International Journal of Information Security manuscript No.
(will be inserted by the editor)

Fujisaki-Okamoto Hybrid Encryption Revisited

David Galindo, Sebastià Martín, Paz Morillo, Jorge L. Villar

Dep. Matemàtica Aplicada IV. Universitat Politècnica de Catalunya
Campus Nord, c/Jordi Girona, 1-3, 08034 Barcelona
e-mail: {dgalindo,sebasm,paz,jvillar}@mat.upc.es

Abstract At Crypto'99, Fujisaki and Okamoto [11] presented a generic transformation from weak secure asymmetric and symmetric schemes into an IND-CCA hybrid encryption scheme in the Random Oracle Model, which has been extensively used in several cryptographic scenarios. The work we present here forms part of the careful revision of the provable security techniques initiated by Shoup in [25], in so far as we find some ambiguities in the proof of this generic conversion, which can lead to false claims. Consequently, the original conversion is modified and the class of asymmetric primitives that can be used is shortened. Furthermore, the concept of *Easy Verifiable Primitive* is formalized, showing its connection with the gap problems introduced in [18]. Using these ideas, a *completely new* security proof for the modified transformation is given, which is phrased using currently widely accepted techniques. The reduction thereby obtained turns out to be *tight*, enhancing the concrete security claimed in the original work for the Easy Verifiable Primitives. For the rest of primitives, the concrete security is improved at the cost of stronger assumptions. Finally, the resistance of the new conversion against reject timing attacks is addressed.

Key words public-key cryptography, chosen-ciphertext security, tight reduction, Random Oracle Model, Okamoto-Uchiyama scheme.

1 Introduction

When developing a new public key encryption scheme, there are two basic criteria that a designer wishes to ensure: *security* and *efficiency*. Security is obviously the main concern, and it is expressed in terms of an attacker's goal against the scheme and the means it uses. The standard security notion for a

general purpose cryptosystem is *indistinguishability against adaptive chosen ciphertext attacks*, IND-CCA for short. Proofs of security are accepted only if they are in the *provable security* model, in which security is polynomially reduced to trusted mathematical assumptions. Regarding efficiency, there are two main aspects to consider. On one hand, the computational complexity of the algorithms involved in the scheme and, on the other hand, the concrete security of the scheme; that is, how the security of the scheme is related to the computational assumptions on which it is based. There are other features of relevance, such as the design simplicity or the length of the messages that can be encrypted.

However, developing a practical provably secure cryptosystem in the sense of IND-CCA is a quite difficult task. In fact, few such schemes are known in the standard model, the exceptions being the schemes designed in the Cramer-Shoup paradigm [8]. In the idealized Random Oracle Model [2], several powerful generic constructions have been designed [11,22,19,6], providing practical IND-CCA schemes from weak asymmetric and symmetric schemes. The proposal by Fujisaki and Okamoto we revisit here is by far the most known conversion.

Among these constructions, [19,6] present a better security reduction than [11,22]. This is mainly due to the use of the *plaintext checking oracle* introduced in [18]. The cost of using this oracle is that the security of the encryption scheme is in general based on (stronger) gap assumptions, when the asymmetric primitive is probabilistic.

But, as recently shown, unexpected difficulties were hidden in the development of secure schemes, in so far as the use of provable security has proved to be more subtle than it was expected. The first example of this fact was the claim by Shoup [25] against the widely believed IND-CCA security of OAEP when applied to a trapdoor permutation. From this and other findings (see [26] for a nice account), we are aware that there are ambiguities and misconceptions in the security model, which can lead to false claims.

Our work forms part of this careful treatment of the provable security techniques, as we applied it to the widely used generic conversion by Fujisaki and Okamoto (FO) presented at Crypto'99. The particular instantiation of this conversion with the Okamoto-Uchiyama scheme [20], known as EPOC-2 [10], has found practical attacks that lead to a total break [15,9,24]. The most serious flaw was found in [15], where the secret key was recovered in the IND-CCA game itself. The authors of [15] pointed out that such a surprising result was related to the vagueness of the IND-CCA model when dealing with invalid ciphertexts. In the case of the original specification of EPOC-2, an attacker could obtain vital information about the system from those ciphertexts. The other attacks mentioned above ([9,24]), make use of extra information available in the real world, such as the running time of the decryption algorithm. This enables us to distinguish among the reasons for rejecting certain ciphertexts, and is used to launch an attack recovering

the secret key again.

Our results. We incorporate the comments made by the authors of EPOC about FO conversion in the appendix to [15]. Then we show that some ambiguities still remain in the proof of security, with the outcome that the security result claimed in [11] cannot be guaranteed in general. This obliges us to slightly modify the conversion and to restrict the class of asymmetric primitives that can be used.

Furthermore, the concept of *Easy Verifiable Primitive* is formalized, and it is used to give a *new* security proof for the modified transformation. We show that the reduction is *tight*, improving the concrete security claimed in the original work for the Easy Verifiable Primitives. For the rest of primitives, the concrete security is improved at the cost of a stronger assumption; that is, a gap assumption (see [18]).

Finally, the resistance of the new conversion against reject timing attacks is addressed. Since the vulnerability of a scheme against these attacks is closely related to the design of the rejection rules in the decryption algorithm, we take this into account when drawing the modification.

2 Preliminaries

In this section, we recall some technical details and notations used in the rest of the paper.

Algorithmic notation. Assigning a value a to a variable x will be in general denoted by $x \leftarrow a$. Nevertheless, this notation can be extended to allow different meanings. If A is a non-empty set, then $x \leftarrow A$ denotes that x has been uniformly chosen in A. If D is a probability distribution over A, then $x \leftarrow D$ means that x has been chosen in A by sampling the distribution D. Finally, if $\mathcal{A}$ is an (probabilistic) algorithm, $x \leftarrow \mathcal{A}$ means that $\mathcal{A}$ has been executed on some specified input and its (random) output has been assigned to the variable x.

Negligible functions. The class of negligible functions on a parameter $\ell \in \mathbb{Z}^+$, denoted as $\mathsf{negl}(\ell)$, is the set of the functions $\epsilon : \mathbb{Z}^+ \to \mathbb{R}^+$ such that, for any polynomial $p \in \mathbb{R}[\ell]$, there exist $M \in \mathbb{R}^+$ such that $\epsilon(\ell) < \frac{M}{p(\ell)}$ for all $\ell \in \mathbb{Z}^+$. Let $\mathsf{poly}(\ell)$ the class of functions $p : \mathbb{Z}^+ \to \mathbb{R}^+$ upper bounded in $\mathbb{Z}^+$ by some polynomial in $\mathbb{R}[\ell]$.

Set sequences. As usual, $\{0,1\}^\star$ and $\{0,1\}^\ell$ will respectively denote the set of all finite binary strings and the set of binary strings with length ℓ. A string set sequence, $X = \{X_\ell\}_{\ell \in \mathbb{Z}^+}$, is a *polynomial size* set if there exist an integer valued function $p_X(\ell) \in \mathsf{poly}(\ell)$ such that $X_\ell \subseteq \{0,1\}^{p_X(\ell)}$ for all $\ell \in \mathbb{Z}^+$. A polynomial size set, X, is *samplable* if there exists a probabilistic polynomial time algorithm (PPT) that on input 1^ℓ, outputs a uniformly distributed random element in X_ℓ. Moreover, X is *recognizable* if there exists a polynomial time algorithm (PT) that on input 1^ℓ and a string

s, with size polynomial in ℓ, outputs 1 if and only if $s \in X_\ell$. The cardinality of a set sequence A (as a function of ℓ) will be denoted by $|A|$.

These notions can easily be extended to non-strings sets by using polynomial size injective encoding maps. In the sequel, the use of natural encodings (e.g. binary representations of integers) is assumed when necessary.

Hereafter, the word 'sequence' in 'set sequence', 'map sequence' and 'probability distribution sequence' will be omitted.

Keypair generators. Let PK and SK polynomial size sets such that the sets PK_ℓ are all disjoint and there exists a PT algorithm that on input $sk \in SK_\ell$ outputs an element $pk \in PK_\ell$. Suppose that there also exists a PT algorithm that on input $pk \in PK_\ell$ outputs ℓ. Let I a polynomial time samplable probability distribution over $PK \times SK$. The triple (PK, SK, I) will be called a *keypair generator*.

Set and map families. Given a keypair generator, the family $\{X_{pk}\}_{pk \in PK}$ is referred to as the *set family* X. In the same way, a *map family* f is defined as $\{f_{pk} : X_{pk} \to Z_{pk}\}_{pk \in PK}$. Given a set family, X, the cardinality $|X|$ as a function of ℓ is defined as the maximal value of $|X_{pk}|$, where $pk \in PK_\ell$.

A set family X is *recognizable* if there exist a PT algorithm that on input $pk \in PK$ and a string s, with size polynomial in ℓ, outputs 1 if and only if $s \in X_{pk}$. A conjectured counterexample is the set family $X_{pk} = Q_n$ of the quadratic residues modulo $n = pq$ (p, q different primes with length ℓ). However, if $sk = (p, q)$ is also provided then there exists an efficient way to recognise the elements in Q_n.

3 Easy verifiable functions

Firstly, recall the definition of a trapdoor one-way function family.

Definition 1 *Let (PK, SK, I) be a keypair generator. Let X and Z be polynomial size set families. Let $f : X \to Z$ be a family of injective maps and $g = \{g_{sk} : Z_{pk} \to X_{pk}\}_{sk \in SK}$ the family of their inverses, i.e. $g_{sk}(f_{pk}(x)) = x$ for all possible pairs (pk, sk) generated by I and for all $x \in X_{pk}$. The map family, f, is called a* Trapdoor One-Way (TOW) *function family (with respect to the probability distribution I) if*

1. *there exists a* PT *algorithm that on input (pk, x) outputs $f_{pk}(x)$ for all $pk \in PK$ and $x \in X_{pk}$.*
2. *there exists a* PT *algorithm that on input (sk, z) outputs $g_{sk}(z)$ for all $sk \in SK$ and $z \in Z_{pk}$.*
3. *for any* PPT *algorithm $\mathcal{A}^{\mathsf{OW}}$,*

$$\mathsf{Pr}\left[\mathcal{A}^{\mathsf{OW}}(pk, f_{pk}(x)) = x \;\middle|\; (pk, sk) \leftarrow I_\ell;\ x \leftarrow X_{pk}\right] \in \mathsf{negl}(\ell)$$

The following definition, based on [22], is somewhat related to the notion of probabilistic one-way (OW) encryption.

