Universitat Politècnica de Catalunya

Department of Applied Mathematics IV

Ph. D. Thesis

Design and Analysis of Semantically Secure Public Key Encryption Schemes

Author: David Galindo Chacón

Advisor: Sebastià Martín Molleví

Jury: Josep Domingo-Ferrer (Universitat Rovira i Virgili)
Rosario Gennaro (IBM Research)
Carles Padró (Universitat Politècnica de Catalunya)
Jordi Quer (Universitat Politècnica de Catalunya)
Tsuyoshi Takagi (Technische Universität Darmstadt)

May 2004

ABSTRACT. An *encryption scheme* is a procedure that enables two parties to securely communicate over a public channel, in such a way that if a malicious party intercepts the information exchanged, it cannot extract the original message. In public key cryptography, the keys needed to encrypt and decrypt are different, the encryption key being public, thus available to legitimate and illegitimate users. Although encryption schemes are basic objects in public key cryptography and have been studied since the birth of this subject, the current demanding security notions and some recent developments in cryptanalysis makes designing encryption schemes an active research area. In this work, encryption schemes with *semantic security* are studied. On the one hand, new schemes are proposed and analysed and, on the other hand, some relevant previous schemes are revisited.

Acknowledgements

First of all, I would like to thank Sebastià Martín, Paz Morillo and Jorge Luis Villar for the fruitful and nice time we have had working together. Thanks also to the members of the research group Mathematics Applied to Cryptography (MAK) at the Universitat Politècnica de Catalunya, with whom this research has been developed. I want also to thank other researchers for sending preprints of their work or for providing fruitful comments: Tsuyoshi Takagi, Dario Catalano, Ronald Cramer and Alex W. Dent among others.

Introduction

Nowadays there is a widespread use of information technologies in many areas of the world, and securing this exchange of information has become a crucial task. Cryptography plays here a fundamental role. Concerned in the beginning with providing privacy when two parties communicate over an insecure channel, the arrival of *public key cryptography* in the late 70's extended the matters that cryptography can deal with. As Oded Goldreich suggests in the introduction to [Gol01], "cryptography can be viewed as concerned with the design of any system that needs to withstand malicious attempts to abuse it". In this section some basic concepts about modern cryptography are briefly presented, as well as the topics to which our research is devoted.

Encryption schemes

An *encryption scheme* is a procedure that enables two parties to securely communicate over a public channel, in such a way that if a malicious party intercepts the information exchanged (commonly called ciphertext and denoted by c), it cannot extract the original message (also called plaintext and denoted by m), while the intended recipient can recover it efficiently. A little more formally, an encryption scheme consists at least of an encryption algorithm Enc, run by the sender, with input a encryption key ek and a message m, and output a ciphertext c; and a decryption algorithm Dec, run by the receiver, with input a decryption key dk and ciphertext c, and output a string m. The minimal requirement these algorithms must satisfy is that $\mathsf{Dec}(\mathsf{dk}, \mathsf{Enc}(\mathsf{ek}, m)) = m$.

The main difference between a legitimate and illegitimate party is that the former is in possession of the decryption key dk, which is kept secret. Generally, the parameters that must be kept secret are contained in the *secret key*. As is well-known, encryption schemes are divided into two categories, depending on whether encryption and decryption keys coincide: *symmetric* or *private-key* encryption schemes, where $\mathsf{ek} = \mathsf{dk}$, and *asymmetric* or *public key* encryption schemes, in which ek and dk are different. In this work we study public key schemes, in which the encryption key is made public (hence the name), i.e. available to all users, including the adversaries, while the decryption key is only known by the recipient of the communication. The main advantage of an

asymmetric scheme with respect to a symmetric scheme is that the parties do not need to agree in a common key before communicating, but it has the disadvantage of being hundreds of times slower than symmetric schemes. For this reason, asymmetric schemes are not suitable for ciphering long messages. A way to solve this problem is to design a so-called *hybrid scheme*, that is, a public key scheme HE obtained by securely integrating an asymmetric scheme PKE and a symmetric scheme SKE. In this case, the asymmetric scheme is used to encrypt a key κ, usually referred to as *session key*, while the symmetric scheme encrypts the long message under key κ.

