The Cryptographers' Track at RSA Conference '2002
(18 – 22 february 2002, San Jose, California, USA)
B. Preneel Ed. Springer-Verlag, LNCS 2271, pages 263–276.

GEM: a Generic Chosen-Ciphertext Secure Encryption Method

Jean-Sébastien Coron[1], Helena Handschuh[1], Marc Joye[2], Pascal Paillier[1], David Pointcheval[3], and Christophe Tymen[1,3]

[1] Gemplus Card International
34 rue Guynemer, 92447 Issy-les-Moulineaux, France
{jean-sebastien.coron, helena.handschuh, pascal.paillier, christophe.tymen}@gemplus.com
[2] Gemplus Card International
Parc d'Activités de Gémenos, B.P. 100, 13881 Gémenos Cedex, France
marc.joye@gemplus.com – http://www.geocities.com/MarcJoye/
[3] École Normale Supérieure, Computer Science Department
45 rue d'Ulm, 75230 Paris Cedex 05, France
david.pointcheval@ens.fr – http://www.di.ens.fr/~pointche/

Abstract. This paper proposes an efficient and provably secure transform to encrypt a message with any asymmetric one-way cryptosystem. The resulting scheme achieves adaptive chosen-ciphertext security in the random oracle model.
Compared to previous known generic constructions (Bellare, Rogaway, Fujisaki, Okamoto, and Pointcheval), our embedding reduces the encryption size and/or speeds up the decryption process. It applies to numerous cryptosystems, including (to name a few) ElGamal, RSA, Okamoto-Uchiyama and Paillier systems.

Keywords: Public-key encryption, hybrid encryption, chosen-ciphertext security, random oracle model, generic conversion, block ciphers, stream ciphers.

1 Introduction

A major contribution of cryptography is *information privacy*: through encryption, parties can securely exchange data over an insecure channel. Loosely speaking this means that unauthorized recipients can learn nothing useful about the exchanged data.

Designing a "good" encryption scheme is a very challenging task. There are basically two criteria to compare the performances of encryption schemes: *efficiency* and *security*. Security is measured as the ability to resist attacks in a given adversarial model [1, 8]. The standard security notion is IND-CCA2 *security*, *i.e.*, indistinguishability under adaptive chosen-ciphertext attacks (cf. Section 2). Usually, an (asymmetric) encryption scheme is proven secure by exhibiting a *reduction*: if an adversary can break the IND-CCA2 security then the same adversary can solve a related problem assumed to be infeasible.

This paper is aimed at simplifying the security proof by providing a *Generic Encryption Method* (GEM) to convert *any* asymmetric one-way cryptosystem into a *provably secure* encryption scheme. Hence, when a new asymmetric one-way function is identified, one can easily design a secure encryption scheme. Moreover, the conversion we propose is very efficient (computationally and memory-wise): the converted scheme has roughly the same cost as that of the one-way cryptosystem it is built from.

1.1 Previous work

In [3], Bellare and Rogaway described OAEP, a generic conversion to transform a *"partial-domain" one-way trapdoor permutation* into an IND-CCA2 secure encryption scheme in the random oracle model [2, 7]. Later, Fujisaki and Okamoto [5] presented a way to transform, in the random oracle model, any *chosen-plaintext* (IND-CPA) secure encryption scheme into an IND-CCA2 one. They improved their results in [6] where they gave a generic method to convert a *one-way* (OW-CPA) cryptosystem into an IND-CCA2 secure encryption scheme in the random oracle model. A similar result was independently discovered by Pointcheval [13]. More recently, Okamoto and Pointcheval [12] proposed a more efficient generic conversion, called REACT, to convert any one-way cryptosystem secure under *plaintext-checking attacks* (OW-PCA) into an IND-CCA2 encryption scheme. Contrary to [5, 6, 13], re-encryption is unnecessary in the decryption process to ensure IND-CCA2 security.

1.2 Our results

This paper presents GEM, a generic IND-CCA2 conversion. The converted scheme, $\mathbb{E}_{pk}$, built from any OW-PCA asymmetric encryption $\mathcal{E}_{pk}$ and any length-preserving IND-secure symmetric encryption scheme $\mathtt{E}_K$, is secure in the sense of IND-CCA2, in the random oracle model. As discussed in Section 2, the security levels we require for $\mathcal{E}_{pk}$ and $\mathtt{E}_K$ are *very* weak and the security level we obtain for $\mathbb{E}_{pk}$ is very high.

1.3 Organization of the paper

The rest of this paper is organized as follows. In Section 2, we review the security notions for encryption, in both the symmetric and the asymmetric settings. Section 3 is the core of the paper. We present our new padding to convert, in the random oracle model, any asymmetric one-way cryptosystem into an encryption scheme that is secure in the *strongest* sense. We prove the security of our construction in Section 4 by providing and proving a *concrete* reduction algorithm. Finally, we illustrate the merits of our conversion in Section 5.

2 Security Notions

In this section, we recall the definition of an encryption scheme and discuss some related security notions. A good reference to the subject is [1].

2.1 Encryption schemes

Definition 1. An *encryption scheme* consists of three algorithms $(\mathcal{K}, \mathcal{E}_{pk}, \mathcal{D}_{sk})$.

1. On input a security parameter k, the key generation algorithm $\mathcal{K}(1^k)$ outputs a random matching pair (pk, sk) of encryption/decryption keys. For the symmetric case, we assume wlog that the encryption and decryption keys are identical, $K := pk = sk$. The key pk is public and the keys sk and K are secret.

2. The encryption algorithm $\mathcal{E}_{pk}(m, r)$ outputs a ciphertext c corresponding to a plaintext $m \in \mathtt{MSPC} \subseteq \{0,1\}^*$, using the random coins $r \in \Omega$. When the process is deterministic, we simply write $\mathcal{E}_{pk}(m)$. In the symmetric case, we note $\mathtt{E}_K$ instead of $\mathcal{E}_{pk}$.
3. The decryption algorithm $\mathcal{D}_{sk}(c)$ outputs the plaintext m associated to the ciphertext c or a notification $\perp$ that c is not a valid ciphertext. In the symmetric case, we use the notations $\mathtt{D}_K$.

