Formal Analysis of the TLS Handshake Protocol

Hanane Houmani and Mourad Debbabi
CIISE, Concordia University, Montreal, Quebec, Canada

Keywords: TLS/SSL Protocol, Formal Analysis, Confidentiality, Secrecy.

Abstract: Most applications in the Internet as e-banking, e-commerce, e-maling, etc., use the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol to protect the communication channel between the client and the server. That is why it is paramount to ensure the security objectives such as confidentiality, authentication and integrity of the SSL/TLS protocol. In this paper we prove the confidentiality (secrecy) property of the SSL/TLS handshake protocol which consititues the main core of the SSL/TLS protocol. To perform this analysis, we introduce a new funcion called DINEK function that safeltly estimates the security level of messages. More precisely, this function which shares a conceptual origin with the idea of a rank function, allows to estimate a security level of a message (including the unknown messages) according to the interaction between the protocol and the intruder. This function could not be used only to verify the TLS protocol as we will show in this paper, but also to verify the secrecy property for large class of protocols and in particular Key Agreement protocols. The verification using the DINEK function is proven in this paper for unbounded number of sessions and unbounded number of nouces.

1 MOTIVATIONS AND BACKGROUND

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that aim to provide secure communication over the Internet (Hickman, 1994; Dierks and Rescorla, 2008). SSL/TLS and their versions are in widespread use in applications such as web browsing, electronic mail, e-commerce, banking, cloud computing, VPN, Internet faxing, instant messaging and voice-over-IP (VoIP). In fact, several version of SSL/TLS are used in each time a secure communication is needed. More precisely, TLS and SSL encrypt the segments of network connections above the transport layer, using asymmetric cryptography to ensure security objectives such as confidentiality, integrity and authentication.

However, these security objectives are broken and many attacks and vulnerabilities (Mitchell et al., 1998; Oppliger and Gajek, 2005; Oppliger et al., 2006; Wagner and Schneier, 1996) have been discovered against the implementation and the cryptographic primitives used by this protocol rather than the protocol itself. For instance, in the implementation of SSL 2.0 some field are not well instanced what could be exploited for man-in-the-middle attack as described in (Oppliger et al., 2006). Also, a weak MAC construction is used as cryptographic primitive in SSL 2.0 as shown in (Wagner and Schneier, 1996). In the last years, many versions of SSL/TLS were been proposed to correct these flaws and vulnerabilities.

Therefore, ensuring the correctness with respect to the security objectives of TLS protocol is paramount. Indeed, most of the communication over the network are based on this protocol and a simple flaw could be dearly-won and costly. Formal methods to verify the security of cryptographic protocols have received much attention in recent years since they allow to give in concrete and formal way the proof of their correctness and security. Some of these works including comparative studies could be found in (Meadows, 2003; Sabelfeld and Myers, 2003; Carlsen, 1994; Clark and Jacob, 1996; Kemmerer et al., 1994; Liebl, 1993; Meadows, 1994; Rubin and Honeyman, 1993; Syverson, 1991; Syverson, 92). However, almost of these methods are not suitable to prove the security of the SSL/TLS protocol due to their restrictions.

Nevertheless, they are some attempt to prove the security of TLS protocol. For example, authors tried to prove in (Paulson, 1997a) some security properties (authentication and secrecy properties) during the handshake phase by using the inductive approach and

Houmani H. and Debbabi M..
Formal Analysis of the TLS Handshake Protocol.
DOI: 10.5220/0004075101920205
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2012), pages 192-205
ISBN: 978-989-8565-24-2

the theorem prover ”Isabelle”. However, the proof is not fully automatic and human interaction is needed to perform the proof which could be error prone. Moreover, the proof concerns only a simplified and abstracted version of SSL/TLS rather than the real version and the proof of the fact that the security of the simplified version of TLS is sufficient to ensure its security is not given. Also, SSL Handshake was been analyzed using a general purpose finite-state enumeration tool called Murϕ (He et al., 2005; Mitchell, 1998). As any model checker, this tool is enable to ensure the security of protocols in the absence of flaws.

In independent line of research, several works (Jager et al., 2011; Morrissey et al., 2008) analyzed the security property (authentication, confidentiality and integrity) of SSL/TLS handshake protocol. However, these works make some unrealistic assumptions and abstraction on the protocol. For instance, in (Morrissey et al., 2008) authors extensively use the random oracle model (Bellare and Rogaway, 1993) to separate the three layers they define in the TLS handshake, and to switch from computational to indistinguishability based security model. While in (Jager et al., 2011), authors use the standard model (some realistic assumptions on the encryption scheme) but they prove the security of only a truncated version of the SSL/TLS handshake protocol rather than the complete and original version.

In this paper, we prove the secrecy (confidentiality) property of the TLS handshake protocol on its original description the protocol. This analysis is conducted by using the interpretation functions-based method (Houmani and Mejri, 2008a; Houmani and Mejri, 2008b) which shares a conceptual origin with the idea of a rank function (Delicata and Schneider, 2005; Schneider, 1997). In fact, the main idea of the rank function-based method is to construct a message space in a way that the authentication will correspond to certain messages kept away from the intruder. The goal is to define a rank function which correctly assigns a positive rank to every message that the intruder may obtain and a negative rank for the others. As for the typing-based method, the idea consists of not decreasing the security levels of sent messages. However, the effort made to define a rank function that allows to guarantee the security of a cryptographic protocols is heavy and non-evident. In that way come the interpretation function-based method to allow defining in a semi-automatic way an interpretation function. An interpretation function could be viewed as a rank function that instead of estimating the security level of message in an absolute way, it allows to estimate in a relative and approximative way. For instance, in the rank function-based method, the rank of a message α is equal to 0 when the message is equal to s_a, and equal to 1 in other cases. In the inetrpretation function, the rank of a message is calculated always by considering a set of messages. For instance, the rank of α in $\{\alpha\}_k$ is equal to the rank of k that may be secret or not, and the rank of α in $\alpha.m$ is equal to 1 (public). This modification on the rank function allows to define rank function for a class of protocols instead of defining rank function for each protocol. Also, it allowed to have a guideline to define such functions.

In addition of that, the intrepretation function-based method generalizes the main result of the rank function-based method by proving the result for any class of protocol and any intruder capacities (including algebraic properties of cryptograhic primitives). Also, the verification is bounded and proven sufficient to guaranty the secrecy property for unbounded sessions and nouces in the presence of an active intruder who can apply an unbounded number of operations to the messages.

However, the guideline of interpretation function is not suitable to define interpretation function that allows to verify the secrecy property of key agreement protocols. This due to the fact, that in this guidline we propose to give to unkown messages unknown security levels. Hence, a key that is freshly shared between two agents and which is consiered for on of them or both as unknown message and could not ensure its confidetiality. In the reminder of ths paper, we will adress this problem by giving new class of interpretation function that could be used to analyze the secrecy property for key agreement protocol. Also, we prove that these kind of functions are sufficient to prove the secrecy for unbounded number of sessions and nouces. Also, we give in this paper, a concret examples (DEK and DINEK funcions) of such functions. With the DINEK function we prove the secrecy property of the TLS handshake protocol.

2 SSL/TLS HANDSHAKE PROTOCOL

The SSL/TLS protocol (Dierks and Rescorla, 2008) is composed of five protocols: Record Layer protocol, Handshake protocol, ChangeCipherSpec protocol, Application Data and Alert protocol. In this paper, we analyze the Handshake protocol that allows to authenticate the client and the server to each other and negotiate a statefull connection by using a handshaking procedure. During this phase, the client and server agree on various parameters used to establish the connection's security. For instance, they must agree on session keys that will be used for securing future connections. The standard description of the SSL/TLS protocol is as follows:

Table 1: The SSL/TLS handshake protocol.

$$\begin{array}{l}
1.\ C \rightarrow S\ :\ m_1 = C, N_c, Ver_c, IdSession \\
2.\ S \rightarrow C\ :\ m_2 = S, N_s, Ver_s, IdSession, CA(S, K_s) \\
3.\ C \rightarrow S\ :\ m_3 = IdSession, \{Ver_c, Secret_c, C, S\}_{K_S}, \\
\qquad\qquad\qquad CA(C, K_c), \{H(g_1(m_1, m_2, Secret_c, C, S))\}_{K_c^{-1}} \\
4.\ S \rightarrow C\ :\ m_4 = \{H(g_2(m_1, m_2, m_3, Secret_c, C, S))\}_{K_{cs}} \\
5.\ C \rightarrow S\ :\ m_5 = \{H(g_3(m_1, m_2, m_3, m_4, Secret_c, C, S))\}_{K_{cs}}
\end{array}$$

Where $K_{cs} = Master(Secrect_c, N_c N_s)$ and $Master()$ is a function that takes the secret $Secret_c$ and the nounces N_a and N_s and returns a key. F_1, F_2 and F_3 are some parameters and preferences chosen by the client C and the server S for the compression.

