ENSURING THE CORRECTNESS OF CRYPTOGRAPHIC PROTOCOLS WITH RESPECT TO SECRECY

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

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

Abstract: This paper gives sufficient conditions to ensure secrecy property of cryptographic protocols that allow to share a session keys. Indeed, this paper proves that if within a protocol agents don't decease or increase the security level of components, then this protocol respect the secrecy property. This sufficient condition holds even we change our context of verification (message algebra, intruder capacities or cryptographic assumptions). To verify this condition we use the notion of interpretation functions. An interpretation function is a safe way allowing an agent to appropriately estimate the security level of message components that he receives so that he can handle them correctly.

1 INTRODUCTION

The verification of cryptographic protocols is paramount, since they are used to make secure our communications and transactions. However, the verification of the security of cryptographic protocols is undecidable in general (see (Comon and Shmatikov, 2002; Comon-Lundh and Cortier, 2003a)). Therefore, researchers have proposed a large variety of methods and tools that can help to find attacks or to prove the security for some classes of these protocols. A general survey of the most used approaches related to the verification of cryptographic protocols could be found in (Boreale and Gorla, 2002; Meadows, 2003; Sabelfeld and Myers, 2003).

Due to the complexity of the problem, almost of all the existing approaches try to simplify it by making some restrictions on cryptographic primitives like the perfect encryption assumption. However, protocols may contain flaws outside the considered assumptions. For example, in (Paulson, 1997), L. Paulson has proven that the Bull protocol preserves secret using an intruder model that does not take into account any algebraic property of cryptographic primitives. However, he proved that attacks are possible on this protocol if some algebraic properties of exponentiation or $\oplus$ are considered. Since then, many researchers (Chevalier et al., 2003; Comon-Lundh and Cortier, 2003b; Goubault-Larrecq, 2005; Jacquemard et al., 2000; Abadi and Cortier, 2006; Shmatikov, 2004; Turuani, 2003), have been trying to study the problem of the security of the cryptographic protocols under equational theories (sets of the algebraic properties).

In (Houmani and Mejri, 2007a; Houmani and Mejri, 2007b; Houmani and Mejri, 2008), we gathered assumptions, restrictions and algebraic properties that can be made on cryptographic protocols on what we called a context of verification. More specifically, a context of verification is basically the specification of messages algebra and the capacities of the intruder (including set of equational theories). This representation allowed us to give sufficient conditions that are independent from specific context of verification (specific class of messages, a specific capacity of the intruder, or specific class of equational theories) and that guarantee the secrecy property of any protocol that respect them. Intuitively, these sufficient conditions state that agents should not decrease the security level of message components when they send them over the network. Protocols that respect this condition were called *Increasing Protocols*.

However, if the analyzed protocol use a temporary key (session key), we will not be able to prove that it respects the secrecy property even if it is the case. This is due to the fact that the sufficient conditions verify wether agents decrease the security level of a message without caring about the exact values

Houmani H. and Mejri M. (2008).
ENSURING THE CORRECTNESS OF CRYPTOGRAPHIC PROTOCOLS WITH RESPECT TO SECRECY.
In *Proceedings of the International Conference on Security and Cryptography*, pages 184-189
DOI: 10.5220/0001927401840189

of these security levels. Suppose for instance that an agent has received the message $\{k\}_{k_{ab}}$, hence from that message we can safely approximate the security level of k by saying that it is less than the security level of k_{ab}. However, to send a message $\{\alpha\}_k$ over the network, it is not sufficient to know a lower bound for the security level of k. but we need an upper bound also and ideally if we know the exact value of this security level so that we can know wether α is correctly protected when sent inside $\{\alpha\}_k$. Therefore, to ensure that a protocol does not leak secret information, we need to ensure that each agent protects the components that he send according to their security types (public key cannot protect secret information for example). More precisely, it is sufficient to restrict each agent so that he does not decrease or increase the security levels of sent components to guarantee the secrecy property of a protocol. Protocols that respect this condition are called, in this paper, *Coherent Protocols*.

However, we need first to find a way allowing to safely communicate the security level of each component send over the network. In fact, an agent cannot appropriately protect a component (especially for received and previously unknown components) if he is not able to deduce his security level. For this purpose, we use what we call interpretation function. The role of this function is to allow each agent involved in the protocol to deduce in a safe way the security level of each received component. Once the interpretation function is defined, it will be enough to restrict each agent so that he does not decrease or increase the security levels of sent components to guarantee that the protocol is correct with respect to the secrecy property.

