A General Construction of IND-CCA2 Secure Public Key Encryption*

Eike Kiltz[1] and John Malone-Lee[2]

[1] Lehrstuhl Mathematik & Informatik, Fakultät für Mathematik,
Ruhr-Universität Bochum, Germany.
E-mail: kiltz@lmi.rub.de
URL: http://www.rub.de/lmi/kiltz

[2] University of Bristol, Department of Computer Science,
Merchant Venturers Building, Woodland Road,
Bristol, BS8 1UB, UK.
E-mail: malone@cs.bris.ac.uk.

Abstract. We propose a general construction for public key encryption schemes that are IND-CCA2 secure in the random oracle model. We show that the scheme proposed in [1, 2] fits our general framework and moreover that our method of analysis leads to a more efficient security reduction.

1 Introduction

Since Diffie and Hellman proposed the idea of public key cryptography [13], one of the most active areas of research in the field has been the design and analysis of public key encryption schemes [4, 5, 11, 14–17, 20, 21]. Initially this research followed two separate paths: practical and theoretical. In [14, 21] efficient primitives were suggested from which to build encryption schemes. Formal models of security were developed in [16, 17, 20]. Schemes were designed in these models using tools from complexity theory to provide proofs of security. While the ideas were ground breaking, the schemes that were proposed where of theoretical interest only since they were not at all efficient.

In recent years much research has been done into methods for designing encryption schemes that are both practical and that may be analyzed formally [5, 11]. One approach that has enjoyed a great deal of success is the *random oracle model* proposed by Bellare and Rogaway [5]. In this model cryptographic hash functions are assumed to be perfectly random. Although a proof of security in this model is a heuristic argument, it is generally accepted as a demonstration of sound design, so much so that several schemes analyzed in this way have enjoyed widespread standardization [4, 6].

* This work was done during a stay of the two authors at BRICS in Aarhus, Denmark. The stay was supported by the Marie-Curie fellowship scheme of the European Union. Both authors wish to thank BRICS and the EU for making this visit possible.

In this paper we propose a very general construction for public key encryption. We prove, in the random oracle model, that our construction yields schemes with indistinguishable encryptions under adaptive chosen ciphertext attack (IND-CCA2) [20]. Our security result is tight. It makes use of the work of Fujisaki and Okamoto [15]. We show that our scheme is a generalization of that proposed in [1, 2], moreover our method of analysis results in a more efficient security reduction.

The paper is organized as follows. In Section 2 we begin by discussing some security notions before defining an abstract computational problem that we call the *Y-computational problem* (YC). Many number-theoretic primitives in the literature fit our abstract definition.

In Section 3 we propose an encryption scheme which is secure in the sense of indistinguishable encryptions under chosen plaintext attack (IND-CPA). Our result is in the random oracle model. We recall the technique of Fujisaki and Okamoto [15] to transform an IND-CPA secure cryptosystem into one that is IND-CCA2 secure in Section 4. In Section 5 we apply this technique to our construction. Some concrete examples of our cryptosystem are given in Section 6. One example, based on the computational Diffie-Hellman problem, turns out to be the cryptosystem of [1, 2]. Our method of security analysis provides an improved reduction than that of [1, 2] however. We give a second example based on Rabin [19]. In this case encryption consists of one squaring and the security is equivalent to factoring.

2 Definitions

2.1 Security Notions

A public key encryption scheme Π consists of three algorithms $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ with the following properties.

- The *key generation algorithm*, $\mathcal{K}$, is a probabilistic algorithm that takes a security parameter $1^k \in \mathbb{N}$, represented in unary, and returns a pair $(\mathrm{pk}, \mathrm{sk})$ of matching public and secret keys.
- The *encryption algorithm*, $\mathcal{E}$, is a probabilistic algorithm that takes a public key pk and a message $m \in \{0,1\}^*$ to produce a ciphertext $c \in \{0,1\}^*$.
- The *decryption algorithm*, $\mathcal{D}$, is a deterministic algorithm that takes a secret key sk and a ciphertext $c \in \{0,1\}^*$ to produce either a message $m \in \{0,1\}^*$ or a special symbol $\perp$. The symbol $\perp$ is used to indicate that the ciphertext was invalid in some way.

The first formal security definitions for public key encryption appeared in [16]. In this work Goldwasser and Micali proved that, if an adversary running in polynomial time can not distinguish which of two chosen messages has been encrypted, then it can learn no information about a message from its ciphertext. This idea underpins all accepted definitions of security used today. The adversary of a public key encryption scheme is modeled as a probabilistic polynomial time algorithm $\mathcal{A}$ that runs in two stages: $\mathcal{A}_1$ and $\mathcal{A}_2$. In the first stage of its attack $\mathcal{A}_1$ is given a public key pk to attack. At the end $\mathcal{A}_1$ outputs two messages m_0 and m_1 of equal length. A bit b is chosen at random and m_b is encrypted under pk to produce a ciphertext c^*. In the second stage of the attack $\mathcal{A}_2$ is given c^* and asked to determine the bit b.

During the first stage $\mathcal{A}_1$ may be given a decryption oracle for the secret key corresponding to the public key it is attacking. This attack is a *non-adaptive chosen ciphertext attack* [17], or CCA1 for short. An attack in which the adversary $\mathcal{A}_2$ is also given the decryption oracle in the second stage is an *adaptive chosen ciphertext attack* (CCA2) [20]. If the adversary has no access to such oracles we call the attack a *chosen plaintext attack* (CPA).

