Secrecy by Witness-Functions

Jaouhar Fattahi[1], Mohamed Mejri[1] and Hanane Houmani[2]

[1]LSI Group, Laval University, Quebec, Canada
[2] University Hassan II, Morocco

Abstract. In this paper, we introduce a new type of functions to analyze cryptographic protocols statically for the property of secrecy: the Witness-Functions. A Witness-Function is a reliable protocol-dependent function intended to prove the correctness of a protocol through its growth. It bases its calculation on the static part of a message in a role-based specification and ignores the dynamic one by introducing the notion of derivative messages. It offers two interesting bounds that enable an analysis of protocols on an unbounded number of sessions. We give here the way to build these functions and we state the theorem of protocol analysis with the Witness-Functions.

Keywords: Protocol, role-based specification, secrecy, sufficient condition.

1 Motivation and background

In this paper, we introduce a new static approach to analyze cryptographic protocols with witness-functions for the property of secrecy. Intuitively, an increasing protocol preserves the secret. In other words, 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 evaluate 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 algebra. These functions were successful in analyzing Needham-Schroeder protocol. However, a such analysis requires the protocol implementation in CSP [2,3]. 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 from 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 protocols that had been implemented with no respect to this restriction. In the same vein, Houmani and 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 proven 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 and the conclusion made on valid traces, 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 restrictive condition, we can build more functions. Nevertheless, freeing a function from a condition may impel us to take additional precautions. In this work, we introduce the witness-functions to analyze cryptographic protocols. We show how to build them. We show that a witness-function offers two bounds that allow us to get rid of the restrictive condition of full-invariance by substitution in Houmani's work by using derivation techniques. We state finally the theorem of protocol analysis with the Witness-Functions that sets an interesting criterion for the protocol correctness.

Notations

Hereafter, we give some definitions and conventions that will be used throughout the 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. $dec(enc(x, y), y^{-1}) = x$.
  - $\models$: is the inference system of the intruder under the equational theory. Let M be a set of messages and m a message. $M \models m$ means that the intruder is able to infer m from M using her capacity. We extend this notation to traces as following: $\rho \models m$ means that the intruder can infer m from the messages exchanged in the trace ρ.
  - $\mathcal{K}$: 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 $\exists m' \in M. \ulcorner m' \urcorner \sqsupseteq \ulcorner m \urcorner$
+ Let p be a protocol, we denote by $R_G(p)$ the set of the generalized roles extracted from p. A generalized role is a protocol abstraction where the emphasis is put on a particular principal and all the unknown messages, and on which no verification could be done by the agent, are replaced by variables. A generalized role ends always by a sending step and an intruder I is introduced to describe that the received messages and the sent messages are probably sent or received by the intruder. More details about the role-based specification are in [12,13,14]. Hereafter, an example of a variation of NSL protocol written in a role-based specification:

$$\begin{array}{l}
m_1 : A \longrightarrow B : \{N_a.A\}_{k_b} \\
m_2 : B \longrightarrow A : \{B.N_a\}_{k_a}.\{B.N_b\}_{k_a} \\
m_3 : A \longrightarrow B : A.B.\{N_b\}_{k_b}
\end{array}$$

Table 1. A variation of NSL protocol

The generalized roles of this protocol in a role-based specification are $\mathcal{R}_{\mathcal{G}}(p_{NSL})= \{A_{\mathcal{G}}^{1}, A_{\mathcal{G}}^{2}, B_{\mathcal{G}}^{1}, B_{\mathcal{G}}^{2}\}$ where:

$$\begin{array}{rllll}
A_{\mathcal{G}}^{1} = & i.1 & A & \longrightarrow I(B) & : \{N_a^i.A\}_{k_b} \\
A_{\mathcal{G}}^{2} = & i.1 & A & \longrightarrow I(B) & : \{N_a^i.A\}_{k_b} \\
 & i.2 & I(B) & \longrightarrow A & : \{B.N_a^i\}_{k_a}.\{B.X\}_{k_a} \\
 & i.3 & A & \longrightarrow I(B) & : A.B.\{X\}_{k_b}
\end{array}$$

$$\begin{array}{rllll}
B_{\mathcal{G}}^{1} = & i.1 & I(A) & \longrightarrow B & : \{Y.A\}_{k_b} \\
 & i.2 & B & \longrightarrow I(A) & : \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a} \\
B_{\mathcal{G}}^{2} = & i.1 & I(A) & \longrightarrow B & : \{Y.A\}_{k_b} \\
 & i.2 & B & \longrightarrow I(A) & : \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a} \\
 & i.3 & I(A) & \longrightarrow B & : A.B.\{N_b^i\}_{k_b}
\end{array}$$

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.

+ 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 assume that the intruder has the full-control of the net, as described in the Dolev-Yao model [15] with no restriction neither on the size of messages nor on the number of sessions.

2 Increasing protocols are correct with respect to the secrecy property

To analyze a protocol, we need reliable functions to estimate the security level of every atomic message. In this section, we state sufficient 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 21 *(Well-formed interpretation function)*
Let F be an interpretation function.
F is well-formed in $\mathcal{C}$ if:
$\forall M, M_1, M_2 \subseteq \mathcal{M}, \forall \alpha \in \mathcal{A}(\mathcal{M})$:

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

For an atom α in a set of messages M, a well-formed interpretation function returns the bottom value "$\bot$" if α appears in clear in M. It returns for it in the union of two sets, the minimum "$\sqcap$" of the two values calculated in each set seperately. It returns the top value "$\top$" if α does not appear at all in M.

Example 22 *Let $m_1 = \{\alpha\}, m_2 = \{\beta.A\}_{k_{ab}}$, $M = \{m_1, m_2\}$ and F a well-formed interpretation function.*
- $F(\alpha, M) = F(\alpha, \{m_1\} \cup \{m_2\}) = F(\alpha, \{\alpha\}) \sqcap F(\alpha, \{\beta.A\}_{k_{ab}}) = \bot \sqcap \top = \bot$
- $F(\beta, M) = F(\alpha, \{m_1\} \cup \{m_2\}) = F(\beta, \{\alpha\}) \sqcap F(\beta, \{\beta.A\}_{k_{ab}}) = \top \sqcap F(\beta, \{\beta.A\}_{k_{ab}})$
$= F(\beta, \{\beta.A\}_{k_{ab}})$

Definition 23 *(Full-invariant-by-intruder interpretation function)*
Let F be an interpretation function.
F is full-invariant-by-intruder in $\mathcal{C}$ if:
$\forall M \subseteq \mathcal{M}, m \in \mathcal{M}.M \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m).(F(\alpha, m) \sqsupseteq F(\alpha, M)) \vee (\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner)$

An interpretation function F is said to be full-invariant-by-intruder if when it assigns a security level to an atomic message α in a set of messages M, this level cannot be decreased by the intruder. That is to say, the intruder cannot infer another message m from M in which the level of security of α decreases (i.e. $F(\alpha, m) \sqsupseteq F(\alpha, M)$) using her capacity in the context of verification, unless α is initially intended to be known by the intruder (i.e. $\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner$).

Definition 24 *(Reliable interpretation function)*
Let F be an interpretation function and $\mathcal{C}$ be a context.

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

Definition 25 *[F-increasing protocol]*
Let F be an interpretation function and p a protocol.
p is F-increasing in $\mathcal{C}$ if:
$\forall R.r \in R_G(p), \forall \sigma \in \Gamma : \mathcal{X} \to \mathcal{M}_p$ we have:

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

A F-increasing protocol is a protocol 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 is intuitively 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 F), the intruder can never reveal it.

Definition 26 *(Secret disclosure)*
Let p be a protocol and C a context.
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 the context of verification $\mathcal{C}$, to infer a secret α that she is not allowed to know (expressed by: $\ulcorner K(I) \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner$ in the definition 26).

Lemma 27
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 819 in [16] ■

The lemma 27 expressess an intuitive result 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 value cannot be decreased by the intruder using her initial knowledge and received messages since a reliable function is 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 then run by induction on the size of the trace and uses the reliability properties of the interpretation function in every step.

Theorem 21 *(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. See the proof 820 in [16] ■

In this section, we have shown 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. Please notice that compared to the sufficient conditions stated in [11], we have one less. Houmani in [11] 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. 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. The problem of substitution migrates to the protocol and becomes less harder to deal with.

