Game-Based Criterion Partition Applied to Computational Soundness of Adaptive Security

M. Daubignard, R. Janvier, Y. Lakhnech, and L. Mazaré

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

Abstract. The composition of security definitions is a subtle issue. As most security protocols use a combination of security primitives, it is important to have general results that allow to combine such definitions. We present here a general result of composition for security criteria (i.e. security requirements). This result can be applied to deduce security of a criterion from security of one of its sub-criterion and an indistinguishability criterion. To illustrate our result, we introduce joint security for asymmetric and symmetric cryptography and prove that it is equivalent to classical security assumptions for both the asymmetric and symmetric encryption schemes. Using this, we give a modular proof of computational soundness of symbolic encryption. This result holds in the case of an adaptive adversary which can use both asymmetric and symmetric encryption.

Keywords: Provable Security, Security Games, Probabilistic Encryption, Computational Soundness of Formal Methods.

1 Introduction

Provable security consists in stating the expected security properties in a formally defined adversarial model and providing a mathematical proof that the properties are satisfied by the designed system/protocol. Micali and Goldwasser are probably the first to put forward the idea that security can be proved in a formally defined model under well-believed rigorously defined complexity-assumptions [GM84]. Although provable security has by now become a very active research field there is a lack of a general "proof theory" for cryptographic systems. As underlined by V. Shoup in [Sho04], security proofs often *become so messy, complicated, and subtle as to be nearly impossible to understand*. Ideally there should be a verification theory for cryptographic systems in the same way as there are verification theories for "usual" sequential and concurrent systems (cf. [Cou90, MP92]).

As security proofs are mostly *proofs by reduction* a promising approach seems to be one that is based on transforming the system to be verified into a system that obviously satisfies the required properties. Sequences of games have

T. Dimitrakos et al. (Eds.): FAST 2006, LNCS 4691, pp. 47–64, 2007.

been recently proposed as a tool for taming the complexity of security proofs [Sho04, BR04] and first implementations of tools that assisted in deriving such sequences have been developed [Bla06]. In particular, three types of transitions between games are proposed. One of the most powerful transitions is based on *indistinguishability.* Informally, to bound the probability of an event E_i in game i and the probability of event E_{i+1} in game $i+1$, one shows that there is a *distinguisher algorithm* D that interpolates between Game i and Game $i+1$, such that given an element from distribution P_i, for $i = 1, 2$, D outputs 1 with probability $Pr[E_i]$. Hence, $Pr[E_i] - Pr[E_{i+1}] = Pr[D(x) \to 1 | x \in P_1] - Pr[D(x) \to 1 | x \in P_2]$, and hence, the indistinguishability assumption implies that $Pr[E_i] - Pr[E_{i+1}]$ is negligible.

In this paper we prove a theorem that provides a powerful instance of the indistinguishability-based transition technique. This theorem can be used for compositional verification of cryptographic libraries as it allows one to reduce a security criterion into simpler ones. A typical use is to allow the comparison of a criterion that involves a set of oracles (which can for example all use the same challenge bit b) with a criterion that only involves a subset of the oracles. As a simple application of this result, we can for instance prove the equivalence of semantic security of one key and semantic security in the multi-party setting [BBM00]. The advantage of applying our theorem in that case is that the proof is done without having to design adversaries, the only thing to do is to provide a partition of the criterion.

Moreover we believe that our main result is helpful when proving computational soundness of symbolic analysis for cryptographic protocols. This recent trend in bridging the gap that separates the computational and symbolic views of protocols has been initiated by Abadi and Rogaway [AR00]. In this paper, they prove that symbolic equivalence of messages implies computational indistinguishability provided that the cryptographic primitives are secure. This result has then been adapted for protocols where the adversary is an eavesdropper and has a passive behavior and the only allowed cryptographic primitive is symmetric encryption [AJ01].

Various extensions of [AR00, AJ01] have been presented recently by adding new cryptographic primitives [BCK05] or by removing the passive adversary hypothesis. There are different ways to consider non-passive adversaries, this can be done by using the simulatability approach [BPW03], by proving trace properties on protocols [MW04, CW05, JLM05]. Another possibility is to consider an adaptive adversary as introduced by Micciancio and Panjwani [MP05]. In this context, the adversary issues a sequence of adaptively chosen equivalent pairs of messages (m_0^1, m_1^1) to (m_0^q, m_1^q). After query (m_0^i, m_1^i) the adversary receives a bit-string that instantiates either m_0^i or m_1^i and it has to tell which is the case. The main improvement with respect to the result of Abadi and Rogaway [AR00] is that the adversary has an adaptive behavior: it can first send a query (m_0^1, m_1^1) then using the result determine a new query and submit it. However Micciancio and Panjwani only consider symmetric encryption. In order to illustrate how our main result can be used in such situations, we prove a similar result when considering both asymmetric and symmetric encryption. Besides by using our partition theorem, the proof we give is modular and hence easier to extend to more cryptographic primitives than the original one. For that purpose, we introduce new security criteria which define *pattern semantic security* and prove that these criteria are equivalent to classical semantic security requirements. The main interest of these criteria is to easily allow encryption of secret keys (either symmetric or private keys).

Organization. In section 2 after recalling some basic definitions, we introduce security criteria and some examples of cryptography-related criteria. A powerful way of composing security criteria is introduced and proved in section 3: the criterion partition theorem. Section 4 shows how to use this result soundly. To illustrate this we prove that some composition of asymmetric and symmetric encryption schemes can be directly stated secure by using the partition theorem. Using this last result, section 5 proves computational soundness of symbolic equivalence for an adaptive adversary using both asymmetric and symmetric encryption schemes. Eventually, section 6 draws some concluding remarks.

2 Preliminaries

2.1 Cryptographic Schemes

We first recall classical definitions for cryptographic schemes in the computational setting. In this setting, messages are bit-strings and a security parameter η is used to characterize the strength of the different schemes, for example η can denote the length of the keys used to perform an encryption.

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 cipher-text and a secret key and outputs the corresponding plain-text, i.e. $\mathcal{D}(\mathcal{E}(m, pk), sk) = m$, if key pair (pk, sk) has been generated by $\mathcal{KG}$. The execution time of the three algorithms is assumed to be polynomially bounded by η.

A *symmetric encryption scheme* $\mathcal{SE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ is also defined by three algorithms. The key generation algorithm $\mathcal{KG}$ is a randomized function which given a security parameter η outputs a key k. The encryption algorithm $\mathcal{E}$ is also a randomized function which given a message and a key outputs the encryption of the message by this key. Finally the decryption algorithm $\mathcal{D}$ takes as input a cipher-text and a key and outputs the corresponding plain-text, i.e. $\mathcal{D}(\mathcal{E}(m, k), k) = m$. The execution time of the three algorithms is also assumed polynomially bounded by η.