Furthermore, we require that for all $m \in \mathtt{MSPC}$ and $r \in \Omega$, $\mathcal{D}_{sk}(\mathcal{E}_{pk}(m, r)) = m$.

The converted scheme we propose in this paper is a combination of an asymmetric encryption scheme and a length-preserving symmetric encryption scheme. We assume very weak security properties from those two schemes. Namely, we require that the asymmetric scheme is OW-PCA and that the symmetric scheme is IND, as defined below.

2.2 Security requirements

An attacker is said *passive* if, in addition to the ciphertext, s/he only obtains some auxiliary information s, which may depend on the potential plaintexts (but not on the key) [9, § 1.5] and other public information (*e.g.*, the security parameter k and the public key pk). Note that in the asymmetric case an attacker can always construct valid pairs of plaintext/ciphertext from the public encryption key pk.

A minimal security requirement for an encryption scheme is *one-wayness* (OW). This captures the property that an adversary cannot recover the *whole* plaintext from a given ciphertext. In some cases, partial information about a plaintext may have disastrous consequences. This is captured by the notion of *semantic security* or the equivalent notion of *indistinguishability* [10]. Basically, indistinguishability means that the only strategy for an adversary to distinguish the encryptions of any two plaintexts is by guessing at random.

In the asymmetric case, suppose that the attacker has access to an oracle telling whether a pair (m, c) of plaintext/ciphertext is valid; *i.e.*, whether $m = \mathcal{D}_{sk}(c)$ holds. Following [12], such an attack scenario is referred to as the *plaintext-checking attack* (PCA). From the pair of adversarial goal (OW) and adversarial model (PCA), we derive the security notion of OW-PCA.

Definition 2. An asymmetric encryption scheme is OW-PCA if no attacker with access to a plaintext-checking oracle $\mathcal{O}^{\text{PCA}}$ can recover the whole plaintext corresponding to a ciphertext with non-negligible probability. More formally, an asymmetric encryption scheme is (τ, q, ε)-secure in the sense of OW-PCA if for any adversary $\mathcal{A}$ which runs in time at most τ, makes at most q queries to $\mathcal{O}^{\text{PCA}}$, its success ε satisfies

$$\Pr_{\substack{m \leftarrow \{0,1\}^* \\ r \leftarrow \Omega}} \left[\begin{array}{c} (sk, pk) \leftarrow \mathcal{K}(1^k), c \leftarrow \mathcal{E}_{pk}(m, r) : \\ \mathcal{A}^{\mathcal{O}^{\text{PCA}}}(c, s) = m \end{array} \right] \leq \varepsilon$$

where the probability is also taken over the random choices of $\mathcal{A}$.

For the symmetric case, we consider a passive attacker who tries to break the indistinguishability property of the encryption scheme. The attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ runs in two stages. In the first stage (or *find stage*), on input k, $\mathcal{A}_1$ outputs a pair of messages (m_0, m_1) and some auxiliary information s. Next, in the second stage (or *guess stage*), given the encryption c_b of either m_0 or m_1 and the auxiliary information s, $\mathcal{A}_2$ tells if the challenge ciphertext c_b encrypts m_0 or m_1.

Definition 3. A symmetric encryption scheme is IND if no attacker can distinguish the encryptions of two equal-length plaintexts with probability non-negligibly greater than 1/2. More formally, a symmetric encryption scheme is (τ, ν)-secure in the sense of IND if for any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ which runs in time at most τ, its advantage ν satisfies

$$\Pr_{b \leftarrow \{0,1\}} \left[\begin{array}{c} K \leftarrow \mathcal{K}(1^k), (m_0, m_1, s) \leftarrow \mathcal{A}_1(k), \\ c_b \leftarrow \mathtt{E}_K(m_b) : \mathcal{A}_2(m_0, m_1, c_b, s) = b \end{array} \right] \leq \frac{1+\nu}{2}$$

where the probability is also taken over the random choices of $\mathcal{A}$.

In contrast, for our converted encryption scheme we require the *highest* security level, namely IND-CCA2 security. The notion of IND-CCA2 for an asymmetric encryption scheme considers an *active* attacker who tries to break the system by probing with chosen-ciphertext messages. Such an attack can be non-adaptive (CCA1) [11] or adaptive (CCA2) [14]. In a CCA2 scenario, the adversary may run a second chosen ciphertext attack upon receiving the challenge ciphertext c_b (the only restriction being not to probe on c_b).

Definition 4. An asymmetric encryption scheme is IND-CCA2 if no attacker with access to a decryption oracle $\mathcal{O}^{\mathcal{D}_{sk}}$ can distinguish the encryptions of two equal-length plaintexts with probability non-negligibly greater than 1/2. More formally, an asymmetric encryption scheme is (τ, q, ε)-secure in the sense of IND-CCA2 if for any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ which runs in time at most τ, makes at most q queries to $\mathcal{O}^{\mathcal{D}_{sk}}$, its advantage ε satisfies

$$\Pr_{\substack{b \leftarrow \{0,1\} \\ r \leftarrow \Omega}} \left[\begin{array}{c} (sk, pk) \leftarrow \mathcal{K}(1^k), \\ (m_0, m_1, s) \leftarrow {\mathcal{A}_1}^{\mathcal{O}^{\mathcal{D}_{sk}}}(k, pk), \\ c_b \leftarrow \mathcal{E}_{pk}(m_b, r) : \\ {\mathcal{A}_2}^{\mathcal{O}^{\mathcal{D}_{sk}}}(m_0, m_1, c_b, s) = b \end{array} \right] \leq \frac{1+\varepsilon}{2}$$

where the probability is also taken over the random choices of $\mathcal{A}$, and $\mathcal{A}_2$ is not allowed to query on c_b.

