Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 254942, 18 pages
doi:10.1155/2012/254942

Research Article

Formal Analysis of SET and NSL Protocols Using the Interpretation Functions-Based Method

Hanane Houmani[1] and Mohamed Mejri[2]

[1] *EAS Group, ENSEM, Hassan II University, Casablanca, Morocco*
[2] *LSFM Group, Laval University, Quebec, QC, Canada*

Correspondence should be addressed to Hanane Houmani, hanane.houmani@ift.ulaval.ca

Received 21 April 2012; Revised 18 June 2012; Accepted 27 June 2012

Academic Editor: Chi-Yao Weng

Most applications in the Internet such as e-banking and e-commerce use the SET and the NSL protocols to protect the communication channel between the client and the server. Then, it is crucial to ensure that these protocols respect some security properties such as confidentiality, authentication, and integrity. In this paper, we analyze the SET and the NSL protocols with respect to the confidentiality (secrecy) property. To perform this analysis, we use the interpretation functions-based method. The main idea behind the interpretation functions-based technique is to give sufficient conditions that allow to guarantee that a cryptographic protocol respects the secrecy property. The flexibility of the proposed conditions allows the verification of daily-life protocols such as SET and NSL. Also, this method could be used under different assumptions such as a variety of intruder abilities including algebraic properties of cryptographic primitives. The NSL protocol, for instance, is analyzed with and without the homomorphism property. We show also, using the SET protocol, the usefulness of this approach to correct weaknesses and problems discovered during the analysis.

1. Motivations and Background

Intuitively, cryptographic protocols are communication protocols that involve cryptography to reach some specific security goals (authentication, secrecy, etc.). Today, these protocols are playing a key role in our daily life. Among others, they protect our banking transactions (e-commerce protocol), our access to private wired and wireless network, and our access to a variety of indispensable services (web, FTP, e-mail, etc.).

Obviously, any flaws in such protocols can have heavy negative consequences on individuals and organizations. It is also a well-known fact that attacks exploiting cryptographic protocols flaws are generally very difficult to detect: tools such as intrusion detections and firewalls are helpless against them since it is difficult (even impossible some-times) to distinguish between legitimate and illegitimate users when the cryptographic protocol is flawed.

Like any sensitive system, cryptographic protocols need to be seriously studied and their correctness should be rigorously analyzed and ideally proved. For that reason, formal specification and verification of security protocols have received much attention in recent years. Some of these works including comparative studies could be found in [1–10].

Today and after more of thirty years of hard work, international community has a better understanding of cryptographic protocols and better support to specify and analyze them. But there remains a lot of work to do in this field. Some of the most important drawbacks of the existing results is that they are limited by either the class of protocols that they can analyze or the context (e.g., limited intruder abilities) in which they can analyze them. For instance, the method described in [11] gives an algorithm that allows to analyze the *pingpong* protocols in a polynomial time but only within the encryption of atomic and closed messages. Others like [12–23] are dedicated to a specific intruder capacity which is usually the standard Dolev and Yao model. These two parameters (the class of the analyzed protocols and the context in which they are analyzed) are generally hard-coded

inside the approach making its adaptation to another context or others protocols very difficult. Also, in [24], Paulson has proven that the Bull protocol preserves the secrecy by using an intruder model that does not consider 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. Thereby, the flexibility of an approach to be adapted to different contexts of analysis is helpful to make the conditions of the analysis closer to the reality.

To deal with this problem, we have introduced in [25, 26] a general and flexible approach that allows to analyze a large variety of protocols in different contexts (different intruder capacities including equational theories) based on interpretation functions. The idea is to give some metaresults (independent of a specific context) that give sufficient conditions allowing to guarantee the correctness of a protocol with respect of the secrecy (confidentiality) property.

In this paper, we recall the main results of the approach and we show its efficiency and its flexibility by analyzing some famous cryptographic protocols such as SET protocol [27–29] and NSL protocol [30] under different assumptions. In fact, we first analyze the secrecy property of the NSL protocol in the Dolev and Yao intruder model and after that we consider the homomorphism property of the encryption. This analysis allows us to prove that the NSL protocol is secure in the Dolev and Yao model whereas it is not when considering the homomorphism encryption property and some weaknesses are found. Second, we analyze the secrecy property of SET protocol which could be considered the first analysis until now that concern this protocol and the secrecy property which is very important in such protocols that allow to transmit secret information such as the card credit number.

This paper is organized as follows. Section 2 gives an overview of the interpretation functions-based method. The analysis of the NSL protocol in different contexts is given in Section 3. Section 4 analyzes the SET protocol. Section 5 compares the interpretation functions-based method with some similar works like the rank functions-based method [31, 32] and the typing-based method [33]. Finally, some concluding remarks are given in Section 6.

2. Overview of the Interpretation Functions-Based Method

As stated before, the main idea of our approach is to propose some conditions that are proven (in [25, 26]) sufficient to guarantee the secrecy property of any protocol that respects them. The proposed conditions can be easily verified in PTIME, and they state intuitively 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 the context of verification (input information) or estimated from received messages. The security levels estimated from received messages is calculated by using some special function called safe interpretation functions.

Table 1: Example of increasing protocol.

1. $A \rightarrow B : \{\{A, B, secret\}_{k_b}\}_{k_a^{-1}}$
2. $B \rightarrow A : \{\{A, B, secret\}_{k_a}\}_{k_b^{-1}}$

Table 2: Exemple of deacreasing protocol.

1. $A \rightarrow B : \{\{A, B, secret\}_{k_b}\}_{k_a^{-1}}$
2. $B \rightarrow A : \{A, B, secret\}_{k_b^{-1}}$

A brief description about these notions could be found in the next sections. In fact, Section 2.1 gives the intuitive definition and some examples of increasing protocols. Section 2.2 presents the notion of safe interpretation functions. After that Section 2.3 presents the definition of the secrecy property (confidentiality). Finally, Section 2.4 states the main result and show how the interpretation functions-based method could be applied to analyze the secrecy property.

2.1. Increasing Protocols. The sufficient condition to guarantee the secrecy property could be summarized as follows. Principals involved in the protocol should not decrease the security levels of sent components. Protocols that satisfy this condition are called in this work "increasing protocols." For instance, the protocol described by Table 1 is increasing since the security level of the message *secret* in step 1 is the same in step 2. Indeed, in step 1 the message *secret* is encrypted by the secret key of A to say that the message is originated from A and also is encrypted by the public key of B to say that the message *secret* is destined to B, hence the security level of the message *secret* is the security level that allows only B and A to know it. In the same way, we can notice that the security level of the message *secret* in step 2 is the security level that allows B and A to know it and hence, the protocol does not decrease the security level of message *secret* form step to another and so the protocol is increasing.

Let us consider the protocol described by Table 2. This protocol is not increasing since the security level of the message *secret* in step 2 is lower than its security level in step 1. Indeed, in step 1 the message *secret* is encrypted by the secret key of A to say that the message is originated from A and also is encrypted by the public key of B to say that the message *secret* is destined to B, hence the security level of the message *secret* is the security level that allows only B and A to know it. In step 2, the message *secret* is encrypted by the secret key of B, so everybody could decrypt it and hence the security level of the message decreases from step 1 to step 2. The formal definition of decreasing protocol could be found in our previous works [25, 26].

2.2. Safe Interpretation Function. To verify whether a protocol is increasing, we need a safe means to correctly estimate the security levels of received components so that we can appropriately handle them. We called it "safe interpretation function" (The name of the approach (interpretation functions-based approach) come for this notions.). Among

the important features of a safe interpretation function is that its results could not be misled by the intruder manipulations. For example, if the interpretation function estimates that the security level of a component α in a set of messages M is top secret, then the intruder can never produce M' from M such that the security level of α in M' is estimated by secret or public. A simple example of a safe interpretation function is the one that attributes a security level for a component α in a message m depending only on the direct keys encrypting α in m. This function, denoted by F_{DEK}, will be referred later by DEK (Direct Encryption Key). Accordingly, $F_{\mathrm{DEK}}(N_b, \{S, N_b\}_{k_{ab}})$ 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 $\{A, B\}$ then we have

$$F_{\mathrm{DEK}}\left(N_b, \{S, N_b\}_{k_{ab}}\right) = \{A, B\}. \tag{1}$$

Another example of safe interpretation function, used in this paper, is the one that attributes a security level of a component α in a message m depending on both the direct keys encrypting α in m and the neighbors of α in m (the components that can be reached from α without going outside encryptions: usually we consider neighbors that are only identities of agents). This function, denoted by F_{DEKAN}, will be referred later by the DEKAN (direct encryption key and neighbors) function. Accordingly, $F_{\mathrm{DEKAN}}(N_b, \{S, N_b\}_{k_{ab}})$ 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 $\{A, B\}$ to design that is a shared secret between A and B, then we have

$$F_{\mathrm{DEKAN}}\left(N_b, \{S, N_b\}_{k_{ab}}\right) = \{A, B, S\}. \tag{2}$$

The DEK function and the DEKAN function can be used to analyze a large variety of cryptographic protocols. When the analysis fails, we should either adapt the protocol or use another safe function. However, the definition of safe functions is a complicated task. For this reason, we have introduced in [25, 26, 34] a helpful guideline allowing to define a safe interpretation function having the following form:

$$\mathsf{F}(\alpha, M) = \mathsf{I} \circ \mathsf{S}(\alpha, M). \tag{3}$$

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*. This type of rank functions has been already introduced and used by Schneider et al. in [31, 32, 35] to attribute to a message a security level (rank).

A simple way to define a selection function is to consider a term (a message in our case) as a tree where its arcs are annotated with real numbers that reflect costs or distances between nodes. After that, it will be easy to define S as a function that selects components that are at some distance from a given component. For instance, let m be the message $\langle B, \{N_b, k_{ab}\}_{k_{as}}\rangle$ then the representation of m can be described as shown by Figure 1, where the annotations of arcs are explained later.

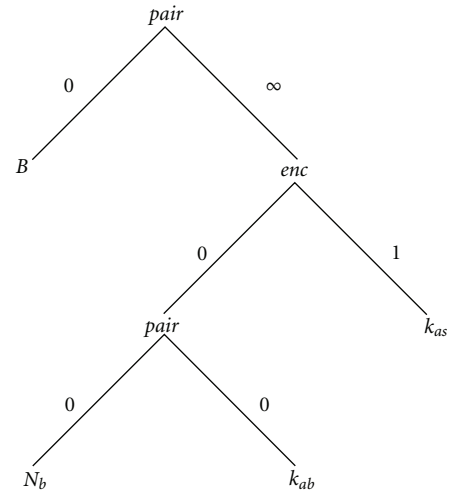

Figure 1: The message ($\langle B, \{N_b, k_{ab}\}_{k_{as}}\rangle$).

Recall that the symbol *enc* is the encryption operator and the symbol *pair* is the concatenation operator. Hence, at distance 1, we could select the keys of encryption, and at distance 0 we could select components that are concatenated to some specific components.

Of course, not all the interpretation functions F which are the composition of the selection function S and a rank function I are safe (could not be misled by an intruder). In the sequel, we give some conditions that must be satisfied by the functions S and I so that their composition is safe. Intuitively, the conditions on the selection function S states that at least the direct encryption keys are selected and no components outside direct encryptions could be selected. The idea behind this condition is that only messages encrypted with a secret key could be safe from the intruder manipulations. 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 $\{A, N_a, B, C\}$. To exclude components outside encryption, we attribute ∞ as a cost between any *enc* node and its father.

In the other side, the conditions on the rank function I state intuitively that the security level of a message, given by this function, should not be greater than its real security level. The idea behind this condition is that the rank function could not give any security level to messages. For instance, if β is a public information (according to the function $\ulcorner \cdot \urcorner$ given within the context of analysis), then I cannot interpret it as secret because elsewhere β could encrypt a secret information whereas β is a public information and so such interpretation functions do give a safe means to estimate the secrurity level. An easy way to construct the rank function is to take it equal to the function $\ulcorner \cdot \urcorner$.