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

2.2 Turing Machines with Oracles

Adversaries are polynomial-time random Turing machines (PRTM) with oracles. Oracles are also implemented using PRTMs. In order to detail the oracles an adversary can query, the definition of an adversary $\mathcal{A}$ is for example:

Adversary $\mathcal{A}/\mathcal{O}_1, \mathcal{O}_2$:
  Code of $\mathcal{A}$ e.g: $s \leftarrow \mathcal{O}_1(x)$

Where the code of $\mathcal{A}$ can call two oracles using names $\mathcal{O}_1$ and $\mathcal{O}_2$. When executing this adversary $\mathcal{A}$, we use the notation $\mathcal{A}/\mathcal{B}_1, \mathcal{B}_2$ where $\mathcal{B}_1$ and $\mathcal{B}_2$ are two PRTMs to denote that names $\mathcal{O}_1$ and $\mathcal{O}_2$ are respectively implemented with oracles $\mathcal{B}_1$ and $\mathcal{B}_2$.

We use the standard λ-notation to concisely describe PRTMs obtained from others by fixing some arguments. For instance, let G be a PRTM that has two inputs. Then, we write $\lambda s.G(s, \theta)$ to describe the machine that is obtained from G by fixing the second argument to the value θ. Thus, $\mathcal{A}/\lambda s.G(s, \theta)$ denotes the machine $\mathcal{A}$ that may query an oracle obtained from G by instantiating its second argument by θ. The argument θ of G is defined in the context of $\mathcal{A}$ and may not be known by $\mathcal{A}$. So typically, $\mathcal{A}$ may be trying to compute some information on θ through successive queries.

Moreover, adversaries are often used as sub-routines in other adversaries. Consider the following description of a randomized algorithm with oracles. Here adversary $\mathcal{A}'$ uses $\mathcal{A}$ as a sub-routine. Moreover, $\mathcal{A}'$ may query oracle $\mathcal{O}_1$. On its turn $\mathcal{A}$ may query the same oracle $\mathcal{O}_1$ and additionally the oracle $\lambda s.F_2(s, \theta_2)$. The latter is obtained from F_2 by fixing the second argument to θ_2 which is generated by $\mathcal{A}'$.

Adversary $\mathcal{A}'/\mathcal{O}_1$:
  $\theta_2 \leftarrow \ldots$
  $s \leftarrow \mathcal{A}/\mathcal{O}_1,$
    $\lambda s.F_2(s, \theta_2)$

2.3 Games and Criteria

A security criterion is defined as a game involving an adversary (represented by a PRTM). The game proceeds as follows. First some parameters θ are generated randomly using a PRTM Θ. The adversary is executed and can query an oracle F which depends on θ. At the end, the adversary has to answer a bit-string whose correctness is checked by an algorithm V which also uses θ (e.g. θ includes a bit b and the adversary has to output the value of b). Thus, a criterion is given by a triple consisting of three randomized algorithms:

- Θ is a PRTM that randomly generates some challenge θ.
- F is a PRTM that takes as arguments a bit-string s and a challenge θ and outputs a new bit-string. F represents the oracles that an adversary can call to solve its challenge.
- V is a PRTM that takes as arguments a bit-string s and a challenge θ and outputs either true or false. It represents the verification made on the result computed by the adversary. The answer true (resp. false) means that the adversary solved (resp. did not solve) the challenge.

As an example let us consider an asymmetric encryption scheme $(\mathcal{KG}, \mathcal{E}, \mathcal{D})$. Semantic security against chosen plain-text attacks (IND-CPA) can be represented using a security criterion $(\Theta; F; V)$ defined as follows: Θ randomly samples the challenge bit b and generates a key pair (pk, sk) using $\mathcal{KG}$; F represents the public key oracle (this oracle returns pk) and the left-right encryption oracle (given bs_0 and bs_1 this oracle returns $\mathcal{E}(bs_b, pk)$); and V checks whether the returned bit equals b.

Note that Θ can generate several parameters and F can represent several oracles. Thus, it is possible to define criteria with multiples Θ and F. For example, a criterion with two challenge generators Θ_1 and Θ_2, two oracles F_1 and F_2 and a verifier V is denoted by $(\Theta_1, \Theta_2; F_1, F_2; V)$.

Let $\gamma = (\Theta; F; V)$. The advantage of a PRTM $\mathcal{A}$ against γ is defined as the probability that $\mathcal{A}$ has to win its game minus the probability that an adversary can get without accessing oracle F.

$$\mathbf{Adv}^{\gamma}_{\mathcal{A}}(\eta) = 2\left(Pr[\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta) = true] - PrRand^{\gamma}(\eta)\right)$$

where $\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta)$ is the Turing machine defined by:

Game $\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta)$:
$\theta \leftarrow \Theta(\eta)$
$d \leftarrow \mathcal{A}(\eta)/\lambda s.F(s, \theta)$
return $V(d, \theta)$

and $PrRand^{\gamma}(\eta)$ is the best probability to solve the challenge that an adversary can have without using oracle F. Formally, let γ' be the criterion $(\Theta; \epsilon; V)$ then $PrRand^{\gamma}(\eta)$ is defined by:

$$PrRand^{\gamma}(\eta) = \max_{\mathcal{A}}\left(Pr[\mathbf{G}^{\gamma'}_{\mathcal{A}}(\eta) = true]\right)$$

where $\mathcal{A}$ ranges over any possible PRTM. For example when considering a criterion $\gamma = (\Theta; F; V)$ where a challenge bit b is generated in Θ and V checks that the adversary guessed the value of b, then $PrRand^{\gamma}(\eta)$ equals 1/2, in particular this is the case for IND-CPA.

3 The Criterion Partition Theorem

Consider a criterion $\gamma = (\Theta_1, \Theta_2; F_1, F_2; V_1)$, composed of two challenge generators Θ_i, their related oracles F_i, and a verifier V_1. Assume that F_1 and V_1

do not depend on θ_2 (which is the part generated by Θ_2). Because of these assumptions, $\gamma_1 = (\Theta_1; F_1; V_1)$ is a valid criterion. We are going to relate the advantages against γ and γ_1. To do so, let us consider the game $\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta)$ played by an adversary $\mathcal{A}$ against γ:

Game $\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta)$:
$\theta_1 \leftarrow \Theta_1(\eta)$
$\theta_2 \leftarrow \Theta_2(\eta)$
$s \leftarrow \mathcal{A}/\lambda s.F_1(s, \theta_1),$
$\quad \lambda s.F_2(s, \theta_1, \theta_2)$
return $V_1(s, \theta_1)$

We define an adversary $\mathcal{A}'$ against γ_1 which tries to act like $\mathcal{A}$. However, $\mathcal{A}'$ does not have access to its challenge θ_1 and hence it generates a new challenge θ'_1 (using Θ_1) and uses it to answer queries made by $\mathcal{A}$ to F_2.