The remainder of this paper is organized as follows. Section 2 gives the definition of a context of verification. Also, it gives the definition of some basic words used within this paper. Section 3 gives a formal definition for the secrecy property. Section 4 introduces the proposed conditions and proves that they are sufficient to ensure the secrecy property of cryptographic protocols. Section 5 shows how to put in practice these conditions with a concrete example. Finally, section 6 provides some concluding remarks.

2 BASIC DEFINITIONS

Basically, this section gives the definition of a context of verification already introduced in (Houmani and Mejri, 2007b; Houmani and Mejri, 2007a). Also, it gives the definition of a set of messages and the definition of the intruder capacities.

Context of Verification. Parameters like the structure of messages exchanged during the protocol, the intruder capacities or the algebraic properties of cryptographic primitives, could affect the class of protocols that could be analyzed by an approach. We found therefore interesting to gather them in what we called a *context of verification*. A context of verification can have the following form $\mathcal{C} = \langle \mathcal{N}, \Sigma, \mathcal{E}, K, \mathcal{L}^{\sqsupseteq}, \ulcorner \cdot \urcorner \rangle$, where:

- *The Names* $\mathcal{N}$ is the set of names (nounce, keys, etc). For instance, let $\mathcal{N}_0$ be the set of names given by the the following BNF grammar:

$$\begin{array}{rcll} n & ::= & A & \text{(Principal Identifier)} \\ & | & N_a & \text{(Nonce)} \\ & | & k_{ab} & \text{(Shared key)} \end{array}$$

- *The Signature* Σ contains all function symbols (encryption and pair symbol for example). For instance, let Σ_0 be the signature defined as follows:

$$\Sigma_0 = \{enc, dec, pair, fst, snd\}$$

  As usual we write $\langle x,y \rangle$ instead of writing $pair(x,y)$.

- *The Equational Theory* $\mathcal{E}$ is the equational theory that represents the algebraic properties of the function symbols (commutativity of the pair symbol for example). For instance, Let $\mathcal{E}_0$ be the equational theory that contains the following equations:

$$\begin{array}{rcl} dec(enc(x,y),y) & = & x \\ fst(pair(x,y)) & = & x \\ snd(pair(x,y)) & = & y \end{array}$$

- *The Intruder Knowledge* K is the set of initial knowledge of the intruder. For instance, let K_0 be the set of knowledge of intruder that contains shared keys k_{ia}, k_{ib}, etc, a public key k_i, a private key k_i^{-1}, and a infinite set of fresh values as sessions keys, nonces, and timestimps .

- *The Lattice of Security* $\mathcal{L}^{\sqsupseteq}$ is a lattice that contains security levels (types). For example the poset $(\{classified, secret, topSecret\}, \sqsubseteq)$ where $classified \sqsubseteq secret \sqsubseteq topSecret$ can define a simple lattice of security. Another interesting security lattice is the one defined by the powerset of agents identities i.e 2^I. Within this lattice the security level of a component is the set of identities of agents allowed to know the value of this component. In the sequel, we denote this powerset lattice by $\mathcal{L}_0^{\subseteq}$.

- *The Typed Environment* $\ulcorner \cdot \urcorner$ is a partial function that assigns to atomic messages their real security

levels (types). This allows us to know the security level of components initially known by each agent. For instance, let $\ulcorner \cdot \urcorner_0$ be a typed environment that assigns to a message α the set of identities of agents that could know α. For example, $\ulcorner k_{ab} \urcorner_0 = \{A, B\}$.

Messages. Given a context of verification $\mathcal{C}$, a set of messages $\mathcal{M}$ can be defined (this definition is inspired from (Abadi and Cortier, 2006).) by the following BNF grammar:

$$\begin{array}{rcll} m & ::= & N & \text{(Name)} \\ & | & X & \text{(Variable)} \\ & | & f(m_1,\ldots,m_n) & \text{(Function application)} \end{array}$$

Notice that the set of messages involved in the verification context will be denoted, in that follows, by $\mathcal{M}$, and the set $\mathcal{A}(M)$ denotes the set of atomic components (nonces, keys and principal identifiers) in M.