The formal definition of these functions and some sufficient conditions to construct some safe interpretation functions can be found in [25, 26, 34].

As stated in Section 1, there is a large variety of formal methods dedicated to the analysis of cryptographic protocols

and particularly with respect to the secrecy property. One of the major differences between them is the assumptions or the restrictions required by each method from the analyzed protocol. They can analyze protocols only under some particular conditions. One of the major drawbacks of these approaches is that the proofs of the their main results are strongly connected to their assumptions making their adaptation to others assumptions a very tedious task.

To overcome this drawback, we introduced in [25, 26] a more general approach that considers the context of verification (some assumptions on protocols, intruder model, etc.) as a parameter of the approach so that it will be clear how it affects any intermediary result. Intuitively, a context of verification contains all the parameters that affect the verification of cryptographic protocols. Among others, a context of verification, denoted by $\mathcal{C}$ in this paper, contains the sorts of messages involved in analyzed protocols, the rules that capturing the intruder capacities, and the algebraic properties of the cryptographic primitives. As we will see later, all the results of our approach consider the context of the verification as a simple input parameter. This great flexibility is useful to change the class of protocols or the intruder capacities and use the approach without any need of reworking the proofs or the conditions. For instance, we can use the approach to analyze protocols that use either symmetric or asymmetric keys. Also, we can employ the approach with or without algebraic properties of cryptographic primitives as it will be shown later within examples.

2.3. Secrecy Property. In this work, the secrecy property is defined in term of information flow security. We adopt also the "no read-up" notion of Bell-La Padula [36] model stating that a subject at a given security level is not allowed to know an information having a higher security level. We suppose, without effective restrictions (notice that it is always possible to define a security lattice that reflects our needs and which is coherent with this hypothesis), that each principal has his security level captured by the highest security level of components in his initial knowledge. In other words, if an agent (including the intruder) knows a message with a security level τ, then he is also eligible to know all messages having a security level lower or equal than τ. Intuitively, we say that a protocol respects the secrecy property if the intruder cannot learn from any valid trace more than what he is eligible to know. For example, if K is the set of the intruder's initial knowledge where α has a security level denoted by $\ulcorner\alpha\urcorner$, then the intruder is illegible to know an information β only if its security level $\ulcorner\beta\urcorner$ is lower or equal to $\ulcorner\alpha\urcorner$.

2.4. Main Result. Now, recall the sufficient conditions allowing to guaranty the correctness of a protocol with respect to the secrecy property. Informally, these conditions state that honest agents should never decrease, according to a safe interpretation function, the security level of any atomic message sent over the network. Protocols that satisfy this function are called increasing protocols. The formalization of an increasing protocol is as follows. 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 use an unbounded number of operations to the messages that he manipulates. Indeed, we proved that, to check if a protocol respects the secrecy property, it is sufficient to verify whether a finite model of the protocol, called in this work a "roles-based specification," is increasing.

The verification process of the interpretation functions-based method can be summarized as described by Figure 2.

Intuitively, 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 about its correctness. Generally, if the correctness of a protocol cannot be ensured using a given safe interpretation function, it does not mean that a positive result cannot be involved using another one. However, even if the verification is not conclusive, it provides helpful information that can be used either 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 as it will be illustrated later. Also, this verification is finite since it is conducted on a finite set of generalized roles.

We believe that the sufficient conditions are not very restrictive, that is, for most of secure protocols we can construct a safe interpretation function allowing to prove the secrecy property. As shown later, even when the verification is not conclusive, the effort made to verify if the protocol is increasing could be helpful to discover flaws or weaknesses in the analyzed protocol or sometimes to have an idea about another safe interpretation function allowing to prove that it is increasing one. It is interesting to notice also that, some times, a slight modification on a protocol could make it an increasing one for a given safe interpretation function and allow to conclude that it is correct for secrecy.

2.5. Protocol. This section gives the syntax and semantics of a protocol and how to infer the roles-based specification from the standard description of a given protocol.

2.6. Syntax. Essentially, a protocol is specified by a sequence of communication steps given in the standard notation. Each step has an unique identifier and specifies the sender, the receiver, and the transmitted message. More precisely, a protocol p has to respect the following BNF grammar:

$$p ::= \langle i : A \longrightarrow B : m \rangle \mid p.p. \tag{4}$$

The statement $\langle i : A \rightarrow B : m \rangle$ denotes the transmission of a message m from the principal A to the principal B in the step i of the protocol.

Table 3 gives an example of a protocol inspired from the Woo and Lam's one [37] and aims to distribute a fresh key that will be shared between two principals A and B.

2.6.1. Roles-Based Specification. To give a semantics to a protocol, we use the notion of generalized roles introduced in [38]. Intuitively, a roles-based specification is a set

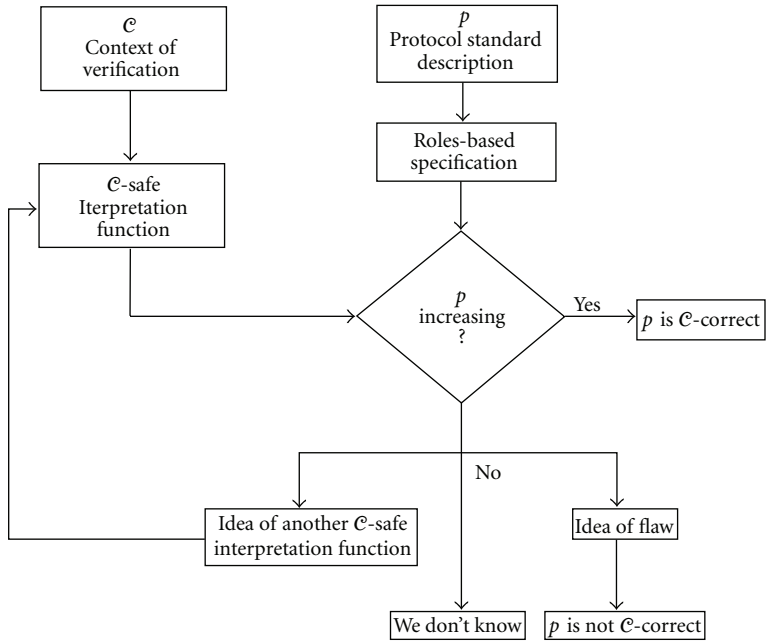

Figure 2: Protocol verification process.

containing the prefixes of generalized roles. A generalized role is a protocol abstraction, where the emphasis is put on a particular principal and where all the unknown messages are replaced by variables. Also, an exponent i (the session identifier) is added to each fresh message to emphasize that these components change their values from one run to another. Intuitively, a generalized role reflects how a particular agent perceives the exchanged messages. To a given principal, different generalized roles can be attributed. These generalized roles are not necessarily prefixes of each others and show the different messages that can be accepted by a role to successfully reach a given step in the protocol. More details related to the generalized roles are in [39]. The roles-based specification (the set of generalized roles) of the Woo and Lam protocol is as follow.

(i) The generalized roles of A are $\mathcal{A}_G^1$ and $\mathcal{A}_G^2$ and are as follows:

$$\begin{aligned}
\mathcal{A}_G^1 &= \langle i.1, A \longrightarrow I(B) : A\rangle,\\
\mathcal{A}_G^2 &= \langle i.1, A \longrightarrow I(B) : A\rangle\\
&\quad \langle i.2, I(B) \longrightarrow A : X\rangle\\
&\quad \left\langle i.3, A \longrightarrow I(B) : \left\{X, k_{ab}^i\right\}_{k_{as}}\right\rangle.
\end{aligned} \tag{5}$$

If a given role cannot make any verification on a part of the received message, then this part needs to be substituted by a variable. The role of A receives a nonce at step 2 but he cannot make any validation on it (we suppose that there are no different types for different messages). For that reason N_b has been substituted by X.

(ii) The generalized roles of B are $\mathcal{B}_G^1$ and $\mathcal{B}_G^2$ and are as follows:

$$\begin{aligned}
\mathcal{B}_G^1 &= \langle i.1, I(A) \longrightarrow B : A\rangle\\
&\quad \langle i.2, B \longrightarrow I(A) : N_b\rangle,\\
\mathcal{B}_G^2 &= \langle i.1, I(A) \longrightarrow B : A\rangle\\
&\quad \langle i.2, B \longrightarrow I(A) : N_b\rangle\\
&\quad \langle i.3, I(A) \longrightarrow B : Y\rangle\\
&\quad \langle i.4, B \longrightarrow I(S) : \{A, Y\}_{k_{bs}}\rangle.
\end{aligned} \tag{6}$$

The variable Y appears since the principal B receives at the third step the unknown message $\{N_b, k_{ab}\}_{k_{as}}$ and he does not have the key k_{as} to make any verification inside it. For a similar reason, the variable Z appears in the fifth step (the key k_{ab} is initially unknown by B and he could not make any verification on it).

(iii) The generalized role of S is

$$\begin{aligned}
\mathcal{S}_G &= \left\langle i.4, I(B) \longrightarrow S : \{A, \{U, V\}_{k_{as}}\}_{k_{bs}}\right\rangle\\
&\quad \langle i.5, S \longrightarrow I(B) : \{U, V\}_{k_{bs}}\rangle.
\end{aligned} \tag{7}$$

Since S does not initially know the value of N_b and k_{ab}, then these messages are, respectively, replaced by the fresh variables U and V.

In the rest of this paper, we denote by $\mathcal{R}_G(p)$ the prefix closed set of generalized roles of the protocol p and if A is an agent, then $\mathcal{R}_G(A)$ denotes the set of its generalized roles.

Table 3: Woo and Lam modified protocol.

$p_{\text{WL}} =$	$\langle 1, A \rightarrow B : A\rangle$
	$\langle 2, B \rightarrow A : N_b\rangle$
	$\langle 3, A \rightarrow B : \{N_b, k_{ab}\}_{k_{as}}\rangle$
	$\langle 4, B \rightarrow S : \{A, \{N_b, k_{ab}\}_{k_{as}}\}_{k_{bs}}\rangle$
	$\langle 5, S \rightarrow B : \{N_b, k_{ab}\}_{K_{bs}}\rangle$

2.6.2. Valid Traces. Based on generalized roles, we define the notion of a *valid trace* of a protocol (an acceptable execution of a protocol). A trace is considered valid if it respects the following tow conditions.

(i) Any session in the trace is an instance (a substitution) of a prefix of a generalized role.

(ii) Any message sent by the intruder should be deductible from his previous received messages, his initial knowledge, and his inference rules.

For instance, the trace of Table 4 is valid for the Woo and Lam protocol described by Table 3. This valid trace contains a breach of secrecy since the intruder can obtain the secret k_{ab}^{α} that should be shared only between A and B.

2.6.3. Semantics. The semantics of a protocol p executed in a context $\mathcal{C}$ is given by the following definition.

Definition 1 (semantics). Let $\mathcal{C}$ be a context of verification and p a protocol. The semantics of p in $\mathcal{C}$, denoted by $[p]_{\mathcal{C}}$, is the set of valid traces of p when executed in $\mathcal{C}$.

3. Analysis of NSL Protocol

This section shows the efficiency and the flexibility of the interpretation functions-based approach when analyzing cryptographic protocols even when taking into consideration cryptographic primitives. We first analyze the NSL-protocol [30] without considering the homomorphism property and we show that it respects the secrecy property. Secondly, we show that if we consider a homomorphism encryption during the analysis, then some weaknesses could appear.

3.1. Analysis of NSL Protocol without the Homomorphism Property. In this section, we consider the NSL-protocol with the standard intruder model of Dolev and Yao (i.e., without algebraic properties).

