NNGT Int. J. of Information Systems , Vol. 1, July 2014

Relaxed Conditions for Secrecy in a Role-Based Specification

Jaouhar Fattahi[a] *jaouhar.fattahi.1@ulaval.ca*
Mohamed Mejri[a] *mohamed.mejri@ift.ulaval.ca*
Hanane Houmani[b] *hanane.houmani@ift.ulaval.ca*

[a] *LSI Group, Laval University, Quebec, Canada*
[b] *University Hassen II, Morocco*

Abstract

In this paper, we look at the property of secrecy through the growth of the protocol. Intuitively, an increasing protocol preserves the secret. For that, we need functions to estimate the security of messages. Here, we give relaxed conditions on the functions and on the protocol and we prove that an increasing protocol is correct when analyzed with functions that meet these conditions.

Keywords: Cryptographic protocol, role-based specification, secrecy.

DOI : 01.IJIS.2014.1.6

1 Introduction

In this paper, we look at the correctness of a protocol for the property of secrecy through its growth. Intuitively, an increasing protocol preserves the secret. That is to say, if the security of any atomic message does not decrease between receiving and sending steps of a protocol, the secret is never leaked. For that, we should define "good" metrics to estimate the security of any atomic message. This way of thinking has been considered in some previous works. In [1], Steve Schneider proposed the concept of rank-functions as metrics to analyze protocols in CSP [2, 3]. These functions were successful in analyzing Needham-Schroeder protocol. However, a such analysis requires the protocol implementation in CSP. Besides, building rank-functions is not a trivial job and their existence is not sure [4]. In [5] Abadi, using Spi-Calculus [6, 7], guarantees that: "If a protocol typechecks, then it does not leak its secret inputs". To do so, he requests the exchanged messages to be composed of four parts having strictly the following types: {secret, public, any, confounder} in order to recognize the security level of every part. However, this approach cannot analyze real protocols that had been implemented with no respect to this restriction. In the same vein, Houmani et al. [8, 9, 10, 11] defined universal functions called interpretation functions able to analyze a protocol statically and operate on an abstraction of the protocol called generalized roles, that generate a space of messages with variables. An interpretation function must meet some sufficient conditions to be reliable for the analysis. Obviously, less we have conditions on functions, more we have functions and more we have chance to get protocols proved correct since one function may fail to prove the growth of a protocol but another may manage to do. However, we notice that the conditions on functions were so restrictive that only two concrete functions had been proposed. We believe that the condition related to the full-invariance by substitution, which is the property-bridge between an analysis run on messages of the generalized roles (messages with variables) and the conclusion made on valid traces (closed messages), is the most restrictive one. Since the aim of our approach is to build as more reliable functions as we are able to do, we think that if we free a function from this condition, we can build more functions.

Notations

Hereafter, we give some definitions and conventions that we will use throughout this paper.

+ We denote by $\mathcal{C} = \langle \mathcal{M}, \xi, \models, \mathcal{K}, \mathcal{L}^{\sqsupseteq}, \ulcorner . \urcorner \rangle$ the context containing the parameters that affect the analysis of a protocol:
  - $\mathcal{M}$: is a set of messages built from the algebraic signature $\langle \mathcal{N}, \Sigma \rangle$ where $\mathcal{N}$ is a set of atomic names (nonces, keys, principals, etc.) and Σ is a set of allowed functions (*enc*:: encryption, *dec*:: decryption, *pair*:: concatenation (denoted by "." here), etc.). i.e. $\mathcal{M} = T_{\langle \mathcal{N}, \Sigma \rangle}(\mathcal{X})$. We use Γ to denote the set of all possible substitution from $\mathcal{X} \rightarrow \mathcal{M}$. We denote by $\mathcal{A}$ all atomic messages in $\mathcal{M}$, by $\mathcal{A}(m)$ the set of atomic messages (or atoms) in m and by $\mathcal{I}$ the set of agents (principals) including the intruder I. We denote by k^{-1} the reverse key of a key k and we consider that $(k^{-1})^{-1} = k$.
  - ξ : is the equational theory that describes the algebraic properties of the functions in Σ by equations. e.g. $dec(enc(x, y), y^{-1}) = x$.
  - $\models_{\mathcal{C}}$: is the inference system of the intruder under the equational theory. Let M be a set of messages and m a message. $M \models_{\mathcal{C}} m$ means that the intruder is able to infer m from M using her capacity. We extend this notation to traces as following: $\rho \models_{\mathcal{C}} m$ means that the intruder can infer m from the messages exchanged in the trace ρ. We assume that the intruder has the full control of the net as described in the Dolev-Yao model [12]. She can intercept, delete, redirect and modify any message. She knows the public keys of all agents, her private keys and the keys she shares with other agents.

