EXTENDING THE STRAND SPACE MODEL FOR SECURITY PROTOCOL COMPOSITION

Genge Bela[1], Haller Piroska[2]
[1, 2] "Petru Maior" University of Targu Mures, ROMANIA
{[1]bgenge, [2]phaller}@upm.ro

Abstract. We extend the regular strand space model with specialized terms and we provide a message construction schema for security protocols, based on grammatical rules. Using the extended model, we present two security protocol specification models that are used by the authors in the sequential and parallel composition of security protocols.

I. INTRODUCTION

Security protocols are communication protocols in which cryptography is used to give participants the capability to transmit encoded information that can be only decoded by the designated receivers. These protocols have been intensively analyzed throughout the last few decades, resulting in a variety of dedicated formal methods and tools [1, 2].

The majority of these methods consider a Dolev-Yao-like penetrator model [3] to capture the actions available to a penetrator, which has complete control over the network. By analyzing each individual protocol in the presence of this penetrator model, the literature has reported numerous types of attacks [4]. However, in practice, there can be multiple protocols running over the same network, thus the penetrator is given new opportunities to construct attacks by combining messages from several protocols, also known as multi-protocol attacks [4].

Because of the above-mentioned attacks, researchers have been mainly focusing throughout the last decades on the development of new design methods. One of the main security protocol design method is based on composition [5, 6, 7, 8, 9].

The composition process involves combining multiple basic protocols having certain security properties (e.g. one-way authentication, key exchange). This process, can be parallel or sequential, depending on the requirements. Because of the above-mentioned attack possibilities, designers should take great care because even a simple message modification can lead to serious attacks [10, 11].

In this paper we present two protocol specification models, proposed by the authors in [12, 13], allowing both sequential and parallel composition of security protocols. The proposed models are based on an extension of the strand space model proposed by Guttman et all in [14]. We show that because of the flexibility of the strand model, we can easily introduce new concepts and further extend the model so that it can be used to develop syntactical analysis methods and algorithms.

The paper is structured as follows. In section II we enrich the regular strand specification model with terms. In section III we present a specification model that can be used to develop new protocols based on parallel composition. In section IV we develop a model that can be used to verify the sequential compositionality of two security protocols. We end with a conclusion in section V.

II. EXTENDING THE STRAND SPACE MODEL

The proposed models are based on the strand space model, introduced by Guttman et all in [14]. In this section we provide an extended version of the original strand space model. In the regular strand-based specification, protocol participants are modeled using *strands*, where a *strand* denotes a sequence of message transmissions.

We propose to specialize the set of terms from the original model using *basic sets* and *function names*. By doing so, we are able to map each term to its corresponding type in the constructions that follow.

Roles (i.e. protocol participants) communicate by exchanging *terms* constructed from elements belonging to the following basic sets: $\mathcal{R}$, denoting the set of role names; $\mathcal{N}$, denoting the set of nonces (i.e. "number once used"); $\mathcal{K}$, denoting the set of cryptographic keys; $\mathcal{I}$, denoting the set of determinable or indexed numbers (e.g. timestamps, counters).

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

$$\begin{aligned} \mathit{FuncName} ::= {} & \mathrm{sk} && \text{(secret key)} \\ | {} & \mathrm{pk} && \text{(public key)} \\ | {} & \mathrm{pvk} && \text{(private key)} \\ | {} & \mathrm{h} && \text{(hash)} \end{aligned}$$

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

$$\begin{aligned} \mathcal{T} ::= . \mid \mathcal{R} \mid \mathcal{N} \mid \mathcal{K} \mid (\mathcal{T}, \mathcal{T}) \\ \mid \{\mathcal{T}\}_{\mathit{FuncName}(\mathcal{T})} \end{aligned}$$

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

The terms that can be constructed by roles, based on the definitions above, include the basic components of a wide variety of security protocols. However, these can be further extended if the modeled protocol requires other types.

