Completing the Picture: Soundness of Formal Encryption in the Presence of Active Adversaries

(Extended Abstract)

Romain Janvier, Yassine Lakhnech, and Laurent Mazaré

VERIMAG - 2, av. de Vignates, 38610 Gières - France
{romain.janvier, yassine.lakhnech, laurent.mazare}@imag.fr

Abstract. In this paper, we extend previous results relating the Dolev-Yao model and the computational model. We add the possibility to exchange keys and consider cryptographic primitives such as signature. This work can be applied to check protocols in the computational model by using automatic verification tools in the formal model.

To obtain this result, we introduce a precise definition for security criteria which leads to a nice reduction theorem. The reduction theorem is of interest on its own as it seems to be a powerful tool for proving equivalences between security criteria. Also, the proof of this theorem uses original ideas that seem to be applicable in other situations.

Note: An extended version of this paper appears as technical report [17].

1 Introduction

There are two approaches to the verification of cryptographic protocols. The so-called *formal approach* [1] that originates from the work of Dolev and Yao and was first described in [8]. The distinguishing feature of this approach is the perfect cryptography hypothesis that essentially postulates that an intruder can only gain information from an encoded message if he knows the inverse key. The other hypothesis is that fresh nonce creation is perfect. Even under these assumptions flaws have been found in protocols that were believed to be secure. Several automatic tools (whether complete in the case of bounded protocols or incomplete in the case of unbounded ones) have been developed (e.g., [18, 6, 11, 4]). In the second approach, encryption schemes are studied using a computational model based on Turing machines. In this context, there is no idealization made concerning the cryptographic schemes: cryptographic functions operate on strings, attackers are Turing machines and correctness is defined in terms of high complexity and weak probability of success [9, 3]. This computational approach is

[1] Formal is not used here in the sense of rigorous but denotes the use of formal methods.

recognized as more realistic than the formal approach. However, its complexity makes it very difficult to develop (semi-)automatic verification methods.

Therefore, a major research goal is to relate both approaches such that a protocol that is verified within the formal approach is guaranteed to be correct in the computational (that is without making the perfect cryptography hypothesis). This research has been initiated by the work of Abadi and Rogaway [2] and a later work [1] where it has been proved that a notion of indistinguishability in the formal model is valid in the computational model. This work has been pushed further in [19] and then in [15] where an active intruder is considered. This last paper proves that if the encryption scheme verifies a certain property (called IND-CCA), then security in the formal model implies security in the computational model. The important part of this work is that it applies to active adversaries. Other related works include Backes, Pflizmann and Waidner [14] where the formal model is not exactly the Dolev-Yao model, although very close. It is not clear whether protocols can be checked automatically in this formalism. Also, the cryptographic primitives are modeled at a rather detailed level in the computational model. P. Laud [12] proves safety of the formal model for symmetric encryption. In particular, he deals with encryption cycles.

Our objective in this paper is to continue this work and weaken some of the restrictions imposed on protocols in previous works. The main restriction in [15] is that secret keys cannot be part of sent messages and that message forwarding is not allowed. To weaken these restrictions, we first give a general definition of a security criterion (like IND-CCA). These criteria can be seen as a game that an intruder should not be able to win. Our first result is a reduction theorem that proves the equivalence between a criterion and simpler criteria. This allows us to prove that the IND-CCA criterion is equivalent to quite richer and useful criteria. Definition of criteria is an important part as they make it possible to release some of the restrictions over protocols made by previous works. These criteria are equivalent to IND-CCA, a well studied notion in provable cryptography. Finally, we use these criteria in order to prove that Dolev-Yao constitutes a safe abstraction of the computational model even for protocols involving both asymmetric encoding and digital signature.

2 Preliminaries

An *asymmetric encryption scheme* $\mathcal{AE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ is defined by three algorithms. The key generation algorithm $\mathcal{KG}$ is a randomized function which given a security parameter η outputs a pair of keys (pk, sk), where pk is a public key and sk the associated secret key. The encryption algorithm $\mathcal{E}$ is also a randomized function which given a message and a public key outputs the encryption of the message by the public key. Finally the decryption algorithm $\mathcal{D}$ takes as input a cyphertext and a secret key and outputs the corresponding plain-text, i.e., $\mathcal{D}(\mathcal{E}(m, pk), sk) = m$. The execution time of the three algorithms is assumed polynomially bounded by η.

A *signature scheme* $\mathcal{SS} = (\mathcal{KG}, \mathcal{S}, \mathcal{V})$ is also defined by three algorithms. The key generation algorithm randomly generates pairs of keys (sik, vk), where sik is the signature key and vk is the verification key. The signature algorithm $\mathcal{S}$ randomly produces a signature of a given message by a given signature key. The verification algorithm $\mathcal{V}$ is given a message m, a signature σ and a verification key vk and tests if σ is a signature of m with the signature key corresponding to vk. Hence, $\mathcal{V}(m, \mathcal{S}(m, sik), vk)$ returns true for any message m and any pair of keys (sik, vk) generated by $\mathcal{KG}$. In this case, we still assume that the algorithms have a polynomial complexity.

