PRACTICAL AND UNIVERSAL INTERPRETATION FUNCTIONS FOR SECRECY

Hanane Houmani and Mohamed Mejri
LSFM Research Group, Computer Science Department
Laval University, Quebec, Canada

Keywords: Cryptographic protocols, Secrecy, Formal verification, Sufficient conditions.

Abstract: Using the notion of interpretation functions, this paper gives some sufficient and practical conditions allowing to guarantee the correctness of a security protocol with respect to the secrecy property. An interpretation function is a safe means by which an agent can estimate the security level of message components that he receives so that he can handle them correctly. An example of an universal interpretation function is given in this paper together with how to use it to analyse a cryptographic protocol.

1 INTRODUCTION

The verification of security of cryptographic protocols is paramount, since they are used to make secure our communications and transactions. The problem of the security verification of cryptographic protocols is undecidable in general (see (Cervesato et al., 1999; Even and Goldreich, 1983; Heintze and Tygar, 1996)). To deal with this problem, researchers proposed a large variety of methods and tools. A general survey could be found in (Boreale and Gorla, 2002; Comon and Shmatikov, 2002; Meadows, 2003).

Theses approaches can be classified in two classes. The first one contains decidable methods where some of them could be find in (Comon-Lundh and Cortier, 2003a; Gangon and Mejri, 2006; Lowe, 1998; Ramanujam and Suresh, 2003; Stoller, 1999). Using these methods, one can decide whether some protocols are correct or not. This is due to the fact that these methods make strong restrictions on the analyzed protocols in order to make the analysis decidable. The second class contains semi-decidable methods and some of them could be found in (Abadi, 1999; Blanchet and Podelski, 2003; Burrows et al., 1990; Comon-Lundh and Cortier, 2003b; Durgin et al., 2001; Houmani and Mejri, 2003; Mao and Boyd, 1993; Paulson, 1997). Some of these approaches aim to prove that a given protocol respects some security properties and others try to detect flaws on them. However, in both cases, if the goal isn't reached, the security of the analyzed protocol cannot be guaranteed.

The major drawback of decidable and semi-decidable approaches allowing to guarantee the secrecy property of a protocol is their strong restrictive assumptions and any new less restrictive approach is more than welcome.

The main result of this paper is to propose some conditions that are not very restrictive but sufficient to guarantee the correctness of any protocol satisfying them with respect to the secrecy property. Besides, the results presented in this paper are generic enough to deal with the verification of a large variety of protocols in different contexts (different intruder capabilities or different syntax of messages).

Intuitively, to guarantee that a protocol is correct with respect to the secrecy property using our approach, one need to find an *interpretation function* and to prove that each role (agent) involved in the protocol does not decrease the security level of component of messages when he sends them over the network. Protocols that satisfy this condition will be called increasing protocol.

An interpretation function could be seen as a means that can be used by agents involved in a protocol to estimate, in a safe way, the security level of unknown components that they receive during the protocol. If the security level of components handled

Houmani H. and Mejri M. (2007).
PRACTICAL AND UNIVERSAL INTERPRETATION FUNCTIONS FOR SECRECY.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 157-164
DOI: 10.5220/0002129101570164
Copyright © SciTePress

by each agent are known or could be estimated in a safe way, then the verification of the protocol will be considerably simplified. In fact, it will be enough to check whether the agents protect in an appropriate way the messages that they send over the network to guarantee the secrecy property of the protocol.

The remainder of this paper is organized as follows. Section 2 formalizes some parameters that affect the verification of protocols are called the context of verification and clarifies the protocol specification and the notion of generalized roles. Section 3 gives a formal definition of the secrecy property based on valid traces. Section 5 introduces the proposed conditions and provides a proof that are sufficient to ensure the secrecy property of a protocol. Section 6 presents a guideline to define a safe interpretation function. Section 7 gives an example showing how to verify the secrecy property of a given protocol by using an interpretation function. Section 8 discusses the approach and compares it with some existing ones. Finally, section 9 provides a few concluding remarks and future works.

2 MODELING PROTOCOL

This section introduces the notions *verification context* and *role-based specification* of a protocol.

2.1 Context of Verification

Intuitively, a verification context is a structure that contains all the parameters that affect the verification of a protocol. Basically, we find in this structure the intruder capabilities, the knowledge of each principal and the security level of each atomic component. More precisely, a verification context, denoted by $\mathbb{M}$, is a quintuple $\langle \mathcal{M}, \models, \mathcal{K}, \mathcal{L}, \ulcorner . \urcorner \rangle$, where :

- $\mathcal{M}$ is the message algebra. It describes the class of messages involved in the analyzed protocol. Notice that we use $\mathcal{A}$ to denote the set of atomic messages in $\mathcal{M}$ and $\mathcal{A}(M)$ to denote the set of atomic messages in a given set of messages M.
- $\models$ is the intruder capabilities. It is a set of inference rules showing how the intruder can infer new messages.
- $\mathcal{K}$ is the sets of messages (fresh and non-fresh messages) that are initially known by each agent including the intruder. More precisely, $\mathcal{K}$ is a set of triples $(A, fr(A), \overline{fr}(A))$, where A is the name of the agent, $fr(A)$ is a set of fresh messages known by A and $\overline{fr}(A)$ is the set of non-fresh messages known by A. Notice that if $\mathbb{M} = \langle \mathcal{M}, \models, \mathcal{K}, \mathcal{L}, \ulcorner . \urcorner \rangle$ is a model, then we denote by K_A the set of messages (fresh and non fresh) known by A in $\mathbb{M}$.
- $(\mathcal{L}, \sqsupseteq)$ or simply $\mathcal{L}$ is a security lattice. It is used to describe the security levels of components involved in the protocol together with an ordering relation between them.
- $\ulcorner . \urcorner$ is a function from $\mathcal{A}$ to $\mathcal{L}$. It specifies the security level of components that are initially known by principals including the intruder. If A is a set of atomic messages and α is an atomic component, we say $\ulcorner M \urcorner \sqsupseteq \ulcorner \alpha \urcorner$ if it exists β in M such that $\ulcorner \beta \urcorner \sqsupseteq \ulcorner \alpha \urcorner$