The composition process of two terms t_1 and t_2 into another term t implies that t has sub-terms. The subterm relation $\sqsubset$ is inductively defined as follows.

Definition 1. *The subterm relation $\sqsubset$ is the smallest relation on terms such that:*

1. $t \sqsubset t$;
2. $t \sqsubset \{t_1\}_{f(t2)}$ *if* $t \sqsubset t_1$ *or* $t \sqsubset t_2$;
3. $t \sqsubset (t_1, t_2)$ *if* $t \sqsubset t_1$ *or* $t \sqsubset t_2$.

Having defined the terms exchanged by participants, we can proceed with the definition of a *strand* and a *strand space* (i.e. a collection of strands). To capture the sending and receiving of terms, the strand model introduces *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 2. *A* signed term *is a pair* $\langle \sigma, t \rangle$ *with* $t \in \mathcal{T}$ *and* σ *one of the symbols +,-. A signed term is written as -t or +t.* $(\pm\mathcal{T})^*$ *is the set of finite sequences of signed terms. A typical element of* $(\pm\mathcal{T})^*$ *is denoted by* $\langle \pm t_1, \pm t_2, \ldots, \pm t_n \rangle$, *with* $t_i \in \mathcal{T}$.

Definition 3. *A* strand *is a sequence of term transmissions and receptions, represented as* $\langle \pm t_1, \pm t_2, \ldots, \pm t_n \rangle \in (\pm\mathcal{T})^*$. *A set of strands is called a strand space and is denoted by* Σ.

1. *A* node *is any transmission or reception of a term, written as* $n = \langle s, i \rangle$, *with* $s \in \Sigma$ *and i an integer satisfying the condition* $1 \le i \le length(s)$. *The set of all nodes is denoted by* $\mathcal{N}$.
2. *Let* $n_1 = \langle s, i \rangle$ *and* $n_2 = \langle s, i+1 \rangle$ *be two consecutive nodes from* $\mathcal{N}$ *on the same strand* $s \in \Sigma$. *Then, there exists an edge* $n_1 \Rightarrow n_2$ *in the same strand.*
3. *Let* $n_1, n_2 \in \mathcal{N}$. *If* n_1 *is a positive node and* n_1 *is a negative node and* $strand(n_1) \neq strand(n_2)$, *then there exists an edge* $n_1 \rightarrow n_2$.

III. PARALLEL COMPOSITION

In this section we present a specification model based on which designers can implement composition algorithms. The proposed model is based on the concept of *binding*, denoting the existing link between security protocol message terms.

We use *Basic Typed Terms* to denote the structure of security protocol messages. In the next sections, however, these will be extended and intensively used in the model construction.

BasicTT ::=	r	(*role type*)
	| n	(*nonce type*)
	| k	(*key type*)
	| b	(*binding type*)

Definition 4. *A* binding *is a tuple written as* $\langle \rho, \nu, \kappa, \beta, f, k, \theta \rangle$, *where* $\rho \in \mathsf{R}^*$, $\nu \in \mathsf{N}^*$, $\kappa \in \mathsf{K}^*$, $\beta \in \mathsf{B}^*$, $f \in FuncName$, $k \in \mathsf{K}$ *and* $\theta \in BasicTT^*$. *We use the* B *symbol to denote the set of all bindings and the symbol* B^* *to denote the set of all subsets of bindings. The '.' symbol is used to denote an empty binding set.*

1. *To obtain the components of a binding b, we use the following projection functions:*

$$Roles(b) = \rho, Nonces(b) = \nu, Keys(b) = \kappa,$$
$$Bindings(b) = \beta, Func(b) = f, BindingKey(b) = k,$$
$$TypeSet(b) = \theta$$

2. *The binding composition operator* $\_+\_ : \mathsf{B} \times \mathsf{B} \rightarrow \mathsf{B}$ *composes all subsequent components using set operators;*
3. *The sub-binding operator* $\sqsubset$ *is defined inductively as follows:*

