Ensuring Confidentiality in Cryptographic Protocols with the Witness-Functions

Jaouhar Fattahi[1]*, Mohamed Mejri[1], Moeiz Miraoui[2], Hanane Houmani[3]

[1] LSI, Computer Science and Software Engineering Department, Laval University, Adrien-Pouliot 1065, Avenue of Medicine, Quebec, G1V 0A6, Canada.
[2] LaTIS, Department of Electrical Engineering, School of Superior Technologies (T. S), 1100, Notre-Dame Ouest, Montreal, H3C 1K3, Canada.
[3] ENSEM, University Hassan II, Casablanca, Morocco.

* Corresponding author. Tel.: 1-581-888-3258; email: jaouhar.fattahi.1@ulaval.ca

Manuscript submitted August 04, 2014; accepted February 25, 2015.
doi: 10.17706/ijcce.2015.4.4.219-233

Abstract: In this paper, we present a new framework to verify cryptographic protocols statically for the property of confidentiality using the witness-functions. A witness-function is a reliable metric able to prove confidentiality in a cryptographic protocol by measuring security in it. Here, we present the theory of witness-functions and we run an analysis on the flawed version of the Woo-Lam protocol using one of these metrics.

Key words: Confidentiality, cryptographic protocols, proof, witness-function.

1. Introduction

Cryptographic protocols are short programs intended to ensure some security properties in communications using cryptographic primitives. Designing a secure protocol is a hard problem [1]-[8]. In fact, many security bugs have been uncovered in lots protocols too many years after of their use [9]-[14]. The man-in-the-middle attack is one of them. It is therefore crucial to obtain a formal token that a protocol is secure for the properties that it must ensure. For that, many verification methods had been introduced to analyze security inside protocols [15]-[27]. In this paper we pose a new static framework for measuring security in protocols that we instilled recently in [28]-[32] using the witness-functions. A witness-function is a mathematical function built locally in the protocol. It could be used to observe the evolution of every atomic message in order to verify that its level of security never falls down in all protocol steps. If a protocol is designed that way, we can conclude that it is secure with respect to confidentiality. A witness-function bases its calculation fully on the static part of a message (static neighborhood) in a role-based specification [26], [33], [34] and ignores the dynamic one (dynamic neighborhood) by construction. It provides two elegant and practical bounds that enable to analyze a protocol on an unbounded number of sessions and with no restriction on the size of valid traces. Here we give the theoretical foundation of the witness-functions and we run an analysis on the flawed version of the Woo-Lam protocol using one of them. We show that it can help to describe the flaw in it.

2. Confidentially Inside Increasing Protocols

To analyze a protocol, we need verification functions to reasonably assess the level of security of all atomic messages. Indeed, if a verification function mistakes the level of security of a single atomic message, this may lead to bad conclusions regarding the protocol security. We call a good verification function a reliable function. If the level of security of every atomic message in the protocol, assessed by a reliable verification functions, never drops, we say that the protocol is increasing. Here, we give general sufficient conditions [28] for a verification function to be reliable. Then, we state that an increasing protocol is secure with respect to confidentiality.

2.1. Reliable Verification Functions

Definition 2.1, [Well-Formed verification Function]:

Let φ be a verification function and C be a context of verification, φ is well–formed in C if:

$$\forall M, M1, M2 \subseteq \mathcal{M}, \forall \alpha \in \mathrm{A}(\mathcal{M}) \begin{cases} \varphi(\alpha, \{\alpha\}) = \bot \\ \varphi(\alpha, M_1 \cup M_2) = \varphi(\alpha, M_1) \sqcap \varphi(\alpha, M_2) \\ \varphi(\alpha, M) = \top, \text{if } \alpha \notin \mathcal{A}(M) \end{cases}$$

A well-formed verification Function φ should return the infimum "$\bot$" for an atom α that appears in clear in M. It returns for it in the union of two sets, the minimum "$\sqcap$" of the two values assessed in each set separately. It returns the supremum "$\top$" for any atom α that does appear in M.

Definition 2.2, [Stable-by-Intruder Verification Function]:

Let φ be a verification function and C be a context of verification, φ is stable-by intruder in C if:

$$\forall\ M \subseteq \mathcal{M}, m \in \mathcal{M}. M \vDash c\ m \Longrightarrow \forall \alpha \in \mathcal{A}(m). \big(\varphi(\alpha, m) \ \sqsupseteq \varphi(\alpha, M)\big) \vee (\lceil K \rceil(\mathrm{I}) \ \sqsupseteq \lceil \alpha \rceil)$$

A Stable-by-Intruder verification Function ' is such that, when it assigns a security level to an atom _ in a set of messages M the intruder can never derive, using her knowledge, from M another message m in which this level drops (i.e. $\varphi(\alpha, m) \not\sqsupseteq \varphi(\alpha, M)$), unless α is initially intended to the intruder (i.e. $\lceil K \rceil(I) \ \sqsupseteq \lceil \alpha \rceil$).

Definition 2.3, [Reliable Verification Function]:

Let φ be a verification function and C be a context of verification.

$$\varphi \text{ is } C\text{-reliable, if } \begin{cases} \varphi \text{ is well formed in } C \\ \varphi \text{ is stabled by intruder in } C \end{cases}$$

A reliable verification function φ is stable-by-intruder and well-formed.

Definition 2.4, [φ-Increasing Protocol]:

Let φ be a verification function, C be a context of verification and p be a protocol, p is φ- Increasing in C if $\forall R.\ r \in R_{G(p)} \forall \sigma \in \Gamma : X \longrightarrow \mathcal{M}_p$ we have:

$$\forall \alpha \in \mathcal{A}(\mathcal{M}).\ \varphi(\alpha, r^{+}\sigma) \sqsupseteq \lceil \alpha \rceil \ \sqcap\ \varphi(\alpha, R^{-}\sigma)$$

A φ-Increasing protocol is a protocol that produces immutably valid traces with atoms having all the time a level of security, assessed by φ, higher on sending (i.e. in $r^{+}\sigma$), than it was on reception (i.e. in $R^{-}\sigma$).

Definition 2.5, [Secret Divulgation]:

Let p be a protocol and C be a context of verification. We say that p reveals a secret $\alpha \in \mathcal{A}(\mathcal{M})$ in C if:

$$\exists p \in [\![p]\!]. (p \vDash c\ \alpha) \wedge (\lceil K(I) \rceil \not\sqsupseteq \lceil \alpha \rceil)$$

A secret divulgation is the fact that the intruder manipulates a valid trace ϱ, using her capacity, to derive a secret α that she should not know (i.e. $\lceil K(I) \rceil \not\sqsupseteq \lceil \alpha \rceil$).

Proposition 2.6:

Let φ be a C-Reliable verification function and pa φ-Increasing Protocol. We have:

$$8\text{m } \forall m \in \mathcal{M}.p \vDash c\, m \Rightarrow \forall \alpha \in \mathcal{A}(m).(\varphi(\alpha, m) \sqsupseteq \alpha) \vee (K(I) \sqsupseteq \alpha$$

The Proposition 2.6 asserts that the level of security of an atom α in a message m generated by an increasing protocol and assessed by a reliable verification function φ is maintained higher than its initial value in the context, if the atom is not initially destined to the intruder. Indeed, initially the atom has a given level of security. This level cannot be decreased by the intruder using her capacity and received messages because a reliable function is Stable-by-Intruder. Besides, in every new step of the evolution of the valid trace, involved atoms have higher level of security assessed by φ since the protocol is φ-increasing. The proof is then directed by induction on the size of the trace.