3.1.1. The NSL-Protocol. The Needham-Schroeder Lowe (NSL) protocol, denoted by p_{NSL} and given in Table 5, is a mutual authentication protocol that uses asymmetric encryption.

The first step of the analysis consists in fixing the context of verification which contains the message algebra, the intruder capabilities, and the security levels of all messages. Second, we define a safe interpretation function

Table 4: Valid trace for the Woo and Lam modified protocol.

$\alpha.1.\ A \rightarrow I(B) : A$
$\alpha.2.\ I(B) \rightarrow A : N_i^{\alpha}$
$\alpha.3.\ A \rightarrow I(B) : \{N_i^{\alpha}, k_{ab}^{\alpha}\}_{k_{as}}$
$\beta.4.\ I(B) \rightarrow S : \{A, \{N_i^{\alpha}, k_{ab}^{\alpha}\}_{k_{as}}\}_{k_{is}}$
$\beta.5.\ S \rightarrow I(B) : \{N_i^{\alpha}, k_{ab}^{\alpha}\}_{k_{is}}$

Table 5: Needham schroeder lowe protocol.

$1.\ A \rightarrow B : \{A, N_a\}_{k_b}$
$2.\ B \rightarrow A : \{\langle N_a, N_b\rangle, B\}_{k_a}$
$3.\ A \rightarrow B : \{N_b\}_{k_b}$

Table 6

$\mathcal{N}_{\text{NSL}}$	::=	A	(Principal Identifier)
	|	N_a	(Nounce)
	|	k_a^{-1}	(Private key)
	|	k_a	(Public key)

in this context. After that, the role-based specification can be analyzed to verify whether the protocol is increasing by using the defined safe interpretation function. The following paragraphs describe in detail these steps.

3.1.2. NSL Context of Verification in the Dolev-Yao Model. Let $\mathcal{C}_{\text{NSL}} = \langle \mathcal{M}_{\text{NSL}}, \vDash_{\text{NSL}} \mathcal{K}_{\text{NSL}}, \mathcal{L}_{\text{NSL}}^{\sqsupseteq}, \ulcorner\cdot\urcorner_{\text{NSL}}\rangle$ be a context of verification where $\mathcal{M}_{\text{NSL}}$ is $(\mathcal{N}_{\text{NSL}}, \Sigma_{\text{NSL}}, \mathcal{E}_{\text{NSL}})$ such that we have the following.

(i) $\mathcal{N}_{\text{NSL}}$ is the set of atomic messages given by BNF grammar shown in Table 6.

(ii) $\Sigma_{\text{NSL}} = \{pair, fst, snd, enc, dec\}$ contains the set of message operators.

Hence, the set of messages of $\mathcal{M}_{\text{NSL}}$ is defined by BNF rules shown in Table 7.

(iii) $\mathcal{E}_{\text{NSL}}$ is an equational theory containing the following equations:

$$\begin{aligned} fst(pair(x, y)) &= x, \\ snd(pair(x, y)) &= y, \\ dec(enc(x, k), k^{-1}) &= x, \\ enc(dec(x, k_a^{-1}), k_a) &= x, \\ \langle\langle m_1, m_2\rangle, m_3\rangle &= \langle m_1, \langle m_2, m_3\rangle\rangle. \end{aligned} \tag{8}$$

As usual, we can write $\{m\}_k$ instead of $\text{enc}(m, k)$. Also, we can write m_1, m_2 or $\langle m_1, m_2\rangle$ instead of $pair(m_1, m_2)$ and $\text{sign}(m, k^{-1})$ instead of $\text{dec}(m, k^{-1})$ if k^{-1} is a private key.

Table 7

$m,\ m_1,\ m_2$	::=	$\mathcal{N}_{\text{NSL}}$	
	|	$pair(m_1, m_2)$	(Pair Function)
	|	$enc(m, k_a)$	(Encryption Function)
	|	$dec(m, k_a^{-1})$	(Decryption Function)
	|	$fst(pair(m_1, m_2))$	(First Function)
	|	$snd(pair(m_1, m_2))$	(Second Function)

Given a set of equations $\mathcal{E}_{\text{NSL}}$ and a set of signatures Σ_{NSL}, the intruder capacities, denoted by the relation $\vDash_{\text{NSL}}$, are defined by the following rules:

$$\begin{aligned}
&\text{(knowledge)}\ \frac{\square}{M \vDash_{\text{NSL}} m} \quad [m \in M],\\
&\text{(construction)}\ \frac{M \vDash_{\text{NSL}} m_1 \cdots M \vDash_{\text{NSL}} m_n}{M \vDash_{\text{NSL}} f(m_1, \ldots, m_n)} \quad [f \in \Sigma_{\text{NSL}}],\\
&(\mathcal{E}\text{-equality})\ \frac{M \vDash_{\text{NSL}} m}{M \vDash_{\text{NSL}} m'} \quad [m =_{\mathcal{E}_{\text{NSL}}} m'].
\end{aligned} \tag{9}$$

The initial knowledge of principals $\mathcal{K}_{\text{NSL}}$ can be as follows: each principal knows his identity, the identities 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 $\mathcal{L}_{\text{NSL}}$ is $\mathcal{L}_0 = (2^{\mathcal{A}}, \subseteq)$. The security level of a message is 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 $\mathcal{A}$.

The typing function $\ulcorner \cdot \urcorner_{\text{NSL}}$ could be any partial function from $\mathcal{M}_{\text{NSL}}$ to $\mathcal{L}_{\text{NSL}}$ that reflects the security levels of components exchanged during the protocol. In this example, we choose this function as follows:

$$\ulcorner . \urcorner = [N_a \longmapsto \{A, B\}, N_b \longmapsto \{A, B\}, k_a^{-1} \longmapsto \{A\}, k_a \longmapsto \bot], \tag{10}$$

where A and B could be substituted by any principal identifier.

3.1.3. NSL Safe Interpretation Function. We use the DEKAN function (that selects the direct encryption keys and neighbors) to analyze the NSL-protocol under the equational theory $\mathcal{E}_{\text{NSL}}$. More precisely, the DEKAN function is a composition of a selection function and an interpretation function, that is $\mathsf{F}_{\text{NSL}} = \mathsf{I}_{\text{NSL}} \circ \mathsf{S}_{\text{NSL}}$

(i) The selection function S_{NSL} takes an atomic component α and a message m and returns keys that directly encrypt α in m with principal identities and variables that are neighbors of α in m. The notion of direct encrypting keys and neighbors are formalized using a special labeled tree representing the analyzed message m. More precisely, given a message m, we start by annotating the arcs of its corresponding tree according to schema given by Figure 3, where d is defined as follows:

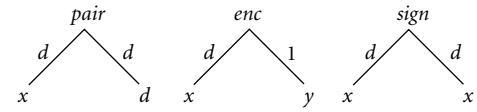

Figure 3: Safe costs assignment.

$$d = \begin{cases} \infty & \text{if } x = \text{enc} \\ 0 & \text{else.} \end{cases} \tag{11}$$

This means that the distance between a node and his father is 0 if it is not *enc* and its father is either *pair* or *sign*. If the node is *enc*, then the distance that separates it from its father is infinite. Finally, if the node is the encryption key of the function *enc*, then the distance to its father is 1. Using these distances between nodes, we consider two nodes as neighbors if the distance that separates them is equal to 0. Also, we consider a key as a direct encrypting one for α if the distance between them is equal to 1. Using these notions, $\mathsf{S}_{\text{NSL}}(\alpha, m)$ returns principal identifiers and variables in m having a distance 0 from α (neighbors) with atomic components having a distance 1 from α (direct encrypting keys). For example,

$$\mathsf{S}_{\text{NSL}}\left(\alpha, \left\{\beta, A, \left\{B, \alpha, \{\alpha, N_a, X\}_{k_3}\right\}_{k_2}\right\}_{k_1}\right) = \{B, X, k_2, k_3\}. \tag{12}$$

(ii) The interpretation function I_{NSL} interprets elements returned by S_{NSL} as follows:

$$\begin{aligned}
&\mathsf{I}_{\text{NSL}}(\emptyset) = \top,\\
&\mathsf{I}_{\text{NSL}}(S_1 \cup S_2) = \mathsf{I}_{\text{NSL}}(S_1) \sqcup \mathsf{I}_{\text{NSL}}(S_2),\\
&\mathsf{I}_{\text{NSL}}(\{\beta\}) = \begin{cases} \{\beta\} & \text{if } \beta \text{ is a principal identifier,} \\ \ulcorner \beta^{-1} \urcorner & \text{if } \beta \text{ is a key,} \\ \tau_\beta & \text{if } \beta \text{ is a variable.} \end{cases}
\end{aligned} \tag{13}$$

For example, if $\ulcorner k_2^{-1} \urcorner = \{A\}$ and $\ulcorner k_3^{-1} \urcorner = \{S\}$ then,

$$\mathsf{I}_{\text{NSL}}(\{B, X, k_2, k_3\}) = \tau_X \sqcup \{A, B, S\}. \tag{14}$$

Notice also that, when an equational theory is taken into consideration, a message can have different equivalent forms. In this situation, to be safe, an interpretation function should give to a message the lowest level of the messages of its class. More precisely, if $m_{/\mathcal{E}} = \{m' \in \mathcal{M} \mid m =_{\mathcal{E}} m'\}$, then, for any atomic message α, we have

$$\mathsf{F}(\alpha, m) = \prod_{m' \in m_{/\mathcal{E}}} \mathsf{F}(\alpha, m'). \tag{15}$$

Table 8: Analysis of NSL: role of A.

α	r	$\ulcorner\alpha\urcorner$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$
N_a^i	$\mathcal{A}_G^1$	$\{A, B\}$	$\{B\}$	$\top$	*Yes*
X	$\mathcal{A}_G^2$	τ_X	$\{B\}$	$\{A, B\}$	*Yes* (for any $\tau_X \in 2^{\mathcal{A}}$)

Table 9: Analysis of NSL protocol: role of B.

α	r	$\ulcorner\alpha\urcorner$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$
N_b^i	$\mathcal{B}_G^1$	$\{A, B\}$	$\{A\}$	$\top$	*Yes*
Y	$\mathcal{B}_G^1$	τ_Y	$\{A, B\}$	$\{A, B\}$	*Yes* (for any $\tau_Y \in 2^{\mathcal{A}}$)

To simplify the computation of F, we usually try to transform the equational theory to a convergent inference system so that the canonical form of a message m, denoted by $m_{\downarrow}$, has always the lowest security level. In other words,

$$\mathsf{F}(\alpha, m_{\downarrow}) \sqsubseteq \prod_{m' \in m/\mathcal{E}} \mathsf{F}(\alpha, m'). \tag{16}$$

To reach this goal, we orient the equation so that the right side is greater to the left side according to F and the obtained rewriting system is convergent (the convergence can be proved based on another ordering relation that is independent of F). (Notice that if such orientation is not possible, the approach cannot be applied. However, for all the equational theories that we have already considered till now we didn't find difficulties to orient them.) For our example, it is simple to see that the following rewriting system is convergent and gives the lowest security level:

$$\begin{aligned} fst(\langle x, y\rangle) &\longrightarrow x, \\ snd(\langle x, y\rangle) &\longrightarrow y, \\ dec(enc(x, k), k^{-1}) &\longrightarrow x, \\ enc\left(dec\left(x, k_y^{-1}\right), k_y\right) &\longrightarrow x, \\ \langle\langle m_1, m_2\rangle, m_3\rangle &\longrightarrow \langle m_1, \langle m_2, m_3\rangle\rangle. \end{aligned} \tag{17}$$

