Introduction to the Witness Functions for Secrecy in Cryptographic Protocols

Jaouhar Fattahi, Mohamed Mejri, and Hanane Houmani

Abstract² In this paper, we examine the property of secrecy in cryptographic protocols from the angle of the growth of the protocol. Intuitively, an increasing protocol preserves the secret. For that, we need functions to estimate the security of messages. Here, we give relaxed conditions on the functions and on the protocol and we prove that an increasing protocol is correct when analyzed with functions that meet these conditions. Then, we shortly introduce the witness functions to analyze protocols for secrecy

Index Terms² Cryptographic protocol, role-based specification, secrecy

I. Introduction

In this paper, we look at the property of secrecy in cryptographic protocols from the point of view of their growth. Intuitively, an increasing protocol does not leak the secret. That is, if the security of every atomic message does not decrease between the reception and the sending steps in a protocol, the secret is kept. For this, we need "good" metrics that reasonably estimate the security of every atomic message. This point of view has been embraced in other previous works. For example, Schneider [1], [2] presented the rank-functions as metrics to calculate the security of messages. These functions succeeded to analyze several protocols such the Needham-Schroeder protocol [2] and the Woo-Lam protocol [3]. However, an analysis with the rank-functions needs to implement the protocol in the CSP algebra [4] - [6] which is not an easy task. In [7], Abadi asserts that: "If a protocol typechecks, then it does not leak its secret inputs". For that, he demanded from the messages generated by the protocol to strictly have the following shape: {secret, public, any, confounder} in order to easily determine the security level of every component. Unfortunately, this approach cannot be used to analyze everyday protocols that had not been implemented with respect to this condition. Besides, such analysis compels the verifier to implement the protocol in the Spi-Calculus [8], [9], that is an extension of the Pi-calculus [10] for cryptographic protocols, which requires special skills from the verifier. In [11] - [14], Houmani et al. proposed universal functions, named interpretation functions, as metrics to calculate the security of messages. These functions are based on selections under the protection of the direct key of encryption and operate in a role-based specification [15] - [18]. An interpretation function must satisfy to some conditions before being certified reliable for protocol analysis. Naturally, less we have conditions, more we are able to build many other functions and better we are equipped to prove protocols correct. Indeed, a function may not succeed to show the growth of a protocol, but another may do so. We draw conclusions from [11] - [14] that the conditions on the interpretation functions were so restrictive that many protocols could not be proven correct with the limited number of functions that Houmani et al. managed to build. In fact, just two functions could be built and proven reliable (DEKAN and DEK). We think that the condition of the full-invariance by substitution is the most restrictive one. So, we believe that if we liberate a function from this condition, we will be able to build more functions. This property is however very important since it enables any decision made on messages of the generalized roles of the protocol (messages with variables) to be exported to valid traces (closed messages). In this work, we introduce the witness functions to analyze cryptographic protocols. We show that they are reliable. They are protocol-independent and use derivation techniques to limit the variable effects at the time of analysis. They provide two bounds that are independent of all substitutions. This negates any need to the full-invariance by substitution property and enables to build more functions.

A. Notations

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

We denote by %L O ç á ág áå áæ ¤ áä À P the context containing the parameters that affect the analysis of a protocol:

M: is a set of messages built from the algebraic signature O 0 á6 P, where è is a set of atomic names (nonces, keys, principals, etc.) and 6 is a set of allowed functions symbols (ex: encryption, dec: decryption, pair:: concatenation denoted by "." here, etc.). We use + to denote the set of all possible substitution from ò to M. We denote by Û all atomic messages in M; by Û(þ) the set of atomic messages (or atoms) in þ and by J the set of agents (principals) including the intruder I. We denote by G Fs the reverse key of a key G and we consider that tG ?5 ;?5 = G

H is the equational theory that describes the algebraic properties of the functions in 6 by equations. e.g. A J :?@ A T á U ; á U ?5).

g: is the inference system of the intruder under the equational theory.