Adversary $\mathcal{A}'/\mathcal{O}_1$:
$\theta'_1 \leftarrow \Theta_1(\eta)$
$\theta_2 \leftarrow \Theta_2(\eta)$
$s \leftarrow \mathcal{A}/\mathcal{O}_1,$
$\quad \lambda s.F_2(s, \theta'_1, \theta_2)$
return s

The game involving $\mathcal{A}'$ against γ_1, $\mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta)$, is given by:

Game $\mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta)$:
$\theta_1 \leftarrow \Theta_1(\eta)$
$\theta'_1 \leftarrow \Theta_1(\eta)$
$\theta_2 \leftarrow \Theta_2(\eta)$
$s \leftarrow \mathcal{A}/\lambda s.F_1(s, \theta_1),$
$\quad \lambda s.F_2(s, \theta'_1, \theta_2)$
return $V_1(s, \theta_1)$

Our aim is to establish a bound on

$$|Pr[\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta) = true] - Pr[\mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta) = true]|$$

To do so, we construct an adversary $\mathcal{B}$ that tries to distinguish game $\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta)$ from game $\mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta)$, i.e. $\mathcal{B}$ tries to distinguish the case where $\mathcal{A}$ uses correlated oracles (i.e. the same θ_1 is used by F_1 and F_2) from the case where $\mathcal{A}$ uses decorrelated oracles (i.e. θ_1 is used by F_1 and a different θ'_1 is used by F_2), figure 1 gives the intuition of how $\mathcal{B}$ works: $\mathcal{B}$ either simulates $\mathcal{A}$ with correlated oracles in the upper part of the figure or $\mathcal{A}$ with decorrelated oracles. Finally, $\mathcal{B}$ uses the answer of $\mathcal{A}$ in order to win its challenge. We introduce a new indistinguishability criterion γ_2 that uses a challenge bit b, in this criterion the adversary has to guess the value of bit b. Our objective is to build a distinguisher $\mathcal{B}$ such that the following equations hold:

$$Pr[\mathbf{G}^{\gamma_2}_{\mathcal{B}} = true \mid b = 1] = Pr[\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta) = true] \tag{1}$$

$$Pr[\mathbf{G}^{\gamma_2}_{\mathcal{B}} = false \mid b = 0] = Pr[\mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta) = true] \tag{2}$$

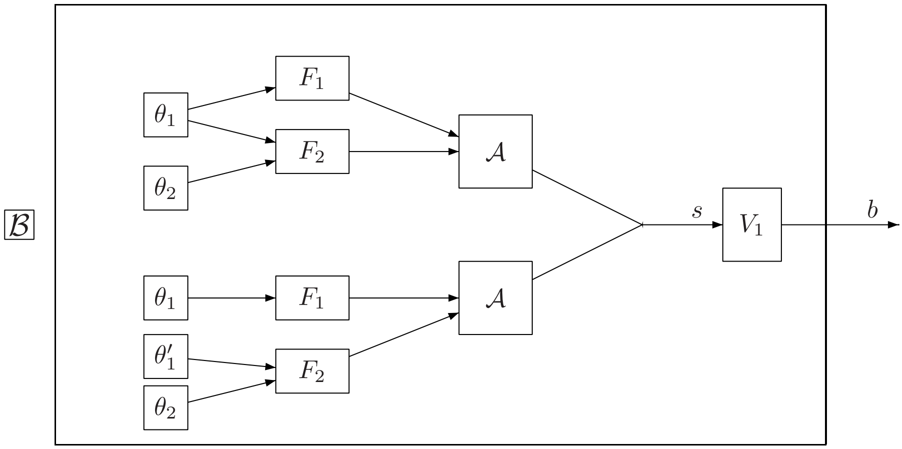

Fig. 1. Correlated and Decorrelated Oracles

Indeed, using these equations we will be able to derive the following bound:

$$|Pr[\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta) = true] - Pr[\mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta) = true]| = \mathbf{Adv}^{\gamma_2}_{\mathcal{B}}(\eta)$$

3.1 Construction of the Distinguisher

In the following, we give a methodology that tells us how to build the indistinguishability criterion γ_2 and the adversary $\mathcal{B}$. To do so, we need an assumption on the form of the second oracle F_2 from γ. This assumption is stated through the following hypothesis.

Hypothesis 1. *There exist three probabilistic random functions f, g and f' such that oracle F_2's implementation consists of two parts: $\lambda s.f(g(s,\theta_1),\theta_2)$ and $\lambda s.f'(s,\theta_2)$. The first part depends on both θ_1 and θ_2 whereas the second depends only on θ_2.*

The idea when introducing two parts for oracle F_2 is to separate the oracles contained in F_2 that really depend on both θ_1 and θ_2 (these oracles are placed in $f(g(...)))$ from the oracles that do not depend on θ_1 (placed in f'). Let us illustrate this on the IND-CPA criterion with two keys: there are one left-right encryption oracle and one public key oracle for each key. Θ_1 generates the challenge bit b and the first key pair (pk_1, sk_1), Θ_2 generates the other key pair (pk_2, sk_2). Oracle F_2 contains the left-right oracle related to pk_2 and the public key oracle that reveals pk_2. Hence f' is used to store the public key oracle whereas the left-right oracle has the form $\lambda s.f(g(s,\theta_1),\theta_2)$ where f performs an encryption using key pk_2 from θ_2 and $g((s_0,s_1),\theta_1)$ returns s_b according to the value of challenge bit b from θ_1. It is possible to split the oracles differently but this would not lead to interesting sub-criteria. In general it is always possible to perform a splitting that satisfies the previous hypothesis (for example,

f' is empty and $g(s, \theta_1)$ outputs both s and θ_1), however this can lead to some criteria against which adversaries may have a non-negligible advantage. In that situation the partition theorem cannot be used to obtain that the advantage of any adversary against the original criterion γ is negligible.

Adversary $\mathcal{B}$ plays against an indistinguishability criterion. It has access to two oracles: $\hat{\mathcal{O}}_1$ is implemented by the left-right oracle $f \circ LR^b$, where LR^b takes as argument a pair and returns either the first or the second element according to the value of bit b, i.e. $LR^b(x_0, x_1) = x_b$. Hence, we have $f \circ LR^b(s_0, s_1) = f(s_b, \theta_2)$ and $\hat{\mathcal{O}}_2$ is simply implemented by f'. Notice now that we have the following equations:

$$f \circ LR^b(g(s, \theta_1'), g(s, \theta_1)) = F_2(s, \theta_1, \theta_2), \text{ if } b = 1$$
$$f \circ LR^b(g(s, \theta_1'), g(s, \theta_1)) = F_2(s, \theta_1', \theta_2), \text{ if } b = 0$$

More formally, our γ_2 criterion is given by $\gamma_2 = (b, \Theta_2; f \circ LR^b, f'; v_b)$, where v_b just checks whether the bit returned by the adversary equals b.