3.1.4. NSL Role-Based Specification. The NSL role-based specification is

$$\mathcal{R}_G(p_{\mathrm{NSL}}) = \{\mathcal{A}_G^1, \mathcal{A}_G^2, \mathcal{B}_G^1\}, \tag{18}$$

where the generalized roles $\mathcal{A}_G^1$ and $\mathcal{A}_G^2$ are as follows:

$$\begin{aligned} \mathcal{A}_G^1 = {} & i.1.\ A \longrightarrow I(B) : \left\{A, N_a^i\right\}_{k_b} \\ & i.1.\ A \longrightarrow I(B) : \left\{A, N_a^i\right\}_{k_b}, \\ \mathcal{A}_G^2 = {} & i.2.\ I(B) \longrightarrow A : \left\{\left\langle N_a^i, X\right\rangle, B\right\}_{k_a} \\ & i.3.\ A \longrightarrow I(B) : \{X\}_{k_b}, \end{aligned} \tag{19}$$

and the generalized role $\mathcal{B}_G^1$ is as follows:

$$\begin{aligned} \mathcal{B}_G^1 = {} & i.1.\ I(A) \longrightarrow B : \{A, Y\}_{k_b} \\ & i.2.\ B \longrightarrow I(A) : \left\{\left\langle Y, N_b^i\right\rangle, B\right\}_{k_a}. \end{aligned} \tag{20}$$

3.1.5. Verifying Whether the NSL Role-Based Specification Is Increasing. Now, we can verify whether the role-based specification of SET is $\mathsf{F}_{\mathrm{SET}}$-increasing. More precisely, we need accordingly to verify that principals do not decrease the security level of components from step to another. To that end, we verify this condition on each generalized role of the protocol (i.e., on each point of view of protocol participants). To that end, we should verify whether the security level of sent messages is a subset of the security level of these messages in the received steps, that is,

$$\forall r \quad \mathsf{F}_{\mathrm{SET}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{SET}}(\alpha, r^-) \quad \forall \alpha. \tag{21}$$

From the generalized roles of A, we deduce that

$$\begin{aligned} \mathcal{A}_G^{1^-} &= \emptyset, \qquad \mathcal{A}_G^{1^+} = \left\{A, N_a^i\right\}_{k_b}, \\ \mathcal{A}_G^{2^-} &= \left\{\left\langle N_a^i, X\right\rangle, B\right\}_{k_a}, \qquad \mathcal{A}_G^{2^+} = \{X\}_{k_b}. \end{aligned} \tag{22}$$

Recall that we are working with the lattice of security $(2^{\mathcal{A}}, \subseteq)$, so the supremum of this lattice $\top$ is equal to $\emptyset$ and the infimum $\bot$ is equal to $\mathcal{A}$. Then, the security levels of sent and received messages in the generalized roles of A according to $\mathsf{F}_{\mathrm{NSL}}$ are as shown in Table 8.

Table 8 shows that the generalized roles $\mathcal{A}_G^1$ and $\mathcal{A}_G^2$ are $\mathsf{F}_{\mathrm{NSL}}$-increasing.

From the generalized roles of B, we deduce that

$$\mathcal{B}_G^{1^-} = \{A, Y\}_{k_b}, \qquad \mathcal{B}_G^{1^+} = \left\{\left\langle Y, N_b^i\right\rangle, B\right\}_{k_a}. \tag{23}$$

Then, the security levels of sent and received messages in the generalized roles of B according to $\mathsf{F}_{\mathrm{NSL}}$ are as in Table 9.

Table 9 shows that the generalized role $\mathcal{B}_G^1$ is $\mathsf{F}_{\mathrm{NSL}}$-increasing. Since all generalized roles of NSL are $\mathsf{F}_{\mathrm{NSL}}$-increasing, then it is an $\mathsf{F}_{\mathrm{NSL}}$-increasing protocol. Furthermore, since $\mathsf{F}_{\mathrm{NSL}}$ is $\mathcal{C}_{\mathrm{NSL}}$-safe interpretation function, we conclude, from our main result, that NSL is $\mathcal{C}_{\mathrm{NSL}}$-correct with respect to secrecy.

3.2. Analysis of the NSL-Protocol with the Homomorphism Property. This section presents the analysis of the Needham-Schroeder Lowe (NSL) protocol with the homomorphism property of encryption:

$$enc(g(x, y)) = g'(enc(x), enc(y)), \tag{24}$$

Table 10: Analysis of NSL protocol: role of A.

α	r	$\ulcorner\alpha\urcorner$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$
N_a^i	$\mathcal{A}_G^1$	$\{A, B\}$	$\{B\}$	$\top$	*Yes*
X	$\mathcal{A}_G^2$	τ_X	$\{B\}$	$\{A\}$	*No*

where g and g' are any operators (usually the same) from the message algebra. For example, $enc(m_1 \cdot m_2) = enc(m_1) \cdot enc(m_2)$ is satisfied when using block ciphers with Electronic Code Block (ECB) and when m_1 has the same size as requested by the encryption system (64 bits for DES). In fact, the ECB consists of splitting the message into blocks of n-bits and encrypting each of them separately. Many asymmetric systems like RSA, Elgamal, Benaloh, and Paillier are also homomorphic for some algebraic operators.

Related works that take into consideration the homomorphism property in the analysis of cryptographic protocols are rare and most of them do not deal with an unbounded number of sessions. Among the important results obtained in this direction are those described in [40–42] which address the intruder deduction problem and propose a PTIME-decision procedure for it. Despite the importance of the result, the resolution of the intruder deduction problem is not enough to analyze a protocol that can exhibit an infinite number of sessions.

The interpretation functions-based method does not suffer from this problem since it is proved that if the protocol is increasing then it is sufficient to guarantee its correctness for the secrecy property under an unbounded number of sessions. In the sequel, we show how the homomorphism property can affect the analysis of a protocol like NSL.

3.2.1. NSL Context of Verification with the Homomorphism Property. First, we need to change the context of verification $\mathcal{C}_{\mathrm{NSL}}$ used in the previous section, so that we consider the homomorphism property. Let $\mathcal{C}'_{\mathrm{NSL}}$ be a context of verification such as $\mathcal{C}'_{\mathrm{NSL}} = \langle(\mathcal{N}_{\mathrm{NSL}}, \Sigma_{\mathrm{NSL}}, \mathcal{E}'_{\mathrm{NSL}}), \vDash_{\mathrm{NSL}} \mathcal{K}_{\mathrm{NSL}}, \mathcal{L}^{\sqsupseteq}_{\mathrm{NSL}}, \ulcorner\cdot\urcorner_{\mathrm{NSL}}\rangle$ where all items are the same as those defined in the context $\mathcal{C}_{\mathrm{NSL}}$ except the equational theory $\mathcal{E}'_{\mathrm{NSL}}$ that includes homomorphism of encryption and it is as follows:

$$\begin{aligned} fst(pair(x, y)) &= x, \\ snd(pair(x, y)) &= y, \\ dec(enc(x, k), k^{-1}) &= x, \\ \langle\langle m_1, m_2\rangle, m_3\rangle &= \langle m_1, \langle m_2, m_3\rangle\rangle, \\ enc(\langle x, y\rangle, z) &= \langle enc(x, z), enc(y, z)\rangle. \end{aligned} \tag{25}$$

The last two equations of $\mathcal{E}'_{\mathrm{NSL}}$ reflect the homomorphic property of the encryption *enc* and the signature *sign* with respect to the concatenation operator $\langle\cdot\rangle$.

3.2.2. NSL Safe Interpretation Function with the Homomorphism Property. We use the DEKAN function (that selects the direct encryption keys and neighbors) to analyze the Needham-Schroeder Lowe protocol under the equational theory $\mathcal{E}'_{\mathrm{NSL}}$. With the homomorphism property, the DEKAN function becomes equal to the DEK function (that selects direct encryption keys). Indeed, the selection is forbidden beyond encryption and the selection is always calculated on the message $m_{\downarrow}$ (the canonical form of m according to the rewriting system obtained from the equational theory $\mathcal{E}'_{\mathrm{NSL}}$, when the equality is replaced by $\rightarrow$) instead of m. For example, with the homomorphism property the message $\{\alpha_1, \ldots, \alpha_n\}_k$, where $\alpha_1, \ldots, \alpha_n$ are atomic, is always reduced, by the homomorphism property, to $\{\alpha_1\}_k, \ldots, \{\alpha_n\}_k$. To calculate the security level of a component α_i in the message $\{\alpha_1, \ldots, \alpha_n\}_k$, the DEKAN function is applied on its canonical form that is $\{\alpha_1\}_k, \ldots, \{\alpha_n\}_k$. It follows that the neighbors are never chosen, and we conclude that the DEKAN function behaves as the DEK function.

Table 11: Flaw in the NSL-protocol under the homomorphic assumption.

1.1. $A \rightarrow I : \{A, N_a^1\}_{k_i}$
2.1. $I(A) \rightarrow B : \{A, N_a^1\}_{k_b}$
2.2. $B \rightarrow I(A) : \{N_a^1, N_b^2, B\}_{k_a}$
1.2. $I \rightarrow A : \{N_a^1, N_b^2, I\}_{k_a}$
1.3. $A \rightarrow I : \{N_b^2\}_{k_i}$
2.3. $I(A) \rightarrow B : \{N_b^2\}_{k_b}$

3.2.3. Verifying Whether the NSL Role-Based Specification Is Increasing. From the generalized roles of A, we deduce that

$$\begin{aligned} \mathcal{A}_G^{1\,-} &= \varnothing, \qquad \mathcal{A}_G^{1\,+} = \left\{A, N_a^i\right\}_{k_b}, \\ \mathcal{A}_G^{2\,-} &= \left\{\left\langle N_a^i, X\right\rangle, B\right\}_{k_a}, \qquad \mathcal{A}_G^{2\,+} = \{X\}_{k_b}. \end{aligned} \tag{26}$$

The security levels of sent and received messages in the generalized roles of A using the DEKAN function $\mathsf{F}_{\mathrm{NSL}}$ are as shown in Table 10.

Table 10 shows that the generalized role $\mathcal{A}_G^2$ is not increasing. Indeed, the message X is sent with the security level $\{B\}$ which is not always in its security level $\ulcorner X\urcorner \cup \{A\}$ estimated from the received messages. In this case, we can never find a safe interpretation function that guarantees the correctness of the protocol simply because the protocol is flawed and it could be attacked as shown by Table 11. Notice that message of step 1.2 could be generated by the intruder from the message 2.2 using the homomorphic property. In fact, $\{N_a^1, N_b^2, B\}_{k_a} = \{N_a^1, N_b^2\}_{k_a}, \{B\}_{k_a}$ and since the intruder can produce $\{I\}_{k_a}$, then using the homomorphic property he can produce $\{N_a^1, N_b^2, I\}_{k_a}$ by simply concatenating $\{N_a^1, N_b^2\}_{k_a}$ with $\{I\}_{k_a}$.

Table 12: Analysis of NSL protocol: role of B.

α	r	$\ulcorner\alpha\urcorner_0$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{NSL}}(\alpha, r^+) \subseteq^? \ulcorner\alpha\urcorner_0 \cup \mathsf{F}_{\mathrm{NSL}}(\alpha, r^-)$
N_b^i	$\mathcal{B}_G^1$	$\{A, B\}$	$\{A\}$	$\top$	*Yes*
Y	$\mathcal{B}_G^1$	$\top$	$\{A\}$	$\{B\}$	*No*

This is because the DEKAN function (that selects direct encryption keys and neighbors) with the homomorphism property acts as the DEK function (that selects the direct encryption keys only) and does not select the neighbors. Therefore, to correct this protocol in the presence of such property, we should find a means (different from the concatenation, such as signatures), allowing A to conclude that N_b is from B. To overcome this problem, we suggest the use of signatures to indicate the correct security level of sent messages. For instance, we can replace the message $\{A, N_a\}_{k_b}$ at step 1 of the NSL-protocol with the message $\{\{N_a\}_{k_a^{-1}}\}_{k_b}$. In this case, signing the message N_a by k_a^{-1} is a safe way to indicate that it comes from the agent A. In the same way, we can replace the message of step 2 of the NSL-protocol by the message $\{\{\langle N_a, N_b\rangle\}_{k_b^{-1}}\}_{k_a}$. Notice that if signature is also homomorphic like encryption, operators like hash functions can be used to forge such type of inseparable links between some elements.