Definition 2 *Let (PK, SK, I) be a keypair generator. Let X, Y and Z be polynomial size set families. Let $f : X \times Y \to Z$ be a family of injective maps and $g = \{g_{sk} : Z_{pk} \to X_{pk}\}_{sk \in SK}$ the family of their partial inverses, i.e. $g_{sk}(f_{pk}(x, y)) = x$ for all possible pairs (pk, sk) generated by I and for all $x \in X_{pk}$ and $y \in Y_{pk}$. The map family, f, is called a* Trapdoor Partial One-Way (TPOW) *function family (with respect to the probability distribution I) if*

1. *there exists a* PT *algorithm that on input (pk, x, y) outputs $f_{pk}(x, y)$ for all $pk \in PK$, $x \in X_{pk}$ and $y \in Y_{pk}$.*
2. *there exists a* PT *algorithm that on input (sk, z) outputs $g_{sk}(z)$ for all $sk \in SK$ and $z \in Z_{pk}$.*
3. *for any* PPT *algorithm $\mathcal{A}^{\mathsf{POW}}$,*

$$\mathsf{Pr}\left[\mathcal{A}^{\mathsf{POW}}(pk, f_{pk}(x, y)) = x \;\middle|\; (pk, sk) \leftarrow I_\ell;\ x \leftarrow X_{pk};\ y \leftarrow Y_{pk}\right] \in \mathsf{negl}(\ell)$$

The last condition can be reformulated in terms of the game

Game POW()
1 $(pk, sk) \leftarrow I_\ell$
2 $x \leftarrow X_{pk}$; $y \leftarrow Y_{pk}$
3 $x' \leftarrow \mathcal{A}^{\mathsf{POW}}(pk, f_{pk}(x, y))$

and the probability $\mathsf{Succ}\left[\mathcal{A}^{\mathsf{POW}}\right] = \mathsf{Pr}\left[x' = x\right] \in \mathsf{negl}(\ell)$.

Notice that the concept of TOW function family can be seen as a particular case of TPOW, in which $|Y| = 1$.

Starting from a TPOW family, f, a probabilistic one-way cryptosystem, $(\mathsf{KeyGen}^f, \mathsf{Enc}^f, \mathsf{Dec}^f)$, is obtained in the following way: the keys $(pk, sk) = \mathsf{KeyGen}^f(1^\ell)$ are generated by using the sampling algorithm for I; the ciphertext for a message $x \in X_{pk}$ with randomness $y \leftarrow Y_{pk}$ is $c = \mathsf{Enc}^f(pk, x) = f_{pk}(x, y)$ and a valid ciphertext $z \in Z_{pk}$ is decrypted by means of $\mathsf{Dec}^f(sk, c) = g_{sk}(c)$. Note that we are implicitly assuming that Y is samplable.

New kinds of attacks and computational problems have been introduced and various applications found in the context of probabilistic cryptosystems (cf [18,19]). In this new scenario, the attacker has access to a *plaintext checking oracle* that checks if a given ciphertext z is an encryption of a given message x.

Definition 3 *A* plaintext checking oracle $\mathcal{O}_{\mathsf{PC}}$ *for a* TPOW *family $f : X \times Y \to Z$, is an oracle such that for a query (pk, x, z), where $pk \in PK$, $x \in X_{pk}$ and $z \in Z_{pk}$, $\mathcal{O}_{\mathsf{PC}}$ answers 1 if there exists $y \in Y_{pk}$ such that $f_{pk}(x, y) = z$, and 0 otherwise. (It is assumed that if x or z are outside their domains, the oracle also answers 0.)*

The new attack is called *Plaintext Checking Attack* (PCA), and it can be reformulated in terms of trapdoor partial one-way functions.

Definition 4 *A* TPOW *function family f is* Partial One-Way against Plaintext Checking Attacks (POW-PCA) *if it is a* TPOW *function even when access to a plaintext checking oracle $\mathcal{O}_{\mathsf{PC}}$ for f is given.*

This notion is stronger than partial one-wayness, since now the adversary is provided with extra computational resources. Now we formalize the concept of *easy verifiability*, informally described in [22], which captures the situation where there exists an efficient algorithm that *verifies* if a pair (x, z) is correct; that is, the algorithm implements a plaintext checking oracle.

Definition 5 *A map family f is* easy verifiable *if it is a* TPOW *family and there exists a (deterministic)* PT *algorithm $\mathcal{V}$, called* plaintext checking algorithm*, with the same input-output behaviour as the plaintext checking oracle for f.*

Obviously, if f is easy verifiable then the plaintext checking oracle for f can be replaced by the algorithm $\mathcal{V}$, without introducing any modification in the adversary's model of computation. These functions are very interesting, since

Lemma 6 *If the map family f is easy verifiable then it is* POW-PCA.

4 Some examples of easy verifiable functions families

It is straightforward to modify a TOW function family $\tilde{f} : X \to \tilde{Z}$ to obtain an easy verifiable function family f. To do this, simply take $Y = \{0,1\}^{p(\ell)}$, where $p(\ell) \in \mathsf{poly}(\ell)$, and define $f_{pk}(x, y) = (\tilde{f}_{pk}(x), y)$, that is, leaving y "in the clear".

For an arbitrary TPOW function family a plaintext checking algorithm could not exist. For instance, this is supposed to be the case for El Gamal and Okamoto-Uchiyama functions. In this situation, we are forced to base POW-PCA on a gap problem, which is a stronger assumption (cf [18,19]).

RSA-Paillier trapdoor bijection defined in [5] can be viewed as a non-trivial example of an easy verifiable function. A generalization of this function is presented below.

4.1 Easy verifiable function families from RSA-Paillier

Let (n, e) be a RSA public key for security parameter ℓ, that is $n = pq$, $e < n$ and p and q are different primes with length ℓ such that $\gcd(e, (p-1)(q-1)) = 1$. For any integer $r > 1$ with size polynomial in ℓ, consider the subset $\Omega_{n,r} \subset \mathbb{Z}_{nr}$ defined as $\Omega_{n,r} = \mathbb{Z}_n^\star + n\mathbb{Z}_r$. Then, the function family

$$\begin{aligned} f_{n,r,e} : \mathbb{Z}_n^\star \times \mathbb{Z}_r &\longrightarrow \Omega_{n,r} \\ (x, y) &\longrightarrow x^e + ny \bmod nr \end{aligned}$$

is a trapdoor bijection family, for $pk = (n, r, e)$ and $sk = (p, q, r, d)$, where d is the inverse of e modulo $(p-1)(q-1)$.

This function is well defined since $z \in \Omega_{n,r}$ iff $z \bmod n \in \mathbb{Z}_n^\star$. Note that $f_{n,r,e}$ is a bijection. Suppose that $f_{n,r,e}(x_0, y_0) = f_{n,r,e}(x_1, y_1)$ for some

x_0, y_0, x_1 and y_1. Reducing the equality modulo n we get $x_0^e = x_1^e \bmod n$, and then $x_0 = x_1 \bmod n$. This implies $ny_0 = ny_1 \bmod nr$, so $y_0 = y_1 \bmod r$ and the function $f_{n,r,e}$ is injective. Finally, given (p, q, r, d), to invert $f_{n,r,e}$ on input $z = f_{n,r,e}(x, y)$, it suffices to compute $x = z^d \bmod n$. Then, y is easily obtained from the equation $ny = z - x^e \bmod nr$. This shows $f_{n,r,e}$ is exhaustive, and therefore it is a bijection.

The above implies that there exist two PT algorithms that compute both $f_{n,r,e}$ and its partial inverse.

Let $\mathcal{G}_\ell$ denote the probability distribution of RSA public keys for security parameter ℓ; that is $(n, e) \leftarrow \mathcal{G}_\ell$ where $n = pq$, $e < n$ and p and q are different primes with length ℓ such that $\gcd(e, (p-1)(q-1)) = 1$.

Proposition 7 *The partial one-wayness of the bijection family $f_{n,r,e}$ is tightly equivalent to the one-wayness of the map family $RSA[n, e](x) = x^e \bmod n$.*

Proof:
$\Rightarrow$) Assume that for some ℓ and r there exists a PPT algorithm, $\mathcal{A}$, breaking the partial one-wayness of $f_{n,r,e}$ in time T and probability ϵ, i.e.

$$\mathsf{Pr}\left[\mathcal{A}(n, r, e, x^e + ny \bmod nr) = x \mid (n, e) \leftarrow \mathcal{G}_\ell;\ x \leftarrow \mathbb{Z}_n^\star;\ y \leftarrow \mathbb{Z}_r\right] = \epsilon$$

The following PPT algorithm, $\mathcal{B}$, can be used to invert the $RSA[n, e]$ function in time $T + O(\ell^2)$ with probability at least ϵ:

$\mathcal{B}(n, e, z)$
1 $y \leftarrow \mathbb{Z}_r$, $z' = z + ny \bmod nr$
2 $x \leftarrow \mathcal{A}(n, r, e, z')$
3 return x

Then, $\mathsf{Pr}\left[\mathcal{B}(n, e, x^e \bmod n) = x \mid (n, e) \leftarrow \mathcal{G}_\ell;\ x \leftarrow \mathbb{Z}_n^\star\right] \geq \epsilon$.
$\Leftarrow$) Trivial. □

Note that the hardness of inverting $RSA[n, e]$ could depend on the probability distribution $\mathcal{G}$. In practice, e is often a priori fixed (e.g. in PGP, $e = 17$) and p, q are uniformly distributed primes with length ℓ such that $p, q \neq 1$ modulo e.

Proposition 8 *The bijection family $f_{n,r,e}$ is easy verifiable.*

Proof: A simple plaintext checking algorithm works as follows. On input (n, r, e, x, z) first, verify if $x \in \mathbb{Z}_n^\star$ and $z \in \Omega_{n,r}$, that is, $z < nr$ and $z \bmod n \in \mathbb{Z}_n^\star$. Then, check if the equation $x^e \equiv z \pmod n$ holds. □

4.2 Easy verifiable function families from pairings

In this subsection, a second non-trivial example of an easy verifiable family is derived from ElGamal encryption, by taking advantage of non-degenerate

bilinear maps to solve the Decisional Diffie-Hellman problem. Currently, the only known efficiently-computable non-degenerate bilinear maps are the modified Weil pairing and the Tate pairing [16].

Let $E(\mathbb{F}_p)$ be the group of points of an elliptic curve over the finite field $\mathbb{F}_p$, with a bilinear non-degenerate function,

$$e : G \times G \longrightarrow G'$$

where G is the subgroup generated by a point $P \in E(\mathbb{F}_p)$ with prime order q and G' is a suitable group. Let us suppose that there is no affine point in $E(\mathbb{F}_p)$ with null x-coordinate. This is accomplished by choosing $a, b \in \mathbb{F}_p$ such that b is not a quadratic residue and using the Weierstrass equation, for characteristic different from 2 and 3, $y^2 = x^3 + ax + b$ to define the elliptic curve.

Let $W = sP$ be the s-multiple of the point P, for some secret value $s \in \mathbb{Z}_q^\star$. Let us consider the function family

$$\begin{aligned} f_W : \mathbb{F}_p^\star \times \mathbb{Z}_q^\star &\longrightarrow (G \setminus \{\mathcal{O}\}) \times \mathbb{F}_p^\star \\ (x, y) &\longrightarrow (yP, (yW)_x x) \end{aligned}$$

where $\mathcal{O}$ stands for the point at infinity and $(yW)_x$ stands for the x-coordinate of the point yW. Then, f_W is a trapdoor bijection family, for $pk = (p, a, b, q, P, W)$ and $sk = (p, a, b, q, P, s)$.

The function f_W is clearly injective. To show the bijectivity of f_W it suffices to compute the preimage (x, y) of any $(Q, r) \in (G \setminus \{\mathcal{O}\}) \times \mathbb{F}_p^\star$ in the following way. $x = ((sQ)_x)^{-1} r$ and y is just the discrete logarithm of Q with respect to P. The computation of x can be done in polynomial time if sk is given. However, there is no known method to compute y in polynomial time.

This shows that there exist two PT algorithms that compute both f_W and its partial inverse.