An adversary for a given scheme is a Polynomial Random Turing Machine (PRTM) which has access to a set of oracles. These oracles depend on the scheme and are given in the different cases thereafter. In the following, we consider Turing machines which execution is polynomially bounded in the security parameter η, i.e. for any input corresponding to security parameter η, the machine stops within $P(\eta)$ steps for some polynomial P.

To model access to oracles, we slightly modify the definition of Turing machines. Our Turing machines have two additional tapes, one for arguments (of function/oracle calls) and one for results. Then, let F be a countable set of function names. We define our PRTM as a pair of a Turing machine $\mathcal{A}$, where transitions can be function calls, and a substitution σ linking function names $\underline{f} \in F$ to functions from string of bits (arguments) to string of bits (results). These functions are also described by polynomial Turing machines (which can also access oracles). To distinguish oracles from real functions (which can be their implementations), function names are always underlined when considering access to an oracle. The semantics of $\mathcal{A}/\sigma$ are the standard semantics of $\mathcal{A}$ except that whenever $\mathcal{A}$ fires a transition labeled by a function call $\underline{f}$, the content of the results tape becomes $f\sigma(args)$, where $args$ is the value of the arguments tape.

To simplify notations, we write directly $\mathcal{A}/f_1, ..., f_n$ where f_i are functions. Thus, we omit the name of the function as soon as this name is not relevant for comprehension and functions are directly called using the $\underline{f_i}$ notation when defining $\mathcal{A}$.

A function $h : \mathbb{R} \to \mathbb{R}$ is *negligible*, if it is ultimately bounded by x^{-c}, for each positive $c \in \mathrm{N}$, i.e., for all $c > 0$ there exists N_c such that $|h(x)| < x^{-c}$, for all $x > N_c$.

The definition of messages and of the intruder in the formal model is by now standard, e.g. [8, 18].

3 A Generic Reduction Theorem

In [15], protocols allowing sending of secret keys are not considered because it is not possible in IND-CCA to encode secret keys. To solve that, we introduce a new criterion N-PAT-IND-CCA and prove it equivalent to IND-CCA. A similar result is needed to introduce signature.

A security criterion γ is defined by an experiment that involves an adversary and two ways W_0 and W_1 of implementing a set of oracles. The adversary is aware of both implementations and is allowed to call the oracles but does not know *a priori* which implementation is really used. The challenge consists in guessing which implementation is used. More precisely, an adversary is a probabilistic polynomial time Turing machine (PRTM) that has access to a set of oracles (either W_0 or W_1). The adversary's *advantage* is the probability that the adversary outputs 1 when the set of oracles is W_1 minus the probability that the adversary outputs 1 when the set of oracles is W_0. An encryption scheme is said γ-secure, if the advantage of any adversary is negligible.

In this section, we present a generic result allowing us to prove that a security criterion γ_1 can be reduced to a criterion γ_2. This means that if there exists an adversary that breaks γ_2 then there exists an adversary that breaks γ_1. The proof is constructive in the sense that such an adversary for γ_1 can be effectively computed.

Given a finite set x_i where i ranges from 1 to n, $\overline{x}$ denotes the whole set of x_i. When more precision is required, this set can also be denoted by $x_{1..n}$. In this section, we give a formal definition of a criterion and show how a criterion can be partitioned in a safe way. The theorem presented here allows us to verify that a criterion is equivalent to another one by using such partitions. This result is applied in the following sections to show an equivalence between a few security criteria.

3.1 Security Criterion

A criterion γ is a collection formed by:

- A bit b, this bit is the challenge that has to be guessed by the adversary.
- A finite number of parameters c_1 to c_{na}. These parameters are shared by the oracles and most of the time, they are chosen randomly at the beginning of the experiment. Θ is the PRTM producing these parameters (usually a key generation algorithm).
- A finite number of oracles f_1 to f_{nb} that depend on their argument, $\overline{c}$ and b. For each f_i, there exist f_i^α and f_i^β such that the corresponding oracles when given argument (l, r) produce $f_i^\beta(f_i^\alpha(l, \overline{c}), \overline{c})$ when $b = 0$ and $f_i^\beta(f_i^\alpha(r, \overline{c}), \overline{c})$ when $b = 1$.
- A finite number of oracles g_1 to g_{nc} that depend on their argument and $\overline{c}$. The corresponding oracles when given argument x produce $g_i(x, \overline{c})$.

Oracles in $\overline{g}$ do not depend on b, they cannot be used directly by the adversary to gain information on b but they can be useful by giving information on the shared parameters $\overline{c}$ that can finally allow the adversary to deduce the value of b. Oracles in $\overline{f}$ have two layers α and β, these layers are used to decompose a criterion into a partition of criteria as the α layer allows the β layers to depend on less parameters.