From the generalized roles of B, we deduce that

$$\mathcal{B}_G^{1^-} = \{A, Y\}_{k_b}, \qquad \mathcal{B}_G^{1^+} = \left\{\left\langle Y, N_b^i\right\rangle, B\right\}_{k_a}. \tag{27}$$

Then, the security level of sent and received messages in the generalized roles of B using the DEKAN function $\mathsf{F}_{\mathrm{NSL}}$ is as shown in Table 12.

Table 12 shows that the generalized role $\mathcal{B}_G^1$ is not increasing. Indeed, the message Y is sent with the security level $\{A\}$ which is not in its security level $\{B\}$ estimated from the received messages. As for the role of A, we suggest to replace the message $\{A, N_a\}_{k_b}$ of step 1 of the NSL-protocol with the message $\{\{N_a\}_{k_a^{-1}}\}_{k_b}$. In this case, signing the message N_a by k_a^{-1} is a safe way to indicate that it comes from the agent A. In the same way, we can replace the message of step 2 of the NSL-protocol with the message $\{\{\langle N_a, N_b\rangle\}_{k_b^{-1}}\}_{k_a}$.

Therefore, the NSL-protocol is not increasing and, so, we cannot deduce anything about its correctness for the secrecy property. However, the protocol, with the changes suggested above, is increasing and it is therefore correct for the secrecy property even in the presence of the homomorphism property. Table 13 summarizes these changes.

By using the DEK function or the DEKAN function, this new version of the Needham-Schroeder-Lowe protocol is increasing and then respects the secrecy property. Indeed, with this version N_a is received with the security level $\{A, B\}$ and so it could be sent to A. Also, N_b is received with security level $\{A, B\}$ and so it could be sent to B.

Table 13: Needham schroeder lowe protocol: corrected version.

1. $A \rightarrow B : \{\{N_a\}_{k_a^{-1}}\}_{k_b}$
2. $B \rightarrow A : \{\{\langle N_a, N_b\rangle\}_{k_b^{-1}}\}_{k_a}$
3. $A \rightarrow B : \{N_b\}_{k_b}$

Table 14: SET protocol.

1. $C \rightarrow M : C, N_c$
2. $M \rightarrow C : \{M, C, XID, N_c, N_M, CA(G)\}_{k_M^{-1}}$
3. $C \rightarrow M : OI, DualSign, \{PI\}_{k_G}$
4. $M \rightarrow G : \{\{AuthReqData, LinkOIPI\}_{k_M^{-1}}\}_{k_G}, DualSign, \{PI\}_{k_G}$
5. $G \rightarrow M : \{\{M, C, XID, AuthRRTags, PurchAmt, AuthCode\}_{k_G^{-1}}\}_{k_M}$
6. $M \rightarrow C : \{M, C, XID, N_C, AuthCode\}_{k_M^{-1}}$

4. Analysis of the SET Protocol

Electronic commerce, commonly denoted by e-commerce, or eCommerce, consists in buying and selling goods or services over the Internet. To make a purchase, a customer usually submits his credit card number to a merchant protected according to a specific cryptographic protocol such as SET and SSL.

Many researchers have addressed the problem of analyzing the SET protocol during the last years. For example, Paulson et al. tried in [43, 44] to prove some security properties (authentication and repudiation properties) during the purchase phase as well as during the cardholder registration one by using the inductive approach and the theorem prover "Isabelle." However, the analysis was very difficult and could not be achieved because of the complexity of the exchanged messages and the complexity of targeted properties. However, they discovered some flaws related to repudiation and authentication properties. In [45], the authors present some weaknesses in the purchase phase of the SET protocol and propose their correct version. The weaknesses concern the repudiation property. However, no results concerning the secrecy property have been given till now. In the following, we show the efficiency of the interpretation function-based method when analyzing such difficult protocol and how it could be used to give a correct version of this protocol.

The SET protocol [27–29] has been proposed by a consortium of credit card and software companies. It aims to protect sensitive cardholder information, to ensure payment integrity, and confidentiality, and to authenticate merchants and cardholders. It contains five subprotocols: cardholder registration; merchant and payment gateway registration; purchase request; payment authorization and transaction payment. We focus here on the analysis of the purchase phase which involves three parties: the cardholder (C); the merchant (M) and a payment gateway (G). The formal

description of the purchase phase of SET, denoted in the sequel by p_{SET}, is defined by Table 14.

The *OI*, *PI*, *DualSign,* and *Link OIPI* are as follows:

$$\begin{aligned}
OI &:= OIData, H(PIData),\\
OIData &:= M, C, XID, N_M, N_C, HOD,\\
HOD &:= H(OrderDesc, PurchAmt),\\
PIData &:= PIHead, PANData,\\
PIHead &:= M, C, XID, HOD, PurchAmt,\\
&\quad MerID, H(XID, CardSecret),\\
PANData &:= PAN, PANSecret,\\
PI &:= PIHead, H(OIData), PANData,\\
DualSign &:= \{H(PIData), H(OIData)\}_{k_c^{-1}},\\
LinkOIPI &:= H(AuthReqData, DualSign, \{PI\}_{K_G}),\\
AuthReqData &:= H(OIData), \mathrm{HOD},\\
&\quad M, C, XID, AuthRRTags.
\end{aligned} \tag{28}$$

PAN is the cardholder's primary account number, and *PANSecret* is a secret number known by the cardholder and used to prove his identity when making purchases. The *OrderDesc* is the description of the customer's detailed order and *PurchAmt* is the total amount of the purchase order. The intuitive meaning of each step is as follows.

(i) Initialization request (step 1): before starting the purchase, the cardholder and the merchant agree upon the order description and its price. This shopping step is out of the SET protocol. The cardholder then sends to the merchant his local ID (C) and a fresh random challenge N_c .

(ii) Initialization response (step 2): The merchant M generates a transaction ID (XID) and sends it to the costumer with the gateway's public encryption key certificate ($CA(G)$).

(iii) Order request (step 3): after validating the signature of the merchant and the certificates of the gateway, the cardholder sends an order request which contains the payment instruction (PI), the order information (OI), and the dual signature ($DualSign$) to the merchant.

(iv) Authorization request (step 4): the message sent during this step contains PI and $DualSign$ sent by the cardholder, the hash codes $H(OIData)$ and HOD. This information enables the gateway to verify the dual signature, the different IDs involved in the transaction, and the authorization request/response tags (*authRRTags*) that should be returned in the authorization response. The purpose of *AuthRRTags* is to match the request/response paired messages; it contains the merchant's financial ID and some optional data that are used by the merchant's bank to authorize the transaction.

(v) Authorization response (step 5): if both PI and OI agree, the gateway proceeds to the transaction authorization using the existing financial networks. If the authorization is allowed, the gateway sends the authorization response containing *authRRTags* copied from step 4, the purchase amount, and the transaction status (a boolean value).

(vi) Purchase response (step 6): the merchant verifies the gateway's signature and whether the IDs and the *authRRTags* in the response match with those sent in his request message. Then, he forwards to the cardholder the authorization status combined with IDs and challenges involved in the transaction.

Table 15

n	::=	A	(Principal Identifier)
	|	N_a	(Nounce)
	|	k_a^{-1}	(Private key)
	|	k_a	(Public key)

Table 16

$m,\ m_1,\ m_2$	::=	n	
	|	$enc(m, k_a)$	(Encryption Function)
	|	$dec(m, k_a^{-1})$	(Signature Function)
	|	$pair(m_1, m_2)$	(Pair Function)
	|	$fst(pair(m_1, m_2))$	(First Function)
	|	$snd(pair(m_1, m_2))$	(Second Function)
	|	$H(m)$	(Hash Function)

In the following, we focus on verifying the secrecy property of this protocol by using the interpretation functions-based method. To that end, we follow the steps described by Figure 2. First, we define the context of verification and a safe interpretation function for it. After that, the role-based specification is generated to verify whether this protocol is increasing.

4.1. SET Context of Verification in the Dolev-Yao Model. Let $\mathcal{C}_{\mathrm{SET}} = \langle(\mathcal{N}_{\mathrm{SET}}, \Sigma_{\mathrm{SET}}), F_0, \mathcal{K}_{\mathrm{SET}}, \mathcal{L}^{\exists}_{\mathrm{SET}}, \ulcorner\cdot\urcorner_{\mathrm{SET}}\rangle$ be a context of verification where $\mathcal{N}_{\mathrm{SET}}$ is the set of atomic messages given by BNF grammar shown in Table 15 and $\Sigma_{\mathrm{SET}} = \{enc, dec, pair, fst, snd, hash\}$.

As usual, we write $\{m\}_k$ instead of $enc(m, k)$ or $sign(m, k)$. Also, we write m_1, m_2 instead of $pair(m_1, m_2)$, $H(m)$ instead of $hash(m)$ and $sign(m, k^{-1})$ instead of $dec(m, k^{-1})$ if k^{-1} is a private key. Hence, the set of messages $\mathcal{M}_{\mathrm{SET}}$ is the set of messages that respect BNF rules shown in Table 16.

Let $\mathcal{E}_{\text{SET}}$ be the equational theory containing the following equations:

$$\begin{aligned} fst(\langle x, y\rangle) &= x, \\ snd(\langle x, y\rangle) &= y, \\ dec(enc(x,k),k^{-1}) &= x, \\ enc(dec(x,k^{-1}),k) &= x, \\ check(sign(m,k_a^{-1}),k_a) &= ok, \\ \langle\langle m_1, m_2\rangle, m_3\rangle &= \langle m_1, \langle m_2, m_3\rangle\rangle. \end{aligned} \tag{29}$$

The intruder model $\vDash_{\text{SET}}$ is the famous Dolev-Yao model for asymmetric key given as follows:

$$\begin{aligned} &\text{(knowledge)}\ \frac{\square}{M \vDash_{\text{SET}} m} \quad [m \in M], \\ &\text{(construction)}\ \frac{M \vDash_{\text{SET}} m_1 \cdots M \vDash_{\text{SET}} m_n}{M \vDash_{\text{SET}} f(m_1, \ldots, m_n)} \quad [f \in \Sigma_{\text{SET}}], \\ &(\mathcal{E}\text{-equality})\ \frac{M \vDash_{\text{SET}} m}{M \vDash_{\text{SET}} m'} \quad [m =_{\mathcal{E}_{\text{SET}}} m']. \end{aligned} \tag{30}$$

The initial knowledge of principals $\mathcal{K}_{\text{SET}}$ is as follows. Each principal knows his identity, the identities of other principals, his public and private key, all the public keys of the other principals and his card numbers (its *PAN* and its *PANSecret*). Also, each principal can generate fresh values.

The security lattice $\mathcal{L}_{\text{SET}}$ is the same as the one defined in the NSL-example, that is, $\mathcal{L}_{\text{SET}} = \mathcal{L}_0 = (2^{\mathcal{I}}, \subseteq)$. 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 $\mathcal{I}$.