Let $\mathcal{G}_\ell$ denote the probability distribution of instances for the Computational Diffie-Hellman (CDH) problem over elliptic curves with a non-degenerate bilinear map. More precisely, for some $\ell \in \mathbb{Z}^+$ let $(p, a, b, q, P, W) \leftarrow \mathcal{G}_\ell$ with the following restrictions:

- p is a prime (power) with length polynomial in ℓ such that the characteristic of $\mathbb{F}_p$ is different from 2 and 3.
- $a, b \in \mathbb{F}_p$ such that b is not a quadratic residue in $\mathbb{F}_p$ and $4a^3 + 27b^2 \neq 0$.
- q is a prime with length ℓ and P is a point of order q on the elliptic curve $E(\mathbb{F}_p)$, defined by the Weierstrass equation $y^2 = x^3 + ax + b$.
- There is a unique subgroup G of order q in $E(\mathbb{F}_p)$, i.e. the subgroup generated by P.
- there is a non-degenerate polynomial-time computable bilinear map $e : G \times G \longrightarrow G'$, for a suitable group G'.
- $W \in G$, different from the point at infinity.

Proposition 9 *The partial one-wayness of the bijection family f_W is tightly equivalent to the hardness of some instance of the Computational Diffie-Hellman Problem (CDH).*

Proof:
$\Rightarrow$) Assume that for some ℓ there exists a PPT algorithm, $\mathcal{A}$, breaking the partial one-wayness of f_W in time T and probability ϵ, i.e.

$$\mathsf{Pr}\left[\mathcal{A}(p,a,b,q,P,W,yP,(yW)_x x) = x \mid (p,a,b,q,P,W) \leftarrow \mathcal{G}_\ell;\ x \leftarrow \mathbb{F}_p^\star;\ y \leftarrow \mathbb{Z}_q^\star\right] = \epsilon$$

The following PPT algorithm, $\mathcal{B}$, can be used to solve CDH in time $T + O(\ell^3) + 2T[e]$ with probability at least ϵ, where $T[e]$ stands for the time involved in the computation of the bilinear map e:

$\mathcal{B}(p,a,b,q,P,Q,W)$
1 $r \leftarrow \mathbb{F}_p^\star$
2 $x \leftarrow \mathcal{A}(p,a,b,q,P,W,Q,r)$
3 $T_x \leftarrow rx^{-1}$
4 $T_y \leftarrow \mathrm{sqrt}(T_x^3 + aT_x + b)$
5 $T \leftarrow (T_x, T_y)$
6 if $e(P,T) = e(Q,W)$; return T; endif
7 return $-T$

where, $\mathrm{sqrt}(z)$ stands for an algorithm that computes one of the two square roots of z in $\mathbb{F}_p$.

Then, $\mathsf{Pr}\left[\mathcal{B}(p,a,b,q,P,yP,W) = yW \mid (p,a,b,q,P,W) \leftarrow \mathcal{G}_\ell;\ y \leftarrow \mathbb{Z}_q^\star\right] \geq \epsilon$, since if $\mathcal{A}$ succeds then $T_x = (yW)x$. Thus, $T = yW$ (so, $e(P,T) = e(Q,W)$) or $T = -yW$.
$\Leftarrow$) Trivial, computing yW from P, yP and W. □

Proposition 10 *The bijection family f_W is easy verifiable.*

Proof: The plaintext checking algorithm works as follows. On input

$$(p,a,b,q,P,W,x,Q,r),$$

firstly verify if $x, r \in \mathbb{F}_p^\star$ and $Q \in G$, i.e. qQ is the point at infinity. Then, compute the point $T = (T_x, T_y)$ such that $T_x = rx^{-1}$ and $T_y = \mathrm{sqrt}(T_x^3 + aT_x + b)$. Now, the existence of $y \in \mathbb{Z}_q^\star$ such that $r = (yW)_x x$ is equivalent to $e(P,T) = e(Q,W)$ or $e(P,T)e(Q,W) = 1$. Notice that the first equality implies that $T = yW$ and the second one implies that $T = -yW$. □

5 Encryption security

Let us briefly recall some definitions about the security of both symmetric and asymmetric encryption.

5.1 Symmetric encryption

Let K and M be two (samplable and recognizable) polynomial size sets that respectively denote the key and message spaces. Let us consider a symmetric encryption scheme $\mathcal{E}^{sym} = (\mathsf{KeyGen}^{sym}, \mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$, over these sets, with the following properties.

- KeyGen^{sym} is a PPT algorithm that on input 1^ℓ outputs an uniformly distributed element in K_ℓ.
- Enc^{sym} and Dec^{sym} are PT algorithms with inputs in $K_\ell \times M_\ell$ and outputs in M_ℓ. Denote $\mathsf{Enc}_k^{sym}(m) = \mathsf{Enc}^{sym}(k, m)$ and $\mathsf{Dec}_k^{sym}(c) = \mathsf{Dec}^{sym}(k, c)$. For each $k \in K_\ell$, Enc_k^{sym} is a bijection on M_ℓ and Dec_k^{sym} is its inverse.
- For each pair $(m, c) \in M_\ell \times M_\ell$ there are at most γ values of $k \in K_\ell$ such that $c = \mathsf{Enc}_k^{sym}(m)$.

Such a cryptosystem has *indistinguishability of encryptions* (IND-SYM), also called Find-Guess security in [11], if any couple of PPT algorithms $\mathcal{A}^{\mathsf{IND-SYM}} = (\mathcal{A}_1, \mathcal{A}_2)$ (called "finding" and "guessing" stages of the adversary) have negligible advantage in the following game:

Game IND-SYM()
1 $b \leftarrow \{0, 1\}$
2 $(m_0, m_1, s) \leftarrow \mathcal{A}_1(1^\ell)$
3 $k \leftarrow K_\ell$; $c^\star = \mathsf{Enc}_k^{sym}(m_b)$
4 $b' \leftarrow \mathcal{A}_2(s, c^\star)$

That is, $\mathcal{E}^{sym}$ is IND-SYM if and only if for all $\mathcal{A}^{\mathsf{IND-SYM}}$,

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right] = |2\mathsf{Pr}\left[b' = b\right] - 1| = |\mathsf{Pr}\left[b' = b\right] - \mathsf{Pr}\left[b' \neq b\right]| \in \mathsf{negl}(\ell)$$

The messages m_0 and m_1 generated by $\mathcal{A}_1$ must be in M_ℓ.

Note that this is a very weak concept of security, but it is all we require in this paper to build a hybrid cryptosystem.

5.2 Asymmetric encryption

Let (PK, SK, I) be a keypair generator, as defined in Section 2. Let us consider an asymmetric encryption scheme $\mathcal{E}^{asym} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ over the polynomial size set families M, R and C, with the following properties:

- The keys $(pk, sk) = \mathsf{KeyGen}(1^\ell)$ are generated by using the (PPT) sampling algorithm for I.
- Enc is a (PPT) encryption algorithm which, on inputs public key $pk \in PK$ and $m \in M_{pk}$, runs on a randomness $r \in R_{pk}$ and returns a ciphertext $c \in C_{pk}$.

– Dec is a (PT) decryption algorithm that, on inputs secret key $sk \in SK$, and a polynomial size string c, returns a string m. We require that if $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\ell)$, then $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m, r)) = m$ for all $(m, r) \in M_{pk} \times R_{pk}$.

The following security notion for a general purpose asymmetric encryption scheme was developed in the works [13,17,23]. We say $\mathcal{E}^{asym}$ has *indistinguishability of encryptions under a chosen ciphertext attack* ($\mathsf{IND\text{-}CCA}$), if any pair of PPT algorithms $\mathcal{A}^{\mathsf{IND-CCA}} = (\mathcal{A}_1, \mathcal{A}_2)$ have negligible advantage in trying to distinguish the encryptions of two selected messages, with access to a couple of decryption oracles $\mathcal{D}_{sk}$ and $\mathcal{D}_{sk,c^\star}$. When queried with a ciphertext c, the first decription oracle answers $\mathsf{Dec}(sk, c)$. The only difference between $\mathcal{D}_{sk}$ and $\mathcal{D}_{sk,c^\star}$ is that the second oracle rejects the query $c^\star$, answering a special reject symbol $\perp$.

More formally, consider the following game:

$\mathsf{Game\ IND\text{-}CCA}()$
1 $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\ell)$
2 $b \leftarrow \{0, 1\}$
3 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{\mathcal{D}_{sk}}(pk)$
4 $c^\star \leftarrow \mathsf{Enc}(pk, m_b)$
5 $b' \leftarrow \mathcal{A}_2^{\mathcal{D}_{sk,c^\star}}(s, c^\star)$

Then, $\mathcal{E}$ is $\mathsf{IND\text{-}CCA}$ if and only if for all $\mathcal{A}^{\mathsf{IND-CCA}}$,

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}\right] = |2\mathsf{Pr}\left[b' = b\right] - 1| = |\mathsf{Pr}\left[b' = b\right] - \mathsf{Pr}\left[b' \neq b\right]| \in \mathsf{negl}(\ell)$$

The messages m_0 and m_1 generated by $\mathcal{A}_1$ must be in M_{pk}.

Notice that the decryption oracle formalizes the access to a decryption machine. Thus, the adversary is free to submit any polynomially bounded string (except for the target ciphertext, $c^\star$, in the guessing stage) to this oracle. This means that $\mathsf{IND\text{-}CCA}$ security depends not only on the encryption algorithm but also on the concrete implementation of the decryption algorithm, including its behaviour for inputs outside the set of valid ciphertexts (i.e. ciphertexts of the form $\mathsf{Enc}(pk, m, r)$ for $m \in M_{pk}$ and $r \in R_{pk}$). This behaviour can give very useful information for an adversary.

5.3 Random oracle model

There are some ways to define random functions or random oracles in the literature. In the seminal work [2], random oracles act as random functions from $\{0,1\}^\star$ to $\{0,1\}^\infty$, while in the influential paper [4] the random functions are collections of functions. Moreover, in the second definition for a given value of the complexity parameter, ℓ, the corresponding function goes from $\{0,1\}^{p_I(\ell)}$ to $\{0,1\}^{p_O(\ell)}$, where $p_I(\ell), p_O(\ell) \in \mathsf{poly}(\ell)$. Furthermore, the security of some schemes (e.g. [12]) depends on which definition of random function is used.

A random oracle can be viewed as a special type of random process or random sequence. The random oracle is defined through its idealised functionality, which is closely related to the random oracle simulations often used in the proofs of security.

Let A be a samplable polynomial size set. A *random function* G over A is a sequence of uniformly distributed independent random variables over A, indexed by the elements of $\{0,1\}^\star$. Notice that $\{0,1\}^\star$ is an ordered set. A *random oracle* over A is an oracle that answers queries exactly as if the random function G was evaluated.

The main property of a random function is that the joint distribution of q_G variables $G(s)$ for distinct values of s is the same regardless which values of s are selected. Thus, an efficient probabilistic (interactive) algorithm can simulate this random function by means of a table $\mathcal{T}_G$ storing all previous queries along with their answers. Any new (still unanswered) query will be answered with a "fresh" random value, which will be annotated in $\mathcal{T}_G$.

$G(s)$
1 if s in $\mathcal{T}_G$; return $\mathcal{T}_G(s)$; endif
2 $g \leftarrow A$
3 insert (s, g) in table $\mathcal{T}_G$
4 return g

Here, s in $\mathcal{T}_G$ will denote the fact that s has been queried to G by some party and $\mathcal{T}_G(s)$ will denote the answer given by G to that query.