3 GEM: Generic Encryption Method

A very appealing way to encrypt a message consists in using a *hybrid encryption* mode. A random session key R is first encrypted with an asymmetric cryptosystem. Then the message is encrypted under that session key with a symmetric cryptosystem. Although seemingly sound, this scheme does not achieve IND-CCA2 security under weak security assumptions for the two underlying cryptosystems. This section shows how to modify the above paradigm for attaining the IND-CCA2 security level.

3.1 REACT transform

The authors of REACT imagined to append a checksum to the previous construction and prove the IND-CCA2 security of the resulting scheme in the random oracle model [12]. Briefly, REACT works as follows. A plaintext m is transformed into the ciphertext (c_1, c_2, c_3) given by

$$\text{REACT}(m) = \underbrace{\mathcal{E}_{pk}(R, u)}_{=c_1} \parallel \underbrace{\mathtt{E}_K(m)}_{=c_2} \parallel \underbrace{\mathrm{H}(R, m, c_1, c_2)}_{=c_3}$$

where u is a random, $K = \mathrm{G}(R)$, and G, H are hash functions.

Building on this, we propose a new generic encryption method. Our method is aimed at shortening the whole ciphertext by incorporating the checksum (*i.e.*, c_3) into c_1 while maintaining the IND-CCA2 security level, in the random oracle model.

3.2 New method

Let $\mathcal{E}_{pk}$ and $\mathtt{E}_K$ denote an asymmetric and a length-preserving symmetric encryption algorithms, respectively, and let $\mathrm{F}, \mathrm{G}, \mathrm{H}$ denote hash functions. Let also $\mathcal{D}_{sk}$ and $\mathtt{D}_K$ denote the decryption algorithms corresponding to $\mathcal{E}_{pk}$ and $\mathtt{E}_K$, respectively. For convenience, for any element x defined over a domain A, we write $\sharp\, x$ for $|\{x \in A\}|$. So, for example, $\sharp\, m$ represents the cardinality of the message space, *i.e.*, the number of different plaintexts.

Encryption

Input: Plaintext m, random $\rho = r \| u$.
Output: Ciphertext (c_1, c_2) given by

$$\mathbb{E}_{pk}(m, \rho) = \underbrace{\mathcal{E}_{pk}(w, u)}_{=c_1} \parallel \underbrace{\mathtt{E}_K(m)}_{=c_2}$$

where $s = \mathrm{F}(m, r)$, $w = s \parallel r \oplus \mathrm{H}(s)$,
and $K = \mathrm{G}(w, c_1)$.

Decryption

Input: Ciphertext (c_1, c_2).
Output: Plaintext $\dot{m}$ or symbol $\perp$ according to

$$\mathbb{D}_{sk}(c_1 \| c_2) = \begin{cases} \dot{m} & \text{if } \dot{s} = \mathrm{F}(\dot{m}, \dot{r}) \\ \perp & \text{otherwise} \end{cases}$$

where $\dot{w} := \dot{s} \| \dot{t} = \mathcal{D}_{sk}(c_1)$, $\dot{K} = \mathrm{G}(\dot{w}, c_1)$,
$\dot{m} = \mathtt{D}_{\dot{K}}(c_2)$, and $\dot{r} = \dot{t} \oplus \mathrm{H}(\dot{s})$.

4 Security Analysis

We now prove the security of our conversion. We show that if the hybrid encryption scheme $\mathbb{E}_{pk}$ can be broken under an adaptive chosen-ciphertext attack then either the length-preserving symmetric encryption scheme $\mathtt{E}_K$ or the asymmetric encryption scheme $\mathcal{E}_{pk}$ underlying our construction is *highly* insecure, namely the IND-security of $\mathtt{E}_K$ or the OW-PCA security of $\mathcal{E}_{pk}$ gets broken.

Theorem 5. *Suppose that there exists an adversary that breaks, in the random oracle model, the* IND-CCA2 *security of our converted scheme* $\mathbb{E}_{pk}$ *within a time bound* τ*, after at most* $q_{\mathrm{F}}, q_{\mathrm{G}}, q_{\mathrm{H}}, q_{\mathbb{D}_{sk}}$ *queries to hash functions* $\mathrm{F}, \mathrm{G}, \mathrm{H}$ *and decryption oracle* $\mathcal{O}^{\mathbb{D}_{sk}}$*, respectively, and with an advantage* ε*. Then, for all* $0 < \nu < \varepsilon$*, there exists*

- *an adversary that breaks the* IND *security of* $\mathtt{E}_K$ *within a time bound* τ *and an advantage* ν*; or*
- *an adversary with access to a plaintext-checking oracle* $\mathcal{O}^{\mathrm{PCA}}$ *(responding in time bounded by* τ_{PCA}*) that breaks the* OW-PCA *security of* $\mathcal{E}_{pk}$ *within a time bound*

$$\tau' = \tau + \left(q_{\mathrm{F}}\, q_{\mathrm{H}} + q_{\mathrm{G}} + q_{\mathbb{D}_{sk}} (q_{\mathrm{F}} + q_{\mathrm{G}})\right) (\tau_{\mathrm{PCA}} + O(1)) \,,$$

after at most

$$q_{\mathrm{PCA}} \leq q_{\mathrm{F}}\, q_{\mathrm{H}} + q_{\mathrm{G}} + q_{\mathbb{D}_{sk}} (q_{\mathrm{F}} + q_{\mathrm{G}})$$

queries to $\mathcal{O}^{\mathrm{PCA}}$*, and with success probability*

$$\varepsilon' \geq \frac{\varepsilon - \nu}{2} - \frac{q_{\mathrm{F}}}{\sharp r} - q_{\mathbb{D}_{sk}} \left(\frac{1}{\sharp s} + q_{\mathrm{F}} \left(\frac{1}{\sharp r} + \frac{1}{\sharp s}\right) + \nu + \frac{1}{\sharp m}\right) \quad .$$