We are now ready to give a distinguisher $\mathcal{B}$ such that equations (1) and (2) hold:

Adversary $\mathcal{B}/\hat{\mathcal{O}}_1, \hat{\mathcal{O}}_2$:
$\theta_1 \leftarrow \Theta_1(\eta)$
$\theta_1' \leftarrow \Theta_1(\eta)$
$s \leftarrow \mathcal{A}/\lambda s.F_1(s, \theta_1),$ // oracle F_1
$\lambda s.\hat{\mathcal{O}}_1(g(s, \theta_1'), g(s, \theta_1)),$ // part f of oracle F_2
$\hat{\mathcal{O}}_2$ // part f' of oracle F_2
$\hat{b} \leftarrow V_1(s, \theta_1)$
return $\hat{b}$

Recall that $\mathcal{A}$ may query two oracles: F_1 and F_2 while $\mathcal{B}$ may query the left-right oracle $f \circ LR^b$ and f'. Therefore, $\mathcal{B}$ uses Θ_1 to generate θ_1 and θ_1'. It is important to notice that θ_1 and θ_1' are generated independently. Then, $\mathcal{B}$ uses $\mathcal{A}$ as a sub-routine using $\lambda s.F_1(s, \theta)$ for $\mathcal{A}$'s first oracle, and the pair of functions $\lambda s.\hat{\mathcal{O}}_1(g(s, \theta_1'), g(s, \theta_1))$ and f' for F_2.

The game corresponding to $\mathcal{B}$ playing against γ_2 can now be detailed:

Game $\mathbf{G}_{\mathcal{B}}^{\gamma_2}(\eta)$:
$b \leftarrow \{0, 1\}$
$\theta_2 \leftarrow \Theta_2(\eta)$
$\hat{b} \leftarrow \mathcal{B}/\lambda s.f(LR^b(s), \theta_2),$
$\lambda s.f'(s, \theta_2)$
return $v_b(\hat{b})$

3.2 Comparing the Games

Let us now check equations (1) and (2). To do so, we first consider that b equals 1. Then game $\mathbf{G}_{\mathcal{B}}^{\gamma_2}$ can be detailed by introducing the definition of $\mathcal{B}$ within the game:

$$
\begin{array}{ll}
\textbf{Game} & \mathbf{G}^{\gamma_2}_{\mathcal{B}}(\eta)|b=1: \\
& \theta_2 \leftarrow \Theta_2(\eta) \\
& \theta_1 \leftarrow \Theta_1(\eta) \\
& \theta'_1 \leftarrow \Theta_1(\eta) \\
& s \leftarrow \mathcal{A}/\lambda s.F_1(s,\theta_1) \\
& \qquad \lambda s.f(g(s,\theta_1),\theta_2), \\
& \qquad \lambda s.f'(s,\theta_2) \\
& \hat{b} \leftarrow V_1(s,\theta_1) \\
& \textbf{return } \hat{b}=1
\end{array}
$$

After the hypothesis we made about the decomposition of oracle F_2, and when detailing $\mathcal{B}$, this game can be rewritten as follows, and rigorously compared to the game played by adversary $\mathcal{A}$ against criterion γ:

$$
\begin{array}{ll}
\textbf{Game} & \mathbf{G}^{\gamma_2}_{\mathcal{B}}(\eta)|b=1: \\
& \theta_1 \leftarrow \Theta_1(\eta) \\
& \theta'_1 \leftarrow \Theta_1(\eta) \\
& \theta_2 \leftarrow \Theta_2(\eta) \\
& s \leftarrow \mathcal{A}/\lambda s.F_1(s,\theta_1), \\
& \qquad \lambda s.F_2(s,\theta_1,\theta_2) \\
& \hat{b} \leftarrow V_1(s,\theta_1) \\
& \textbf{return } \hat{b}=1
\end{array}
\qquad
\begin{array}{ll}
\textbf{Game} & \mathbf{G}^{\gamma}_{\mathcal{A}}(\eta): \\
& \theta_1 \leftarrow \Theta_1(\eta) \\
& \theta_2 \leftarrow \Theta_2(\eta) \\
& s \leftarrow \mathcal{A}/\lambda s.F_1(s,\theta_1), \\
& \qquad \lambda s.F_2(s,\theta_1,\theta_2) \\
& \textbf{return } V_1(s,\theta_1)
\end{array}
$$

Therefore these two games are equivalent and so equation (1) holds:

$$Pr[\mathbf{G}^{\gamma_2}_{\mathcal{B}} = true \mid b=1] = Pr[\mathbf{G}^{\gamma}_{\mathcal{A}}(\eta) = true]$$

We now detail the game played by adversary $\mathcal{B}$ against γ_2 when the challenge bit b is 0. This game is compared to the game played by $\mathcal{A}'$ against γ_1.

$$
\begin{array}{ll}
\textbf{Game} & \mathbf{G}^{\gamma_2}_{\mathcal{B}}(\eta)|b=0: \\
& \theta_2 \leftarrow \Theta_2(\eta) \\
& \theta_1 \leftarrow \Theta_1(\eta) \\
& \theta'_1 \leftarrow \Theta_1(\eta) \\
& s \leftarrow \mathcal{A}/\lambda s.F_1(s,\theta_1), \\
& \qquad \lambda s.F_2(s,\theta'_1,\theta_2) \\
& \hat{b} \leftarrow V_1(s,\theta_1) \\
& \textbf{return } \hat{b}=0
\end{array}
\qquad
\begin{array}{ll}
\textbf{Game} & \mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta): \\
& \theta_1 \leftarrow \Theta_1(\eta) \\
& \theta'_1 \leftarrow \Theta_1(\eta) \\
& \theta_2 \leftarrow \Theta_2(\eta) \\
& s \leftarrow \mathcal{A}/\lambda s.F_1(s,\theta_1), \\
& \qquad \lambda s.F_2(s,\theta'_1,\theta_2) \\
& \textbf{return } V_1(s,\theta_1)
\end{array}
$$

It is easy to see that these two games can be compared: adversary $\mathcal{B}$ wins anytime $\mathcal{A}'$ loses, and thus:

$$Pr[\mathbf{G}^{\gamma_1}_{\mathcal{A}'}(\eta) = false] = Pr[\mathbf{G}^{\gamma_2}_{\mathcal{B}}(\eta) = true|b=0]$$

We can therefore evaluate our distinguisher's advantage. For that purpose let us first notice that as γ_2 consists in guessing the value of a random bit b, $PrRand^{\gamma_2}$ equals 1/2. Furthermore γ and γ_1 have the same verifier V_1, hence $PrRand^{\gamma}$ is equal to $PrRand^{\gamma_1}$.