Example 1. Let γ be the criterion $(b, \{pk_1, sk_1, pk_2, sk_2\}, \{f_1, f_2\}, \emptyset)$. Functions f_1 and f_2 have no α layer, i.e. $f_i^\alpha(x, ...) = x$ and $f_i(m_0, m_1)$ corresponds to the encryption of message m_b using key pk_i. Thus γ corresponds to 2-IND-CPA as introduced in [5]: the adversary has to guess the value of bit b by using only two oracles that encrypt the left or the right message according to b. The 2-IND-CCA criterion can be obtained by adding two oracles g_1 and g_2. These oracles decrypt messages encoded respectively with key pk_1 and pk_2 assuming that these messages have not been produced by oracle f_1 or f_2. □

The advantage of a PRTM $\mathcal{A}$ against γ is

$$\mathbf{Adv}_{\mathcal{A}}^{\gamma}(\eta) = Pr[\mathbf{Exp}_{\mathcal{A}}^{\gamma}(\eta, 1) = 1] - Pr[\mathbf{Exp}_{\mathcal{A}}^{\gamma}(\eta, 0) = 1]$$

Where *Exp* is the Turing machine defined by:

```
Experiment $\mathbf{Exp}_{\mathcal{A}}^{\gamma}(\eta, b)$:
    $\overline{c} \overset{R}{\leftarrow} \Theta(\eta)$
    if $b = 0$ then
        $f_i \leftarrow \lambda(l, r).f_i^{\beta}(f_i^{\alpha}(l, \overline{c}), \overline{c})$    for $i$ in $1...nb$
    else
        $f_i \leftarrow \lambda(l, r).f_i^{\beta}(f_i^{\alpha}(r, \overline{c}), \overline{c})$    for $i$ in $1...nb$
    $d \overset{R}{\leftarrow} \mathcal{A}/\eta, \overline{f}, \overline{g}$
    return $d$
```

$\mathcal{A}$ has access to an oracle giving η and to the oracles $\overline{f}$ and $\overline{g}$ as defined above. Oracles in $\overline{f}$ depend on b, this dependence is explicited by "creating" oracles f according to the value of b.

The advantage of $\mathcal{A}$ is the probability to answer 1 when the value of b is 1 minus the probability to answer 1 when the value of b is 0. Thus, if we consider a machine $\mathcal{A}$ that always outputs the same result, its advantage is 0, this also holds when considering a machine that gives a random output. Advantages are between -1 and 1, however, if $\mathcal{A}$ has a negative advantage, it is easy to build a PRTM $\mathcal{B}$ that has the opposite of $\mathcal{A}$'s advantage (we simply need to run $\mathcal{A}$ and to return the inverse of its output).

3.2 Criterion Partition and the Reduction Theorem

Example 2. Let us consider the 2-IND-CPA criterion γ defined before. Then, we say that $\gamma' = (b, \{pk_1, sk_1\}, \{f_1\}, \emptyset)$ and $\gamma'' = (b, \{pk_2, sk_2\}, \{f_2\}, \emptyset)$ constitutes a valid partition of γ when both criteria are valid (i.e. f_1 and f_2 are in the same criterion as their respective parameters pk_1 and pk_2). γ' and γ'' correspond to the IND-CPA criterion (only one oracle is available). By the reduction theorem, if an encryption scheme is IND-CPA secure (advantage of any PRTM against γ' and γ'' is negligible), then it is 2-IND-CPA secure. □

A pair of criteria γ', γ'' defines a *valid partition* of γ if there exist na', nb' and nc' such that

- $\gamma' = \big(b, c_{1..na'}, f_{1..nb'}^{\beta}, g_{1..nc'}\big)$
- $\gamma'' = \big(b, c_{(na'+1)..na}, f_{(nb'+1)..nb}, g_{(nc'+1)..nc}\big)$

- For $i \leq nb'$, f_i^α only depends on $c_{(na'+1)..na}$.
- For $i \leq nb'$, f_i^β only depends on $c_{1..na'}$.
- For $i \leq nc'$, g_i only depends on $c_{1..na'}$.
- For $i > nb'$, f_i only depends on $c_{(na'+1)..na}$.
- For $i > nc'$, g_i only depends on $c_{(na'+1)..na}$.

The four last conditions are necessary for γ' and γ'' to remain valid: oracles from a criterion only have access to parameters generated by the same criterion. The reduction theorem states that an advantage against a criterion γ can be used to produce an advantage over criterion γ' or criterion γ''.

Theorem 1 (Reduction Theorem). *If γ', γ'' is a valid partition of γ and $\mathcal{A}$ is a PRTM, then there exist two PRTM $\mathcal{A}^o$ and $\mathcal{B}$ such that*

$$|\mathbf{Adv}_{\mathcal{A}}^{\gamma}(\eta)| \leq 2.|\mathbf{Adv}_{\mathcal{B}}^{\gamma'}(\eta)| + |\mathbf{Adv}_{\mathcal{A}^o}^{\gamma''}(\eta)|$$

Proof Idea for the Reduction Theorem. The purpose of this section is to explain the main ideas underlying the proof of the reduction theorem, the detailed proof appears in [17]. An application of this proof to a simple example is given below.

The adversary $\mathcal{A}^o$ against the criterion γ'' simulates $\mathcal{A}$. To do so, he has to answer the queries made to oracles from γ'. Since $\mathcal{A}^o$ cannot construct faithfully these oracles (as it does not have access to parameters from γ''), it returns incorrect answers to $\mathcal{A}$. Finally $\mathcal{A}^o$ uses the output of $\mathcal{A}$ to answer its own challenge. If the advantage of $\mathcal{A}^o$ is comparable to the advantage of $\mathcal{A}$, then γ can be reduced to criterion γ''.