Let / be a set of messages and m a message. g ¼ I means that the intruder is able to infer m from / using her capacity. We extend this notation to traces as following:

Q: g % I means that the intruder can infer m from the

Manuscript received April 24, 2014; revised June 10, 2015.

Jaouhar Fattahi and Mohamed Mejri are with LSI Group, Laval University, Quebec, Canada (e-mail: jaouhar.fattahi.1@ulaval.ca, Mohamed.Mejri@ift.ulaval.ca).

Hanane Houmani is with University Hassan II, Morocco (e-mail: Hanane.Houmani@ift.ulaval.ca).

messages exchanged in the trace. We assume that the intruder has the full control of the net as described in the Dolev-Yao model [19]. She can intercept, delete, redirect and modify any message. She knows the public keys of all agents, her private keys and the keys she shares with other agents. She can encrypt or decrypt any message with known keys.

Formally, the intruder has generically the following rules of building messages:

:E J; ã P s / g% I [I Õ/ ë - : +;]

:K L; ã / g% I s å / g% I J / g% B: I s å á I J ; [B Õ Ç]

:A M; ã / g% I "á I "L% I / g% I [(I "L¼ I I "L :¼ ; I)]

Example 1.1. The intruder capacity may be described by the following rules:

:E J; ã P s / g% I [I Õ/ ë - : +;]

:@ A; ã / g% G^Fs á / g% <I =_G / g% I

:A J; ã / g% G á / g% I / g% <I =_G

:? K J ?; ã P / g% I á / g% I " / g% I äI "

:@ A ? K J; ã P = P / g% I äI " / g% I

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

å: is a function from J to M, that assigns to any agent (principal) a set of atomic messages describing her initial knowledge. We denote by - %: +; the initial knowledge of the intruder, or simply - : +; where the context is clear.

æ: is the security lattice : æá Ó á Q á P á c á a ; used to attribute security levels to messages. A concrete example of a lattice is : t ªá C á æ á æ á ã á å ; that will be used to attribute to a message α the set of principals that are allowed to know it.

ä À is a partial function that assigns a value of security (type) to a message in . Let / be a set of messages and be a message. We write ¿ ÀO ¿ ÀÍ Ì I ñ Õ/ ä ¿ "ÀO ¿ À

+ Our analysis takes place in a role-based specification. A role-based specification is a set of generalized roles. A generalized role is a protocol abstraction where the emphasis is made on a specific principal and where all the messages that the principal does not know are replaced by variables. An exponent i (the session identifier) is added to each fresh message to emphasize that these components change their values from one run to another. More details about the role-based specification are in [1], [15]-[18].

+ 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 ä L ; å the set of valid traces of p.

+ We denote by ç $_2^{á}$ the set of messages with variables generated by 4 À (p), by ç $_2^{À}$ the set of closed messages generated by substituting terms in ç $_2^{á}$.

We denote by 4^E (respectively 4^F) the set of sent messages (respectively received messages) by a honest agent in the role 4. Commonly, we reserve the uppercase letters for sets or sequences of elements and the lowercase for single elements. For instance / denotes a set of messages, m a single message, 4 a role composed of a sequence of steps, step and 4.r the role ending by the step r.

II. CORRECTNESS OF INCREASING PROTOCOLS

In this section, we prove that an increasing protocol is correct with respect to the secrecy property when analyzed with functions that satisfy to few conditions.

A. C-reliable Interpretation Functions

Definition 2.1. (Well-formed interpretation function)

Let F be an interpretation function and C be a context of verification. F is well-formed in C if ˆ ‘ ” ƒ Ž / Ž á / 5 á / 6 C ç we have:

:DáDÐ=; L A

Ê=Ð :ç ;äP :Dá/ 5 ë / 6; L :Ùá/ 5 ; P (:Ùá/ 6;

:Dá/ ; L aá ˆD Ñ Û :/ ;

A well-formed interpretation function attributes for an atomic message α, that appears in clear in a set of messages /, the bottom value "Ô" to express the fact that everybody knows it. It attributes for it in the union of sets, the minimum "P" of the returned values calculated in each set separately. It attributes for it the top value "T", if it does not show in this set.