DOI : 01.IJIS.2014.1.6

She can encrypt or decrypt any message with known keys. Formally, the intruder has generically the following rules of building messages:

$(int) : \frac{\square}{M \models_{\mathcal{C}} m}[m \in M \cup K(I)]$

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

$(eq) : \frac{M \models_{\mathcal{C}} m', m' =_{\mathcal{C}} m}{M \models_{\mathcal{C}} m}$, with $(m' =_{\mathcal{C}} m) \equiv (m' =_{\xi_{(\mathcal{C})}} m)$

Example 1.1.

The intruder capacity may be described by the following rules:

$(int) : \frac{\square}{M \models_{\mathcal{C}} m}[m \in M \cup K(I)]$
$(dec) : \frac{M \models_{\mathcal{C}} k, M \models_{\mathcal{C}} m_k}{M \models_{\mathcal{C}} m}$
$(enc) : \frac{M \models_{\mathcal{C}} k, M \models_{\mathcal{C}} m}{M \models_{\mathcal{C}} \{m\}_k}$
$(concat) : \frac{M \models_{\mathcal{C}} m_1, M \models_{\mathcal{C}} m_2}{M \models_{\mathcal{C}} m_1.m_2}$
$(deconcat) : \frac{M \models_{\mathcal{C}} m_1.m_2}{M \models_{\mathcal{C}} m_i}[i \in \{1, 2\}]$

In this example, from a set of messages, an intruder can infer any message in this set, encrypt any message when she possesses previously the encryption key, decrypt any message when she possesses previously the decryption key, concatenate any two messages and deconcatenate them.

- $\mathcal{K}$: is a function from $\mathcal{I}$ to $\mathcal{M}$, that assigns to any agent (principal) a set of atomic messages describing her initial knowledge. We denote by $K_{\mathcal{C}}(I)$ the initial knowledge of the intruder, or simply $K(I)$ where the context is clear.
- $\mathcal{L}^{\sqsupseteq}$: is the security lattice $(\mathcal{L}, \sqsupseteq, \sqcup, \sqcap, \bot, \top)$ used to attribute security levels to messages. A concrete example of a lattice is $(2^{\mathcal{I}}, \subseteq, \cap, \cup, \mathcal{I}, \emptyset)$ that will be used to attribute to a message α the set of principals that are allowed to know it.
- $\ulcorner . \urcorner$: is a partial function that assigns a value of security (type) to a message in $\mathcal{M}$. Let M be a set of messages and m a message. We write $\ulcorner M \urcorner \sqsupseteq \ulcorner m \urcorner$ if $\exists m' \in M . \ulcorner m' \urcorner \sqsupseteq \ulcorner m \urcorner$

+ Our analysis takes place in a role-based specification. A role-based specification 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. Also, an exponent i (the session identifier) is added to each fresh message to emphasize that these components change their values from one run to another. Basically, a generalized role reflects how a particular principal perceives the exchanged messages. A generalized role could be extracted from a protocol by these following steps:

1. Extract the roles from a protocol.
2. Replace the unknown messages by fresh variables for each role.

Roles can be extracted by following these steps:

1. For each principal (agent), extract from the protocol all the steps in which this principal participates. After that, add to that abstraction a session identifier i in the steps identifiers and in fresh values. For instance, from the variation of Woo and Lam protocol given by the Table 1, three roles could be extracted, denoted by R_A (for the principal A), R_B (for the principal B), and R_S (for the principal S).

DOI : 01.IJIS.2014.1.6

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

Table 1: A variation of Woo and Lam Protocol

2. Introduce explicitly an intruder I to capture the fact that the received messages and the sent messages are potentially sent or received by an intruder.
3. Finally, extract all prefixes from those roles where a prefix ends always by a sending step.