We formalize these notions in Definition 1 below. Here we denote the fact that the encryption of one of two messages must be indistinguishable to the adversary by IND.

Definition 1. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary. For $atk \in \{cpa, cca1, cca2\}$ and $1^k \in \mathbb{N}$ let*

$$\mathrm{Adv}_{\mathcal{A},\Pi}^{ind-atk}(1^k) = 2 \cdot \Pr \begin{bmatrix} (\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{K}(1^k); \\ (m_0, m_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}_1}(\mathrm{pk}); \\ b \leftarrow \{0,1\}; \\ c^* \leftarrow \mathcal{E}_{\mathrm{pk}}(m_b); \\ \mathcal{A}_2^{\mathcal{O}_2}(m_0, m_1, c^*, state) = b \end{bmatrix} - 1$$

where

$$\begin{array}{lll} atk = cpa & \Rightarrow \mathcal{O}_1(\cdot) = \epsilon & \text{and } \mathcal{O}_2(\cdot) = \epsilon, \\ atk = cca1 & \Rightarrow \mathcal{O}_1(\cdot) = \mathcal{D}_{\mathrm{sk}}(\cdot) & \text{and } \mathcal{O}_2(\cdot) = \epsilon, \\ atk = cca2 & \Rightarrow \mathcal{O}_1(\cdot) = \mathcal{D}_{\mathrm{sk}}(\cdot) & \text{and } \mathcal{O}_2(\cdot) = \mathcal{D}_{\mathrm{sk}}(\cdot). \end{array}$$

We insist that $\mathcal{A}_1$ outputs m_0 and m_1 with $|m_0| = |m_1|$. Also, $\mathcal{A}_2$ is not permitted make the query $\mathcal{O}_2(c^*)$.

The encryption scheme Π is IND-ATK secure if $\mathcal{A}$ being polynomial-time implies that $\mathrm{Adv}_{\mathcal{A},\Pi}^{ind-atk}(1^k)$ is negligible.

We define the advantage function for the scheme

$$\mathrm{Adv}_{\Pi}^{ind-atk}(1^k, t, q_d) = \max\{\mathrm{Adv}_{\mathcal{A},\Pi}^{ind-atk}(1^k)\},$$

where the maximum is taken over all adversaries that run for time t and make at most q_d queries to the decryption oracle.

NOTE: In the ROM we also consider the number of RO queries made by an adversary in the advantage function.

Our construction will make use of a symmetric encryption scheme SE. This consists of two algorithms (E, D) with the following properties.

- The *encryption algorithm*, E, is a deterministic algorithm that takes a key $\kappa \in \{0,1\}^l$ and a message $m \in \{0,1\}^*$ to produce a ciphertext $c \in \{0,1\}^*$.
- The *decryption algorithm*, D, is a deterministic algorithm that takes a key $\kappa \in \{0,1\}^l$ and a ciphertext $c \in \{0,1\}^*$ to produce a message $m \in \{0,1\}^*$.

As in the public key case, the security definition for SE that we use is based on indistinguishability of encryptions. We give a formal definition in Definition 2 below. Here OTE means "one time encryption". Our definition is similar to the notion of find-then-guess security from [3]; however, in [3] an adversary may be able to access an encryption oracle for the key that it is attacking. We require security in a weaker sense where no such oracle is considered.

Definition 2. *Let $SE = (E, D)$ be a symmetric encryption scheme. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary that runs in two stages. Define*

$$\mathrm{Adv}^{ote}_{\mathcal{A},SE} = 2 \cdot \Pr\left[\begin{array}{c} \kappa \leftarrow \{0,1\}^l; \\ (m_0, m_1, state) \leftarrow \mathcal{A}_1(); \\ b \leftarrow \{0,1\}; \\ c^* \leftarrow \mathcal{E}_\kappa(m_b); \\ b \leftarrow \mathcal{A}_2(m_0, m_1, c^*, state) \end{array}\right] - 1.$$

We insist that $\mathcal{A}_1$ outputs m_0 and m_1 with $|m_0| = |m_1|$.

The encryption scheme SE is OTE secure if $\mathcal{A}$ being polynomial-time implies that $\mathrm{Adv}^{ote}_{\mathcal{A},SE}$ is negligible.

We define the advantage function for the scheme

$$\mathrm{Adv}^{ote}_{SE}(t) = \max\{\mathrm{Adv}^{ote}_{\mathcal{A},SE}\},$$

where the maximum is taken over all adversaries that run for time t.

2.2 Computational Problems

In Definition 3 below we define the general computational problem that our construction will use. Our formalization captures many of the most widely used cryptographic primitives such as RSA [21] and Diffie-Hellman [13]. Some illustrative examples are given following the definition. We call our general problem the Y-computational problem (YC). The reason for this can be seen in the shape of Figure 1.

Definition 3. *An instance generator $\mathcal{I}_{\mathrm{YC}}(1^k)$ for* YC *outputs a description of (S_1, S_2, f_1, f_2, t). Here S_1 and S_2 are sets with $|S_1| = k$,*

$$f_1, f_2 : S_1 \to S_2$$

are functions and $t : S_2 \to S_2$ is a (trapdoor) function such that for all $x \in S_1$, $t(f_1(x)) = f_2(x)$. The functions f_1, f_2 and t should be easy to evaluate and it should be possible to sample efficiently from S_1.