Definition 2.2. (Full-invariant-by-intruder Interpretation Function)

Let F be an interpretation function and C be a context of verification. F is full-invariant-by-intruder in C if for all ˆ ‘ ” ƒ Ž Ž ç á þ Ð ç we have / g¼ þ :

Ê=Ð Û :þ ;äk :=áþ ; O (:Ùá/ ;oé :¿ :$;À O ¿ÙÀ;

/ g¼ þ : Ê. Ð Û :þ ;äkî :Ùáþ ; O î :Ùá/ ;oé :¿ Àÿ; O ¿ÙÀ; A full-invariant-by-intruder function F is such that when it attributes a security level to an atomic message _ in a set of messages M, the intruder can never deduce from M another message m that decreases this level (i.e. :=áþ ; O (:Ùá/ ;) unless D is initially intended to the intruder (i.e. :¿ :$;À O ¿ÙÀ;).

Definition 2.3. (Reliable Interpretation Function)

Let F be an interpretation function and C a context of verification. F is C-reliable if F is well-formed and F is full-invariant-by-intruder in C.

A reliable interpretation function is simply a function that is well-formed and full-invariant-by-intruder in a given context of verification C.

Definition 2.4. (F-increasing Protocol)

Let F be an interpretation function, C be a context of verification and p a protocol. p is F-increasing in C if ˆ ‘ ” ƒ Ž Ž ä Ð :¹; ÊD Ð ã Õ 7 ç ', we have ÊÙ Ð Û :ç ;,

(: =áN? P; O ¿ÙÀP : =á4? P;

An F-increasing protocol is a protocol such that every involved principal sends continuously valid traces (interleaving of substituted generalized roles) in such way that every atom has a level of security, estimated with an interpretation function F, higher or equal to its level of security in the context on reception (¿ÙÀP : =á4? P;).

Definition 2.5. (Secret Disclosure)

Let p be a protocol and C a context of verification.

We say that p discloses a secret Ð Û : ç ; in C if:

Ì Ð äLåä g¼ =; • : ¿ : +; À¢ ¿ÙÀ;

We say that a protocol discloses a secret if the intruder can exploit a valid trace generated by the protocol using her knowledge - : +; in a context of verification C, to deduce a secret _ that she is not initially intended to her (expressed by: ¿ : ; À¢ ¿=À).

Lemma 2.6. Let F be a C-reliable interpretation function and p a F-increasing protocol. We have:

Êþ Ð ç äåLå g¼ þ œ
Ê= Ð Û : þ ; ä (: Ùáþ ; O ¿ÙÀ; é : ¿ : +; ÀO ¿ÙÀ;

The lemma 2.6 states that for an atom _ in a message m generated by an increasing protocol, its level of security estimated by a reliable interpretation function remains greater or equal to its initial value in the context, if the intruder is not initially allowed to know it. Indeed, initially the atom has some security level. This level cannot be maliciously decreased by the intruder using her initial knowledge and received messages since a reliable interpretation function is full-invariant-by-intruder and then cannot be misled by her. In every new step, 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 properties of reliability of the interpretation function in all the steps.

Theorem 2.7. (Correctness of Increasing Protocols)

Let F be a C-reliable interpretation function and p a F-increasing protocol. p is C-correct with respect to the secrecy property.

See the proof in [20].