Example 2.1 *In the sequel, we denote by $\mathbb{M}_0$, the following verification model:*

- *$\mathcal{M}_0$ is the message algebra given by the the following BNF grammar.*

$$\begin{array}{lcll} m & ::= & A & \textit{(Principal Identifier)} \\ & | & N_a & \textit{(Nonce)} \\ & | & k_{ab} & \textit{(Shared key)} \\ & | & X & \textit{(Variable)} \\ & | & \{m\}_{k_{ab}} & \textit{(Encrypted Message)} \\ & | & m, m' & \textit{(Concatenated Message)} \end{array}$$

  In the sequel, we denote by $\mathcal{I}_{\mathcal{M}}$ the set of identifiers in $\mathcal{M}$.
- *$\models_0$ the following commun intruder rules:*

$$\text{(init)} \quad \frac{\square}{M \models m}[m \in M]$$

$$\text{(decrypt)} \quad \frac{M \models k \qquad M \models \{m\}_k}{M \models m}$$

$$\text{(encrypt)} \quad \frac{M \models k \qquad M \models m}{M \models \{m\}_k}$$

$$\text{(concat)} \quad \frac{M \models m_1 \qquad M \models m_2}{M \models m_1.m_2}$$

$$\text{(deconcat)} \quad \frac{M \models m_1.m_2}{M \models m_i}[i = 1,2]$$

  where $M \models m$ means that the intruder can infer m from M using the previous rules.
- *$\mathcal{K}_0 = \{(A, fr(A), \overline{fr}(A)),\ (B, fr(B), \overline{fr}(B)),\ (S, fr(S), \overline{fr}(S)),\ (I, fr(I), \overline{fr}(I))\}$, where :*

$$\begin{array}{lcl} fr(A) & = & \{k_{ab}\} \\ fr(B) & = & \{N_b\} \\ fr(S) & = & \emptyset \\ fr(I) & = & \{N_i^1, N_i^2, \ldots\} \\ \overline{fr}(A) & = & \{k_{as}, A, B, S\} \\ \overline{fr}(B) & = & \{k_{bs}, A, B, S\} \\ \overline{fr}(S) & = & \{k_{bs}, k_{as}, A, B, S\} \\ \overline{fr}(S) & = & \{k_{is}, A, B, S\} \end{array}$$

- $\mathcal{L}_0 = (2^{\mathcal{I}_\mathcal{M}}, \supseteq)$.
- $\ulcorner . \urcorner_0$. *The security type of an atomic message involved in the protocol is the set of principals that are allowed to know this message.*

$$\begin{array}{rcl} \ulcorner . \urcorner_0 & = & [N_b, N_i^1, N_i^2, \ldots \mapsto \bot,\ A, B, S \mapsto \bot, \\ & & k_{ab} \mapsto \{A,B,S\},\ k_{is} \mapsto \{I,S\} \\ & & k_{as} \mapsto \{A,S\},\ k_{bs} \mapsto \{B,S\}] \end{array}$$

2.2 Protocol

Basically, a protocol P is specified by a sequence of communication steps given in the standard notation. Hereafter, we give an example which is a variant extracted from the Woo and Lam (Woo and Lam, 1994) authentication protocol. As shown by Table 1, this variant aims to distribute a fresh key that will be shared between two agents A and B.

Table 1: Woo and Lam modified protocol.

$$\begin{array}{ll} P = & \langle 1, A \rightarrow B : A \rangle. \\ & \langle 2, B \rightarrow A : N_b \rangle. \\ & \langle 3, A \rightarrow B : \{N_b, k_{ab}\}_{k_{as}} \rangle. \\ & \langle 4, B \rightarrow S : \{A, \{N_b, k_{ab}\}_{k_{as}}\}_{k_{bs}} \rangle. \\ & \langle 5, S \rightarrow B : \{N_b, k_{ab}\}_{K_{bs}} \rangle \end{array}$$

From a verification context $\mathbb{M}$ and a protocol P, we extract a role-based specification (a set of generalized roles) that we denote by $\mathcal{R}_G(P)$. A generalized role is a protocol abstraction, where the emphasis is put on a particular principal and where all the unknown messages are replaced by variables. An exponent i (the session identifier) to each fresh message to reflect the fact that these components change their values from one run to another is added. Basically, a generalized role reflects how a particular agent perceives the exchanged messages. More details about generalized roles and how they extracted from a protocol could be find in (Houmani and Mejri, 2003). For instance, the role-based specification of the protocol described in Table 1, $\mathcal{R}_G(P)$, is $\{\mathcal{A}_G^1, \mathcal{A}_G^2, \mathcal{B}_G^1, \mathcal{B}_G^2, \mathcal{B}_G^3, \mathcal{S}_G^1\}$ (see Table 2).

The role-based specification is used to formalize the notion of valid trace. A trace is considered as valid if all honest participants act according to the protocol specification and all the messages sent by the intruder are derivable from his intercepted messages, his initial knowledge and his inference rules given within the verification context. It is considered that an honest agent acts according to the protocol specification if any given run in which he participates is an instance

Table 2: Generalized roles of Woo and Lam protocol.