$$\begin{aligned}\mathbf{Adv}_{\mathcal{B}}^{\gamma_2}(\eta) &= 2\big(Pr[\mathbf{G}_{\mathcal{B}}^{\gamma_2}(\eta) = true] - PrRand^{\gamma_2}\big)\\ &= 2Pr[\mathbf{G}_{\mathcal{B}}^{\gamma_2}(\eta) = true|b=1]Pr[b=1] +\\ &\quad 2Pr[\mathbf{G}_{\mathcal{B}}^{\gamma_2}(\eta) = true|b=0]Pr[b=0] - 1\\ &= Pr[\mathbf{G}_{\mathcal{A}}^{\gamma}(\eta) = true] + Pr[\mathbf{G}_{\mathcal{A}'}^{\gamma_1}(\eta) = false] - 1\\ &= Pr[\mathbf{G}_{\mathcal{A}}^{\gamma}(\eta) = true] - Pr[\mathbf{G}_{\mathcal{A}'}^{\gamma_1}(\eta) = true]\\ &= Pr[\mathbf{G}_{\mathcal{A}}^{\gamma}(\eta) = true] - PrRand^{\gamma}\\ &\quad + PrRand^{\gamma_1} - Pr[\mathbf{G}_{\mathcal{A}'}^{\gamma_1}(\eta) = true]\\ &= \frac{1}{2}\mathbf{Adv}_{\mathcal{A}}^{\gamma}(\eta) - \frac{1}{2}\mathbf{Adv}_{\mathcal{A}'}^{\gamma_1}(\eta)\end{aligned}$$

Given an adversary $\mathcal{A}$ against γ, we were able to build an adversary $\mathcal{A}'$ against γ_1 and an adversary $\mathcal{B}$ against γ_2 such that:

$$\forall\eta, \mathbf{Adv}_{\mathcal{A}}^{\gamma}(\eta) = 2\mathbf{Adv}_{\mathcal{B}}^{\gamma_2}(\eta) + \mathbf{Adv}_{\mathcal{A}'}^{\gamma_1}(\eta)$$

This is summed up in the following theorem which is our core result.

Theorem 1 (Criterion Partition). *Let γ be the criterion $(\Theta_1, \Theta_2; F_1, F_2; V_1)$ where:*

1. *V_1 and F_1 only depend on the challenge generated by Θ_1, denoted by θ_1.*
2. *There exist some PRTMs f, f' and g such that F_2 is constituted of two parts: $\lambda s.f(g(s,\theta_1),\theta_2)$ and $\lambda s.f'(s,\theta_2)$*

Then, for any adversary $\mathcal{A}$ against criterion γ, there exist two adversaries $\mathcal{B}$ and $\mathcal{A}'$, such that:

$$\forall\eta, \boldsymbol{Adv}_{\mathcal{A}}^{\gamma}(\eta) = 2\boldsymbol{Adv}_{\mathcal{B}}^{\gamma_2}(\eta) + \boldsymbol{Adv}_{\mathcal{A}'}^{\gamma_1}(\eta)$$

where $\gamma_2 = (\Theta_2, b; f \circ LR^b, f'; v_b)$ is an indistinguishability criterion and $\gamma_1 = (\Theta_1; F_1; V_1)$.

This theorem can be used to prove that the advantage of any adversary against a criterion γ is negligible. For that purpose, one has to provide a partition of γ such that the advantage of any adversary against γ_1 or γ_2 is negligible. Then we get that for an adversary $\mathcal{A}$ against γ, the advantage of $\mathcal{A}$ can be bounded by the advantage of an adversary against γ_1 and the advantage of an adversary against γ_2. The advantage of these two new adversaries are negligible and so the advantage of $\mathcal{A}$ is also negligible.

4 Mixing Asymmetric and Symmetric Encryption

4.1 Cryptographic Game: *N*-PAT-IND-CCA

We introduce a security criterion that turns out to be useful for protocols where secret keys are exchanged. This criterion is an extension of semantic security against chosen cipher-text attacks (IND-CCA). In the classical N-IND-CCA criterion (see [BBM00] about N-IND-CCA and its reduction to IND-CCA), a random bit b is sampled. For each key, the adversary has access to a left-right oracle (the adversary submits a pair of bit-strings bs_0, bs_1 and receives the encoding of bs_b) and a decryption oracle (that does not work on the outputs of the left-right oracle). The adversary has to guess the value of b. Criterion IND-CPA is the same as IND-CCA except that the adversary does not have access to the decryption oracle.

Since it has no information concerning secret keys, the adversary cannot get the encryption of a challenge secret key under a challenge public key. Therefore, we introduce the N-PAT-IND-CCA criterion where the adversary can obtain the encryption of messages containing challenge secret keys, even if it does not know their values. For that purpose, the adversary is allowed to give pattern terms to the left-right oracles.

Pattern terms are terms where new atomic constants have been added: pattern variables. These variables represent the different challenge secret keys and are denoted by $[i]$ (this asks the oracle to replace the pattern variable by the value of sk_i). Variables can be used as atomic messages (data pattern) or at a key position (key pattern). When a left-right oracle is given a pattern term, it replaces patterns by values of corresponding keys and encodes the so-obtained message.

More formally, patterns are given by the following grammar where bs is a bit-string and i is an integer. In the definition of pattern terms, we use two binary operators: concatenation and encryption. Concatenation of patterns pat_0 and pat_1 is written (pat_0, pat_1). Encryption of pat with key bs is denoted by $\{pat\}_{bs}$. Similarly, when the key is a challenge key, it is represented by a pattern variable $[i]$.

$$\begin{aligned} pat ::= &\ (pat, pat) \mid \{pat\}_{bs} \mid \{pat\}_{[i]} \\ &\mid bs \mid [i] \end{aligned}$$

The computation (evaluation) made by the oracle is easily defined recursively in a context θ associating bit-string values to the different keys. Its result is a bit-string and it uses the encryption algorithm $\mathcal{E}$ and the concatenation denoted by "·" in the computational setting.

$$\begin{aligned} v(bs, \theta) &= bs \\ v([i], \theta) &= \theta(sk_i) \\ v((p_1, p_2), \theta) &= v(p_1, \theta) \cdot v(p_2, \theta) \end{aligned} \qquad \begin{aligned} v(\{p\}_{bs}, \theta) &= \mathcal{E}(v(p, \theta), bs) \\ v(\{p\}_{[i]}, \theta) &= \mathcal{E}(v(p, \theta), \theta(pk_i)) \end{aligned}$$

There is yet a restriction. Keys are ordered and a pattern $[j]$ can only be encrypted under pk_i if $i < j$ to avoid key cycles. This restriction is well-known in cryptography and widely accepted [AR00]. When the left-right pattern encryption oracle related to key i is given two pattern terms pat_0 and pat_1, it tests that none contains a pattern $[j]$ with $j \leq i$. If this happens, it outputs an error message, else it produces the encryption of the message corresponding to pat_b, $v(pat_b, \theta)$, using public key pk_i. To win, the adversary has to guess the value of secret bit b. In fact our acyclicity hypothesis only occurs on secret keys: when considering pattern $\{\{p\}_{[j]}\}_{[j]}$, the public key oracle related to key j can be called and returns bit-string bs, then pattern $\{\{p\}_{bs}\}_{[j]}$ can be used to get the awaited result. We do not detail restrictions on the length of arguments submitted to the left-right oracle, an interesting discussion on that point appears in [AR00]. The most simple restriction is to ask that both submitted patterns can only be evaluated (using v) to bit-strings of equal length.