The theorem 2.7 states that an increasing protocol is correct with respect to the secrecy property when analyzed with a reliable interpretation function. Compared to the sufficient conditions proposed by Houmani et al. in [11], [14], we have one condition less. In fact, Houmani et al. demanded from a protocol to be increasing on the messages of the generalized roles of the protocol (that contain variables), and from the interpretation function to resist to the problem of substitution of variables, hence to be full-invariant by substitution. Even if they gave a comprehensive guideline to safely build these functions, just two functions have been given: DEK and DEKAN. This is due to the difficulty to find, and then to prove, that a function meets the full-invariance by substitution property. In this paper, we free our functions from this restrictive condition in the hope to be able to build more functions. We put this condition in our definition of an increasing protocol, that is demanded now to be increasing on valid traces (closed messages).

III. Introduction to the Witness-Functions

In [21], [22], we give a constructive way of reliable interpretation functions that operate on valid traces (closed messages) and based on selections of atomic messages. We define first a generic class of selections of atoms inside the protection of the external key such that when they are composed to an appropriate morphism give reliable interpretation functions. We prove that for any atom a in a message l, any selection of atoms that takes place inside the encryption by the most external protective key K such that: fG$^{Fs}_{=>}$‡O ¿=À; is a reliable selection. Thus, an intruder cannot modify this selection when she does not have the key K (i.e. ¿ : +; ÀO fG$^{Fs}_{=>}$‡). This selection can only be modified by people who are initially authorized to know K (¿ : +; ÀO ¿=À) by transitivity. So, such class of selections is full-invariant by intruder. In addition, we build this class so that it is well-formed by construction.

Example 3.1. Let a be an atomic message and l be a message such that: ¿=ÀL < á =and l L < ä ä=ä =$_{Pif}$. Let 5$_s$ á5$_t$ and 5$_u$ be three selections such that:

5$_s$: =ál ; L <G$^{Fs}_{=>}$= , 5$_t$: =ál ; L <#á%á&G$^{Fs}_{=>}$= and 5$_u$: =ál ; L <#á%á&á5$^{Fs}_{=>}$= These three selections are reliable.

Then, we define specific functions that are a composition of an appropriate morphism and instances of this class of selections. This morphism exports the properties of reliability from a selection to a function and transforms selected atoms to security levels. A such morphism could be defined as follows:

Öã : : t^{Û} ; C ; 7 æ^{O}
a ‹ ˆ / L Î ä
/ 7 \ P ÖE Ö: =; ‹ • ‘ –ä

such that:

Ö: =; L \ <== ‹ ˆ= Õã : ’ ” ‹ • … ‹ ‹ ƒ ‡ • –; ä– ›
¿=À ‹ • ‘ –ä

This morphism returns for a principal in a selection its identity. It returns for a key, its level of security in the context of verification.

Example 3.2. Let a be an atom, l a message and G$_{=>}$ a key such that:

¿=ÀL < á ÷; l L < ä ä=ä =$_{Pif}$; fG$^{Fs}_{=>}$‡L < á á ÷;
5$_5$: =ál ; L <G$^{25}_{OO}$=
5$_6$: =ál ; L <#á%á&=
5$_7$: =ál ; L <#á%á&á5$^{25}_{OO}$=
(5 : =ál ; L Ö Ü5$_5$: =ál ; L ¿G$^{25}_{OO}$ÀL < á á ÷;
(6 : =ál ; L Ö Ü5$_6$: =ál ; L <#á%á&=
(7 : =ál ; L Ö Ü5$_7$: =ál ; L <#á%á&=ë ¿G$^{25}_{OO}$ÀL <#á%á&á5=ä
(u : =ál ; L Ö Ü5$_u$: =ál ; L <#á%á&=ë fG$^{Fs}_{=>}$‡L <#á%á&á5=

Unfortunately, these interpretation functions operate only on valid traces (closed messages). However, a static protocol analysis should be run over the finite set of messages of the generalized roles because the set of valid traces is infinite. The finite set of the generalized roles contains variables. The interpretation 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) [23]-[25]. The full-invariance by substitution is the property that allows us to perform an analysis over messages with variables and propagate the conclusion made on to closed messages. To solve this problem, we introduce the notion of derivative messages to reduce variable effects and we define the derivative message as follows:

Definition 3.3. (Derivative Message)

$$\begin{aligned}
&w(X) = X\\
&w(k) = k\\
&w(x) = x\\
&w(u,v) = w(u), w(v)\\
&\delta_{\{x\}}(u) = \delta_{\{x\}}(v) \text{ if } w(u) = w(v)\\
&\delta_{\{x\}}(u) = \delta_{\{x\}}(v) \text{ if } \delta_{\{x\}}(u) = \delta_{\{x\}}(v)\\
&w(\{u\}_v) = \{w(u)\}_{w(v)} \text{ if } \delta_{\{x\}}(v) = \delta_{\{x\}}(k)\\
&\delta_{\{x\}}(\{u\}_v) = \{\delta_{\{x\}}(u)\}_{\delta_{\{x\}}(v)}\\
&\delta_{\{x\}}(\{u\}_k) = \{\delta_{\{x\}}(u)\}_k \text{ if } \delta_{\{x\}}(k) = k
\end{aligned}$$

$\delta_{\{x\}}(u)$ consists in eliminating the variable: in u and $\delta_{\overline{\{x\}}}(u)$ consists in eliminating all variables, except x in u. Therefore, x when overlined is considered as a constant in it. The derivative message $\delta(u)$ consists in eliminating all the variables in u. We may think now to calculate the level of security of an atom X in any closed message m ($m \in \mathcal{C}_0$) in the derivative message $\delta(m)$ rather than in m as shown in the definition 3.4.

Definition 3.4. Let $m \in \mathcal{C}_0$, $X \in \mathcal{V}$ and $\mathcal{P}$ be a valid trace.

For all $\mathcal{P}(P) = (I, \ldots)$, we denote by:

$$\begin{aligned}
&\Phi(X, m, \mathcal{P}) =\\
&\quad \Phi(X, \delta(m), I) \text{ if } X \notin \delta(m)\\
&\quad \min_{m' \in \mathcal{P}(I)} \{\Phi(X, \delta(m'), I) \mid w(m') = w(m)\} \text{ if } X \in \delta(m)
\end{aligned}$$

In fact, for an atom X in the static part of m (i.e. in $\delta(m)$), the application $\Phi(X, \delta(m), I)$ ignores the variables in m and gives it the value $\Phi(X, \delta(m), \emptyset)$. For anything that is not an atom of the static part, that comes so by substitution of some variable X in m, $\Phi(X, \delta(m), I)$ considers it as the variable itself, treated as a constant and as a block, and gives it all the time the same value $\Phi(\_, \delta(m), I)$. It gives the top value for an atom that does not appear in m. The major advantage of the application in the definition 3.4 is that it does not depend on substitutions thanks to the operator of derivation that uses. When an interpretation function F is reliable, the application in the definition 3.4 remains full-invariant by intruder (because the derivation just removes atoms, so the atoms returned by this function remain always beyond the knowledge of the intruder). It remains well-formed also. However, it can lose its property as a function since it can return more than one image for the same preimage because the operator of derivation may cause a "loss of details" as shown in the example 3.5.

Example 3.5. Let I_s and I_t be two messages of a generalized role of a protocol p such that $I_s = \{x, a\}_k$ and $I_t = \{x, a, b\}_k$ and $\delta(I) = \{\_, a\}_k$. Let $I = \{x, a, b\}_{pub(i)}$ be a closed message in a valid trace generated by p. Let F be the function based on the selection of the most external key of encryption and all the neighbors inside. We have:

$$\begin{aligned}
&\text{knows}(\delta(I), \mathcal{P}) = \begin{cases} \{x, a\}_k & \text{if } I = I_s\theta_s : \theta_s = \{\}\\ \{x, a, b\}_k & \text{if } I = I_t\theta_t : \theta_t = \{\} \end{cases}
\end{aligned}$$

Hence $\text{knows}(\delta(I), \mathcal{P})$ is not even a function on the closed message I since it can return more than one image for I. This leads us directly to the witness functions. A witness function looks for all the sources of any closed message in input and returns the minimum calculated by their derivative messages. This minimum exists and is unique in the finite set $\mathcal{C}_0^{\alpha}$.