$\mathcal{A}_G^1 =$	$\langle i.1,$	A	$\rightarrow$	$I(B)$	$:$	$A \rangle$
$\mathcal{A}_G^2 =$	$\langle i.1,$	A	$\rightarrow$	$I(B)$	$:$	$A \rangle.$
	$\langle i.2,$	$I(B)$	$\rightarrow$	A	$:$	$X \rangle.$
	$\langle i.3,$	A	$\rightarrow$	$I(B)$	$:$	$\{X, k_{ab}^i\}_{k_{as}} \rangle$
$\mathcal{B}_G^1 =$	$\langle i.1,$	$I(A)$	$\rightarrow$	B	$:$	$A \rangle.$
	$\langle i.2,$	B	$\rightarrow$	$I(A)$	$:$	$N_b \rangle$
$\mathcal{B}_G^2 =$	$\langle i.1,$	$I(A)$	$\rightarrow$	B	$:$	$A \rangle.$
	$\langle i.2,$	B	$\rightarrow$	$I(A)$	$:$	$N_b \rangle.$
	$\langle i.3,$	$I(A)$	$\rightarrow$	B	$:$	$Y_1 \rangle.$
	$\langle i.4,$	B	$\rightarrow$	$I(S)$	$:$	$\{A, Y_1\}_{k_{bs}} \rangle$
$\mathcal{B}_G^3 =$	$\langle i.1,$	$I(A)$	$\rightarrow$	B	$:$	$A \rangle.$
	$\langle i.2,$	B	$\rightarrow$	$I(A)$	$:$	$N_b \rangle.$
	$\langle i.3,$	$I(A)$	$\rightarrow$	B	$:$	$Y_2 \rangle.$
	$\langle i.4,$	B	$\rightarrow$	$I(S)$	$:$	$\{A, Y_2\}_{k_{bs}} \rangle.$
	$\langle i.5,$	$I(S)$	$\rightarrow$	B	$:$	$\{N_b^i, Z\}_{k_{bs}} \rangle$
$\mathcal{S}_G^1 =$	$\langle i.4,$	$I(B)$	$\rightarrow$	S	$:$	$\{A, \{U, V\}_{k_{as}}\}_{k_{bs}} \rangle.$
	$\langle i.5,$	S	$\rightarrow$	$I(B)$	$:$	$\{U, V\}_{k_{bs}} \rangle$

(variables are replaced by constant messages) of one of his generalized roles. The notation $[\![P]\!]$ means the set of the valid traces associated to the protocol P. Also, $[\![P]\!] \models_{K_I} \alpha$ means that the protocol P exhibits a trace allowing an intruder, having an initial knowledge K_I and using the inference rules defined by $\models$, modeled by $\models_{K_I}$, to know the message α. To verify the secrecy property, this paper proves that it is sufficient to verify whether the generalized roles of the analyzed protocol respect some conditions. This verification is decidable, because the set of generalized roles of a protocol is finite.

3 SECRECY PROPERTY

Intuitively, a protocol keeps a component m secret, if it has not a valid trace that decrease the security level of m. More precisely, the formal definition of the secrecy property given hereafter states that the intruder cannot learn from any valid trace more than what he is eligible to know. We suppose that if an agent (including the intruder) knows a message with a security level τ, then he is also eligible to know all messages having security level lower than τ[1].

Definition 3.1 (Secrecy Property) *Let P be a protocol and $\mathbb{M} = \langle \mathcal{M}, \models, \mathcal{K}, \mathcal{L}, \ulcorner . \urcorner \rangle$ a verification context. The protocol P is $\mathbb{M}$-correct with respect the secrecy property, if:*

$$\forall \alpha \in \mathcal{A}(\mathcal{M}) \cdot \ [p] \models_{K_I} \alpha \ \Rightarrow \ \ulcorner K_I \urcorner \sqsupseteq \ulcorner \alpha \urcorner$$

4 SAFE INTERPRETATION FUNCTION

The aim of safe interpretation functions is to give a correct means to the principals involved in the protocol by which they can estimate the security level of the received components.

Definition 4.1 (Interpretation function) *Let $\mathbb{M} = \langle \mathcal{M}, \models, \mathcal{K}, \mathcal{L}, \ulcorner . \urcorner \rangle$ be a verification context. An $\mathbb{M}$-Interpretation function F is function from $\mathcal{A}(\mathcal{M}) \times \mathcal{M}$ to $\mathcal{L}$.*

Before introducing the notion of safe interpretation functions, we need to define the following ordering relation.

Definition 4.2 ($\sqsupseteq_{\mathsf{F}}$) *Let F be an $\mathbb{M}$-Interpretation function. We say that $M_1 \sqsupseteq_{\mathsf{F}} M_2$ if:*

$$\forall \alpha \in \mathcal{A}(M_1) \cdot \ \mathsf{F}(\alpha, M_1) \sqsupseteq \mathsf{F}(\alpha, M_2)$$

For the sake of simplicity, $\{m\} \sqsupseteq_{\mathsf{F}} M$ is replaced by $m \sqsupseteq_{\mathsf{F}} M$, and $\mathsf{F}(\alpha, \{m\})$ is replaced by $\mathsf{F}(\alpha, m)$.