3 Building reliable interpretation functions

As seen in the previous section, to analyze a protocol statically we need reliable interpretation functions to evaluate the level of security of each atom in a message. In this section, we propose a constructive way to build these functions. We first show how to build a generic class of reliable selections inside the protection of the most external key (or simply the external key), then we provide specialized selections that are instances of this class, and finally we show the way to build reliable selection-based interpretation functions. Similar techniques have been used in some previous works and especially in [8,10,11] to build functions based on the direct key of encryption and in [17] to verify correspondences for security in protocols. But first, we introduce the notion of well-protected messages that have interesting properties that we will use in the definition of reliable selections.

3.1 Well-protected messages

We denote by $\mathcal{E}_\mathcal{C}$ the set of encryption functions and by $\overline{\mathcal{E}}_\mathcal{C}$ the complementary set $\Sigma\backslash\mathcal{E}_\mathcal{C}$ in a context of verification $\mathcal{C}$. In the definition 31, we define the application *keys* that returns the encryption keys of any atom α in a message m.

Definition 31 *(Keys)*
Let $M \subseteq \mathcal{M}$, $f \in \Sigma$ and $m \in M$.
We define the application Keys as follows:

$$Keys : \mathcal{A} \times \mathcal{M} \longrightarrow \mathcal{P}(\mathcal{P}(\mathcal{A}))$$

$\forall t_1, t_2 ... t_n$ subterms of m:

$$\begin{array}{ll} Keys(\alpha,\alpha) & = \{\emptyset\} \\ Keys(\alpha,\beta) & = \emptyset, \textit{ if } \ \alpha \neq \beta \ \textit{ and } \beta \in \mathcal{A} \\ Keys(\alpha, f_k(t_1,...,t_n)) & = \{k\} \otimes \bigcup_{i=1}^{n} Keys(\alpha, t_i), \textit{ if } \ f_k \in \mathcal{E}_\mathcal{C} \\ Keys(\alpha, f(t_1,...,t_n)) & = \bigcup_{i=1}^{n} Keys(\alpha, t_i), \textit{ if } f \in \overline{\mathcal{E}}_\mathcal{C} \end{array}$$

We extend the application Keys to sets as follows:

$$\forall M \subseteq \mathcal{M}. Keys(\alpha, M) = \bigcup_{m \in M} Keys(\alpha, m) \textit{ and } Keys(\alpha, \emptyset) = \emptyset.$$

Definition 32 *(Equational theory, rewriting system and normal form)*
We assume that we can transform the equational theory ξ given in the context of verification to a convergent rewriting system $\rightarrow_\xi$ such that:

$$\forall m \in \mathcal{M}, \forall \alpha \in \mathcal{A}(m), \forall l \rightarrow r \in \rightarrow_\xi, \qquad Keys(\alpha, r) \subseteq Keys(\alpha, l) \qquad (3.1.1)$$

We denote by $m_{\Downarrow}$ the normal form of m in $\rightarrow_\xi$.

The normal form of a message in the definition 32 is the one that has the smallest set of encryption keys and eliminates all the unnecessary keys (e.g. $e(k, d(k^{-1}, m)) \rightarrow m$). This kind of rewriting systems orientation poses no problem with the most of equational theories [18,19,20].

In the definition 33 we introduce the application $Access$ where every element of $Access(\alpha, m)$ contains a set of required keys to decrypt α in m after elimination of unnecessary keys by the normal form defined in 32.

Definition 33 *(Access)*
Let $M \subseteq \mathcal{M}$, $f \in \Sigma$ and $m \in M$.
We define the application Accessor as follows:

$$Access : \mathcal{A} \times \mathcal{M} \longrightarrow \mathcal{P}(\mathcal{P}(\mathcal{A}))$$

$\forall t_1, t_2 ... t_n$ subterms of m:

$$\begin{array}{ll}
Access(\alpha, \alpha) & = \{\emptyset\} \\
Access(\alpha, \beta) & = \emptyset, \textit{ if } \ \alpha \neq \beta \ \textit{ and } \beta \in \mathcal{A} \\
Access(\alpha, f_k(t_1, ..., t_n)) & = \{k^{-1}\} \otimes \overset{n}{\underset{i=1}{\cup}} Access(\alpha, t_i), \textit{ if } f_k \in \mathcal{E}_{\mathcal{C}} \textit{ and } f_k(t_1, ..., t_n) = f_k(t_1, ..., t_n)_{\Downarrow} \\
Access(\alpha, f(t_1, ..., t_n)) & = \overset{n}{\underset{i=1}{\cup}} Access(\alpha, t_i), \textit{ if } f \in \overline{\mathcal{E}}_{\mathcal{C}} \textit{ and } f_k(t_1, ..., t_n) = f_k(t_1, ..., t_n)_{\Downarrow} \\
Access(\alpha, f(t_1, ..., t_n)) & = Access(\alpha, f(t_1, ..., t_n)_{\Downarrow}), \textit{ if not.}
\end{array}$$

We extend the application Access to sets as follows:

$$\forall M \subseteq \mathcal{M}. Access(\alpha, M) = \underset{m \in M}{\cup} Access(\alpha, m) \textit{ and } Access(\alpha, \emptyset) = \emptyset.$$

Example 34 *Let m be a message such that: $m = \{\{A.D.\alpha\}_{k_{ab}}.\alpha.\{A.E.\{C.\alpha\}_{k_{ef}}\}_{k_{ab}}\}_{k_{ac}}$;*
$Access(\alpha, m) = \{\{k_{ac}^{-1}, k_{ab}^{-1}\}, \{k_{ac}^{-1}\}, \{k_{ac}^{-1}, k_{ab}^{-1}, k_{ef}^{-1}\}\}$.

In the definition 35, we define a well-protected message. Informally, a well-protected message is a message such that every atom α in it such that $\ulcorner \alpha \urcorner \sqsupset \bot$ is encrypted by at least one key k such that $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$ after elimination of unnecessary keys by the normal form defined in 32.

Definition 35 *(Well-protected message)*
Let $\mathcal{C}$ be context of verification, $m \in \mathcal{M}$, $M \subseteq \mathcal{M}$ and $\alpha \in \mathcal{A}(m)$ such that $\ulcorner \alpha \urcorner \sqsupset \bot$.

We say that α is well-protected in m if:

$$\forall K \in Access(\alpha, m). \ulcorner K \urcorner \sqsupseteq \ulcorner \alpha \urcorner$$

We say that α is well-protected in M if:

$$\forall m \in M. \alpha \textit{ is well-protected in } m$$

We say that m is well-protected in $\mathcal{C}$ if:

$$\forall \alpha \in \mathcal{A}(m). \alpha \textit{ is well-protected in } m$$

We say that M is well-protected in $\mathcal{C}$ if:

$$\forall m \in M. m \textit{ is well-protected in } \mathcal{C}.$$

In the definition 36, we define $Clear(m)$. Informally, $Clear(m)$ is the set of all atoms that appear in clear in m after elimination of unnecessary keys by the normal form defined in 32.

Definition 36 *(Clear)*
Let $m \in \mathcal{M}$ and $M \subset \mathcal{M}$.
$Clear(m) = \{\alpha \in \mathcal{A}(m) | \ \emptyset \in Access(\alpha, m)\}$
We extend this definition to sets as follows:

$$Clear(M) = \underset{m \in M}{\cup} Clear(m)$$

Lemma 37
Let M be a set of well-protected messages in $\mathcal{M}$. We have:

$$M \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m).(\alpha \textit{ is well-protected in } m) \vee (\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner)$$

Proof. See the proof 912 in [16] ■

The lemma 37 states that from a set of well-protected messages, all atomic messages beyond the knowledge of the intruder (i.e. $\ulcorner K(I) \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner$) remain well-protected in any message that the intruder could infer. In fact, since we operate on well-protected messages, every atom such that $\ulcorner \alpha \urcorner \sqsupset \bot$ is encrypted by at least one key k such that $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$. So the intruder must retrieve k^{-1} before she sees α not well-protected in any message. Yet, the key k^{-1}, if it appears in M, should be in its turn encrypted by at least one key k' such that $\ulcorner k'^{-1} \urcorner \sqsupseteq \ulcorner k^{-1} \urcorner$ since we operate on well-protected messages. The proof is then run by induction on the encryption keys.