Provable security

Parties in a cryptographic protocol are modeled as algorithms executed by a computer. Specifying its computational capacity is a primary step to determine which tasks the parties can efficiently perform. In other words, we look at cryptography from a *complexity-theoretic* point of view. Roughly speaking, efficient computations are those that can be carried out by algorithms that run in polynomial time. In this context, some well-defined problems arise that are conjectured to be unsolvable in polynomial time without the knowledge of some secret information. These problems are used to design cryptographic protocols, and will be referred to as *atomic primitives*.

To devise a way to prove formally that a given cryptographic protocol is *secure* has required a lot of effort from researchers. The techniques developed to solve this question have led to the so-called *provable security* paradigm. The idea of provable security was introduced in the pioneering work of Goldwaser and Micali [GM84]. Bellare explains this paradigm in [Bel98] as follows: take some cryptography goal, like achieving privacy via encryption. The first step is to make a formal adversarial model and *define* what it means for an encryption scheme to be secure. With this in hand, a particular scheme, based on some particular atomic primitive, can be analyzed from the point of view of meeting the definition. Eventually, one shows that the scheme "works" via a reduction. The reduction shows that the *only way* to defeat the protocol is to break the underlying atomic primitive. In other words, there is no need to directly cryptanalyze the protocol: if it were possible to find a weakness in it, there would be an unexpected one in the underlying atomic primitive. So one might as well focus on the atomic primitive; and if we believe the latter to be secure, we know without further cryptanalysis of the protocol that the protocol is secure.

An important point in the last step is that in order to enable a reduction one must also have a formal notion of what is meant by the security of the underlying atomic primitive: which attacks exactly does it resist?

The main generic disadvantage of the schemes delivered by the standard provable security approach is that they are inefficient. For this reason, standard makers did not

take it into consideration and simply worked for several years by trial and error. Only in the late nineties, when subtle attacks against standardized schemes were found in [Ble98, CNS99, CHJ99], standard bodies were convinced this ad-hoc approach was not correct. In this way, Bellare and Rogaway introduced in [BR93] the Random Oracle Model (ROM), an idealized model of computation aiming to bridge the gap between provable security and efficiency. In this model, concrete objects such as cryptographic hash functions are treated as *truly* random functions. This allows us to derive security proofs more easily and, usually, the schemes in this model are simpler and more efficient. The problem is that the significance of a proof in the ROM is somewhat debatable, since hash functions are deterministic and hence not even probabilistic. For this reason, security proofs using the ROM cannot be considered as actual proofs but rather as heuristic arguments. The idea is that a proof in the ROM gives a good indication about the security of a protocol and, in fact, several standardisation related efforts such as NESSIE [Nes03] or ISO/IEC [Sho04] accept cryptographic protocols with proofs in this model.

With the deployment of this model, the idea of a concrete or quantitative treatment of security arises. One would like to derive concrete estimates from the proof: if a reduction is efficient, the security loss is small and the existence of an efficient adversary leads to an algorithm for solving the underlying mathematical problem, which is almost as efficient. Then, the more efficient the reduction is, the shorter the key size of the scheme. For this reason, the concrete security performance has become a very important feature when evaluating a cryptographic protocol.

Our contributions

This thesis is devoted to the study of public key encryption schemes with semantic security within the provable security paradigm. We deal with the design of new schemes with appealing features as well as with the careful revision of some existing proposals.

The reader is assumed to be familiar with cryptography basics, but the main conceptual tools needed to describe our research are included in the exposition. In this sense, this document is aimed at being self-contained. We emphasize that our approach is both theoretical and practical, that is, we state definitions and theorems in a rigorous way, but we have in mind that, in the end, these theoretical results must be applied in a real setting. Therefore, cryptographic practice plays an important role in our discussions.