From this, we immediately obtain:

Corollary 6. *For any* OW-PCA *asymmetric encryption* $\mathcal{E}_{pk}$ *and any length-preserving* IND*-secure symmetric encryption scheme* $\mathtt{E}_K$*, our converted scheme* $\mathbb{E}_{pk}$ *is* IND-CCA2 *secure in the random oracle model.* □

To prove Theorem 5, we suppose that there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ able to break the INC-CCA2 security of $\mathbb{E}_{pk}$. We further suppose that $\mathtt{E}_K$ is (τ, ν)-secure in the sense of IND. From $\mathcal{A}$, we then exhibit an adversary $\mathcal{B}$ (*i.e.*, a reduction algorithm) that inverts $\mathcal{E}_{pk}$ using a plaintext-checking oracle, and thus breaks the OW-PCA security of $\mathcal{E}_{pk}$.

4.1 A useful lemma

The assumption that $\mathtt{E}_K$ is length-preserving and (τ, ν)-IND secure implies the following lemma.

Lemma 7. *Assume that* $\mathtt{E}_K$ *is a length-preserving* (τ, ν)-IND *symmetric encryption scheme, where* τ *denotes the time needed for evaluating* $\mathtt{E}_K(\cdot)$*. Then given a pair* (m, c) *of plaintext/ciphertext, we have*

$$\Pr_K \left[\mathtt{E}_K(m) = c\right] \leq \nu + \frac{1}{\sharp m} \quad .$$

Proof. Given the pair (m, c), we consider the following distinguisher $\mathcal{A}$. $\mathcal{A}$ randomly chooses a bit $d \in \{0,1\}$, sets $m_d = m$ and $m_{\neg d}$ to a random value m'. The pair $(m_d, m_{\neg d})$ is then sent to the encryption oracle which returns $c_b = \mathtt{E}_K(m_b)$ for a random key K and a random $b \in \{0,1\}$. $\mathcal{A}$ then checks if $c_b = c$, returns d if the equality holds and $\neg d$ otherwise. Letting ε the advantage of $\mathcal{A}$, we have

$$\begin{aligned}
\varepsilon &= 2 \Pr_{m',K,d,b}[\mathcal{A} \text{ returns } b] - 1 \\
&= 2 \Pr_{m',K,d,b}[(c_b = c) \wedge (d = b)] + \\
&\qquad 2 \Pr_{m',K,d,b}[(c_b \neq c) \wedge (\neg d = b)] - 1 \\
&= 2 \Pr_{m',K,d,b}[(\mathtt{E}_K(m) = c) \wedge (d = b)] + \\
&\qquad 2 \Pr_{m',K,d,b}[(\mathtt{E}_K(m') \neq c) \wedge (\neg d = b)] - 1 \\
&= \Pr_{K}[\mathtt{E}_K(m) = c] + \Pr_{m',K}[\mathtt{E}_K(m') \neq c] - 1 \\
&= \Pr_{K}[\mathtt{E}_K(m) = c] - \Pr_{m',K}[\mathtt{E}_K(m') = c] \leq \nu \ .
\end{aligned}$$

The proof follows by noting that as $\mathtt{E}_K$ is length-preserving, it permutes the set of messages for any key K and so $\Pr_{m',K}[\mathtt{E}_K(m') = c] = 1/\sharp\, m$. □

4.2 Description of the reduction algorithm

$\mathcal{B}$ is given a challenge encryption $y = \mathcal{E}_{pk}(\widetilde{w}, *)$, an oracle $\mathcal{O}^{\text{PCA}}$ which answers plaintext-checking requests on $\mathcal{E}_{pk}$, and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that breaks the IND-CCA2 security of $\mathbb{E}_{pk}$. $\mathcal{B}$'s goal is to retrieve all the bits of $\widetilde{w}$. Wlog, we assume that $\mathcal{O}^{\text{PCA}}$ responds to any of $\mathcal{B}$'s requests with no error and within a time bounded by τ_{PCA}.

Throughout, the following notations are used. For any predicate $\mathrm{R}(x)$, $\mathrm{R}(*)$ stands for $\exists x$ s.t. $\mathrm{R}(x)$. If $\mathcal{O}$ is an oracle to which $\mathcal{A}$ has access, we denote by *query* $\mapsto$ *response* the correspondence $\mathcal{O}$ establishes between $\mathcal{A}$'s request *query* and the value *response* returned to $\mathcal{A}$. $\textsc{Hist}\,[\mathcal{O}]$ stands for the set of correspondences established by $\mathcal{O}$ as time goes on: $\textsc{Hist}\,[\mathcal{O}]$ can be seen as an history tape which gets updated each time $\mathcal{A}$ makes a query to $\mathcal{O}$. We denote by $q_{\mathcal{O}}$ the number of calls $\mathcal{A}$ made to $\mathcal{O}$ during the simulation.

Overview of $\mathcal{B}$ At the beginning, $\mathcal{B}$ chooses a random value $\widetilde{K}$. $\mathcal{B}$ then runs $\mathcal{A}$ and provides a simulation for F, G, H and $\mathbb{D}_{sk}$. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ runs in two stages. At the end of the first stage (find stage), $\mathcal{A}_1$ outputs a pair (m_0, m_1). $\mathcal{B}$ then randomly chooses $b \in \{0,1\}$, computes $\widetilde{c_2} = \mathtt{E}_{\widetilde{K}}(m_b)$ and builds $(y, \widetilde{c_2})$. This challenge is provided to $\mathcal{A}_2$, which outputs some bit at the end of the second stage (guess stage). Once finished, $\mathcal{B}$ checks whether some $\widetilde{w}$ has been defined during the game. If so, $\widetilde{w}$ is returned as the inverse of $\mathcal{E}_{pk}$ on y; otherwise a failure answer is returned. The detailed description of the simulation follows.