Notice that the above algorithm runs in polynomial time and space if,

- A is a samplable polynomial size set
- during the game, in which the different parties have access to the random oracle, no more than a polynomial quantity, $q_G(\ell) \in \mathsf{poly}(\ell)$, of queries are made
- the size of each query is limited by a polynomial function in ℓ

This last condition will be obviously fulfilled if all parties are modeled by polynomial time machines.

Finally, IND-CCA security in the Random Oracle Model (ROM) of an asymmetric cryptosystem $\mathcal{E}$ is defined in the same way as above, but providing the adversary with oracle access to one or more random functions. In order to formalize the random coins of the random functions, a step $G \leftarrow \mathcal{R}(A)$ will be added at the beginning of the IND-CCA game for each random function used.

Since random functions have exponential size description, in real implementations they have to be adequately replaced by function families with polynomial size description, cryptographic hash functions being the most popular choice. This concrete specification will be included in the public data available to all parties in a protocol (e.g. the public key of an encryption scheme). The resulting gap between a random function and the function actually implemented in the protocol implies that security proofs are not

completely meaningful but a heuristic argument. In fact, several works (for instance [4,14]) have found theoretical weaknesses in any real implementation of protocols proven secure in the ROM. However, these flaws are so contrived that they do not affect protocols found in the literature. Research is currently being devoted to studying the meaning of the ROM heuristic in protocols closer to real cryptographic uses, for instance in [3].

6 Revisiting Fujisaki-Okamoto hybrid scheme

In this section, the transformation introduced in [11] of weak symmetric and asymmetric schemes into an IND-CCA hybrid encryption scheme is revisited.

6.1 The original construction

Let $\mathcal{E}^f = (\mathsf{KeyGen}^f, \mathsf{Enc}^f, \mathsf{Dec}^f)$ be a probabilistic asymmetric encryption scheme, defined from a TPOW function family f over the sets X, Y and Z, and $\mathcal{E}^{sym} = (\mathsf{KeyGen}^{sym}, \mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$ be a symmetric encryption scheme over the sets K and M. Let G be a random function over K, and H an independent random function over Y. The hybrid scheme $\mathcal{E}^{FO} = (\mathsf{KeyGen}^{FO}, \mathsf{Enc}^{FO}, \mathsf{Dec}^{FO})$, proposed by Fujisaki and Okamoto, works as follows.

Key generation. The public and secret keys are generated as in KeyGen^f.

Encryption. The ciphertext for a message $m \in M_\ell$ is

$$c = (f_{pk}(x, y), \mathsf{Enc}^{sym}_{G(x)}(m)),$$

where $y = H(x, m)$ and x is uniformly chosen in X_{pk}.

Decryption. To decrypt a ciphertext $c = (c_1, c_2)$, firstly compute $x = g_{sk}(c_1)$. Then, compute $m = \mathsf{Dec}^{sym}_{G(x)}(c_2)$ and return m if $c_1 = f_{pk}(x, H(x, m))$. Otherwise, return the reject symbol $\perp$. If it is not possible to compute $g_{sk}(c_1)$ or $\mathsf{Dec}^{sym}_{G(x)}(c_2)$, return $\perp$.

Let $\mathcal{A}^{\mathsf{IND-CCA}}[T, \epsilon, q_G, q_H, q_D]$ denote an adversary against the IND-CCA security of the above cryptosystem that runs in time T with advantage ϵ, doing no more than q_G, q_H and q_D queries respectively to the random oracles G, H and to the decryption oracles $\mathcal{D}_{sk}$ and $\mathcal{D}_{sk,c^\star}$. Then, the result claimed in [11] can be reformulated in the following way:

Theorem 11 *If there exists an adversary* $\mathcal{A}^{\mathsf{IND-CCA}}[T, \epsilon, q_G, q_H, q_D]$, *then there exists an adversary* $\mathcal{A}^{\mathsf{POW}}$ *against the POW of* f *in time* T_1 *with success probability* ϵ_1 *and an adversary* $\mathcal{A}^{\mathsf{IND-SYM}}$ *against the IND-SYM security of* $\mathcal{E}^{sym}$ *in time* T_2 *with advantage* ϵ_2 *such that*

$$\epsilon \leq (2(q_G + q_H)\epsilon_1 + \epsilon_2 + 1)\left(1 - 2\epsilon_1 - 2\epsilon_2 - \frac{1}{|Y|} - \frac{1}{|M|}\right)^{-q_D} - 1$$

and

$$T = \min(T_1, T_2) - O((q_G + q_H)\log(|X||M|))$$

The main drawback of this scheme is that the security reduction obtained in the proof is not tight, due to the quantity q_G+q_H multiplying ϵ_1. However, the same authors improved in [12] this result for the particular case of the Okamoto-Uchiyama scheme [20] (known as EPOC-2) and claimed, without proof, that a tight reduction is obtained for trivial easy verifiable primitives, using our terminology.

6.2 Identifying dangerous ambiguities

However, as pointed out in the introduction, several attacks against EPOC-2 have been found [15,9,24]. Despite the changes introduced in FO conversion after [15], there are still some ambiguities both in the scheme and in the security proof, which compromise the validity of the above theorem.

For instance, let us consider a TPOW function family f, and $X_{pk} \subset \overline{X}_{pk}$ such that $f_{pk}(x, y)$ is computable in polynomial time for any $x \in \overline{X}_{pk}$ and $y \in Y_{pk}$. Then, some badly generated ciphertexts

$$c = (f_{pk}(x, H(x, m)), \mathsf{Enc}^{sym}_{G(x)}(m))$$

for $x \in \overline{X}_{pk} \setminus X_{pk}$ may be accepted. This was the case for Okamoto-Uchiyama function in the original EPOC-2, where $\overline{X}_{pk} = \mathbb{Z}_{2^\ell+1}$ and $X_{pk} = \mathbb{Z}_{2^\ell}$, for $2^\ell < p < 2^\ell + 1$. This information was used in [15] to obtain the secret value p.

As Fujisaki and Okamoto proposed later in [12], this attack is avoided if all ciphertexts (c_1, c_2) such that $g_{sk}(c_1) \notin X_{pk}$ are rejected. However, when this change is included in the general conversion a problem of a different kind arises. If X is not a recognizable set, the checking cannot be performed in polynomial time. In this case the simulation of the $\mathcal{D}_{sk}$ in the proof is not correct.

Nevertheless, an additional oracle could be used to solve this problem. In this situation, an adversary can use the decryption oracle to solve a *difficult* decisional problem. As a result, we could only guarantee that breaking security of the cryptosystem is equivalent to solving a gap problem, that is, a stronger assumption than claimed.

This is the case for the Blum-Williams one-way trapdoor bijection family (i.e. squaring quadratic residues modulo $n = pq$), where $X_{pk} = Q_n$ and $\overline{X}_{pk} = Q_n \cup -Q_n$. Rejection of all ciphertexts (c_1, c_2) such that $g_{sk}(c_1) \notin X_{pk}$ means that the adversary will know if an arbitary $x \in \mathbb{Z}_n$ is a quadratic residue. Thus, the IND-CCA security of the hybrid cryptosystem will be based on the gap between the quadratic residuosity modulo n and factoring n assumptions.

6.3 The new proposal

From the above discussion, we know that although it is necessary to check if $g_{sk}(c_1) \in X_{pk}$ to prevent leaking vital information, this cannot be done in all cases.

In this section, we restrict the asymmetric primitives to those which admit a correct and unambiguous proof of security for the general transformation. We also take into account the results in [9,24], which use the ability to distinguish among rejection rules in the hybrid scheme to launch a total break. Thus, we slightly modify the specification of the decryption algorithm in the conversion. Finally, the recent developments in [19,6,7] can be applied to this transformation, and together with the concept of easy verifiable primitives, they are used to give a *new proof of security* improving the concrete security result presented in the original work.

Let $\mathcal{E}^f = (\mathsf{KeyGen}^f, \mathsf{Enc}^f, \mathsf{Dec}^f)$ be a probabilistic asymmetric encryption scheme obtained from a TPOW function family f over the sets X, Y and Z, and $\mathcal{E}^{sym} = (\mathsf{KeyGen}^{sym}, \mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$ be a symmetric encryption scheme over the sets K and M. Let G be a random function over K, and H an independent random function over Y.

The first change we introduce is that the random functions G and H are defined with unrestricted inputs, as explained in subsection 5.3. We believe it is not realistic to restrict the inputs of the random functions, as suggested in [11], since in a practical implementation random functions are replaced by cryptographic hash functions. Then, if a proof of security can be driven for unrestricted domains, this choice is preferable.

Now, X and M must be recognizable sets. Note that this is a restriction only for X, since almost always $M_\ell = \{0,1\}^{p(\ell)}$, for some polynomial p. In contrast, Z is not required to be a recognizable set. Instead of this, it is assumed that there exists a recognizable set $\bar{Z}$ such that $Z_{pk} \subseteq \overline{Z}_{pk}$, and that the partial inverse of f_{pk} can also be computed (in polynomial time) on elements of the extended set $\overline{Z}_{pk}$.

The proposed hybrid cryptosystem, $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, is almost the same as the original. The only, but nevertheless important, change is that now two different reject symbols are produced in the decryption algorithm Dec. Thus, when a ciphertext is rejected, the adversary will know the reason, obtaining different reject symbols without mounting a timing attack. Then, if the computing time of each step in the algorithm is independent of the data, the scheme seems to be robust against reject timing attacks.

$\mathsf{Dec}(sk, c)$
1 if $c \notin \overline{Z}_{pk} \times M_\ell$; return $\perp_1$; endif
2 $(c_1, c_2) = c$
3 $x \leftarrow g_{sk}(c_1)$
4 $m \leftarrow \mathsf{Dec}^{sym}_{G(x)}(c_2)$
5 $y \leftarrow H(x, m)$

6 if $x \notin X_{pk}$ or $f_{pk}(x,y) \neq c_1$; return $\perp_2$; endif
7 return m

We point out that in the OR operation in step 6 of the algorithm both predicates have *always* to be evaluated in order to prevent the adversary from detecting an extra rejection reason.

Now, the security results are stated. The first theorem is for the special case when f is an easy verifiable function family, while the second theorem works for general TPOW function families.

Theorem 12 *If there exists an adversary* $\mathcal{A}^{\mathsf{IND-CCA}}[T, \epsilon, q_G, q_H, q_D]$ *against the IND-CCA security of the proposed cryptosystem for an easy verifiable function family* f*, then there exists an adversary* $\mathcal{A}^{\mathsf{POW}}$ *that in time* T_1 *breaks the POW of* f *with success probability* ϵ_1 *and an adversary* $\mathcal{A}^{\mathsf{IND-SYM}}$ *that in time* T *breaks IND-SYM security of* $\mathcal{E}^{sym}$ *with advantage* ϵ_2 *such that*

$$\epsilon \leq \epsilon_1 + 3\epsilon_2 + \frac{3q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{3q_D}{|Y| - q_D}$$

and

$$T_1 \leq (q_G + q_H + q_D + q_G q_D)T[\mathcal{V}] + q_D\Big(T[f] + T[\mathsf{Dec}^{sym}]\Big) + T$$

where $T[\mathcal{V}]$ *is the time complexity of the plaintext checking algoritm for* f *and* $T[f]$ *is the time complexity of* f*.*

Proof: The proof is given in the appendix. □

Notice that now the probabilities are tightly related. In the general case, the plaintext checking algorithm could not exist . Using the access to a plaintext checking oracle instead, the following result is straightforward.

