A Theorem for Secrecy in Tagged Protocols Using the Theory of Witness-Functions

Jaouhar Fattahi

Valcartier Research Centre. Defence Research and Development Canada.
2459 de la Bravoure Road, Québec, Canada. G3J 1X5.
Laboratory of Computer Security. Department of Computer Science and Software Engineering.
Pavillon Adrien-Pouliot, local 3770. Laval University. Québec, Canada, G1K 7P4.
E-mail: jaouhar.fattahi@drdc-rddc.gc.ca | jaouhar.fattahi.1@ulaval.ca

Abstract—**In this paper, we enunciate the theorem of secrecy in tagged protocols using the theory of witness-functions and we run a formal analysis on a new tagged version of the Needham-Schroeder public-key protocol using this theorem. We discuss the significance of tagging in securing cryptographic protocols as well.**

Index Terms—**cryptographic protocols, intruder, secrecy, security, tag, unification, witness-function.**

Notice [1]

I. Introduction

Recently, a new category of analytic functions, called witness-functions, has been put forward to analyze cryptographic protocols for secrecy [1]–[4]. These functions assign to every single atomic message involved in the protocol a reasonable level of security. An analysis with a witness-function is the process that tries to make sure that this level of security never goes down between any two consecutive steps, a receiving step and a sending one, from the very first appearance of the atomic message in the protocol until its final destination. This is obviously sufficient to guarantee that any secret will never fall into the hands of an unauthorized agent including an evil intruder. In that case, the protocol is said to be increasing. Certainly, the witness-functions are able to analyze any protocol. However, we notice that they present interesting features when they are used on tagged protocols. In fact, the theorem of analysis acquires a reduced and elegant form and the analysis becomes much quicker. This is because there is a subtle relationship between tagging, on the one hand, and a witness-function definition, on the other hand. In this paper,

we discuss these aspects and we analyze a tagged protocol with a witness-function. The paper is organized as follows. In section II, we recall the theory of witness-functions. In section III, we give an overview on tagged protocol. In section IV, we enunciate the theorem of secrecy in tagged protocols using witness-functions. In section V, we propose a tagged version of the Needham-Schroeder public-key protocol and we analyze it with that theorem. In section VI, we discuss some interesting related works dealing with tagged protocols and we compare them to our approach. In section VII, we conclude.

II. The theory of witness-functions

The theory of witness-functions has been proposed by Fattahi et al. [1]–[4] to statically verify cryptographic protocols for secrecy. A witness-function is an analytic function that attributes a safe level of security to every atomic message in the protocol and the analysis using a witness-function closely follows the growth of this value during the lifecycle of this atom. In this section, we recall the fundaments of this theory. Please notice that we will give the meaning of every notation we use in a natural language as soon as it shows up first.

A. Context of verification

A protocol analysis using the witness-functions runs in a role-based specification [5], [6] under the hypotheses of Dolev-Yao [7]. In this paper, we assume that a protocol is always analyzed under the perfect encryption assumption which means that we do not deal with flaws caused by the cryptographic system in use or the implementation of cryptographic primitives. Equally, we suppose that there is no special equational theory and all secrets, keys and other names are atomic.

B. Reliable function

Definition 1: (Well-formed Function) Let F be a function. F is well-formed iff: $\forall M, M_1, M_2 \subseteq \mathcal{M}, \forall \alpha \in \mathcal{A}(\mathcal{M})$:

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

A well-formed function F should assign the infimum level of security (i.e. $\bot$) to an atomic message α that shows up in clear (not encrypted) in a set M of messages. This is obviously

to express that anybody who knows M inevitably knows α. It assigns to an atomic message in the union of two sets of messages the minimum of the two levels (i.e. $\sqcap$) assigned in each set alone. It assigns the supremum (i.e. $\top$) to an atomic message α that does not even appear in M. This is to express the fact that nobody is able to know α when he knows M. We note by $\mathcal{A}(M)$ the atomic messages of M.

Definition 2: (Full-invariant-by-intruder Function) Let F be a function. F is full-invariant-by-intruder iff: $\forall M \subseteq \mathcal{M}, m \in \mathcal{M}, \alpha \in \mathcal{A}(m)$:
$M \models m \Rightarrow (F(\alpha, m) \sqsupseteq F(\alpha, M)) \vee (\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner)$.

A full-invariant-by-intruder function F should resist against any malicious tentative to lower the level of security by an intruder once F assigns to an atomic message α a level of security in a set of messages M. That is to say that the intruder can never infer (i.e. $\models$) from this set M any other message m in which this level may be lower than the one given in M (i.e. $F(\alpha, m) \not\sqsupseteq F(\alpha, M)$), exception made when the intruder is explicitly authorized to know α (i.e. $\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner$). We say that a function F is reliable when it is well-formed and full-invariant-by-intruder.

Definition 3: (F-Increasing Protocol) Let F be a function and p be a protocol. p is F-increasing iff: $\forall R.r, \forall \sigma, \forall \alpha \in \mathcal{A}(r^+)$, we have: $F(\alpha, r^+\sigma) \sqsupseteq \ulcorner \alpha \urcorner \sqcap F(\alpha, R^-\sigma)$

An F-increasing protocol is a protocol that constantly pumps traces (substituted generalized roles in a role-based specification) with atomic messages α that always have a security level, calculated by F, higher (i.e. $\sqsupseteq$) upon a sending step (i.e. in the generalized role $r^+\sigma$, the sign $+$ denotes a sending operation and σ a substitution corresponding to a possible execution of the protocol) than the one calculated by the same function in the messages received in the latest receiving step (i.e. in the generalized role $R^-\sigma$, the sign $-$ denotes a receiving operation), or higher than the level of security of α obtained directly from within the context of verification (i.e. $\ulcorner \alpha \urcorner$), if it is available.