Wlog, we assume that $\mathcal{A}$ keeps track of all the queries throughout the game so that $\mathcal{A}$ never has to make the same query twice to the same oracle.

Simulation of F For each new query (m, r),

(Event E_1) if processing guess stage and $m = m_b$ and there exists $s \mapsto h \in \text{Hist}[\text{H}]$ such that $y = \mathcal{E}_{pk}(s\|r \oplus h, *)$ then F sets $\widetilde{w} := s\|r \oplus h$, returns s and updates its history,
(no event) else F outputs a random value and updates its history.

Simulation of G For each new query (w, c_1),

(Event E_2) if $c_1 = y$ and $y = \mathcal{E}_{pk}(w, *)$ then G sets $\widetilde{w} := w$, returns $\widetilde{K}$ and updates its history,
(Event E_3) else if $c_1 \neq y$ and $y = \mathcal{E}_{pk}(w, *)$ then G sets $\widetilde{w} := w$, returns a random value and updates its history,
(no event) else G outputs a random value and updates its history.

Simulation of H For each new query s, H outputs a random value and updates its history.

Simulation of $\mathbb{D}_{sk}$ (plaintext extractor) For each new query (c_1, c_2), $\mathbb{D}_{sk}$ first checks (this verification step only stands while the guess stage $\mathcal{A}_2$ is running) that $(c_1, c_2) \neq (y, \widetilde{c_2})$ since if this equality holds, the query must be rejected as $\mathcal{A}$ attempts to decrypt its own challenge ciphertext. Then, $\mathbb{D}_{sk}$ tries to find the only (if any) message m matching the query. To achieve this, $\mathbb{D}_{sk}$ invokes the simulation of G, H and F provided by $\mathcal{B}$ as follows.

- Find the unique pair (r, s) such that $(*, r) \mapsto s \in \text{Hist}[\text{F}]$, $s \mapsto h \in \text{Hist}[\text{H}]$ and $c_1 = \mathcal{E}_{pk}(s\|r \oplus h, *)$. If such a pair exists,
  - query G to get $K = \text{G}(s\|r \oplus h, c_1)$,
  - letting $m = \mathtt{D}_K(c_2)$, query F to check if $\text{F}(m, r) = s$. If the equality holds, return m; otherwise reject the query (Event RJ_1).
- If the search for (r, s) is unsuccessful, check if there exists w with $(w, c_1) \mapsto K \in \text{Hist}[\text{G}]$ and $c_1 = \mathcal{E}_{pk}(w, *)$. If such an w exists,
  - define s and t by $w = s\|t$, and query H to get $h = \text{H}(s)$,
  - letting $m = \mathtt{D}_K(c_2)$, query F to check if $\text{F}(m, t \oplus h) = s$. If the equality holds, return m; otherwise reject the query (Event RJ_2).
- If the search for w is unsuccessful, reject the query (Event RJ_3).

4.3 Soundness of $\mathcal{B}$

Unless otherwise mentioned, all probabilities are taken over the random choices of $\mathcal{A}$ and $\mathcal{B}$.

Simulation of random oracles The plaintext $\widetilde{w}$ uniquely defines $\widetilde{s}$ and $\widetilde{t}$ such that $\widetilde{w} = \widetilde{s}\|\widetilde{t}$. We note $\widetilde{r}$ the random variable $\widetilde{t} \oplus \text{H}(\widetilde{s})$.

Soundness of F. The simulation of F fails when $(m_b, \widetilde{r})$ is queried and answered with some value $s \neq \widetilde{s}$ before $\widetilde{s}$ appears in $\textsc{Hist}\,[\mathrm{H}]$.

Let q_{F}^1 denote the number of oracle queries $\mathcal{A}_1$ made to F during the find stage. Since $\tilde{r}$ is a uniformly-distributed random variable throughout the find stage, we certainly have $\Pr[\mathrm{F}\ \mathsf{incorrect\ in\ the\ find\ stage}] \leq q_{\mathrm{F}}^1/\sharp\, r$.

Moreover, throughout the guess stage, $\mathcal{A}_2$ cannot gain any information about $\widetilde{r}$ without knowing $\mathrm{H}(\widetilde{s})$ because H is a random function. Hence, letting q_{F}^2 the number of oracle queries $\mathcal{A}_2$ made to F during the guess stage, we have $\Pr[\mathrm{F}\ \mathsf{incorrect\ in\ the\ guess\ stage}] \leq q_{\mathrm{F}}^2/\sharp\, r$.

Consequently, the probability that an error occurs while $\mathcal{B}$ simulates the oracle F is upper-bounded by

$$\Pr[\mathrm{F}\ \mathsf{incorrect}] \leq \frac{q_{\mathrm{F}}^1 + q_{\mathrm{F}}^2}{\sharp\, r} = \frac{q_{\mathrm{F}}}{\sharp\, r}\ .$$

Soundness of G. The simulation is perfect.

Soundness of H. The simulation is perfect.

Plaintext extraction The simulation of $\mathbb{D}_{sk}$ fails when $\mathbb{D}_{sk}$ returns $\perp$ although the query $c = (c_1, c_2)$ is a valid ciphertext. Let then m, r, s, t, h, w, K be the unique random variables associated to c in this case. Obviously, c was rejected through event RJ_3, because a rejection through RJ_1 or RJ_2 refutes the validity of c. Therefore, if $\mathbb{D}_{sk}$ is incorrect for c, we must have

$$(\underbrace{(m,r) \notin \textsc{Hist}\,[\mathrm{F}]}_{:=\neg\mathsf{E}_{\mathrm{F}}} \vee \underbrace{s \notin \textsc{Hist}\,[\mathrm{H}]}_{:=\neg\mathsf{E}_{\mathrm{H}}}) \wedge (\underbrace{(w, c_1) \mapsto K \notin \textsc{Hist}\,[\mathrm{G}]}_{:=\neg\mathsf{E}_{\mathrm{G}}})\ .$$