In fact, the client C and the server S exchange the messages m_1 and m_2 to synchronize with each other. In step 2, S provides a public key certificate to C in a certificate message. In step 3, C provides a public key certificate in a certificate message, a pseudorandomly generated master secret "*$secret_c$*" for the SSL/TLS session encrypted with the servers public key (found in the certificate message). Finally, C and S exchange all messages that are subsequently transmitted between C and S cryptographically protected in terms of authenticity, integrity, and confidentiality with cryptographic keys derived from the master secret "*$secret_c$*".

3 OVERVIEW OF THE INTERPRETATION FUNCTIONS-BASED METHOD

The main idea of the interpretation function-based method is based on some conditions that are proven sufficient to guarantee the secrecy property of any protocol that respects them. The proposed conditions can be easily verified in PTIME and they intuitively state that principals involved in the protocol should not decrease the security levels of sent components. The security level of an atomic message is either given within a context of verification (input information) or/and estimated from received messages. The protocols that satisfy this condition are called in this work "increasing protocols".

To verify whether a protocol is increasing, we should have a safe means, called "safe interpretation functions", to appropriately estimate the security levels of exchanged messages. By a safe means, we mean basically that the interpretation function could not be misled by intruder manipulations. Indeed, the intruder can make some changes on the received messages to affect the security of the components. Therefore, a safe interpretation function is a function that always gives the correct security level of a message even when the message is altered by an intruder. For instance, a safe interpretation function could be a function that attributes the security level of a message according to its direct encrypted key, this function was called the DEK (Direct Encrypted Key) function (Houmani and Mejri, 2008c) and denoted by F_{DEK}. In this case, $F_{DEK}(N_b, \{A, N_B\}_{k_{ab}})$ calculates the security level of N_b in the message $\{A, N_B\}_{k_{ab}}$, and it is equal to the security level of k_{ab}. For example, if the security level of k_{ab} is *secret* the we have:

$$F_{DEK}(N_b, \{A, N_B\}_{k_{ab}}) \quad = \quad secret$$

The main result of the interpretation functions-based method are general and do not depend on a specific intruder capacities or a pecific class of protocols. Indeed, the authors introduced the concept of a "context of verification" and proved all results for any context of verification. A context of verification contains basically the class of protocols, the class of intruder capacities, and the class of algebraic properties of the cryptographic primitives. This concept is a great flexibility that is useful to change the class of protocols or the intruder capacities and still be able to use the approach without any need of reworking the proofs and/or the conditions. For instance, we can apply the approach to the protocols that use either symmetric or asymmetric keys. Also, we can apply the approach with or without algebraic properties of cryptographic primitives.

The secrecy property of increasing protocols is guaranteed even for an unbounded number of sessions and in the presence of an active intruder who can apply an unbounded number of operations to the messages that he manipulates. Indeed, verifying whether the specification of the protocol is increasing, is proven sufficient to guarantee the secrecy property. In other words, the interpretation functions-based method makes some static conditions on the protocol that are sufficient to the secrecy property.

To sum up, the verification of the secrecy property consists of verifying whether the protocol is increasing according to a safe interpretation function and a context of verification. In fact, if the protocol is increasing according to a specific safe interpretation function, then we can deduce that the protocol respects the secrecy property, otherwise we cannot make any statement. In this case, the analyzed protocol may be increasing by using another safe interpretation function. Nevertheless, even if the verification is not conclusive, it could be helpful to discover flaws or weaknesses in the analyzed protocol or to deduce another safe interpretation function allowing us to prove the secrecy property of a protocol. All these cases are illustrated in the case studies section.

4 A NEW AND PRACTICAL SAFE INTERPRETATION FUNCTIONS TO ANALYZE KEYS-AGREEMENT PROTOCOLS

To prove the secrecy property of a cryptographic protocol by the interpretation functions-based method, as seen in the previous Section, we need to have a suitable safe interpretation function. That is why, in (Houmani and Mejri, 2008a; Houmani and Mejri, 2008b) authors proposed a guideline to help to define safe interpretation functions having the following form:

$$\mathsf{F}(\alpha, M) = \mathsf{I} \circ \mathsf{S}(\alpha, M)$$

The function S selects from M some atomic components on which the security level of α depends. This function is called a *selection function*. The function I interprets what S returns as a security type. This function is called a *rank function*.

In addition to the fact that a safe interpretation function F should be a composition of the selection function S and a rank function I, the selection function S should select at least the direct encryption keys. For example $\mathsf{S}(\alpha, \{S, R, \{\alpha, A, N_a, B, C\}_{k_1}\}_{k_2})$ should return k_1 and any subset in $\{A, N_a, B, C\}$. Also, the rank function I should attribute to a message a security level at least equal to its real security level. For instance, if β is a public information, then I cannot interpret it as secret.

As an example of such functions, authors proposed in (Houmani and Mejri, 2008a; Houmani and Mejri, 2008b; Houmani and Mejri, 2007) the DEK and DEKAN functions. the DEK function, denoted by F_{DEK}, attributes a security level of a component α in a message m depending only on the direct keys encrypting α in m. Accordingly, $F_{DEK}(N_b, \{S, N_b\}_{k_{ab}})$ calculates the security level of N_b in the message $\{S, N_B\}_{k_{ab}}$, and it is equal to the security level of k_{ab}. For example, if the security level of k_{ab} is $\{A, B\}$ (meaning that only A and B are eligible to know k_{ab}), then we have:

$$F_{DEK}(N_b, \{S, N_b\}_{k_{ab}}) \quad = \quad \{A, B\}$$

The DEKAN function, denoted by F_{DEKAN} attributes a security level of a component α in a message m depending only on the direct keys encrypting α in m and the neighbours of α in m (the components that can be reach for α without going outside encryptions and usually we consider neighbours that are only identities of agents). Accordingly, $F_{DEKAN}(N_b, \{S, N_b\}_{k_{ab}})$ calculates the security level of N_b in the message $\{S, N_b\}_{k_{ab}}$, and it depends on both the security level of k_{ab} and S. For example, if the security level of k_{ab} is $\{A, B\}$ (meaning that is a shared secret between A and B), then we can fix it as follows:

$$F_{DEKAN}(N_b, \{S, N_b\}_{k_{ab}}) \quad = \quad \{A, B, S\}$$

However, both the DEK function and DEKAN function do not allow to prove the secrecy property of keys-agreement protocols (protocols that allow principals to agree with fresh keys) such as the SSL/TLS protocol. This restriction is due, basically, to the fact that fresh keys are considered by the proposed interpretation functions as initially unknown keys that have unknown security levels and so there is no way to verify if they can encrypt secret information and more in general when these unknown messages affect the security level of other messages.

Since the interpretation functions-based method are not dedicated only to the DEK and DEKAN functions, we refine in this paper these interpretation functions in order to analyze keys-agreement protocols. More precisely, we propose a new way on how we assign the security levels of unknown keys and more in general unknown messages and when the unknown messages affect the security level of other messages.

4.1 Security Levels of Unknown Messages

Almost of formal methods dedicated to analyze cryptographic protocols in the literature (Abadi, 1999; Bugliesi et al., 2004; Debbabi et al., 2001; Gordon and Jeffrey, 2004; Schneider, 1992; Fabrega et al., 1999) consider messages that are not initially known by principals as a message variables in the protocol specification. For instance in Spi-calculus model (Abadi, 1999), CSP model (Schneider, 1992) and strand spaces model (Fabrega et al., 1999), these messages variables are denoted in these models by x, y, z, $\ldots$.

However, these methods differ from each other in how they consider the security levels of these messages variables. In fact, the first works in the formal methods such as CSP-based method (Schneider, 1992), have considered only two levels of security 0 and 1 or *secret* and *public*. However, these kind of security levels does not allow to formalize the fact that principals could not authenticate these messages and they could receive either *secret* or *public* messages. To deal with this problem, Abadi introduced in (Abadi, 1999) a new security level of unknown messages that he called *any*.

In the interpretation functions-based method, authors have proposed to generalize these concepts by introducing the concept of a lattice of security that could be $\{0,1\}$, $\{secret, public, any\}$ or $2^I \cup \overline{\mathcal{X}}$ (I is the set of principals identities and $\overline{\mathcal{X}}$ represents the set of variable security levels). This last set (basically 2^I) aims to attribute to a message a security level of principals that are eligible to know it. For instance, if α has a security level $\{A,B,S\}$, then that means that α is for A, B and S. The set of principals identities gives a precise way to represent the security levels. In fact, the key k_{ab} and the key k_{as} are both *secret* but they are designated to different principals and so they should have different security levels instead of the same (*secret*). The set 2^I allows to express such difference. In the same way, the set $\overline{\mathcal{X}}$, that represents variable security levels, makes difference between variables by giving, for example, to the variable x and the variable y different variable security levels τ_x and τ_y.