Let $\mathcal{A}$ be an adversary and define

$$\mathrm{Adv}_{\mathcal{A},\mathcal{I}_{\mathrm{YC}}}(1^k) = \Pr\left[\begin{array}{c}(S_1, S_2, f_1, f_2, t) \leftarrow \mathcal{I}_{\mathrm{YC}}(1^k);\\ x \leftarrow S_1;\\ f_2(x) \leftarrow A\big(S_1, S_2, f_1, f_2, f_1(x)\big)\end{array}\right].$$

We define the advantage function

$$\mathrm{Adv}_{\mathcal{I}_{\mathrm{YC}}}(1^k, t) = \max\{\mathrm{Adv}_{\mathcal{A},\mathcal{I}_{\mathrm{YC}}}(1^k)\}$$

where the maximum is taken over all adversaries that run for time t. We say that YC *is hard for $\mathcal{I}_{\mathrm{YC}}(1^k)$ if t being polynomial in k implies that the advantage function* $\mathrm{Adv}_{\mathcal{I}_{\mathrm{YC}}}(1^k, t)$ *is negligible in k. Figure 1 illustrates the hard Y-computational problem.*

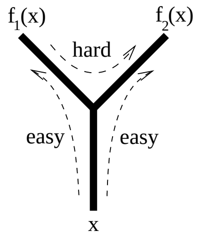

Fig. 1. The Y-computational problem: Given $f_1(x)$, compute $f_2(x)$.

2.3 Examples of hard Y-computational problems

In this subsection we show that many known cryptographic primitives fit the general definition of YC problems.

El Gamal. For the El Gamal cryptosystem, the instance generator $\mathcal{I}_{\mathrm{YC}}(1^k)$ computes a random k-bit prime p and a random generator g_1 of the multiplicative group $\mathbb{Z}_p^*$. The sets S_1 and S_2 are $\mathbb{Z}_p^*$ together with the generator g_1. A random value $s \in \{1, \ldots, p-1\}$ is chosen and $g_2 \in \mathbb{Z}_p^*$ is computed as $g_2 = g_1^s$. The functions f_1 and f_2 are defined as $f_1(x) = g_1^x$ and $f_2(x) = g_2^x$. The trapdoor function is $t(x) = x^s$.
Obviously, $t(f_1(x)) = (g_1^x)^s = g_1^{xs} = g_2^x = f_2(x)$ holds and YC is hard if the computational Diffie-Hellman assumption [13] holds.

Pointcheval [18]. For the Pointcheval cryptosystem, the instance generator $\mathcal{I}_{\mathrm{YC}}(1^k)$ computes a random k-bit composite $n = pq$. The sets S_1 and S_2 are $\mathbb{Z}_n^*$. A random exponent e is chosen with $\gcd(e, \varphi(n)) = 1$ and its inverse $d = e^{-1}$

modulo $\varphi(n)$ is computed. The functions f_1 and f_2 are defined as $f_1(x) = x^e$ and $f_2(x) = (x+1)^e$. The trapdoor function is $t(x) = (x^d + 1)^e$.
Obviously, $t(f_1(x)) = f_2(x)$ holds and YC is hard if the *computational dependent RSA problem* (see also [18]) is hard.

Arbitrary trapdoor oneway functions. Let $\mathcal{I}_{towf}$ be an instance generator for trapdoor oneway functions. Informally speaking, on input 1^k, $\mathcal{I}_{towf}$ outputs the description of two sets S_1 and S_2, together with a oneway function $f_1 : S_1 \to S_2$, and its trapdoor t such that $t(f_1(x)) = x$, for all $x \in S_1$. The functions f_1 and t should be easy to evaluate.
The instance generator $\mathcal{I}_{\mathrm{YC}}(1^k)$ runs $\mathcal{I}_{towf}$ on input 1^k, sets $f_2(x) = x$ (f_2 is the identity), and outputs (S_1, S_2, f_1, f_2, t) as an instance of YC. Then obviously YC is hard if inverting the oneway function f_1 is hard.
The two most important examples of trapdoor oneway functions are the RSA [21] and the Rabin [19] function. The latter is especially interesting for cryptographic purposes because its onewayness is provably equivalent to factoring.

RSA-Paillier [9]. For the RSA-Paillier cryptosystem, the instance generator $\mathcal{I}_{\mathrm{YC}}(1^k)$ computes a random k-bit composite $n = pq$. It outputs the sets $S_1 = \mathbb{Z}_n$ and $S_2 = \mathbb{Z}_n^*$. Then it computes a random exponent e with $\gcd(e, \varphi(n)) = 1$ and its inverse $d = e^{-1}$ modulo $\varphi(n)$. For $x = aN + b \in \mathbb{Z}_{N^2}^*$, we define $[x]_1$ as $b \in \mathbb{Z}_n$ and $[x]_2$ as $a \in \mathbb{Z}_n^*$. The functions f_1 and f_2 are defined as $f_1(x) = [x^e \bmod n^2]_1 = x^e \bmod n$ and $f_2(x) = [x^e \bmod n^2]_2$. The trapdoor function $t(x) = [(x^d \bmod N)^e \bmod n^2]_2$.
Obviously, $t(f_1(x)) = f_2(x)$ holds. In [10] it was shown that YC is hard if the RSA problem is hard.

3 IND-CPA under YC in the RO model

In this section we present a general construction of an IND-CPA secure cryptosystem based on the hardness of YC. The method uses a hash function which is modelled as a random oracle [5] in the security analysis.