Theorem 2.7, [Security of Increasing Protocols]:

Let φ be a C-Reliable verification Function and p a φ-increasing protocol.

p is C-secure with respect to confidentiality.

The Theorem 2.7 states that an increasing protocol is secure with respect to confidentiality when analyzed with a reliable function φ. The proof of the Theorem 2.7 derives directly from the Proposition 6 and the Definitions 2.1 and 2.5. For further details on the proof, please see [28]-[32].

3. Reliable Verification Functions

We define in [29], [32] a class S_{Gen}^{EK} of reliable selection-based verification functions. A selection $S \in S_{Gen}^{EK}$ returns for an atom α in a message m:

1) If α is encrypted by a key k, where k is the most external key that satisfies to $\lceil k^{-1} \rceil \sqsupseteq \lceil \alpha \rceil$(or simply the external protective key), a subset among the reverse form k^{-1} and atoms that travel with α under the same protection by k (α itself is not selected);
2) For two messages joined by an operation other than an encryption by a protective key (e.g. pair), the union of two subselections in the two messages separately;
3) If α does not have a protective key in m, the infimum value (all atoms);
4) If α does not appear in m, the supremum value (the empty set).

We prove that any $S \in S_{Gen}^{EK}$ is C-reliable.

Among the elements of S_{Gen}^{EK}, we define three practical selections:

1) The selection S_{MAX}^{EK} : returns for an atom α in a message having the key k as an external protective, all the principal identities inside the same protection by k, in addition to the key k^{-1};
2) The selection S_{EK}^{EK} : returns for an atomαin a message m having as an external protective key k, the key k^{-1};
3) The selection S_{N}^{EK}: returns for an atom α in a message m having as an external protective key k, all the principal identities inside the same protection by k;

These selections when composed to a proper homomorphism ψ render reliable verification functions $\varphi = \psi\, o\, S$.

We choose the homomorphism that returns for:

1) A principal, its identity;
2) The key k^{-1}, if selected, the set of principals that know it in the context.

We denote by φ_{MAK}^{EK}, φ_{EK}^{EK} and φ_{N}^{EK} respectively the compositions $\psi\, o\, S_{MAX}^{EK}$, $\psi\, o\, S_{EK}^{EK}$ and $\psi\, o\, S_{N}^{EK}$. We prove that these functions are C-reliable. The main idea of the proof is that the selection for protected atoms (secrets) is always performed inside an invariant area protected by the external protective keyk. Hence, to alter this area (to decrease the level of security of an atom α), the intruder should have derived the atomic key k^{-1} before. In this case, her knowledge should necessarily satisfy $\lceil K(I) \rceil \sqsupseteq \lceil k^{-1} \rceil$. Since the key k^{-1} satisfies $\lceil k^{-1} \rceil \sqsupseteq \lceil \alpha \rceil$, then the knowledge of the intruder must satisfy $\lceil K(I) \rceil \sqsupseteq \lceil \alpha \rceil$ too by transitivity of the relation "$\sqsupseteq$" in the lattice, which is the definition of a Stable-by-Intruder function. These functions are in addition Well-Formed by construction. So, they are C-reliable. For further details on the proof, please see [30]-[32]. In the rest of this paper, φ denotes any of the functions φ_{MAK}^{EK}, φ_{EK}^{EK} and φ_{N}^{EK}.

Example:

Let α be an atom, m a message and k_{ab}a key such that $\lceil \alpha \rceil$ ={A, B, S}, m= {A. C. { $\lceil \alpha \rceil . D$} $k_{as}\}k_{ab}$; $k_{\mathrm{ab}}^{-1} = k_{ab}, k_{\mathrm{as}}^{-1} = k_{as;}$ $\lceil k_{as} \rceil$={A, S}, $\lceil k_{ab} \rceil = \{A, B\}$;

$$S_{EK}^{EK}(\alpha, m) = \{k_{ab}^{-1}\};\ S_{N}^{EK}(\alpha, m) = \{A, C, D\};\ S_{MAX}^{EK}(\alpha, m) = \{A, C, D, k_{ab}^{-1}\};$$

$$\varphi_{EK}^{EK}(\alpha, m) = \psi\ o\ S_{EK}^{EK}(\alpha, m) = k_{ab} = \{A, B\}; \varphi_{N}^{EK}(\alpha, m) = \ \psi\ o\ S_{N}^{EK}(\alpha, m) = \{A, C, D\};\ \varphi_{MAX}^{EK}(\alpha, m)$$
$$= \ \psi\ o\ S_{MAX}^{EK}(\alpha, m) = \{A, C, D\} \sqcap \left\lceil k_{ab}^{-1} \right\rceil = \{A, C, D\} \cup \{A, B\} = \{A, C, D, B\}$$

4. The Witness-Functions

The verification functions defined previously are good to verify a protocol through its valid traces (closed messages). Unfortunately, the set of valid traces is infinite. A static verification should be run on the finite set of generalized roles. But, the generalized roles contain variables and our functions are not ready to deal with this problem. Here, we propose a safe way to use these functions on generalized roles. First, in order to reduce variable effects, we introduce the notion of derivative messages that are messages from which we remove variables as described in the Definition 4.1.

Definition 4.1, [Derivation]:

We define the derivative message as follows:

$$\partial X\ \alpha = \alpha$$
$$\partial X\ \epsilon = \epsilon$$
$$\partial X\ X = \epsilon$$
$$\partial X\ Y = Y$$
$$\partial \{X\}\ m = \partial x^{m}$$
$$\partial [\overline{X}]^{m} = \partial \{_{x_m} \backslash X\}^{m}$$
$$\partial_X f(m) = f(\partial X\ m), f \in\ \Sigma$$
$$\partial_{s_1 \cup s_2} m = \ \partial_{s_1} \partial_{s_2} m$$
$$\partial_{s_1 \cup s_2} m = \partial_{s_2 \cup s_1} m$$

Then, we apply any of the previous functions φ to derivative message instead of the message with variables. For an atom of the static part (i.e. in ∂m) we analyze it with no regard to variables at all. For a variable, it is analyzed as a constant block with no regard on its content and with no respect to other variables in the message, if any. Hence, every component is calculated according to the static neighborhood only. This is described by the Definition 4.2. For any secret, a reliable function φ applied to derivative messages preserves its property of reliability since its associative selection might just ignore some candidates (principal identities that come dynamically by variable substitution), but remains a sub-selection of S_{Gen}^{EK}, so reliable

Definition 4.2:

Let $m \in \mathcal{M}_P^{\mathcal{G}}$, $X \in \mathcal{X}_m$ and $m\sigma$ be a valid trace.