Else the advantage of $\mathcal{A}^o$ is negligible compared to the advantage of $\mathcal{A}$, then another adversary, $\mathcal{B}$, has the same advantage as $\mathcal{A}$. The adversary $\mathcal{B}$ is playing against the criterion γ'. It generates a challenge for $\mathcal{A}$. Moreover, if $b = 1$ the answers to the queries made by $\mathcal{A}$ are correct and if $b = 0$ the answers are forged in the same way as in $\mathcal{A}^o$. When $\mathcal{A}$ answers its challenge, $\mathcal{B}$ verifies it. If it is correct, $\mathcal{B}$ supposes that $b = 1$, else it supposes that $b = 0$. Indeed, $\mathcal{A}^o$ probably has a lower advantage than $\mathcal{A}$.

Example 3. Consider our previous (IND-CPA) example. Machine $\mathcal{A}^o$ is opposed to γ'', it creates the missing key pk_1 and uses it to simulate the missing oracle: the simulation is achieved by always encoding the left argument, ${}_{fake}f_1(l, r) = \mathcal{E}(l, pk_1)$. Machine $\mathcal{B}$ is opposed to γ' with the challenge bit b'. It creates its missing key pk_2 and a random bit b. The fake oracle ${}_{fake}f_2$ uses this bit b. Oracle f_1 is also faked using b': ${}_{fake}f_1(m_0, m_1) = f_1(m_0, m_b)$. The faked oracles behave like the original oracles when $b' = 1$. They behave like the oracles faked in $\mathcal{A}^o$ when $b' = 0$. This behavior is summed up in the following array:

$oracles$	$b' = 0$	$b' = 1$
$\underline{\mathcal{E}_{pk_1}}(m_0, m_1)$	$\mathcal{E}(m_0, pk_1)$	$\mathcal{E}(m_b, pk_1)$
$\underline{\mathcal{E}_{pk_2}}(m_0, m_1)$	$\mathcal{E}(m_b, pk_2)$	$\mathcal{E}(m_b, pk_2)$

Then, if the underlying $\mathcal{A}$ machine answers b correctly, we assume that it was confronted to the right oracles and thus machine $\mathcal{B}$ answers 1, else it answers 0. The intuition behind machine $\mathcal{B}$ is that its advantage tells whether oracles from γ' are useful or can be faked without losing the advantage. □

4 Applications of the Reduction Theorem

We now introduce a new security criterion and prove it equivalent to IND-CCA by using our reduction theorem. N-PAT-IND-CCA allows the adversary to obtain the encryption of messages containing challenge secret keys, even if it does not know the value of these secret keys. For that purpose, the adversary is allowed to give pattern terms to the left-right oracles.

The pattern terms are terms where new atomic constants have been added: pattern variables. These variables denote the different challenge secret keys ($[i]$ asks the oracle to replace it with the value of sk_i). Variables can be used as atomic messages (data pattern). When a left-right oracle is given a pattern term, it replaces patterns by values of corresponding keys and encodes the message. More formally, patterns are given by the following grammar where bs is a bit-string and i is an integer.

$$pat ::= \langle pat, pat\rangle | \{pat\}_{bs} | bs | [i]$$

The computation (valuation) made by the oracle is easily defined recursively in a context giving the bit-string values for the different keys. Its result is a bit-string and it uses the encryption algorithm $\mathcal{E}$ and the concatenation denoted by operator $\cdot$.

$$\begin{array}{ll} v(bs, \overline{pk, sk}) = bs & v(\{p\}_{bs}, \overline{pk, sk}) = \mathcal{E}(v(p, \overline{pk, sk}), bs)) \\ v([i], \overline{pk, sk}) = sk_i & v(\langle p_1, p_2\rangle, \overline{pk, sk}) = v(p_1, \overline{pk, sk}).v(p_2, \overline{pk, sk}) \end{array}$$

There is yet a restriction: we exclude encryption cycles. Hence keys are ordered and a pattern $[i]$ can only be encrypted under pk_j if $i > j$. References concerning this restriction appear in [2].

The related criterion is γ_N where $\overline{c}$ is a list containing N pairs of keys (pk_i, sk_i). Oracles in $\overline{f}$ are the encryption oracles. They behave like the oracles defined in the previous example except that they perform the replacement of pattern variables with key values. The v operation is performed in the α layer whereas the β layer corresponds to the previous oracles (i.e. simple encoding). Formally, $f_i^\alpha(x) = v(x, \overline{c})$ and $f_i^\beta(x) = \mathcal{E}(x, sk_i)$. Oracles in $\overline{g}$ decrypt a message using secret keys as long as their argument has not been produced by an oracle in $\overline{f}$.

An asymmetric encryption scheme $\mathcal{AE}$ is said to be N-PAT-IND-CCA iff for any adversary $\mathcal{A}$, $\mathbf{Adv}^{\gamma_N}_{\mathcal{AE},\mathcal{A}}(\eta)$ is negligible. Note that 1-PAT-IND-CCA corresponds to IND-CCA.