Now a safe interpretation function could be defined as following:

Definition 4.3 (Safe Interpretation Function) *Let $\mathbb{M} = \langle \mathcal{M}, \models, \mathcal{K}, \mathcal{L}, \ulcorner . \urcorner \rangle$. An $\mathbb{M}$-Interpretation function is called $\mathbb{M}$-Safe if the following conditions are respected:*

1. *F is well formed, i.e:* $\mathsf{F}(\alpha, \{\alpha\}) = \bot$ *and* $\mathsf{F}(\alpha, M_1 \cup M_2) = \mathsf{F}(\alpha, M_1) \sqcap \mathsf{F}(\alpha, M_2)$ *and* $\mathsf{F}(\alpha, M) = \top$ *where* $\alpha \notin \mathcal{A}(M)$
2. *F is full-invariant by substitution, i.e: for all M_1 and M_2 two set of messages in $\mathcal{M}$ such that $Var(M_1) \subseteq Var(M_2)$ and $M_1 \sqsupseteq_{\mathsf{F}} M_2$ we have:*
$$\forall \sigma \in \Gamma \cdot \ M_1\sigma \sqsupseteq_{\mathsf{F}} M_2\sigma$$
where Γ is the set of possible substitution form $\mathcal{X}$ to close messages in $\mathcal{M}$.
3. *F is full-invariant by intruder, i.e:* $\forall M \subseteq \mathcal{M}, \forall \alpha \in \mathcal{A}(M)$ *such that* $\mathsf{F}(\alpha, M) \sqsupseteq \ulcorner \alpha \urcorner$ *and* $\forall m \in \mathcal{M}$ *such that* $M \models_{K_I} m$ *we have:*
$$\forall \alpha \in \mathcal{A}(m) \cdot \ (\mathsf{F}(\alpha, m) \sqsupseteq \mathsf{F}(\alpha, M) \vee (\ulcorner K_I \urcorner \sqsupseteq \ulcorner \alpha \urcorner)$$

[1]Notice that it is always possible to define a security lattice that reflects our needs and which is coherent with this hypothesis.

5 MAIN RESULT

Now, it is time to give the sufficient conditions allowing to guarantee the correctness of a protocol with respect to the secrecy property. Informally, these conditions state that *(a)* honest agents should never decrease the security level of any atomic message and *(b)* they have to use a safe interpretation function to estimate the security level of component that they didn't initially know. To formalize the condition *(a)*, we introduce the following definition.

Definition 5.1 (Increasing Protocol)
Let $\mathbb{M}$ be a verification context, F an $\mathbb{M}$-interpretation function and P a protocol. The protocol P is said to be F-increasing if:

$$\forall r \in \mathcal{R}_G(P), \forall \alpha \in \mathcal{A}(r^+) \cdot \ \mathsf{F}(\alpha, r^+) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathsf{F}(\alpha, r^-)$$

where r^+ is a set containing the messages sent during the last step of r and r^- contains the set of messages received by the honest agent in r.

Now the main theorem could be formalized as following:

Theorem 5.2 *Let P be a protocol, $\mathbb{M}$ a verification context and F a $\mathbb{M}$-interpretation function . If F is $\mathbb{M}$-safe and P is F-increasing, then P is $\mathbb{M}$-correct with respect to the secrecy property.*

Proof:
Due to the lake of space, proofs have been removed and can be found in (Houmani and Mejri, 2007a).
□

6 GUIDELINES TO DEFINE SAFE INTERPRETATION FUNCTIONS

According to theorem 5.2, the first step of the verification of secrecy property is to find a *safe interpretation function* and this is the delicate part of the approach. In fact, an interpretation function needs to be full-invariant by substitution and full-invariant by intruder. For that reason, this section provides some guidelines allowing to easily construct a safe interpretation function.

Basically, we show hereafter that any interpretation function that has some given form is safe. In particular, we focus on interpretation functions having the following form:

$$\mathsf{F}(\alpha, M) = \mathsf{I} \circ \mathsf{S}(\alpha, M)$$

where S is a function that selects from M some atomic components having some links with α and where I is a function that interprets what S returns as a security type.

If we consider a message, which is a term in the message algebra, as a tree and we attach to each arc of this tree an integer value (reflecting costs or distance between nodes), then it will be easy to define S that select some components that are at some distance from a given component. To formalize the the notion of distance, we introduce a function $\mathcal{T}$ that takes a message and attach a value to each arc in its corresponding tree. An example of $\mathcal{T}$ is as following:

$$\begin{array}{rcl} \mathcal{T}((c,\alpha)) & = & 0 \\ \mathcal{T}((c,e)) & = & \infty \\ \mathcal{T}((e,c)) & = & 0 \\ \mathcal{T}((e,\alpha)) & = & 1 \end{array}$$

where e is the encryption operator, c is the concatenation operator and α is an atomic message. If we apply this function to the message "$B,\{N_b,k_{ab}\}_{k_{as}}$", we obtain $\mathcal{T}$("$B,\{N_b,k_{ab}\}_{k_{as}}$") given by Fig. 1.

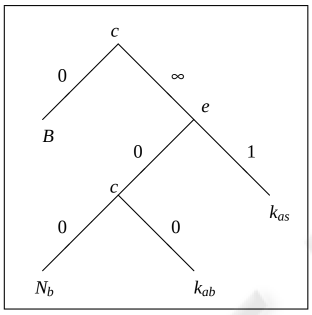

Figure 1: $\mathcal{T}$("$B,\{N_b,k_{ab}\}_{k_{as}}$").