Theorem 1: (Secrecy in Increasing Protocols) Let F be a reliable function and p be an F-increasing protocol.
$$p \text{ is correct for secrecy.}$$

Theorem 1 brings up a very important result. It establishes that a protocol is correct for secrecy if it could be demonstrated increasing using a reliable function F. The sketch of the proof is quite straightforward. That is, if the attacker manages to discover an initially protected secret α (get it in clear) then its security level calculated by F should be the infimum seeing as F is well-formed. This scenario cannot be rooted in the rules of the protocol seeing as this latter is F-increasing and its rules constantly raise the level of security of α. This scenario could not happen either if the intruder uses his capabilities seeing as F is full-invariant-by-intruder and then the intruder could not forge any message in which the security level of α may decline. Hence, this scenario could simply never happen and the secret could never be disclosed. The complete formal proof could be found in [8].

C. Construction of Reliable Function

Here we give one constructive way to build a reliable function. Let's consider the function F defined as follows:

Definition 4: (Reliable Function)

1.	$F(\alpha, \{\alpha\})$	$=$	$\perp$
2.	$F(\alpha, M_1 \cup M_2)$	$=$	$F(\alpha, M_1) \sqcap F(\alpha, M_2)$
3.	$F(\alpha, M)$	$=$	$\top$, if $\alpha \notin \mathcal{A}(M)$
4.	$F(\alpha, m_1.m_2)$	$=$	$F(\alpha, \{m_1, m_2\})$
5.	$F(\alpha, \{m\}_k)$	$=$	$F(\alpha, \{m\})$, if $\ulcorner k^{-1} \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner$
6.	$F(\alpha, \{m\}_k)$	$=$	$\ulcorner k^{-1} \urcorner \sqcap \text{ID}(m)$, if $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$

The first three steps 1., 2. and 3. directly grant the function F the property of being well-formed. The step 4. deconcatenates a message $m_1.m_2$ into two messages m_1 and m_2 and F returns the same level of security as in the set $\{m_1, m_2\}$. That is because an intruder, although he can deconcatenate any message $m_1.m_2$, he cannot infer about α in $m_1.m_2$ more than he could infer about it in each of m_1 or m_2 separately. The step 5. ignores encryption with an outer weak key (i.e. $\ulcorner k^{-1} \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner$) and looks for a deeper strong key. That is because if α is encrypted with a weak key, it can fall into the hands of an unauthorized agent. The step 6. makes sure that α is encrypted with a strong key k (i.e. $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$ meaning the reverse key k^{-1} must be known only by a part of agents who are authorized to know α in the context) and F returns the set of agent identities who know the reverse key (i.e. $\ulcorner k^{-1} \urcorner$) as well as the identity of all the neighbors of α in m (i.e. $\text{ID}(m)$). The step 6. transforms F into a full-invariant-by-intruder function. In fact, an unauthorized intruder who attempts to mislead F should obtain the key k^{-1} beforehand. Hence, his knowledge must include k^{-1} (i.e. $\ulcorner K(I) \urcorner \sqsupseteq \ulcorner k^{-1} \urcorner$). Since the key k^{-1} is such that $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$ then the knowledge of the intruder must satisfy $\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner$ as well owing to the transitivity of the comparator "$\sqsupseteq$". This is contradictory to the fact that the intruder is unauthorized to know α.

Example 1: Let us have the following context of verification: $\ulcorner \alpha \urcorner = \{A, B, S\}$; $m = \{\{C.\{\alpha.D\}_{k_{as}}\}_{k_{ab}}\}_{k_{ac}}$; $k_{ac}^{-1} = k_{ac}, k_{ab}^{-1} = k_{ab}, k_{as}^{-1} = k_{as}$; $\ulcorner k_{ac} \urcorner = \{A, C\}, \ulcorner k_{as} \urcorner = \{A, S\}, \ulcorner k_{ab} \urcorner = \{A, B\}$. We have:
$F(\alpha, m) = F(\alpha, \{C.\{\alpha.D\}_{k_{as}}\}_{k_{ab}}) = \{C, D\} \cup \ulcorner k_{ab}^{-1} \urcorner = \{C, D\} \cup \{A, B\} = \{A, B, C, D\}$.
Please notice that the outermost encryption by k_{ac} has been ignored by F because it is a weak key since the agent C is not authorized to know α in the context (i.e. $\ulcorner \alpha \urcorner = \{A, B, S\}$). This case falls into the step 5.

Other reliable functions could be found in [2], [4]. In the rest of this paper, we will only use the function defined in this subsection and we refer to it by F.