Lemma 1. *Let $\mathcal{AE}$ be an asymmetric encryption scheme. If $\mathcal{AE}$ is N-PAT-IND-CCA, then $\mathcal{AE}$ is also IND-CCA.*

Proposition 1. *Let $\mathcal{AE}$ be an asymmetric encryption scheme. If $\mathcal{AE}$ is N-PAT-IND-CCA, then $\mathcal{AE}$ is also (N+1)-PAT-IND-CCA.*

Proof. We have $c_i = (pk_i, sk_i)$. Then let γ' and γ'' be the partitions obtained with $na' = nb' = nc' = 1$; f_1^β and g_1^β only need c_1; f_1^α only needs $c_{2..(N+1)}$, this would not hold if we release the acyclicity hypothesis. Finally, f_i and g_i with $i \geq 2$ only need $c_{2..(N+1)}$ and so this is a valid partition. Hence, criterion γ_{N+1} has a valid partition constituted by γ_1 and γ_N.

The reduction theorem applies and gives:

$$|\mathbf{Adv}_{\mathcal{A}}^{\gamma_{N+1}}(\eta)| \leq 2.|\mathbf{Adv}_{\mathcal{B}}^{\gamma_1}(\eta)| + |\mathbf{Adv}_{\mathcal{A}^o}^{\gamma_N}(\eta)|$$

By hypothesis, $\mathcal{AE}$ is N-PAT-IND-CCA (hence IND-CCA). Then advantages of $\mathcal{B}$ and $\mathcal{A}^o$ are negligible and we can conclude that the advantage of $\mathcal{A}$ is negligible too. □

Corollary 1. *For any N, $\mathcal{AE}$ is IND-CCA if and only if $\mathcal{AE}$ is also N-PAT-IND-CCA.*

This result tells us that if an encryption scheme is IND-CCA secure, then it is still secure when adding the possibility to ask for encryption of patterns instead of just encryption of messages.

4.1 Signature

In order to extend previous results to the case of protocols using signature, we present here a new definition of security for signature scheme, UNF-CCA, which is an adaptation of Selective (Un)Forgery Against Adaptive Chosen Message Attack [10].

The main requirement is that an adversary should not be able to forge a pair containing a message m and the signature of m using the secret signature key. An N-UNF-CCA adversary $\mathcal{A}$ is given N verification keys and has to produce a message and its signature under one of the keys. It has access to the security parameter η, N verification keys vk_i and N signature oracles $\mathcal{S}_{sik_i}(.)$. The experiment outputs bit 1 if $\mathcal{A}$ managed to produce a compromising pair $(m, \{m\}_{sik_i})$ which right part is not the result of a call to a signature oracle. Otherwise, the experiment outputs bit 0. Formally the experiment is detailed below.

```
Experiment $\mathbf{Exp}_{\mathcal{SS},\mathcal{A}}^{N-UNF}(\eta)$:
    for $i = 1$ to $N$ do
        $(sik_i, vk_i) \overset{R}{\leftarrow} \mathcal{KG}(\eta)$
    $(m, \sigma) \overset{R}{\leftarrow} \mathcal{A}/\eta, vk_1, ..., vk_N,$
                  $\mathcal{S}_{sik_1}(.), ..., \mathcal{S}_{sik_N}(.),$
    if $\sigma$ is a valid signature of $m$ under one of the $sik_i$
            not produced by $\mathcal{S}_{sik_i}(.)$
        return 1
    else return 0
```

The advantage of adversary $\mathcal{A}$ in winning the UNF-CCA challenge is defined as:

$$\mathbf{Adv}^{N-UNF}_{\mathcal{SS},\mathcal{A}}(\eta) = Pr[\mathbf{Exp}^{N-UNF}_{\mathcal{SS},\mathcal{A}}(\eta) = 1]$$

A signature scheme $\mathcal{SS}$ is said to be N-UNF-CCA iff for any adversary $\mathcal{A}$, $\mathbf{Adv}^{N-UNF}_{\mathcal{SS},\mathcal{A}}(\eta)$ is negligible. Instead of 1-UNF-CCA, we write UNF-CCA.

As the challenge is not anymore guessing the value of a bit b, our reduction theorem cannot apply directly. However, by modifying the proof scheme given above, it is possible to deduce the following property relating the UNF-CCA and N-UNF-CCA criteria. The proof is given in [17].

Proposition 2. *For any signature scheme $\mathcal{SS}$, if $\mathcal{SS}$ is UNF-CCA, then it is also N-UNF-CCA.*

4.2 N-PAT-UNF-IND-CCA

To be able to deal with protocols using both an encryption scheme and a signature scheme, we define a new criterion N-PAT-UNF-CCA, a combination of N-PAT-IND-CCA and N-UNF-CCA. Let us consider an asymmetric encryption scheme $\mathcal{AE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ and a signature scheme $\mathcal{SS} = (\mathcal{KG}', \mathcal{S}, \mathcal{V})$. We use two types of adversary: those who try to find the secret bit b used in the N left-right pattern encryption oracles and those who try to produce a message and its signature under one of the N challenge signature keys. Each of these corresponds to an experiment, we denote by $N-PUI_1$ and $N-PUI_2$, respectively. The left-right pattern encryption oracles accept patterns of the form $[sik_i]$ where sik_i is one of the challenge signature keys. Then, the corresponding advantages are:

$$\mathbf{Adv}^{N-PUI_1}_{(\mathcal{AE},\mathcal{SS}),\mathcal{A}}(\eta) = Pr[\mathbf{Exp}^{N-PUI_1}_{(\mathcal{AE},\mathcal{SS}),\mathcal{A}}(\eta, 1) = 1] - Pr[\mathbf{Exp}^{N-PU_1}_{(\mathcal{AE},\mathcal{SS}),\mathcal{A}}(\eta, 0) = 1]$$

$$\mathbf{Adv}^{N-PUI_2}_{(\mathcal{AE},\mathcal{SS}),\mathcal{A}}(\eta) = Pr[\mathbf{Exp}^{N-PUI_2}_{(\mathcal{AE},\mathcal{SS}),\mathcal{A}}(\eta) = 1]$$