Henceforth, let $\mathcal{AE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ be an asymmetric encryption scheme. Then, criterion N-PAT-IND-CCA is given by $\gamma_N = (\Theta; F; V)$, where Θ randomly generates N pairs of keys (pk_1, sk_1) to (pk_N, sk_N) using $\mathcal{KG}$ and a bit b; V verifies whether the adversary gave the right value for bit b; and F gives access to three oracles for each i between 1 and N: a left-right encryption oracle that takes as argument a pair of patterns (pat_0, pat_1) and outputs pat_b completed with the secret keys $(v(pat_b, \theta))$ and encoded using pk_i; a decryption oracle that decodes any message that was not produced by the former encryption oracle; and an oracle that simply makes the public key pk_i available.

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

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

Proof. We want to establish first that an IND-CCA asymmetric encryption scheme is an N-PAT-IND-CCA secure one. We use the criterion reduction theorem on N-PAT-IND-CCA (denoted by δ_N). We now consider $\delta_N = (\Theta_1, \Theta_2; F_1, F_2; V_1)$, where the criterion partition has been performed the following way:

- Θ_1 randomly generates the bit b and $N - 1$ pairs of matching public and secret keys (pk_2, sk_2) to (pk_N, sk_N) using $\mathcal{KG}$.
- Θ_2 randomly generates the first key pair (pk_1, sk_1).
- F_1 contains the oracles related to θ_1; hence as neither pk_1 nor sk_1 can be asked to this oracle (because of acyclicity), F_1 does not depend on θ_2.
- F_2 contains the oracles related to key pair (pk_1, sk_1), it uses θ_1 for the bit b and the different keys needed to fill in patterns.
- V_1 compares the output to b, and therefore only depends on θ_1.

This splitting complies with the first hypothesis of theorem 1. Let us then check whether the second hypothesis holds. The decryption and public key oracles included in F_2 only depend on θ_2, we place them in f'. We let the encryption oracle be $\lambda s.f(g(s, \theta_1), \theta_2)$ where $g((pat_0, pat_1), \theta_1) = v(pat_b, \theta_1)$ plays the role of a left-right oracle, b being the challenge bit included in θ_1, composed with the valuation function v that completes patterns, and $f(bs, \theta_2) = \mathcal{E}(bs, pk_1)$ is the original encryption oracle.

The theorem can now be applied. It thus follows that for any adversary $\mathcal{A}$ against criterion δ_N, there exist two adversaries $\mathcal{B}$ and $\mathcal{A}'$, such that:

$$\forall \eta, \mathbf{Adv}_{\mathcal{A}}^{\delta_N}(\eta) = 2\mathbf{Adv}_{\mathcal{B}}^{\gamma_2}(\eta) + \mathbf{Adv}_{\mathcal{A}'}^{\gamma_1}(\eta)$$

where $\gamma_2 = (\Theta_2, b; f \circ LR^b, f'; v_b)$ is IND-CCA and $\gamma_1 = (\Theta_1; F_1; V_1)$ is criterion δ_{N-1}.

Hence if we suppose that the asymmetric encryption scheme $\mathcal{AE}$ is IND-CCA and $N-1$-PAT-IND-CCA, then the advantages of $\mathcal{A}'$ and $\mathcal{B}$ are negligible, so the advantage of $\mathcal{A}$ is also negligible and $\mathcal{AE}$ is N-PAT-IND-CCA. Moreover, as 0-PAT-IND-CCA consists in guessing a challenge bit without access to any oracle, any adversary's advantage against it is thus null, which obviously implies that any encryption scheme is 0-PAT-IND-CCA. Using a quick recursion, it now appears clearly that if an asymmetric encryption scheme is IND-CCA, it is also N-PAT-IND-CCA for any integer N.

In this proof, we bound the advantage against N-PAT-IND-CCA by $2N$ times the advantage against IND-CCA. This bound is not contradictory with the one proposed by [BBM00] as the number of queries to each oracle is unbounded in our model.

4.2 Cryptographic Game: *N*-PAT-SYM-CPA

In this section, we introduce a new criterion describing safety of a symmetric encryption scheme. This definition is an extension of semantic security against chosen plain-text attacks. The main difference with the N-PAT-IND-CCA criterion is that there are no public key oracles and no decryption oracles. Hence the left-right encryption oracles are similar to those presented in the previous section and the adversary still has to guess the value of the challenge bit b. The hypothesis related to acyclicity of keys still holds: k_i can only appear encoded by k_j if $i > j$.

The N-PAT-SYM-CPA criterion is $\gamma_N = (\Theta, F, V)$ where Θ generates N symmetric keys and a bit b; F gives access to one oracle for each key: a left-right encryption oracle that takes as argument a pair of patterns (pat_0, pat_1) and outputs pat_b completed with the secret keys $(v(pat_b, \theta))$ and encoded with k_i. Finally, V returns true when the adversary returns bit b.

Let γ_N be a criterion including the oracles detailed above. A symmetric encryption scheme $\mathcal{SE}$ is said N-PAT-SYM-CPA iff for any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ against γ_N, $\mathbf{Adv}^{\gamma_N}_{\mathcal{SE},\mathcal{A}}(\eta)$, is negligible in η.

Using the criterion partition theorem, it is possible to reduce criterion N-PAT-SYM-CPA to criterion SYM-CPA. This can be done by using the same partition as for criterion N-PAT-IND-CCA.

Proposition 2. *Let N be an integer. If a symmetric encryption scheme $\mathcal{SE}$ is SYM-CPA, then $\mathcal{SE}$ is N-PAT-SYM-CPA.*

4.3 Cryptographic Games: *N*-PAS-CCA and *N*-PAS-CPA

These criteria combine both precedent ones. N asymmetric and symmetric keys are generated along with a single challenge bit b. The adversary can access oracles it was granted in both previous criteria (left-right encryption, public key and decryption for the asymmetric scheme in N-PAS-CCA) and has to deduce the value of the challenge bit b. The acyclicity condition still holds on both primitives. However, we authorize patterns using symmetric keys when accessing left-right oracles from the asymmetric part. Hence symmetric encryption and symmetric keys can be used under asymmetric encryption but the converse is forbidden. The pattern definition has to be extended so that the adversary can ask for both asymmetric and symmetric encryptions and asymmetric and symmetric keys.