However, either the set $\{secret, public, any\}$ or $2^I \cup \overline{\mathcal{X}}$ could not allow to analyze key argreement protocols (protocols that allow two or more participants to agree with fresh keys to secure their future communication). Indeed, a fresh key is an unknown message in the view of some protocol principals and these unknown messages (variables) have security level *any* or security levels in $\overline{\mathcal{X}}$. Hence, these unknown messages (variables) could not be used as a keys to encrypt messages since we are not sure about their security levels.

To deal with this problem, we propose in this paper, to attribute to variables a precise security levels (for example a security level in $\{secret, public\}$ or in 2^I) according to their possible values. In fact, we consider in this paper a security level of a message as the maximum of the security levels of its possible values. Formally, let Γ be a set of substitutions that represents all possible values of the variable x and i be a rank function (function that attributes to a non-variable message a security level), the rank function denoted in the following by $\mathsf{I}^{\mathsf{i}}_{\Gamma}$ and that allows to take into account the security level of all possible values of a variable could be defined as follows:

$$\mathsf{I}^{\mathsf{i}}_{\Gamma}(x) = \begin{cases} \mathsf{i}(\alpha) & \text{if } \alpha \notin \mathcal{X} \\ \bigsqcap\limits_{\sigma\in\Gamma} \mathsf{I}^{\mathsf{i}}_{\Gamma}(x\sigma) & \text{else} \end{cases}$$

4.2 What Affect the Security Levels of Messages

In the interpretation functions-based method, a selection function S selects elements at some distances meaning that these elements could affect the security levels. For instance, let suppose that an intruder could have the message $\{s,A\}_k$ and the message $\{s,x\}_k$ and the security level of s depends on the encryption key k and the identity A. In this case, the intruder could send the message $\{s,x\}_k$ instead of the message $\{s,A\}_k$ if he could substitute the variable x by its principal identity for example. Hence, the unknown message x in this case could affect the security of s.

Now, an intruder could have the message $\{s\}_x$ and the message $\{z\}_y$. It is obvious that the message $\{s\}_x$ could be sent instead of the message $\{z\}_y$, and this could lower the security level of s. Indeed, if for example the security level of z is *public* and the security level of s is *secret*, then if the intruder replace the message $\{z\}_k$ by $\{s\}_k$, then the receiver will think that s has security level *public* since it instance the unknown message z and so he could send s in clear what will be a breach of secrecy. Therefore, the unknown message y here could affect the security of s.

To sum up, the selection function S should select only the unknown messages that could affect the security levels of messages. The unknown messages that could affect the security level are those when instanced by some values are selected by the selection function. Formally, let s be a selection function and Γ a set of possible substitutions (a set of possible values) of unknown messages. Then, the selection function that select the unknown messages which affect the security levels of message denoted by S^{s}_{Γ} could be defined as follows:

$$\mathsf{S}^{\mathsf{s}}_{\Gamma}(\alpha,m) = (s(\alpha,m)\backslash\mathcal{X}) \cup \{x_i \in Dom(\Gamma) \cap \mathcal{X} | \exists\sigma \in \Gamma, \exists\beta \in \{\alpha\} \cup (s(\alpha,m\sigma)\backslash\mathcal{X}) \cdot \beta \in \{x_i\sigma\}_{\downarrow C}\}$$

4.3 Safe Interpretation Functions to Analyze Key-agreement Protocols

In the following, we prove that by selecting the unknown messages that only affect the security levels of messages and by assigning to those unknown messages the maximum of the security level of theirs possible values, we can construct a safe interpretation functions that could be used to analyze the security of keys-agreement protocols such as SSL/TLS. In fact, let s be a selection function, i be a rank function and Γ be a set of substitutions. Suppose that the rank function $\mathsf{I}^{\mathsf{i}}_{\Gamma}$ and the selection function S^{s}_{Γ} are those defined respectively in 4.1 and in 4.2. let define the interpretation functions that have the following form:

$$\mathsf{F}_\Gamma(\alpha, m) = \mathsf{I}_\Gamma^{\mathsf{i}} \circ S_\Gamma^{s}(\alpha, m)$$

Now, we denote by DEK_Γ the interpretation function that allows to give to a message a security level according to its encryption keys. Formally:

Definition 1. *Let $\mathcal{C} = \langle \mathcal{M},\ \models,\ \mathcal{K},\ \mathcal{L}^{\sqsupseteq},\ \ulcorner\cdot\urcorner\rangle$ be a context of verification, s^k be a selection function that allows to select direct encryption key and i^k be a rank function that allows to give to an atomic message a security level as follows: $\mathsf{i}^k(\alpha) = \ulcorner(\alpha)^{-1}\urcorner$. We define the DEK_Γ function as follows:*

$$\mathsf{DEK}_\Gamma = \mathsf{I}_\Gamma^{\mathsf{i}^k} \circ \mathsf{S}_\Gamma^{\mathsf{s}^k}$$

Recall that $\mathsf{I}_\Gamma^{\mathsf{i}^k}$ will give to keys their exact security level according to their possible values given by Γ. The selection function $\mathsf{S}_\Gamma^{\mathsf{s}^k}$ will allow to select only keys and the unknown keys that could affect the security level of messages.

Example 1. *Let $\mathcal{C} = \langle \mathcal{M},\ \models,\ \mathcal{K},\ \mathcal{L}^{\sqsupseteq},\ \ulcorner\cdot\urcorner\rangle$ be a context of verification and $\Gamma = \{[x \mapsto k_a^{-1}], [x \mapsto k_{ab}]\}$. Then, the security level of α in the message $\{S, \{\alpha, A, B, N_a\}_{k_{as}}\}_{ab}$ according to Γ is as follows:*

$$\begin{aligned}
&\mathsf{DEK}_\Gamma(\alpha, \{S, \{\alpha, A, B, N_a\}_{k_{as}}\}_{ab})\\
&\quad = \mathsf{I}_\Gamma^{k} \circ \mathsf{S}^{k}(\alpha, \{S, \{\alpha, A, B, N_a\}_{k_{as}}\}_{ab})\\
&\quad = \mathsf{I}_\Gamma^{k}(k_{as})\\
&\quad = \ulcorner k_{as} \urcorner\\
&\quad = \{A, S\}
\end{aligned}$$

the security level of α in the message $\{\alpha, A, B, N_a\}_x$ according to Γ is as follows:

$$\begin{aligned}
&\mathsf{DEK}_\Gamma(\alpha, \{\alpha, A, B, N_a\}_x)\\
&\quad = \mathsf{I}_\Gamma^{k} \circ \mathsf{S}^{k}(\alpha, (\alpha, \{\alpha, A, B, N_a\}_x)\\
&\quad = \mathsf{I}_\Gamma^{k}(x)\\
&\quad = \ulcorner k_{ab} \urcorner \cup \ulcorner k_a \urcorner\\
&\quad = \{A, B\} \cup I\\
&\quad = I\\
&\quad = \bot
\end{aligned}$$

The interpretation function DEK_Γ is safe (could not be misled by intruder manipulations). Indeed, the security level of a message depend on its direct encrypted key and so the message could be known only by the agent whose know the keys of encryption. Formally, we have:

Theorem 1. *Let $\mathcal{C}$ be a context of verification and Γ be a set of substitutions. DEK_Γ is a $\mathcal{C}$-safe interpretation function.*

Proof. As proved in (Houmani and Mejri, 2008a; Houmani and Mejri, 2008b), any interpretation function that is a composition of a selection function and a rank function and in which the selection function selects at least the direct encryption keys and the rank function attributes to these keys the default security levels of their inverse keys, is safe interpretation function. The function DEK_Γ respects these conditions and so it is safe. □

Now, let define DIN_Γ as an interpretation function that attributes to a message a security level according to the principal identities that are neighbors to this message. Formally:

Definition 2. *Let s^n be a selection function that allows to select direct identities neighbors and i^n be a rank function that allows to give a security level to an atomic message as follows $\mathsf{i}^n(A) = \{A\}$ if $A \in I$ and $\mathsf{i}^n(\alpha) = \ulcorner\alpha\urcorner$ else. The interpretation function DIN_Γ could be defined as follows:*

$$\mathsf{DIN}_\Gamma = \mathsf{I}_\Gamma^{\mathsf{i}^n} \circ \mathsf{S}_\Gamma^{\mathsf{s}^n}$$

The interpretation function DIN_Γ is not safe since it does not take into account whether a message is encrypted or not and what is its encrypted keys. Hence, there si nor way to ensure the confidentality of the messages or to know who can know them. Nevertheless, we can combine it with the interpretation function DEK_Γ to have a safe one. Formally:

Definition 3. *Let $\mathcal{C}$ be a context of verification, Γ be a set of substitutions and DINEK_Γ be an interpretation function that respect the following syntax:*

$$\mathsf{DINEK}_\Gamma = \mathsf{DIN}_\Gamma \sqcap \mathsf{DEK}_\Gamma$$

The interpretation function DINEK_Γ allows to attribute to a message a security level according to its direct encryption keys and the direct identities neighbors. The following example shows how this function works.