Intruder. Given a context of verification $\mathcal{C} = \langle \mathcal{N}, \Sigma, \mathcal{E}, K, \mathcal{L}^{\sqsupseteq}, \ulcorner \cdot \urcorner \rangle$ and a set of message M that represents the information available to an intruder, the message m can be deduced by the intruder from M in the context $\mathcal{C}$ and we write $M \models_{\mathcal{C}} m$ if m can be obtained by using these rules:

Table 1: Generic capacities of intruder.

$$(\mathsf{Init}) \quad \frac{\square}{M \models_{\mathcal{C}} m}[m \in M \cup K_I]$$

$$(\mathsf{Eq}) \quad \frac{M \models_{\mathcal{C}} m_1 \quad m_1 =_{\mathcal{C}} m_2}{M \models_{\mathcal{C}} m_2}$$

$$(\mathsf{Op}) \quad \frac{M \models_{\mathcal{C}} m_1,\ \ldots,\ M \models_{\mathcal{C}} m_n}{M \models_{\mathcal{C}} f(m_1,\ldots,m_n)}[f \in \Sigma]$$

Given a context of verification $\mathcal{C} = \langle \mathcal{N}, \Sigma, \mathcal{E}, K, \mathcal{L}^{\sqsupseteq}, \ulcorner \cdot \urcorner \rangle$, we write $m_1 =_{\mathcal{C}} m_2$ if and only if $m_1 =_{\mathcal{E}} m_2$, where $m_1 =_{\mathcal{E}} m_2$ means that the message m_1 and m_2 are equal under the equational theory $\mathcal{E}$.

Protocol. Basically, a protocol is specified by a sequence of communication steps given in the standard notation. More precisely a protocol p has to respect the following BNF grammar:

$$p \quad ::= \langle i :\ A \rightarrow B : m \rangle \mid p.p$$

The statement $\langle i :\ A \rightarrow B : m \rangle$ denotes the transmission of a message m from the principal A to the principal B in the step i. Let p_0 be a variant of the Woo and Lam (Woo and Lam, 1994) authentication protocol. This variant, given by Table 2, aims to distribute a new key that will be shared between two agents A and B.

Table 2: Example of a Protocol.

$$\begin{array}{rl} p_0 = & \langle 1, A \rightarrow B : A \rangle. \\ & \langle 2, B \rightarrow S : \{N_b, A, k_{ab}\}_{k_{bs}} \rangle. \\ & \langle 3, S \rightarrow A : \{N_b, B, k_{ab}\}_{k_{as}} \rangle. \\ & \langle 4, A \rightarrow B : \{N_b, s_{ab}\}_{k_{ab}} \rangle \end{array}$$

In this paper, we denote by $[\![p]\!]$ all valid traces (executions) of a protocol p. Also, we denote by $\mathcal{R}_G(p)$ the role-based specification (Debbabi et al., 1997; Houmani and Mejri, 2007b) of p which is a set of generalized roles. 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. A session identifier i is added as an exponent to each fresh message to reflect the fact that these components change their values from one run to another. Basically, a generalized role reflects how a particular agent perceives the exchanged messages. For instance, the role-based specification of the protocol described in Table 2, $\mathcal{R}_G(P)$, 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 3).

Table 3: Example of Roles-Based Specification.

$$\begin{array}{|rl|} \hline \mathcal{A}_G^1 = & \langle i.1, A \rightarrow I(B) : A \rangle \\ \hline \mathcal{A}_G^2 = & \langle i.1, A \rightarrow I(B) : A \rangle. \\ & \langle i.3, I(S) \rightarrow A : \{X_1, B, X_2\}_{k_{as}} \rangle. \\ & \langle i.4, A \rightarrow I(B) : \{X_1, s_{ab}\}_{X_2} \rangle \\ \hline \mathcal{B}_G^1 = & \langle i.1, I(A) \rightarrow B : A \rangle. \\ & \langle i.2, B \rightarrow I(S) : \{N_b, A, k_{ab}\}_{k_{bs}} \rangle \\ \hline \mathcal{B}_G^2 = & \langle i.1, I(A) \rightarrow B : A \rangle. \\ & \langle i.2, B \rightarrow I(S) : \{N_b, A, k_{ab}\}_{k_{bs}} \rangle. \\ & \langle i.4, I(A) \rightarrow B : \{N_b, Y_1\}_{k_{ab}} \rangle \\ \hline \mathcal{S}_G^1 = & \langle i.2, I(A) \rightarrow S : \{U, A, V\}_{k_{bs}} \rangle. \\ & \langle i.3, S \rightarrow I(A) : \{U, B, V\}_{k_{as}} \rangle \\ \hline \end{array}$$

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 1 (Secrecy Property). *Let p be a protocol and $\mathcal{C} = \langle \mathcal{N}, \Sigma, \mathcal{E}, K, \mathcal{L}^{\sqsupseteq}, \ulcorner\cdot\urcorner \rangle$ a verification context. The protocol p is $\mathcal{C}$-correct with respect the secrecy property, if:*

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

where $\sqsupseteq$ is the order involved from the security lattice given in $\mathcal{C}$ and the notation $\ulcorner K \urcorner \sqsupseteq \ulcorner \alpha \urcorner$ is an abbreviation of: $\exists \beta \in K \cdot \ulcorner \beta \urcorner \sqsupseteq \ulcorner \alpha \urcorner$. Notice that this abbreviation will be used throughout the rest of this paper.