Let γ_N be the criterion including the oracles detailed above. A cryptographic library $(\mathcal{AE}, \mathcal{SE})$ is said N-PAS-CCA iff for any adversary $\mathcal{A}$ the advantage of $\mathcal{A}$, $\mathbf{Adv}^{\gamma_N}_{\mathcal{AE},\mathcal{SE},\mathcal{A}}(\eta)$, is negligible. The challenge bit b is common to asymmetric and symmetric encryption, thus it is non trivial to prove that IND-CCA and SYM-CPA imply N-PAS-CCA. However using our partition theorem, it is possible to prove this implication.

Proposition 3. *Let N be an integer. If an asymmetric encryption scheme $\mathcal{AE}$ is IND-CCA and a symmetric encryption scheme $\mathcal{SE}$ is SYM-CPA, then the cryptographic library $(\mathcal{AE}, \mathcal{SE})$ is N-PAS-CCA.*

This can easily be adapted to prove variants of this property, for example let us consider the IND-CPA criterion for the symmetric encryption scheme (the adversary only has access to the left-right oracle and has to guess the challenge bit) and the N-PAS-CPA criterion for a cryptographic library (the adversary has access to public keys for the asymmetric encryption scheme, to left-right oracles using patterns such that asymmetric secret keys cannot be asked to symmetric encryption oracles).

Proposition 4. *Let N be an integer. If an asymmetric encryption scheme $\mathcal{AE}$ is IND-CPA and a symmetric encryption scheme $\mathcal{SE}$ is SYM-CPA, then the cryptographic library $(\mathcal{AE}, \mathcal{SE})$ is N-PAS-CPA.*

5 Computational Soundness of Adaptive Security

In this section, we prove computational soundness of symbolic equivalence for messages that use both asymmetric and symmetric encryption in the case of an adaptive adversary. This model has been introduced in [MP05]. Roughly, speaking it corresponds to the case of a passive adversary that however can adaptively chose symbolic terms and ask for their computational evaluation whereas in the passive case [AR00], the adversary is confronted with two fixed symbolic terms. The practical significance of this model is discussed in [MP05]. Our result is an extension of the soundness result from [MP05], moreover we propose a more modular approach which does not use any hybrid argument but is based on proposition 4. Another improvement is that we allow the adversary to reuse computational values within symbolic terms, constants in messages can be used to represent any bit-string. To simplify things up, we do not consider polynomial sequences of messages as in [MP05] but rather bounded sequences of messages. In fact, to cope with the polynomial case, we need to extend theorem 1 in order to handle a polynomial number of challenges. This extension is presented in [Maz06].

5.1 A Symbolic Treatment of Cryptography

Let **SymKeys**,**PKeys**,**SKeys** and **Const** be four disjoint sets of symbols representing *symmetric keys*, *public keys*, *secret keys* and *constants*. Let **Atoms** be the union of the previous sets. We assume the existence of a bijection $[]^{-1}$ from **PKeys** to **SKeys** that associates to each public key the corresponding secret key. The inverse of this function is also denoted $[]^{-1}$. The set **Msg** of messages is defined by the following grammar.

$$\mathbf{Msg} ::= \mathbf{SymKeys} \mid \mathbf{Const} \mid (\mathbf{Msg}, \mathbf{Msg}) \mid \{\mathbf{Msg}\}^s_{\mathbf{SymKeys}} \mid \{\mathbf{Msg}\}^a_{\mathbf{PKeys}}$$

Elements of **SymKeys** can be thought of as randomly sampled keys, elements of **Const** as bit-strings. Term (m, n) represents the pairing of message m and n, $\{m\}^s_k$ represents the symmetric encryption of m using key k and $\{m\}^a_{pk}$ represents the asymmetric encryption of m using public key pk. In the sequel, when presenting examples, we use symbols 0 and 1. These are to be understood as elements of **Const** which computational interpretations are respectively bit-strings 0 and 1.

Next we define when a message $m \in \mathbf{Msg}$ can be deduced from a set of messages $E \subseteq \mathbf{Msg}$ (written $E \vdash m$) by a passive eavesdropper. The deduction relation $\vdash$ is defined by the standard Dolev-Yao inference system [DY83] and is given by the following rules:

$$\frac{m \in E}{E \vdash m} \quad \frac{E \vdash (m_1, m_2)}{E \vdash m_1} \quad \frac{E \vdash (m_1, m_2)}{E \vdash m_2} \quad \frac{E \vdash m_1 \quad E \vdash m_2}{E \vdash (m_1, m_2)}$$
$$\frac{E \vdash m \quad E \vdash k}{E \vdash \{m\}^s_k} \quad \frac{E \vdash \{m\}^s_k \quad E \vdash k}{E \vdash m} \quad \frac{E \vdash m \quad E \vdash pk}{E \vdash \{m\}^a_{pk}} \quad \frac{E \vdash \{m\}^a_{pk} \quad E \vdash pk^{-1}}{E \vdash m}$$

The information revealed by a symbolic expression can be characterized using *patterns* [AR00, MP05]. For a message $m \in \mathbf{Msg}$ its pattern is defined by the following inductive rules:

$$\begin{array}{ll}
pattern\big((m_1, m_2)\big) = \big(pattern(m_1), pattern(m_2)\big) & \\
pattern\big(\{m'\}^s_k\big) = \{pattern(m')\}^s_k & \text{if } m \vdash k \\
pattern\big(\{m'\}^s_k\big) = \{\Box\}^s_k & \text{if } m \not\vdash k \\
pattern\big(\{m'\}^a_{pk}\big) = \{pattern(m')\}^a_{pk} & \text{if } m \vdash pk^{-1} \\
pattern\big(\{m'\}_{pk}\big) = \{\Box\}^a_{pk} & \text{if } m \not\vdash pk^{-1} \\
pattern(m') = m' & \text{if } m' \in \mathbf{Atoms}
\end{array}$$

The symbol $\Box$ represents a cipher-text that the adversary cannot decrypt. As $\Box$ does not store any information on the length or structure of the corresponding plain-text, we assume that the encryption schemes used here do not reveal plain-text lengths (see [AR00] for details). Two messages are said to be *equivalent* if they have the same pattern: $m \equiv n$ if and only if $pattern(m) = pattern(n)$. Two messages are *equivalent up to renaming* if they are equivalent up to some renaming of keys: $m \cong n$ if there exists a renaming σ of keys from n such that $m \equiv n\sigma$.

Example 1. Let us illustrate this equivalence notion. We have that:

- $\{0\}_k^s \cong \{1\}_k^s$ encryptions with different plain-text cannot be distinguished if the key is not deducible.
- $(\{0\}_k^s, \{k\}_{pk}^a, pk^{-1}) \not\cong (\{1\}_k^s, \{k\}_{pk}^a, pk^{-1})$ but it is not the case if the key can be deduced.