Example 2. *In this example, let $\mathcal{C} = \langle \mathcal{M},\ \models,\ \mathcal{K},\ \mathcal{L}^{\sqsupseteq},\ \ulcorner\cdot\urcorner\rangle$ be a context of verification, $\Gamma_1 = \{[x \mapsto N_b], [y \mapsto IdSession]\}$ $\Gamma_2 = \{[x \mapsto N_b], [x \mapsto I], [y \mapsto IdSession]\}$. Then, the security level of α in the message $\{S, y, \{\alpha, A, B, N_a, x\}_{k_{as}}\}_{ab}$ according to Γ_1 is as follows:*

$$\mathsf{DINEK}_{\Gamma_1}(\alpha, \{S, y, \{\alpha, A, B, N_a, x\}_{k_{as}}\}_{k_{ab}}) = \{A, B, S\}$$

Indeed, we have:

$$\begin{aligned}
&\mathsf{DIN}_{\Gamma_1}(\alpha, \{S, y, \{\alpha, A, B, N_a, x\}_{k_{as}}\}_{k_{ab}})\\
&\quad = \mathsf{I}_{\Gamma_1}^{\mathsf{i}^n} \circ \mathsf{S}^{n}(\alpha, \{S, y, \{\alpha, A, B, N_a, x\}_{k_{as}}\}_{k_{ab}})\\
&\quad = \mathsf{I}_{\Gamma_1}^{\mathsf{i}^n}(A, B)\\
&\quad = \mathsf{i}^n(A) \cup \mathsf{i}^n(B)\\
&\quad = \{A\} \cup \{B\}\\
&\quad = \{A, B\}
\end{aligned}$$

and we have:

$$\begin{aligned}&\mathsf{DEK}_{\Gamma_1}(\alpha,\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{k_{ab}})\\&\quad=\mathsf{I}^k_{\Gamma_1}\circ\mathsf{S}^k(\alpha,\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{k_{ab}})\\&\quad=\mathsf{I}^k_{\Gamma_1}(k_{as})\\&\quad=\{A,S\}\end{aligned}$$

The security level of α *in the message* $\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{ab}$ *according to* Γ_2 *is as follows:*

$$\mathsf{DINEK}_{\Gamma_2}(\alpha,\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{ab})=\{A,B,S,I\}$$

Indeed, we have:

$$\begin{aligned}&\mathsf{DIN}_{\Gamma_2}(\alpha,\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{k_{ab}})\\&\quad=\mathsf{I}^{\mathsf{i}^n}_{\Gamma_2}\circ\mathsf{S}^n(\alpha,\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{k_{ab}})\\&\quad=\mathsf{I}^{\mathsf{i}^n}_{\Gamma_2}(A,B,)\\&\quad=\mathsf{i}^n(A)\cup\mathsf{i}^n(B)\cup\mathsf{i}^n(x[x\mapsto N_b])\cup\mathsf{i}^n(x[x\mapsto I])\\&\quad=\{A\}\cup\{B\}\cup\emptyset\cup\{I\}\\&\quad=\{A,B,I\}\end{aligned}$$

and we have:

$$\begin{aligned}&\mathsf{DEK}_{\Gamma_2}(\alpha,\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{k_{ab}})\\&\quad=\mathsf{I}^k_{\Gamma_2}\circ\mathsf{S}^k(\alpha,\{S,y,\{\alpha,A,B,N_a,x\}_{k_{as}}\}_{k_{ab}})\\&\quad=\mathsf{I}^k_{\Gamma_2}(k_{as})\\&\quad=\ulcorner k_{as}\urcorner\\&\quad=\{A,S\}\end{aligned}$$

Notice, that we can use the interpretation function DINEK_Γ *with other lattice of security like* $\{0,1\}$, $\{secret, public\}$ *and* $\{secret, any, public\}$. *Recall that the function* DINEK_Γ *is a safe interpretation function (lemma 2).*

In the following theorem we prove that DINEK_Γ is safe.

Theorem 2. *Let* $\mathcal{C}$ *be a context of verification,* Γ *be a set of substitutions. Then, the interpretation function* DINEK_Γ *is* $\mathcal{C}$*-safe.*

Proof. As proved in (Houmani and Mejri, 2008a; Houmani and Mejri, 2008b), any interpretation function that is a composition a selection function and rank function and in which the selection function selects at least the direct encryption keys and the rank function attributes to these keys the default security levels of the their inverse keys, is safe interpretation function. The function DINEK_Γ respects these conditions and so it is safe.

□

4.4 Bounded Verification for Unbounded Executions

We have defined safe interpretation functions by using the set of possible substitutions of variables Γ that represents the set of possible values that could be taken by variables in the set of all possible protocol executions. However, the set of possible executions of a protocol is infinite and hence the set Γ is also infinite. This last fact could make the verification process infinite and so impossible. To deal with this problem we prove hereafter that the set Γ could be reduced to a finite one. In fact, we can reduce the set of all possible values of protocol variables to $\Gamma_{\mathcal{C}}(p)$ the set of the most general unifiers (mgu) that unify the messages that could be inferred by the intruder from the protocol specification. Formally, let $\mathcal{C}$ be a context of verification, p be a protocol and $\mathcal{M}(p)$ is the set of messages that are in the specification of the protocol and $\mathcal{M}(p)_{\downarrow_{\mathcal{C}}}$[1] is the normal form obtained by applying the intruder rules and capacities defined in $\mathcal{C}$ to the set $\mathcal{M}(p)$, we define $\Gamma_{\mathcal{C}}(p)$ as follows:

$$\begin{aligned}\Gamma_{\mathcal{C}}(p)\quad=\quad&\{\sigma\in\Gamma|\exists m_1,m_2\in(\mathcal{M}(p))_{\downarrow_{\mathcal{C}}}\cdot\\&\sigma=mgu(m_1,m_2)\}\end{aligned}$$

The idea behind using the set $\Gamma_{\mathcal{C}}(p)$ could be summarized by these tree facts:

1. Any execution of a protocol is a substitution of a role-based specification where the received messages are deduced from the intruder capacities and the sent messages. Hence, the set of all possible protocol executions could be represented by the set of all possible substitutions of protocol roles-based specification including the substitutions made by an intruder in order to misled a principal.
2. The behaviors of the honest principals when executing a protocol are the same. For instance, if the protocol have two roles A and B, and the principal C wants to execute the protocol they should play the role of A or B and in this case C could not do what A or B are not able to do. Therefore, we can reduce the set of all possible honest executions to the set of one execution of the protocol. Notice that an execution of a protocol in the model considered here, is a substitution of roles-based specification. The set of all possible substitutions that represents the set of all possible executions conducted by honest principals could be reduced to the substitutions that unifies the roles-based specification.
3. A dishonest principal (an intruder) could perform an attack and execute a protocol in our model if

[1] The set of messages that could be inferred by an intruder is finite when the orienting the equational theory form left to right and by bounding the number of functions that contruct the messages. In this paper, we do not deal with non-convergent equational theories which is could be subject to future works.

he could deduce all its sent messages from the received ones. Hence, the set of all its possible executions could be represented by the set denoted by $\mathcal{M}(p)_{\downarrow_{\mathcal{C}}}$ represents the set of all substitutions that could be obtained by unifying the messages deduced from the protocol messages and its capacities. Also, suppose that an intruder receives the message $\{\alpha, s\}_k$ and he is able to know s by using his capacities and suppose that this message is the instantiation of the message $\{\alpha, x\}_k$ in the protocol roles-based specification, then the intruder will be able to deduce also x from the roles-based specification and its capacities. Therefore, the set of the substitutions of possible attacks could be reduced to the set of substitutions obtained by unifying messages that could be inferred from the exchanged messages and the intruder capacities.

To sum up, the set of all possible executions of a protocol could be reduced to the set of substitutions obtained by unifying messages that could be inferred from the exchanged messages in the roles-based specification and the intruder capacities. Hence, we prove hereafter that the set $\Gamma_{\mathcal{C}}(p)$ is sufficient to analyze the secrecy property of the protocol p.