In the rest of this paper, $d_{\mathcal{T}(m)}(\alpha,\beta)$ (or simply $d_m(\alpha,\beta)$ if $\mathcal{T}(m)$ is clear from the context) denotes the smallest distance between α and β in $\mathcal{T}(m)$. This notion can be easily extended to a set of messages M as follows: $d_M(\alpha,\beta) = Min_{m\in M}(d_m(\alpha,\beta))$

Now, we introduce the n-selection function $\mathsf{S}^n_{\mathcal{T}}$ as following:

$$\mathsf{S}^n_{\mathcal{T}}(\alpha,M) = \begin{cases} \mathcal{A} & \text{if } \alpha \text{ appears in clear in } M \\ \{\beta\in\mathcal{A}(M) \mid d_{\mathcal{T}(M)}(\alpha,\beta) = n\} & \text{else} \end{cases}$$

The definition of $\mathsf{S}^n_{\mathcal{T}}$ is extended to $\mathsf{S}^{\vec{n}}_{\mathcal{T}}$, called the n-vector selection function, as following:

$$\mathsf{S}^{\vec{n}}_{\mathcal{T}}(\alpha,M) = (\mathsf{S}^0_{\mathcal{T}}(\alpha,M), \mathsf{S}^1_{\mathcal{T}}(\alpha,M),\ldots,\mathsf{S}^n_{\mathcal{T}}(\alpha,M))$$

This n-vector selection function will play a key role in the definition of a safe interpretation function.

Any n-vector selection function $\mathsf{S}^{\vec{n}}_{\mathcal{T}}$ from $\mathcal{A}\times 2^{\mathcal{M}}$ to $((2^{\mathcal{A}})^n,\subseteq)$ can be proved full-invariant by substitution, where $\subseteq$ is extended to deal with vectors as follows:

$$(M_1,M_2,\ldots,M_n)\subseteq(M'_1,M'_2,\ldots,M'_n) \Leftrightarrow \forall i\in\{1,\ldots,n\}\cdot M_i\subseteq \bigcup_{j=1}^{i} M'_j$$

This full-invariance of n-vector selection function is a useful property to help us to construct an interpretation function that is full-invariant by substitution.

Now, some restrictions on $\mathcal{T}$ are needed so that $\mathsf{S}^{\vec{n}}_{\mathcal{T}}$ will be also full-invariant by intruder manipulation. A simple and practical condition that $\mathcal{T}$ needs to satisfy so that $\mathsf{S}^{\vec{n}}_{\mathcal{T}}$ becomes full-invariant by intruder is given by definition 6.1.

Definition 6.1 (Safe Cost Attribution (SCA)) *We say that $\mathcal{T}$ is an SCA if $\mathcal{T}(o,e)=\infty$, where where e is an encryption operator and o is any another node.*

Having a selection function that is full-invariant by substitution and full-invariant by intruder is not enough to construct a safe function $\mathsf{F}=\mathsf{I}\circ\mathsf{S}$ that is also full-invariant by substitution and full-invariant by intruder. The function I needs also to be restricted so that it can preserves the full-invariance properties of S. For instance, the selection function $\mathsf{S}^n_{\mathcal{T}}$ returns a vector of sets of atomic components that will be interpreted by I as a security type. These atomic components could contain variables and therefore the function I needs to carefully handle them so that the "full-invariance" properties of $\mathsf{S}^n_{\mathcal{T}}$ are preserved.

Before giving how to choose I, we need to introducing the following notations: Given a security lattice $(\mathcal{L},\sqsupseteq)$, we introduce the two following useful extensions.

- $(\mathcal{L}_{\overline{X}},\sqsupseteq_{\overline{X}})$: where $\overline{X}$ is a set of variables that range over security types, $\mathcal{L}_{\overline{X}}=\mathcal{L}\cup\overline{X}$ and the relation $\sqsupseteq_{\overline{X}}$ is defined as follows:

$$\tau_1\sqsupseteq_{\overline{X}}\tau_2\Leftrightarrow\forall\sigma\in\overline{X}\times\mathcal{L}:\ \tau_1\sigma\sqsupseteq\tau_2\sigma$$

- $(\mathcal{L}^n_{\overline{X}},\sqsupseteq^n_{\overline{X}})$ is an extension of $(\mathcal{L}_{\overline{X}},\sqsupseteq_{\overline{X}})$ as following:
  - $\mathcal{L}^n_{\overline{X}}=\underbrace{\mathcal{L}_{\overline{X}}\times\ldots\times\mathcal{L}_{\overline{X}}}_{n}$
  - $\sqsupseteq^n_{\overline{X}}$ is defined as follows:

$$(\tau_1,\ldots,\tau_n)\sqsupseteq^n_{\overline{X}}(\tau'_1,\ldots,\tau'_n)\Leftrightarrow \forall i\in\{1,\ldots,n\}\cdot\tau_i\sqsupseteq_{\overline{X}}\bigsqcap_{j=1}^{i}\tau'_j$$

The following theorem shows the condition that need to be respected by I so that we get a safe interpretation function.

Theorem 6.2 *Let $\mathbb{M}$ be a context of verification and $\mathsf{S}^{\vec{n}}_{\mathcal{T}}$ an n-vector selection function. If:*

- *$\mathcal{T}$ be a SCA, and*
- *I is a morphism from $((2^{\mathcal{A}})^n, \subseteq)$ to $(\mathcal{L}^n_{\overline{X}}, \sqsupseteq^n_{\overline{X}})$*

Then, $\mathsf{F} = \mathsf{I} \circ \mathsf{S}^{\overrightarrow{n}}_{\mathcal{T}}$ is $\mathbb{M}$-safe.

Proof:
Due to the lake of space, proofs have been removed and can be found in (Houmani and Mejri, 2007b).
□

Practical steps to construct F: To define a safe interpretation function F, one can follow these steps:

1. Define a n-vector selection function $\mathsf{S}^{\overrightarrow{n}}_{\mathcal{T}}$. To do this, we should define the following elements:
   - Choose a positive integer value for n.
   - Choose $\mathcal{T}$ such that it is an SCA.