$b \sqsubset b$

$b_1 \sqsubset b_2$ *if*

$$\begin{aligned}
&Roles(b_1) \subseteq Roles(b_2) \wedge \\
&Nonces(b_1) \subseteq Nonces(b_2) \wedge \\
&Keys(b_1) \subseteq Keys(b_2) \wedge \\
&Bindings(b_1) \subseteq Bindings(b_2) \wedge \\
&Func(b_1) = Func(b_2) \wedge \\
&BindingKey(b_1) = BindingKey(b_2)
\end{aligned}$$

Definition 5. *A* b-strand $s : (\pm \mathsf{B})^* \times C$ *is a sequence of binding transmissions and receptions attached to a strand classifier. A set of b-strands is called a b-strand space and is represented as* Σ. We use the Σ^* *symbol to denote a set of b-strand space subsets.*

1. *A* node *is any transmission or reception of a binding, written as* $n_i = \langle s, i \rangle$, *with* $s \in \Sigma$ *and i an integer satisfying the condition* $1 \le i \le length(s)$, *where* $length(s)$ *is a function returning the number of nodes from a b-strand. The set of all nodes is denoted by* $\mathcal{N}$.
2. *Let* $n_1 = \langle s, i \rangle$ *and* $n_2 = \langle s, i+1 \rangle$ *be two consecutive nodes from* $\mathcal{N}$ *on the same b-strand s. Then, there exists an edge* $n_1 \Rightarrow n_2$ *in the same b-strand s.*
3. *Let* $n_1, n_2 \in \mathcal{N}$. *If* n_1 *is positive and* n_2 is *negative, and* $bstrand(n_1) \neq bstrand(n_2)$, *then there exists an edge* $n_1 \rightarrow n_2$.
4. *Let* $n \in \mathcal{N}$. *Then* $sign(n)$ *is a function returning the sign and* $binding(n)$ *is a function returning the binding corresponding to a given node.*
5. *Let* $s \in \Sigma$ *with* $s = \langle \beta_s, c \rangle$. *Then we define the following projection functions:* $(s)_1 = \beta_s \quad (s)_2 = c$

Definition 6. A *role specification* is a pair $\langle r, \xi \rangle$, *such that* $r \in \mathsf{R}$ *and* $\xi \in \Sigma^*$. *Let* r_S *be a role specification. Then Role*(r_S) *is a projection function returning the role name and BStrands*(r_S) *is a projection function returning the set of b-strands corresponding to a role specification.*

A set of role specifications is denoted by RoleSpec and $RoleSpec^*$ *denotes the set of all subsets of role specifications.*

Simply specifying the sequence of messages for a protocol is not enough. A specification must also include initial role knowledge defined as follows.

Definition 7. Role knowledge *is a pair* $\langle r, b \rangle$, *where* $r \in \mathsf{R}$ *and* $b \in \mathsf{B}$, *such that Func*(*b*)=*BindingKey*(*b*)= . . *We use RoleKnow to denote a set of role knowledge and* $RoleKnow^*$ *to denote the set of all subsets of role knowledge.*

In a constructed b-protocol, each message is accompanied by the knowledge required to construct the given message and the message sequence that has to be exchanged by protocol participants. Thus, each message can be treated independently

from other protocol messages, allowing the implementation of composition algorithms.

IV. SEQUENTIAL COMPOSITION

The goal of sequential composition is to create a protocol that allows the *secure* running of two or more security protocol alongside each other. The involved protocols can be completely independent or they can achieve sequential composition of security properties. In both cases, the independence of the protocol must be verified so that protocols are not subject to multi-protocol attacks, as described in [12].

In this section we construct a security protocol specification model that outlines the message components, which are used to verify the sequential compositionality of the modeled protocols.