Hence,

$$\begin{aligned}
\Pr[\mathbb{D}_{sk}\ \mathsf{incorrect\ for}\ c] &= \Pr[(\neg\mathsf{E}_{\mathrm{F}} \vee \neg\mathsf{E}_{\mathrm{H}}) \wedge \neg\mathsf{E}_{\mathrm{G}}] \\
&= \Pr[((m,r) \neq (m_b, \widetilde{r})) \wedge (\neg\mathsf{E}_{\mathrm{F}} \vee \neg\mathsf{E}_{\mathrm{H}}) \wedge \neg\mathsf{E}_{\mathrm{G}}] + \\
&\quad \Pr[((m,r) = (m_b, \widetilde{r})) \wedge (\neg\mathsf{E}_{\mathrm{F}} \vee \neg\mathsf{E}_{\mathrm{H}}) \wedge \neg\mathsf{E}_{\mathrm{G}}] \\
&\leq \Pr[((m,r) \neq (m_b, \widetilde{r})) \wedge (\neg\mathsf{E}_{\mathrm{F}} \vee \neg\mathsf{E}_{\mathrm{H}})] + \Pr[((m,r) = (m_b, \widetilde{r})) \wedge \neg\mathsf{E}_{\mathrm{G}}] \\
&= \Pr[((m,r) \neq (m_b, \widetilde{r})) \wedge \neg\mathsf{E}_{\mathrm{F}}] + \Pr[((m,r) \neq (m_b, \widetilde{r})) \wedge (\mathsf{E}_{\mathrm{F}} \wedge \neg\mathsf{E}_{\mathrm{H}})] + \\
&\quad \Pr[((m,r) = (m_b, \widetilde{r})) \wedge \neg\mathsf{E}_{\mathrm{G}}]\ .
\end{aligned}$$

1. $\textsc{Assume}$ $(m,r) \neq (m_b, \widetilde{r})$ $\textsc{and}$ $(m,r) \notin \textsc{Hist}\,[\mathrm{F}]$. Since F is a random function, $\mathrm{F}(m,r)$ is a uniformly distributed random value unknown to $\mathcal{A}$. The fact that c is a valid ciphertext implies that $\mathrm{F}(m,r) = s$, which happens with probability

$$\Pr[c\ \mathsf{is\ valid} \wedge (m,r) \notin \textsc{Hist}\,[\mathrm{F}]] = \frac{1}{\sharp\, s}\ .$$

2. $\textsc{Assume}$ $(m,r) \neq (m_b, \widetilde{r})$ $\textsc{and}$ $(m,r) \in \textsc{Hist}\,[\mathrm{F}] \wedge s \mapsto h \notin \textsc{Hist}\,[\mathrm{H}]$. Suppose that $s \neq \widetilde{s}$. Since H is a random function, $\mathrm{H}(s)$ is a uniformly

distributed random value unknown to $\mathcal{A}$. The validity of c implies that $(m, t \oplus \mathrm{H}(s)) \mapsto s \in \textsc{Hist}\,[\mathrm{F}]$, which happens with probability

$$\Pr[(m, t \oplus \mathrm{H}(s)) \mapsto s \in \textsc{Hist}\,[\mathrm{F}]] \leq \frac{q_\mathrm{F}}{\sharp\, r} \ .$$

Now assume $s = \widetilde{s}$. In this case, we must have $(m, r) \mapsto \widetilde{s} \in \mathrm{F}$ which occurs with probability

$$\Pr[(m, r) \mapsto \widetilde{s} \in \textsc{Hist}\,[\mathrm{F}]] \leq \frac{q_\mathrm{F}}{\sharp\, s} \ .$$

3. $\textsc{Assume}\ (m, r) = (m_b, \widetilde{r})\ \textsc{and}\ (w, c_1) \mapsto K \notin \textsc{Hist}\,[\mathrm{G}]$. This implies $s = \widetilde{s}$, $t = \widetilde{t}$, $w = \widetilde{w}$ and $c_1 \neq y$. Hence, if c is valid, we must have $\mathtt{E}_K(m_b) = c_2$ for a uniformly distributed K. By virtue of Lemma 7, this is bounded by

$$\Pr_K[\mathtt{E}_K(m_b) = c_2] \leq \nu + \frac{1}{\sharp\, m} \ .$$

Gathering all preceding bounds, we get

$$\begin{aligned} &\Pr[c \text{ is valid} \wedge \mathbb{D}_{sk} \text{ incorrect for } c] \\ &\quad \leq \frac{1}{\sharp\, s} + q_\mathrm{F} \left(\frac{1}{\sharp\, r} + \frac{1}{\sharp\, s} \right) + \nu + \frac{1}{\sharp\, m} \,, \end{aligned}$$

which, taken over all queries of $\mathcal{A}$, leads to

$$\Pr[\mathbb{D}_{sk} \text{ incorrect}] \leq q_{\mathbb{D}_{sk}} \left(\frac{1}{\sharp\, s} + q_\mathrm{F} \left(\frac{1}{\sharp\, r} + \frac{1}{\sharp\, s} \right) + \nu + \frac{1}{\sharp\, m} \right) \ .$$

Conclusion We have

$$\begin{aligned} \Pr[\mathcal{B} \text{ incorrect}] &= \Pr[\mathrm{F} \text{ incorrect}] + \Pr[\mathbb{D}_{sk} \text{ incorrect}] \\ &\leq \frac{q_\mathrm{F}}{\sharp\, r} + q_{\mathbb{D}_{sk}} \left(\frac{1}{\sharp\, s} + q_\mathrm{F} \left(\frac{1}{\sharp\, r} + \frac{1}{\sharp\, s} \right) + \nu + \frac{1}{\sharp\, m} \right) \ . \end{aligned}$$