Definition 4 (The Cryptosystem Π_0).

- *The key generator $\mathcal{K}(1^k)$ runs the instance generator $\mathcal{I}_{\mathrm{YC}}(1^k)$ for* YC *as in Definition 3 and outputs the description of (S_1, S_2, f_1, f_2) as the public key* pk. *The corresponding secret key* sk *is the trapdoor $t : S_2 \to S_2$.*
- *The cryptosystem uses a symmetric encryption scheme $SE = (E, D)$ with keys of length l. It also uses a hash function*
$$G : S_2 \to \{0,1\}^l.$$
- *The encryption function works as follows. Choose $x \xleftarrow{r} S_1$. Compute $\kappa = G(f_2(x))$.*
$$\mathcal{E}_{\mathrm{pk}}(m, x) = \big(f_1(x), E_\kappa(m)\big) = (\alpha, \beta).$$

– *To decrypt* (α, β) *one computes* $\kappa = G(t(\alpha))$ *and outputs*

$$D_\kappa(\beta).$$

The following Theorem proves the IND-CPA security of the encryption scheme Π_0 in the random oracle model.

Theorem 5 (IND-CPA security of Π_0). *For the encryption scheme Π_0 we have*

$$\mathrm{Adv}_{\Pi_0}^{ind-cpa}(t, q_g) \leq 2q_g \cdot \mathrm{Adv}_{\mathcal{I}_{\mathrm{YC}}}(1^k, t') + \mathrm{Adv}_{SE}^{ote}(t'),$$

where $t' \approx t$.

Proof. We prove the theorem by constructing algorithms using an adversary $\mathcal{A}$ as a subroutine to show that if $\mathcal{A}$ is to have any advantage then, with overwhelming probability, it must either solve an instance of YC or it must break the symmetric encryption scheme $SE = (E, D)$.

We begin by constructing an algorithm $\mathcal{B}$ to solve YC. Let us assume that $\mathcal{I}_{\mathrm{YC}}(1^k)$ has been run to produce (S_1, S_2, f_1, f_2, t) and that we are given the description of (S_1, S_2, f_1, f_2) and $X = f_1(x)$ for some $x \in S_1$. We make (S_1, S_2, f_1, f_2) the public key pk which $\mathcal{A}$ attacks.

The task of $\mathcal{B}$ is to compute $f_2(x)$. We run $\mathcal{B}$ responding to its hash queries with a simulator G_{sim}. Simulator G_{sim} keeps a list G_L of query/response pairs (y, κ) to maintain consistency between calls.

We may now describe $\mathcal{B}$.

Algorithm $\mathcal{B}(X)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1^{G_{sim}}(\mathrm{pk})$
$b \xleftarrow{r} \{0, 1\}$
$\kappa^* \xleftarrow{r} \{0, 1\}^l$
$\alpha^* \leftarrow X$
$\beta^* \leftarrow E_{\kappa^*}(m_b)$
$c^* = (\alpha^*, \beta^*)$
$b' \leftarrow \mathcal{A}_2^{G_{sim}}(m_0, m_1, c^*, state, \mathrm{pk})$
$(y, \kappa) \xleftarrow{r} G_L$
Return y

Let us now analyse our simulation. We will consider how $\mathcal{A}$ runs in a real run ($real$) and in our simulation (sim). We define an event ERR to be one that would cause $\mathcal{A}$'s view to differ in $real$ and sim. We have

$$\begin{aligned}
\Pr[\mathcal{A} \text{ wins} \wedge \neg\mathsf{ERR}]_{sim} &= \Pr[\mathcal{A} \text{ wins} \wedge \neg\mathsf{ERR}]_{real} \\
&\geq \Pr[\mathcal{A} \text{ wins}]_{real} - \Pr[\mathsf{ERR}]_{real} \\
&= \frac{1}{2} + \frac{1}{2}\mathrm{Adv}_{\mathcal{A},\Pi}^{ind-cpa} - \Pr[\mathsf{ERR}]_{real} \\
&= \frac{1}{2} + \frac{1}{2}\mathrm{Adv}_{\mathcal{A},\Pi}^{ind-cpa} - \Pr[\mathsf{ERR}]_{sim}. \qquad (1)
\end{aligned}$$

The final equality follows from that fact that, by definition of ERR, $\mathcal{A}$'s view in *real* and in *sim* are identical up until ERR occurs.

We now consider $\Pr[\mathsf{ERR}]_{sim}$. The event can only be caused by an error in G_{sim}. The only possible error here is caused by $\mathcal{A}$ making the query $t(X) = f_2(x)$ to which G_{sim} should respond κ^*. Moreover, if such a query is made algorithm $\mathcal{B}$ succeeds with probability $1/q_g$. We infer that

$$\Pr[\mathsf{ERR}]_{sim} \leq q_g \cdot \mathrm{Adv}_{\mathcal{B},\mathcal{I}_{\mathrm{YC}}}(1^k) \tag{2}$$

Let us now reconsider $\Pr[\mathcal{A}\text{ wins} \wedge \neg\mathsf{ERR}]_{sim}$. We show that $\mathcal{A}$ can have no advantage in this situation unless it can break the one-time security of the symmetric encryption function SE. To do this we construct an adversary $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ of SE. This adversary will again run $\mathcal{A}$ as a subroutine. The simulator to respond to $\mathcal{A}$'s queries to G will be as above.