A couple $(\mathcal{AE}, \mathcal{SS})$ is said to be N-PAT-UNF-IND-CCA iff for all adversary $\mathcal{A}$, $\mathbf{Adv}^{N-PUI_1}_{(\mathcal{AE},\mathcal{SS}),\mathcal{A}}(\eta)$ and $\mathbf{Adv}^{N-PUI_2}_{(\mathcal{AE},\mathcal{SS}),\mathcal{A}}(\eta)$ are negligible.

The following property states that the combination of a secure signature scheme and a secure encryption scheme is still secure. Its proof can be done using the same proof scheme as for the reduction theorem.

Proposition 3. *If $\mathcal{AE}$ is N-PAT-IND-CCA and if $\mathcal{SS}$ is N-UNF-CCA, $(\mathcal{AE}, \mathcal{SS})$ is N-PAT-UNF-IND-CCA.*

To sum up, we proved the following equivalences between criteria:

$$\mathrm{IND-CCA} \Leftrightarrow \mathrm{N-PAT-IND-CCA}$$
$$\mathrm{UNF-CCA} \Leftrightarrow \mathrm{N-UNF-CCA}$$
$$(\mathrm{IND-CCA}, \mathrm{UNF-CCA}) \Leftrightarrow \mathrm{N-PAT-UNF-IND-CCA}$$

5 Dolev-Yao Is a Safe Abstraction

In this section, we give a precise formalization of the link between the two commonly used approaches for verification of cryptographic protocols, i.e. the computational approach and the formal approach. For that purpose, we first define cryptographic protocols, then we relate traces from both models. This relation is used to prove the main theorem.

5.1 Description of Cryptographic Protocols

A multi-party protocol is defined by a list of triples (m_1, m_2, R), called actions. The action (m_1, m_2, R) means that an agent playing role R sends a message m_2 after receiving a message m_1. It is possible to replace m_1 with an empty message denoted by "-" for the first action of the protocol. For the last action, the same thing can be done with m_2. A role R represents a program that is executed by an agent Ag during a session S. In session S, we say that agent Ag impersonates role R. Let us consider the Needham-Schroeder-Lowe protocol (NSL introduced in [13]) there are two roles: the initiator and the receiver.

For a session S, an agent Ag has some local variables used during the execution of his role: his local copy of variable Var is denoted by $Ag.S.Var$. Each agent has five variables for each of his sessions: $Init, Rec, Na, Nb, Pc$. The list of actions is:

- $(-, \{\langle Init, Na\rangle\}_{Pk_{Rec}}, Init)$
- $(\{\langle init, na\rangle\}_{Pk_{Rec}}, \{\langle Rec, na, Nb\rangle\}_{Pk_{init}}, Rec)$
- $(\{\langle Rec, Na, nb\rangle\}_{Pk_{Init}}, (\{nb\}_{Pk_{Rec}}, Init)$
- $((\{Nb\}_{Pk_{Rec}}, -, Rec)$

We make a distinction between values known before an action and values received during the action: values already known are denoted using a capital for their first letter. Let Ag be an agent impersonating the initiator. In the first action, Ag chooses B as a receiver, and the value of nonce Na for session s. He sets variables $Ag.s.Init$ to Ag, $Ag.s.Rec$ to B and $Ag.s.Na$ with the chosen value. Then he sends message $\{\langle Ag.s.Init, Ag.s.Na\rangle\}_{Ag.s.Rec}$. In the second action, B receives a message encrypted with his public key. He decrypts it and parses the plain-text as a pair $\langle init, na\rangle$. After that, he chooses a new session number s' and sets $B.s'.Init$ to $init$, $B.s'.Rec$ to B, $B.s'.Na$ to na and finally chooses a fresh value for $B.s'.Nb$. Finally, he sends message $\{\langle B.s'.Rec, B.s'.Na, B.s'.Nb\rangle\}_{Pk_{B.s'.Init}}$ to Ag. The remaining actions have similar semantics.

Hypothesis over Protocols. The following restrictions are made over the protocol Π considered in this section. Π has to be executable, that is each role can be run by a PRTM. In the formal world, for any execution, secret keys of honest agents remain secrets, there is no encryption cycle, there is a nonce in each signed message that also remains secret. Moreover, we ask that messages include enough typing information to allow parsing of received messages. We also assume that any agent knows identities of all the other agents.

Computational and Formal Models. For both models, agents involved in a protocol send their messages through a network. This network is modeled by the Adversary. The Adversary intercepts any message sent by an agent. He can forge new messages using his knowledge and send these messages to agents usurping an agent's identity.