A witness function is so a function. The general form of a witness function is:

$$\begin{aligned}
&W_{F,\alpha}(m, P) = \min(P' : \text{knows}(\delta(P'), \mathcal{P}) \sqsubseteq \text{knows}(\delta(m), \mathcal{P}),\\
&\qquad P' \in \mathcal{C}_0^{\alpha}\\
&\text{with } \delta(P') \sqsubseteq \delta(m) \text{ and } P' \sqsubseteq m)
\end{aligned}$$

It is easy to prove that $W_{F,\alpha}$ remains reliable for any function F based on a selection inside the external protective key. 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. A witness function offers two elegant bounds that are independent of all substitutions as follows:

$$\begin{aligned}
&\text{knows}(\delta(P), \mathcal{P}) \sqsubseteq \text{knows}(\delta(m), \mathcal{P}) \sqsubseteq \text{knows}(\delta(P'), \mathcal{P}),\\
&\qquad P' \in \mathcal{C}_0^{\alpha}\\
&\text{with } \delta(P') \sqsubseteq \delta(m) \text{ and } P' \sqsubseteq m
\end{aligned}$$

The upper bound of a witness function estimates the security level of a given atomic message x in a given closed message m in P from one confirmed source m in $\mathcal{C}_0^{\alpha}$ (m is naturally a source of m in P), the witness function itself estimates it from the exact sources of m in P in $\mathcal{C}_0^{\alpha}$ (i.e. when the protocol is executed), and the lower bound estimates it from all possible sources of m in P (i.e. the messages that can be unified with m in $\mathcal{C}_0^{\alpha}$). The unification in the lower bound looks for all the candidates in all the possible sources of the closed message in the protocol. These bounds allows us to state the protocol analysis with a witness function theorem that sets a criterion for protocols correctness with respect to the secrecy property (see the Theorem 3.6).

Theorem 3.6. (Protocol Analysis with a Witness Function)

Let $W_{F,\alpha}$ be a witness function and F an interpretation function based on a selection inside the external key. A sufficient condition of correctness of with respect to the

secrecy property is:

P (k=áò⋝ ? þ ñ ê ñ o O ¿=À P (k=áò⋝ ? 4 ? o
þ ¨ Ð ç á 2
Ì ê ñ Ð áþ ñ ê ñ L Ǹ ê ñ

See the proof 20 in [22]

The result in the theorem 3.6 comes directly from the theorem 2.7 and the bounds of the witness-function. Thanks to the independence of the criterion given in the theorem 3.6 of all substitutions, any decision made on the generalized roles can be exported to valid traces. This replaces the restrictive condition of full invariance by substitution stated in [11], [14].

Example 3.7. Let p be a protocol analyzed in a role-based specification. We extract first the roles of all the agents that participate in it. Then, we extract the generalized roles 4À(p) where any message that an agent does not know upon it she could perform no verification is replaced by a variable. Let ç á 2 be the set of messages generated by 4À(p) where variables, nonces and principal identities are renamed to express some typing rules in the messages of the protocol. An analysis of p with a witness-function Í L á consists in verifying in every generalized role 4. Ǹ that the message Ǹ when sending, and the message 4 when receiving, respect the criterion set by the theorem 3.6. Let

ç á 2 L ≪; s ä#s =-$s á≪$t ä; t =-#u á< s =-$u =

where the variables are denoted by ;s and ;t and the static names by #s, -$s, $t, -#u and -$u.