Lemma 38 *(Lemma of non-disclosure of atomic secrets in well-protected messages)*
Let M be a set of well-protected messages in $\mathcal{M}$ and α an atomic message in M.
We have:

$$M \models_{\mathcal{C}} \alpha \Rightarrow \ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner.$$

Proof. See the proof 914 in [16] ■

3.2 Well-protected messages and increasing protocols

The lemma 38 expresses an important result. It states that a set of well-protected messages never discloses a secret. Thus, an intruder cannot infer from this set of messages something that she is not initially allowed to know. The notion of well-protected messages is linked to the notion of increasing protocols. Indeed, having a reliable interpretation function F and a protocol p, p could not be F-increasing if it does not operate on a set of well-protected messages $\mathcal{M}_p$. Thus, a F-increasing protocol does not leak secrets as established by the theroem 21. This could not happen if $\mathcal{M}_p$ is not well-protected. Thus, an atomic message that is not well-protected in $\mathcal{M}_p$ could be deduced by agents that have the decryption keys even if they are not intented to know it. To be coherent, a reliable interpretation function must give for a non-well-protected atom α in any message m in $\mathcal{M}_p$ a value less than $\ulcorner \alpha \urcorner$ (e.g. $F(\alpha, m) = \bot$).

3.3 Reliable selections in well-protected messages

Now, we will focus on building selections such that when they are composed to suitable homomorphisms, provide reliable interpretation functions. The definition 39 introduces the notion of a well-formed selection and the definition 310 introduces the notion of a full-invariant-by-intruder selection.

Definition 39 *(Well-formed selection)*
Let $M, M_1, M_2 \subseteq \mathcal{M}$ such that M, M_1 and M_2 are well-protected.
Let $S : \mathcal{A} \times \mathcal{M} \longmapsto 2^{\mathcal{A}}$ be a selection.
We say that S is well-formed in $\mathcal{C}$ if:

$$\begin{cases} S(\alpha, \{\alpha\}) & = \mathcal{A}, \\ S(\alpha, M_1 \cup M_2) & = S(\alpha, M_1) \cup S(\alpha, M_2), \\ S(\alpha, M) & = \emptyset, \textit{ if } \alpha \notin \mathcal{A}(M) \end{cases}$$

For an atom α in a set of messages M, a well-formed selection returns all the atoms in $\mathcal{M}$ if $M = \{\alpha\}$. It returns for it in the union of two sets of messages, the union of the two selections performed in each set separately. It returns the empty set if the atom does not appear in M.

Definition 310 *(Full-invariant-by-intruder selection)*
Let $M \subseteq \mathcal{M}$ such that M is well-protected.
Let $S : \mathcal{A} \times \mathcal{M} \longmapsto 2^{\mathcal{A}}$ be a selection.
We say that S is full-invariant-by-intruder in $\mathcal{C}$ if:
$\forall M \subseteq \mathcal{M}, m \in \mathcal{M}$, we have:

$$M \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m).(S(\alpha, m) \subseteq S(\alpha, M)) \vee (\ulcorner K(I) \urcorner \sqsupseteq \ulcorner \alpha \urcorner)$$

The goal of a full-invariant-by-intruder selection is to create a full-invariant-by-intruder function when composed to an appropriate homomorphism that transforms its returned values into security levels. Since a full-invariant-by-intruder function is requested to resist to any attempt of the intruder to generate a message m from any set of messages M in which the level of security of an atom, that she is not allowed to know, decreases compared to it its value in M, a full-invariant-by-intruder selection is requested to resist to any attempt of the intruder to generate a message m from any set of messages M in which the selection associated to an atom, that she is not allowed to know, could be enlarged compared to the selection associated to this atom in M. This fact is expressed by the definition 310.

Definition 311 *(Reliable selection)*
Let $S : \mathcal{A} \times \mathcal{M} \longmapsto 2^{\mathcal{A}}$ be a selection and $\mathcal{C}$ be a context of verification.

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

3.3.1 Reliable selections inside the protection of an external key

Now, we define a generic class of selections that we call S_{Gen}^{EK} and we prove that any instance of this class is $\mathcal{C}$-reliable. Then we instantiate concrete selections from this class.

Definition 312 (S_{Gen}^{EK}: selection inside the protection of an external key)
We denote by S_{Gen}^{EK} the class of all selections S that meet the following conditions:

- $S(\alpha, \alpha) = \mathcal{A}$;
$$\tag{3.3.1}$$
- $S(\alpha, m) = \emptyset$, *if* $\alpha \notin \mathcal{A}(m)$;
$$\tag{3.3.2}$$
- $\forall \alpha \in \mathcal{A}(m),$ *where* $m = f_k(m_1, ..., m_n)$:

$$S(\alpha, m) \subseteq (\underset{1 \leq i \leq n}{\cup} \mathcal{A}(m_i) \cup \{k^{-1}\} \backslash \{\alpha\}) \text{ if } f_k \in \mathcal{E}_{\mathcal{C}} \text{ and } \ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner \text{ and } m = m_{\Downarrow} \tag{3.3.3}$$

- $\forall \alpha \in \mathcal{A}(m),$ *where* $m = f(m_1, ..., m_n)$:

$$S(\alpha, m) = \begin{cases} \underset{1 \leq i \leq n}{\cup} S(\alpha, m_i) & \text{if } f_k \in \mathcal{E}_{\mathcal{C}} \text{ and } \ulcorner k^{-1} \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner \text{ and } m = m_{\Downarrow} \quad (a) \\ \underset{1 \leq i \leq n}{\cup} S(\alpha, m_i) & \text{if } f \in \overline{\mathcal{E}}_{\mathcal{C}} \text{ and } m = m_{\Downarrow} \quad (b) \\ S(\alpha, m_{\Downarrow}) & \text{if } m \neq m_{\Downarrow} \quad (c) \end{cases} \tag{3.3.4}$$

- $S(\alpha, \{m\} \cup M) = S(\alpha, m) \cup S(\alpha, M)$
$$\tag{3.3.5}$$

For an atom α in an encrypted message $m = f_k(m_1, ..., m_n)$, a selection S as defined above returns a subset (see "$\subseteq$" in equation 3.3.3) among atoms that are neighbors of α in m inside the protection of the most external protective key k including its reverse form k^{-1}. The atom α itself is not selected. This set of candidate atoms is denoted by $\underset{1\leq i\leq n}{\cup} \mathcal{A}(m_i) \cup \{k^{-1}\} \backslash \{\alpha\}$ in the equation 3.3.3. The most external protective key (or simply the external key) is the most external one that satisfies $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$. By neighbor of α in m, we mean any atom that travels with it inside the protection of the external key. Outside the protection of the external key, we have:

- no effective selection is performed;
- any two messages joined by an operation other than an encryption by the external key (e.g. a concatenation or an encryption by a weak key) are assimilated to two distinct messages and the selection returns the union of the two selections performed separately in each one (see the equations 3.3.4(a) and 3.3.4(b));
- the selection in two sets of messages is the union of the two selections performed separately in each one(see the equation 3.3.5);
- the selection relative to a clear atom returns all atoms in $\mathcal{M}$ (see the equation 3.3.1);
- the selection relative to an atom that does not appear in a message returns the empty set(see the equation 3.3.2).

S_{Gen}^{EK} defines a generic class of selections since it does not identify what atoms to select precisely inside the protection of the external key. It identifies only the atoms that are candidates for selection and among them we are allowed to return any subset.

Proposition 313
Let $S \in S_{Gen}^{EK}$ and $\mathcal{C}$ be a context of verification.
Let's have a rewriting system $\rightarrow_\xi$ such that $\forall m \in \mathcal{M}, \forall \alpha \in \mathcal{A}(m) \wedge \alpha \notin Clear(m)$, we have:

$$\forall l \rightarrow r \in \rightarrow_\xi, S(\alpha, r) \subseteq S(\alpha, l) \tag{3.3.6}$$

We have:

S is full-invariant-by-intruder in $\mathcal{C}$.