D. Witness-functions to reduce the impact of variables

The function F as defined above may be suitable to assign security level for atomic messages but in ground terms only. Nevertheless, when we analyze a protocol, messages are not necessarily ground and may contain variables. To cope with this situation, the idea is to use the derivative function F' of F that operates like F but after eliminating variables

from the neighborhood of α (e.g. $F'(\alpha, \{\alpha.X.B\}_{k_{cd}}) = F(\alpha, \{\alpha.B\}_{k_{cd}}) = \{B, C, D\}$). Although the derivative function remains well-formed and full-invariant-by-intruder, it may lose its quality as a function and may return multiple and contradictory values for the same trace generated by a substitution in the generalized roles. For example, if the trace is $\{\alpha.A.B\}_{k_{cd}}$ that could be produced by substitution in two generalized roles $\{\alpha.X.B\}_{k_{cd}}$ and $\{\alpha.Y\}_{k_{cd}}$, the function F' assigns to α the level of security $\{B, C, D\}$ when the trace originates from the first generalized role, and the level of security $\{C, D\}$ if the trace originates from the second one. To overcome this incoherence, we define the witness-functions.

Definition 5: [Witness-Function]

$$\mathcal{W}_{p,F}(\alpha, m\sigma) = \underset{\{(m', \sigma') \in \tilde{\mathcal{M}}_p^{\mathcal{G}} \otimes \Gamma \mid m'\sigma' = m\sigma\}}{\sqcap} F'(\alpha, m'\sigma')$$

A witness-function $\mathcal{W}_{p,F}$ calculates the level of security of an atomic message α in a trace $m\sigma$ by using F' applied to all the possible origins m' in the messages $\tilde{\mathcal{M}}_p$ generated by the generalized roles and returns the minimum, which is obviously a single value. Nevertheless, a witness-function could not be used as is to analyze a protocol since the analysis runs statically on the generalized roles not on the traces (i.e. $m\sigma$) which are dynamic entities. For that, we bound a witness-function by two static bounds as follows.

Lemma 1: [binding a witness-function]

$$F'(\alpha, m) \sqsupseteq \mathcal{W}_{p,F}(\alpha, m\sigma) \sqsupseteq \underset{\{(m', \sigma') \in \tilde{\mathcal{M}}_p^{\mathcal{G}} \otimes \Gamma \mid m'\sigma' = m\sigma'\}}{\sqcap} F'(\alpha, m'\sigma')$$

The upper-bound $F'(\alpha, m)$ returns a minimal set of identities from m after removing all variables in m. The lower-bound $\underset{\{(m', \sigma') \in \tilde{\mathcal{M}}_p^{\mathcal{G}} \otimes \Gamma \mid m'\sigma' = m\sigma'\}}{\sqcap} F'(\alpha, m'\sigma')$ returns all the identities gathered from all the messages that could be unified with m. The witness-function returns certain identities in between which are known only when the protocol is executed from the actual origins of the trace only. The inequality is quite intuitive since m is a guaranteed origin of the trace $m\sigma$ and the actual origins of the trace $m\sigma$ is a subset of the messages that are unifiable with m. The two bounds are obviously statically computable.

Theorem 2: [Decision Procedure for Secrecy with a Witness-Function] Let p be a protocol. Let $\mathcal{W}_{p,F}$ be a witness-function. p is correct for secrecy if: $\forall R.r \in R_G(p), \forall \alpha \in \mathcal{A}(r^+)$ we have:

$$\underset{\{(m', \sigma') \in \tilde{\mathcal{M}}_p^{\mathcal{G}} \otimes \Gamma \mid m'\sigma' = r^+\sigma'\}}{\sqcap} F'(\alpha, m'\sigma') \sqsupseteq \ulcorner \alpha \urcorner \sqcap F'(\alpha, R^-)$$

Theorem 2 establishes a decision procedure for secrecy using the bounds a witness-function. When a message is sent (i.e. r^+), it is analyzed largely with the lower-bound of a witness-function. When a message is received (i.e R^-), it is analyzed strictly with the upper-bound of a witness-function. Any dishonest identity ambushed by the lower-bound that is not returned by the upper-bound will be interpreted as an intrusion. The protocol is then decided not increasing and the analysis halts with a failure flag. Theorem 2 is a direct result of Theorem 1 and Lemma 1. Please notice that Theorem 2 does not imply the witness-function itself (i.e. $\mathcal{W}_{p,F}$). It involves its bounds only.

III. TAGGED PROTOCOLS

A tag is any subtlety or any syntactic annotation put inside a message to differentiate it from another message. A tagged protocol is a protocol such that every message received by any agent has a unique and regular origin. That implies that every single message (an encryption pattern) containing a variable (i.e. something that the receiver does not know) is distinguishable from any other message (any other encryption pattern) and does not unify with any message other than the regular message that the receiver is expecting to get through the network from the right agent. Tagging a protocol could be reached by inserting an identity beside some atom in the message. For example, if an agent A receives the message $\{\alpha.B.X\}_{k_{ab}}$ where the variable X is supposed to be a nonce N_b sent by a regular agent B and the protocol generates also the message $\{\alpha.B.C\}_{k_{ab}}$, we can change the message $\{\alpha.B.X\}_{k_{ab}}$ in the definition of the protocol by $\{A.\alpha.B.X\}_{k_{ab}}$ in the new tagged version of the protocol and hence the message $\{\alpha.B.C\}_{k_{ab}}$ will not unify with it. All the same, a signature could be an efficient tag, too. For example, we can change the message $\{\alpha.B.X\}_{k_{ab}}$ in the definition of the protocol by $\{\{\alpha\}_{k_b^{-1}}.B.X\}_{k_{ab}}$ in the new tagged version of the protocol to prevent $\{\alpha.B.C\}_{k_{ab}}$ from unifying with it. Tagging could be also reached by inserting an ordinal number into an encrypted message or inserting a string describing the type of certain components inside. In general, tagging prevents man-in-the-middle attacks from happening by offering the receiver the way to distinguish a regular message from an irregular one.