4.4 Reduction cost

Success probability Let us suppose that $\mathcal{A}$ distinguishes $\mathbb{E}_{pk}$ within a time bound τ with advantage ε in less than q_F, q_H, q_G, $q_{\mathbb{D}_{sk}}$ oracle calls. Defining $\Pr_2[\cdot] = \Pr[\cdot \mid \neg(\mathcal{B} \text{ incorrect})]$, this means that

$$\Pr_2[\mathcal{A} \to b] \geq \frac{1 + \varepsilon}{2} \ .$$

Suppose also that $\mathtt{E}_K$ is (τ, ν)-indistinguishable. Assuming that the oracles are correctly simulated, if none of the events E_1, E_2 or E_3 occurs, then $\mathcal{A}$ never asked $\widetilde{r}$ to the random oracle F, neither did it learn the key $\widetilde{K}$ under which m_b was encrypted in $\widetilde{c_2}$ (this is due to the randomness of F and G). This upper-limits the information leakage on b by ν, since $\mathcal{A}$'s running time is bounded by τ. Noting $\mathsf{E}_{win} = \mathsf{E}_1 \vee \mathsf{E}_2 \vee \mathsf{E}_3$, this implies

$$\Pr_2[\mathcal{A} \to b \mid \neg \mathsf{E}_{win}] \leq \frac{1 + \nu}{2} \ .$$

We then get

$$\frac{1+\varepsilon}{2} \leq \Pr_2[\mathcal{A} \to b] \leq \Pr_2[\mathcal{A} \to b \mid \neg \mathsf{E}_{win}] + \Pr_2[\mathsf{E}_{win}] \leq \frac{1+\nu}{2} + \Pr_2[\mathsf{E}_{win}]\,,$$

whence $\Pr_2[\mathsf{E}_{win}] \geq (\varepsilon - \nu)/2$. But $\Pr_2[\mathcal{B} \to \widetilde{w}] = \Pr_2[\mathsf{E}_{win}]$ and finally,

$$\begin{aligned}\Pr[\mathcal{B} \to \widetilde{w}] &\geq \Pr_2[\mathcal{B} \to \widetilde{w}] - \Pr[\mathcal{B} \text{ incorrect}] \\ &\geq \frac{\varepsilon - \nu}{2} - \frac{q_{\mathrm{F}}}{\sharp\, r} - \\ &\qquad q_{\mathbb{D}_{sk}} \left(\frac{1}{\sharp\, s} + q_{\mathrm{F}} \left(\frac{1}{\sharp\, r} + \frac{1}{\sharp\, s} \right) + \nu + \frac{1}{\sharp\, m} \right) \ .\end{aligned}$$

Hence $\mathcal{B}$ succeeds with non-negligible probability.

Total number of calls to $\mathcal{O}^{\text{PCA}}$ Checking that a pair of the form (w, y) satisfies $y = \mathcal{E}_{pk}(w, *)$ is done thanks to the plaintext-checking oracle $\mathcal{O}^{\text{PCA}}$. Therefore, oracle F makes at most $q_{\mathrm{F}} \cdot q_{\mathrm{H}}$ queries to $\mathcal{O}^{\text{PCA}}$, and oracle G makes at most $q_{\mathrm{G}} \cdot 1$ queries to $\mathcal{O}^{\text{PCA}}$. Moreover, it is easy to see that oracle $\mathcal{O}^{\mathbb{D}_{sk}}$ makes at most $q_{\mathbb{D}_{sk}}(q_{\mathrm{F}} + q_{\mathrm{G}})$ calls since in the worst case $\mathbb{D}_{sk}$ has to call $\mathcal{O}^{\text{PCA}}$ for all elements $((*, r) \mapsto s) \in \textsc{Hist}\,[\mathrm{F}]$ and for all elements $((w, c_1) \mapsto K) \in \textsc{Hist}\,[\mathrm{G}]$. In conclusion, the total number of calls actually needed by $\mathcal{B}$ is upper-bounded by

$$q_{\text{PCA}} \leq q_{\mathrm{F}}\, q_{\mathrm{H}} + q_{\mathrm{G}} + q_{\mathbb{D}_{sk}}(q_{\mathrm{F}} + q_{\mathrm{G}}) \ .$$

Total running time The reduction algorithm runs in time bounded by

$$\tau_{\mathcal{B}} = \tau + (q_{\mathrm{F}}\, q_{\mathrm{H}} + q_{\mathrm{G}} + q_{\mathbb{D}_{sk}}(q_{\mathrm{F}} + q_{\mathrm{G}}))\,(\tau_{\text{PCA}} + O(1)) \ .$$

This completes the proof of Theorem 5.

5 Concluding Remarks

A very popular way to (symmetrically) encrypt a plaintext is to use a *stream cipher*. The simplest example is the Vernam cipher where a plaintext m is processed bit-by-bit to form the ciphertext c under the secret key K. With this cipher, each plaintext bit m_i is XOR-ed with the key bit K_i to produce the ciphertext bit $c_i = m_i \oplus K_i$. If K is truly random and changes for each plaintext m being encrypted, then the system is unconditionally secure. This ideal situation is, however, impractical for a real-world implementation. To resolve the key management problem, a stream cipher is usually combined with a public-key cryptosystem. Contrary to the obvious solution consisting in encrypting a random session key with an asymmetric cryptosystem and then using that key with a stream cipher, our hybrid scheme achieves provable security. Compared to a purely asymmetric solution, our scheme presents the advantage to encrypt

long messages *orders of magnitude faster* thanks to the use of a symmetric cryptosystem.

Another merit of our scheme resides in its generic nature. The set of possible applications of the new conversion scheme is similar to that of REACT: it concerns *any* asymmetric function that is OW-PCA under a conjectured intractability assumption. A specificity of REACT is that it can operate "on the fly". The session key does not depend on the plaintext to be encrypted and can thus be computed in advance. This property is particularly advantageous when the asymmetric encryption is expensive, as is the case for discrete-log based cryptosystems. Remark that our scheme does not allow "on-the-fly" encryption, that was the price to pay for shortening the ciphertext.