The rest of this work is organized as follows. In Chapter 1 the fundamental concepts of provable security for public key encryption are presented. This includes basics from probability and complexity theories, formal definitions of symmetric and asymmetric encryption schemes, security notions and mathematical hard problems to build secure protocols. For the sake of completeness, a model for hybrid encryption design, known

as KEM-DEM methodology, is described. This model has gained wide acceptance in the cryptographic community.

In Chapter 2, two new schemes with semantic security against passive adversaries in the standard model are presented. Both schemes base their security in factoring related hard problems and have a fast encryption. In the first place, we present Rabin-Paillier scheme [GMMV02], whose encryption has one-wayness equivalent to factoring. We also construct a new trapdoor permutation based on factoring, which has interest in itself. The semantic security of the scheme is based on an appropiate decisional assumption, named as Decisional Small $2e$-Residues assumption. The robustness of this assumption is also discussed. We point out that an improvement of our results has been presented in [KT03]. In the second place, an elliptic curve scheme is proposed, named as Lifted-Rabin scheme [GMTV04]. It provides one-wayness equivalent to factoring and faster encryption than the previous known semantically secure elliptic curve schemes. Indeed, it is three times faster in encryption than the standard El Gamal scheme over elliptic curves. On the negative side, its decryption procedure is quite slow and presents large key sizes. We point out that several interesting techniques and cryptographic objects have been developed in its design. In this work some ideas from our previous research in [GMMV03b, GMMV03c] have been used.

Chapter 3 is devoted to revisiting some of the most relevant asymmetric schemes with semantic security against adaptive adversaries appearing in the literature. This includes both widely used theoretical results as well as schemes for commercial applications. On the one hand, we identify some ambiguities in the security proof of the popular conversion proposed by Fujisaki and Okamoto in 1999. From private and public key encryption schemes with weak security, this conversion designs a hybrid scheme with strong security. The importance of this conversion stems from the fact of being the most used generic conversion to date in the literature. In doing so, we continue with the careful revision of the provable security techniques initiated by Shoup in [Sho01], where this author questioned some properties of the OAEP scheme [BR95], which were accepted without proof. We modify the Fujisaki-Okamoto transformation to remove the ambiguities detected, and to prove that the resulting conversion is secure using the Random Oracle heuristic. The security proof is phrased using current widely accepted proof techniques. We also improve the concrete security with respect to certain primitives. This research has been published in [GMMV03a, GMMV04].

Furthermore, we re-evaluate the elliptic curve based KEMs presented to become standards (for instance in ISO/IEC 18033 and in the NESSIE project) which are called ACE-KEM, ECIES-KEM and PSEC-KEM. We analyse both their security properties and performance when elliptic curves with efficiently computable bilinear maps are used. It is also shown that ECIES-KEM arises as the best option among these KEMs when such curves are used. This is remarkable, since NESSIE [Nes03] did not se-

lect ECIES-KEM as a candidate for standardization. This work has been presented in [GMV04].

Contents

Chapter 1

Preliminaries

In this chapter we introduce the main tools needed to present our results. In this way, Section 1.1 includes some concepts about probability and complexity theory and definitions of basic cryptographic objects that are used in subsequent sections. Section 1.2 deals with the formal definitions involved in public key encryption schemes, containing also a widely accepted model for hybrid encryption. In Section 1.3, standard security definitions for encryption schemes are presented, while in Section 1.4 some mathematical assumptions to be used in protocol design are described. Widely used as well as recently proposed assumptions are included, some of them arising from this research.

1.1 Basic tools

Modern cryptography is a subject that takes basic ideas from probability and complexity theories to build its core concepts. In this section, the key components we need from these disciplines are presented. Most of this material resembles [DK02].