The typing relation $\ulcorner \cdot \urcorner_{\text{SET}}$ (or shortly denoted by $\ulcorner \cdot \urcorner$ if there is no confusion) is a partial function from $\mathcal{M}_{\text{SET}}$ to $\mathcal{L}_{\text{SET}}$ defined as follows:

$$\begin{aligned} &[N_c \longmapsto \bot, PurchAmt \longmapsto \bot, MerID \longmapsto \bot, \\ &CardSecret \longmapsto \{C, G\}, \\ &PANSecret \longmapsto \{C, G\}, PAN \longmapsto \{C, G\}, XID \longmapsto \bot, \\ &N_M \longmapsto \bot, CA(G) \longmapsto \bot, \\ &AuthRRTags \longmapsto \{M, G\}, AuthCode \longmapsto \bot]. \end{aligned} \tag{31}$$

4.2. SET Safe Interpretation Function in the Dolev-Yao Model. We use an extended version of the DEKAN function to deal with hash functions. To that end, we follow the guideline defined in Section 2.2. In fact, the guideline states that we have to consider the interpretation function as a composition of a selection and rank function (i.e., $\mathsf{F}_{\text{SET}} = \mathsf{I}_{\text{SET}} \circ \mathsf{S}_{\text{SET}}$). In this example, we consider the selection function S_{SET} that allows to select principal identifiers that are at distance 0 (neighbors) and component at distance 1 (encryption keys). A hashed message is considered in this example as a black box for the selection function. This means that selection function could not select components from a hashed message and this done by making the hashed components at distance strictly greater than 1 (the maximum distance that the selection function can reach to select components). Also, to have a safe function, we forbid the selection of components that are beyond an encryption operator. To that end, we set distances so that any two components separated by an encryption are at distance equal to ∞. More precisely, the selection function S_{SET} could be defined by considering the costs between nodes in messages as described by Figure 4, where d is as follows:

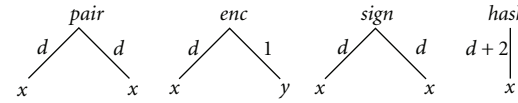

Figure 4: Safe costs assignment.

$$d = \begin{cases} \infty & \text{if } x \in \{\text{enc}, \text{hash}\} \\ 0 & \text{else.} \end{cases} \tag{32}$$

The DEKAN function F_{SET}, used to analyze the SET protocol, could be built by composing the selection function and the rank function as follows:

$$\mathsf{F}_{\text{SET}} = \mathsf{I}_{\text{SET}} \circ \mathsf{S}_{\text{SET}}, \tag{33}$$

where I_{SET} gives the rank$\{A\}$ to any principal identity A that is selected as a neighbor and the rank $\ulcorner k^{-1} \urcorner$ to the keys k which are selected as direct keys of encryption.

4.3. SET Roles-Based Specification. For more details about how we compute a roles-based specification from a protocol and a context of verification, we refer the reader to Section 2.6.1. Following the steps described in Section 2.6.1, the SET roles-based specification is

$$\mathcal{R}_G(p_{\text{SET}}) = \left\{\mathcal{C}_G^1, \mathcal{C}_G^2, \mathcal{C}_G^3, \mathcal{M}_G^1, \mathcal{M}_G^2, \mathcal{M}_G^3, \mathcal{G}_G^1\right\}, \tag{34}$$

where $\mathcal{C}_G^1$, $\mathcal{C}_G^2$, and $\mathcal{C}_G^3$ (the generalized roles of C) are as follows:

$$\begin{aligned} \mathcal{C}_G^1 &= i.1.\ C \longrightarrow I(M) : C, N_c^i, \\ \mathcal{C}_G^2 &= \begin{cases} i.1.\ C \longrightarrow I(M) : C, N_c^i, \\ i.2.\ I(M) \longrightarrow C : \{M, C, X_1, N_c^i, X_2, CA(G)\}_{k_M^{-1}}, \\ i.3.\ C \longrightarrow I(M) : IO_G^C, DualSign_G^C, \left\{PI_G^C\right\}_{k_G}, \end{cases} \\ \mathcal{C}_G^3 &= \begin{cases} i.1.\ C \longrightarrow I(M) : C, N_c^i, \\ i.2.\ I(M) \longrightarrow C : \{M, C, X_1, N_c^i, X_2, CA(G)\}_{k_M^{-1}}, \\ i.3.\ C \longrightarrow I(M) : IO_G^C, DualSign_G^C, \left\{PI_G^C\right\}_{k_G}, \\ i.6.\ I(M) \longrightarrow C : \left\{M, C, X_1, N_C^i, X_3\right\}_{k_M^{-1}}. \end{cases} \end{aligned} \tag{35}$$

The OI_G^C, $DualSign_G^C$, and PI_G^C are, respectively, OI, $DualSign$, and PI in which XID, N_M and N_c are, respectively, replaced by X_1, X_2, and N_c^i. The generalized roles of M ($\mathcal{M}_G^1$, $\mathcal{M}_G^2$, and $\mathcal{M}_G^3$) are as follows:

$$\mathcal{M}_G^1 = \begin{cases} i.1.\ I(C) \longrightarrow M : C, Y_1, \\ i.2.\ M \longrightarrow I(C) : \\ \quad \left\{M, C, XID^i, Y_1, N_M^i, CA(G)\right\}_{k_M^{-1}}, \end{cases}$$

$$\mathcal{M}_G^2 = \begin{cases} i.1.\ I(C) \longrightarrow M : C, Y_1, \\ i.2.\ M \longrightarrow I(C) : \\ \quad \left\{M, C, XID^i, Y_1, N_M^i, CA(G)\right\}_{k_M^{-1}}, \\ i.3.\ I(C) \longrightarrow M : OIData_G^M, Y_7, DualSign_G^M, Y_8, \\ i.4.\ M \longrightarrow I(G) : \\ \quad \left\{\left\{AuthReqData, LinkOIPI\right\}_{k_M^{-1}}\right\}_{k_G}, \\ \quad DualSign_G^M, Y_8, \end{cases}$$

$$\mathcal{M}_G^3 = \begin{cases} i.1.\ I(C) \longrightarrow M : C, Y_1, \\ i.2.\ M \longrightarrow I(C) : \\ \quad \left\{M, C, XID^i, Y_1, N_M^i, CA(G)\right\}_{k_M^{-1}}, \\ i.3.\ I(C) \longrightarrow M : OIData_G^M, Y_7, DualSign_G^M, Y_8, \\ i.4.\ M \longrightarrow I(G) : \\ \quad \left\{\left\{AuthReqData_G^M, LinkOIPI_G^M\right\}_{k_M^{-1}}\right\}_{k_G}, \\ \quad DualSign_G^M, Y_8, \\ i.5.\ I(G) \longrightarrow M : \\ \quad \left\{\left\{M, C, XID^i, authRRTags, Y_3, Y_9\right\}_{k_G^{-1}}\right\}_{k_M}, \\ i.6.\ M \longrightarrow I(C) : \left\{M, C, XID^i, Y_1, Y_9\right\}_{k_M^{-1}}. \end{cases} \tag{36}$$

The $OIData_G^M$, $DualSign_G^M$, $AuthReqData_G^M$ and $LinkOIPI_G^M$ are as follows:

$$\begin{aligned} OIData_G^M &:= M, C, XID^i, N_M^i, Y_1, Y_2, \\ DualSign_G^M &:= \left\{Y_7, H\left(OIData_G^M\right)\right\}_{k_c^{-1}}, \\ LinkOIPI_G^M &:= H\left(AuthReqData, DualSign_G^M, Y_8\right), \\ AuthReqData_G^M &:= H\left(OIData_G^M\right), Y_2, M, C, XID^i, \\ &\quad AuthRRTags. \end{aligned} \tag{37}$$

The generalized role of G ($\mathcal{G}_G^1$) is as follows:

$$\mathcal{G}_G^1 = \begin{cases} i.4.\ I(M) \longrightarrow G : \\ \quad \left\{\left\{AuthReqData_G^G, LinkOIPI_G^G\right\}_{k_M^{-1}}\right\}_{k_G}, \\ \quad DualSign_G^G, \left\{PI_G^G\right\}_{k_G}, \\ i.5.\ G \longrightarrow I(M) : \\ \quad \left\{\left\{M, C, Z_1, Z_{11}, Z_6, AuthCode\right\}_{k_G^{-1}}\right\}_{k_M}. \end{cases} \tag{38}$$

The PI_G^G, $DualSign_G^G$, $AuthReqData_G^G$, and $LinkOIPI_G^G$ are as follows:

$$\begin{aligned} PIData_G^G &:= PIHead_G^G, PANData_G^G \\ PIHead_G^G &:= M, C, Z_1, Z_4, Z_6, MerID, Z_8 \\ PANData_G^G &:= Z_9, Z_{10} \\ PI_G^G &:= PIHead_G^G, Z_{12}, PANData_G^G \\ DualSign_G^G &:= \left\{H\left(PIData_G^G\right), Z_{12}\right\}_{k_c^{-1}} \\ LinkOIPI_G^G &:= H\left(AuthReqData_G^G, DualSign_G^G, \{PI\}_{K_G}\right) \\ AuthReqData_G^G &:= Z_{12}, Z_4, M, C, Z_1, Z_{11}. \end{aligned} \tag{39}$$

4.4. Verifying Whether the SET Role-Based Specification is Increasing. Now, we can verify whether the role-based specification of SET is $\mathsf{F}_{\mathrm{SET}}$-increasing. More precisely, we need accordingly to verify that principals do not decrease the security level of components from step to another. To that end, we verify this condition on the each generalized role of the protocol (i.e., on each point of view of protocol participants). To that end, we should verify whether the security level of sent messages is a subset of the security level of these messages in the receieved steps that is,

$$\forall r \cdot \mathsf{F}_{\mathrm{SET}}(\alpha, r^+) \subseteq \ulcorner \alpha \urcorner \cup \mathsf{F}_{\mathrm{SET}}(\alpha, r^-). \tag{40}$$

(i) Let us start with the generalized roles of C. According to the definition of r^+ and r^-, we have

$$\begin{aligned} \mathcal{C}_G^{1^-} &= \varnothing, \\ \mathcal{C}_G^{1^+} &= C, N_c^i, \\ \mathcal{C}_G^{2^-} &= \left\{M, C, X_1, N_c^i, X_2, CA(G)\right\}_{k_M^{-1}}, \\ \mathcal{C}_G^{2^+} &= IO_G^C, DualSign_G^C, \left\{PI_G^C\right\}_{k_G}. \end{aligned} \tag{41}$$

From the definition of $\mathsf{F}_{\mathrm{SET}}$ and $\mathcal{C}_G^{2^+}$, it follows that, for all atomic message α, we have that

$$\begin{aligned} &\mathsf{F}_{\mathrm{SET}}\left(\alpha, \mathcal{C}_G^{2^+}\right) \\ &\quad = \mathsf{F}_{\mathrm{SET}}\left(\alpha, IOData_G^C, \{PIHead, PANData\}_{k_G}\right). \end{aligned} \tag{42}$$

Now, the security levels of sent and received messages in the generalized roles of C using the $\mathsf{F}_{\mathrm{SET}}$ function are as shown in Table 17.

From the analysis of the role of C shown in Table 17, we deduce that its generalized roles are not increasing. This is due to the fact that an unknown message X_1 is put beside the secret components *CardSecret*, *PAN*, and *PANSecret* and this lowers their security level. Moreover, putting the identity

Table 17: SET analysis: role of C.

α	r	$\ulcorner\alpha\urcorner$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$
N_c^i	$\mathcal{C}_G^1$	$\bot$	$\bot$	$\top$	*Yes*
N_c^i	$\mathcal{C}_G^2$	$\bot$	$\bot$	$\top$	*Yes*
X_1	$\mathcal{C}_G^2$	τ_{X_1}	$\bot$	$\bot$	*Yes*
X_2	$\mathcal{C}_G^2$	τ_{X_2}	$\bot$	$\bot$	*Yes*
PurchAmt	$\mathcal{C}_G^2$	$\bot$	$\{C, G\} \cup \tau_{X_1}$	$\top$	*Yes*
MerID	$\mathcal{C}_G^2$	$\bot$	$\bot$	$\top$	*Yes*
CardSecret	$\mathcal{C}_G^2$	$\{C, G\}$	$\{C, M, G\} \cup \tau_{X_1}$	$\top$	*No*
PAN	$\mathcal{C}_G^2$	$\{C, G\}$	$\{C, M, G\} \cup \tau_{X_1}$	$\top$	*No*
PANSecret	$\mathcal{C}_G^2$	$\{C, G\}$	$\{C, M, G\} \cup \tau_{X_1}$	$\top$	*No*

Table 18: SET analysis: role of M.