Corollary 13 *If there exists an adversary* $\mathcal{A}^{\mathsf{IND-CCA}}[T, \epsilon, q_G, q_H, q_D]$ *against the IND-CCA security of the proposed cryptosystem, then there exists an adversary* $\mathcal{A}^{\mathsf{POW-PCA}}$ *that in time* T_1 *breaks the POW-PCA of* f *with success probability* ϵ_1 *and an adversary* $\mathcal{A}^{\mathsf{IND-SYM}}$ *that in time* T *breaks IND-SYM security of* $\mathcal{E}^{sym}$ *with advantage* ϵ_2 *such that*

$$\epsilon \leq \epsilon_1 + 3\epsilon_2 + \frac{3q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{3q_D}{|Y| - q_D}$$

and

$$T_1 \leq (q_G + q_H + q_D + q_G q_D) + q_D\Big(T[f] + T[\mathsf{Dec}^{sym}]\Big) + T$$

where $T[f]$ *is the time complexity of* f*.*

Proof: It suffices to invoke the PC oracle into the plaintext checking algorithm $\mathcal{V}$ for f. Thus, by definition of oracle access, $T[\mathcal{V}] = 1$. □

6.4 Particular cases

Both in the case of the trivial construction of partial one-way function families and in the non-trivial family defined in subsection 4.1, the simulation in the security proof can be improved introducing only technical modifications.

In both cases, there exist a polynomial size set family $\tilde{Z}$ and two very efficiently computable function families $\tilde{f} : X \to \tilde{Z}$ and $\tilde{\pi} : \overline{Z} \to \tilde{Z}$ such that for all $pk \in PK$, $x \in X_{pk}$ and $z \in \overline{Z}_{pk}$, $\mathcal{V}(pk, x, z) = 1$ if and only if $\tilde{f}_{pk}(x) = \tilde{\pi}_{pk}(z)$. Notice that this property implies the injectivity of $\tilde{f}$. It is shown in the appendix that

$$T[\mathcal{A}^{\mathsf{POW}}] \leq (q_G + q_H)T[\tilde{f}] + q_D\Big(T[f] + T[\tilde{\pi}] + T[\mathsf{Dec}^{sym}]\Big) + T[\mathcal{A}^{\mathsf{IND-CCA}}]$$

then providing a *very-tight* security reduction.

If the trivial constructions are considered, $f_{pk}(x, y) = (\tilde{f}_{pk}(x), y)$ and $\tilde{\pi}_{pk}(\tilde{z}, y) = \tilde{z}$ so $T[\tilde{\pi}]$ can be neglected. Moreover, $T[f] \approx T[\tilde{f}]$ so

$$T[\mathcal{A}^{\mathsf{POW}}] \leq (q_G + q_H + q_D)T[\tilde{f}] + q_D T[\mathsf{Dec}^{sym}] + T[\mathcal{A}^{\mathsf{IND-CCA}}]$$

On the other hand, using the generalized RSA-Paillier function, $\tilde{f}_{n,r,e}(x) = x^e \bmod n$ and $\tilde{\pi}_{n,r,e}(z) = z \bmod n$. Note that $\overline{Z}_{n,r,e} = Z_{n,r,e} = \Omega_{n,r}$ and $\tilde{Z}_{n,r,e} = \mathbb{Z}_n^\star$. Then,

$$T[\mathcal{A}^{\mathsf{POW}}] \leq (q_G + q_H + q_D)O(\ell^2 \log e) + q_D T[\mathsf{Dec}^{sym}] + T[\mathcal{A}^{\mathsf{IND-CCA}}]$$

References

1. Bellare M, Desai A, Pointcheval D, Rogaway P (1998) Relations Among Notions of Security for Public-Key Encryption Schemes. In: Krawczyk H (ed) Advances in Cryptology - CRYPTO 1998. (Lecture Notes in Computer Science 1462) Springer, Berlin Heidelberg New York, pp 26-45
2. Bellare M, Rogaway P (1993) Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. ACM Press, New York, pp 62-73
3. Bellare M, Boldyreva A, Palacio A (2004) An Uninstantiable Random-Oracle-Model Scheme for a Hybrid-Encryption Problem. In: Cachin C, Camenisch J (eds) Advances in Cryptology - EUROCRYPT 2004. (Lecture Notes in Computer Science 3027) Springer, Berlin Heidelberg New York, pp 449-461
4. Canetti R, Goldreich O, Halevi S (1998) The random oracle methodology, revisited. In: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing. ACM Press, New York, pp 209-218
5. Catalano D, Gennaro R, Howgrave-Graham N, Nguyen PQ (2001) Paillier's Cryptosystem Revisited. In: Proceedings of the 8th ACM Conference on Computer and Communications Security. ACM Press, New York, pp 206-214
6. Coron J, Handschuh H, Joye M, Paillier P, Pointcheval D, Tymen C (2002) GEM: a Generic Chosen-Ciphertext Secure Encryption Method. In: Preneel B (ed) Topics in Cryptology - CT-RSA 2002. (Lecture Notes in Computer Science 2271) Springer, Berlin Heidelberg New York, pp 263-276

7. Coron J, Handschuh H, Joye M, Paillier P, Pointcheval D, Tymen C (2002) Optimal Chosen-Ciphertext Secure Encryption of Arbitrary-Length Messages. In: Naccache D, Paillier P (eds) Public Key Cryptography, PKC 2002 (Lecture Notes in Computer Science 2274) Springer, Berlin Heidelberg New York, pp 17-33
8. Cramer R, Shoup V (2002) Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen L R (ed) Advances in Cryptology - EUROCRYPT 2002. (Lecture Notes in Computer Science 2332) Springer, Berlin Heidelberg New York, pp 45-64
9. Dent A W (2002) An implementation attack against the EPOC-2 public-key cryptosystem. Electronics Letters 38(9):412-413
10. EPOC, Efficient Probabilistic Public-Key Encryption. `http://info.isl.ntt.co.jp/epoc/`
11. Fujisaki E, Okamoto T (1999) Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener M J (ed) Advances in Cryptology - CRYPTO 1999. (Lecture Notes in Computer Science 1666) Springer, Berlin Heidelberg New York, pp 537-554
12. Fujisaki E, Okamoto T (2001) A Chosen-Cipher Secure Encryption Scheme Tightly as Secure as Factoring. IEICE Transactions on Fundamentals E84-A(1): 179-187
13. Golwasser S, Micali S (1984) Probabilistic encryption. J Computer and System Sciences 28:270-299
14. Goldwasser S, Tauman Y (2003) On the (In)security of the Fiat-Shamir Paradigm. In: Proceedings of 44th Symposium on Foundations of Computer Science (FOCS 2003). IEEE Computer Society, pp 102-115
15. Joye M, Quisquater J J, Yung M (2001) On the Power of Misbehaving Adversaries and Security Analysis of the Original EPOC. In: Naccache D (ed) Topics in Cryptology - CT-RSA 2001. (Lecture Notes in Computer Science 2020) Springer, Berlin Heidelberg New York, pp 208-222
16. Menezes A (1993) Elliptic Curve Public-Key Cryptosystems. The Kluwer International Series in Engineering and Computer Science; SECS 234. Kluwer Academic Publishers, Boston Dordrecht London.
17. Naor M, Yung M (1990) Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing. ACM Press, New York, pp 427-437
18. Okamoto T, Pointcheval D (2001) The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes. In: Kim K (ed) Public Key Cryptography, PKC 2001. (Lecture Notes in Computer Science 1992) Springer, Berlin Heidelberg New York, pp 104-118
19. Okamoto T, Pointcheval D (2001) REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In: Naccache D (ed) Topics in Cryptology - CT-RSA 2001. (Lecture Notes in Computer Science 2020) Springer, Berlin Heidelberg New York, pp 159-175
20. Okamoto T, Uchiyama S (1998) A New Public-Key Cryptosystem as Secure as Factoring. In: Nyberg K (ed) Advances in Cryptology - EUROCRYPT 1998 (Lecture Notes in Computer Science 1403) Springer, Berlin Heidelberg New York, pp 308-318
21. PSEC, Provably Secure Encryption Scheme. `http://info.isl.ntt.co.jp/psec/`
22. Pointcheval D (2000) Chosen-Ciphertext Security for any One-Way Cryptosystem. In: Imai H, Zheng Y (eds): Public Key Cryptography, PKC 2000.

(Lecture Notes in Computer Science 1751) Springer, Berlin Heidelberg New York, pp 129-146
23. Rackoff C, Simon D R (1992) Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum J (ed) Advances in Cryptology - CRYPTO 1991 (Lecture Notes in Computer Science 576) Springer, Berlin Heidelberg New York, pp 433-444
24. Sakurai K, Takagi T (2002) A Reject Timing Attack on an IND-CCA2 Public-Key Cryptosystem. In: Joong Lee P, Hoon Lim C (eds) Information Security and Cryptology - ICISC 2002. (Lecture Notes in Computer Science 2587) Springer, Berlin Heidelberg New York, pp 359–373
25. Shoup V (2001) OAEP Reconsidered. In: Kilian J (ed) Advances in Cryptology - CRYPTO 2001. (Lecture Notes in Computer Science 2139) Springer, Berlin Heidelberg New York, pp 239-259
26. Stern J (2003) Why Provable Security Matters? In: Biham E (ed) Advances in Cryptology - EUROCRYPT 2003. (Lecture Notes in Computer Science 2656) Springer, Berlin Heidelberg New York, pp 449-461
27. Watanabe Y, Shikata J, Imai H (2002) Equivalence between Semantic Security and Indistinguishability against Chosen Ciphertext Attacks. In: Biham E (ed) Advances in Cryptology - EUROCRYPT 2003. (Lecture Notes in Computer Science 2567) Springer, Berlin Heidelberg New York, pp 71-84

A Proof of theorem 12

Let $\mathcal{A}^{\mathsf{IND-CCA}}[T, \epsilon, q_G, q_H, q_D] = (\mathcal{A}_1, \mathcal{A}_2)$ be the adversary aiming to attack the IND-CCA security of the hybrid encryption scheme, $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ described in subsection 6.3.

In order to prove the theorem, some different games will be considered. Starting from the IND-CCA game, we will introduce several intermediate games before designing the game for an adversary who tries to break the partial one-wayness (POW) of f. Each game will be obtained by introducing slight modifications into the previous game in such a way that the adversary success probabilities are easily related.

Although in all games the adversary uses the same coins, this might not be the case for the coins used in the games along the simulation. Thus, different games might lie in different probability spaces, and the events defined from the view of $\mathcal{A}^{\mathsf{IND-CCA}}$ might occur with different probabilities. Let us denote by $\mathsf{Pr}_i[F]$ the probability of event F in game i.

Each game will be described as a main algorithm along with some auxiliar algorithms used as oracles by $\mathcal{A}^{\mathsf{IND-CCA}}$. The bulleted steps in the algorithms will indicate the main changes introduced in the game, with respect to the previous one.

The following trivial lemma will be very useful in this proof, since it allows us to relate the probabilities of some event across different games.