5.2 Computational Soundness

This model is parameterized by an asymmetric encryption scheme $\mathcal{AE} = (\mathcal{KG}^a, \mathcal{E}^a, \mathcal{D}^a)$ and a symmetric encryption scheme $\mathcal{SE} = (\mathcal{KG}^s, \mathcal{E}^s, \mathcal{D}^s)$. Computational semantics are given by a concretization function *concr* which can be derived from the v function that was introduced previously. This algorithm uses a computational substitution θ which stores bit-string values for keys. Constants from **Const** represents bit-strings so the concretization of c from **Const** is c itself.

$$\begin{aligned} concr((m_1, m_2), \theta) &= concr(m_1, \theta) \cdot concr(m_2, \theta) \\ concr(\{m\}_{pk}^a, \theta) &= \mathcal{E}^a(concr(m, \theta), \theta(pk)) \\ concr(\{m\}_k^s, \theta) &= \mathcal{E}^s(concr(m, \theta), \theta(k)) \end{aligned} \qquad \begin{aligned} concr(k, \theta) &= \theta(k) \\ concr(c, \theta) &= c \end{aligned}$$

Thus the computational distribution generated by a message can be obtained by randomly sampling the necessary keys and using the *concr* function.

We consider a model where the adversary can see the computational version of a bounded sequence of adaptively chosen messages. Let α be a bound on the sequence length. The adaptive experiment proceeds as follows: the adversary has access to one oracle which takes as argument a pair of messages (m_0, m_1) and either outputs a concretization of m_0 (oracle $\mathcal{O}_0$) or a concretization of m_1 (oracle $\mathcal{O}_1$). These oracles work by randomly sampling the necessary keys then using the *concr* function on either m_0 or on m_1. Finally, the adversary has to tell against which oracle it is playing, $\mathcal{O}_0$ or $\mathcal{O}_1$. The advantage of $\mathcal{A}$ is defined by:

$$\mathbf{Adv}_{\mathcal{AE},\mathcal{SE},\mathcal{A}}^{adpt}(\eta) = Pr[\mathcal{A}/\mathcal{O}_1 = 1] - Pr[\mathcal{A}/\mathcal{O}_0 = 1]$$

Moreover there are restrictions on the sequence of messages submitted by the adversary (m_0^1, m_1^1) to (m_0^q, m_1^q). Such a sequence is said to be *legal* if:

1. Messages $(m_0^1, ..., m_0^q)$ and $(m_1^1, ..., m_1^q)$ are equivalent up to renaming.
2. Messages $(m_0^1, ..., m_0^q)$ and $(m_1^1, ..., m_1^q)$ contain no encryption cycles, moreover secret keys cannot be sent under symmetric encryptions.
3. The lengths of $(m_0^1, ..., m_0^q)$ and $(m_1^1, ..., m_1^q)$ are lower than α.

Proposition 5. *If $\mathcal{AE}$ is an IND-CPA secure encryption scheme and $\mathcal{SE}$ is a SYM-CPA secure encryption scheme, then the advantage of any legal adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{AE},\mathcal{SE},\mathcal{A}}^{adpt}(\eta)$, is a negligible function in η.*

This result can be used to model secure multicast as presented in [MP05].

6 Conclusion

This paper contributes to the development of a proof theory for cryptographic systems by providing a theorem that allows to decompose the proof of correctness of a security criterion to the correctness of a sub-criterion and an indistinsguishability criterion. We apply this decomposition result to prove that given secure asymmetric and symmetric encryption schemes we can combine them to obtain a secure cryptographic library.

This security result can be used to easily prove computational soundness of formal methods. This has been illustrated in the case of the adaptive setting for asymmetric and symmetric encryption.

In future works, we intend to develop this computational soundness result to the case of security protocols in general against an active adversary. We believe that our partition theorem will also be useful in this situation, in particular by giving simpler and more modular proofs of soundness.

References

[AJ01] Abadi, M., Jürjens, J.: Formal eavesdropping and its computational interpretation. In: Kobayashi, N., Pierce, B.C. (eds.) TACS 2001. LNCS, vol. 2215, pp. 82–94. Springer, Heidelberg (2001)

[AR00] Abadi, M., Rogaway, P.: Reconciling two views of cryptography (the computational soundness of formal encryption). In: IFIP International Conference on Theoretical Computer Science (IFIP TCS2000), Sendai, Japan, Springer, Berlin (2000)

[BBM00] Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (2000)

[BCK05] Baudet, M., Cortier, V., Kremer, S.: Computationally sound implementations of equational theories against passive adversaries. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, Springer, Heidelberg (2005)

[Bla06] Blanchet, B.: A computationally sound mechanized prover for security protocols. In: IEEE Symposium on Security and Privacy, Oakland, California (May 2006)

[BPW03] Backes, M., Pfitzmann, B., Waidner, M.: A composable cryptographic library with nested operations. In: Proceedings of the 10th ACM conference on Computer and communication security, pp. 220–230 (2003)

[BR04] Bellare, M., Rogaway, P.: The game-playing technique. Cryptology ePrint Archive, Report 2004/331 (2004), `http://eprint.iacr.org/`

[Cou90] Cousot, P.: Methods and Logics for Proving Programs. In: Handbook of Theoretical Computer Science, vol. B: Formal Methods and Semantics, pp. 841–994. Elsevier Science Publishers B.V, Amsterdam (1990)

[CW05] Cortier, V., Warinschi, B.: Computationally sound, automated proofs for security protocols. In: Sagiv, M. (ed.) ESOP 2005. LNCS, vol. 3444, Springer, Heidelberg (2005)

[DY83] Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Transactions on Information Theory 29(2), 198–208 (1983)

[GM84] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)

[JLM05] Janvier, R., Lakhnech, Y., Mazaré, L.: Completing the picture: Soundness of formal encryption in the presence of active adversaries. In: Sagiv, M. (ed.) ESOP 2005. LNCS, vol. 3444, Springer, Heidelberg (2005)

[Maz06] Mazaré, L.: Computational Soundness of Symbolic Models for Cryptographic Protocols. PhD thesis, INPG, Grenoble (October 2006) (to appear)

[MP92] Manna, Z., Pnueli, A.: The temporal logic of reactive and concurrent systems. Springer, Heidelberg (1992)

[MP05] Micciancio, D., Panjwani, S.: Adaptive security of symbolic encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 169–187. Springer, Heidelberg (2005)

[MW04] Micciancio, D., Warinschi, B.: Soundness of formal encryption in the presence of active adversaries. In: Proceedings of the Theory of Cryptography Conference, pp. 133–151. Springer, Heidelberg (2004)

[Sho04] Shoup, V.: Sequences of games: a tool for taming complexity in security proofs (2004)