Theorem 3. *Let $\mathcal{C}$ be a context of verification, p be a protocol, Γ the set of all possible substitutions that represents the values of variables in all possible executions of the protocol p and F_{Γ} is a safe interpretation function. Then, if p is F_{Γ}-increasing if and only if p is $\mathsf{F}_{\Gamma_{\mathcal{C}}(p)}$-increasing.*

Proof. The detailled proofs is removed due the number of pages but in the following, we present the scetch of this proof. We use the set $\mathcal{M}(p)$ (the set of all messages exchanged in the protocol specification) because a valid trace of a protocol is an interweaving of substitutions of prefixes of the protocol specification where the sent messages could be inferred by the intruder. Therefore, we need to know what messages could be inferred from the protocol and intruder capacities. More precisely, we need to know what messages could be used to replace other messages by using the protocol and the intruder capacities. Hence, we search for messages that are in the set $\mathcal{M}(p))_{\downarrow_{\mathcal{C}}}$ that represents the set of all messages that could be obtained by the intruder by listening to the network and by applying his capacities to deduce new messages. Also, any substitution of role-based specification σ could be written as a composition of two substitutions σ_1 and σ_2 (i.e $\sigma = \sigma_1 \circ \sigma_2$), where σ_2 is in $\Gamma_{\mathcal{C}}(p)$ and σ_1 is a substitution that rename identities. Hence, the intruder could perform any attack by considering only the number of protocol participants in the description of that protocol. □

For the sake of simplicity, we will denote, in the remainder of this paper, F_p, S_p and I_p instead of $\mathsf{F}_{\Gamma_{\mathcal{C}}(p)}$, $S_{\Gamma_{\mathcal{C}}(p)}$ and $\mathsf{I}_{\Gamma_{\mathcal{C}}(p)}$ respectively.

Accordingly, the secrecy property of a protocol p is guaranteed when the protocol is increasing according to a safe interpretation function and the set $\Gamma_{\mathcal{C}}(p)$. Hence, to analyze the secrecy property of a protocol by using the interpretation functions-based method, we have to compute first the set $\Gamma_{\mathcal{C}}(p)$ and after that we can define an interpretation function that will use the set $\Gamma_{\mathcal{C}}(p)$ to calculate the security levels in the sent and received messages in the protocol in order to verify whether the protocol is increasing. For instance in the case of SSL/TLS protocol, suppose that the set $\Gamma_{\mathcal{C}_{TLS}}(p_{TLS})$ (or simply Γ_{TLS}, is defined as follows:

$$\begin{array}{rcl}\Gamma_{TLS} & = & \{[X_1 \mapsto N_s, X_2 \mapsto Ver_s, Y_1 \mapsto N_c, \\ & & Y_2 \mapsto Verc_c, Y_3 \mapsto IdSession^i, \\ & & Y_4 \mapsto Secret_c]\}\end{array}$$

Then, the security level of α in the message $\{\alpha, X_1, B\}_{k_{as}}$ according to the function DINEK and the set Γ_{TLS}, is as follows:

$$\begin{array}{l}\mathsf{DINEK}_{TSL}(\alpha, \{\alpha, X_1, B\}_{k_{as}}) \\ \quad = i^n(B) \cup i^n(X_1[X_1 \mapsto N_s]) \cup i^k(k_{as}) \\ \quad = \{B\} \cup \emptyset \cup \{A, S\} \\ \quad = \{A, B, S\}\end{array}$$

Hence, only A, B and S are eligible to know α in this case. For the sake of simplicity, we will use F_{TSL} instead of DINEK_{TSL}.

5 ANALYSIS OF THE SSL/TLS HANDSHAKE PROTOCOL

In this section we analyze the SSL/TLS handshake protocol. To that end we need to define first as shown by Figure **??**, the context of verification. Second, we model the SSL/TLS handshake protocol as a roles-based specification. Finally, we prove that roles-based specification of SSL/TLS handshake protocol is increasing according to the DINEK function and the Dolev and Yao intruder model (we suppose the perfect encryption hypothesis) and so the secrecy property of the SSL/TLS handshake protocol is guaranteed.

Context of Verification. A context of verification in the interpretation function method is basically the class of protocols that could be defined by the message algebra and the set of intruder capacities. Let $\mathcal{C}_{TLS}$ be the context of verification that we will consider for the analysis of the SSL/TLS handshake protocol. The message algebra, in this example, is given by the set of names $\mathcal{N}_{TLS}$ and the set Σ_{TLS}. The set of intruder capacities is the set of intruder rules denoted by $\models_{TLS}$ and the set of equational theory denoted by $\mathcal{E}_{TLS}$. In addition, we consider that the context of verification contains the lattice of security $\mathcal{L}_{TLS}$, the initial knowledge of principals $\mathcal{K}_{TLS}$ and the security levels of messages given in the description of the protocol and described by $\ulcorner . \urcorner_{TLS}$. The lattice of security describes security levels space. Initial knowledge of principals are what the principals know before executing the protocol. The security levels of atomic messages involved in the protocol is an environment that attributes to each message its security level.

In this example, the set of names $\mathcal{N}_{TLS}$ could be the set of atomic messages given by the following BNF grammar:

n	$::=$	A	(Principal Identifier)
	$\mid$	$IdSession$	(Session Identifier)
	$\mid$	Ver_a	(Protocol Version)
	$\mid$	$Secret_c$	(Secret)
	$\mid$	N_a	(Nounce)
	$\mid$	k_a^{-1}	(Private key)
	$\mid$	k_a	(Public key)
	$\mid$	k_{ab}	(Shared key)

and $\Sigma_{TLS} = \{pair, fst, snd, enc, dec, sign, check, H, g_1, g_2, g_3\}$

As usual we can write $\{m\}_k$ instead of $enc(m,k)$ or $sign(m,k)$. Also, we can write m_1, m_2 instead of $pair(m_1, m_2)$.

Hence, the set of messages $\mathcal{M}_{TLS}$ is defined by the following BNF rules:

m	$::=$	n	
	$\mid$	$pair(m_1, m_2)$	(Pair Function)
	$\mid$	$g_i(m)$	(Compression Function)
	$\mid$	$H(m)$	(Hash Function)
	$\mid$	$enc(m,k)$	(Encryption Function)
	$\mid$	$dec(m,k)$	(Decryption Function)
	$\mid$	$sign(m, k_a^{-1})$	(Signature Function)
	$\mid$	$check(m, k_a)$	(Checking Signature)

In this paper, we consider a hashed message as a message that is encrypted by a public key K_h and no one could know the inverse of this key. Thus assumption is used to say that any one could hash a message and no one could know some thing about the original message from the hashed message.

The intruder rules $\models_{TLS}$ are as follows:

The equational theory $\mathcal{E}_{TLS}$ contains the following equations:

$$\begin{aligned} fst(pair(x,y)) &= x \\ snd(pair(x,y)) &= y \\ dec(enc(x,k_y),k_y^{-1}) &= x \\ g_i(g_i(x)) &= x \qquad i \in \{1,2,3\} \\ check(sign(x,k_y^{-1}),k_y) &= ok \end{aligned}$$

Let $\models_{TLS}$ denotes the following rules of intruder:

$$\text{(knowledge)} \qquad \frac{\Box}{M \models_0 m}[m \in M]$$

$$\text{(construct)} \qquad \frac{M \models_{TLS} m_1 \ \dots \ M \models_{TLS} m_n}{M \models_{TLS} f(m_1, \dots, m_n)}[f \in \Sigma_0]$$

$$(\mathcal{E}\text{-equality}) \qquad \frac{M \models_{TLS} m}{M \models_{TLS} m'}[m =_{\mathcal{E}_0} m']$$

Therefore, when an intruder could deduce a message m from a set of messages, we denote by $M \models_{\mathcal{E}} m$. The intruder model $\models_{TLS}$ and the equational theory $\mathcal{E}_{TLS}$ represents the famous Dolev and Yao model.

The initial knowledge of principals $\mathcal{K}_{TLS}$ could be as follows: each principal knows his identity, the identity of other principals, his public and private key and all the public keys of the other principals. Also, each principal can generate fresh values.

The security lattice $\mathcal{L}_{TLS}$ is $\mathcal{L}_0 = (2^I, \subseteq)$. In fact, the security level of a message is simply the set of principals that are eligible to know its value. Therefore, the supremum of this lattice $\top$ is equal to $\emptyset$ and the infimum $\bot$ is equal to I_X (the set of principal identities).

The types environment $\ulcorner . \urcorner_{TLS}$ could be any partial function from $\mathcal{M}_{TLS}$ to $\mathcal{L}_{TLS}$. In this example, we choose this environment as follows:

$$\begin{aligned} &[Secret_c \mapsto \{C,S\}, N_c, N_s, Ver_c, Vers_s, IdSession \mapsto \bot, \\ &\ K_c, K_s \mapsto \bot, k_s^{-1} \mapsto \{S\}, k_c^{-1} \mapsto \{c\}] \end{aligned}$$

5.1 SSL/TLS Roles-based Specification

Recall that the roles-based specification is a set of the prefixes of generalized roles. A generalized role is a protocol abstraction, where the emphasis is put upon a particular principal and where all the unknown messages are replaced by variables. Also, an exponent i (the session identifier) is added to each fresh message to emphasize that these components change their values from one run to another. For more details on how we can compute a roles-based specification from a protocol and a context of verification we refer the reader to (Houmani and Mejri, 2008a; Houmani and Mejri, 2008b). Also, any other specification could be used to conduct this proof as strand spaces (Fabrega et al., 1999), CSP (Schneider, 1996) or Pi-calcul (Abadi, 1999).