4 MAIN RESULT

Now, it is time to give the sufficient conditions allowing to guarantee the correctness of a cryptographic protocol with respect to the secrecy property. Informally, these conditions state that honest agents should never decrease or increase the security level of any atomic message. However, to reach this goal, principals involved in the protocol need a "safe" way allowing them to compute the security level of a component received within message during the protocol execution. By a "safe" way, we mean that the computed security level can neve be mislead by the intruder. To this end, we introduce what we call a *safe interpretation function* allowing to safely compute the security level of components send and received with messages. Formally:

Definition 2 (Safe Interpretation Function). *Let $\mathcal{C} = \langle \mathcal{N}, \Sigma, \mathcal{E}, K, \mathcal{L}^{\sqsupseteq}, \ulcorner\cdot\urcorner \rangle$ be a context of verification. A function function F, from $\mathcal{A}(\mathcal{M}) \times 2^{\mathcal{M}}$ to $\mathcal{L}$, is called $\mathcal{C}$-safe interpretation function if the following conditions hold:*

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 $\mathcal{C}$-full-invariant by substitutions, i.e: for all M_1 and M_2 two set of messages in $2^{\mathcal{M}}$ such that $\forall \alpha \in \mathcal{A}(M_1) \cdot \ \mathsf{F}(\alpha, M_1) = \mathsf{F}(\alpha, M_2)$ we have for every $\sigma \in \Gamma$ that:*
   $$\forall \alpha \in \mathcal{A}(M_1) \cdot \ \mathsf{F}(\alpha, M_1\sigma) = \mathsf{F}(\alpha, M_2\sigma)$$
   where Γ is the set of possible substitution form $\mathcal{X}$ to close messages in $\mathcal{M}$.
3. *F is $\mathcal{C}$-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 $M \models_{\mathcal{C}} m$ we have for every α in $\mathcal{A}(m)$ that:
   $$(\mathsf{F}(\alpha, m) = \mathsf{F}(\alpha, M) \vee (\ulcorner K \urcorner \sqsupseteq \ulcorner \alpha \urcorner)$$

Let $\mathcal{C}_0 = \langle \mathcal{N}_0, \Sigma_0, \mathcal{E}_0, K_0, \mathcal{L}_0^{\sqsupseteq}, \ulcorner\cdot\urcorner_0 \rangle$ be the context of verification, where its elements are those given as example in section 2. As an example of a safe interpretation function, is the function F_0 that attributes to message a security level according to the keys that encrypt it and agents identities that are neighbors to it. For instance, $\mathsf{F}_0(k_{ab}, \{A, \{B, N_b, k_{ab}\}_{k_{as}}\}_{k_{bs}}) = \{B\} \cup \ulcorner k_{as} \urcorner_0 = \{A, S, B\}$. This function is called the DEKAN function and it is proved in (Houmani and Mejri, 2007a) to be $\mathcal{C}_0$-safe.

The sufficient conditions that states that agents of a protocols should not decrease or increase the security level of messages when they send them over the network, can be formalized as follows:

Definition 3 (Coherent Protocol). *Let $\mathcal{C} = \langle \mathcal{N}, \Sigma, \mathcal{E}, K, \mathcal{L}^{\sqsupseteq}, \ulcorner\cdot\urcorner \rangle$ be a verification context, F a $\mathcal{C}$-interpretation function and p a protocol. The protocol p is said to be F-coherent if:*

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

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 follows:

Theorem 4. *Let p be a protocol, $\mathcal{C}$ a verification context and F a $\mathcal{C}$-interpretation function . If F is $\mathcal{C}$-safe and p is F-coherent, then p is $\mathcal{C}$-correct with respect to the secrecy property.*

Proof.
The proof is almost similar to the ones in (Houmani and Mejri, 2007a).The intuition behind our proof is as follows: Since the protocols is F-coherent and F full-invariant by substitution, then the valid traces of the protocol, which are the interleaving of substituted