2. Define a morphism I from $((2^{\mathcal{A}})^n, \subseteq)$ to $(\mathcal{L}^n_{\overline{X}}, \sqsupseteq^n_{\overline{X}}$). Given a security lattice $(\mathcal{L}, \sqsupseteq)$, a morphism can be built simply by following these steps:
   - Define mapping I from $\mathcal{A}$ to $\mathcal{L}$.
   - Extend the mapping I to deal with variable as following:
     $$\mathsf{I}(x) = \bar{x} \qquad \text{if } x \text{ is a variable}$$
     where $\bar{x}$ is a variable that ranges over security types. This extension is done to appropriately deal with variables that could be returned by $\mathsf{S}^n_{\mathcal{T}}$.
   - Extend I to be able to handle sets a another as following:
     $$\mathsf{I}(A_1 \cup A_2) = \mathsf{I}(A_1) \sqcap \mathsf{I}(A_2)$$
   - Extend the function I to be able to deal with vectors as following:
     $$\mathsf{I}((A_1, \ldots, A_n)) \;=\; (\mathsf{I}(A_1), \ldots, \mathsf{I}(A_n))$$

3. Define an $\mathbb{M}$-safe function F as:
   $$\mathsf{F} = \mathsf{I} \circ \mathsf{S}^{\overrightarrow{n}}_{\mathcal{T}}$$

6.1 Example

In this example, we define an $\mathbb{M}_0$-safe function, denoted by F_0, where $\mathbb{M}_0$ is the verification context described in section 2.1. To that end we proceed according to the previous steps:

1. Define a n-vector selection function $\mathsf{S}^{\overrightarrow{n}}_{\mathcal{T}}$:
   - Choose a value for n: In this example, we choose $n = 0$.

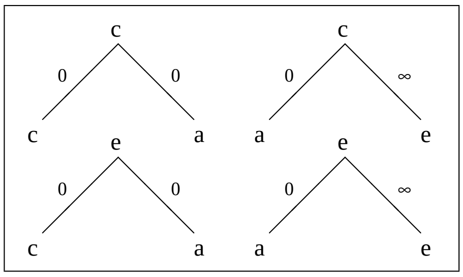

Figure 2: Example of costs.

   - Define an SCA $\mathcal{T}$: In this example, $\mathcal{T}$ attaches values to arcs as shown by Fig. 2, where a is an atomic message. Intuitively, $\mathcal{T}$ allow $\mathsf{S}^{\overrightarrow{n}}_{\mathcal{T}}(\alpha, M)$ to select the innermost keys that encrypt α in M together with the atomic components that are concatenated to α in M (called direct neighbors).

2. Define a morphism $\mathsf{I}_{\overline{X}}$ from $(2^{\mathcal{A}}, \subseteq)$ to $(2^{\mathcal{I}_{\overline{X}}}, \subseteq_{\overline{X}})$. In this example, let I be the mapping defined as follows $\mathsf{I}_(A) = A$ and $\mathsf{I}(\alpha_{ab}) = \{A, B\}$. After that we can extend this mapping with following steps described previously to the mapping I that is from $(2^{\mathcal{A}}, \subseteq)$ to $(2^{\mathcal{I}_{\overline{X}}}, \subseteq_{\overline{X}})$.

3. The interpretation function is now as following:
   $$\mathsf{F}_0 = \mathsf{I} \circ \mathsf{S}^{\overrightarrow{0}}_{\mathcal{T}}$$
   For instance:
   $$\begin{array}{rcl}\mathsf{F}_0(\alpha, \{\{\alpha, B, X\}_{k_{as}}\}_{k_{bs}}) & = & \mathsf{I}(\{B, X, k_{as}\}) \\ & = & \{A, S, B, \} \cup \overline{X}\end{array}$$
   This function is called DEKAN (**D**irect **E**ncrypting **K**eys **a**nd **N**eighbors). We believe that such a DEKAN function is powerful enough to allow us to appropriately analyze a large class of protocols with respect to the secrecy property.

7 CASES STUDY

By using the DEKAN safe interpretation function F_0 and the theorem 5.2, one can prove that P (the version of Woo and Lam protocol given by Table 1) is $\mathbb{M}_0$-correct with respect to the secrecy property. To this end, we need only to prove that the protocol is F_0-increasing, i.e., for each generalized role r in Table 2, we have:

$$\forall \alpha \in \mathcal{A}(r^+) \cdot \mathsf{F}_0(\alpha, r^+) \subseteq_{\overline{X}} \ulcorner \alpha \urcorner \cup \mathsf{F}_0(\alpha, r^-)$$

For the role $\mathcal{A}_G^1$, since $\ulcorner A \urcorner = \bot$ and F is well-defined ($\mathsf{F}(A,A) = \bot$), then the role $\mathcal{A}_G^1$ is F-increasing:

$$\mathsf{F}_0(A,A) \subseteq_{\overline{X}} \ulcorner A \urcorner \cup \mathsf{F}_0(A,A)$$

For the role $\mathcal{A}_G^2$, since $\mathsf{F}_0(k_{ab}^i, \{\{X, k_{ab}^i\}_{k_{as}}\}) = \{A,S\} \cup \overline{X}$, $\mathsf{F}_0(X, \{\{X, k_{ab}^i\}_{k_{as}}\}) = \{A,B,S\}$, $\ulcorner k_{ab}^i \urcorner = \{A,B,S\}$, $\ulcorner X \urcorner = \top$ and F is well-defined ($\mathsf{F}_0(X,X) = \bot,\ \mathsf{F}_0(k_{ab}^i, X) = \top$), then the role $\mathcal{A}_G^2$ is not F-increasing. Indeed, we have:

$$\left\{ \begin{array}{l} \mathsf{F}_0(k_{ab}^i, \{\{X, k_{ab}^i\}_{k_{as}}\}) \not\subseteq_{\overline{X}} \ulcorner k_{ab}^i \urcorner \cup \mathsf{F}_0(k_{ab}^i, X) \\ \mathsf{F}_0(X, \{\{X, k_{ab}^i\}_{k_{as}}\}) \subseteq_{\overline{X}} \ulcorner X \urcorner \cup \mathsf{F}_0(X, X) \end{array} \right.$$