1.1.1 Finite probability spaces and random variables

Definition 1 (Probability space)

- *A* probability distribution *(or simply a* distribution*)* $p = (p_1, \ldots, p_n)$ *is a tuple of elements* $p_i \in \mathbb{R},\ 0 \leq p_i \leq 1$*, called* probabilities*, such that* $\sum_{i=1}^{n} p_i = 1$.
- *A* probability space (Ω, p_Ω) *is a finite set* $\Omega = \{\omega_1, \ldots, \omega_n\}$ *with a probability distribution* $p_\Omega = (p_1, \ldots, p_n)$*; that is,* $p_\Omega(\omega_i) = p_i$*.* Ω *is also called the* sample space.
- *An event* E *in a probability space* (Ω, p_Ω) *is a subset* E *of* Ω*. The probability measure is extended to events:* $p_\Omega(E) = \sum_{y \in E} p_\Omega(y)$.

Typically the probability space consists of the set of all binary strings of a certain length ℓ, taken with the uniform probability distribution. That is, the sample space is $\Omega = \{0,1\}^\ell$, and each string is assigned with probability measure $2^{-\ell}$. Let us denote by $\{0,1\}^*$ the set of all finite length binary strings.

Definition 2 (Conditional probability) *Let (Ω, p_Ω) be a probability space and $A, B \subseteq \Omega$ be events, with $p_\Omega(B) \neq 0$. The* conditional probability *of A assuming B is*

$$p_\Omega(A|B) = \frac{p_\Omega(A,B)}{p_\Omega(B)},$$

where separating events by commas means combining them with AND.

Definition 3 (Random variable) *Let (Ω, p_Ω) be a probability space. A map $X : \Omega \longrightarrow Y$ is called a Y-valued* random variable *on Ω. The* distribution p_X of a random variable X *is the image of p_Ω under X:*

$$p_X(y) = p_\Omega(\{\omega \in \Omega | X(\omega) = y\}) \text{ for } y \in Y.$$

Considering the distribution of a random variable $X : \Omega \longrightarrow Y$ means considering the distribution of the probability space induced as image on Y by X. It is usual to look at a Y-valued random variable as a probability distribution over Y.

Often in the literature the probability space is not specified when dealing with a random variable. For example, we may say that X is a random variable assigning values in the set of all strings, so that $\mathsf{Pr}[X = 00] = \frac{1}{3}$ and $\mathsf{Pr}[X = 0111] = \frac{2}{3}$. It is assumed that X depends on a certain probability space (Ω, p_Ω), but it is not needed to further specify it. As mentioned before, this probability space consists of all strings of a particular length. Typically, these strings represent random choices made by some randomized process (see Section 1.1.2), and the random variable is the output of the process.

Definition 4 (Expected value) *Let X be a random variable mapping to real numbers. Then its* expected value *or* mean *is* $\mathsf{E}[X] = \sum_{\omega \in \Omega} X(\omega) \cdot p_\Omega(\omega)$.

Definition 5 (Joint distribution)

– *Let $X_1, \ldots, X_r$ be random variables, defined over some (finite) probability space and each one with its image set. $X_1, \ldots, X_r$ are called* jointly distributed *if there is a joint probability distribution p_X of $X_1, \ldots, X_r$, that is,*

$$\mathsf{Pr}[X_1 = x_1, \ldots, X_r = x_r] = p_X(x_1, \ldots, x_r).$$

– *The marginal distribution p_i, $1 \leq i \leq r$, is the image of p_X under the projection*

$$\pi_i : X_1 \times \ldots \times X_r \longrightarrow X_i,\ (x_1, \ldots, x_r) \longrightarrow x_i$$

which means

$$p_i(x_i) = p_X(\pi_i^{-1}(x_i)),\ \text{for } 1 \leq i \leq r \text{ and } x_i \in X_i.$$

– *They are called* independent *if and only if*

$$p_X(x_1, \ldots, x_r) = \prod_{i=1}^{r} p_i(x_i) \text{ for all } (x_1, \ldots, x_r) \in X.$$

If $x \in \{0,1\}^*$, then $|x|$ denotes its length. If X is a set and p_X is a probability distribution over X, then $x \stackrel{p_X}{\leftarrow} X$ denotes that x has been randomly chosen from X with the distribution p_X. In particular, the expression $x \leftarrow X$ implies the uniform distribution. If D is a Y-valued random variable, then $y \leftarrow D$ or $y \stackrel{p_D}{\longleftarrow} Y$ denote that y has been assigned a value from Y with distribution p_D.