From the roles, we generate the generalized roles. A generalized role is an abstraction of a role where unknown messages are replaced by variables. Indeed, a message or a component of a message is replaced by a variable when the receiver cannot make any verification on it, and so she cannot be sure about its integrity or its origin. The generalized roles give a precise idea about the behavior of principals during the protocol runs. The generalized roles of A are:

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

$$
\begin{array}{lllllll}
\mathcal{A}^2_G = & \langle i.1, & A & \rightarrow & I(B) & : & A \rangle. \\
& \langle i.2, & I(B) & \rightarrow & A & : & X \rangle. \\
& \langle i.3, & A & \rightarrow & I(B) & : & \{X, k^i_{ab}\}_{k_{as}} \rangle
\end{array}
$$

The generalized roles of B are:

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

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

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

The generalized role of S is:

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

Hence, the role-based specification of the protocol described by the Table 1 is $\mathcal{R}_G(p) = \{\mathcal{A}^1_G, \mathcal{A}^2_G, \mathcal{B}^1_G, \mathcal{B}^2_G, \mathcal{B}^3_G, \mathcal{S}^1_G\}$. The role-based specification is used to formalize the notion of valid traces of a protocol. More details about the role-based specification are in [13, 14, 15, 16].

DOI : 01.IJIS.2014.1.6

+ A valid trace is an interleaving of instantiated generalized roles where each message sent by the intruder can be produced by her using her capacity and the previous received messages. We denote by $[\![p]\!]$ the set of valid traces of p.
+ We denote by $\mathcal{M}_p^{\mathcal{G}}$ the set of messages with variables generated by $R_G(p)$, by $\mathcal{M}_p$ the set of closed messages generated by substituting terms in $\mathcal{M}_p^{\mathcal{G}}$. We denote by R^+ (respectively R^-) the set of sent messages (respectively received messages) by a honest agent in the role R. Commonly , we reserve the uppercase letters for sets or sequences of elements and the lowercase for single elements. For instance M denotes a set of messages, m a single message, R a role composed of a sequence of steps, r a step and $R.r$ the role ending by the step r.
+ We assume no restriction on the size of messages or the number of sessions in the protocols we analyze.

2 Secrecy in increasing protocols

To analyze a protocol, we need reliable functions to estimate the security level of every atomic message. In this section, we state relaxed conditions allowing to guarantee that a function is reliable. We prove that an increasing protocol is correct with respect to the secrecy property when analyzed with such functions.

2.1 $\mathcal{C}$-reliable interpretation functions

Definition 2.1. (Well-formed interpretation function)
Let F be an interpretation function and $\mathcal{C}$ a context of verification.
F is well-formed in $\mathcal{C}$ if:
$\forall M, M_1, M_2 \subseteq \mathcal{M}, \forall \alpha \in \mathcal{A}(\mathcal{M})$:

$$\begin{cases} F(\alpha, \{\alpha\}) & = & \bot \\ F(\alpha, M_1 \cup M_2) & = & F(\alpha, M_1) \sqcap F(\alpha, M_2) \\ F(\alpha, M) & = & \top, \text{ if } \alpha \notin \mathcal{A}(M) \end{cases}$$

For an atom α in a set of messages M, a well-formed interpretation function returns the bottom value "$\bot$", if $M = \{\alpha\}$. It returns for it in the union of two sets, the minimum "$\sqcap$" of the two values calculated in each set separately. It returns the top value "$\top$", if it does not appear in this set.

Definition 2.2. (Full-invariant-by-intruder interpretation function)
Let F be an interpretation function and $\mathcal{C}$ a context of verification.
F is full-invariant-by-intruder in $\mathcal{C}$ if:
$\forall M \subseteq \mathcal{M}, m \in \mathcal{M}. M \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m). (F(\alpha, m) \sqsupseteq F(\alpha, M)) \vee (\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner)$

A reliable function F should be full-invariant-by-intruder. That is to say, if F attributes a security level to a message α in M, then the intruder can never produce from M another message m that decrease this level (i.e. $F(\alpha, m) \sqsupseteq F(\alpha, M)$) except when α is intended to be known by the intruder (i.e. $\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner$).

Definition 2.3. (Reliable interpretation function)
Let F be an interpretation function and $\mathcal{C}$ a context of verification.

F is $\mathcal{C}$-reliable if F is well-formed and F is full-invariant-by-intruder in $\mathcal{C}$.