In other cases such as for RSA (with a low encryption exponent, *e.g.*, 3 or $2^{16}+1$) or Rabin cryptosystem, on-the-fly encryption is not an issue and our scheme may be preferred because the resulting ciphertext is shorter. Furthermore, it is worth noting that for a deterministic asymmetric encryption scheme $\mathcal{E}_{pk}$, the notions of OW-PCA and OW-CPA are identical: the validity of a pair (m, c) of plaintext/ciphertext can be publicly checked as $c = \mathcal{E}_{pk}(m)$. So, our method allows one to construct, for example, an efficient IND-CCA2 hybrid encryption scheme whose security relies on the hardness of inverting the RSA function or factoring large numbers.

In conclusion, our generic conversion may be seen as the best alternative to REACT when the underlying asymmetric encryption is relatively fast or when memory/bandwidth savings are a priority.

References

1. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. Full paper (30 pages), February 1999. An extended abstract appears in H. Krawczyk, ed., Advances in Cryptology – CRYPTO '98, volume 1462 of Lecture Notes in Computer Science, pages 26–45, Springer-Verlag, 1998.
2. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
3. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In A. De Santis, editor, *Advances in Cryptology – EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer-Verlag, 1995.
4. Victor Boyko. On the security properties of OAEP as an all-or-nothing transform. Full paper (28 pages), August 1999. An extended abstract appears in M. Wiener, ed., Advances in Cryptology – CRYPTO '99, volume 1666 of Lecture Notes in Computer Science, pages 503–518, Springer-Verlag, 1999.
5. Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. *IEICE Transaction on of Fundamentals of Electronic Communications and Computer Science* **E83-A**(1): 24–32, January 2000.
6. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer-Verlag, 1999.
7. Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274. Springer-Verlag, 2001.
8. Oded Goldreich. On the foundations of modern cryptography. In B. Kaliski, editor, *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 46–74. Springer-Verlag, 1997.

9. Oded Goldreich. *Modern cryptography, probabilistic proofs and pseudo-randomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, 1999.
10. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
11. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM Annual Symposium on the Theory of Computing (STOC'90)*, pages 427–437. ACM Press, 1990.
12. Tatsuaki Okamoto and David Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In D. Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 159–175. Springer-Verlag, 2001.
13. David Pointcheval. Chosen-ciphertext security for any one-way cryptosystem. In H. Imai and Y. Zheng, editors, *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 129–146. Springer-Verlag, 2000.
14. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer-Verlag, 1992.
15. Ronald L. Rivest. All-or-nothing encryption and the package transform. In E. Biham, editor, *Fast Software Encryption*, volume 1267 of *Lecture Notes in Computer Science*, pages 210–218. Springer-Verlag, 1997.

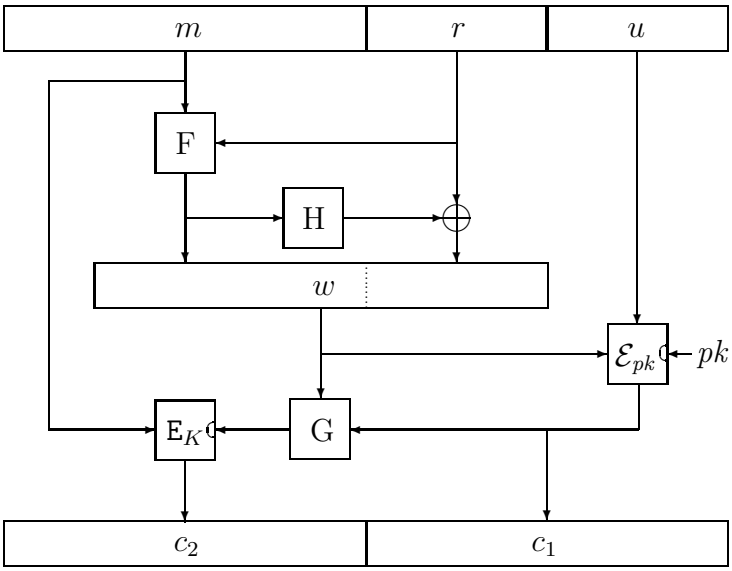

Fig. 1. Description of GEM in encryption mode.

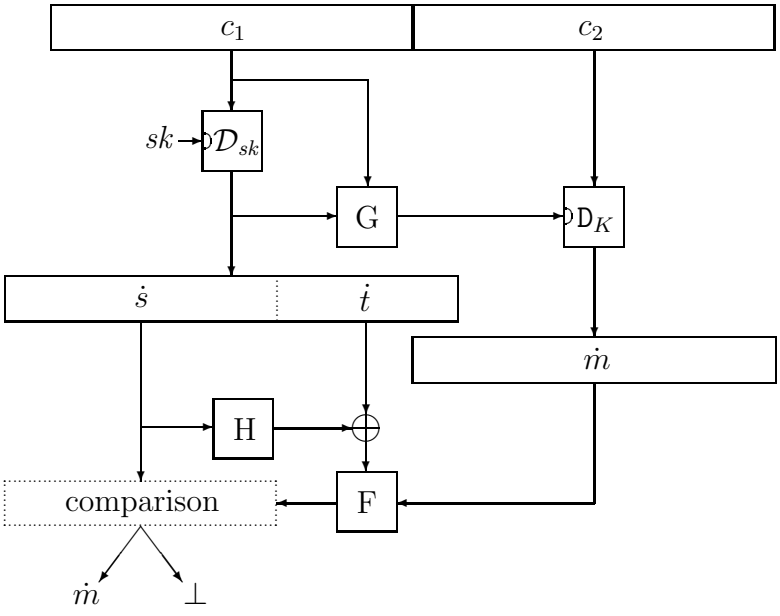

Fig. 2. Description of GEM in decryption mode.