In the formal model, the Adversary is a classical Dolev-Yao Intruder [8]. In the computational model, both the Adversary and the implementation of the protocol are PRTM, denoted by $\mathcal{A}_c$ and Π_c. Π_c is used as an oracle by $\mathcal{A}_c$. Messages are bit-strings, new nonces generated by agents are random bit-strings. Keys used by agents are generated using $\mathcal{KG}$. Encryptions and decryptions are obtained using algorithms from $\mathcal{AE}$. Signatures related functions use $\mathcal{SS}$.

$\mathcal{A}_c$ can create new valid identities and thus impersonate some dishonest agents.

5.2 Non Dolev-Yao Traces

A trace is a list of tuples (m_1, m_2, Ag, s) called transitions where m_1 is a message sent by the Adversary to agent Ag for session s and m_2 is the answer from Ag. As before, message m_1 and m_2 can be "-". Assignment $t_c \leftarrow \mathcal{A}_c/\eta, \Pi_c$ denotes that the trace t_c is obtained by the computational Adversary $\mathcal{A}_c$ confronted to Π_c . We assume that only messages accepted by an agent appear in the trace. We now transform a computational trace into a *pseudo formal trace*. The resulting trace is only pseudo formal because even if messages are expressed using Dolev-Yao terms, this does not imply that there exists a formal Adversary producing this trace. The transformation given here can be seen as verification of the trace by all the honest agents working together. Their goal is to check if the adversary performed an action that is not in the Dolev-Yao model. To achieve this, messages in the computational trace (which are bit-strings) are replaced with Dolev-Yao terms using the following:

- Bit-strings corresponding to identities are associated to fresh atoms: H_1, $H_2, ...$ for honest agents and $I_1, I_2, ...$ for dishonest agents.
- Bit-strings corresponding to long-term keys are associated to fresh keys: $Pk_{H_1}, Pk_{H_2}, ...$ and $Pk_{I_1}, Pk_{I_2}, ...$ for public keys, Sk_{Id} for associated secret keys.
- Bit-strings corresponding to nonces N generated in session S by an honest agent are associated to fresh atoms N^s.
- Bit-strings corresponding to fresh public or secret keys generated by an honest agent in session s are associated with fresh keys Pk^s and Sk^s.
- Bit-strings corresponding to keys generated by the Adversary are associated with fresh keys Pk_I^j and Sk_I^j.
- Bit-strings corresponding to concatenation of a message associated to term T_1 with a message associated to term T_2 are associated with $\langle T_1, T_2 \rangle$.
- Bit-strings corresponding to encryption of messages associated to term T with a key associated to term K are associated to term $\{T\}_K$.

Note that this is not complete as we have not yet taken into account nonces generated by the Adversary. As the Adversary is a PRTM, whenever he has to send a new nonce, he does not have to generate it randomly: he can send composed messages instead of nonces or perform operations over bit-strings (XOR, changing bit order, adding or removing bits...). Hence for the honest agents it is impossible to guess how the Adversary has chosen his nonces. This is why when transforming a bit-string corresponding to a message where an honest agent receives a new nonce, we only test if the corresponding bit-string is an already known nonce. In this case, the bit-string is associated to the nonce term. Else, it is associated to a fresh variable X_i. If later this bit-string is parsed as something else (tuple, encoding), variable X_i is replaced by the appropriate term. The same thing is done when an honest agent receives a message encrypted with a key which inverse is not known or a signature impossible to verify (at reception time). When every message in the trace has been transformed, each remaining X_j is replaced by the fresh atom N_I^j, i.e. remaining variables are considered as fresh nonces. The pseudo-formal trace corresponding to computational trace t_c is denoted by $\alpha(t_c)$.

Definition 1 (Non Dolev-Yao Traces). *A formal trace t_f is said* Non Dolev-Yao *(NDY) iff there exists a message sent by the Adversary which cannot be deduced from previous messages using Dolev-Yao's deduction, this message is called a NDY message. A computational trace t_c is said NDY iff $\alpha(t_c)$ is NDY.*

5.3 A Computational Trace Is Certainly a Dolev-Yao Trace

In this section, we prove that if the encryption and signature schemes verify IND-CCA resp. UNF-CCA and if the number of possible nonces is exponential in η, then the probability that a computational trace is NDY is negligible. This means that the computational Adversary, even with all the computing power of PRTM, cannot have a behavior not represented by a formal adversary.

Theorem 2. *Let Π be a protocol. Let $\mathcal{AE}$ be the encryption scheme and $\mathcal{SS}$ the signature scheme used in Π_c. If $\mathcal{AE}$ is IND-CCA and $\mathcal{SS}$ is UNF-CCA then for any concrete Adversary $\mathcal{A}_c$:*
$Pr\big[t_c \leftarrow \mathcal{A}_c/\eta, \Pi_c \ ; \ t_c \ NDY\big]$ is negligible.

The proof of this theorem can be found in [17].