DOI : 01.IJIS.2014.1.6

Definition 2.4. (F-increasing protocol)
Let F be an interpretation function, $\mathcal{C}$ a context of verification and p a protocol.
p is F-increasing in $\mathcal{C}$ if:
$\forall R.r \in R_G(p), \forall \sigma \in \Gamma : \mathcal{X} \rightarrow \mathcal{M}_p$ we have:

$$\forall \alpha \in \mathcal{A}(\mathcal{M}).F(\alpha, r^+\sigma) \sqsupseteq \ulcorner \alpha \urcorner \sqcap F(\alpha, R^-\sigma)$$

A F-increasing protocol produces valid traces (interleaving of substituted generalized roles) where every involved principal (every substituted generalized role) never decreases the security levels of received components. When a protocol is F-increasing and F is a reliable function, it will be easy to prove its correctness with respect to the secrecy property. In fact, if every agent appropriately protects her sent messages (if she initially knows the security level of a component, she has to encrypt it with at least one key having a similar or higher security level, and if she does not know its security level, she estimates it using a reliable function), the intruder can never reveal it.

Definition 2.5. (Secret disclosure)
Let p be a protocol and $\mathcal{C}$ a context of verification.
We say that p discloses a secret $\alpha \in \mathcal{A}(\mathcal{M})$ in $\mathcal{C}$ if:

$$\exists \rho \in [\![p]\!].(\rho \models_{\mathcal{C}} \alpha) \wedge (\ulcorner K(I) \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner)$$

A secret disclosure consists in exploiting a valid trace of the protocol (denoted by $[\![p]\!]$) by the intruder using her knowledge $K(I)$ in a context of verification $\mathcal{C}$, to infer a secret α that she is not allowed to know (expressed by: $\ulcorner K(I) \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner$).

Lemma 2.6.
Let F be a $\mathcal{C}$-reliable interpretation function and p a F-increasing protocol.
We have:

$$\forall m \in \mathcal{M}.[\![p]\!] \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m).(F(\alpha, m) \sqsupseteq \ulcorner \alpha \urcorner) \vee (\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner).$$

Proof. See the proof 4 in [17][1] □

The lemma 2.6 asserts that for any atom α in a message generated by an increasing protocol, its security level calculated by a reliable interpretation function is maintained greater than its initial value in the context, if the intruder is not initially allowed to know it. Thus, initially the atom has a certain security level. This level cannot be decreased by the intruder using her initial knowledge and received messages since reliable functions are full-invariant-by-intruder. In each new step of any valid trace, involved messages are better protected since the protocol is increasing. The proof is run by induction on the size of the trace and uses the reliability properties of the interpretation function in every step.

Theorem 2.7. (Correctness of increasing protocols)
Let F be a $\mathcal{C}$-reliable interpretation function and p a F-increasing protocol.

$$p \text{ is } \mathcal{C}\text{-correct with respect to the secrecy property.}$$

Proof.
Let's suppose that p discloses an atomic secret α.

1. The proofs could be downloaded from the following URL: `http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Tun2/preuvesJ.pdf`

DOI : 01.IJIS.2014.1.6

From the definition 2.5 we have:

$$\exists \rho \in [\![p]\!].(\rho \models_{\mathcal{C}} \alpha) \wedge (\ulcorner K(I) \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner) \tag{1}$$

Since F is a $\mathcal{C}$-reliable interpretation function and p an F-increasing protocol, we have from the lemma 2.6:

$$(F(\alpha, \alpha) \sqsupseteq \ulcorner \alpha \urcorner) \vee (\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner) \tag{2}$$

From 1 and 2, we have:

$$F(\alpha, \alpha) \sqsupseteq \ulcorner \alpha \urcorner \tag{3}$$

Since F is well-formed in $\mathcal{C}$, then:

$$F(\alpha, \alpha) = \bot \tag{4}$$

From 3 and 4 we have:

$$\bot = \ulcorner \alpha \urcorner \tag{5}$$

5 is impossible because it is contradictory with: $\ulcorner K(I) \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner$ in 1.

Then p is $\mathcal{C}$-correct with respect to the secrecy property.

3 Comparison with related works

The theorem 2.7 states that an increasing protocol is correct with respect to the secrecy property when analyzed with an interpretation function that is full-invariant by intruder and well-formed, or simply reliable. Compared to the sufficient conditions stated by Houmani et al. in [8, 11], we have one less. Houmani et al. requested that a protocol must be increasing on the messages of the generalized roles of the protocol (that contain variables), and demanded from the interpretation function to resist to the problem of substitution of variables. Even if they gave a clear guideline to build safe functions, just two functions have been defined: DEK and DEKAN. That is due to the complexity to find, and then to prove, that a function meets the full-invariance by substitution property. Here, we free our functions from this restrictive condition in order to be able to build more functions. We relocate this condition in our new definition of an increasing protocol, that is requested now to be increasing on valid traces (closed messages). The problem of substitution migrates to the protocol and becomes easier to handle.