IV. THEOREM FOR SECRECY IN TAGGED PROTOCOLS

As a matter of fact, when a protocol is tagged (all its messages are distinguishable one from another), it becomes nonsense talking about message that overlap (unifiable). This has a direct impact on the reduction of Theorem 2. In fact, the expression $\underset{\{(m', \sigma') \in \tilde{\mathcal{M}}_p^{\mathcal{G}} \otimes \Gamma \mid m'\sigma' = r^+\sigma'\}}{\sqcap} F'(\alpha, m'\sigma')$ in Theorem 2 will be reduced to $F'(\alpha, r^+)$. That is, the lower-bound $\underset{\{(m', \sigma') \in \tilde{\mathcal{M}}_p^{\mathcal{G}} \otimes \Gamma \mid m'\sigma' = r^+\sigma'\}}{\sqcap} F'(\alpha, m'\sigma')$ means F' applied to all the patterns in the generalized roles that are unifiable with the message r^+ and its goal is to ambush dishonest identities that could be inserted in the neighborhood of an analyzed atomic message α. Nevertheless, this could never happen when the protocol is tagged. A tagged protocol creates in fact a series of *from regular to regular* data flow in which the intruder is hopeless to launch any man-in-the-middle attack. In that case, if the protocol happens to be incorrect, that will definitely be because it is not increasing by construction because of a bad reasoning on the knowledge of every agent and without any intervention from the intruder. This brings us to the following theorem.

$$p= \begin{array}{llll} \langle 1, & A & \longrightarrow & B: & \{N_a.A.B\}_{k_b}\rangle \\ \langle 2, & B & \longrightarrow & A: & \{A.B.N_a\}_{k_a}.\{B.A.N_b\}_{k_a}\rangle \\ \langle 3, & A & \longrightarrow & B: & \{N_b.B.A.N_a\}_{k_b}\rangle. \end{array}$$

<div align="center">

TABLE I

A TAGGED VERSION OF THE NEEDHAM-SCHROEDER PROTOCOL

</div>

Theorem 3: [Theorem of Secrecy for Tagged Protocols] Let p be a tagged protocol. Let $\mathcal{W}_{p,F}$ be a witness- function. p is correct for secrecy if: $\forall R.r \in R_G(p), \forall \alpha \in \mathcal{A}(r^+)$ we have:

$$F'(\alpha, r^+) \sqsupseteq \ulcorner\alpha\urcorner \sqcap F'(\alpha, R^-)$$

Theorem 3 enables tagged protocols to use simply the derivative function F' on both the received generalized role and the sent one to determine whether or not the tagged protocol is increasing, with no need to perform any further unifications. It is worth mentioning that verifying whether or not a protocol is tagged is an easy task that is carried out only once before analyzing the protocol. It is equally worth noticing that Theorem 3 sets just sufficient conditions for the tagged protocol correctness regarding secrecy, which conditions are not inevitably necessary since the problem of secrecy remains undecidable in general. In the rest of the paper, we will refer to Theorem 3 by the acronym TSTP.

V. FORMAL ANALYSIS OF A TAGGED VERSION OF THE NEEDHAM-SCHROEDER PUBLIC-KEY PROTOCOL

In this section, we propose our new tagged version of the Needham-Schroeder public-key protocol (different from the NSL protocol) and we analyze it with Theorem 3 (TSTP) for secrecy. This version is given in Table I.

A. Context setting

The generalized roles of p are defined by $\mathcal{R}_G(p) = \{\mathcal{A}_G, \mathcal{B}_G\}$ where:

$$\begin{array}{lllll} \mathcal{A}_G = & i.1 & A & \longrightarrow I(B): & \{N_a^i.A.B\}_{k_b} \\ & i.2 & I(B) & \longrightarrow A: & \{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a} \\ & i.3 & A & \longrightarrow I(B): & \{X.B.A.N_a^i\}_{k_b} \\ \mathcal{B}_G = & j.1 & I(A) & \longrightarrow B: & \{Y.A.B\}_{k_b} \\ & j.2 & B & \longrightarrow I(A): & \{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a} \\ & j.3 & I(A) & \longrightarrow B: & \{N_b^j.B.A.Y\}_{k_b} \end{array}$$

Initial knowledge :
$\ulcorner A\urcorner = \bot$; $\ulcorner B\urcorner = \bot$; (i.e. two public identities)
$\ulcorner N_a\urcorner = \{A, B\}$ (i.e. secret shared between A and B);
$\ulcorner N_b\urcorner = \{A, B\}$ (i.e. secret shared between A and B);
$\ulcorner k_a^{-1}\urcorner = \{A\}$; (i.e. private key of A)
$\ulcorner k_b^{-1}\urcorner = \{B\}$; (i.e. private key of B)
$\ulcorner k_a\urcorner = \bot$; (i.e. public key of A)
$\ulcorner k_b\urcorner = \bot$; (i.e. public key of B)
$(\mathcal{L}, \sqsupseteq, \sqcup, \sqcap, \bot, \top) = (2^{\mathcal{I}}, \subseteq, \cap, \cup, \mathcal{I}, \emptyset)$; (i.e. security lattice)
$\mathcal{I} = \{I, A, B\}$; (i.e. intruder and regular agents present on the net)
$\mathcal{X}_p = \{X, Y\}$ is the set of variables. F is the function given by Definition 4 and F' is its derivative form.