1.1.2 Some basics from complexity theory

Our aim is to present a core concept in modern cryptography, the concept of feasible computations, that is, the class of computations we assume the parties in a protocol can perform, both legitimate users and adversaries. Our definitions are not completely formal but suffice for the standard purposes. The reason is that the formal approach needs the concept of *Turing machine*, a way to model a *deterministic algorithm*. This is out of our scope, and we refer the interested reader to [HU79]. For us, a deterministic algorithm $\mathcal{A}$ behaves like a mathematical mapping from strings to strings: applying $\mathcal{A}$ to the same input x several times always yields the same output y, which is computed by a sequence of steps decided in advance by the programmer. In contrast, a *probabilistic algorithm* $\mathcal{A}$ may yield different outputs when applied more than once to the same input x. We emphasize that both the input and the output of an algorithm are represented as bit strings.

Definition 6 (Probabilistic algorithm) *Given an input x, a* probabilistic algorithm $\mathcal{A}$ *may toss a coin a finite number of times during its computation of the output y, and the next step may depend on the results of the preceeding coin tosses. The number of coin tosses may depend on the outcome of the previous ones, but it is bounded by some constant t_x, for a given input x. The coin tosses are independent and the coin is fair, that is, each side appears with probability $1/2$.*

Remarks:

– Another way to view a probabilistic algorithm $\mathcal{A}$ is to consider the outcome of the coin tosses as an additional input. We call the corresponding deterministic algorithm $\mathcal{A}_D$ the *deterministic extension* of $\mathcal{A}$, taking as inputs the original x and a string r containing the coin tosses.

– Given x, the output $\mathcal{A}(x)$ is not a single constant value, but a Y-valued random variable, provided the outputs of $\mathcal{A}$ are in Y. Then, we can ask for the probability of the event "$\mathcal{A}$ outputs y on input x". From the definition of a probabilistic algorithm, the number of coin tosses for a given x is bounded by a constant t_x. We can assume this number is exactly t_x, and then coin tosses can be viewed as given by the uniform distribution in $\{0,1\}^{t_x}$. The probability of an outcome r is $1/2^{t_x}$, and hence

$$\mathsf{Pr}\left[\mathcal{A}(x) = y\right] = \frac{|\{r \in \{0,1\}^{t_x} \mid \mathcal{A}_D(x,r) = y\}|}{2^{t_x}}\,.$$

– Let p_X be a probability distribution on the domain X of a probabilistic algorithm $\mathcal{A}$ with outputs in Y. Randomly selecting an $x \in X$ with distribution p_X and computing $\mathcal{A}(x)$ can be viewed as a random variable over Y. We can define then a probability distribution over Y:

$$p_{\mathcal{A},p_X}(y) = \mathsf{Pr}\left[\mathcal{A}(x) = y \mid x \overset{p_X}{\leftarrow} X\right].$$

When describing the behaviour of a probabilistic algorithm, it is also useful to view its running time for any input x as a random variable, denoted $T_{\mathcal{A}}(x)$. Let $\mathsf{poly}(\ell)$ be the class of functions $p : \mathbb{Z}^+ \to \mathbb{R}^+$ upper bounded in $\mathbb{Z}^+$ by some polynomial in $\mathbb{R}[\ell]$. Hereafter ℓ denotes a positive integer.

Definition 7 (PPT algorithms) *A* probabilistic polynomial time (PPT) *algorithm is a probabilistic algorithm $\mathcal{A}$, such that $T_{\mathcal{A}}(x)$ is bounded by $P(|x|)$, where $P(\ell) \in \mathsf{poly}(\ell)$. The running time is measured as the number of steps in our model of algorithms, i.e. the number of steps of the probabilistic Turing machine. Tossing a coin is one step in this model.*

Definition 8 (PT algorithm) *A* polynomial time (PT) *algorithm is a a deterministic algorithm $\mathcal{A}$, such that $T_{\mathcal{A}}(x)$ is bounded by $P(|x|)$, where $P(\ell) \in \mathsf{poly}(\ell)$. The running time is measured as defined above.*

The following definition provides a useful tool to analyze the output distribution and running time of particular algorithms.