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

generalized roles of the protocol, are also F-coherent. This means that all the sent messages are encrypted with keys having an appropriate level of security. Furthermore, since F is invariant by intruder manipulation, then it follows that the intruder can never deduce, from appropriately protected messages, an inappropriately protected component. Hence, the intruder can never learn from the protocol what he is not eligible to know. □

5 CASES STUDY

According to the theorem 4, the first step of verification is to define a safe interpretation function. To that end, we consider the DEKAN function F_0 which is a safe interpretation function (see (Houmani and Mejri, 2007a)), that selects the direct encrypting keys and neighbors of a message and after that interprets the selected elements to deduce the security level of that message. For instance, we have:

$$\mathsf{F}_0(\alpha,\{A,B,\alpha\}_{k_{as}}) = \{A,B,S\}$$

By using the DEKAN safe interpretation function F_0 and the theorem 4, we will try to prove that p (the version of Woo and Lam protocol given by Table 2) is $\mathcal{C}_0$-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 3, we have:

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

For the role $\mathcal{A}_G^1$, since $\ulcorner A\urcorner = \bot$ and F_0 is well-defined ($\mathsf{F}(A,A)=\bot$), then the role $\mathcal{A}_G^1$ is F_0-coherent:

$$\mathsf{F}_0(A,A) = \ulcorner A\urcorner \cup \mathsf{F}_0(A,A)$$

For the role $\mathcal{A}_G^2$, since:

$$\begin{array}{rcl} \ulcorner s_{ab}^i\urcorner & = & \{A,B,S\} \\ \mathsf{F}_0(X_1,\{X_1,B,X_2\}_{k_{as}}) & = & \{A,B,S\} \\ \mathsf{F}_0(X_1,\{X_1,S,s_{ab}^i\}_{X_2}) & = & \{A,B,S\} \\ \mathsf{F}_0(s_{ab}^i,\{X_1,S,s_{ab}^i\}_{X_2}) & = & \{A,B,S\} \end{array}$$

then the role $\mathcal{A}_G^2$ is F_0-coherent. Indeed, we have:

$$\mathsf{F}_0(X_1,\{X_1,S,s_{ab}^i\}_{X_2}) = \ulcorner X_1\urcorner \cup \mathsf{F}_0(X_1,\{X_1,B,X_2\}_{k_{as}})$$
$$\mathsf{F}_0(s_{ab}^i,\{X_1,S,s_{ab}^i\}_{X_2}) = \ulcorner s_{ab}^i\urcorner \cup \mathsf{F}_0(s_{ab}^i,\{X_1,B,X_2\}_{k_{as}})$$

For the role $\mathcal{B}_G^1$, since:

$$\begin{array}{rcl} \ulcorner A\urcorner & = & \bot \\ \mathsf{F}_0(k_{ab}^i,\{N_b^i,A,k_{ab}^i\}_{k_{bs}}) & = & \{A,B,S\} \\ \mathsf{F}_0(N_b^i,\{N_b^i,A,k_{ab}^i\}_{k_{bs}}) & = & \{A,B,S\} \\ \ulcorner k_{ab}^i\urcorner & = & \ulcorner N_b^i\urcorner = \{A,B,S\} \end{array}$$

and since F_0 is well-defined ($\mathsf{F}_0(A,A)=\bot, \mathsf{F}_0(N_b^i,A)=\top$), then the role $\mathcal{B}_G^1$ is F_0-coherent. Indeed, we have:

$$\mathsf{F}_0(N_b^i,\{N_b^i,A,k_{ab}^i\}_{k_{bs}}) = \ulcorner N_b^i\urcorner \cup \mathsf{F}_0(N_b^i,A)$$

For the role $\mathcal{B}_G^2$, since:

$$\begin{array}{rcl} \ulcorner A\urcorner & = & \bot \\ \mathsf{F}_0(k_{ab}^i,\{N_b^i,A,k_{ab}^i\}_{k_{bs}}) & = & \{A,B,S\} \\ \mathsf{F}_0(N_b^i,\{N_b^i,A,k_{ab}^i\}_{k_{bs}}) & = & \{A,B,S\} \\ \ulcorner k_{ab}^i\urcorner & = & \ulcorner N_b^i\urcorner = \{A,B,S\} \end{array}$$

and since F_0 is well-defined ($\mathsf{F}_0(A,A)=\bot, \mathsf{F}_0(N_b^i,A)=\top$), then the role $\mathcal{B}_G^1$ is F_0-coherent. Indeed, we have:

$$\mathsf{F}_0(N_b^i,\{N_b^i,A,k_{ab}^i\}_{k_{bs}}) = \ulcorner N_b^i\urcorner \cup \mathsf{F}_0(N_b^i,A)$$