Algorithm $\mathcal{C}_1()$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1^{G_{sim}}(\mathrm{pk})$
Return $(m_0, m_1, state)$

Now outside of $\mathcal{C}$'s view a random bit b is chosen and m_b is encrypted under a random key κ^* to produce c^*.

Algorithm $\mathcal{C}_2(m_0, m_1, c^*, state)$
$\alpha^* \stackrel{r}{\leftarrow} S_2$
$\beta^* \leftarrow c^*$
$c^* \leftarrow (\alpha^*, \beta^*)$
$b' \leftarrow \mathcal{A}_2^{G_{sim}}(m_0, m_1, c^*, state)$
Return b'.

The important things to note are first of all, in the event $\neg\mathsf{ERR}$, adversary $\mathcal{C}$ runs $\mathcal{A}$ in exactly the same way that the latter would be run in *sim*. Secondly, if $\mathcal{A}$ wins in *sim* then $\mathcal{C}$ wins. We infer that

$$\Pr[\mathcal{A}\text{ wins} \wedge \neg\mathsf{ERR}]_{sim} \leq \frac{1}{2} + \frac{1}{2}\mathrm{Adv}^{ote}_{\mathcal{C},SE}. \tag{3}$$

The result now follows from (1), (2), (3) and the construction of $\mathcal{B}$ and $\mathcal{C}$. $\square$

Now consider $\Pr[\mathsf{ERR}]_{sim}$ from equation (2). If we had access to an efficient verify algorithm $\mathcal{V}$ that on input $f_1(x_1)$ and $f_2(x_2)$, checks if $x_1 = x_2$, then we could drop this error probability to

$$\Pr[\mathsf{ERR}]_{sim} \leq \mathrm{Adv}_{\mathcal{B},\mathcal{I}_{\mathrm{YC}}}(1^k).$$

This is done by simply running $\mathcal{V}$ on input (y, X) for all queries y from the list G_L (that contains all queries made to the oracle G_{sim}). Indeed, if such an algorithm $\mathcal{V}$ exists (we say that YC has the "easy to verify" property), we get the improved result in Remark 6 below.

Remark 6. If YC has the "easy to verify" property, then for the encryption scheme Π_0 we have

$$\mathrm{Adv}_{\Pi_0}^{ind-cpa}(t, q_g) \leq 2 \cdot \mathrm{Adv}_{\mathcal{I}_{\mathrm{YC}}}(1^k, t') + \mathrm{Adv}_{SE}^{ote}(t'),$$

where $t' = t + q_G(T_{\mathcal{V}}(1^k) + O(k))$. Here $T_{\mathcal{V}}(1^k)$ denotes the running time of the verify algorithm $\mathcal{V}$.

Note that, with the exception of El Gamal, all the Y-computational problems presented in subsection 2.3 have the easy to verify property.

Remark 7. If one removes the symmetric encryption algorithm in the IND-CPA scheme of Definition 4, and merely output the symmetric key, then the scheme becomes a key encapsulation mechanism as introduced by Cramer and Shoup [12]. This is another approach to the problem of desiging IND-CCA2 secure encryption schemes.

4 The Fujisaki-Okamoto Transform

In [15] Fujisaki-Okamoto (FO) described a method to transform a cryptosystem with IND-CPA security into one with IND-CCA2 security. The method uses a hash function which is modelled as a random oracle [5] in the security analysis. The reduction is very tight. In this section we define the necessary notions and state the FO result.

Definition 8. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an IND-CPA secure cryptosystem. We define the transformed scheme $\Pi' = (\bar{\mathcal{K}}, \bar{\mathcal{E}}^H, \bar{\mathcal{D}}^H)$ as follows.*

- *The key generator $\bar{\mathcal{K}}(1^k)$ runs the key generator $\mathcal{K}(1^k)$.*
- *The cryptosystem uses a hash function*

$$H : \{0,1\}^* \to \{0,1\}^{k_0}$$

- *The encryption function works as follows. Choose $x \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$ and compute*

$$\bar{\mathcal{E}}_{\mathrm{pk}}^H(m, x) = \mathcal{E}_{\mathrm{pk}}\left((m||x), H(m||x)\right).$$

- *To decrypt the ciphertext c, one computes $m'||x' = \mathcal{D}_{\mathrm{sk}}(c)$ and outputs*

$$\bar{\mathcal{D}}_{\mathrm{sk}}^H = \begin{cases} m' & \textit{if } \bar{\mathcal{E}}_{\mathrm{pk}}^H(m', x') = c \textit{ and} \\ \perp & \textit{otherwise.} \end{cases}$$

Definition 9 (λ-uniformity). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key cryptosystem taking random input from $\{0,1\}^{k_0}$ and messages from $\{0,1\}^{\mathrm{mlen}}$. For given $x \in \{0,1\}^{\mathrm{mlen}}$ and $y \in \{0,1\}^*$, define*

$$\lambda(x, y) := \Pr_{x \stackrel{R}{\leftarrow} \{0,1\}^{k_0}} [y = \mathcal{E}_{\mathrm{pk}}(m, x)].$$

We say that Π is λ-uniform if, for any $x \in \{0,1\}^{\mathrm{mlen}}$ and $y \in \{0,1\}^*$, $\lambda(x, y) \leq \lambda$.

Fujisaki and Okamoto proved the following result about Π'.