5.4 Formal and Computational Properties

Let P_c be a property in the computational world represented by a predicate over computational traces. A protocol Π verifies P_c (denoted by $\Pi \models_c P_c$) iff for any Adversary $\mathcal{A}_c$, $Pr\big[t_c \leftarrow \mathcal{A}_c/\eta, \Pi_c \ ; \ \neg P_c(t_c)\big]$ is negligible. A property in the formal world is represented by a predicate P_f over formal traces. Hence, a protocol Π verifies P_f (denoted by $\Pi \models_f P_f$) iff any trace produced by a Dolev-Yao adversary against Π verifies P_f.

Using theorem 2, we prove the following result which states that proving formally P_f allows us to deduce P_c.

Theorem 3. *Let P_f and P_c be a formal and a computational property such that*

$$\forall t_c, \forall t_f, \big(P_f(t_f) \wedge \alpha(t_c) = t_f\big) \Rightarrow P_c(t_c)$$

If $\mathcal{AE}$ is IND-CCA and $\mathcal{SS}$ is UNF-CCA, then

$$\Pi \models_f P_f \Rightarrow \Pi \models_c P_c$$

This theorem states that if the formal property correctly under-approximates the computational property then the formal abstraction is correct.

This theorem has been applied to mutual authentification in [15] and holds for nonce secrecy [7].

6 Conclusion

In this paper, we considered active intruders. Our main result is that an adversary behavior follows the formal model with overwhelming probability, if the encryption scheme is IND-CCA and the signature scheme is UNF-CCA. This result has immediate applications as automatic verification of security protocols is quite developed now and as there are encryption algorithms that verify the required properties. Our result extends previous ones and allow:

- Multi-party protocols.
- More cryptographic primitives: combination of digital signature and asymmetric encryption.
- Protocols where encoding of secret keys and message forwarding are allowed.

A second main contribution of our paper is a formal definition for security criteria and a reduction theorem. This theorem and its proof scheme seem to apply in a wide variety of cases. It allows to prove equivalences between a security criterion and some of its sub-criteria. This theorem allowed us to give a quick proof of already known results, to generalize this to new results and we believe that it could be useful whenever one wants to relate two security criteria.

Concerning extensions of this work, in [16], we extend these results to protocols using simultaneously all the classical cryptographic primitives: asymmetric and symmetric encoding, signature and hashing. This paper also deals with simple equational theories.

Acknowledgments

This work has been partially supported by the RNTL project PROUVE-03V360 and the ACI project ROSSIGNOL

References

1. M. Abadi and J. Jürgens. Formal eavesdropping and its computational interpretation. In *4th Intern. Symp. on Theoretical Aspects of Computer Software*, volume 2215 of *LNCS*. Springer-Verlag, 2001.
2. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP International Conference on Theoretical Computer Science (IFIP TCS2000)*, Sendai, Japan, 2000. Springer-Verlag, Berlin Germany.

3. Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo G. Desmedt, editor, *Proc. CRYPTO 94*, pages 341–358. Springer, 1994. Lecture Notes in Computer Science No. 839.
4. B. Blanchet. Abstracting cryptographic protocols by prolog rules. In *International Static Analysis Symposium*, volume 2126 of *LNCS*, pages 433–436, 2001.
5. A. Boldyreva, M. Bellare, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *Advances in Cryptology-EUROCRYPT 2000, Lecture Notes in Comput. Sci., Vol. 1807*, pages 259–274. Springer, 2000.
6. L. Bozga, Y. Lakhnech, and M. Périn. Hermes: An automatic tool for verification of secrecy in security protocols. In *15th International Conference on Computer Aided Verification (CAV),*, volume 2725 of *LNCS*, 2003.
7. V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. Research Report RR-5341, INRIA, October 2004.
8. D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.
9. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.
10. Shafi Goldwasser, Silvio Micali, and Ron L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
11. Jean Goubault-Larrecq. A method for automatic cryptographic protocol verification. In *International Workshop on Formal Methods for Parallel Programming: Theory and Applications*, volume 1800 of *LNCS*, 2000.
12. P. Laud. Symmetric encryption in automatic analyses for confidentiality against adaptive adversaries. In *Proc. of 2004 IEEE Symposium on Security and Privacy*, pages 71–85, 2004.
13. G. Lowe. An attack on the Needham-Schroeder public-key authentification protocol. *Information Processing Letters*, 56(3):131–133, 1995.
14. B. Pfitzmann M. Backes and M. Waidner. U universally composable cryptographic library. In ACM Press, editor, *Computer and Communication Security*, Oct. 2003.
15. D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In *Proceedings of the Theory of Cryptography Conference*, pages 133–151. Springer, 2004.
16. Y. Lakhnech R. Janvier and L. Mazaré. (de)compositions of cryptographic schemes and their applications to protocols. Technical report, Verimag, Centre Équation, 38610 Gières, To Appear 2004.
17. Y. Lakhnech et L. Mazare R. Janvier. Completing the picture: Soundness of formal encryption in the presence of active adversaries. Technical Report TR-2004-19, Verimag, Centre Équation, 38610 Gières, November 2004.
18. M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *IEEE Computer Security Foundations Workshop*, 2001.
19. B. Warinschi. A computational analysis of the needham-schroeder(-lowe) protocol. In *Proceedings of 16th Computer Science Foundation Workshop*, pages 248–262. ACM Press, 2003.