α	r	$\ulcorner\alpha\urcorner$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$
Y_1	$\mathcal{M}_G^1$	τ_{Y_1}	$\bot$	$\bot$	*Yes*
XID^i	$\mathcal{M}_G^1$	$\bot$	$\bot$	$\top$	*Yes*
N_M^i	$\mathcal{M}_G^1$	$\bot$	$\bot$	$\top$	*Yes*
$CA(G)$	$\mathcal{M}_G^1$	$\bot$	$\bot$	$\top$	*Yes*
Y_2	$\mathcal{M}_G^2$	τ_{Y_2}	$\bot$	$\bot$	*Yes*
XID^i	$\mathcal{M}_G^2$	$\bot$	$\bot$	$\bot$	*Yes*
AuthRRTags	$\mathcal{M}_G^2$	$\{M, G\}$	$\bot$	$\top$	*No*

M beside *CardSecret*, *PAN*, and *PANSecret* means that these messages could be known by M according to our interpretation function. Therefore, if X_1 (XID in the description of the protocol step 3) and M are hashed in *PIHead*, then this role can be proved increasing. More precisely, we redefine *PIHead* as follows:

$$PIHead = C, HOD, PurchAmt, MerID, H(M, XID), H(CardSecret). \tag{43}$$

(ii) From the generalized roles of M, we deduce that

$$\begin{aligned}
\mathcal{M}_G^{1^-} &= C, Y_1,\\
\mathcal{M}_G^{1^+} &= \left\{M, C, XID^i, Y_1, N_M^i, CA(G)\right\}_{k_M^{-1}},\\
\mathcal{M}_G^{2^-} &= \mathcal{M}_G^{1^-} \cup \left\{OIData_G^M, Y_7, DualSign_G^M, Y_8\right\},\\
\mathcal{M}_G^{2^+} &= \left\{\left\{AuthReqData, LinkOIPI\right\}_{k_M^{-1}}\right\}_{k_G},\\
&\quad DualSign_G^M, Y_8,\\
\mathcal{M}_G^{3^-} &= \mathcal{M}_G^{2^-}\\
&\quad \cup \left\{\left\{\left\{M, C, XID^i, AuthRRTags, Y_3, Y_9\right\}_{k_G^{-1}}\right\}_{k_M}\right\},\\
\mathcal{M}_G^{3^+} &= \left\{M, C, XID^i, Y_1, Y_9\right\}_{k_M^{-1}}.
\end{aligned} \tag{44}$$

The security levels of sent and received messages in the generalized roles of M using the DEKAN function are as shown in (Table 18).

From Table 19, we can deduce that the generalized roles of M are not increasing. This is because the message *AuthRRTags* is sent at step 4 of the generalized role $\mathcal{M}_G^2$, with the security level $\{C, M, G\}$, which is not greater than its real security level ($\{M, G\}$). This is since the identity C is beside *AuthRRTags* and that means that this message is for C which also lowers the security level of *AuthRRTags*. One way to make this role increasing is as follows. We remove (or put it beyond the encryption) the identity C from the message *AuthReqData* of step 4 of the generalized role. More precisely, we redefine *AuthReqData* as follows:

$$AuthReqData = H(OIData), HOD, M, XID, AuthRRTags. \tag{45}$$

(iii) From the generalized roles of G, we deduce that

$$\begin{aligned}
\mathcal{G}_G^{1^-} &= \left\{\left\{AuthReqData_G^G, LinkOIPI_G^G\right\}_{k_M^{-1}}\right\}_{k_G},\\
&\quad DualSign_G^G, \left\{PI_G^G\right\}_{k_G}\\
\mathcal{G}_G^{1^+} &= \left\{\left\{M, C, Z_1, Z_{11}, Z_6, AuthCode\right\}_{k_G^{-1}}\right\}_{k_M}.
\end{aligned} \tag{46}$$

Then, the security levels of sent and received messages in the generalized roles of M using the DEKAN function are as in Table 20.

Table 19: SET analysis: role of M.

α	r	$\ulcorner\alpha\urcorner$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$
Y_8	$\mathcal{M}_G^2$	τ_{Y_8}	$\bot$	$\bot$	*Yes*
XID^i	$\mathcal{M}_G^3$	$\bot$	$\{M, C, G\} \cup \tau_{Y_2}$	$\{M, C, Y_3\} \cup \tau_{Y_9}$	*Yes*
Y_1	$\mathcal{M}_G^3$	τ_{Y_1}	$\bot$	$\bot$	*Yes*
Y_9	$\mathcal{M}_G^3$	τ_{Y_9}	$\bot$	$\bot$	*Yes*

Table 20: SET analysis: role of G WLMV:prot.

α	r	$\ulcorner\alpha\urcorner$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$	$\mathsf{F}_{\mathrm{SET}}(\alpha, r^+) \subseteq \ulcorner\alpha\urcorner \cup \mathsf{F}_{\mathrm{SET}}(\alpha, r^-)$
Z_1	$\mathcal{G}_G^1$	τ_{Z_1}	$\{M, C, G\} \cup \tau_{Z_6} \cup \tau_{Z_{11}}$	$\{M, C, G, \tau_{Z_4}\} \cup \tau_{Z_6} \cup \tau_{Z_8} \cup \tau_{Z_{11}} \cup \tau_{Z_{12}}$	*Yes*
Z_6	$\mathcal{G}_G^1$	τ_{Z_6}	$\{M, C, G\} \cup \tau_{Z_1} \cup \tau_{Z_{11}}$	$\{M, C, G\} \cup \tau_{Z_1} \cup \tau_{Z_4} \cup \tau_{Z_8} \cup \tau_{Z_{11}} \cup \tau_{Z_{12}}$	*Yes*
Z_{11}	$\mathcal{G}_G^1$	$\tau_{Z_{11}}$	$\{M, C, G\} \cup \tau_{Z_1} \cup \tau_{Z_6}$	$\{M, C, G\} \cup \tau_{Z_1} \cup \tau_{Z_4} \cup \tau_{Z_{12}}$	*No*
AuthCode	$\mathcal{G}_G^1$	$\bot$	$\{M, C, G\} \cup \tau_{Z_1} \cup \tau_{Z_4} \cup \tau_{Z_6}$	$\top$	*Yes*

Table 21: A Secure version of the SET protocol.

1. $C \rightarrow M : C, N_c$
2. $M \rightarrow C : \{M, C, XID, N_c, N_M, CA(G)\}_{k_M^{-1}}$
3. $C \rightarrow M : OI, DualSign, \{PI\}_{k_G}$
4. $M \rightarrow G : \{\{AuthReqData, LinkOIPI\}_{k_M^{-1}}\}_{k_G}, DualSign, \{PI\}_{k_G}$
5. G → M : {{M, C, XID, AuthRRTags, PurchAmt, AuthCode}$_{k_G^{-1}}$}$_{k_M}$

5′. G → M : {{**M, XID, AuthRRTags, AuthCode**}$_{\mathbf{k_G^{-1}}}$}$_{\mathbf{k_M}}$, {{**M, C, XID, PurchAmt, AuthCode**}$_{\mathbf{k_G^{-1}}}$}$_{\mathbf{k_M}}$

6. $M \rightarrow C : \{M, C, XID, N_C, AuthCode\}_{k_M^{-1}}$

From Table 20, we can deduce that the generalized roles of G are not increasing. This is since the messages Z_{11} (*AuthRRTags*) and Z_6 (*PurchAmt*) are neighbors in step five while it is not the case in the step four. So, to make this role increasing we should put Z_6 (*PurchAmt*) beyond the encryption that contains *AuthRRTags* in step five. For instance, we replace the message of step five as follows:

$$5.\ G \longrightarrow M : \left\{\{M, C, XID, AuthRRTags, AuthCode\}_{k_G^{-1}}\right\}_{k_M}, \quad \left\{\{M, C, XID, PurchAmt, AuthCode\}_{k_G^{-1}}\right\}_{k_M}. \tag{47}$$

To sum up, we propose hereafter a secure version of the SET protocol for the secrecy property. It replaces the message of step 5 by message 5′ in Table 21.

Also, the *OI*, *PI*, *DualSign,* and *LinkOIPI* messages are as follows:

$$\begin{aligned}
OI &:= OIData, H(PIData),\\
OIData &:= M, C, XID, N_M, N_C, HOD,\\
HOD &:= H(OrderDesc, PurchAmt),\\
PIData &:= PIHead, PANData,\\
\text{PIHead(oldversion)} &:= C, \text{HOD}, \text{PurchAmt},\\
&\quad \text{MerID}, \text{M}, \text{XID},\\
&\quad \text{H(XID, CardSecret)},\\
\mathbf{PIHead} &:= C, HOD, PurchAmt, MerID,\\
&\quad \mathbf{H(M, XID)}, \mathbf{H(CardSecret)},\\
PANData &:= PAN, PANSecret,\\
PI &:= PIHead, H(OIData), PANData,\\
DualSign &:= \{H(PIData), H(OIData)\}_{k_C^{-1}},\\
LinkOIPI &:= H(AuthReqData, DualSign,\\
&\quad \{PI\}_{K_G}),\\
AuthReqData &:= H(OIData), HOD,\\
&\quad M, \not{C}, XID, AuthRRTags.
\end{aligned} \tag{48}$$

Such modifications allow to prove, by using the DEKAN function, that the new version is increasing and therefore respects the secrecy property.

5. Comparison with Related Works

The interpretation functions-based method is flexible framework allowing to verify different classes of protocols against

a variety of intruder models. This flexibility gives to it a great advantage when compared to the related works where almost all the approaches can deal only with a specific class of protocols and under some specific assumptions. When using the interpretation functions-based method, we do not have any restriction neither on the class of protocols that we can analyze nor on the intruder model and the algebraic properties. We are limited only by our capacity to find a suitable and safe interpretation function for the protocol that we want to analyze under a selected intruder model (context of verification).

Also, the analysis of protocols using the interpretation function-based method can help to discover the algebraic properties that should not be satisfied by an operator of encryption to guarantee the correctness of the analyzed protocol. This information is useful for the implementation of the protocol when time comes to select a specific encryption system. For instance, the commutativity property of the "exclusive or" and the "nilpotence" property should not be combined since they allow to decrypt a message without knowing the key.

Moreover, the interpretation function-based method has the significant advantage to ensure the correctness of cryptographic protocols for unbound number of sessions without any restriction neither on the number of principals nor on the size of exchanged messages. In fact, may existing approaches restrict their results by either limiting the number of sessions, the sizes of messages, or/and the number of involved principals to be able to overcome infinite number of traces that can be exhibited by a protocol. Other works tried to find a finite number of traces that are representative for all the others, that is, the analysis of the selected traces is enough to conclude about the correctness of the protocol. However, finding this representative and finite set of traces, if it exists, can be as complicated as the original problem even for some particular class of protocols. There are, however, some tentatives to analyze protocols without restriction on their traces, but they still suffer from some heavy restrictions both on the class of protocols that they can analyze and on the intruder model that they use.

In the remaining part of this section, we focus on the typing-based method [33] and the rank function-based method [32] which have some similarities with the interpretation functions-based method.

5.1. Typing-Based Method. In 1997, Abadi introduced in [33] a typing system to verify the secrecy property. This system uses three types {*secret, public, any*} as security levels and verifies whether the sent messages during the protocol are appropriately protected, that is, according to their security levels. The approach uses the Spi-calculus as a formal language to specify the analyzed protocol. Then, a typing system is applied on the protocol: if it typechecks we conclude that it ensures the secrecy property.

Compared to interpretation functions-based method, the secrecy by typing approach has the following restrictions.