Proof. *See the proof 107 in [16]* ■

Remark 314 *(Scope)*

The condition on the rewriting system $\rightarrow_\xi$ given by the equation 3.3.6 in the definition 313 is introduced to make sure that the selection in the normal form is the smallest among all forms of a given message. This prevents the selection S to select atoms that may be inserted maliciously by the intruder by manipulating the equational theory. Hence, we are sure that all selected atoms by S are honest and do not come by an intruder manipulation of the message. For example, let $m = \{\alpha.C\}_{k_{ab}}$ be a message in a homomorphic cryptography (i.e. $\{\alpha.C\}_{k_{ab}} = \{\alpha\}_{k_{ab}}.\{C\}_{k_{ab}}$). In the form $\{\alpha.C\}_{k_{ab}}$, the selection $S(\alpha, \{\alpha.C\}_{k_{ab}})$ may select C, but in the form $\{\alpha\}_{k_{ab}}.\{C\}_{k_{ab}}$, the selection $S(\alpha, \{\alpha\}_{k_{ab}}.\{C\}_{k_{ab}})$ may not. Then, we must make sure that the rewriting system $\rightarrow_\xi$ we are using is oriented in such way that it chooses the form $\{\alpha\}_{k_{ab}}.\{C\}_{k_{ab}}$ rather than the form $\{\alpha.C\}_{k_{ab}}$ because there is no guarantee that C is a honest neighbor and that it had not been inserted maliciously by the intruder using the homomorphic property in the theory. We assume that the equational theory in the context of verification allows the extraction of a convergent rewriting system that meets this condition.

As for the proposition 313, it is quite easy to verify that by constuction a selection S that is instance of S_{Gen}^{EK} is well-formed. The proof of full-invariance-by-intruder is run by induction on the tree of construction of any message. The main idea of the proof is that the selection of the candidate atoms is performed inside the encryption by the most external protective key and therfore beyond the knowledge of the intruder, thus the intruder cannot alter when she does not detain this key. Besides, according to the lemma 38, in a set of well-protected messages the intruder can never infer this key since it is atomic. The intruder cannot neither use the equational theory to alter this selection for the reasons given in the remark 314. Therefore the set of atoms returned by S cannot be altered (enlarged) by the intruder as required by a full-invariant-by-intruder selection.

Example 315 *Let α be an atomic message and m a message such that $\ulcorner\alpha\urcorner = \{A, B\}$ and $m = \{A.C.\alpha.D\}_{k_{ab}}$. Let S_1, S_2 and S_3 be three selections such that: $S_1(\alpha, m) = \{k_{ab}^{-1}\}$, $S_2(\alpha, m) = \{A, C, k_{ab}^{-1}\}$ and $S_3(\alpha, m) = \{A, C, D, k_{ab}^{-1}\}$. These three selections are $\mathcal{C}$-reliable.*

3.4 Instantiation of reliable selections from the class S^{EK}_{Gen}

Now that we defined a generic class of reliable selections S^{EK}_{Gen}, we will instantiate some concrete selections from it, that are naturally reliable. Instantiating S^{EK}_{Gen} consists in defining selections that return precise sets of atoms among the candidates allowed by S^{EK}_{Gen}.

3.4.1 The selection S^{EK}_{MAX} : is the instance of the class S^{EK}_{Gen} that returns for an atom in a message m all its neighbors, that are principal identities, inside the protection of the external protective key k in addition to its reverse form k^{-1}. (MAX means: the MAXimum of principal identities)

3.4.2 The selection S^{EK}_{EK} : is the instance of the class S^{EK}_{Gen} that returns for an atom in a message m only the reverse form of the external protective key. (EK means: External Key)

3.4.3 The selection S^{EK}_{N} : is the instance of the class S^{EK}_{Gen} that returns only its neighbors, that are principal identities, inside the protection of the external protective key. (N means: Neighbors)

Several other selections could be defined such that the selections inside the most internal key or inside all the keys together since they are all subselections inside the external key.

Example 316
Let α be an atom and m a message such that: $\ulcorner\alpha\urcorner = \{A, C\}$ and $m = \{\{\{\alpha.C\}_{k_{ab}}.B\}_{k_{ac}}.D\}_{k_{ad}}$
$S^{EK}_{MAX}(\alpha, m) = \{C, B, k_{ac}^{-1}\}$; $S^{EK}_{EK}(\alpha, m) = \{k_{ac}^{-1}\}$; $S^{EK}_{N}(\alpha, m) = \{C, B\}$

3.5 Specialized $\mathcal{C}$-reliable selection-based interpretation functions

The following proposition gives the way to transform the elements returned by a selection to security levels.

Proposition 317
Let ψ be a homomorphism defined as follows:

$$\begin{array}{lcl} \psi : (2^{\mathcal{A}})^{\subseteq} & \mapsto & \mathcal{L}^{\sqsupseteq} \\ M & \mapsto & \begin{cases} \top & \text{if } M = \emptyset \\ \underset{\alpha \in M}{\sqcap} \psi(\alpha) & \text{if not.} \end{cases} \end{array}$$

$$\text{such that: } \psi(\alpha) = \begin{cases} \{\alpha\} & \text{if } \alpha \in \mathcal{I} \text{ (Principal Identities)} \\ \ulcorner\alpha\urcorner & \text{if not.} \end{cases}$$

We have: $F^{EK}_{MAX} = \psi \circ S^{EK}_{MAX}$, $F^{EK}_{EK} = \psi \circ S^{EK}_{EK}$ and $F^{EK}_{N} = \psi \circ S^{EK}_{N}$ are $\mathcal{C}$-reliable.

Proof. See the proof 1110 in [16] ■

The homomorphism ψ defined in the proposition 317 returns for a principal in a selection, its identity. It returns for a key its level of security in the context of verification. ψ ensures the mapping from the operator "$\subseteq$" to the operator "$\sqsupseteq$" in the lattice which enables an interpretation function to inherit the full-invariance-by-intruder from its associated selection. It ensures also the mapping from the operator "$\cup$" to the operator "$\sqcap$" in the lattice, which enables an interpretation function to be well-formed when its associated selection is well-formed. Generally, any function $\psi \circ S$ remains reliable for any selection S that is an instance of S_{Gen}^{EK}.

Example 318
Let α be an atom, m a message and k_{ab} a key such that: $\ulcorner \alpha \urcorner = \{A, B\}$; $m = \{A.C.\alpha.D\}_{k_{ab}}$;
$k_{ab}^{-1} = k_{ab}$; $\ulcorner k_{ab} \urcorner = \{A, B\}$;
$S_{EK}^{EK}(\alpha, m) = \{k_{ab}^{-1}\}$; $S_{N}^{EK}(\alpha, m) = \{A, C, D\}$; $S_{MAX}^{EK}(\alpha, m) = \{A, C, D, k_{ab}^{-1}\}$;
$F_{EK}^{EK}(\alpha, m) = \psi \circ S_{EK}^{EK}(\alpha, m) = \ulcorner k_{ab}^{-1} \urcorner = \{A, B\}$; $F_{N}^{EK}(\alpha, m) = \psi \circ S_{N}^{EK}(\alpha, m) = \{A, C, D\}$;
$F_{MAX}^{EK}(\alpha, m) = \psi \circ S_{MAX}^{EK}(\alpha, m) = \{A, C, D\} \sqcap \ulcorner k_{ab}^{-1} \urcorner = \{A, C, D\} \sqcap \{A, B\} = \{A, C, D, B\}$.

4 The Witness-Functions

In the previous section, we presented a constructive way of a class of selection-based functions that have the required properties to analyze protocols statically. Unfortunatly, they operate only on valid traces that contain closed messages. However, a static protocol analysis should run over a finite set of messages of the generalized roles of the protocol because the set of valid traces is infinite. The finite set of the generalized roles contains variables. The functions we defined are not "enough prepared" to analyze messages with variables because they are not supposed to be full-invariant by substitution (or stable by substitution) [21,22,23]. The full-invariance by substitution is the property-bridge that allows us to perform an analysis over messages with variables and propagate the conclusion made-on to closed messages. In the following section, we deal with the substitution question. We introduce the notion of derivation to reduce the impact of variables and we build the witness-functions that operate on derivative messages rather than messages themselves. As we will see, the witness-functions provide two interesting bounds that are independent of all substitutions and hence we solve the problem of substitution. We last define a criterion of protocol correctness based on these two bounds. The fact that the criterion is independent of all substitutions enables an analysis to be run on generalized roles and the decision made-on to be exported to valid traces.

4.1 Derivative message

Let $m, m_1, m_2 \in \mathcal{M}$; $\mathcal{X}_m = Var(m)$; $S_1, S_2 \subseteq 2^{\mathcal{X}_m}$; $\alpha \in \mathcal{A}(m)$; $X, Y \in \mathcal{X}_m$ and ϵ be the empty message.