Definition 9 (Expected running time) *The* expected running time *of a* PPT *algorithm $\mathcal{A}$ is defined as $\mathsf{E}\left[T_{\mathcal{A}}(x)\right]$, i.e the expected value of $T_{\mathcal{A}}(x)$, the random variable measuring its running time.*

The concept of a *negligible function* is a key step in order to define a feasible computation.

Definition 10 (Negligible function) *The class of* negligible functions *on a parameter* $\ell \in \mathbb{Z}^+$*, denoted as* $\mathsf{negl}(\ell)$*, is the set of functions* $\varepsilon : \mathbb{Z}^+ \to \mathbb{R}^+$ *such that, for any polynomial* $p \in \mathbb{R}[\ell]$*, there exists* $M \in \mathbb{R}^+$ *such that* $\varepsilon(\ell) < \frac{M}{p(\ell)}$ *for all* $\ell \in \mathbb{Z}^+$*.*

When an event happens with probability at least $1 - \nu(\ell)$, where $\nu(\ell)$ is a negligible function, we say it occurs with *overwhelming* probability. Roughly speaking, a computation is *feasible* when it can be carried out by a PPT algorithm with overwhelming success probability with respect to the size of the input. Formally,

Definition 11 (Feasible computation) *A computational problem* $\mathcal{P}$ *is* feasible *if there exists a* PPT *algorithm* $\mathcal{A}$ *such that for any instance* x *of* $\mathcal{P}$*,* $\mathcal{A}(x)$ *yields the correct answer with overwhelming probability with respect to* $|x|$*.*

This definition may seem too restrictive, since we are asking that $\mathcal{P}$ is feasible if and only if any instance of $\mathcal{P}$ can be correctly computed with probability almost 1. However, the following lemma states that it suffices to correctly answer with probability $1/2$ plus a non-negligible quantity.

Lemma 12 *Let* $P, Q \in \mathsf{negl}(\ell)$ *and* $\mathcal{A}$ *be a* PPT *algorithm which computes a function* $f : X \longrightarrow Y$*, with*

$$\mathsf{Pr}\left[\mathcal{A}(x) = f(x)\right] \geq \frac{1}{2} + \frac{1}{P(|x|)} \text{ for all } x \in X.$$

Then, by repeating the computation $\mathcal{A}(x)$ *and returning the most frequent result, we obtain a* PPT *algorithm* $\tilde{\mathcal{A}}$*, such that*

$$\mathsf{Pr}\left[\tilde{\mathcal{A}}(x) = f(x)\right] \geq 1 - \frac{1}{Q(|x|)} \text{ for all } x \in X.$$

Proof: See [DK02] pp. 118–119. ∎

Finally, we state and prove a useful lemma when dealing with two non-independent calls to a probabilistic algorithm, which is a common situation in cryptographic reductions. In this case, it is no longer possible to use independence to compute the resulting success probability. The proof is quite technical, and the reader can skip it without affecting his/her understanding of the rest.

Lemma 13 *Consider a probabilistic algorithm $\mathcal{A}$ with input $x \in X$, a surjective map $f : X \to Y$ and a predicate P on the input and the output of $\mathcal{A}$ (e.g. $P(x, \mathcal{A}(x))$, which is true if $A(x)$ is the correct output). Let $\varepsilon = \mathsf{Pr}\left[P(x, \mathcal{A}(x)) \mid x \leftarrow X\right]$. Then,*

$$\mathsf{Pr}\left[P(x_1, \mathcal{A}(x_1)) \wedge P(x_2, \mathcal{A}(x_2)) \mid x_1 \leftarrow X;\ x_2 \leftarrow f^{-1}(f(x_1))\right] \geq \varepsilon^2$$

where the internal random coins used by $\mathcal{A}$ in the two calls are independent.