Lemma 14 *Let E_1, F_1 be two events defined in a probability space $\mathcal{X}_1$, and E_2, F_2 another two events defined in a probability space $\mathcal{X}_2$, such that $p = \mathsf{Pr}_{\mathcal{X}_2}[F_2] = \mathsf{Pr}_{\mathcal{X}_1}[F_1]$ and $\mathsf{Pr}_{\mathcal{X}_2}[E_2 \wedge \neg F_2] = \mathsf{Pr}_{\mathcal{X}_1}[E_1 \wedge \neg F_1]$. Then*

$$|\mathsf{Pr}_{\mathcal{X}_2}[E_2] - \mathsf{Pr}_{\mathcal{X}_1}[E_1]| \leq p$$

This lemma is a generalization of the lemma used by Shoup in [25] since different probability spaces are considered.

Game0. The IND-CCA attack. There are some minor differences between Game0 and the standard IND-CCA game, described in subsection 5.2, but they do not modify any probability.

Game0()
1 $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\ell)$; $G \leftarrow \mathcal{R}(K_\ell)$; $H \leftarrow \mathcal{R}(Y_{pk})$
2 $b \leftarrow \{0,1\}$; $x^\star \leftarrow X_{pk}$
3 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G,H,\mathcal{D}_{sk}}(pk)$
4 $y^\star \leftarrow H(x^\star, m_b)$; $c^\star \leftarrow \left(f_{pk}(x^\star, y^\star), \mathsf{Enc}^{sym}_{G(x^\star)}(m_b)\right)$
5 $b' \leftarrow \mathcal{A}_2^{G,H,\mathcal{D}_{sk,c^\star}}(s, c^\star)$

where the oracle's answer $\mathcal{D}_{sk}(c)$ is exactly the same as the value returned by the $\mathsf{Dec}(sk, c)$ (see subsection 6.3 for details).

Let Askx be the event that, during the game, either $x^\star \in X$ is queried (by $\mathcal{A}^{\mathsf{IND-CCA}}$) to G or $(x^\star, m)$ is queried to H, for some m. Then,

$$\begin{aligned}\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}\right] &= |\mathsf{Pr}_0[b' = b] - \mathsf{Pr}_0[b' \neq b]| \leq \\ &\leq |\mathsf{Pr}_0[b' = b \wedge \mathsf{Askx}] - \mathsf{Pr}_0[b' \neq b \wedge \mathsf{Askx}]| + \\ &\quad + |\mathsf{Pr}_0[b' = b \wedge \neg\mathsf{Askx}] - \mathsf{Pr}_0[b' \neq b \wedge \neg\mathsf{Askx}]| \leq \\ &\leq \mathsf{Pr}_0[\mathsf{Askx}] + |\mathsf{Pr}_0[b' = b \wedge \neg\mathsf{Askx}] - \mathsf{Pr}_0[b' \neq b \wedge \neg\mathsf{Askx}]|\end{aligned}$$

In order to improve the readability of the rest of the proof, let us define $\mathsf{S}_1 = \mathsf{Askx}$, $\mathsf{S}_{01} = \neg\mathsf{Askx} \wedge b' = b$ and $\mathsf{S}_{00} = \neg\mathsf{Askx} \wedge b' \neq b$. The above equation can be rewritten as

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}\right] \leq \mathsf{Pr}_0[\mathsf{S}_1] + |\mathsf{Pr}_0[\mathsf{S}_{01}] - \mathsf{Pr}_0[\mathsf{S}_{00}]|$$

Game1. In this game, the queries made by $\mathcal{A}^{\mathsf{IND-CCA}}$ to the random oracles are intercepted in order to immediately abort the execution of the game if Askx (i.e. S_1) occurs. The following functions will perform this task:

$G1(x)$
1 if $x = x^\star$; $b' \leftarrow \{0,1\}$; exit game; endif
2 return $G(x)$

$H1(x, m)$
1 if $x = x^\star$; $b' \leftarrow \{0,1\}$; exit game; endif
2 return $H(x, m)$

and the new game is the same except for replacing the oracles given to $\mathcal{A}^{\mathsf{IND-CCA}}$ by the above functions.

Game1()

1 $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\ell)$; $G \leftarrow \mathcal{R}(K_\ell)$; $H \leftarrow \mathcal{R}(Y_{pk})$
2 $b \leftarrow \{0,1\}$; $x^\star \leftarrow X_{pk}$
• 3 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G1, H1, \mathcal{D}_{sk}}(pk)$
4 $y^\star \leftarrow H(x^\star, m_b)$; $c^\star \leftarrow \left(f_{pk}(x^\star, y^\star), \mathsf{Enc}^{sym}_{G(x^\star)}(m_b)\right)$
• 5 $b' \leftarrow \mathcal{A}_2^{G1, H1, \mathcal{D}_{sk, c^\star}}(s, c^\star)$

Since the games are identical when $\neg \mathsf{S}_1$, the events S_1, S_{01} and S_{00} remain unchanged in Game1. Then,

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}\right] \leq \mathsf{Pr}_1[\mathsf{S}_1] + |\mathsf{Pr}_1[\mathsf{S}_{01}] - \mathsf{Pr}_1[\mathsf{S}_{00}]|$$

Game2. In this game, the decryption oracle is modified in such a way that it is disallowed from making new queries to the random oracle G. Let $\mathcal{Q}_G$ be the set of all values queried by $\mathcal{A}^{\mathsf{IND-CCA}}$ to oracle $G1$ to the execution point. Now in this game, all ciphertexts (c_1, c_2) submitted to the decryption oracle such that $g_{sk}(c_1) \notin X_{pk} \cap \mathcal{Q}_G$ are rejected by returning $\perp_2$, even when some of them may be valid ciphertexts.

$\mathcal{D}2_{sk}(c)$
1 if $c \notin \overline{Z}_{pk} \times M_\ell$; return $\perp_1$; endif
2 $(c_1, c_2) = c$
3 $x \leftarrow g_{sk}(c_1)$
• 4 if $x \notin X_{pk}$ or $x \notin \mathcal{Q}_G$; return $\perp_2$; endif
5 $m \leftarrow \mathsf{Dec}^{sym}_{G(x)}(c_2)$
6 $y \leftarrow H(x, m)$
7 if $f_{pk}(x, y) \neq c_1$; return $\perp_2$; endif
8 return m

and decryption oracle $\mathcal{D}_{sk,c^\star}$ is modified in the same way. In the main game algorithm, $\mathcal{A}^{\mathsf{IND-CCA}}$ is provided with oracles $\mathcal{D}2_{sk}$ and $\mathcal{D}2_{sk,c^\star}$ instead of $\mathcal{D}_{sk}$ and $\mathcal{D}_{sk,c^\star}$.

Let F be the event that, in some query to the decryption oracle, the ciphertext is accepted in Game1, but rejected at step 4 of $\mathcal{D}2_{sk}$. Before F occurs, both games are indentical. Then, by lemma 14,

$$|\mathsf{Pr}_2[\mathsf{S}_1] - \mathsf{Pr}_1[\mathsf{S}_1]| \leq \mathsf{Pr}[\mathsf{F}]$$
$$|\mathsf{Pr}_2[\mathsf{S}_{01}] - \mathsf{Pr}_1[\mathsf{S}_{01}]| \leq \mathsf{Pr}[\mathsf{F}]$$
$$|\mathsf{Pr}_2[\mathsf{S}_{00}] - \mathsf{Pr}_1[\mathsf{S}_{00}]| \leq \mathsf{Pr}[\mathsf{F}]$$

From these inequalities, it can be easily shown that

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}\right] \leq \mathsf{Pr}_2[\mathsf{S}_1] + |\mathsf{Pr}_2[\mathsf{S}_{01}] - \mathsf{Pr}_2[\mathsf{S}_{00}]| + 3\mathsf{Pr}[\mathsf{F}]$$

The following lemma gives an upper bound for $\mathsf{Pr}[\mathsf{F}]$.

Lemma 15

$$\mathsf{Pr}[\mathsf{F}] \leq \frac{q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{q_D}{|Y| - q_D}$$

Proof: Let F_k be the event that F occurs exactly at the k-th query to the decryption oracle. Clearly, $\mathsf{Pr}[\mathsf{F}] = \sum_{k=1}^{q_D} \mathsf{Pr}[\mathsf{F}_k]$. Note that Askx cannot occur before the q_D-th query has been made. In order to compute an upper bound of $\mathsf{Pr}[\mathsf{F}_k]$, it will be better to consider the conditional probability $p_k = \mathsf{Pr}\left[\mathsf{F}_k \mid \bigvee_{i=1}^{k-1} \neg \mathsf{F}_i\right]$, which means the probability that F occurs exactly at the k-th query supposing that games 1 and 2 run identically until this query. Thus, $\mathsf{Pr}[\mathsf{F}_k] \leq p_k$ and

$$\mathsf{Pr}[\mathsf{F}] \leq \sum_{k=1}^{q_D} p_k$$

Let us compute an upper bound for p_k. Now, games 1 and 2 run identically just until the k-th query, which will be denoted by $\bar{c}$.

Suppose for a while that $\mathcal{A}^{\mathsf{IND-CCA}}$ is in the 'finding' stage. The only information available to the adversary in order to generate the cyphertext $\bar{c}$ is the view of the game at this execution point, that is $\mathsf{View} = (pk, \mathcal{T}_G, \mathcal{T}_H, \mathcal{T}_{\mathcal{D}})$, where $\mathcal{T}_{\mathcal{O}}$ denotes the sequence of all queries made by $\mathcal{A}^{\mathsf{IND-CCA}}$ to the oracle $\mathcal{O}$ (that is G, H or the decryption oracles), along with the corresponding answers. To find an upper bound for p_k, we will consider the best choice for $\bar{c}$, for each possible View.

The event F_k occurs if and only if $\mathcal{D}2_{sk}(\bar{c}) \neq \mathcal{D}_{sk}(\bar{c})$; that is, $\mathcal{D}2_{sk}$ rejects $\bar{c}$ (returning $\perp_2$) while $\mathcal{D}_{sk}$ accepts it. This means that $\bar{c} = (f_{pk}(\bar{x}, \bar{y})), \bar{c}_2)$, where $\bar{x} \in X_{pk} \backslash \mathcal{Q}_G$, $\bar{y} \in Y_{pk}$, $\bar{c}_2 \in M_\ell$, and the equation $\bar{y} = H(\bar{x}, \mathsf{Dec}^{sym}_{G(\bar{x})}(\bar{c}_2))$ holds.

If View and $\bar{c}$ are fixed, then p_k depends only on the joint probability distribution of $G(\bar{x})$ and $H(\bar{x}, \mathsf{Dec}^{sym}_{G(\bar{x})}(\bar{c}_2))$. But this distribution is conditioned by the answers given by H to the queries $(\bar{x}, m)$ for some m, and the answers given by $\mathcal{D}_{sk}$ to the queries $(f_{pk}(\bar{x}, y), c_2)$ for some $y \in Y_{pk}$ and $c_2 \in M_\ell$. Notice that any queried ciphertext $c \notin Z_{pk} \times M_\ell$ is rejected by $\mathcal{D}_{sk}$, independently of the values taken by the random functions.