Definition 41 *(Derivation)*
We define the derivative message as follows:

$$\begin{array}{ll}
\partial_X \epsilon & = \epsilon \\
\partial_X \alpha & = \alpha \\
\partial_X X & = \epsilon \\
\partial_X Y & = Y \\
\partial_X f(m) & = f(\partial_X m), f \in \mathcal{E}_\mathcal{C} \cup \overline{\mathcal{E}}_\mathcal{C} \\
\partial_{\{X\}} m & = \partial_X m \\
\partial[\overline{X}] m & = \partial_{\{\mathcal{X}_m \backslash X\}} m \\
\partial_{S_1 \cup S_2} m & = \partial_{S_2 \cup S_1} m = \partial_{S_1} \partial_{S_2} m = \partial_{S_2} \partial_{S_1} m
\end{array}$$

For the sake of simplification, we denote by ∂m the expression $\partial_{\mathcal{X}_m} m$. The operation of derivation in the definition 41 (denoted by ∂) is used to eliminate variables in a message.

$\partial_X m$ consists in eliminating the variable X in m. $\partial[\overline{X}]m$ consists in eliminating all variables, except X, in m. Hence, X when overlined is considered as a constant in m. ∂m consists in eliminating all the variables in m.

Example 42 *Let $m = \{A.X.Y\}_{k_{ab}}$ where A and k_{ab} are static and X and Y are two variables. We have: $\partial m = \{A\}_{k_{ab}}$; $\partial_X m = \{A.Y\}_{k_{ab}}$; $\partial[\overline{X}]m = \{A.X\}_{k_{ab}}$*

Definition 43
Let $m \in \mathcal{M}_p^{\mathcal{G}}$, $X \in \mathcal{X}_m$ and $m\sigma$ be a closed message.
For all $\alpha \in \mathcal{A}(m\sigma)$, $\sigma \in \Gamma$, we denote by:

$$F(\alpha, \partial[\overline{\alpha}]m\sigma) = \begin{cases} \top & \textit{if } \alpha \notin \mathcal{A}(m\sigma), \\ F(\alpha, \partial m) & \textit{if } \alpha \in \mathcal{A}(\partial m), \\ F(X, \partial[\overline{X}]m) & \textit{if } \alpha \in \mathcal{A}(X\sigma) \wedge \alpha \notin \mathcal{A}(\partial m). \end{cases}$$

A message m in a generalized role is composed of two parts: a static part and a dynamic part. The dynamic part is described by variables. For an atom α in the static part (i.e. ∂m), $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ removes the variables in m and gives it the value $F(\alpha, \partial m)$. For anything that is not an atom of the static part, so comes by substitution of some variable X in m, $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ considers it as the variable itself, treated as a constant (as a block), and gives it the value $F(X, \partial[\overline{X}]m)$. It gives the top value for any atom that does not appear in $m\sigma$. For any F such that its associated selection is an instance of the class S_{Gen}^{EK}, $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ depends only on the static part of m since α is never selected. The introduction of the derivation could suggest that we give to $F(\alpha, m\sigma)$ the value of $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ and hence we neutralize the variable effects. Unfortunately, this does not happen without undesirable "side-effects" because derivation causes a "loss of details". Let's examine the following case in the example 44.

Example 44
Let m_1 and m_2 be two messages of a generalized role of a protocol p such that $m_1 = \{\alpha.B.X\}_{k_{ad}}$ and $m_2 = \{\alpha.Y.C\}_{k_{ad}}$ and $\ulcorner\alpha\urcorner = \{A, D\}$. Let $m = \{\alpha.B.C\}_{k_{ad}}$ be in a valid trace generated by p.

$$F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m) = \begin{cases} \{B, A, D\} & \textit{if } m \textit{ comes by the substitution of } X \textit{ by } C \textit{ in } m_1 \\ \{A, D, C\} & \textit{if } m \textit{ comes by the substitution of } Y \textit{ by } B \textit{ in } m_2 \end{cases}$$

Hence $F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m)$ is not even a function on the closed message m since it may return more than one image for the same preimage. This leads us straightly to the witness-functions.

4.2 The Witness-Functions

Definition 45 *(Witness-Function)*
Let $m \in \mathcal{M}_p^{\mathcal{G}}$, $X \in \mathcal{X}_m$ and $m\sigma$ be a closed message.
Let p be a protocol and F be a $\mathcal{C}$-reliable interpretation function.
We define a witness-function $\mathcal{W}_{p,F}$ for all $\alpha \in \mathcal{A}(m\sigma)$, $\sigma \in \Gamma$, as follows:

$$\mathcal{W}_{p,F}(\alpha, m\sigma) = \underset{\substack{m' \in \mathcal{M}_p^{\mathcal{G}} \\ \exists \sigma' \in \Gamma . m'\sigma' = m\sigma}}{\sqcap} F(\alpha, \partial[\overline{\alpha}]m'\sigma')$$

$\mathcal{W}_{p,F}$ is said to be a witness-function inside the protection of an external key when F is an interpretation function such that its associated selection is an instance of the class S_{Gen}^{EK}.

As seen in the example 44, the application defined in 43 is not necessary a function in the set $\mathcal{M}_p^{\mathcal{G}}$ of messages generated by the generalized roles of p since a valid trace could have more than one source in $\mathcal{M}_p^{\mathcal{G}}$ such that each source has a different static part. A witness-function is though a function since it looks for all the sources of any closed message and takes the minimum (the union). This minimum exists and is unique in the finite set $\mathcal{M}_p^{\mathcal{G}}$. Although a witness-function is protocol-dependent since it depends on messages in the generalized roles of the protocol, it is built in a standard way for any pair (protocol, interpretation function) in input.

Remark 46 *For a witness-function inside the protection of an external key, since its associated interpretation function calculates the security level of an atom always in a message m having an encryption pattern, i.e. when $f_k \in \mathcal{E_C}$, the search of all sources of m in the set $\{m' \in \mathcal{M}_p^{\mathcal{G}} | \exists \sigma' \in \Gamma . m'\sigma' = m\sigma\}$ is reduced to a search in the encryption patterns in $\mathcal{M}_p^{\mathcal{G}}$.*

4.3 Inheritance of reliability properties from F

Proposition 47
Let $\mathcal{W}_{p,F}$ be a witness-function inside the protection of an external key.
We have:

$$\mathcal{W}_{p,F} \textit{ inherits reliability from } F$$

Proof. See the proof 122 in [16] ■

The selection associated with a witness-function inside the protection of an external key is the union of selections associated with the interpretation function F restricted to derivative messages. It is easy to verify that a witness-function is by construction well-formed. For the full-invariance-by-intruder property, since the derivation just removes variables and since each selection in the union returns a subset among acceptable candidates, then the union itself returns a subset among acceptable candidates (the union of subsets is a subset). Therefore the selection associated with a witness-function remains an instance of the class S_{Gen}^{EK} and so full-invariant-by-intruder. Since the witness-function is the composition of the homomorphism associated with F and an instance of the class S_{Gen}^{EK}, then it is reliable.

Example 48
Let $\mathcal{M}_p^{\mathcal{G}} = \{\{\alpha.B.X\}_{k_{ad}}, \{\alpha.Y.C\}_{k_{ad}}, \{A.Z\}_{k_{bc}}\}$; $m_1 = \{\alpha.B.C\}_{k_{ad}}$; $Var(\mathcal{M}_p^{\mathcal{G}}) = \{X, Y, Z\}$.

