version 0.9.8h, available at <http://www.openssl.org/>, 2008.
- [Pat06] Victor Valeriu Patriciu, Semnături electronice și securitatea informatică, Editura All, București, 2006.
- [SPP01] Song Dawn, Adrian Perrig, and Doantam Phan, AGVI - Automatic Generation, Verification, and Implementation of Security Protocols, In Proceedings of the 13th Conference on Computer Aided Verification (CAV), Paris, France, July 2001.
- [SRW05] Adam Stubblefield, Aviel D. Rubin, Dan S. Wallach, Managing the Performance Impact of Web Security, Electronic Commerce Research, No. 5, Springer Science + Business Media, Inc. Manufactured in the Netherlands, pages 99-116, 2005.
- [TH05] Benjamin Tobler and Andrew C. M. Hutchison, Generating Network Security Protocol Implementations from Formal Specifications, IFIP International Federation for Information Processing, Springer Boston, pages 33-54, 2005.
- [Tsa02] Chii-Ren Tsai, Non-Repudiation in Practice, The Second International Workshop for Asian Public Key Infrastructures, Taipei, Taiwan, October 30-November 01, 2002.
- [VW01] M. Viredaz and D. Wallach, Power evaluation of a handheld computer: A case study, Compaq Western Research Lab, Tech. Rep. 2001/1, 2001.
- [WWWC04] World Wide Web Consortium, OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004.
- [WWWC05] World Wide Web Consortium, Web Service Semantics - WSDL-S, W3C Member Submission, 7 November, 2005.
- [WWWC06] World Wide Web Consortium, WS-Policy, available at <http://www.w3.org/Submission/WS-Policy/>, April 2006.
- [Ylo06] T. Ylonen, The Secure Shell (SSH) Protocol Architecture, SSH Communications Security Corp, RFC4251, January 2006.
- [Zha05] Meiyuan Zhao, Performance Evaluation of Distributed Security Protocols Using Discrete Event Simulation, Dartmouth Computer Science Technical Report TR2005-559, PhD Thesis, Dartmouth College, Hanover, New Hampshire, October, 2005.

# List of publications

## Indexed papers:

1. **Genge Bela**, Haller Pirooska, Middleware for Automated Implementation of Security Protocols, 6<sup>th</sup> European Semantic Web Conference, Heraklion, Greece, Lecture Notes in Computer Science (LNCS 5554), Springer-Verlag, pages 476-490, 2009 (DBLP, EI Compendex, INSPEC, ACM Portal).
2. **Genge Bela**, Haller Pirooska, Towards Automated Secure Web Service Execution, Networking 2009, Aachen, Germany, Lecture Notes in Computer Science (LNCS 5550), Springer-Verlag, pages 943-954, 2009 (DBLP, EI Compendex, INSPEC, ACM Portal).
3. **Genge Bela**, Iosif Ignat, Syntactic Sequential Composition of Security Protocols, Journal of Automation Computers Applied Mathematics (ACAM), Volume 17, No. 2, pages 169-178, 2008 (Mathematical Reviews).
4. **Genge Bela**, Haller Pirooska, A Modeling Framework for Generating Security Protocol Specifications, 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'08), Workshop on Global Computing Models and Technologies, Timisoara, Romania, pages 362-365, 2008, IEEE Computer Society Press (DBLP, INSPEC, EI Compendex).
5. Ovidiu Ratoi, Haller Pirooska, Ioan Salomie, **Genge Bela**, Component Based Platform for Multimedia Applications, 7th IEEE RoEduNet International Conference, Cluj-Napoca, Romania, pages 40-43, Aug. 2008 (ISI Web of Knowledge).
6. **Genge Bela**, Haller Pirooska, Ovidiu Ratoi, Constructing Security Protocol Specifications for Web Services, Intelligent Distributed Computing, Italy, Appears in Studies In Computational Intelligence, Springer-Verlag Berlin Heidelberg, Sept. 2008, Volume 162/2008, pages 245-250, 2008 (DBLP, Ulrichs, SCOPUS, MathSciNet, Zentralblatt).
7. **Genge Bela**, Haller Pirooska, Ovidiu Ratoi, Iosif Ignat, Term-based composition of security protocols, 2008 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, pages 233-238, May 2008 (IEEE Xplore, ISI Web of Knowledge).
8. **Genge Bela**, Haller Pirooska, Iosif Ignat, Ovidiu Ratoi, Informal specification-based performance evaluation of security protocols, 4th IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, pages 193-200, Aug. 2008 (IEEE Xplore, ISI Web of Knowledge).
9. **Genge Bela**, Haller Pirooska, Bindings for Security Protocol Composition, In the Proceedings of the 6th IEEE RoEduNet International Conference, Craiova, Romania, pages 64-69, November 2007 (ISI Web of Knowledge).
10. **Genge Bela**, Iosif Ignat, An Abstract Model for Security Protocol Analysis, WSEAS Transactions on Computers, Issue 2, Volume 6, pages, 207-215, February 2007 (INSPEC, Zentralblatt, Ulrichs).
11. **Genge Bela**, Iosif Ignat, Verifying the Independence of Security Protocols, 3rd IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, pages 155-163, September 2007 (IEEE Xplore, ISI Web of Knowledge).
12. **Genge Bela**, Iosif Ignat, A typed specification for security protocols, In the Proceedings of the 5th WSEAS Int. Conf. on Data Networks, Communications & Computers, Bucharest, Romania, pages 113-118, October 2006 (INSPEC, Zentralblatt, Ulrichs).
13. **Genge Bela**, Haller Pirooska, Towards a distributed authentication system in Coordinated Mobile Virtual Organizations, In the Proceedings of the 5th RoEduNet IEEE International Conference, Sibiu, Romania, pages 119-123, July 2006 (ISI Web of Knowledge).