(i) The exchanged messages have to respect some particular form. They must always be composed of four separated parts having the following type {*secret, public, any, confounder*}. This restriction helps to recognize the parts that are "secret," "public," and "any" (component with unknown security level) of messages. But, this involves that this approach cannot be applied to analyze existing protocols which have not been developed with this restriction in mind. The interpretation functions-based method, on the other side, does not require any particular form related to the exchanged messages during a protocol.

(ii) The secrecy by typing approach uses only the the Dolev and Yao model for the intruder, while interpretation functions-based method can deal with different intruder models.

5.2. Rank Functions-Based Method. In 1997, Schneider suggested, in [31, 32], an interesting approach to verify the cryptographic protocols, specified as a process in CSP [46]. A protocol is considered as correct for the secrecy property if all its exchanged messages have a suitable rank according to a well-designed rank function. A message is either secret or public and the result returned by the ranking function should be in harmony with these security levels, that is, it assigns ranks (negative value) to secret messages different from the ones (positive value) assigned to public messages.

The main ideas behind the typing system, the interpretation function, and the rand function approach are the same, that is, find someway to evaluate the security levels of exchanged messages and then evaluate if they are appropriately protected to guaranty the correctness of the protocol for the secrecy property. However, there are some fundamental differences between them. Hereafter, we focus on the difference between the approach presented in this paper and the rank function.

(i) For each protocol, we need to define a suitable rank function which is a complicated task. There are no universal functions, like DEK or DEKAN for the interpretation function approach, that are independent of the analyzed protocols. A rank function is extracted from the analyzed protocol itself and should respect some specific conditions (like safe condition for the interpretation function). Though the author makes a great effort, in [35, 47], to help find rank functions, the task remains complicated. On the other hand, in [25], the interpretation functions-based method gives a guideline which helps to define in an easier way safe interpretation functions.

(ii) The results given within the rank function approach are linked to a specific intruder ability. The approach based on interpretation function, on the other hand, is more flexible since it can handle a large variety of intruder abilities without reworking proofs.

(iii) When using a given interpretation function to analyze a protocol, even if we are unable to ensure its correctness the result is generally very helpful to either adapt the protocol or to build another

interpretation function. This is not generally the case with the rank function approach.

6. Conclusion

By analyzing the SET and the NSL protocols, this paper is an attempt to show that the interpretation functions-based method is an efficient technique to analyze and to ensure the correctness of cryptographic protocols for the secrecy property. Based on some special functions called "Interpretation Functions," this technique allows to guarantee the secrecy property under an unbound number of sessions and without any restriction on the size of messages sent by the intruder. To verify the secrecy property, it is sufficient to check whether a finite specification of the protocol, called generalized roles, respects some precise conditions. Intuitively, these conditions state that involved principals could not decrease, for a safe interpretation function, the security levels of exchanged messages. Another interesting feature of this approach is that it can handle different verification contexts with different intruder abilities including algebraic properties.

As future works, we want to implement the approach, extend it to authentication property, and propose more safe interpretation functions.

References

[1] M. J. Banks and J. L. Jacob, "Unifying theories of confidentiality," in *Proceedings of the International Symposium on Unifying Theories of Programming (UTP '10)*, vol. 6445 of *Lecture Notes in Computer Science*, pp. 120–136, 2010.

[2] V. Cortier, S. Delaune, and P. Lafourcade, "A survey of algebraic properties used in cryptographic protocols," *Journal of Computer Security*, vol. 14, no. 1, pp. 1–43, 2006.

[3] V. Cortier, S. Kremer, and B. Warinschi, "A survey of symbolic methods in computational analysis of cryptographic systems," *Journal of Automated Reasoning*, vol. 46, no. 3-4, pp. 225–259, 2011.

[4] S. Escobar, C. Meadows, and J. Meseguer, "A rewriting-based inference system for the NRL protocol analyzer: grammar generation," in *Proceedings of the ACM Workshop on Formal Methods in Security Engineering (FMSE '05)*, pp. 1–12, ACM, New York, NY, USA, November 2005.

[5] J. Feigenbaum, A. Johnson, and P. Syverson, "Probabilistic analysis of onion routing in a black-box model," in *Proceedings of the 6th ACM Workshop on Privacy in the Electronic Society (WPES '07)*, pp. 1–10, October 2007.

[6] P. Lafourcade, V. Terrade, and S. Vigier, "Comparison of cryptographic verification tools dealing with algebraic properties," in *Proceedings of the 6th international conference on Formal Aspects in Security and Trust (FAST '09)*, pp. 173–185, Springer, Berlin, Germany, 2010.

[7] C. Meadows, "What makes a cryptographic protocol secure?" in *Proceedings of the European Symposium on Programming (ESOP '03)*, Springer, April 2003.

[8] "Security and trust management," in *Proceedings of the 7th International Workshop (STM '11)*, C. Meadows, M. Carmen, and F. Gago, Eds., vol. 7170 of *Lecture Notes in Computer Science*, Springer, Copenhagen, Denmark, June 2011.

[9] A. Sabelfeld and A. C. Myers, "Language-based information-flow security," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, pp. 5–19, 2003.

[10] A. Sinha, "A survey of system security in contactless electronic passports," *Journal of Computer Security*, vol. 19, no. 1, pp. 203–206, 2011.

[11] D. Dolev, S. Even, and R. M. Karp, "On the security of ping-pong protocols," *Information and Control*, vol. 55, no. 1–3, pp. 57–68, 1982.

[12] R. M. Amadio, D. Lugiez, and V. Vanackère, "On the symbolic reduction of processes with cryptographic functions," *Theoretical Computer Science*, vol. 290, no. 1, pp. 695–740, 2003.

[13] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proceedings of the 14th IEEE Workshop on Computer Security Foundations (CSFW '01)*, pp. 82–96, Novia Scotia, Canada, June 2001.

[14] H. Comon-Lundh, F. Jacquemard, and N. Perrin, "Tree automata with one memory, set constraints, and ping-pong protocols," in *Proceedings of the 8th International Conference on Automata, Languages and Programming (ICALP '07)*, vol. 2076 of *Lecture Notes in Computer Science*, pp. 682–695, 2007.

[15] H. Comon-Lundh and V. Shmatikov, "Intruder deductions, constraint solving and insecurity decision in presence of exclusive or," in *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS '03)*, pp. 271–280, June 2003.

[16] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov, "Undecidability of bounded security protocols," in *Proceedings of the Workshop on Formal Methods and Security Protocols (FMSP '99)*, N. Heintze and E. Clarke, Eds., Trento, Italy, July 1999.

[17] M. Fiore and M. Abadi, "Computing symbolic models for verifying cryptographic protocols," in *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW '14)*, pp. 160–173, June 2001.

[18] D. Kindred, Theory generation for security protocols, 1999.

[19] G. Lowe, "Towards a completeness result for model checking of security protocols," in *Proceedings of the 11th IEEE Computer Security Foundations Workshop (CSFW '98)*, pp. 96–105, June 1998.

[20] J. Millen and V. Shmatikov, "Constraint solving for bounded-process cryptographic protocol analysis," in *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*, pp. 166–175, November 2001.

[21] M. Rusinowitch and M. Turuani, "Protocol insecurity with a finite number of sessions and composed keys is NP-complete," *Theoretical Computer Science*, vol. 299, no. 1–3, pp. 451–475, 2003.

[22] S. D. Stoller, "A bound on attacks on payment protocols," in *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS '01)*, pp. 61–70, June 2001.

[23] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron, "Extending the Dolev-Yao intruder for analyzing an unbounded number of sessions," vol. 2803, pp. 128–141.

[24] L. C. Paulson, "Mechanized proofs for a recursive authentication protocol," in *Proceedings of the 10th IEEE Computr Security Foundations Workshop (CSFW '97)*, pp. 84–94, June 1997.

[25] H. Houmani and M. Mejri, "Analysis of some famous cryptographic protocols using the interpretation-function-based method," *International Journal of Security and Its Applications*, vol. 2, no. 4, pp. 99–116, 2008.

[26] H. Houmani and M. Mejri, "Ensuring the correctness of cryptographic protocols with respect to secrecy," in *Proceedings of the International Conference on Security and Cryptography*

(SECRYPT '08), pp. 184–189, INSTICC Press, Porto, Portugal, July 2008.

[27] SetCo., "Set secure electronic transaction Specification: business description," Tech. Rep., 1997.

[28] SetCo, "Set secure electronic transaction specification: formal protocol definition," Tech. Rep., 1997.

[29] SetCo, "Set secure electronic transaction specification: programmer's guide," Tech. Rep., 1997.

[30] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications of the ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[31] R. Delicata and S. Schneider, "Temporal rank functions for forward secrecy," in *Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW '05)*, pp. 126–139, Washington, DC, USA, June 2005.

[32] S. Schneider, "Verifying authentication protocols in CSP," *IEEE Transactions on Software Engineering*, vol. 24, no. 9, pp. 741–758, 1998.

[33] M. Abadi, "Secrecy by typing in security protocols," *Journal of the ACM*, vol. 46, no. 5, pp. 749–786, 1999.

[34] H. Houmani and M. Mejri, "Secrecy by interpretation functions," *Knowledge-Based Systems*, vol. 20, no. 7, pp. 617–635, 2007.

[35] B. Dutertre and S. Schneider, "Using a pvs embedding of csp to verify authentication protocols," in *Proceedings of the Theorem Proving in Higher Order Logics (TPHOL's '97)*, pp. 121–136, Springer, 1997.

[36] D. E. Bell and L. J. La Padula, *Secure Computer Systems: Mathematical Foundations*, vol. I, The MITRE Corporation, 1973.

[37] T. Y. C. Woo and S. S. Lam, "A lesson on authentication protocol design," *Operating Systems Review*, pp. 24–37, 1994.

[38] M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi, "From protocol specifications to flaws and attack scenarios: an automatic and formal algorithm," in *Proceedings of the 2nd International Workshop on Enterprise Security*, Massachusetts Institute of Technology (MIT), IEEE Press, Cambridge, Mass, USA, June 1997.

[39] M. Debbabi, N. A. Durgin, M. Mejri, and J. C. Mitchell, "Security by typing," *International Journal on Software Tools for Technology Transfer*, vol. 4, no. 4, pp. 472–495, 2003.

[40] H. Comon-Lundh and R. Treinen, "Easy intruder deductions," in *Verification: Theory and Practice*, vol. 2772/2004 of *Lecture Notes in Computer Science*, pp. 182–184, Springer, Berlin, Germany, 2004.

[41] S. Delaune, "Easy intruder deduction problems with homomorphisms," *Information Processing Letters*, vol. 97, no. 6, pp. 213–218, 2006.

[42] P. Lafourcade, D. Lugiez, and R. Treinen, "Intruder deduction for the equational theory of Abelian groups with distributive encryption," *Information and Computation*, vol. 205, no. 4, pp. 581–623, 2007.

[43] G. Bella, F. Massacci, and L. C. Paulson, "Verifying the SET registration protocols," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, pp. 77–87, 2003.

[44] L. C. Paulson, "Verifying the SET protocol: overview," in *Proceedings of the International Conference on Formal Aspects of Security (FASec '03)*, vol. 2629 of *Lecture Notes in Computer Science*, pp. 4–14, 2003.

[45] S. Brlek, S. Hamadou, and J. Mullins, "A flaw in the electronic commerce protocol SET," *Information Processing Letters*, vol. 97, no. 3, pp. 104–108, 2006.

[46] S. Schneider, "An operational semantics for timed CSP," in *Proceedings of the Chalmers Workshop on Concurrency*, Report PMGR63, pp. 428–456, Chalmers University of Technology and University of Göteborg, 1992.

[47] J. W. Bryans and S. Schneider, "Mechanical verification of the full needhamschroeder public key protocol," Tech. Rep. CSD-TR-97-11, University of London, 1997.