Let 4. Ǹ be a generalized role in 4À(P) and 4 and Ǹ be the two messages, respectively, in the receiving step and the sending step of it such that 4 L{ <; ä#=G } and Ǹ L{ ≪$ ä; =G }.

Let F be the function based on the selection of the most external protective key of encryption and all the neighbors (principals) inside. Let us have a context such that:

¿#ÀL c, ¿$ÀL c, ƒG s ‡L≪#=, ƒG s ‡L≪$= ã L
< :‹•–"—; ǻ ǂ"á á#5 á$5 á#6 á$6 á#7 á$7 å =

The principal identities are not analyzed since they are set public.

We denote by Í L á . m) the lower bound:

P (: =áò⋝ ? þ ñ ê ñ;
þ ¨ Ð ç á 2
Ì ê ñ Ð áþ ñ ê ñ L I ê ñ

Of the witness-function Í L á . m 1

a) When sending: Ǹ L{ ≪$ ä; =G } (in a sending step, we use the lower bound)

ÊY.{ I ïÕç á É Ì ê ñ Ð áþ ñ ê ñ L ≪$ ä; =Þ ê ñ}=
{ ≪$t ä; t =-#u ês, < s =-$u êt} with:

ês L ≪$t = $á; t = ; á-#u = G=
P
êt L [: s = $ä; á-$u = G_

Í L á(Y ≪$ ä; =G) ={Definition of the lower bound of the witness-function}

(:; áò⋝g ≪$6 ä 6 =À₂ / ê5 ñ; P (:; áò⋝g < 5 =À₃ / ê6 ñ;={Setting the static neighborhood by renaming the static names}

(: áò⋝g ≪$ ä 6 =Þ ê5 ñ; P (: áò⋝g < 5 =Þ ê6 ñ; ={Definition 3.4}

(:;6 áò⋝ g ≪$ ä 6 =Þ ; P (::5 áò⋝ g < 5 =Þ ;={Derivation}

(:;6 á≪$ ä 6 =Þ ; P (::5 á< 5 =Þ ; = {F is based on the selection the external key of encryption and all the neighbors}

≪#á$=ë ≪#=L ≪#á$=: E;

b) When receiving: 4 L{ <; ä#=G } (in a receiving step, we use the upper bound)

ÊY. (:; áò⋝g <; ä#=Þ ; = (:; á<; ä#=Þ ; ={A, B} : E;E

From : E; and : E;E we have:

Í L á(Y, ≪$ ä; =G)= {A, B} O
¿ ÀP (:; áò⋝ ? <; ä#=G ; = ¿ Àë ≪#á$=: E E; E

From : E E;E Ǹ respects the correctness criterion set by the theorem 3.6.

IV. Conclusion and Future Work

In this work, we gave relaxed conditions on functions and on the protocol and we proved that an increasing protocol is correct with respect to the secrecy property when analyzed with these functions. Then we briefly introduced the witness functions. A witness-function is protocol-dependent that meets these conditions and uses derivation techniques to solve the question of substitution locally in the protocol. Its two bounds, that are independent of all substitutions, enable any decision made on the generalized roles (messages with variables) to be exported to valid traces (closed messages). The witness-functions were successful to prove the correctness of many protocols such the NSL protocol [26] and they even helped to locate flaws as in the Needham-Schroeder protocol [27]. In a future work, we will give the full details of the witness-functions and we will run analyzes on real protocols.

References

[1] S. A. Schneider and R. Delicata, "Verifying security protocols: An application of csp," Years Communicating Sequential Processes, pp. 243-263, 2004.

[2] S. Schneider and R. Holloway, Using CSP for Protocol Analysis: The Needham-Schroeder Public-Key Protocol, Technical report, 1996.

[3] S. A. Shaikh and V. J. Bush, "Analysing the woo-lam protocol using csp and rank functions," WOSIS, pp. 3-12, 2005.