For all $\alpha \in \mathcal{A}(m\sigma)$, $\sigma \in \Gamma$, we denote by:

$$\varphi(\alpha, \partial[\overline{\alpha}]m\sigma) = \begin{cases} \varphi(\alpha, \partial m) \text{if } \alpha \in A(\partial m), \\ \varphi(X, \partial[\overline{X}]m) \text{ if } \alpha \notin A(\partial m) \text{ and } \alpha = X\sigma. \end{cases}$$

Example 4. 3:

Let m_1 and m_2 be two messages of a generalized role of a protocol p such that m_1 = $\{\alpha.C.X\}k_{ab}$ and m_2 = $\{\alpha.Y\}\,k_{ab}$ and $\lceil\alpha\rceil$= $\{A, B, S\}$ and $\lceil k_{ab}^{-1} \rceil$ = $\{A, B, S\}$. Let m = $\{\alpha.C.B\}\,k_{ab}$ be in a valid trace generated by p.

$$\varphi_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m) = \begin{cases} \{A, B, S, C\} \text{ if } m = m_1\sigma_1 | X\sigma_1 = B \\ \{A, B, S\} \text{ if } m = m_2\sigma_2 | Y\sigma_2 = C.B \end{cases}$$

Therefore, $\varphi_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m)$ is not even a function on m since it could return more than one image for the same atom α.

To solve this problem, we define the witness-functions. A witness-function as described in the Definition 4.4 considers all the sources of a closed message $m\sigma$ in the finite set of generalized roles M_p^g and takes the minimum. This minimum naturally exists and is unique in a lattice.

Definition 4.4, [Witness-Function]:

Let $m \in M_p^{\mathcal{G}}$, $X \in \mathcal{X}_m$ and $m\sigma$ be a valid trace. Let p be a protocol and φ be a C-reliable verification Function. We define a witness-function $\Phi p\,,\varphi$ for all $\alpha \in A(m\sigma)$, $\sigma \in \Gamma$, as follows:

$$\Phi p\,, \varphi(\alpha, m\sigma) = \sqcap_{\substack{m' \in \mathcal{M}_P^{\mathcal{G}} \\ \exists\sigma' \in \Gamma, m'\sigma' = m\sigma}} \varphi(\alpha, \partial[\overline{\alpha}]m'\sigma')$$

We notice that a witness-function depends on the set of sources of $m\sigma$, (i.e. the set$\{m' \in \mathcal{M}_P^{\mathcal{G}} | \exists\sigma' \in \Gamma, m'\sigma' = m\sigma\}$). So it depends on the substitution σ that is not known statically. For that, we bind the witness-function in two bounds that are independent of any substitution in order to be able to verify a protocol statically, with no regard to its runs. These bounds are given by the Proposition 4.5.

Proposition 4.5:

Let $m \in \mathcal{M}_P^{\mathcal{G}}$, let ϕ_p, φ be a witness-function. For all $\sigma \in \Gamma$ we have:

$$\varphi(\alpha, \partial[\overline{\alpha}]m) \sqsupseteq \Phi p\,, \varphi(\alpha, \mathrm{m\sigma}) \sqsupseteq \cup_{\substack{m' \in \mathcal{M}_P^{\mathcal{G}} \\ \exists\sigma' \in \Gamma, m'\sigma' = m\sigma'}} \varphi(\alpha, \partial[\overline{\alpha}]m'\sigma')$$

The proof of the Proposition 4.5 is trivial since we have always $m \in \{m' \in \mathcal{M}_P^{\mathcal{G}} | \exists\sigma' \in \Gamma, m'\sigma' = m\sigma\}$ and $\{m' \in \mathcal{M}_P^{\mathcal{G}} | \exists\sigma' \in \Gamma, m'\sigma' = m\sigma\} \in \{m' \in \mathcal{M}_P^{\mathcal{G}} | \exists\sigma' \in \Gamma, m'\sigma' = m\sigma'\}$. The upper bound (the tighter) provides a minimal set of confirmed principals whereas the lower bound (the looser) provides the set of all possible principals in all the likely runs of a protocol. This latter contains, in addition to the honest principals, all possible intrusions, if any. It is an intrusion trap.

Theorem 4.6, [Protocol Analysis Theorem]:

Let p be a protocol. Let $\phi_{p,\varphi}$ be a witness-function. p is secure for the property of confidentiality if: $\forall$ R, r $\in R_G$ $(\mathcal{P})$, $\forall_\alpha \in (r^+)$ we have:

$$\begin{gathered} \sqcap\ \varphi(\alpha, \partial[\overline{\alpha}]m'\sigma\,') \sqsupseteq \lceil a \rceil \sqcap \varphi(\alpha, \partial[\overline{\alpha}]R^-) \\ m' \in \mathcal{M}_P^{\mathcal{G}} \\ \exists \sigma\,' \in \Gamma, m'\sigma\,' = r^+\sigma\,' \end{gathered}$$

The proof derives directly from the Proposition 4.5 and the Theorem 2.7. The Theorem 4.6 sets a criterion for protocol growth, so correctness with a witness-function using its bounds. Since these bounds do not depend on substitutions, then we can now, and only now, verify a protocol through its set of generalized roles and extend any conclusion to the traces.

Although, the application given in the Definition 4.2 is naturally independent of any substitution in m, using it to verify protocols is an error-prone process. Let us see that in the following example.

5. Case Study: Verification of the Woo-Lam Protocol (Flawed Version) Using a Witness-Function

In this section, we run an analysis of the flawed version of the Woo-Lam protocol using the witness-function $\phi_p, \varphi_{MAK}^{EK}$. The flawed version of the Woo-Lam protocol is denoted by p and defined in Table 1.

Table 1. The Woo-Lam Protocol

$$\begin{array}{l} p = \langle 1, A \ \to\ B : \ A \rangle. \\ \quad\ \ \langle 2, B \ \to\ A : \ N_b \rangle. \\ \quad\ \ \langle 3, A \ \to\ B : \ \{N_b.k_{ab}\}\, k_{as} \rangle \\ \quad\ \ \langle 4, B \ \to\ S : \ \{A.\{\, N_b.k_{ab}\}\, k_{as}\}k_{bs} \rangle \\ \quad\ \ \langle 5, S \ \to\ B : \ \{N_b.k_{ab}\ \}k_{bs} \rangle \end{array}$$

The role-based specification of p is $R_G\ (p)$, = $\{\mathcal{A}_{G,}^1 \mathcal{A}_{G,}^2 \mathcal{B}_{G,}^1 \mathcal{B}_{G,}^2 \mathcal{B}_{G,}^3 \mathcal{S}_G^1\}$, where the generalized roles $\mathcal{A}_{G,}^1 \mathcal{A}_{G,}^2$, of A are as follows:

$$\mathcal{A}_G^1 = \langle i.1, A \ \to I(B) : A \rangle$$

$$\begin{gathered} \mathcal{A}_G^2 \ = \langle i.1, A \ \to I(B) : A \rangle \\ \langle i.2, I(B) \ \to\ A : X \rangle \\ \langle i.3, A \ \ I(B) \to\ I(B) : \{X.k_{ab}^i\}k_{as} \ \rangle \end{gathered}$$