The SSL/TLS roles-based specification is:

$$\mathcal{R}_G(p_{NSL}) = \{\mathcal{C}_G^1, \mathcal{C}_G^2, \mathcal{C}_G^3, \mathcal{S}_G^1, \mathcal{S}_G^2, \mathcal{S}_G^3, \}$$

The generalized roles $\mathcal{C}_G^1$, $\mathcal{C}_G^2$ and $\mathcal{C}_G^3$ are as follows:

$$\mathcal{C}_G^1 = \begin{array}{lllll} i.1. & C \to I(S) & : & m_1^C \end{array}$$

$$\mathcal{C}_G^2 = \begin{array}{lllll} i.1. & C \to I(S) & : & m_1^C \\ i.2. & I(S) \to C & : & m_2^C \\ i.3. & C \to I(S) & : & m_3^C \end{array}$$

$$\mathcal{C}_G^3 = \begin{array}{lllll} i.1. & C \to I(S) & : & m_1^C \\ i.2. & I(S) \to C & : & m_2^C \\ i.3. & C \to I(S) & : & m_3^C \\ i.4. & I(S) \to C & : & m_4^C \\ i.5. & C \to I(S) & : & m_5^C \end{array}$$

where

$$\begin{array}{lcl} m_1^C & = & C, N_c, Ver_c, IdSession^i \\ m_2^C & = & S, X_1, X_2, IdSession^i, CA(S, K_s) \\ m_3^C & = & IdSession^i, \{Ver_c, Secret_c, C, S\}_{K_s} \\ & & CA(C, K_c), \{H(g_1(m_1^C, m_2^C, Secret_c, C, S))\}_{K_c^{-1}} \\ m_4^C & = & \{H(g_2(m_1^C, m_2^C, m_3^C, Secret_c, C, S))\}_{K_{cs}} \\ m_5^C & = & \{H(g_3(m_1^C, m_2^C, m_3^C, m_4^C, Secret_c, C, S))\}_{K_{cs}} \\ K_{cs} & = & Master(Secrect_c, N_c, X_1) \end{array}$$

The generalized roles $\mathcal{S}_G^1$, $\mathcal{S}_G^2$ and $\mathcal{S}_G^3$ are as follows:

$$\mathcal{S}_G^1 = \begin{array}{lllll} i.1. & I(C) \to S & : & m_1^S \\ i.2. & S \to I(C) & : & m_2^S \end{array}$$

$$\mathcal{S}_G^2 = \begin{array}{lllll} i.1. & I(C) \to S & : & m_1^S \\ i.2. & S \to I(C) & : & m_2^S \\ i.3. & I(C) \to S & : & m_3^S \\ i.4. & S \to I(C) & : & m_4^S \end{array}$$

$$\mathcal{S}_G^3 = \begin{array}{lllll} i.1. & I(C) \to S & : & m_1^S \\ i.2. & S \to I(C) & : & m_2^S \\ i.3. & I(C) \to S & : & m_3^S \\ i.4. & S \to I(C) & : & m_4^S \\ i.5. & I(C) \to S & : & m_5^S \end{array}$$

where

$$\begin{array}{lcl} m_1^S & = & C, Y_1, Y_2, Y_3 \\ m_2^S & = & S, N_s, Ver_s, Y_3, CA(S, K_s) \\ m_3^S & = & Y_3, \{Y_1, Y_3, C, S\}_{K_s}, \\ & & CA(C, K_c), \{H(g_1(m_1^S, m_2^S, Y_4, C, S))\}_{K_c^{-1}} \\ m_4^S & = & \{H(g_2(m_1^S, m_2^S, m_3^S, Y_4, C, S))\}_{K_{cs}} \\ m_5^S & = & \{H(g_3(m_1^S, m_2^S, m_3^S, m_4^S, Y_4, C, S))\}_{K_{cs}} \\ K_{cs} & = & Master(Y_4, Y_1, N_s) \end{array}$$

To define the interpretation function that will help to verify the secrecy property of the SSL/TLS handshake protocol we should first (as we have seen in Section 4) the set $\Gamma_{\mathcal{C}}(p)$. In fact, we have:

$$\mathcal{M}(p) = \{m_1^C, m_2^C, m_3^C, m_4^C, m_5^C, m_1^S, m_2^S, m_3^S, m_4^S, m_5^S\}$$

The set of messages that could be inferred by the intruder is as follows:

$$\begin{array}{lcl} (\mathcal{M}(p))_{\downarrow C} & = & \mathcal{M}(p) \cup \{Y_3, \{Y_1, Y_4, C, S\}_{K_s}\} \cup \\ & & \{CA(C, K_c)\}\ cup \\ & & \{\{H(g_1(m_1^S, m_2^S, Y_4, C, S))\}_{K_c^{-1}}\} \cup \\ & & \{\{IdSession^i, Ver_c, Secret_c, C, S\}_{K_s}\} \\ & & \{\{H(g_1(m_1^1), m_2^C, Secret_c, C, S))\}_{K_c^{-1}}\} \end{array}$$

Therefore, the set $\Gamma_{\mathcal{C}_{TLS}}(p_{TLS})$ or simply Γ_{TLS} is as follows:

$$\begin{array}{c} \Gamma_{TLS} = \{[X_1 \mapsto N_s, X_2 \mapsto Ver_s, Y_1 \mapsto N_c, Y_2 \mapsto Verc_c\} \\ \{Y_3 \mapsto IdSession^i, Y_4 \mapsto Secret_c]\} \end{array}$$

Now, we are ready to choose or define a safe interpretation function. In this example, we will use DINEK_{TLS} function. Recall that this function allows to assign to a message a security level according to it its direct identities neighbors and the direct encryption keys. Also, recall that the DINEK_{TLS} allows to take into account the variables that could take values as identities neighbors or encryptions keys (see the section 4 for formal definition). For the sake of simplicity, we will use in the remainder of this paper the notation F_{TLS} instead of DINEK_{TLS}.

5.2 Secrecy Property of the SSL/TLS Handshake Protocol

In this section, we analyze the secrecy property of the SSL/TLS Handshake Protocol. To that end, we verify whether the roles-based specification is increasing according to the $sfDinek$ function. Unformally, we verify whether principals do not decrease the security levels of messages when sending them over the networks. The security levels are estimated by using the $sfDinek$ function denoted by F_{TLS} and that gives a security level to α in according to its direct identities neighbors and direct encryption Keys.

From the generalized role$\mathcal{C}_G^1$, we deduce that:

$$\begin{array}{lcl} \mathcal{C}_G^{1^-} & = & \emptyset \\ \mathcal{C}_G^{1^+} & = & (m_1^C = C, N_c, Ver_c, IdSession^i) \end{array}$$

In this role, the sent messages are C, N_c, Ver_c and $IdSession^i$. These messages have the security level $\perp$, i.e $\ulcorner\alpha\urcorner = \perp$ for all $\alpha \in \{C, N_c, Ver_c, IdSession^i\}$. Hence, the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{1^+}) \sqsupseteq \ulcorner\alpha\urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{1^-})$ will be always true for all $\alpha \in \{C, N_c, Ver_c, IdSession^i\}$ and so the role $\mathcal{C}_G^1$ is increasing.

From the generalized role $\mathcal{C}_G^2$, we deduce that:

$$\begin{array}{lcl} \mathcal{C}_G^{2^-} & = & (m_2^C = S, X_1, X_2, IdSession^i, CA(S, K_s)) \\ \mathcal{C}_G^{2^+} & = & (m_3^C = IdSession^i, \{Ver_c, Secret_c, C, S\}_{K_s}, \\ & & CA(C, K_c), \{H(g_1(m_1^C, m_2^C, Secret_c, C, S))\}_{K_c^{-1}}) \end{array}$$

In the role $\mathcal{C}_G^2$, the sent messages are C, S, Ver_c, $Secret_c$, X_1, X_2 and $IdSession^i$. The messages C, S, Ver_c, and $IdSession^i$ have the security level $\bot$, i.e $\ulcorner\alpha\urcorner = \bot$ for all $\alpha \in \{C, S, Ver_c, IdSession^i\}$. Hence, the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2+}) \sqsupseteq \ulcorner\alpha\urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2-})$ will be always true for all $\alpha \in \{C, S, Ver_c, IdSession^i\}$.