[4] S. Schneider, "Security properties and CSP," in Proc. the IEEE Symposium on Security and Privacy, 1996, pp. 174-187.

[5] S. Schneider, "Verifying authentication protocols in csp," IEEE Trans. Software Eng. vol. 24, no. 9, pp. 741-758, 1998.

[6] J. Heather and S. Schneider, "A decision procedure for the existence of a rank function," J. Comput. Secur., vol. 13, no. 2, pp. 317-344, March 2005.
[7] M. Abadi, "Secrecy by typing in security protocols," J. ACM, vol. 46, pp. 611-638, 1998.
[8] M. Abadi and A. D. Gordon, "Reasoning about cryptographic protocols in the spi calculus," in Concur, pp. 59-73, 1997.
[9] M. Abadi and A. D. Gordon, "A calculus for cryptographic protocols: The SPI calculus," in Proc. the ACM Conference on Computer and Communications Security, 1997, pp. 36-47.
[10] R. Milner, "The PI calculus and its applications (keynote address)," in Proc. the International Joint Conference on Natural Language Processing, 1998, pp. 3-4.
[11] H. Houmani and M. Mejri, "Practical and universal interpretation functions for secrecy," in Proc. the International Conference on Security and Cryptography, 2007, pp. 157-164.
[12] H. Houmani and M. Mejri, "Ensuring the correctness of cryptographic protocols with respect to secrecy," in Proc. the International Conference on Security and Cryptography, 2008, pp. 184-189.
[13] H. Houmani and M. Mejri, "Formal analysis of set and NSL protocols using the interpretation functions-based method," J. Journal Comp. Netw. and Communic., 2012
[14] H. Houmani, M. Mejri, and H. Fujita, "Secrecy of cryptographic protocols under equational theory," Knowl.-Based Syst., vol. 22, no. 3, pp. 160-173, 2009.
[15] J. Fattahi, M. Mejri, and H. Houmani. Context of Verification and Role-Based Specification [Online]. Available: http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/WFCRCH
[16] M. Debbabi, Y. Legaré, and M. Mejri, "An environment for the specification and analysis of cryptoprotocols," in Proc. the Annual Computer Security Applications Conference, Annual Computer Security Applications Conference, 1998, pp. 321-332.
[17] M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi, "Formal automatic verification of authentication crytographic protocols," in Proc. the International Conference on Formal Engineering Methods, 1997, pp. 50-59.
[18] M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi, "From protocol specifications to flaws and attack scenarios: An automatic and formal algorithm," in Proc. the International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1997, pp. 256-262.
[19] D. Dolev and A. C. C. Yao, "On the security of public key protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198-207, 1983.
[20] J. Fattahi, M. Mejri, and H. Houmani. Sufficient conditions for secrecy in cryptographic protocols: Proofs and intermediate results [Online]. Available: http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/WFCRCH/preuvesCH.pd
[21] J. Fattahi, M. Mejri, and H. Houmani. New functions for secrecy in cryptographic protocols [Online]. Available: http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/WFCRCH/
[22] J. Fattahi, M. Mejri, and H. Houmani. The witness functions: Proofs and intermediate results [Online]. Available: http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/WFCRCH/WitFunProofs.pdf.
[23] F. Baader and T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.
[24] N. Dershowitz and D. A. Plaisted, Rewriting in Handbook of Automated Reasoning, pp. 535-610, 2001.
[25] H. C. Lundh, C. Kirchner, and H. Kirchner, Lecture Notes in Computer Science, Springer, 2007.
[26] J. Fattahi, M. Mejri, and H. Houmani. NSL protocol analysis with a witness function. [Online]. Available: http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/WFCRCH/NSL
[27] J. Fattahi, M. Mejri, and H. Houmani. A variation of needham-schroeder protocol analysis with a witness function. [Online]. Available: http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/WFCRCH/

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

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

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