Theorem 10 (IND-CCA2 security [15]). *Suppose that the encryption scheme Π' is λ-uniform. Then we have*

$$\mathrm{Adv}_{\Pi'}^{ind-cca2}(1^k, t, q_d, q_h) \leq \mathrm{Adv}_{\Pi}^{ind-cpa}(1^k, t') \cdot (1-\lambda)^{-q_d} + q_h \cdot 2^{-k_0-1}.$$

where $t' = t + q_h(T_{\mathcal{E}}(1^k) + O(k))$. Here $T_{\mathcal{E}}(1^k)$ denotes the running time of $\mathcal{E}_{\mathrm{pk}}(\cdot)$.

5 IND-CCA2 under YC in the RO model

As proved in Section 3, the cryptosystem Π_0 is IND-CPA secure in the random oracle model if the Y-computational problem YC is hard and the symmetric encryption function is OTE secure. It is now natural to apply the FO construction from the last section to this cryptosystem to get a cryptosystem Π_1 that is IND-CCA2 secure in the random oracle model. The construction uses two hash functions which are modelled as random oracles [5] in the security analysis. The reduction is very tight.

Definition 11 (The Cryptosystem Π_1).

- *The key generator $\mathcal{K}(1^k)$ runs the instance generator $\mathcal{I}_{\mathrm{YC}}$ for* YC *as in Definition 3 and outputs the description of (S_1, S_2, f_1, f_2) as the public key* pk. *The corresponding secret key* sk *is the trapdoor $t : S_2 \to S_2$.*
- *The cryptosystem uses a symmetric encryption scheme $SE = (E, D)$ with keys of length l. It also uses two hash functions*

$$G : S_2 \to \{0,1\}^l \ \text{and} \ \ H : \{0,1\}^* \to S_1.$$

- *The encryption function works as follows. Choose $x \xleftarrow{r} \{0,1\}^{k_1}$. Compute $h = H(m||x)$ and $\kappa = G(f_2(h))$.*

$$\mathcal{E}_{\mathrm{pk}}(m, x) = \big(f_1(h), E_\kappa(m||x)\big) = (\alpha, \beta).$$

- *To decrypt (α, β) one computes $\kappa = G(t(\alpha))$, $m'||x' = D_\kappa(\beta)$, $h' = H(m'||x')$.*

$$D_{\mathrm{sk}}(\alpha, \beta) = \begin{cases} m' & \text{if } \alpha = f_1(h') \\ \perp & \text{otherwise} \end{cases}$$

The symbol $\perp$ denotes the fact that the ciphertext was rejected.

The following Theorem proves the IND-CCA2 security of the encryption scheme Π_1 in the random oracle model.

Theorem 12 (IND-CCA2 security of Π_1). *For the encryption scheme Π_1 we have*

$$\mathrm{Adv}_{\Pi_1}^{ind-cca2}(1^k, t, q_d, q_h) \leq$$
$$(2q_g \cdot \mathrm{Adv}_{\mathcal{I}_{\mathrm{YC}}}(1^k, t') + \mathrm{Adv}_{SE}^{ote}(t')) \cdot (1 - 2^{-k})^{-q_d} + \frac{q_h}{2^{k_0+1}}.$$

where $t' = t + q_h(T_{\mathcal{E}}(1^k) + O(k))$. Here $T_{\mathcal{E}}(1^k)$ denotes the running time of $\mathcal{E}_{\mathrm{pk}}(\cdot)$.

The proof of this Theorem directly follows by applying the Theorem 10 and Theorem 5, and the following Lemma about the λ-uniformity of the cryptosystem Π_0.

Lemma 13. *The cryptosystem Π_0 from Definition 4 is 2^{-k}-uniform.*

Proof. By definition of λ-uniformity (see Definition 9), we have that

$$\begin{aligned}\lambda(m,\alpha,\beta) &= \Pr_{x \xleftarrow{R} \{0,1\}^{k_0}} \begin{bmatrix} \alpha = f_1(x) \\ \beta = E_\kappa(m) \end{bmatrix} \\ &\leq \Pr_{x \xleftarrow{R} \{0,1\}^{k_0}} [\alpha = f_1(x)] \\ &= \frac{1}{|S_1|} = \frac{1}{2^k}.\end{aligned}$$

□

Remark 14. If YC has the "easy to verify" property, then for the encryption scheme Π_1 we have

$$\mathrm{Adv}_{\Pi_1}^{ind-cca2}(1^k, t, q_d, q_h) \leq$$
$$\left(2 \cdot \mathrm{Adv}_{\mathcal{I}_{\mathrm{YC}}}(1^k, t') + \mathrm{Adv}_{SE}^{ote}(t')\right) \cdot (1 - 2^{-k})^{-q_d} + \frac{q_h}{2^{k_0+1}}.$$

where $t' = t + q_h(T_{\mathcal{E}}(1^k) + O(k)) + q_G(T_{\mathcal{V}}(1^k) + O(k))$. Here $T_{\mathcal{E}}(1^k)$ denotes the running time of $\mathcal{E}_{\mathrm{pk}}(\cdot)$. and $T_{\mathcal{V}}(1^k)$ denotes the running time of $\mathcal{V}(\cdot)$.

6 Examples

In this section we apply our construction of the cryptosystem Π_1 from section 5 to two important examples of instances of YC mentioned in Section 2.3, the El Gamal function and the Rabin function.

ENHANCED EL GAMAL ENCRYPTION SCHEME.