Now, let's verify the equation for the messages X_1, X_2 and $Secret_c$. The security level of these messages are as follows:

$$\ulcorner Secret_c\urcorner = \{C, S\} \quad \text{and} \quad \ulcorner X_1\urcorner = \ulcorner X_2\urcorner = \top$$

The security level of X_1, X_2 and $Secret_c$ obtained by the function F_{TLS} according to sent and received messages in $\mathcal{C}_G^2$ are as follows:

α	m	$\mathsf{DIN}_{TLS}(\alpha,m)$	$\mathsf{DEK}_{TLS}(\alpha,m)$	$\mathsf{F}_{TLS}(\alpha,m)$
X_1	$\mathcal{C}_G^{2-}$	$\{S\}$	$\bot$	$\bot$
X_1	$\mathcal{C}_G^{2+}$	$\{C,S\}$	$\top$	$\{C,S\}$
X_2	$\mathcal{C}_G^{2-}$	$\{S\}$	$\bot$	$\bot$
X_2	$\mathcal{C}_G^{2+}$	$\{C,S\}$	$\top$	$\{C,S\}$
$Secret_c$	$\mathcal{C}_G^{2-}$	$\emptyset t$	$\top$	$\top$
$Secret_c$	$\mathcal{C}_G^{3+}$	$\{C,S\}$	$\{S\}$	$\{C,S\}$

Recall that $\mathsf{F}_{TLS} = \mathsf{DIN} \cup \mathsf{DEK}$ and allows to attribute to a message a security level that depends on its direct encrypted keys and direct principals identities.

From the previous equations we can also deduce that the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2+}) \sqsupseteq \ulcorner\alpha\urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2-})$ is true for all $\alpha \in \{X_1, X_2, Secret_c\}$ and so the role $\mathcal{C}_G^2$ is increasing.

From the generalized role $\mathcal{C}_G^3$, we deduce that:

$$\mathcal{C}_G^{3-} = (m_4^C = \{H(g_2(m_1^C, m_2^C, m_3^C, Secret_c, C, S))\}_{K_{cs}})$$
$$\mathcal{C}_G^{3+} = (m_5^C = \{H(g_3(m_1^C, m_2^C, m_3^C, m_4^C, Secret_c, C, S))\}_{K_{cs}})$$

In the role $\mathcal{C}_G^3$, the sent messages are C, S, Ver_c, $Secret_c$, X_1, X_2 and $IdSession^i$. The messages C, S, Ver_c, and $IdSession^i$ have the security level $\bot$, i.e $\ulcorner\alpha\urcorner = \bot$ for all $\alpha \in \{C, S, Ver_c, IdSession^i\}$. Hence, the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2+}) \sqsupseteq \ulcorner\alpha\urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2-})$ will be always true for all $\alpha \in \{C, S, Ver_c, IdSession^i\}$. Now, let's verify the equation for the messages X_1, X_2 and $Secret_c$. The security level of these messages are as follows:

$$\ulcorner Secret_c\urcorner = \{C, S\} \quad \text{and} \quad \ulcorner X_1\urcorner = \ulcorner X_2\urcorner = \top$$

The security level of X_1, X_2 and $Secret_c$ obtained by the function F_{TLS} according to sent and received messages in $\mathcal{C}_G^3$ are as follows:

α	m	$\mathsf{DIN}_{TLS}(\alpha,m))$	$\mathsf{DEK}_{TLS}(\alpha,m)$	$\mathsf{F}_{TLS}(\alpha,m)$
X_1	$\mathcal{C}_G^{3-}$	$\{S\}$	$\bot$	$\bot$
X_1	$\mathcal{C}_G^{3+}$	$\{C,S\}$	$\top$	$\{C,S\}$
X_2	$\mathcal{C}_G^{3-}$	$\{S\}$	$\bot$	$\bot$
X_2	$\mathcal{C}_G^{3+}$	$\{C,S\}$	$\top$	$\{C,S\}$
$Secret_c$	$\mathcal{C}_G^{3-}$	$\emptyset$	$\top$	$\top$
$Secret_c$	$\mathcal{C}_G^{3+}$	$\{C,S\}$	$\{S\}$	$\{C,S\}$

From the previous Table we can also deduce that the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2+}) \sqsupseteq \ulcorner\alpha\urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{2-})$ is true for all $\alpha \in \{X_1, X_2, Secret_c\}$ and so the role $\mathcal{C}_G^{2+}$ is increasing.

To sum up, the generalized roles of C are increasing since they satisfy the equation

$$(eq1) \quad \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{i+}) \sqsupseteq \ulcorner\alpha\urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{i-})$$

Indeed, we have:

α	r	$\ulcorner\alpha\urcorner_0$	$\mathsf{F}_{TLS}(\alpha,r^+)$	$\mathsf{F}_{TLS}(\alpha,r^-)$	$eq1$
X_1	$\mathcal{C}_G^2$	$\top$	$\bot$	$\top$	*Yes*
X_1	$\mathcal{C}_G^3$	$\top$	$\bot$	$\top$	*Yes*
X_2	$\mathcal{C}_G^2$	$\top$	$\bot$	$\top$	*Yes*
X_2	$\mathcal{C}_G^3$	$\top$	$\bot$	$\top$	*Yes*
$Secret_c$	$\mathcal{C}_G^2$	$\{C,S\}$	$\{C,S\}$	$\top$	*Yes*
$Secret_c$	$\mathcal{C}_G^3$	$\{C,S\}$	$\{C,S\}$	$\{C,S\}$	*Yes*

From the generalized roles of S, we deduce that:

$$\begin{array}{rcl} \mathcal{S}_G^{1\,-} & = & (m_1^S = C, Y_1, Y_2, Y_3) \\ \mathcal{S}_G^{1\,+} & = & (m_2^S = S, N_s, Ver_s, Y_3, CA(S, K_s)) \end{array}$$

In this role, the sent messages are S, N_s, Ver_s and Y_3. The messages S, N_s and Ver_s have the security level $\perp$, i.e $\ulcorner \alpha \urcorner = \perp$ for all $\alpha \in \{C, N_c, Ver_c, IdSession^i\}$. Hence, the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{1\,+}) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{C}_G^{1\,-})$ will be always true for all $\alpha \in \{C, N_c, Ver_c, IdSession^i\}$ and so the role $\mathcal{C}_G^1$ is increasing. Now, let's verify the equation for the message Y_3. In fact, the security level of Y_3 obtained by the function F_{TLS} according to sent and received messages in $\mathcal{S}_G^1$ is as follows:

α	m	$\mathsf{DIN}_{TLS}(\alpha, m)$	$\mathsf{DEK}_{TLS}(\alpha, m)$	$\mathsf{F}_{TLS}(\alpha, m)$
Y_3	$\mathcal{S}_G^{1-}$	$\{C\}$	$\perp$	$\perp$
Y_3	$\mathcal{S}_G^{1+}$	$\{S\}$	$\perp$	$\perp$

From the previous equations we can also deduce that the equation $\mathsf{F}_{TLS}(Y_3, \mathcal{C}_G^{2\,+}) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathsf{F}_{TLS}(Y_3, \mathcal{C}_G^{2\,-})$ and so the role $\mathcal{S}_G^1$ is increasing.

From the generalized role $\mathcal{S}_G^2$, we deduce that:

$$\begin{array}{rcl} \mathcal{S}_G^{2\,-} & = & (m_3^S = Y_3, \{Y_1, Y_4, C, S\}_{K_s}, \\ & & CA(C, K_c), \{H(F_1(m_1^S, m_2^S, Y_4, C, S))\}_{K_c^{-1}}) \\ \mathcal{S}_G^{2\,+} & = & m_4 = \{H(F_2(m_1^S, m_2^S, m_3^S, Y_4, C, S))\}_{K_{cs}} \end{array}$$

In the role $\mathcal{S}_G^2$, the sent messages are C, S, N_s, Ver_s, Y_1, Y_2, Y_3 and Y_4. The messages C, S, N_s and Ver_s have the security level $\perp$, i.e $\ulcorner \alpha \urcorner = \perp$ for all $\alpha \in \{C, S, Ver_c, IdSession^i\}$. Hence, the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{S}_G^{2\,+}) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{S}_G^{2\,-})$ will be always true for all $\alpha \in \{C, S, Ver_c, IdSession^i\}$. Now, let's verify the equation for the messages Y_1, Y_2, Y_3 and Y_4. Theirs security levels obtained by the function F_{TLS} according to sent and received messages in $\mathcal{S}_G^2$ are as follows:

α	m	$\mathsf{DIN}_{TLS}(\alpha, m)$	$\mathsf{DEK}_{TLS}(\alpha, m)$	$\mathsf{F}_{TLS}(\alpha, m)$
Y_1	$\mathcal{S}_G^{2-}$	$\{C, S\}$	$\perp$	$\perp$
Y_1	$\mathcal{S}_G^{2+}$	$\{C, S\}$	$\top$	$\{C, S\}$
Y_2	$\mathcal{S}_G^{2-}$	$\{C\}$	$\perp$	$\perp$
Y_2	$\mathcal{S}_G^{2+}$	$\{C, S\}$	$\top$	$\{C, S\}$
Y_3	$\mathcal{S}_G^{2-}$	$\{C\}$	$\perp$	$\perp$
Y_3	$\mathcal{S}_G^{2+}$	$\{C, S\}$	$\top$	$\{C, S\}$
Y_4	$\mathcal{S}_G^{2-}$	$\{C, S\}$	$\{S\}$	$\{C, S\}$
Y_4	$\mathcal{S}_G^{2+}$	$\{C, S\}$	$\top$	$\{C, S\}$

From the previous Table we can also deduce that the equation $\mathsf{F}_{TLS}(\alpha, \mathcal{S}_G^{2\,+}) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{S}_G^{2\,-})$ is true for all $\alpha \in \{Y_1, Y_2, Y_3, Y_4\}$ and so the role $\mathcal{S}_G^{2\,+}$ is increasing.