Typed terms are a vital component of the proposed model. These are created by applying term forming constructs, defined below, to *basic typed terms* and *function names*. Our notion of a *basic typed term* has been formalized in the previous section, however, these must be extended with typed keys and unknown types, such as:

$$\begin{aligned} BasicTT ::= \; & BasicTT \\ & \mid \mathsf{K}_t \qquad (\textit{typed keys}) \\ & \mid \mathsf{u} \qquad (\textit{nonce type}) \end{aligned}$$

Typed terms are constructed using the following grammatical productions:

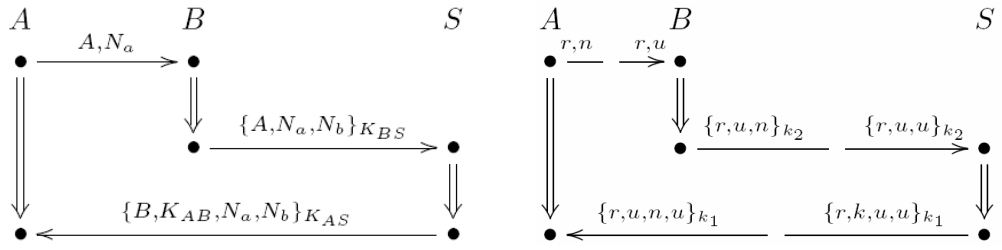
$$\begin{aligned} \mathcal{T}_t ::= . \mid BasicTT \mid & \left(\mathcal{T}_t, \mathcal{T}_t\right) \\ \mid & \left\{\mathcal{T}_t\right\}_{FuncName(\mathcal{T}_t)} \end{aligned}$$

Definition 8. *A* signed typed term *is a pair* $\langle \sigma, t \rangle$ *with* $t \in \mathcal{T}_t$ *and* σ *one of the symbols +,-. A signed typed term is written as -t or +t.* $\left(\pm\mathcal{T}_t\right)^*$ *is the set of finite sequences of signed terms. A typical element of* $\left(\pm\mathcal{T}_t\right)^*$ *is denoted by* $\langle \pm t_1, \pm t_2, \ldots, \pm t_n \rangle$, *with* $t_i \in \mathcal{T}_t$.

Definition 9. *A* t-strand *is a sequence of typed term transmissions and receptions, represented as* $\langle \pm t_1, \pm t_2, \ldots, \pm t_n \rangle \in \left(\pm\mathcal{T}_t\right)^*$. *A collection of t-strands is called a t-strand space and is denoted by* Σ_t.

4. *A* typed node *is any transmission or reception of a typed term, written as* $n_t = \langle s, i \rangle$, *with* $s_t \in \Sigma_t$ *and* i *an integer satisfying the condition* $1 \le i \le length(s)$. *The set of all nodes is denoted by* $\mathcal{N}_t$.
5. *Let* $n_{t1} = \langle s, i \rangle$ *and* $n_{t2} = \langle s, i+1 \rangle$ *be two consecutive nodes from* $\mathcal{N}_t$ *on the same t-strand* $s_t \in \Sigma_t$. *Then, there exists an edge* $n_{t1} \Rightarrow n_{t2}$ *in the same t-strand.*

Let $n_{t1}, n_{t2} \in \mathcal{N}_t$. *If* n_{t1} *is a positive node and* n_{t1} *is a negative node and* $strand\left(n_{t1}\right) \neq strand\left(n_{t2}\right)$, *then there exists an edge* $n_{t1} \rightarrow n_{t2}$.