$\mathcal{W}_{p,F_{MAX}^{EK}}(\alpha, m_1)$
$=$ *{Definition 45}*
$$\underset{\substack{m' \in \mathcal{M}_p^{\mathcal{G}} \\ \exists \sigma' \in \Gamma . m'\sigma' = m_1}}{\sqcap} F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m'\sigma') = \underset{\substack{\{\{\alpha.B.X\}_{k_{ad}}, \{\alpha.Y.C\}_{k_{ad}}\} \\ \sigma' = \{X \longmapsto C, Y \longmapsto B\}}}{\sqcap} F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m'\sigma')$$
$= \{\mathcal{W}_{p,F_{MAX}^{EK}}$ *is well-formed from the proposition 47* $\}$
$F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]\{\alpha.B.X\}_{k_{ad}}[X \longmapsto C]) \sqcap F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]\{\alpha.Y.C\}_{k_{ad}}[Y \longmapsto B])$
$=$*{Definition 43 and derivation in 41}*
$F_{MAX}^{EK}(\alpha, \{\alpha.B\}_{k_{ad}}) \sqcap F_{MAX}^{EK}(\alpha, \{\alpha.C\}_{k_{ad}})$
$=$ *{Definition of* F_{MAX}^{EK}*}*
$\{B, A, D\} \cup \{C, A, D\} = \{B, A, D, C\}$

4.4 Bounding a Witness-Function

Lemma 49
Let $m \in \mathcal{M}_p^{\mathcal{G}}$ and $\mathcal{W}_{p,F}$ be a witness-function inside the protection of an external key.
$\forall \sigma \in \Gamma, \forall \alpha \in \mathcal{A}(\mathcal{M}_p)$ we have:

$$F(\alpha, \partial[\overline{\alpha}]m) \sqsupseteq \mathcal{W}_{p,F}(\alpha, m\sigma) \sqsupseteq \underset{\substack{m' \in \mathcal{M}_p^{\mathcal{G}} \\ \exists \sigma' = mgu(m', m)}}{\sqcap} F(\alpha, \partial[\overline{\alpha}]m'\sigma')$$

Proof. See the proof 124 in [16] ■

The upper bound of a witness-function estimates the security level of an atom from one *confirmed* source of the closed message, the witness-function it-self estimates it from the *exact* sources of the closed message (i.e. when the protocol is executed), and the lower bound estimates it from all *likely* sources of the closed message. The unification in the lower bound "hunts" the candidates from all the *likely* sources of the closed message in the protocol.

4.5 Sufficient condition for protocol correctness with a Witness-Function

Now, it is time to state the protocol analysis with a Witness-Function theorem that states a criterion for protocol correctness with respect to the secrecy property which is independent of all substitutions.

Theorem 41 *(Protocol analysis with a Witness-Function)*
Let $\mathcal{W}_{p,F}$ be a witness-function inside the protection of an external key.
A sufficient condition of correctness of p with respect to the secrecy property is:
$\forall R.r \in R_G(p), \forall \alpha \in \mathcal{A}(r^+)$ *we have:*

$$\underset{\substack{m' \in \mathcal{M}_p^{\mathcal{G}} \\ \exists \sigma' = mgu(m', r^+)}}{\sqcap} F(\alpha, \partial[\overline{\alpha}]m'\sigma') \sqsupseteq \ulcorner \alpha \urcorner \sqcap F(\alpha, \partial[\overline{\alpha}]R^-)$$

Proof. See the proof 125 in [16] ■

Thanks to the independence of the criterion stated by the theorem 41 of all substitutions, any decision made on the generalized roles can be transmitted to valid traces.

5 NSL protocol analysis with a witness-function

Hereafter, we analyze the NSL protocol given in Table 1 with a witness-function.

Let's have a context of verification such that: $\ulcorner A \urcorner = \bot$; $\ulcorner B \urcorner = \bot$; $\ulcorner N_a^i \urcorner = \{A, B\}$ (secret shared between A and B); $\ulcorner N_b^i \urcorner = \{A, B\}$ (secret shared between A and B); $\ulcorner k_a^{-1} \urcorner = \{A\}$; $\ulcorner k_b^{-1} \urcorner = \{B\}$; $(\mathcal{L}, \sqsupseteq, \sqcup, \sqcap, \bot, \top) = (2^{\mathcal{I}}, \subseteq, \cap, \cup, \mathcal{I}, \emptyset)$; $\mathcal{I} = \{I, A, B, A_1, A_2, B_1, B_2, ...\}$;
The set of messages generated by the protocol is $\mathcal{M}_p^{\mathcal{G}} = \{\{N_{A_1}.A_1\}_{k_{B_1}}, \{B_2.N_{A_2}\}_{k_{A_2}}, \{B_3.X_1\}_{k_{A_3}}, \{X_2\}_{k_{B_4}}, \{Y_1.A_4\}_{k_{B_5}}, \{B_6.Y_2\}_{k_{A_5}}, \{B_7.N_{B_7}\}_{k_{A_6}}, \{N_{B_8}\}_{k_{B_8}}\}$
The variables are denoted by X_1, X_2, Y_1 and Y_2;
The static names are denoted by N_{A_1}, A_1, k_{B_1}, B_2, N_{A_2}, k_{A_2}, B_3, k_{A_3}, k_{B_4}, A_4, k_{B_5}, B_6, k_{A_5}, B_7, N_{B_7}, k_{A_6}, N_{B_8} and k_{B_8}.

After elimination of duplicates, $\mathcal{M}_p^{\mathcal{G}} = \{\{N_{A_1}.A_1\}_{k_{B_1}}, \{B_2.N_{A_2}\}_{k_{A_2}}, \{B_3.X_1\}_{k_{A_3}}, \{X_2\}_{k_{B_4}}, \{Y_1.A_4\}_{k_{B_5}}, \{B_7.N_{B_7}\}_{k_{A_6}}, \{N_{B_8}\}_{k_{B_8}}\}$
Let's select the Witness-Function as follows:
$p = NSL$; $F = F_{MAX}^{EK}$; $\mathcal{W}_{p,F}(\alpha, m\sigma) = \underset{\substack{m' \in \mathcal{M}_p^{\mathcal{G}} \\ \exists \sigma'.m'\sigma' = m\sigma}}{\sqcap} F(\alpha, \partial[\overline{\alpha}]m'\sigma')$;

Let's denote the lower bound of the Witness-Function in the theorem 41 by:

$$\mathcal{W}'_{p,F}(\alpha, r^+) = \underset{\substack{m' \in \mathcal{M}_p^{\mathcal{G}} \\ \exists \sigma' = mgu(m', r^+)}}{\sqcap} F(\alpha, \partial[\overline{\alpha}]m'\sigma')$$

Please notice that all messages are in their normal form since no equational theory is defined. Principal identities in the context are not analyzed since they are declared public. The protocol is analyzed for the property of secrecy only.

5.1 Analysis of the generalized role of A

As defined in the generalized roles of p, an agent A can participate in two consequent sessions: S^i and S^j such that $j > i$. In the former session S^i, the agent A receives nothing and sends the message $\{N_a^i.A\}_{k_b}$. In the consequent session S^j, she receives the message $\{B.N_a^i\}_{k_a}.\{B.X\}_{k_a}$ and she sends the message $A.B.\{X\}_{k_b}$. This is described by the following schema:

$$S^i : \frac{\square}{\{N_a^i.A\}_{k_b}} \qquad\qquad S^j : \frac{\{B.N_a^i\}_{k_a}.\{B.X\}_{k_a}}{A.B.\{X\}_{k_b}}$$

-Analysis of the messages exchanged in the session S^i:

1- For any N_a^i:
a- When sending: $r_{S^i}^{+} = \{N_a^i.A\}_{k_b}$ *(in a sending step, the lower bound is used)*
$\forall N_a^i.\{m' \in \mathcal{M}_p^{\mathcal{G}} | \exists \sigma' = mgu(m', r_{S^i}^{+})\}$
$= \forall N_a^i.\{m' \in \mathcal{M}_p^{\mathcal{G}} | \exists \sigma' = mgu(m', \{N_a^i.A\}_{k_b})\}$
$= \{(\{N_{A_1}.A_1\}_{k_{B_1}}, \sigma'_1), (\{X_2\}_{k_{B_4}}, \sigma'_2), (\{Y_1.A_4\}_{k_{B_5}}, \sigma'_3)\}$ such that:

$$\begin{cases} \sigma'_1 = \{N_{A_1} \longmapsto N_a^i, A_1 \longmapsto A, k_{B_1} \longmapsto k_b\} \\ \sigma'_2 = \{X_2 \longmapsto N_a^i.A, k_{B_4} \longmapsto k_b\} \\ \sigma'_3 = \{Y_1 \longmapsto N_a^i, A_4 \longmapsto A, k_{B_5} \longmapsto k_b\} \end{cases}$$