To sum up, To sum up, the generalized roles of C are increasing since they satisfy the equation

$$(eq2) \quad \mathsf{F}_{TLS}(\alpha, \mathcal{S}_G^{i\,+}) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathsf{F}_{TLS}(\alpha, \mathcal{S}_G^{i\,-})$$

Indeed, we have:

α	r	$\ulcorner \alpha \urcorner_0$	$\mathsf{F}_{TLS}(\alpha, r^+)$	$\mathsf{F}_{TLS}(\alpha, r^-)$	$(eq2)$
Y_3	$\mathcal{S}_G^1$	$\top$	$\perp$	$\perp$	*Yes*
Y_3	$\mathcal{S}_G^2$	$\top$	$\{C, S\}$	$\perp$	*Yes*
Y_2	$\mathcal{S}_G^2$	$\top$	$\{C, S\}$	$\perp$	*Yes*
Y_1	$\mathcal{S}_G^2$	$\top$	$\{C, S\}$	$\perp$	*Yes*
Y_4	$\mathcal{S}_G^2$	$\top$	$\{C, S\}$	$\{C, S\}$	*Yes*

The previous table shows that the generalized role of S is increasing. Therefore, we can deduce that the SSL/TLS protocol respects the secrecy property in the context $\mathcal{C}_{TLS}$

6 CONCLUSIONS

This paper presents the analysis of the SSL/TLS handshake protocol by using the interpretation functions-based method. In fact, we proved that the SSL/TLS protocol is correct with respect to the secrecy property. This result is conducted by considering the famous Dolev and Yao intruder model. In our future works, we will extend this model with more algebraic properties of cryptographic primitives in order to analyze the secrecy properties in more and realistic intruder model. In fact, in (Paulson, 1997b), L. Paulson has proven that the Bull protocol preserves the secrecy by using an intruder model that does not take into account any algebraic property of cryptographic primitives. However, he proved that attacks are possible on this protocol if some algebraic properties of $\oplus$ or of exponentiation are considered in the intruder model.

Also, we gave in this paper, a new and practical safe interpretation functions (DEK and DINEK functions) that could be used to analyze all kind of keys-agreement protocols. Therefore, we want to investigate in our future works the analysis of others keys-agreement protocols such as Kereberos with some interesting algebraic properties. Also, we want to study and give more safe interpretation functions.

REFERENCES

Abadi, M. (1999). Secrecy by typing in security protocols. *Journal of the ACM*, 46(5):749–786.

Bellare, M. and Rogaway, P. (1993). Random oracles are practical: A paradigm for designing efficient protocols. pages 62–73. ACM Press.

Bugliesi, M., Focardi, R., and Maffei, M. (2004). Authenticity by tagging and typing. In *FMSE ’04: Proceedings of the 2004 ACM workshop on Formal methods in security engineering*, pages 1–12. ACM Press.

Carlsen, U. (1994). *Formal Specification and Analysis of Cryptographic Protocols*. PhD thesis, Université PARIS XI.

Clark, J. and Jacob, J. (1996). A survey of authentication protocol literature. Unpublished Article Available at.

Debbabi, M., Durgin, N., Mejri, M., and Mitchell, J. (2001). Security by typing. *Accpeted for publication in the International Journal on Software Tools for Technology Transfer (STTT), Springer Verlag*.

Delicata, R. and Schneider, S. (2005). Temporal rank functions for forward secrecy. In *CSFW ’05: Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW’05)*, pages 126–139, Washington, DC, USA. IEEE Computer Society.

Dierks, T. and Rescorla, E. (2008). Rfc 5246 - the transport layer security (tls) protocol version 1.2. Technical report, IETF.

Fabrega, F. J. T., Javier, F., Herzog, J. C., and Guttman, J. D. (1999). Strand spaces: Proving security protocols correct.

Gordon, A. D. and Jeffrey, A. (2004). Authenticity by Typing for Security Protocols. *Journal of Computer Security*, 11(4):451–519.

He, C., Sundararajan, M., Datta, A., Derek, A., and Mitchell, J. C. (2005). A modular correctness proof of ieee 802.11i and tls. In *In CCS 05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 2–15. ACM Press.

Hickman, K. E. B. (1994). The ssl protocol version 2.0.

Houmani, H. and Mejri, M. (2007). Secrecy by interpretation functions. *Journal of Knowledge-Based Systems*, 20(7):617–635.

Houmani, H. and Mejri, M. (2008a). Analysis of some famous cryptographic protocols using the interpretation-function-based method. *International Journal of Security and Its Applications (IJSIA)*, 2(4):99–116.

Houmani, H. and Mejri, M. (2008b). Ensuring the correctness of cryptographic protocols with respect to secrecy. In PRESS, I., editor, *International Conference on Security and Cryptography (Secrypt)*, Porto, Portugal.

Houmani, H. and Mejri, M. (2008c). Toward an automatic verification of secrecy without the perfect encryption assumption. *International Journal of Computers, North Atlantic University Union (NAUN)*, 2(2):183–192.

Jager, T., Kohlar, F., Schage, S., and Schwenk, J. (2011). A standard-model security analysis of tls. Cryptology ePrint Archive.

Kemmerer, R., Meadows, C., and Millen, J. (1994). Three Systems for Cryptographic Protocol Analysis. *Journal of Cryptology*, 7(2):79–130.

Liebl, A. (1993). Authentication in distributed systems: A bibliography. *Operating Systems Review*, 27(4):122–136.

Meadows, C. (1994). The NRL Protocol Analyzer: An Overview. *Journal of Logic Programming*.

Meadows, C. (2003). What makes a cryptographic protocol secure? In *Proceedings of ESOP 03*. Springer-Verlag.

Mitchell, J. C. (1998). Finite-state analysis of security protocols. In in Computer Science, L. N., editor, *Computer Aided Verification*, volume 1427, pages 71–76.

Mitchell, J. C., Shmatikov, V., and Stern, U. (1998). Finite-state analysis of SSL 3.0. In *Proceedings of the 7th USENIX Security Symposium (SECURITY-98)*, pages 201–216, Berkeley. Usenix Association.

Morrissey, P., Smart, N. P., and Warinschi, B. (2008). A modular security analysis of the tls handshake protocol. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, pages 55–73.

Oppliger, R. and Gajek, S. (2005). Effective protection against phishing and web spoofing. In *Proceedings* *of the 9th IFIP TC6 and TC11 Conference on Communications and Multimedia Security (CMS 2005), Springer-Verlag, LNCS 3677*, pages 32–41.

Oppliger, R., Hauser, R., and Basin, D. (2006). Ssl/tls session-aware user authenticationor how to effectively thwart the man-in-the-middle. *Computer Communications*, 29:2238–2246.

Paulson, L. C. (1997a). Inductive analysis of the internet protocol tls. *ACM Transactions on Information and System Security*, 2:332–351.

Paulson, L. C. (1997b). Mechanized proofs for a recursive authentication protocol. In *10th Computer Security Foundations Workshop*, pages 84–95. IEEE Computer Society Press.

Rubin, A. D. and Honeyman, P. (1993). Formal methods for the analysis of authentication protocols. Technical Report 93–7, Center for Information Technology Integration. University of Michigan. Internal Draft.

Sabelfeld, A. and Myers, A. (2003). Language-based information-flow security.

Schneider, S. (1992). An operational semantics for timed CSP. In *Proceedings Chalmers Workshop on Concurrency, 1991*, pages 428–456. Report PMG-R63, Chalmers University of Technology and University of Göteborg.

Schneider, S. (1996). Security properties and csp. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 174–187. IEEE Computer Society Press.

Schneider, S. (1997). Verifying authentication protocols with CSP. In *PCSFW: Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press.

Syverson, P. (1991). The use of logic in the analysis of cryptographic protocols. In Lunt, T. F. and McLean, J., editors, *Proceedings of the 1991 IEEE Symposium on Security and Privacy*, pages 156–170. IEEE Computer Society.

Syverson, P. (92). Knowledge, belief, and semantics in the analysis of cryptographic protocols. *Journal of Computer Security*, 1(3):317–334.

Wagner, D. and Schneier, B. (1996). Analysis of the ssl 3.0 protocol. In *Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce - Volume 2*, pages 4–4, Berkeley, CA, USA. USENIX Association.