The generalized roles $\mathcal{B}_{G,}^1 \mathcal{B}_{G,}^2 \mathcal{B}_{G,}^3$ are as follows:

$$\begin{gathered} \mathcal{B}_G^1 \ = \langle i.1, I(A) \ \to B : A \rangle \\ \langle i.2, B \ \to\ I(A) : N_b^i \rangle \end{gathered}$$

$$\begin{gathered} \mathcal{B}_G^2 \ \ = \langle i.1, A \ \to B : A \rangle \\ \langle i.2, B \ \to\ I(A) : N_b^i \rangle \\ \langle i.3, I(A) \ \to\ B : Y \rangle \end{gathered}$$

$$\langle i.4, B \rightarrow I(S) : \{A.Y\}k_{bs}\rangle$$

$$\begin{gathered}\mathcal{B}_G^3 = \langle i.1, A \rightarrow B : A\rangle \\ \langle i.2, B \rightarrow I(A): N_b^i\rangle \\ \langle i.3, I(A) \rightarrow B: Y\rangle \\ \langle i.4, B \rightarrow I(S): \{A.Y\}k_{bs}\rangle \\ \langle i.5, I(S) \rightarrow B : \{N_b^i.Z\}k_{bs}\rangle\end{gathered}$$

The generalized roles $\mathcal{S}_G^1$ are as follows:

$$\begin{gathered}\mathcal{S}_G^1 = \langle i.4, B \rightarrow \{A:\{U.V\}k_{as}\}k_{bs}\rangle \\ \langle i.5, S \rightarrow I(B) : \{U.V\}k_{bs}\rangle\end{gathered}$$

Let us have a context of verification such that: $\lceil k_{as}\rceil = \{A, S\}$;$\lceil k_{bs}\rceil = \{B, S\}$; $\lceil k_b^i\rceil = \{A, B, S\}$; $\lceil N_b^i\rceil = \bot$; $\forall A \in$ I, $\lceil A\rceil = \bot$.

The principal identities are not verified since they are set public in the context.

Let $\varphi = \varphi_{MAX;}^{EK}\, \Phi_{p,\varphi} = \Phi_{p,\varphi_{MAX}^{EK}}$;

We denote by $\Phi'_{p,\varphi}(a,m)$ the lower bound: $\sqcap\ \varphi(\alpha, \partial[\overline{\alpha}]m'\sigma')$ of the witness-function $\Phi_{p,\varphi}(a,m)$

$$\begin{gathered}m' \in \mathcal{M}_P^{\mathcal{G}} \\ \exists\sigma' \in \Gamma, m'\sigma' = m\sigma'\end{gathered}$$

Let

$$M_{\mathcal{P}}^{\mathcal{G}} = \{A_1, X_1, \{X_2.k_{A_2B_1}^i\}k_{A_2S_1}, A_3, N_{B_2}^i, Y_1, \{A_4.Y_1\}k_{B_3S_2}, \{N_{B_4}^i.Z_1\}k_{B_4S_3}, \{A_5.\{U_1.V_1\}k_{A_5S_4}\}k_{B_5S_4}, \{U_2.V_2\}k_{B_6S_5}\}$$

After elimination of duplicates $M_{\mathcal{P}}^{\mathcal{G}} = \{A_1, X_1, \{X_2.k_{A_2B_1}^i\}K_{A_2S_1}, N_{B_2}^i, Y_1, \{A_4.Y_1\}K_{B_3S_2}, \{N_{B_4}^i.Z_1\}K_{B_4S_3}, \{A_5.\{U_1.V_1\}k_{A_5S_4}\}K_{B_5S_4}, \{U_2.V_2\}K_{B_6S_5}\}$

The variables are denoted by $X_1, X_2\ Y_1, Z_1, U_1, U_2, V_1$ and V_2.

The static names are denoted by A_1, A_2, B_1, $K_{A_2S_1}$, $N_{B_2}^i$, A_4, $K_{B_3S_2}$, $N_{B_4}^i$, $K_{B_4S_3}$, A_5, $K_{A_5S_4}$, $K_{B_5S_4}$ and $K_{B_6S_5}$.

5.1. Verification of the Generalized Role of *A*

As defined in the generalized role of A, an agent A can participate in some session S^i in which she receives an unknown message X and sends the message $\{X.k_{ab}^i\}k_{as}$. This is described by the following rule:

$$S^i : \frac{X}{\{X.k_{ab}^i\}k_{as}}$$

1) For k_{ab}^i:

- When receiving: $R_{S^i}^- = X$ (on receiving, we use the upper bound)

$$\varphi\left(k_{ab}^i, \partial\left[k_{ab}^i\right]X\right) = \top(1.0)$$

- When sending: $r_{S^i}^+ = \{X.k_{ab}^i\}k_{as}$, (on sending, we use the lower bound)

$$k_{ab}^i.\{m' \in \mathrm{M}_{\mathcal{P}}^{\mathcal{G}} \,|\, \exists \sigma' \in \Gamma, m'\sigma' = r_{S^i}^{+}\sigma'\} = \forall k_{ab}^i.\{m' \in \mathrm{M}_{\mathcal{P}}^{\mathcal{G}} \,|\, \exists \sigma' \in \Gamma, m'\sigma' = \{X.k_{ab}^i\}k_{as}\,\sigma'\}$$
$$= \{(\{X_2.k_{A_2B_1}^i\}k_{A_2S_1}, \sigma_1'), (\{A_4.Y_1\}k_{B_3S_2}, \sigma_2'), (\{N_{B_4}^i.Z_1\}k_{B_4S_3}, \sigma_3'), (\{U_2.V_2\}k_{B_6S_5}, \sigma_4')\}$$

Such that:

$$\begin{cases} \sigma_1' = \{X_2 \longmapsto X, k_{A_2B_1} \longmapsto k_{ab}, k_{A_2S_1} \longmapsto k_{as}\} \\ \sigma_2' = \{X \longmapsto A_4, Y_1 \longmapsto k_{ab}^i, k_{A_2S_1} \longmapsto k_{as}\} \\ \sigma_3' = \{X \longmapsto N_{B_4}^i, Z_1 \longmapsto k_{ab}^i, k_{B_4S_3} \longmapsto k_{as}\} \\ \sigma_4' = \{U_2 \longmapsto X, V_2 \longmapsto k_{ab}^i, k_{B_6S_5} \longmapsto k_{as}\} \end{cases}$$

$$\Phi'_{p,\varphi}(k_{ab}^i, \{X.k_{ab}^i\}k_{as}) = \{\text{Definition of the lower bound of the Witness-Function}\}$$

$$\varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{X_2.k_{A_2B_1}^i\}k_{A_2S_1}\sigma_1'\right) \sqcap \varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{A_4.Y_1\}k_{B_3S_2}\,\sigma_2'\right) \sqcap \varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{N_{B_4}^i.Z_1\}k_{B_4S_3}\sigma_3'\right) \sqcap$$
$$\varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{U_2.V_2\}k_{B_6S_5}\sigma_4'\right) = \{\text{Renaming the static neighborhood}\}$$

$$\varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{X_2.k_{ab}^i\}k_{as}\sigma_1'\right) \sqcap \varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{A_4.Y_1\}k_{as}\sigma_2'\right) \sqcap \varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{N_{B_4}^i.Z_1\}k_{as}\sigma_3'\right) \sqcap$$
$$\varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{U_2.V_2\}k_{as}\sigma_4'\right) = \{\text{Definition 4.2}\}$$

$$\varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]\{X_2.k_{ab}^i\}k_{as}\right) \sqcap \varphi(Y_1, \partial[\overline{Y_1}]\{A_4.Y_1\}k_{as}) \sqcap \varphi(Z_1, \partial[Z_1]\{N_{B_4}^i.Z_1\}k_{as}) \sqcap$$
$$\varphi(V_2, \partial[\overline{V_2}]\{U_2.V_2\}k_{as}) = \{\text{Derivation in the Definition 4.1}\}$$

$$\varphi(k_{ab}^i, \{k_{ab}^i\}k_{as}) \sqcap \varphi(Y_1, \{A_4.Y_1\}k_{as}) \sqcap \varphi(Z_1, \{N_{B_4}^i.Z_1\}k_{as}) \sqcap \varphi(V_2, \{V_2\}k_{as}) = \{\text{Since } \varphi = \varphi_{MAX}^{EX}\}$$

$$\{A, S\} \cup \{A_4, A, S\} \cup \{A, S\} \cup \{A, S\} = \{A, S, A_4\} \tag{5.1.1}$$