For the role $\mathcal{B}_G^1$, since $\ulcorner N_b^i \urcorner = \bot$ and F is well-defined ($\mathsf{F}_0(N_b^i, N_b^i) = \bot,\ \mathsf{F}_0(N_b^i, A) = \top$), then the role $\mathcal{B}_G^1$ is F-increasing. Indeed, we have:

$$\mathsf{F}_0(N_b^i, N_b^i) \subseteq_{\overline{X}} \ulcorner N_b^i \urcorner \cup \mathsf{F}_0(N_b^i, A)$$

For the role $\mathcal{B}_G^2$, since $\mathsf{F}_0(Y_1, \{\{A, Y_1\}_{k_{bs}}\}) = \{A,B,S\}$, $\mathsf{F}_0(A, \{\{A, Y_1\}_{k_{bs}}\}) = \{B,S\} \cup \overline{Y_1}$, $\ulcorner A \urcorner = \bot$, $\ulcorner Y_1 \urcorner = \top$ and F is well-defined ($\mathsf{F}_0(Y_1, \{Y_1, A\}) = \bot, \mathsf{F}_0(A, \{Y_1, A\}) = \bot$), then the role $\mathcal{B}_G^2$ is F-increasing. Indeed, we have:

$$\left\{ \begin{array}{l} \mathsf{F}_0(Y_1, \{\{A, Y_1\}_{k_{bs}}\}) \subseteq_{\overline{X}} \ulcorner Y_1 \urcorner \cup \mathsf{F}_0(Y_1, \{Y_1, A\}) \\ \mathsf{F}_0(A, \{\{A, Y_1\}_{k_{bs}}\}) \subseteq_{\overline{X}} \ulcorner A \urcorner \cup \mathsf{F}_0(A, \{Y_1, A\}) \end{array} \right.$$

For the role $\mathcal{S}_G^1$, since $\mathsf{F}_0(U, \{U,V\}_{k_{bs}}) = \{B,S\} \cup \overline{V}$, $\mathsf{F}_0(V, \{U,V\}_{k_{bs}}) = \{B,S\} \cup \overline{U}$, $\mathsf{F}_0(U, \{A, \{U,V\}_{k_{as}}\}_{k_{bs}}) = \{A,S\} \cup \overline{V}$, $\mathsf{F}_0(V, \{A, \{U,V\}_{k_{as}}\}_{k_{bs}}) = \{A,S\} \cup \overline{U}$ and $\ulcorner U \urcorner = \ulcorner V \urcorner = \top$, then the role $\mathcal{S}_G^1$ is not F-increasing. Indeed, we have:

$$\left\{ \begin{array}{l} \mathsf{F}_0(U, \{U,V\}_{k_{bs}}) \not\subseteq_{\overline{X}} \ulcorner U \urcorner \cup \mathsf{F}_0(U, \{A, \{U,V\}_{k_{as}}\}_{k_{bs}}) \\ \mathsf{F}_0(V, \{U,V\}_{k_{bs}}) \not\subseteq_{\overline{X}} \ulcorner V \urcorner \cup \mathsf{F}_0(V, \{A, \{U,V\}_{k_{as}}\}_{k_{bs}}) \end{array} \right.$$

Therefore, this protocol is not an F_0-increasing and we can not ensure its correctness with respect to the secrecy property. Moreover, it is not difficult to see that this protocol contains a flaw and therefore we will never be able to construct a a safe interpretation function that makes it an increasing one. What is interesting however, is that the previous verifications help as to localize the origin of the problem which are in steps 3 and 5. For instance in step 3, the message k_{ab}^i is sent with security level equal to $\{A,S\} \cup \overline{X}$ which is not in its security level specified in the protocol $\{A,B,S\}$ for all the possible values of $\overline{X}$ (a similar problem appears in step 5). To resolve this problem and prove the correctness by using F_0 as interpretation function, we need to modify the protocol. One possible way of modifying this protocol is as shown by Table 3).

We can now prove that this new version is F_0-increasing.

Table 3: Woo and Lam modified protocol: Second version.

$$\begin{array}{ll} P = & \langle 1, A \to B : A \rangle. \\ & \langle 2, B \to A : N_b \rangle. \\ & \langle 3, A \to B : \{\{N_b\}_A, B, k_{ab}\}_{k_{as}} \rangle. \\ & \langle 4, B \to S : \{A, \{\{N_b\}_A, B, k_{ab}\}_{k_{as}}\}_{k_{bs}} \rangle. \\ & \langle 5, S \to B : \{\{N_b\}_A, A, k_{ab}\}_{k_{bs}}\}_{k_{bs}} \rangle \end{array}$$

8 DISCUSSION

The approach proposed in this paper shares some characteristics with others existing works like the secrecy by typing technique proposed in (Abadi, 1999) and rank functions proposed in (Schneider, 1998; Delicata and Schneider, 2005).

With the secrecy by typing technique, we share the idea of having some secure way allowing to propagate the security level of exchanged information during a protocol so that they can be correctly handled by the receivers. To this end, the secrecy by typing approach uses standard forms of messages ($\{secret, any, public, confounder\}_k$) so that it will be easy to deduce its "secret", "public" and "any" (component with unknown security level) parts. This involves that this approach can only ensure the security of protocols that exchange messages having the standard format. With our approach, however, the security level of any component can be inferred in a safe way by using what we called interpretation function. It is possible for us to define an interpretation function, when the message have the standard form giving a safe way to know the security level of its components (like the one used in the secrecy by typing approach), but also it still possible to define an interpretation function even such kind of standard are not respected by the protocol. This gives to our approach the ability of certifying the security of a larger class of protocols.