B. Tagging verification

Before we dive into the analysis, let us make sure that this protocol is a tagged one. At the first sight, an attentive eye should remark that the protocol is tagged by the position of the identities in its messages. In fact, the encrypted message $\{N_a.A.B\}_{k_b}$ is the

only one that contains the identity of the receiver (i.e. B) at the last position. The encrypted message $\{A.B.N_a\}_{k_a}$ is the only one that contains the identity of the receiver (i.e. A) at the first position and the identity of the sender (i.e. B) at the second position. The encrypted message $\{B.A.N_b\}_{k_a}$ is the only one that contains the identity of the receiver (i.e. A) at the second position and the identity of the sender (i.e. B) at the first position. Finally, the encrypted message $\{N_b.B.A.N_a\}_{k_b}$ is the only one that contains the identity of the receiver (i.e. B) followed by the identity of the sender (i.e. A) that must show up in the middle of the message only. This makes all the encryptions distinguishable one from another from a receiver point of view. More rigorously, according to the generalized roles $\mathcal{A}_G$, the agent A is a receiver in the step $i.2$. The first message he receives is $\{A.B.N_a^i\}_{k_a}$ which is the regular message expected by B. The other message is $\{B.A.X\}_{k_a}$ which unifies only with $\{B.A.N_b^i\}_{k_a}$ which is the regular message that A is expecting.

According to the generalized roles $\mathcal{B}_G$, the agent B is a receiver in two steps.
1) In the step $j.1 : B$ receives $\{Y.A.B\}_{k_b}$. This message unifies only with the message $\{N_a^i.A.B\}_{k_b}$, which is the regular message that B is expecting;
2) In the step $j.3 : B$ receives $\{N_b^j.B.A.Y\}_{k_b}$. This message unifies only with $\{X.B.A.N_a^i\}_{k_b}$. Upon replacing X by N_b^j and Y by N_a^i, the received message becomes $\{N_b^j.B.A.N_a^i\}_{k_b}$, which is the regular message that B is expecting.

Therefore, this protocol is a tagged one and Theorem TSTP applies.

C. Analyzing the generalized role of A

As defined in the generalized role $\mathcal{A}_G$, an agent A may participate in two receiving-sending steps. In the first step, he receives nothing and sends the message $\{N_a^i.A.B\}_{k_b}$. In the subsequent step, he receives the message $\{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a}$ and sends the message $\{X.B.A.N_a^i\}_{k_b}$. This is represented by the following two rules.

$$S_A^1 : \frac{\square}{\{N_a^i.A.B\}_{k_b}} \qquad\qquad S_A^2 : \frac{\{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a}}{\{X.B.A.N_a^i\}_{k_b}}$$

1) Analyzing exchanged messages in S_A^1:

1- For N_a^i:

a- On sending: $r_{S_A^1}^+ = \{N_a^i.A.B\}_{k_b}$

$$F'(N_a^i, r_{S_A^1}^+) = F'(N_a^i, \{N_a^i.A.B\}_{k_b})$$
{No variable in the neighborhood of N_a^i to be removed by derivation}
$$\begin{aligned} &= F(N_a^i, \{N_a^i.A.B\}_{k_b}) \\ &\quad \{\text{Definition 4}\} \\ &= \{A, B\} \cup \ulcorner k_b^{-1}\urcorner \\ &\quad \{\text{Since } \ulcorner k_b^{-1}\urcorner = \{B\}\} \\ &= \{A, B\} \cup \{B\} \\ &= \{A, B\} \qquad (1.1) \end{aligned}$$

b- On receiving: $R_{S^i}^- = \emptyset$

$$F'(N_a^i, R_{S_A^1}^-) = F'(N_a^i, \emptyset)$$
{No variable in the neighborhood of N_a^i to be removed by derivation}
$$\begin{aligned} &= F(N_a^i, \emptyset) \\ &\quad \{\text{Definition 4}\} \\ &= \top \qquad (1.2) \end{aligned}$$

2- Accordance with Theorem TSTP:

From (1.2) and since $\ulcorner N_a\urcorner = \{A, B\}$, we have:

$$\ulcorner N_a^i\urcorner \sqcap F'(N_a^i, R_{S_A^1}^-) = \{A, B\} \sqcap \top = \{A, B\} \quad (1.3)$$

From (1.1) and (1.3), we have :

$$F'(N_a^i, r_{S_A^1}^+) \sqsupseteq \ulcorner N_a^i \urcorner \sqcap F'(N_a^i, R_{S_A^1}^-) \qquad (1.4)$$

From (1.4), S_A^1 respects Theorem TSTP. $\qquad$ (I)

2) Analyzing exchanged messages in S_A^2:

1- For N_a^i:

a- On sending: $r_{S_A^2}^+ = \{X.B.A.N_a^i\}_{k_b}$

$$
\begin{aligned}
F'(N_a^i, r_{S_A^2}^+) &= F'(N_a^i, \{X.B.A.N_a^i\}_{k_b}) \\
&\quad \{\text{The variable } X \text{ is removed by derivation}\} \\
&= F(N_a^i, \{B.A.N_a^i\}_{k_b}) \\
&\quad \{\text{Definition 4}\} \\
&= \{A, B\} \cup \ulcorner k_b^{-1} \urcorner \\
&\quad \{\text{Since } \ulcorner k_b^{-1} \urcorner = \{B\}\} \\
&= \{A, B\} \cup \{B\} \\
&= \{A, B\} \qquad (2.1)
\end{aligned}
$$

b- On receiving: $R_{S_A^2}^- = \{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a}$

$$
\begin{aligned}
F'(N_a^i, R_{S_A^2}^-) &= F'(N_a^i, \{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a}) \\
&\quad \{\text{The variable } X \text{ is removed by derivation}\} \\
&= F(N_a^i, \{A.B.N_a^i\}_{k_a}.\{B.A\}_{k_a}) \\
&\quad \{\text{Definition 4 and } F \text{ is well-formed}\} \\
&= F(N_a^i, \{A.B.N_a^i\}_{k_a}) \sqcap F(N_a^i, \{B.A\}_{k_a}) \\
&\quad \{F \text{ is well-formed}\} \\
&= F(N_a^i, \{A.B.N_a^i\}_{k_a}) \sqcap \top \\
&\quad \{\text{Security lattice property}\} \\
&= F(N_a^i, \{A.B.N_a^i\}_{k_a}) \\
&\quad \{\text{Definition 4}\} \\
&= \{A, B\} \cup \ulcorner k_a^{-1} \urcorner \\
&\quad \{\text{Since } \ulcorner k_a^{-1} \urcorner = \{A\}\} \\
&= \{A, B\} \cup \{A\} \\
&= \{A, B\} \qquad (2.2)
\end{aligned}
$$

2- For X:

c- On sending: $r_{S_A^2}^+ = \{X.B.A.N_a^i\}_{k_b}$

$$
\begin{aligned}
F'(X, r_{S_A^2}^+) &= F'(X, \{X.B.A.N_a^i\}_{k_b}) \\
&\{\text{No variable in the neighborhood of } X \text{ to be removed by derivation}\} \\
&= F(X, \{X.B.A.N_a^i\}_{k_b}) \\
&\quad \{\text{Definition 4}\} \\
&= \{A, B\} \cup \ulcorner k_b^{-1} \urcorner \\
&\quad \{\text{Since } \ulcorner k_b^{-1} \urcorner = \{B\}\} \\
&= \{A, B\} \cup \{B\} \\
&= \{A, B\} \qquad (2.3)
\end{aligned}
$$

d- On receiving: $R_{S_A^2}^- = \{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a}$

$$
\begin{aligned}
F'(X, R_{S_A^2}^-) &= F'(X, \{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a}) \\
&\{\text{No variable in the neighborhood of } X \text{ to be removed by derivation}\} \\
&= F(X, \{A.B.N_a^i\}_{k_a}.\{B.A.X\}_{k_a}) \\
&\quad \{\text{Definition 4 and } F \text{ is well-formed}\} \\
&= F(X, \{A.B.N_a^i\}_{k_a}) \sqcap F(X, \{B.A.X\}_{k_a}) \\
&\quad \{F \text{ is well-formed}\} \\
&= \top \sqcap F(X, \{B.A.X\}_{k_a}) \\
&\quad \{\text{Security lattice property}\} \\
&= F(X, \{B.A.X\}_{k_a}) \\
&\quad \{\text{Definition 4}\} \\
&= \{A, B\} \cup \ulcorner k_a^{-1} \urcorner \\
&\quad \{\text{Since } \ulcorner k_a^{-1} \urcorner = \{A\}\} \\
&= \{A, B\} \cup \{A\} \\
&= \{A, B\} \qquad (2.4)
\end{aligned}
$$

3- Accordance with Theorem TSTP:

From (2.1), (2.2), we have directly:

$$F'(N_a^i, r_{S_A^2}^+) \sqsupseteq \ulcorner N_a^i \urcorner \sqcap F'(N_a^i, R_{S_A^2}^-) \qquad (2.5)$$

From (2.3) and (2.4), we have directly:

$$F'(X, r_{S_A^2}^+) \sqsupseteq \ulcorner X \urcorner \sqcap F'(X, R_{S_A^2}^-) \qquad (2.6)$$

From (2.5) and (2.6), S_A^2 respects Theorem TSTP. $\qquad$ (II)

D. Analyzing the generalized role of B

As defined in the generalized role $\mathcal{B_G}$, an agent B may participate in just one receiving-sending step in which he receives the message $\{Y.A.B\}_{k_b}$ and sends the message $\{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a}$. This is represented by the following rule.

$$S_B : \frac{\{Y.A.B\}_{k_b}}{\{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a}}$$

1) Analyzing exchanged messages in S_B:

1- For N_b^j:

a- On sending: $r_{S_B}^+ = \{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a}$

$$
\begin{aligned}
F'(N_b^j, r_{S_B}^+) &= F'(N_b^j, \{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a}) \\
&\quad \{\text{The variable } Y \text{ is removed by derivation}\} \\
&= F(N_b^j, \{A.B\}_{k_a}.\{B.A.N_b^j\}_{k_a}) \\
&\quad \{\text{Definition 4 and } F \text{ is well-formed}\} \\
&= F(N_b^j, \{A.B\}_{k_a}) \sqcap F(N_b^j, \{B.A.N_b^j\}_{k_a}) \\
&\quad \{\text{Definition 4 and } F \text{ is well-formed}\} \\
&= F(N_b^j, \{A.B\}_{k_a}) \sqcap F(N_b^j, \{B.A.N_b^j\}_{k_a}) \\
&\quad \{F \text{ is well-formed}\} \\
&= \top \sqcap F(N_b^j, \{B.A.N_b^j\}_{k_a}) \\
&\quad \{\text{Security lattice property}\} \\
&= F(N_b^j, \{B.A.N_b^j\}_{k_a}) \\
&\quad \{\text{Definition 4}\} \\
&= \{A, B\} \cup \ulcorner k_a^{-1} \urcorner \\
&\quad \{\text{Since } \ulcorner k_a^{-1} \urcorner = \{A\}\} \\
&= \{A, B\} \cup \{A\} \\
&= \{A, B\} \qquad (3.1)
\end{aligned}
$$

b- On receiving: $R_{S_B}^- = \{Y.A.B\}_{k_b}$

$$
\begin{aligned}
F'(N_b^j, R_{S_B}^-) &= F'(N_b^j, \{Y.A.B\}_{k_b}) \\
&\quad \{\text{The variable } Y \text{ is removed by derivation}\} \\
&= F(N_b^j, \{A.B\}_{k_b}) \\
&\quad \{F \text{ is well-formed}\} \\
&= \top \qquad (3.2)
\end{aligned}
$$

2- For Y:

a- On sending: $r_{S_B}^+ = \{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a}$

$$
\begin{aligned}
F'(Y, r_{S_B}^+) &= F'(Y, \{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a}) \\
&\{\text{No variable in the neighborhood of } Y \text{ to be removed by derivation}\} \\
&= F(Y, \{A.B.Y\}_{k_a}.\{B.A.N_b^j\}_{k_a}) \\
&\quad \{\text{Definition 4 and } F \text{ is well-formed}\} \\
&= F(Y, \{A.B.Y\}_{k_a}) \sqcap F(Y, \{B.A.N_b^j\}_{k_a}) \\
&\quad \{F \text{ is well-formed}\} \\
&= F(Y, \{A.B.Y\}_{k_a}) \sqcap \top \\
&\quad \{\text{Security lattice property}\} \\
&= F(Y, \{A.B.Y\}_{k_a}) \\
&\quad \{\text{Definition 4}\} \\
&= \{A, B\} \cup \ulcorner k_a^{-1} \urcorner \\
&\quad \{\text{Since } \ulcorner k_a^{-1} \urcorner = \{A\}\} \\
&= \{A, B\} \cup \{A\} \\
&= \{A, B\} \qquad (3.3)
\end{aligned}
$$

b- On receiving: $R_{S_B}^- = \{Y.A.B\}_{k_b}$

$$
\begin{aligned}
F'(Y, R_{S_B}^-) \quad &= \quad F'(Y, \{Y.A.B\}_{k_b}) \\
&\{\text{No variable in the neighborhood of } Y \text{ to be removed by derivation}\} \\
&= \quad F(Y, \{Y.A.B\}_{k_b}) \\
&\quad \{\text{Definition 4}\} \\
&= \quad \{A, B\} \cup \ulcorner k_b^{-1} \urcorner \\
&\quad \{\text{Since } \ulcorner k_b^{-1} \urcorner = \{B\}\} \\
&= \quad \{A, B\} \cup \{B\} \\
&= \quad \{A, B\} \qquad\qquad\qquad (3.4)
\end{aligned}
$$

3- Accordance with Theorem TSTP:

From (3.1), (3.2) and since $\ulcorner N_b \urcorner = \{A, B\}$ we have:

$$F'(N_b^j, r_{S_B}^+) \sqsupseteq \ulcorner N_b^j \urcorner \sqcap F'(N_b^j, R_{S_B}^-) \qquad (3.5)$$

From (3.3) and (3.4), we have directly:

$$F'(Y, r_{S_B}^+) \sqsupseteq \ulcorner Y \urcorner \sqcap F'(Y, R_{S_A}^-) \qquad (3.6)$$

From (3.5) and (3.6), S_B respects Theorem TSTP. $\qquad$ (III)

VI. Comparison with related works

From (I), (II) and (II), we deduce that the tagged version of the Needham-Schroeder public-key protocol given in Table I fully respects Theorem TSTP. Hence, we conclude that it is correct for secrecy. In fact, tagging constitutes an efficient way to create well-structured protocols that help avoid mis-interpretation of received messages and a regular agent is always assured that he is receiving messages from the right regular agent. A tagged protocol is a good candidate for an analysis by witness-functions that can verify it quickly owing to the simplified theorem we have exhibited so far. By the same token, the authors in [9] add a tag for each type by adding an explicit name in every message generated by the protocol. For instance, they use the notation (nonce, N) to indicate that the value N is supposed to be a nonce. This extra information is in fact added by honest agents to precise the intended type of the message and the receiver uses it to recognize the message. This way, tags ensure that any message having originally a given type will not be interpreted as having another type which prevents any possible type-flaw attack. In [10], tagging schemes are used for a decidability proof purpose. The tag is represented as a fresh number that marks all encrypted sub-terms in the protocol. As a result, tagging prevents the unification of different encrypted sub-terms which transforms an undecidable general problem to a decidable particular one even with an infinite number of nonces. In [11], tagging allows to change the inherent non-termination property caused by inference rules. An approach based on Horn clauses [12] is adopted in which attacker abilities and protocol rules are translated into Horn clauses, then the algorithm infers progressively new clauses by resolution. After some resolution steps, the authors show that it is possible to generate an infinite number of sessions which may lead to non-termination. However, after adding a tag on each use of a cryptographic primitive, every encrypted message becomes distinguishable from others, obviously owing to the tag. To practically high-light the effect of tagging, they apply their approach on untagged protocols whose their resolution algorithm does not