$\mathcal{W}'_{p,F}(N_a^i, \{N_a^i.A\}_{k_b})$
$= \{$Definition of the lower bound of the Witness-Function$\}$
$F(N_a^i, \partial[\overline{N_a^i}]\{N_{A_1}.A_1\}_{k_{B_1}}\sigma'_1) \sqcap F(N_a^i, \partial[\overline{N_a^i}]\{X_2\}_{k_{B_4}}\sigma'_2) \sqcap F(N_a^i, \partial[\overline{N_a^i}]\{Y_1.A_4\}_{k_{B_5}}\sigma'_3)$
$= \{$Setting the static neighborhood$\}$
$F(N_a^i, \partial[\overline{N_a^i}]\{N_a^i.A\}_{k_b}\sigma'_1) \sqcap F(N_a^i, \partial[\overline{N_a^i}]\{X_2\}_{k_b}\sigma'_2) \sqcap F(N_a^i, \partial[\overline{N_a^i}]\{Y_1.A\}_{k_b}\sigma'_3)$
$= \{$Definition 43$\}$
$F(N_a^i, \{N_a^i.A\}_{k_b}) \sqcap F(X_2, \partial[\overline{X_2}]\{X_2\}_{k_b}) \sqcap F(Y_1, \partial[\overline{Y_1}]\{Y_1.A\}_{k_b})$
$= \{$Derivation in the definition 41$\}$
$F(N_a^i, \{N_a^i.A\}_{k_b}) \sqcap F(X_2, \{X_2\}_{k_b}) \sqcap F(Y_1, \{Y_1.A\}_{k_b})$
$= \{$Since $F = F_{MAX}^{EK}\}$
$\{A, B\} \cup \{B\} \cup \{A, B\} = \{A, B\}(1.0)$

b- When receiving: $R_{S^i}^{-} = \emptyset$ *(in a receiving step, the upper bound is used)*
$F(N_a^i, \partial[\overline{N_a^i}]\emptyset) = F(N_a^i, \emptyset) = \top$ (1.1)

2- Compliance with the correctness criterion stated in the theorem 41:
From (1.0) and (1.1), we have: $\mathcal{W}'_{p,F}(N_a^i, \{N_a^i.A\}_{k_b}) = \{A, B\} \sqsupseteq \ulcorner N_a^i \urcorner \sqcap F(N_a^i, \emptyset) = \{A, B\}$ (1.2)
From (1.2) we have: the messages exchanged in the session S^i respect the correctness criterion stated in the theorem 41. (I)

-Analysis of the messages exchanged in the session S^j:

1- For any X:
a- When sending: $r_{S^j}^{+} = A.B.\{X\}_{k_b}$
$\mathcal{W}'_{p,F}(X, A.B.\{X\}_{k_b}) = \mathcal{W}'_{p,F}(X, A) \sqcap \mathcal{W}'_{p,F}(X, B) \sqcap \mathcal{W}'_{p,F}(X, \{X\}_{k_b}) = \top \sqcap \top \sqcap \mathcal{W}'_{p,F}(X, \{X\}_{k_b}) = \mathcal{W}'_{p,F}(X, \{X\}_{k_b})$ (2.0)
$\forall X.\{m' \in \mathcal{M}_p^{\mathcal{G}} | \exists \sigma' = mgu(m', \{X\}_{k_b})\} = \{(\{X_2\}_{k_{B_4}}, \sigma'_1)\}$ such that:

$$\sigma'_1 = \{X_2 \longmapsto X, k_{B_4} \longmapsto k_b\}$$

$\mathcal{W}'_{p,F}(X, \{X\}_{k_b})$
$= \{$Definition of the lower bound of the Witness-Function$\}$

$F(X, \partial[\overline{X}]\{X_2\}_{k_{B_4}}\sigma'_1)$
$= \{\text{Setting the static neighborhood}\}$
$F(X, \partial[\overline{X}]\{X_2\}_{k_b}\sigma'_1)$
$= \{\text{Definition 43}\}$
$F(X_2, \partial[\overline{X_2}]\{X_2\}_{k_b})$
$= \{\text{Derivation in the definition 41}\}$
$F(X_2, \{X_2\}_{k_b})$
$= \{\text{Since } F = F_{MAX}^{EK}\}$
$\{B\}$ (2.1)

b- When receiving: $R_{S^j}^- = \{B.N_a^i\}_{k_a}.\{B.X\}_{k_a}$ *(in a receiving step, the upper bound is used)*
$F(X, \partial[\overline{X}]\{B.N_a^i\}_{k_a}.\{B.X\}_{k_a}) = F(X, \partial[\overline{X}]\{B.N_a^i\}_{k_a}) \sqcap F(X, \partial[\overline{X}]\{B.X\}_{k_a}) =$
$F(X, \{B.N_a^i\}_{k_a}) \sqcap F(X, \{B.X\}_{k_a}) = \top \sqcap \{A, B\} = \{A, B\}$ (2.2)

3-Compliance with the correctness criterion stated in the theorem 41:
From (2.0), (2.1) and (2.2), we have:
$\mathcal{W}'_{p,F}(X, A.B.\{X\}_{k_b}) = \{B\} \sqsupseteq \ulcorner X \urcorner \sqcap F(X, \partial[\overline{X}]\{B.N_a^i\}_{k_a}.\{B.X\}_{k_a}) = \ulcorner X \urcorner \cup \{A, B\}$ (2.3)
From (2.3), we have: the messages exchanged in the session S^j respect the correctness criterion stated in the theorem 41. (II)
From (I) and (II), the messages exchanged in the generalized role of A respect the correctness criterion stated in the theorem 41. (III)

5.2 Analysis of the generalized role of B

As defined in the generalized roles of p, an agent B can participate in a session ${S'}^i$, in which she receives the message $\{Y.A\}_{k_b}$ and she sends the message $\{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}$. This is described by the following schema:

$${S'}^i : \frac{\{Y.A\}_{k_b}}{\{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}}$$

1- For any N_b^i:
a- When sending: $r_{{S'}^i}^+ = \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}$ *(in a sending step, the lower bound is used)*
$\mathcal{W}'_{p,F}(N_b^i, \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}) = \mathcal{W}'_{p,F}(N_b^i, \{B.Y\}_{k_a}) \sqcap \mathcal{W}'_{p,F}(N_b^i, \{B.N_b^i\}_{k_a}) =$
$\top \sqcap \mathcal{W}'_{p,F}(N_b^i, \{B.N_b^i\}_{k_a}) = \mathcal{W}'_{p,F}(N_b^i, \{B.N_b^i\}_{k_a})$ (3.0)
$\forall N_b^i.\{m' \in \mathcal{M}_p^{\mathcal{G}} | \exists \sigma' = mgu(m', \{B.N_b^i\}_{k_a})\}$
$= \{(\{B_3.X_1\}_{k_{A_3}}, \sigma'_1), (\{X_2\}_{k_{B_4}}, \sigma'_2), (\{B_7.N_{B_7}\}_{k_{A_6}}, \sigma'_3)\}$ such that:

$$\begin{cases} \sigma'_1 = \{B_3 \longmapsto B, X_1 \longmapsto N_b^i, k_{A_3} \longmapsto k_a\} \\ \sigma'_2 = \{X_2 \longmapsto B.N_b^i, k_{B_4} \longmapsto k_a\} \\ \sigma'_3 = \{B_7 \longmapsto B, N_{B_7} \longmapsto N_b^i, k_{A_6} \longmapsto k_a\} \end{cases}$$

$\mathcal{W}'_{p,F}(N_b^i, \{B.N_b^i\}_{k_a})$
$= \{\text{Definition of the lower bound of the Witness-Function}\}$
$F(N_b^i, \partial[\overline{N_b^i}]\{B_3.X_1\}_{k_{A_3}}\sigma'_1) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{X_2\}_{k_{B_4}}\sigma'_2) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{B_7.N_{B_7}\}_{k_{A_6}}\sigma'_3)$
$= \{\text{Setting the static neighborhood}\}$
$F(N_b^i, \partial[\overline{N_b^i}]\{B.X_1\}_{k_a}\sigma'_1) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{X_2\}_{k_a}\sigma'_2) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{B.N_b^i\}_{k_a}\sigma'_3)$
$= \{\text{Definition 43}\}$
$F(X_1, \partial[\overline{X_1}]\{B.X_1\}_{k_a}) \sqcap F(X_2, \partial[\overline{X_2}]\{X_2\}_{k_a}) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{B.N_b^i\}_{k_a})$
$= \{\text{Derivation in the definition 41}\}$
$F(X_1, \{B.X_1\}_{k_a}) \sqcap F(X_2, \{X_2\}_{k_a}) \sqcap F(N_b^i, \{B.N_b^i\}_{k_a})$
$= \{\text{Since } F = F_{MAX}^{EK}\}$
$\{A, B\} \cup \{A\} \cup \{A, B\} = \{A, B\}$ (3.1)

b- When receiving: $R^{-}_{S'^{i}} = \{Y.A\}_{k_b}$ *(in a receiving step, the upper bound is used)*
$F(N_b^i, \partial[\overline{N_b^i}]\{Y.A\}_{k_b}) = \top$ (3.2)