## Non-indexed papers:

1. **Genge Bela**, Haller Pirooska, Extending WS-Security to Implement Security Protocols for Web Services, 1st International Conference on Recent Achievements in Mechatronics, Automation, Computer-Sciences and Robotics, Targu Mures, March 20, Appears in Acta Universitatis Sapientiae - Electrical and Mechanical Engineering, In Press, 2009.
2. Magyari Attila, **Genge Bela**, Haller Pirooska, Certificate-based Single Sign-On Mechanism for Multi-Platform Distributed Systems, 1st International Conference on Recent Achievements in Mechatronics, Automation, Computer-Sciences and Robotics, Targu Mures, March 20, Appears in Acta Universitatis Sapientiae - Electrical and Mechanical Engineering, In Press, 2009.
3. **Genge Bela**, Haller Pirooska, Middleware for Implementing Security Protocols, 8th International Conference on Computer Science and Energetics-Electrical Engineering, Sumuleu-Ciuc, Romania, pp. 139-144, October 2008.
4. **Genge Bela**, Haller Pirooska, A Chained Authentication Model for Virtual Organizations, Acta Universitatis Cibiniensis, VOL. LV, Technical Series, Sibiu, Romania, pages 60-68, 2007.
5. **Genge Bela**, Programming Manual for Smart Houses, Eliberatica – The benefits of Open and Free Technologies, Brasov, Romania, Locally edited booklet, June 2007.
6. **Genge Bela**, Haller Pirooska, Extending the Strand Space Model for Security Protocol Composition, International Scientific Conference "Interdisciplinarity in Engineering", Inter-Ing 2007, Targu Mures, Romania, pages 1-7, November 2007.
7. **Genge Bela**, Haller Pirooska, Attacks in single and multi-protocol environments, 7th International Conference on Computer Science and Energetics-Electrical Engineering, Oradea, Romania, pages 54-58, October 2007.
8. Haller Pirooska, **Genge Bela**, Security Issues in Wireless Distance Vector Routing Protocols, International Scientific Conference "Interdisciplinarity in Engineering", Inter-Ing 2005, Targu Mures, Romania, pages 662-668, 2005.