- The key generator $\mathcal{K}(1^k)$ runs the instance generator $\mathcal{I}_{\mathrm{YC}}$ for the El Gamal case and gets S_1 and S_2 as $\mathbb{Z}_p^*$ together with a generator g_1. Furthermore it gets $f_1(x) = g_1^x$ and $f_2(x) = g_2^x$. (S_1, S_2, f_1, f_2) form the public key pk. The corresponding secret key sk is the trapdoor $t(x) = x^s$ (where $s = \log_{g_1} g_2$).
- The cryptosystem uses a symmetric encryption scheme $SE = (E, D)$ with keys of length l. It also uses two hash functions

$$G : S_2 \to \{0,1\}^l \text{ and } H : \{0,1\}^* \to \mathbb{Z}_p^*.$$

- The encryption function works as follows. Choose $x \xleftarrow{r} \{0,1\}^{k_1}$. Compute $h = H(m||x)$ and $\kappa = G(g_2^h)$.

$$\mathcal{E}_{\mathrm{pk}}(m, x) = \left(g_1^h, E_\kappa(m||x)\right) = (\alpha, \beta).$$

– To decrypt (α, β) one computes $\kappa = G(\alpha^s)$, $m'||x' = D_\kappa(\beta)$, $h' = H(m'||x')$.

$$D_{\rm sk}(\alpha, \beta) = \begin{cases} m' & \text{if } \alpha = g_1^{h'} \\ \perp & \text{otherwise} \end{cases}$$

The symbol $\perp$ denotes the fact that the ciphertext was rejected.

Corollary 15. *In the random oracle model, the enhanced El Gamal encryption scheme is IND-CCA2 secure if the computational Diffie-Hellman problem is intractable and the symmetric encryption scheme SE is OTE secure.*

We note that our Enhanced El Gamal scheme is exactly that proposed in [1, 2] (we refer to it as the BKL-scheme henceforth), when we use the onetime pad as the symmetric encryption function SE (i.e., $E_\kappa(m) = \kappa \oplus m$). Moreover, our method of security reduction is *tight* (linear in terms of both, time and probability), opposed to that of [1, 2] which gives a reduction that is cubic in the time parameter.

In [1, 2] a comparison of the BKL-scheme is made with El Gamal encryption once the FO-transform has been applied. It is claimed that the BLK-scheme is preferable since its security is guaranteed by the computational Diffie-Hellman problem rather than the, possibly easier, decisional Diffie-Hellman problem [7]. This argument is misleading since, if G is a random oracle, the distributions $\left(g, g^a, g^b, G(g^{ab})\right)$ and $\left(g, g^a, g^b, G(g^r)\right)$ are indistinguishable if and only if the computational Diffie-Hellman problem is hard. It is easy to see that, with a random oracle a decisional problem comes for free from a computational problem.

Enhanced Rabin encryption scheme. As an example of how to use our scheme with a trapdoor-oneway function we use the Y-computational problem induced by the Rabin function.

Applying our result to the Rabin function requires care because square roots modulo $n = pq$ are not unique. To this end we slightly modify our decryption algorithm and make the trapdoor function t act from $S_2 \to S_2 \times S_2$.

– The key generator $\mathcal{K}(1^k)$ runs the instance generator $\mathcal{I}_{\rm YC}$ for Rabin and gets a modulus $n = pq$ where n is a $(k-1)$ bit number and p and q are two primes of roughly the same size with $p = q = 1 \bmod 4$. The set S_1 is

$$S_1 = \{1, \ldots, (n-1)/2\} \cap \mathbb{Z}_n^*,$$

and S_2 is the set of quadratic residues modulo n. The function f_1 is $f_1(x) = x^2$ and $f_2(x) = x$. The quadruple (S_1, S_2, f_1, f_2) forms the public key pk. The corresponding secret key sk is the trapdoor $t(x)$ which maps $y \in S_2$ to a pair $(z_1, z_2) \in S_1 \times S_1$ such that $z_1^2 = z_2^2 = y$ (z_1 and z_2 differ in their Jacobi symbol).
– The cryptosystem uses a symmetric encryption scheme $SE = (E, D)$ with keys of length l. It also uses two hash functions

$$G : S_2 \to \{0,1\}^l \text{ and } H : \{0,1\}^* \to S_1.$$

- The encryption function works as follows. Choose $x \xleftarrow{r} \{0,1\}^{k_1}$. Compute $h = H(m||x)$ and $\kappa = G(h)$.
$$\mathcal{E}_{\mathrm{pk}}(m,x) = \left(h^2, E_\kappa(m||x)\right) = (\alpha,\beta).$$
- To decrypt (α,β) one computes $(z_1,z_2) = t(\alpha)$ and for $i \in \{1,2\}$, $\kappa_i = G(z_i)$, $m_i'||x_i' = D_{\kappa_i}(\beta)$, $h_i' = H(m_i'||x_i')$.
$$D_{\mathrm{sk}}(\alpha,\beta) = \begin{cases} m_1' & \text{if } \alpha = (h_1')^2 \\ m_2' & \text{if } \alpha = (h_2')^2 \\ \perp & \text{otherwise} \end{cases}$$
The symbol $\perp$ denotes the fact that the ciphertext was rejected.