2) ∀ X:

Since when receiving, we have $\varphi(X, \partial\{\overline{X}\}X) = \varphi(X, X) = \bot$, then we derive:

$$\Phi'_{p,\varphi}(X, \{X.k_{ab}^i\}k_{as}) \sqsupseteq \lceil X \rceil \sqcap \varphi(X, \partial\{\overline{X}\}X) = \bot \tag{5.1.2}$$

3) Compliance with the security criterion in the theorem 4.6:

From (5.1.1) and (5.1.2), we have

$$\Phi'_{p,\varphi}(k_{ab}^i, \{X.k_{ab}^i\}k_{as}) = \{A, S, A_4\} \not\sqsupseteq \lceil k_{ab}^i \rceil \sqcap \varphi\left(k_{ab}^i, \partial\left[\overline{k_{ab}^i}\right]X\right) = \{S, A, B, \} \tag{5.1.3}$$

From (5.1.3), we have: the messages of the session S^i $(\text{i.e.}\, k_{ab}^i)$ are not compliant with the security criterion in the Theorem 4.6. (I).

5.2. Verification of the Generalized Role of B

As defined in the generalized roles of B, an agent B can participate in two consequent sessions: S^i and S^j such that *j > i*. In the former session S^i, the agent B receives the identity A and sends the nonce N_b^i. In the consequent session S^j, she receives an unknown message Y and she sends the message $\{A.N_b^i.Y\}k_{bs}$.

This is described by the following rules:

$$S^i : \frac{X}{N_b^i} \qquad S^j : \frac{Y}{\{A.Y\}k_{bs}}$$

5.2.1. Verification of the messages in the session S^i:

1) For N_b^i:

Since N_b^i is declared public in the context $\left(i.e.\ \lceil N_b^i \rceil = \bot\right)$, then we derive:

$$\Phi'_{p,\varphi}\left(N_b^i, N_b^i\right) \sqsupseteq \lceil N_b^i \rceil \sqcap \varphi\left(N_b^i, \partial\left[\overline{N_b^i}\right] A\right) = \bot \tag{5.2.1}$$

5.2.2. Verification of the messages in the session S^j:

1) $\forall Y$:

Since when receiving, we have: $\varphi\left(Y, \partial\left[\overline{Y}\right]Y\right) = \varphi(Y.Y) = \bot$, then we derive:

$$\Phi'_{p,\varphi}\left(Y, \{A.\mathrm{N}_b^i.Y\}k_{bs}\right) \sqsupseteq \lceil Y \rceil \sqcap \varphi\left(Y, \partial\left[\overline{Y}\right]Y\right) = \bot \tag{5.2.2}$$

2) Compliance with the security criterion in the Theorem 4.6:

From (5.2.1) and (5.2.2) we have: the messages of the session S^i and S^j respect the security criterion in the Theorem 4.6. (II)

5.3. Verification of the Generalized Role of S

As defined in the generalized role S, an agent S can participate in some session S^i in which she receives the message $\{A.\{U.V\}k_{as}\}k_{bs}$ and sends the message $\{U.V\}k_{bs}$. This is described by the following rule:

$$\mathrm{S}^i : \frac{\{A.\{U.V\}k_{as}\}k_{bs}}{\{U.V\}k_{bs}}$$

1) $\forall U$:

- When receiving: $R_{S^i}^- = \{A.\{U.V\}k_{as}\}k_{bs}$ (on receiving, we use the upper bound)

$$\begin{array}{ll} & \varphi\left(U, \partial\left[\overline{U}\right]\{A.\{U.V\}k_{as}\}k_{bs}\right) = \varphi(U, \{A.\{U\}k_{as}\}k_{bs}) = \\ \begin{cases} \{A.B.S\} \\ \{A.S\} \end{cases} & \begin{array}{l} \text{if } k_{bs} \text{ is the external protective of } V \text{ in } \{A.\{U.V\}k_{as}\}k_{bs} \text{ (5.3.1.1)} \\ \text{if } k_{as} \text{ is the external protective of } V \text{ in } \{A.\{U.V\}k_{as}\}k_{bs} \text{ (5.3.1.2)} \end{array} \end{array} \tag{5.3.1}$$

- When sending: $r_{S^i}^+ = \{U.V\}k_{bs}$ (on sending, we use the lower bound)

$$\begin{aligned} \forall U.\left\{m' \in M_{\mathcal{P}}^{\mathcal{G}} \middle| \exists\sigma' \in \Gamma, m'\sigma' = r_{S^i}^+\sigma'\right\} &= \forall \mathrm{U}.\left\{m' \in M_{\mathcal{P}}^{\mathcal{G}} \middle| \exists\sigma' \in \Gamma, m'\sigma' = \{U.V\}k_{bs}\sigma'\right\} \\ &= \left\{\{X_2.k_{A_2B_1}^i\}k_{A_2S_1}, \sigma'_1), (\{U_2.V_2\}k_{B_6S_5}, \sigma'_2)\right\} \end{aligned}$$

Such that:

$$\begin{cases} \sigma'_1 = \left\{X_2 \longmapsto U, V \longmapsto k_{A_2B_1}^i, k_{A_2S_1} \longmapsto k_{bs}\right\} \\ \sigma'_2 = \left\{U_2 \longmapsto U, V_2 \longmapsto V, k_{B_6S_5} \longmapsto k_{bs}\right\} \end{cases}$$

$$\Phi'_{p,\varphi}(U,\{U.V\}k_{bs})\text{= \{definition of the lower bound of the witness-function\}}$$

$$\varphi\big(U,\partial[\overline{U}]\{X_2.k^i_{A_2B_1}\}k_{A_2S_1}\sigma'_1\big)\sqcap\varphi\big(U,\partial[\overline{U}]\,\{U_2\,.V_2\}k_{B_6S_5}\sigma'_2\big)\text{= \{Renaming the static neighborhood\}}$$

$$\varphi\big(U,\partial[\overline{U}]\{X_2.k^i_{A_2B_1}\}k_{bs}\sigma'_1\big)\sqcap\varphi\big(U,\partial[\overline{U}]\,\{U_2\,.V_2\}k_{bs}\sigma'_2\big)\text{= \{Definition 4.2\}}$$

$$\varphi\big(X_2,\partial[\overline{X_2}]\{X_2.k^i_{A_2B_1}\}k_{bs}\sigma'_1\big)\sqcap\varphi\big(U_2\,,\partial[\overline{U_2}]\,\{U_2\,.V_2\}k_{bs}\sigma'_2\big)\text{= \{Derivation in the Definition 4.1\}}$$

$$\varphi\big(X_2,\{X_2.k^i_{A_2B_1}\}k_{bs}\big)\sqcap\varphi(U_2\,,\{U_2\,\}k_{bs})\text{= \{Since }\varphi\text{=}\varphi^{\text{EX}}_{\text{MAX}}\text{\}}$$

$$\{B,S\}\cup\{B,S\}=\{B,S\}\qquad(5.3.2)$$

2) $\forall V$:

- When receiving: $R^-_{S^i}=\{\mathrm{A}.\{U.V\}k_{as}\}k_{bs}$ (on receiving, we use the upper bound)

$$\varphi\big(V,\partial[\overline{V}]\{A.\{U.V\}k_{as}\}k_{bs}\big)=\varphi(V,\{A.\{V\}k_{as}\}k_{bs})=$$
$$\begin{cases}\{A,B,S\} & \text{if } k_{bs} \text{ is the external protective key of } V \text{ in } \{A.\{U.V\}k_{as}\}k_{bs}\ (5.3.3.1)\\ \{A.S\} & \text{if } k_{as} \text{ is the external protective key of } V \text{ in } \{A.\{U.V\}k_{as}\}k_{bs}\ (5.3.3.2)\end{cases}\qquad(5.3.3)$$

- When sending: $r^+_{S^i}=\{U.V\}k_{bs}$ (on sending, we use the lower bound)

$$\forall\mathrm{V}.\{m'\in\mathrm{M}^{\mathcal{G}}_{\mathcal{P}}\,|\exists\sigma'\in\Gamma.m'\sigma'=r^+_{S^i}\sigma'\}$$
$$=\forall\mathrm{V}.\{m'\in\mathrm{M}^{\mathcal{G}}_{\mathcal{P}}\,|\exists\sigma'\in\Gamma.m'\sigma'=\{U.V\}k_{bs}\sigma'\}$$
$$=\Big\{\{A_4.Y_1\}k_{B_3S_2},\sigma'_1),\big(\{N^i_{B_4}.Z_1\}k_{B_4S_3},\sigma'_2\big),\big(\{U_2\,.V_2\}k_{B_6S_5},\sigma'_3\big)\Big\}$$

Such that:

$$\begin{cases}\sigma'_1=\{U\longmapsto A_4,Y_1\longmapsto V,k_{B_3S_2}\longmapsto k_{bs}\}\\ \sigma'_2=\{U\longmapsto N^i_{B_4},Z_1\longmapsto V,k_{B_4S_3}\longmapsto k_{bs}\}\\ \sigma'_3=\{U_2\longmapsto U,V_2\longmapsto V,k_{B_6S_5}\longmapsto k_{bs}\}\end{cases}$$

$$\Phi'_{p,\varphi}(V,\{U.V\}k_{bs})\text{= \{definition of the lower bound of the witness-function\}}$$

$$\varphi\big(V,\partial[\overline{V}]\{A_4.Y_1\}k_{B_3S_2}\sigma'_1\big)\sqcap\varphi\big(V,\partial[\overline{V}]\,\{N^i_{B_4}.Z_1\}k_{B_4S_3}\sigma'_2\big)\sqcap\varphi\big(V,\partial[\overline{V}]\,\{U_2\,.V_2\}k_{B_6S_5}\sigma'_3\big)\text{= \{Renaming the static neighborhood\}}$$

$$\varphi\big(V,\partial[\overline{V}]\{A_4.Y_1\}k_{bs}\sigma'_1\big)\sqcap\varphi\big(V,\partial[\overline{V}]\,\{N^i_{B_4}.Z_1\}k_{bs}\sigma'_2\big)\sqcap\varphi\big(V,\partial[\overline{V}]\,\{U_2\,.V_2\}k_{bs}\sigma'_3\big)\text{= \{Definition 4.2\}}$$

$$\varphi\big(Y_1,\partial[\overline{Y_1}]\{A_4.Y_1\}k_{bs}\sigma'_1\big)\sqcap\varphi\big(Z_1,\partial[\overline{Z_1}]\,\{N^i_{B_4}.Z_1\}k_{bs}\sigma'_2\big)\sqcap\varphi\big(V_2,\partial[\overline{V_2}]\,\{U_2\,.V_2\}k_{bs}\sigma'_3\big)\text{= \{Derivation in the Definition 4.1\}}$$

$$\varphi(Y_1,\{A_4.Y_1\}k_{bs})\sqcap\varphi\big(Z_1,\{N^i_{B_4}.Z_1\}k_{bs}\big)\sqcap\varphi(V_2,\{V_2\}k_{bs})\text{= \{Since }\varphi\text{=}\varphi^{\text{EK}}_{\text{MAX}}\text{\}}$$

$$\{A_4,B,S\}\cup\{B,S\}\cup\{B,S\}=\{A_4,B,S\}\qquad(5.3.4)$$

3) Compliance with the security criterion in the Theorem 4.6

For any U, from (5.3.1) and (5.3.2) we have:

$$\begin{cases} \Phi'_{p,\varphi}(\{U,\{U.V\}k_{bs}\}k_{bs}) = \{A,B,S\} \sqsupseteq \lceil U \rceil \sqcap \varphi\big(U,\partial[\overline{U}]\{A.\{U.V\}k_{as}\}k_{bs}\big) = \\ \lceil U \rceil \sqcap \{A,B,S\} \text{ if } k_{bs} \text{ is the external protective key of } U \text{ in } \{A.\{U.V\}k_{as}\}k_{bs} (5.3.5.1) \\ \Phi'_{p,\varphi}(\{U,\{U.V\}k_{bs}\}k_{bs}) = \{A,B,S\} \not\sqsupseteq \lceil U \rceil \sqcap \varphi\big(U,\partial[\overline{U}]\{A.\{U.V\}k_{as}\}k_{bs}\big) = \\ \lceil U \rceil \sqcap \{A,S\} \text{ if } k_{as} \text{ is the external protective key of } U \text{ in } \{A.\{U.V\}k_{as}\}k_{bs} \text{ (5.3.5.2)} \end{cases} \quad (5.3.5)$$

For any V, from (5.3.3) and (5.3.4) we have:

$$\begin{cases} \Phi'_{p,\varphi}(V,\{U.V\}k_{bs}) = \{A_4,B,S\} \not\sqsupseteq \lceil V \rceil \sqcap \varphi\big(V,\partial[\overline{V}]\{A.\{U.V\}k_{as}\}k_{bs}\big) = \\ \lceil V \rceil \sqcap \{A,B,S\} \text{ if } k_{bs} \text{ is the external protective key of } V \text{ in } \{A.\{U.V\}k_{as}\}k_{bs} (5.3.6.1) \\ \Phi'_{p,\varphi}(V,\{U.V\}k_{bs}) = \{A_4,B,S\} \not\sqsupseteq \lceil V \rceil \sqcap \varphi\big(U,\partial[\overline{V}]\{A.\{U.V\}k_{as}\}k_{bs}\big) = \\ \lceil V \rceil \sqcap \{A,S\} \text{ if } k_{as} \text{ is the external protective key of } V \text{ in } \{A.\{U.V\}k_{as}\}k_{bs} \text{ (5.3.6.2)} \end{cases} \quad (5.3.6)$$

From (5.3.5) and (5.3.6) we have: the messages of the session S^i are not compliant with the security criterion in the Theorem 4.6 (III).