For the role $\mathcal{S}_G^1$, since:

$$\begin{array}{rcl} \mathsf{F}_0(U,\{U,A,V\}_{k_{bs}}) & = & \{A,B,S\} \\ \mathsf{F}_0(V,\{U,A,V\}_{k_{bs}}) & = & \{A,B,S\} \\ \mathsf{F}_0(U,\{U,B,V\}_{k_{as}}) & = & \{A,B,S\} \\ \mathsf{F}_0(V,\{U,V\}_{k_{as}}) & = & \{A,B,S\} \end{array}$$

and since $\ulcorner U\urcorner = \ulcorner V\urcorner = \top$, then the role $\mathcal{S}_G^1$ is F_0-coherent. Indeed, we have:

$$\mathsf{F}_0(U,\{U,B,V\}_{k_{as}}) = \ulcorner U\urcorner \cup \mathsf{F}_0(U,\{U,A,V\}_{k_{bs}})$$
$$\mathsf{F}_0(V,\{U,B,V\}_{k_{as}}) = \ulcorner V\urcorner \cup \mathsf{F}_0(V,\{U,A,V\}_{k_{bs}})$$

Therefore, this protocol is F_0-coherent and so $\mathcal{C}_0$-correct with respect the secrecy property.

6 CONCLUSIONS

In this paper, we have extended the result obtained in our previous works (Houmani and Mejri, 2007a; Houmani and Mejri, 2007b; Houmani and Mejri, 2008), to deal with protocols that use temporary keys (also called session keys). The main idea is that agents always include implicitly or explicitly within the messages the exact security levels of components so that this information can never be manipulated by an intruder.

According the main result of this paper, 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. That is why, we have proposed in (Houmani and Mejri, 2007a; Houmani and Mejri, 2007b; Houmani and Mejri, 2008) some guidelines allowing to define some examples of such functions. As a future work, we would like to extend this guideline to give more safe interpretation functions and so to handle more cryptographic protocols.

REFERENCES

Abadi, M. and Cortier, V. (2006). Deciding knowledge in security protocols under equational theories. *Theor. Comput. Sci.*, 367(1):2–32.

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.

Chevalier, Y., Ksters, R., Rusinowitch, M., and Turuani, M. (2003). An np decision procedure forprotocol insecurity with xor. In *LICS '03*, volume 25. IEEE Computer Society Press.

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). *New Decidability Results for Fragments of First-Order Logic and Application to Cryptographic Protocols*, volume 2706 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg.

Debbabi, M., Mejri, M., Tawbi, N., and Yahmadi, I. (1997). From Protocol Specifications to Flaws and Attack Scenarios: An Automatic and Formal Algorithm. In *Proceedings of the Second International Workshop on Enterprise Security, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA*. IEEE Press.

Goubault-Larrecq, J. (2005). Deciding h1 by resolution. *Inf. Process. Lett.*, 95(3):401–408.

Houmani, H. and Mejri, M. (2007a). Practical and universal interpretation functions for secrecy. In *International Conference on Security and Cryptography: Secrypt*, Barcelona, Spain.

Houmani, H. and Mejri, M. (2007b). Secrecy by interpretation functions. *Journal of Knowledge-Based Systems*, 20(7):617–635.

Houmani, H. and Mejri, M. (2008). Sufficient conditions for secrecy under equational theories. In *The 2nd International Conference on Information Security and Assurance*, Busan, Korea. IEEE CS.

Jacquemard, F., Rusinowitch, M., and Vigneron, L. (2000). Compiling and verifying security protocols. In *Logic Programming and Automated Reasoning*, pages 131–160.

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). Mechanized proofs for a recursive authentication protocol. In *10th Computer Security Foundations Workshop*, pages 84–95. IEEE Computer Society Press.

Sabelfeld, A. and Myers, A. (2003). Language-based information-flow security.

Shmatikov, V. (2004). *NP Decidable Analysis of Cryptographic Protocols with Products and Modular Exponentiation*, volume 2986 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg.

Turuani, M. (2003). *Scurit des protocoles cryptographiques : dcidabilit et complexit*. PhD thesis, Universit Henri Poincar, Nancy.

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