Corollary 16. *In the random oracle model, the enhanced Rabin encryption scheme is IND-CCA2 secure if the factoring problem is intractable and the symmetric encryption scheme SE is OTE secure.*

The encryption procedure of the Enhanced Rabin encryption scheme seems to be very efficient. When we neglect the cost of using the hash functions G and H, and the symmetric encryption scheme SE, the scheme uses only one squaring modulo n. Decryption requires two exponentiation, one modulo p and the other modulo q.

As already noted before, the "easy to verify" property (see Remark 6) is true in the case of the Rabin function, since $f_2(x) = x$ is the identity function. Therefore, the running time of the reduction algorithm is tight (Remark 14).

7 Conclusions

We have introduced a general construction for public key encryption schemes that are IND-CCA2 secure in the random oracle model. Our construction may be used with many of the number theoretic primitives in the literature. The scheme generalises that of [1, 2] and we have provided an improved security reduction.

There is some doubt concerning the meaning of a proof of security in the random oracle model. In [8] it is demonstrated that there exist cryptosystems that are provably secure in the random model, but insecure when the random oracle is instantiated with any hash function. Following from our remark in Section 6, it may be interesting to investigate the possibility of Y-computational problems where there is a separation between the computational problem and the decisional analogue no matter what hash function is used to instantiate the random oracle.

8 Acknowledgments

We would like to thank Ronald Cramer for useful advice and the anonymous referees for their suggestions of how to improve the paper.

References

1. J. Baek and B. Lee and K. Kim. Provably secure length-saving public-key encryption scheme under the computational Diffie-Hellman assumption. Electronics and Telecommunications Research Institute (ETRI) Journal, Vol 22, No. 4, Dec., pages 25-31, 2000.
2. J. Baek, B. Lee, and K. Kim. Secure Length-Saving El Gamal Encryption Under the Computational Diffie-Hellman Assumption. In Proceedings of the Fifth Australian Conference on Information Security and Privacy (ACISP 2000), volume 1841 of Lecture Note in Computer Science, pages 49-58. Springer-Verlag, 2000.
3. M. Bellare and A. Desai and E. Jokipii and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption. In 38^{th} Annual Symposium on Foundations of Computer Science, pages 394-403. IEEE Computer Science Press, 1997.
4. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption - How to Encrypt with RSA. In Advances in Cryptology - EUROCRYPT '94, volume 950 of Lecture Notes in Computer Science, pages 92-111. Springer-Verlag, 1994.
5. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of the First ACM Conference on Computer and Communications Security, pages 62-73. 1993.
6. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures - How to sign with RSA and Rabin. In Advances in Cryptology - EUROCRYPT '96, volume 1070 of Lecture Notes in Computer Science, pages 399-416. Springer-Verlag, 1996.
7. D. Boneh. The Decisional Diffie-Hellman Problem. In proceedings of the 3^{rd} Algoritmic Number Theory Symposium, volume 1423 of Lecture Notes in Computer Science, pages 48-63. Springer-Verlag, 1998.
8. R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology, Revisited. In Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing - STOC '98, pages 209-218. ACM, 1998.
9. D. Catalano, R. Gennaro, N. Howgrave-Graham, and P. Q. Nguyen. Paillier's Cryptosystem Revisited. Proceedings of the 8th ACM Conference on Computer and Communications Security, 2001.
10. D. Catalano, P. Q. Nguyen, and J. Stern. The Hardness of Hensel Lifting: The Case of RSA and Discrete Logarithm. In Advances in Cryptology - ASIACRYPT 2002, volume 2501 of Lecture Notes in Computer Science, pages 299-310. Springer Verlag, 2002.
11. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In Advances in Cryptology - CRYPTO '98, volume 1462 of Lecture Notes in Computer Science, pages 13-25. Springer-Verlag, 1998.
12. R. Cramer and V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext attack. To appear, SIAM Journal of Computing.
13. W. Diffie and M. E. Hellman. New Directions in Cryptography. In IEEE Transactions on Information Theory, volume IT-22(6), pages 644-654. 1976.
14. T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In IEEE Transactions on Information Theory, volume IT-31, pages 469-472. 1985.
15. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In Public Key Cryptography - PKC '99, volume 1560 of Lecture Notes in Computer Science, pages 53-68. Springer-Verlag, 1999.

16. S. Goldwasser and S. Micali. Probabilistic Encryption. In Journal of Computer and System Sciences, volume 28, pages 270-299. 1984.
17. M. Naor and M. Yung. Public-key Cryptosystems Provably Secure Against Chosen Ciphertext Attack. In Proceedings of 22^{nd} ACM Symposium on Theory of Computing, pages 427-437. ACM Press, 1990.
18. D. Pointcheval. New Public Key Cryptosystems based on the Dependent-RSA Problems. In Advances in Cryptology - Proceedings of EUROCRYPT '99, volume 1592 of Lecture Notes in Computer Science, Pages 239-254, Springer-Verlag, 1999.
19. M. O. Rabin. Digitalized signatures and public key cryptosystems as intractable as factorization. MIT/LCS/TR-212, Technical Report MIT. 1979.
20. C. Rackoff and D. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In Advances in Cryptology - CRYPTO '91, volume 576 of Lecture Notes in Computer Science, pages 433-444. Springer-Verlag, 1992.
21. R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In Communications of the ACM, volume 21(1), pages 120–126. 1978.
22. Y. Tsiounis and M. Yung. On the Security of El Gamal Based Encryption. In Public Key Cryptography '98, volume 1431 of Lecture Notes in Computer Science, pages 117-134, Springer-Verlag, 1998.