5.4. Discussion

The verification results of the Woo and Lam protocol are given in Table 2.

From Table 2, we conclude that this version of the Woo-Lam protocol is not compliant with the security criterion in the Theorem 4.6. For this reason, we cannot make any conclusion concerning its confidentiality. All that we can say is: "If the protocol contains a law, it must be because of one of the security decay in the rows 1, 5 and 6 of Table 2.

Consistent with this conclusion, in the literature, we report a law in this protocol that exploits the decay of security of the variable V in the generalized role of the server S as shown in the row 6 of Table 2. The attack scenario is described by Table 3. In fact, the server S may receive a message (the session key k_{ab}^i) substituting the variable V in the message $\{A.\{U.V\}k_{as}\}k_{bs}$ such that it is protected by k_{as} only (see (5.3.3.2)), after that he sends it encrypted by the key k_{bs} in the message $\{A.\{U.V\}k_{as}\}k_{bs}$. This encryption key was not unfortunately enough strong to ensure its confidentiality in that message as shown in the statement (5.3.6.2).

Now that we are aware that this protocol involves this flaw, we can conclude by modus-tollens of the Theorem 4.6 that: "*There is no hope to find any Witness-Function such that this protocol might be increasing using it*".

Table 2. Compliance of the Woo-Lam Protocol with the Theorem 4.6

	α	Gen. Role	R^-	r^+	Theorem 4.6
1	k_{ab}^i	A	X	$\{X.k_{ab}^i\}k_{as}$	✖
2	$\forall X$	A	X	$\{X.k_{ab}^i\}k_{as}$	✔
3	N_b^i	B	A	N_b^i	✔
4	$\forall Y$	B	Y	$\{A.Y\}k_{bs}$	✔
5	$\forall U$	S	$\{A,\{U.V\}k_{as}\}k_{bs}$	$\{U.V\}k_{bs}$	✖
6	$\forall V$	S	$\{A,\{U.V\}k_{as}\}k_{bs}$	$\{U.V\}k_{bs}$	✖

Table 3. Attack Scenario in the Woo-Lam Protocol

$$\begin{array}{l}
\alpha.1.\ \ A \rightarrow I(B){:}\, A \\
\alpha.2.\ \ I(B) \rightarrow A{:}\, N_b^i \\
\alpha.3.\ \ A \rightarrow I(B){:}\, \{N_b^i.k_{ab}^i\}k_{as} \\
\quad \beta.4.\ \ I(B) \rightarrow S{:}\, \{A.\{N_b^i.k_{ab}^i\}k_{as}\}k_{is} \\
\quad \beta.5.\ \ S \rightarrow I(B){:}\, \{N_b^i.k_{ab}^i\}k_{is}
\end{array}$$

We conjecture that the amended version p' of the Woo-Lam protocol given in Table 4 is secure for confidentiality and we prove this using the witness-functions in a future work.

Table 4. The Woo and Lam Protocol (Amended Version)

$$\begin{array}{ll}
p'= & \langle 1, A \rightarrow B : A\rangle \\
 & \langle 2, B \rightarrow A : N_b\rangle \\
 & \langle 3, A \rightarrow B : \{N_b.k_{ab}\}\, k_{as}\rangle \\
 & \quad \langle 4, B \rightarrow S : \{A.\{\, N_{b.}k_{ab}\}\, k_{as}\}k_{bs}\rangle \\
 & \quad \langle 5, S \rightarrow B : \{N_b.k_{ab}\,\}k_{bs}\rangle
\end{array}$$

6. Related Works

We compare our method of protocol verification by the witness-functions to the method by Interpretation Functions proposed by Houmani in [35]-[38] and the method by Rank-Functions proposed by Steve Schneider in [39] and the method by typing proposed by Abadi in [25], [40], [41]. We believe that our method is more efficient and flexible than Houmani's one, simpler than Schneider's one and less restrictive than Abdi's one. We think that it can be used on wider range of protocols.

7. Conclusion and Future Works

In this paper, we presented a new framework for verifying cryptographic protocols statically using the witness-functions for the property of confidentiality. We tested them on the lawed version of the Woo-Lam protocol and we showed that they can even describe laws. In a future work, we intend to extend our witness-functions to authentication. In this respect, we believe that this property could be reached by slightly modifying the criterion set in the Theorem 4.6.

References

[1] Colin, B., & Wenbo, M. (1993). On a limitation of BAN logic. *Proceedings of Workshop on the Theory and Application of Cryptographic Techniques EUROCRYPT '93* (pp. 240-247). Lofthus, Norway: EUROCRYPT.

[2] Abadi, M., & Needham, R. M. (1996). Prudent engineering practice for cryptographic protocols. *IEEE Trans. Software Eng., 22(1),* 6-15.

[3] Rusinowitch, M., & Turuani, M. (2001). Protocol insecurity with finite number of sessions is np-complete. *Proceedings of 14th Computer Security Foundations Workshop CSFW'01* (pp. 174-190). Cape Breton, Canada: IEEE Comp. Soc. Press.

[4] Lowe, G. (1998). Towards a completeness result for model checking of security protocols. *Proceedings of 11th Computer Security Foundations Workshop CSFW'98* (pp. 96-106). Massachusetts, USA: IEEE Comp. Soc. Press.

[5] Nessett, D. M. (1990). A critique of the burrows, abadi and needham logic. *SIGOPS Oper. Syst. Rev., 24(2),* 35-38.

[6] Durgin, N. A., Lincoln, P. D., Mitchell, D. L., & Scedrov, A. (1999). Undecidability of bounded security protocols. *Proceedings of Workshop on Formal Methods and Security Protocols.* Trento, Italy: FMSP.

[7] Comon-Lundh, H., & Cortier, V. (2003). New decidability results for fragments of first-order logic and application to cryptographic protocols. *Proceedings of 14th International Conference on Rewriting Techniques and Applications (RTA'2003): Vol. 2706* (pp. 148-164). Valencia, Spain: Springer-Verlag.

[8] Chevalier, Y., Küsters, R., Rusinowitch, M., & Turuani, M. (2003). Deciding the security of protocols with diffie-hellman exponentiation and products in exponents. *Proceedings of 23rd Conference on Foundations of Software Technology and Theoretical Computer Science: Vol. 2914* (pp. 124-135). Mumbai, India: Springer.

[9] Lowe, G. (1995). An attack on the needham-schroeder public key authentication protocol. *Information Processing Letters.*

[10] Lowe, G. (1995). Breaking and fixing the needham-schroeder public-key protocol using FDR. *Software, Concepts and Tools, 17(3),* 93-102.

[11] Lowe, G., & Roscoe, A. W. (1997). Using CSP to detect errors in the TMN protocol. *IEEE Trans. Software Eng., 23(10),* 659-669.

[12] Nesi, M., & Nocera, G. (2006). Deriving the type law attacks in the otway-rees protocol by rewriting. *Nordic Journal of Computing, 13(1),* 78-97.