2- For any Y:
a- When sending: $r^{+}_{S'^{i}} = \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}$ *(in a receiving step, the upper bound is used)*
$\mathcal{W}'_{p,F}(Y, \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}) = \mathcal{W}'_{p,F}(Y, \{B.Y\}_{k_a}) \sqcap \mathcal{W}'_{p,F}(Y, \{B.N_b^i\}_{k_a}) =$
$\mathcal{W}'_{p,F}(Y, \{B.Y\}_{k_a}) \sqcap \top = \mathcal{W}'_{p,F}(Y, \{B.Y\}_{k_a})$ (3.3)

$\forall Y.\{m' \in \mathcal{M}_p^{\mathcal{G}} | \exists \sigma' = mgu(m', \{B.Y\}_{k_a})\} = \{(\{B_3.X_1\}_{k_{A_3}}, \sigma_1), (\{X_2\}_{k_{B_4}}, \sigma_2)\}$ such that:

$$\begin{cases} \sigma'_1 = \{B_3 \longmapsto B, X_1 \longmapsto Y, k_{A_3} \longmapsto k_a\} \\ \sigma'_2 = \{X_2 \longmapsto B.Y, k_{B_4} \longmapsto k_a\} \end{cases}$$

$\mathcal{W}'_{p,F}(Y, \{B.Y\}_{k_a})$
$= \{$Definition of the lower bound of the Witness-Function$\}$
$F(Y, \partial[\overline{Y}]\{B_3.X_1\}_{k_{A_3}}\sigma'_1) \sqcap F(Y, \partial[\overline{Y}]\{X_2\}_{k_{B_4}}\sigma'_2)$
$= \{$Setting the static neighborhood$\}$
$F(Y, \partial[\overline{Y}]\{B.X_1\}_{k_a}\sigma'_1) \sqcap F(Y, \partial[\overline{Y}]\{X_2\}_{k_a}\sigma'_2) =$
$= \{$Definition 43$\}$
$F(X_1, \partial[\overline{X_1}]\{B.X_1\}_{k_a}) \sqcap F(X_2, \partial[\overline{X_2}]\{X_2\}_{k_a})$
$= \{$Derivation in the definition 41$\}$
$F(X_1, \{B.X_1\}_{k_a}) \sqcap F(X_2, \{X_2\}_{k_a})$
$= \{$Since $F = F_{MAX}^{EK}\}$
$\{A, B\} \cup \{A\} = \{A, B\}$ (3.4)

b- When receiving: $R^{-}_{S'^{i}} = \{Y.A\}_{k_b}$ *(in a receiving step, the upper bound is used)*
$F(Y, \partial[\overline{Y}]\{Y.A\}_{k_b}) = \{A, B\}$ (3.5)

3- Compliance with the correctness criterion stated in the theorem 41:
From (3.0), (3.1) and (3.2) we have:
$\mathcal{W}'_{p,F}(N_b^i, \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}) = \{A, B\} \sqsupseteq \ulcorner N_b^i \urcorner \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{Y.A\}_{k_b}) = \{A, B\}$ (3.6)
From (3.3), (3.4) and (3.5) we have:
$\mathcal{W}'_{p,F}(Y, \{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}) = \{A, B\} \sqsupseteq \ulcorner Y \urcorner \sqcap F(Y, \partial[\overline{Y}]\{Y.A\}_{k_b}) = \ulcorner Y \urcorner \sqcap \{A, B\}$ (3.7)
From (3.6) and (3.7), the messages exchanged in the session S'^{i} respect the correctness criterion stated in the theorem 41. (IV)
From (IV), the messages exchanged in the generalized role of B respect the correctness criterion stated by the theorem 41. (V)
From (III) and (V), the protocol NSL respects the correctness criterion stated by the theorem 41, then it is correct with respect to the secrecy property.

6 Comparison with related works

Houmani et al. in [8,9,10,11] have defined universal functions to analyze increasing protocols. 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 difficulty to prove that a function is full-invariant by substitution. Henceforth, with a witness-function, there is no need to request the property of full-invariance by substitution from an interpretation function. The witness-function takes it in input, without this property, and solves the problem of substitutions locally in the protocol, thanks to its construction that depends fully on the static part of a message and thanks to its two bounds that do not depend on substitutions (bring two properties and have one for free). This enables us to build more functions, and then, to analyze more protocols. The bounds of a witness-function having an output free of variables are expected to prove the correctness of protocols that could not be proven with universal functions because of variables in output, for instance, Kerberos protocol and SET protocol with Houmanis' functions. Our approach does not need a complicated formalism as do the rank-functions suggested by Schneider in [1] that require the CSP formalism [24]. It does not need, neither, a strongly message-typing as suggested by Abadi in [5]. Our approach intersects Schneider's work and Houmani's work in the way of seeing the protocol correctness through its growth and through the need to have reliable metrics to evaluate security. Besides, all of these methods handle an unbounded number of sessions and provide a semi-decidable procedure since they decide only when the protocol is increasing.

7 Conclusion and future work

In this paper, we introduced the witness-functions as a technique to analyze cryptographic protocols for secrecy. We gave the way to build them. In a future work, we intend to define witness-functions based on the selection of other keys (other than the most external key) like the most internal key or all encryption keys together. In this respect, we believe that a witness-function based on the selection of all neighbors and all encryption keys (i.e. $S(\alpha, \{\{E.\alpha\}_{k_{AB}}\}_{k_{CD}}) = \{E, k_{AB}^{-1}, k_{CD}^{-1}\}$) could efficiently deal with protocols with the Diffie-Hellman property in a non-empty equational theory. In this respect, we believe that the witness-functions are ready to deal with algebraic properties in convergent theories. In addition, we intend to take benefice of the bounds of a witness-function to consider exchanged messages in a protocol as keys of encryption. We intend also to experiment our witness-functions on compose protocols. Indeed, many protocols could be secure when they are analyzed separately but may show undesirable interactions when analyzed together. In this regard, similar researches had been led by V. Cortier, S. Ciobaca and S. Delaune in [25,26,27] where they suggest protocol-tags to secure compose protocols. We think that our witness-functions are prepared to deal with this question.

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

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

14. 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.

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

16. 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/WFFMS/WFAppend.pdf`. pages 1–28, 2014.

17. Bruno Blanchet. Automatic verification of correspondences for security protocols. *Journal of Computer Security*, 17(4):363–434, 2009.

18. Hubert Comon-Lundh, Véronique Cortier, and Eugen Zalinescu. Deciding security properties for cryptographic protocols. application to key cycles. *ACM Trans. Comput. Log.*, 11(2), 2010.

19. Véronique Cortier and Stéphanie Delaune. Decidability and combination results for two notions of knowledge in security protocols. *J. Autom. Reasoning*, 48(4):441–487, 2012.

20. Véronique Cortier, Steve Kremer, and Bogdan Warinschi. A survey of symbolic methods in computational analysis of cryptographic systems. *J. Autom. Reasoning*, 46(3-4):225–259, 2011.

21. Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998.

22. Nachum Dershowitz and David A. Plaisted. Rewriting. In *Handbook of Automated Reasoning*, pages 535–610. 2001.

23. Hubert Comon-Lundh, Claude Kirchner, and Hélène Kirchner, editors. *Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday*, volume 4600 of *Lecture Notes in Computer Science.* Springer, 2007.

24. Steve A. Schneider, Helen Treharne, and Neil Evans. Chunks: Component verification in csp||b. In *IFM*, pages 89–108, 2005.

25. Stefan Ciobaca and Veronique Cortier. Protocol composition for arbitrary primitives. *2012 IEEE 25th Computer Security Foundations Symposium*, 0:322–336, 2010.

26. Véronique Cortier. Secure composition of protocols. In *TOSCA*, pages 29–32, 2011.

27. Véronique Cortier and Stéphanie Delaune. Safely composing security protocols. *Formal Methods in System Design*, 34(1):1–36, 2009.