(a) (b)

Figure 1. (a) The Yahalom-Lowe protocol
(b) The typed specification of the Yahalom-Lowe protocol

The transformation functions used to create the typed model from the regular strand model are the following:

$$TTr(t) = \begin{cases} r, & \text{if } t \in \mathsf{R}, \\ n, & \text{if } t \in \mathsf{N}, \\ k_i, & \text{if } t \in \mathsf{K}, \\ (TTr(t_1), TTr(t_2)), & \text{if } t = (t_1, t_2), \\ \{TTr(t_1)\}_{f(TTr(t_2))}, & \text{if } t = \{t_1\}_{f(t_2)}. \end{cases}$$

$$KTTr(\kappa, t) = \begin{cases} (KTTr(\kappa, t_1), KTTr(\kappa, t_2)), & \text{if } t = (t_1, t_2), \\ \{KTTr(\kappa, t_1)\}_{f(KTTr(\kappa, t_2))}, & \text{if } t = \{t_1\}_{f(t_2)}, \\ TTr(t), & \text{if } t \in \kappa, \\ u, & \text{otherwise.} \end{cases}$$

By using the above-mentioned functions, we exemplify the transformation process by using the Yahalom-Lowe security protocol. The regular and typed specifications are presented in Figure 1.

V. CONCLUSIONS

In this paper we have presented two extensions of the strand space model. We have enriched the regular specification with terms, bindings and typed terms. For the typed strand model we have also provided transformation functions so that the transformation process from one model to another is error-prone.

The binding-based model presented in this paper has been used by the authors to create a new protocol [13] from two existing security protocols. The resulting protocol is not only a composition of the involved messages, but a composition of the security properties embodied in the two protocols.

The type-based model has also been used by the authors in the past to verify the independence of security protocols [12]. This model is not limited, however, to the composition process of two different security protocols. It can also be used to check for replay attacks in single protocols, as shown by the authors in [10, 15].

REFRENCES

[1] F.J.T. Fabrega, J.C. Herzog, J.D. Guttman, “Strand Spaces: Proving security protocols correct”, Journal of Computer Science, Vol. 7, pp. 191-230, 1999.
[2] C. Weidenbach, “Towards an automatic analysis of security protocols”, In the Proc. of the 16th International Conference on Automated Deduction, pp. 378-382, 1999.
[3] D. Dolev, A. Yao, “On the security of public-key protocols”, IEEE Transactions on Information Theory, Vol. 29, pp. 198-208, 1983.
[4] C.J.F. Cremers, “Feasability of Multi-Protocol Attacks”, In the Proc. of the first ARAS conference, 2006.
[5] Hyun-Jin Choi, “Security protocol design by composition”, Cambridge University, UK, Technical report Nr. 657, 2006.
[6] A. Datta, A. Derek, J. C. Mitchell, A. Roy, “Protocol Composition Logic (PCL)”, Electronic Notes in Theoretical Computer Science, to appear, 2007.
[7] Cas J. F. Cremers, “Compositionality of Security Protocols: A Research Agenda”, Electr. Notes Theor. Comput. Sci., 142, pp. 99-110, 2006.
[8] 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, pp. 265-281, 2003.
[9] S. Andova, Cas J.F. Cremers, K. Gjosteen, S. Mauw, S. Mjolsnes, and S. Radomirovic, “A framework for compositional verification of security protocols”, to appear, 2007.
[10] Genge Bela, Iosif Ignat, “An Abstract Model for Security Protocol Analysis”, WSEAS Transactions on Computers, Issue 2, Volume 6, pp. 207-215, 2007.
[11] Genge Bela, Haller Piroska, “Attacks in single and multi-protocol environments”, XVII Szamokt, Oradea, pp. 54-58, 2007.
[12] Genge Bela, Iosif Ignat, “Verifying the Independence of Security Protocols”, IEEE 3rd International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, pp.155-163, 2007.
[13] Genge Bela, Haller Piroska, “Binding groups for security protocol message composition”, to appear, 2007.
[14] F. Javier Thayer Fabrega, Jonathan C. Herzog, Joshua D. Guttman, “Strand spaces: Proving security protocols correct”, Journal of Computer Security 7, 191-230, 1999.
[15] Genge Bela, Iosif Ignat, “A typed specification for security protocols”, Proceedings of the 5th WSEAS Int. Conf. on Data Networks, Communications and Computers, Bucharest, Romania, October 16-17, pp. 113-118, 2006.