[13] Shaikh, S. A., Bush, V. J. (2006). Analysing the woo-lam protocol using CSP and rank functions*. Journal of Research and Practice in Information Technology, 38(1).* from http://www.jrpit.acs.org.au/jrpit/JRPITVolumes/JRPIT38/JRPIT38.1.19.pdf

[14] Schneider. S., & Holloway, R. (1996). Using CSP for protocol analysis: The needham-schroeder public key protocol. Royal Holloway Technical Report.

[15] Chrétien, R., Cortier, V., & Delaune, S. (2013). From security protocols to pushdown automata. *Proceedings of 40th International Colloquium, Part II* (pp. 137-149). Riga, Latvia: Automata, Languages, and Programming.

[16] Paiola, M., & Blanchet, B. (2013). Verification of security protocols with lists: From length one to unbounded length. *Journal of Computer Security, 21(6),* 781-816.

[17] Blanchet, B. (2012). Security protocol verification: Symbolic and computational models. *Proceedings of First International Conference on Theory and Practice of Software, ETAPS* (pp. 3-29). Tallinn, Estonia: Principles of Security and Trust.

[18] Arapinis, M., Delaune, S., Kremer, S. (2014). Dynamic tags for security protocols. *Logical Methods in Computer Science, 10(2),* 1-50.

[19] Delicata, R., & Schneider, S. (2007). An algebraic approach to the verification of a class of diffie-hellman protocols. *Int. J. Inf. Sec., 6(2-3),* 183-196.

[20] Heather, J., Schneider, S., & Teague, V. (2014). Cryptographic protocols with everyday objects. *Formal Asp. Comput., 26(1),* 37-62.

[21] Burrows, M., Abadi, M., & Needham, R. M. (1990). A logic of authentication. *ACM Trans. Comput. Syst., 8(1),* 18-36.

[22] Müller-Olm, M., David, S., & Steffen, B. (1999). Model-checking: A tutorial introduction. *Proceedings of the 6th International Symposium on Static Analysis* (pp. 330–354). London, UK: Springer-Verlag.

[23] Saleh, M., & Debbabi, M. (2007). Modeling security protocols as games. *Proceedings of Third International Symposium on Information Assurance and Security* (pp. 253-260). Manchester, UK: IEEE.

[24] Otrok, H., Mohammed, N., Wang, L., Debbabi, M., & Bhattacharya, P. (2008). A game-theoretic intrusion detection model for mobile ad hoc networks. *Computer Communications, 31(4),* 708-721.

[25] Abadi, M. (1998). Secrecy by typing in security protocols. *Journal of the ACM, 46,* 611-638.

[26] Debbabi, M., Mejri, M., Tawbi, N., & Yahmadi, I. (1997). From protocol specifications to flaws and attack scenarios: An automatic and formal algorithm. *Proceedings of the 6th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises* (pp. 256-262). Washington, DC, USA: IEEE.

[27] Blanchet, B. (2001). An efficient cryptographic protocol verifier based on prolog rules. *Proceedings of 14th Computer Security Foundations Workshop (CSFW'01)* (pp. 82-96). Cape Breton, Canada: IEEE Comp. Soc. Press.

[28] Fattahi, J., Mejri, M., & Houmani, H. (2014). Relaxed conditions for secrecy in a role-based specification. *International Journal of Information Security, 1(1),* 33-36.

[29] Fattahi, J., Mejri, M., & Houmani, H. (2014). Introduction to the witness-functions for secrecy in cryptographic protocols. *Proceedings of the 2014 International Conference on Networks and Information*. Nanjing, China: WIT Press.

[30] Fattahi, J., Mejri, M., & Houmani, H. (2014). New functions for secrecy on real protocols. *Proceedings of Fourth International Conference on Computer Science, Engineering and Applications* (pp. 229-250). Chennai, India: AIRCC.

[31] Fattahi, J., Mejri, M., & Houmani, H. (2014). Secrecy by witness functions. *Proceedings of the Formal Methods for Security Workshop Co-located with the PetriNets: Vol. 1158* (pp. 34-52). Tunis, Tunisia: CEUR.

[32] Fattahi, J., Mejri, M., & Houmani, H. (2014). Secrecy by witness-functions on increasing protocols. *Proceedings of 2014 in Conjunction with the 6th Electronics, Computers & Artificial Intelligence International Conference, Vol 6* (pp. 79-84). Bucharest, Romania: ECAI.

[33] Debbabi, M., Legaré, Y., & Mejri, M. (1998). An environment for the specification and analysis of cryptoprotocols. *Proceedings of IEEE Annual Computer Security Application Conference, ACSAC'98* (pp 321-332). Arizona, USA: IEEE Press.

[34] Debbabi, M., Mejri, M., Tawbi, N., & Yahmadi, I. (1997). Formal automatic verification of authentication crytographic protocols. *Proceedings of the First IEEE International Conference on Formal Engineering Methods, ICFEM'97* (pp. 50-59). Hiroshima, Japan: IEEE Press.

[35] Houmani, H., & Mejri. M. (2007). Practical and universal interpretation functions for secrecy. *Proceedings of International Conference on Security and Cryptography* (pp. 157-164). Barcelona, Spain: IEEE Computer Society.

[36] Houmani, H., & Mejri, M. (2008). Ensuring the correctness of cryptographic protocols with respect to secrecy. *Proceedings of International Conference on Security and Cryptography* (pp. 184-189). Porto, Portugal: IEEE Computer Society.

[37] Houmani, H., & Mejri. M. (2012). Formal analysis of set and nsl protocols using the interpretation functions-based method. *Journal of Computer Networks and Communications.* Retrieved June 27, 2014, from http://www.hindawi.com/journals/jcnc/2012/254942/

[38] Houmani, H., Mejri, M., & Fujita, H. (2009). Secrecy of cryptographic protocols under equational theory. *Knowl.-Based Syst., 22(3),* 160-173.

[39] Schneider, S. (1998). Verifying authentication protocols in CSP. *IEEE Trans. Software Eng., 24(9),* 741-758.

[40] Abadi, M., & Gordon, A. D. (1997). Reasoning about cryptographic protocols in the SPI calculus. *Proceedings of the 8th International Conference on Concurrency Theory* (pp. 59-73). London, UK: Springer-Verlag.

[41] Abadi, M., & Gordon, A. D. (1997). A calculus for cryptographic protocols: The SPI calculus. *Information and Computation, 148(1),* 1–70.

Jaouhar Fattahi is a PhD student in computer science at the Laval University, Canada. His research topics cover protocol security and formal methods. He is a graduate engineer in computer science. He is also a NATO consultant, Sun certified for JEE and a University teacher.

Mohamed Mejri received his Ph.D. degree in specification and analysis of cryptographic protocols in 2001 from Laval University, Canada. He is a professor in the Computer Science and Software Engineering Department of Laval University. His research topics cover computer security, formal methods and software engineering.

Moeiz Miraoui received his Ph.D. degree in computer science from the École de Technologie Supérieure (E.T.S.) University of Quebec, Montreal, Canada in 2009. He is a professor at Umm Al-Qura University, Makkah, Saudi Arabia and a member of the LATIS Laboratory at the E.T.S. His research interests include pervasive and ubiquitous computing, context-aware systems, security and smart spaces.

Hanane Houmani received her Ph.D. degree in specification and analysis of cryptographic protocols from the Laval University, Canada in 2009. She is a professor in the Computer Science Department of Hassan II University, Morocco. Her research topics cover computer security, formal methods and software engineering.