With the rank functions techniques, we share the idea of using inductive proofs and the use of a function that allows to know whether a message is appropriately protected or not. However, interpretation functions are more precise than rank functions. This is due to fact that rank functions are global functions that ensure if a message is globally constructed in a secure way. However, interpretation functions are local functions than allows to know whether each component of messages are convenably protected or not. Therefore, the interpretation functions give more help to detect problems in a protocol, and how to modify it so that its security could be guaranteed. In fact, when they fail to ensure the security of a protocol they give the exact components that could be inappropriately protected. We believe also that interpretation functions are more helping during the conception of new protocols and could give some guidelines.

9 CONCLUSION

By using the notion of safe interpretation functions, this paper gives sufficient conditions to guarantee the correctness of a cryptographic protocol with respect to the the secrecy property. These conditions could be verified in a linear time on the protocol. It gives also a practical way to construct these kind of interpretation functions.

As future works, we want to show the efficiency of this approach by verifying real life protocols such as SSL, SET, Kerberos, etc. We would like also to define other safe and universal interpretation functions. Finally, it will be interesting to see wether this approach could help in order analyze others security properties.

REFERENCES

Abadi, M. (1999). Secrecy by typing in security protocols. *Journal of the ACM*, 46(5):749–786.

Blanchet, B. and Podelski, A. (2003). Verification of cryptographic protocols: Tagging enforces termination. In *Foundations of Software Science and Computational Structures*, volume 2620 / 2003, pages 136–152, Warsaw, Poland. Springer-Verlag Heidelberg.

Boreale, M. and Gorla, D. (2002). Process calculi and the verification of security properties. *Journal of Telecommunication and Information Technology—Special Issue on Cryptographic Protocol Verification*, (4/02):28–40.

Burrows, M., Abadi, M., and Needham, R. (1990). Rejoinder to Nessett. *ACM Operating Systems Review*, 24(2):39–40.

Cervesato, I., Durgin, N. A., Lincoln, P., Mitchell, J. C., and Scedrov, A. (1999). A meta-notation for protocol analysis. In *CSFW*, pages 55–69.

Comon, H. and Shmatikov, V. (2002). Is it possible to decide whether a cryptographic protocol is secure or not. *Journal of Telecommunications and Information Technolog,*.

Comon-Lundh, H. and Cortier, V. (2003a). New decidability results for fragments of first-order logic and application to cryptographic protocols. In *RTA*, pages 148–164.

Comon-Lundh, H. and Cortier, V. (2003b). Security properties: Two agents are sufficient. In *ESOP*, pages 99–113.

Delicata, R. and Schneider, S. (2005). Temporal rank functions for forward secrecy. In *CSFW '05: Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW'05)*, pages 126–139, Washington, DC, USA. IEEE Computer Society.

Durgin, N., Mitchell, J., and Pavlovic, D. (2001). A compositional logic for protocol correctness.

Even, S. and Goldreich, O. (1983). On the security of multiparty ping-pong protocols. In *IEEE Symposium on Foundations of Computer Science*, pages 34–39.

Gangon, F. and Mejri, M. (2006). A decision procedure for structured cryptographic protocols. In *New Trends in Software Methodologies, Tools and Techniques*, pages 272–286. IOS Press.

Heintze, N. and Tygar, J. D. (1996). A model for secure protocols and their compositions. *Software Engineering*, 22(1):16–30.

Houmani, H. and Mejri, M. (2003). Secure protocols for secrecy. In *Foundations of Computer Security Afiliated with LICS'03*, pages 85–96, Ottawa, Canada.

Houmani, H. and Mejri, M. (2007a). Secrecy by interpretation functions. *Knowledge-Based Systems*, doi:10.1016/j.knosys.2007.05.003.

Houmani, H. and Mejri, M. (2007b). Secrecy by interpretation functions: Extended version. Thechnical Report, `www.ift.ulaval.ca\~hahou\techReport1.pdf`.

Lowe, G. (1998). Towards a Completeness Result for Model Checking of Security Protocols. In *Proceedings of 11th IEEE Computer Security Foundations Workshop*, pages 96–108.

Mao, W. and Boyd, C. (1993). Towards the Formal Analysis of Security Protocols. In *Proceedings of the Computer Security Foundations Workshop VI*, pages 147–158. IEEE Computer Society Press.

Meadows, C. (2003). What makes a cryptographic protocol secure? the evolution of requirements specification in formal cryptographic protocol analysis. In *Proceedings of ESOP 03*. Springer-Verlag.

Paulson, L. C. (1997). Proving properties of security protocols by induction. In *10th Computer Security Foundations Workshop*, pages 70–83. IEEE Computer Society Press.

Ramanujam, R. and Suresh, S. (2003). Tagging makes secrecy decidable with unbounded nonces as well. In *Lecture Notes in Computer Science*, volume 2914/2003. FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, Publisher Springer Berlin / Heidelberg.

Schneider, S. (1998). Verifying authentication protocols in csp. *IEEE Trans. Softw. Eng.*, 24(9):741–758.

Stoller, S. D. (1999). Lower and upper bounds for attacks on authentication protocols. In *Symposium on Principles of Distributed Computing*, page 283.

Woo, T. Y. C. and Lam, S. S. (1994). A Lesson on Authentication Protocol Design. *Operating Systems Review*, pages 24–37.