In the worst case, all queries in $\mathcal{T}_H$ and $\mathcal{T}_{\mathcal{D}}$ are related to $\bar{x}$; that is, $h_i = H(\bar{x}, m_i)$ for $i = 1, \ldots, q_H$, and $c^{(j)} = (f_{pk}(\bar{x}, y_j), c_2^{(j)})$ for $j = 1, \ldots, k-1$. Since $\bar{x} \notin \mathcal{Q}_G$, then $\mathcal{D}_{sk}(c^{(j)}) = \mathcal{D}2_{sk}(c^{(j)}) = \perp_2$, and then $y_j \neq H(\bar{x}, \mathsf{Dec}^{sym}_{G(\bar{x})}(c_2^{(j)}))$. These equations could be incompatible for some values of $G(\bar{x})$, namely those $g \in K_\ell$ such that $m_i = \mathsf{Dec}^{sym}_g(c_2^{(j)})$ and $h_i = y_j$ for some (i, j). In the (unfeasible) worst case, all h_i and y_j are equal and there can be up to $q_H(k-1)\gamma$ forbidden values for $G(\bar{x})$. Then, the random variable $G(\bar{x})$ is uniformly distributed over a set of at least $|K_\ell| - (k-1)q_H\gamma$ elements.

There are at most $q_H\gamma$ different values of g such that $(\bar{x}, \mathsf{Dec}^{sym}_g(\bar{c}_2)) \in \mathcal{Q}_H$, where $\mathcal{Q}_H$ is the set of pairs (x, m) queried to H by $\mathcal{A}^{\mathsf{IND-CCA}}$. For these values, $\bar{y} = H(\bar{x}, \mathsf{Dec}^{sym}_g(\bar{c}_2))$ can be ensured if all h_i are equal to $\bar{y}$. Thus,

$$\mathsf{Pr}\left[\mathsf{F}_k \wedge (\bar{x}, \mathsf{Dec}^{sym}_{G(\bar{x})}(\bar{c}_2)) \in \mathcal{Q}_H \mid \mathsf{View}\right] \leq \frac{q_H\gamma}{|K_\ell| - (k-1)q_H\gamma}$$

For any g such that $(\bar{x}, \mathsf{Dec}_g^{sym}(\bar{c}_2)) \notin \mathcal{Q}_H$, the variable $H(\bar{x}, \mathsf{Dec}_g^{sym}(\bar{c}_2))$ is uniformly distributed over a set of at least $|Y_{pk}|-(k-1)$ elements, because if $\bar{c}_2 = c_2^{(j)}$, then the value y_j is forbidden. Consequently,

$$\mathsf{Pr}\left[\mathsf{F}_k \wedge (\bar{x}, \mathsf{Dec}_{G(x)}^{sym}(c_2)) \notin \mathcal{Q}_H \;\middle|\; \mathsf{View}\right] \leq \frac{1}{|Y_{pk}| - (k-1)}$$

and summing up, we obtain

$$p_k^{\text{find}} \leq \frac{q_H \gamma}{|K_\ell| - (k-1) q_H \gamma} + \frac{1}{|Y_{pk}| - (k-1)}$$

If $\mathcal{A}^{\mathsf{IND-CCA}}$ is in the 'guessing' stage, then $c^\star$ holds valuable information. In fact, $\mathsf{View} = (pk, \mathcal{T}_G, \mathcal{T}_H, \mathcal{T}_\mathcal{D}, c^\star)$, but $c^\star$ depends only on $G(x^\star)$ and $H(x^\star, m_b)$. Thus, if $\bar{x} \neq x^\star$, $c^\star$ does not give any additional information about F_k, and everything goes the same way as in the 'finding' stage.

If $\bar{x} = x^\star$, the restriction $\bar{c} \neq c^\star$ must also be considered. Moreover, there are no queries in $\mathcal{Q}_H$ related to $x^\star$. Then, in the worst case, the joint distribution of $G(\bar{x})$ and $H(\bar{x}, \mathsf{Dec}_{G(\bar{x})}^{sym}(\bar{c}_2))$ is conditioned by the equations $y_j \neq H(x^\star, \mathsf{Dec}_{G(x^\star)}^{sym}(c_2^{(j)}))$, for $j = 1 \ldots, k-1$, $y^\star = H(x^\star, m_b)$ and $m_b = \mathsf{Dec}_{G(x^\star)}^{sym}(c_2^\star)$.

The equality $y^\star = H(x^\star, m_b)$ is useless, since the only valid ciphertext related to $H(x^\star, m_b)$ is $c^\star$. Nevertheless, from $m_b = \mathsf{Dec}_{G(x^\star)}^{sym}(c_2^\star)$, only a reduced number of values of $G(x^\star)$ remain possible, but, as above, $H(x^\star, \mathsf{Dec}_{G(x^\star)}^{sym}(\bar{c}_2))$ is uniformly distributed over a set of at least $|Y_{pk}| - (k-1)$ elements, and $p_k^{\text{guess}} \leq \dfrac{1}{|Y_{pk}| - (k-1)}$.

Finally,

$$\mathsf{Pr}[\mathsf{F}] \leq \sum_{k=1}^{q_D} \left(\frac{q_H \gamma}{|K_\ell| - (k-1) q_H \gamma} + \frac{1}{|Y_{pk}| - (k-1)} \right) \leq \frac{q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{q_D}{|Y| - q_D}$$

□

Game2′. In this game, oracles G and H are simulated by using tables $\mathcal{T}_G$ and $\mathcal{T}_H$, as described in subsection 5.3. The generation of the ciphertext, which is also different, is equivalent to redefining some values of the random functions used in Game2. Namely, $G(x^\star) = g^\star$ and $H(x^\star, m_b) = y^\star$. But these changes in the oracles do not affect the probability distribution of the view of $\mathcal{A}^{\mathsf{IND-CCA}}$, since in Game2 $\mathcal{A}^{\mathsf{IND-CCA}}$ neither queried $x^\star$ to G nor $(x^\star, m)$ to H, for any m. (Note that, at step 6 of $\mathcal{D}2_{sk}$, $x \neq x^\star$ since $x^\star \notin \mathcal{T}_{G1}$.)

$\mathsf{Game2'}()$

- 1 $\mathcal{T}_G \leftarrow \mathsf{empty}$; $\mathcal{T}_H \leftarrow \mathsf{empty}$
  2 $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\ell)$

3 $b \leftarrow \{0,1\};\ x^\star \leftarrow X_{pk}$
4 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G2',H2',\mathcal{D}2'_{sk}}(pk)$
• 5 $g^\star \leftarrow K_\ell;\ y^\star \leftarrow Y_{pk};\ c^\star \leftarrow \left(f_{pk}(x^\star, y^\star), \mathsf{Enc}_{g^\star}^{sym}(m_b)\right)$
6 $b' \leftarrow \mathcal{A}_2^{G2',H2',\mathcal{D}2'_{sk,c^\star}}(s, c^\star)$

$\mathcal{D}2'_{sk}(c)$
1 if $c \notin \overline{Z}_{pk} \times M_\ell$; return $\perp_1$; endif
2 $(c_1, c_2) = c$
3 $x \leftarrow g_{sk}(c_1)$
• 4 if $x \notin X_{pk}$ or x not in $\mathcal{T}_G$; return $\perp_2$; endif
• 5 $g \leftarrow \mathcal{T}_G(x)$
6 $m \leftarrow \mathsf{Dec}_g^{sym}(c_2)$
7 $y \leftarrow H2'(x, m)$
8 if $f_{pk}(x, y) \neq c_1$; return $\perp_2$; endif
9 return m

$G2'(x)$
• 1 if x in $\mathcal{T}_G$; return $\mathcal{T}_G(x)$; endif
2 if $x = x^\star$; exit game; endif
• 3 $r \leftarrow K_\ell$
• 4 insert (x, r) in table $\mathcal{T}_G$
5 return r

$H2'(x, m)$
• 1 if (x, m) in $\mathcal{T}_H$; return $\mathcal{T}_H(x, m)$; endif
2 if $x = x^\star$; exit game; endif
• 3 $r \leftarrow Y_{pk}$
• 4 insert $((x, m), r)$ in table $\mathcal{T}_H$
5 return r

Game3. In this game, we introduce some modifications to avoid the use of m_b in the generation of the target ciphertext. In fact, the differences between using m_b and using a random message can be tapped by a new adversary $\mathcal{A}^{\mathsf{IND-SYM}} = (\mathcal{A}_1^{sym}, \mathcal{A}_2^{sym})$ who tries to break the IND-SYM security of $\mathcal{E}^{sym}$ (see 5.1).

Game3()
1 $\beta \leftarrow \{0,1\}$
2 $(\mu_0, \mu_1, \sigma) \leftarrow \mathcal{A}_1^{sym}(1^\ell)$
3 $g^\star \leftarrow K_\ell;\ \kappa^\star = \mathsf{Enc}_{g^\star}^{sym}(\mu_\beta)$
4 $\beta' \leftarrow \mathcal{A}_2^{sym}(\sigma, \kappa^\star)$

$\mathcal{A}_1^{sym}(1^\ell)$
1 $\mathcal{T}_G \leftarrow$ empty; $\mathcal{T}_H \leftarrow$ empty

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\ell)$
$b \leftarrow \{0,1\};\ x^\star \leftarrow X_{pk}$
$(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G3,H3,\mathcal{D}3_{sk}}(pk)$
$m \leftarrow M_\ell$
$\sigma = (\mathcal{T}_G, \mathcal{T}_H, pk, sk, b, x^\star, s)$
return (m_b, m, σ)

$\mathcal{A}_2^{sym}(\sigma, \kappa^\star)$
$(\mathcal{T}_G, \mathcal{T}_H, pk, sk, b, x^\star, s) = \sigma$
$y^\star \leftarrow Y_{pk};\ c^\star \leftarrow (f_{pk}(x^\star, y^\star), \kappa^\star)$
$b' \leftarrow \mathcal{A}_2^{G3,H3,\mathcal{D}3_{sk,c^\star}}(s, c^\star)$
if $b' = b$
  $\beta' \leftarrow 0$
else
  $\beta' \leftarrow 1$
endif
$\beta'' \leftarrow 0$

$G3(x)$
if x in $\mathcal{T}_G$; return $\mathcal{T}_G(x)$; endif
if $x = x^\star$
• $\beta' \leftarrow \{0,1\}$
• $\beta'' \leftarrow 1$
  exit game
endif
$r \leftarrow K_\ell$
insert (x, r) in table $\mathcal{T}_G$
return r

$H3(x, m)$
if (x, m) in $\mathcal{T}_H$; return $\mathcal{T}_H(x, m)$; endif
if $x = x^\star$
• $\beta' \leftarrow \{0,1\}$
• $\beta'' \leftarrow 1$
  exit game
endif
$r \leftarrow Y_{pk}$
insert $((x, m), r)$ in table $\mathcal{T}_H$
return r

The only difference in the decryption oracles is that $H2'$ is replaced by $H3$.

In this game, $\mathcal{A}^{\mathsf{IND-SYM}}$ has two different ways to guess the value of β: β' indicates if $\mathcal{A}^{\mathsf{IND-CCA}}$ guesses the correct value of b and β'' indicates if S_1 occurs. Then, two different advantages can be taken into account:

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right] = |2\mathsf{Pr}_3\left[\beta' = \beta\right] - 1| \quad \text{and}$$

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right]' = \left|2\mathsf{Pr}_3\left[\beta'' = \beta\right] - 1\right|.$$

If $\beta = 1$, the value of m_b is used nowhere in the game. So, the view of $\mathcal{A}^{\mathsf{IND-CCA}}$ is independent of b and

$$\mathsf{Pr}_3\left[\beta' = 1 \mid \beta = 1 \wedge \neg \mathsf{S}_1\right] = \mathsf{Pr}_3\left[b' \neq b \mid \beta = 1 \wedge \neg \mathsf{S}_1\right] = \frac{1}{2}.$$

Moreover, $\mathsf{Pr}_3\left[\beta' = 1 \mid \beta = 1 \wedge \mathsf{S}_1\right] = \frac{1}{2}$ and then $\mathsf{Pr}_3\left[\beta' = 1 \mid \beta = 1\right] = \frac{1}{2}$.