4 Conclusion and future work

Freeing a function from a condition may impel us to take additional precautions when using it. In a future work, we introduce the notion of witness-functions [18, 19] to analyze cryptographic protocols. A witness-function is protocol-dependent that uses derivation techniques to solve the question of substitution locally in the protocol. It offers two bounds that are independent of all substitutions which enables any decision made on the generalized roles (messages with variables) to be exported to valid traces (closed messages). This replaces the restrictive condition of full-invariance by substitution stated in Houmani's work [8, 11]. The witness-functions are successful to prove the correctness of protocols [20]. They even help to locate flaws [21].

DOI : 01.IJIS.2014.1.6

References

[1] Steve Schneider. Verifying authentication protocols in csp. *IEEE Trans. Software Eng.*, 24(9):741–758, 1998.

[2] Steve Schneider. Security properties and csp. In *IEEE Symposium on Security and Privacy*, pages 174–187, 1996.

[3] Steve A. Schneider and Rob Delicata. Verifying security protocols: An application of csp. In *25 Years Communicating Sequential Processes*, pages 243–263, 2004.

[4] James Heather and Steve Schneider. A decision procedure for the existence of a rank function. *J. Comput. Secur.*, 13(2):317–344, March 2005.

[5] Martín Abadi. Secrecy by typing in security protocols. *Journal of the ACM*, 46:611–638, 1998.

[6] Martín Abadi and Andrew D. Gordon. Reasoning about cryptographic protocols in the spi calculus. In *CONCUR*, pages 59–73, 1997.

[7] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *ACM Conference on Computer and Communications Security*, pages 36–47, 1997.

[8] Hanane Houmani and Mohamed Mejri. Practical and universal interpretation functions for secrecy. In *SECRYPT*, pages 157–164, 2007.

[9] Hanane Houmani and Mohamed Mejri. Ensuring the correctness of cryptographic protocols with respect to secrecy. In *SECRYPT*, pages 184–189, 2008.

[10] Hanane Houmani and Mohamed Mejri. Formal analysis of set and nsl protocols using the interpretation functions-based method. *Journal Comp. Netw. and Communic.*, 2012, 2012.

[11] Hanane Houmani, Mohamed Mejri, and Hamido Fujita. Secrecy of cryptographic protocols under equational theory. *Knowl.-Based Syst.*, 22(3):160–173, 2009.

[12] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[13] Jaouhar Fattahi, Mohamed Mejri, and Hanane Houmani. Context of verification and role-based specification `http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Tun2/Context.pdf`. (4):1–4, 2014.

[14] Mourad Debbabi, Y. Legaré, and Mohamed Mejri. An environment for the specification and analysis of cryptoprotocols. In *ACSAC*, pages 321–332, 1998.

[15] Mourad Debbabi, Mohamed Mejri, Nadia Tawbi, and I. Yahmadi. Formal automatic verification of authentication crytographic protocols. In *ICFEM*, pages 50–59, 1997.

[16] Mourad Debbabi, Mohamed Mejri, Nadia Tawbi, and I. Yahmadi. From protocol specifications to flaws and attack scenarios: An automatic and formal algorithm. In *WETICE*, pages 256–262, 1997.

[17] Jaouhar Fattahi, Mohamed Mejri, and Hanane Houmani. Relaxed conditions for secrecy in cryptographic protocols: Proofs and intermediate results `http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Tun2/preuvesJ.pdf`. (9):1–9, 2014.

[18] Jaouhar Fattahi, Mohamed Mejri, and Hanane Houmani. New functions for secrecy in cryptographic protocols `http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Tun2/WF.pdf`. (17):1–17, 2014.

[19] Jaouhar Fattahi, Mohamed Mejri, and Hanane Houmani. The witness-functions: Proofs and intermediate results. `http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Tun2/WitFunProofs.pdf`. (33):1–33, 2014.

DOI : 01.IJIS.2014.1.6

[20] Jaouhar Fattahi, Mohamed Mejri, and Hanane Houmani. Nsl protocol analysis with a witness-function `http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Tun2/NSL.pdf`. (5):1–5, 2014.

[21] Jaouhar Fattahi, Mohamed Mejri, and Hanane Houmani. A variation of needham-schroeder protocol analysis with a witness-function `http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Tun2/Needham.pdf`. (6):1–6, 2014.

DOI : 01.IJIS.2014.1.6