terminate (i.e. the Needham–Schroeder shared-key protocol, the Woo-Lam shared key protocol, etc.). Then, they show that after tagging the protocol, messages become unambiguously identified and the infinite loop observed before never happens again. Therefore, the algorithm terminates. In [13], Arapinis et al. give a scheme to transform a secure protocol for a single session, which is a decidable problem, to a secure one for an unbounded number of sessions using tagging. In [14], Cortier et al. show that if a protocol running alone is secure, it remains secure even if it runs simultaneously with other protocols if we carefully add a tag to every encryption in such a way that we can differentiate between all the protocols by adding the name of the protocol for example. Similarly, Bauer et al. [15] show that if a protocol is correct for secrecy with a probability higher than some threshold, a protocol composition remains correct for secrecy provided that protocol messages are tagged. Our work in this paper is one of these efforts with the clear advantage that our proposed theorem helps to prove secrecy statically with no need to go through dynamic complexities.

VII. Conclusion

In this paper, we put forward a new theorem to prove secrecy inside tagged protocols using witness-functions. Then, we run a detailed analysis on a tagged version of the Needham-Schroeder public-key protocol. Finally, we discussed some works pinpointing multiple advantages of protocol tagging.

References

[1] J. Fattahi, M. Mejri, and E. Pricop, "The theory of witness functions," in *Recent Advances in Systems Safety and Security*, ch. 1, pp. 1–19, Switzerland: Springer International Publishing, June 2016.

[2] J. Fattahi, *Analyse des Protocoles Cryptographiques par les Fonctions Témoins*. PhD thesis, Université Laval. Québec. Canada, February 2016.

[3] J. Fattahi, M. Mejri, M. Ziadia, T. Omrani, and E. Pricop, "Witness-functions versus interpretation-functions for secrecy in cryptographic protocols: What to choose?," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2649–2654, Oct 2017.

[4] J. Fattahi, M. Mejri, and H. Houmani, "Secrecy by witness functions," in *Proceedings of the Formal Methods for Security Workshop co-located with the PetriNets-2014 Conference , Tunis, Tunisia, June 23rd, 2014.* (V. Cortier and R. Robbana, eds.), vol. 1158 of *CEUR Workshop Proceedings*, pp. 34–52, CEUR-WS.org, 2014.

[5] M. Debbabi, Y. Legaré, and M. Mejri, "An environment for the specification and analysis of cryptoprotocols," in *14th Annual Computer Security Applications Conference (ACSAC 1998), 7-11 December 1998, Scottsdale, AZ, USA*, pp. 321–332, IEEE Computer Society, 1998.

[6] M. Mejri, *From Type Theory to the Verification of Security Protocols*. PhD thesis, Université Laval. Québec. Canada, December 2000.

[7] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, pp. 198–208, Mar 1983.

[8] J. Fattahi, M. Mejri, and H. Houmani, "Relaxed conditions for secrecy in a role-based specification," *CoRR*, vol. abs/1801.08410, 2018.

[9] J. Heather, G. Lowe, and S. Schneider, "How to prevent type flaw attacks on security protocols," in *Proceedings 13th IEEE Computer Security Foundations Workshop. CSFW-13*, pp. 255–268, 2000.

[10] R. Ramanujam and S. P. Suresh, *Tagging Makes Secrecy Decidable with Unbounded Nonces as Well*, pp. 363–374. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[11] B. Blanchet and A. Podelski, "Verification of cryptographic protocols: tagging enforces termination," vol. 333, pp. 67 – 90, 2005. Foundations of Software Science and Computation Structures.

[12] B. Blanchet, "Using Horn Clauses for Analyzing Security Protocols," in *Formal Models and Techniques for Analyzing Security Protocols* (V. Cortier and S. Kremer, eds.), vol. 5 of *Cryptology and Information Security Series*, pp. 86 – 111, IOS Press, 2011.

[13] M. Arapinis, S. Delaune, and S. Kremer, "From one session to many: Dynamic tags for security protocols," in *Logic for Programming, Artificial Intelligence, and Reasoning* (I. Cervesato, H. Veith, and A. Voronkov, eds.), pp. 128–142, Springer Berlin Heidelberg, 2008.

[14] V. Cortier, J. Delaitre, and S. Delaune, "Safely composing security protocols," in *FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science* (V. Arvind and S. Prasad, eds.), (Berlin, Heidelberg), pp. 352–363, Springer Berlin Heidelberg, 2007.

[15] M. S. Bauer, R. Chadha, and M. Viswanathan, "Composing protocols with randomized actions," in *Principles of Security and Trust* (F. Piessens and L. Viganò, eds.), (Berlin, Heidelberg), pp. 189–210, Springer Berlin Heidelberg, 2016.