If $\beta = 0$, Game3 and Game2$'$ are identical. Thus

$$\begin{array}{rl} \mathsf{Pr}_3\left[\beta' = 0 \mid \beta = 0\right] & = \mathsf{Pr}_3\left[\beta' = 0 \wedge \mathsf{S}_1 \mid \beta = 0\right] + \mathsf{Pr}_3\left[\beta' = 0 \wedge \neg \mathsf{S}_1 \mid \beta = 0\right] = \\ & = \frac{1}{2}\mathsf{Pr}_3\left[\mathsf{S}_1 \mid \beta = 0\right] + \mathsf{Pr}_3\left[b' = b \wedge \neg \mathsf{S}_1 \mid \beta = 0\right] = \\ & = \frac{1}{2}\mathsf{Pr}_2\left[\mathsf{S}_1\right] + \mathsf{Pr}_2\left[\mathsf{S}_{01}\right] \end{array}$$

Putting altogether,

$$\begin{array}{rl} \mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right] & = \left|2\mathsf{Pr}_3\left[\beta' = 0 \wedge \beta = 0\right] + 2\mathsf{Pr}_3\left[\beta' = 1 \wedge \beta = 1\right] - 1\right| = \\ & = \left|\mathsf{Pr}_3\left[\beta' = 0 \mid \beta = 0\right] + \mathsf{Pr}_3\left[\beta' = 1 \mid \beta = 1\right] - 1\right| = \\ & = \left|\mathsf{Pr}_2\left[\mathsf{S}_{01}\right] + \frac{1}{2}\mathsf{Pr}_2\left[\mathsf{S}_1\right] - \frac{1}{2}\right| = \frac{1}{2}\left|\mathsf{Pr}_2\left[\mathsf{S}_{01}\right] - \mathsf{Pr}_2\left[\mathsf{S}_{00}\right]\right| \end{array}$$

If β'' is used instead of β', then

$$\begin{array}{rl} \mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right]' & = \left|2\mathsf{Pr}_3\left[\mathsf{S}_1 \wedge \beta'' = \beta\right] + 2\mathsf{Pr}_3\left[\neg \mathsf{S}_1 \wedge \beta'' = \beta\right] - 1\right| = \\ & = \left|2\mathsf{Pr}_3\left[\mathsf{S}_1 \wedge \beta = 1\right] + 2\mathsf{Pr}_3\left[\neg \mathsf{S}_1 \wedge \beta = 0\right] - 1\right| = \\ & = \left|\mathsf{Pr}_3\left[\mathsf{S}_1 \mid \beta = 1\right] + \mathsf{Pr}_3\left[\neg \mathsf{S}_1 \mid \beta = 0\right] - 1\right| = \\ & = \left|\mathsf{Pr}_3\left[\mathsf{S}_1 \mid \beta = 1\right] - \mathsf{Pr}_3\left[\mathsf{S}_1 \mid \beta = 0\right]\right| = \\ & = \left|\mathsf{Pr}_3\left[\mathsf{S}_1 \mid \beta = 1\right] - \mathsf{Pr}_2\left[\mathsf{S}_1\right]\right| \end{array}$$

Finally,

$$\begin{array}{rl} \mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}\right] & \leq \mathsf{Pr}_2\left[\mathsf{S}_1\right] + \left|\mathsf{Pr}_2\left[\mathsf{S}_{01}\right] - \mathsf{Pr}_2\left[\mathsf{S}_{00}\right]\right| + 2\mathsf{Pr}\left[\mathsf{F}\right] = \\ & = \mathsf{Pr}_2\left[\mathsf{S}_1\right] + 2\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right] + 2\mathsf{Pr}\left[\mathsf{F}\right] \leq \\ & \leq \mathsf{Pr}_3\left[\mathsf{S}_1 \mid \beta = 1\right] + 2\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right] + \\ & \quad + \mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right]' + 3\mathsf{Pr}\left[\mathsf{F}\right] \end{array}$$

Game4. Game3 with $\beta = 1$ can be modified to obtain an implementation of an adversary, $\mathcal{A}^{\mathsf{POW}}$, trying to break the partial one-wayness of f. This adversary will know neither sk nor $x^\star$. The use of sk in the decryption oracle simulator is avoided by using the deterministic plaintext checking algorithm $\mathcal{V}$ for f. The use of $x^\star$ in the random oracle simulators is also avoided. To do this, S_1 is detected by using $\mathcal{V}$. In fact, when S_1 occurs $\mathcal{A}^{\mathsf{POW}}$ learns the value of $x^\star$ and stores it in x'.

$\mathsf{Game4}()$
1 $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\ell)$
2 $x^\star \leftarrow X_{pk};\ y^\star \leftarrow Y_{pk};\ z \leftarrow f_{pk}(x^\star, y^\star)$
3 $\mathcal{A}^{\mathsf{POW}}(pk, z)$

$\mathcal{A}^{\mathsf{POW}}(pk, z)$

$b \leftarrow \{0,1\}$
$m \leftarrow M_\ell$; $g^\star \leftarrow K_\ell$; $c^\star \leftarrow (z, \mathsf{Enc}_{g^\star}^{sym}(m))$
$\mathcal{T}_G \leftarrow$ empty ; $\mathcal{T}_H \leftarrow$ empty
$(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G4,H4,\mathcal{D}4_{pk}}(pk)$
$b' \leftarrow \mathcal{A}_2^{G4,H4,\mathcal{D}4_{pk,c^\star}}(s, c^\star)$
• $x' \leftarrow X_{pk}$

$G4(x)$

if x in $\mathcal{T}_G$; return $\mathcal{T}_G(x)$; endif
• if $x \in X_{pk}$ and $\mathcal{V}(pk, x, z) = 1$
• $x' \leftarrow x$
exit game
endif
$r \leftarrow K_\ell$
insert (x, r) in table $\mathcal{T}_G$
return r

$H4(x, m)$

if (x, m) in $\mathcal{T}_H$; return $\mathcal{T}_H(x, m)$; endif
• if $x \in X_{pk}$ and $\mathcal{V}(pk, x, z) = 1$
• $x' \leftarrow x$
exit game
endif
$r \leftarrow Y_{pk}$
insert $((x, m), r)$ in table $\mathcal{T}_H$
return r

$\mathcal{D}4_{pk}(c)$

if $c \notin \overline{Z}_{pk} \times M_\ell$; return $\perp_1$; endif
$(c_1, c_2) = c$
• foreach x in $\mathcal{T}_G$
• if $x \in X_{pk}$ and $\mathcal{V}(pk, x, c_1) = 1$
$g \leftarrow \mathcal{T}_G(x)$
$m \leftarrow \mathsf{Dec}_g^{sym}(c_2)$
$y \leftarrow H4(x, m)$
if $f_{pk}(x, y) \neq c_1$; return $\perp_2$; endif
return m
endif
• endforeach
return $\perp_2$

These changes do not modify any probability. Moreover, the views of $\mathcal{A}^{\mathsf{IND-CCA}}$ in games 3 (with $\beta = 1$) and 4 are identically distributed. So,

$$\mathsf{Succ}\left[\mathcal{A}^{\mathsf{POW}}\right] = \mathsf{Pr}_4\left[x' = x^\star\right] \geq \mathsf{Pr}_4\left[\mathsf{S}_1\right] = \mathsf{Pr}_3\left[\mathsf{S}_1 \mid \beta = 1\right]$$

and, from the above results,

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}\right] \leq \mathsf{Succ}\left[\mathcal{A}^{\mathsf{POW}}\right] + 2\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right] + \\ + \mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}\right]' + \frac{3q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{3q_D}{|Y| - q_D}$$

In terms of time complexity of the algorithms, the overhead introduced by the simulation of the random oracles, G and H, into games 3 and 4 can be reduced by using standard hashing techniques for table insertion and searching. In fact, in almost all security proofs in the Random Oracle Model in the literature, this time overhead is neglected. It is also supposed that the time needed to check if $c \in \overline{Z}_{pk} \times M_\ell$ and $x \in X_{pk}$ is negligible.

Neglecting lower order terms, the running time of $\mathcal{A}^{\mathsf{POW}}$ in Game4 is bounded by

$$T[\mathcal{A}^{\mathsf{POW}}] \leq (q_G + q_H + q_D + q_G q_D)T[\mathcal{V}] + q_D\Big(T[f] + T[\mathsf{Dec}^{sym}]\Big) + T[\mathcal{A}^{\mathsf{IND-CCA}}]$$

where $T[\mathcal{V}]$ is the time complexity of the plaintext checking algorithm and $T[f]$ is the time complexity of f. Also, $T[\mathcal{A}^{\mathsf{IND-SYM}}] = T[\mathcal{A}^{\mathsf{IND-CCA}}]$.

A.1 Particular cases

Both in the case of the trivial construction of easy verifiable functions, and in the non-trivial family in subsection 4.1, the algorithm $\mathcal{D}4_{pk}$ can be improved without modifying the behavior of the game to avoid exhaustive search in $\mathcal{T}_G$. To do this, $(\tilde{f}(x), (x, G(x)))$ is stored in another table $\tilde{\mathcal{T}}_G$ for each query $x \in X_{pk}$ to $G4$.

$\tilde{G}4(x)$

```
 1  if x in T_G; return T_G(x); endif
 2  if x ∈ X_pk and f̃_pk(x) = π̃_pk(z)
 3      x' ← x
 4      exit game
 5  endif
 6  r ← K_ℓ
 7  insert (x, r) in table T_G
 8  if x ∈ X_pk
 9      insert (f̃_pk(x), (x, r)) in table T̃_G
10  endif
11  return r
```

$\tilde{\mathcal{D}}4_{pk}(c)$

```
1  if c ∉ Z̄_pk × M_ℓ; return ⊥_1; endif
2  (c_1, c_2) = c
3  c̃_1 ← π̃_pk(c_1)
4  if c̃_1 in T̃_G
```

- 5 $(x, g) \leftarrow \tilde{\mathcal{T}}_G(\tilde{c}_1)$
  6 $m \leftarrow \mathsf{Dec}^{sym}_{g}(c_2)$
  7 $y \leftarrow H4(x, m)$
  8 if $f_{pk}(x, y) \neq c_1$; return $\perp_2$; endif
  9 return m
  10 endif
  11 return $\perp_2$

The plaintext checking algorithm call $\mathcal{V}(pk, x, z)$ is replaced by the condition $\tilde{f}_{pk}(x) = \tilde{\pi}_{pk}(z)$, and $\tilde{\pi}_{pk}(z)$ for the target z can be precomputed by $\mathcal{A}^{\mathsf{POW}}$. Moreover, the same standard hashing techniques used in the simulation of G and H can also be used here to maintain $\tilde{\mathcal{T}}_G$, so the time overhead of step 4 in $\tilde{\mathcal{D}}4_{pk}$ and step 9 in $\tilde{G}4$ can be neglected. Then,

$$T[\mathcal{A}^{\mathsf{POW}}] \leq (q_G + q_H)T[\tilde{f}] + q_D\Big(T[f] + T[\tilde{\pi}] + T[\mathsf{Dec}^{sym}]\Big) + T[\mathcal{A}^{\mathsf{IND-CCA}}]$$