Proof: Let $w_y = \mathsf{Pr}\left[f(x) = y \mid x \leftarrow X\right]$ and $\varepsilon_y = \mathsf{Pr}\left[P(x, \mathcal{A}(x)) \mid x \leftarrow f^{-1}(y)\right]$, for $y \in Y$. Then $\sum_{y \in Y} w_y = 1$ and $\sum_{y \in Y} w_y \varepsilon_y = \varepsilon$.

Given the following experiment: $x_1 \leftarrow X;\ x_2 \leftarrow f^{-1}(f(x_1))$, then

$$\begin{aligned}
&\mathsf{Pr}\left[P(x_1, \mathcal{A}(x_1)) \wedge P(x_2, \mathcal{A}(x_2))\right] = \\
&\quad = \sum_{y \in Y} \mathsf{Pr}\left[P(x_1, \mathcal{A}(x_1)) \wedge P(x_2, \mathcal{A}(x_2)) \wedge f(x_1) = y\right] = \\
&\quad = \sum_{y \in Y} \mathsf{Pr}\left[P(x_1, \mathcal{A}(x_1)) \wedge P(x_2, \mathcal{A}(x_2)) \mid f(x_1) = y\right] \mathsf{Pr}\left[f(x_1) = y\right]
\end{aligned}$$

But conditioning by $f(x_1) = y$ is equivalent to modifying the experiment into $x_1 \leftarrow f^{-1}(y);\ x_2 \leftarrow f^{-1}(y)$. So, in this probability space, x_1 and x_2 are identically distributed independent random variables and

$$\mathsf{Pr}\left[P(x_1, \mathcal{A}(x_1)) \wedge P(x_2, \mathcal{A}(x_2)) \mid f(x_1) = y\right] = \left(\mathsf{Pr}\left[P(x_1, \mathcal{A}(x_1)) \mid f(x_1) = y\right]\right)^2 = \varepsilon_y^2$$

By using for instance the Cauchy-Schwartz inequality for a suitable weighted inner product (i.e. $\mathbf{a} \cdot \mathbf{b} = \sum_{y \in Y} w_y a_y b_y$), it is straightforward to see that $\sum_{y \in Y} w_y \varepsilon_y^2 \geq \varepsilon^2$. ■

Observe that, although the two calls to $\mathcal{A}$ are not independent, they share part of the input. There may be then a positive correlation between their outputs due to the map f, which allows us to upper bound the success probability by the square of the success probability on a single call.

1.1.3 Atomic primitives

Speaking informally, a *one-way function* is a map $f : X \longrightarrow Y$ which is easy to compute but hard to invert. It is infeasible to compute pre-images of $y \in Y$. But if f is intended to be used for encryption purposes, then f must be a special one-way function, called *trapdoor one-way* (TOW) function. Knowing some information, the trapdoor information, it must be feasible to invert f, and f remains one-way only if this trapdoor is kept secret. In order to define one-wayness, we have to consider not only single

functions, but, more generally, families of functions defined over appropiate index sets. In the following, the components needed to define a TOW function are described, while the definition is given afterwards. The concept of a key pair generator, which plays a central role hereafter, deserves special attention.

A *polynomial size set* is a set sequence, $X = \{X_\ell\}_{\ell\in\mathbb{Z}^+}$, such that there exists a function $p_X(\ell) \in \mathsf{poly}(\ell)$, and $X_\ell \subseteq \{0,1\}^{p_X(\ell)}$ for all $\ell \in \mathbb{Z}^+$. A sequence of probability distributions $D = \{D_\ell\}_{\ell\in\mathbb{Z}^+}$ over X is *polynomial time samplable* (samplable for short) if there exists a PPT algorithm sampling X_ℓ with distribution D_ℓ for all $\ell \in \mathbb{Z}^+$. A polynomial size set X is *samplable* if there exists a PPT algorithm that on input 1^ℓ, outputs a uniformly distributed random element in X_ℓ. To simplify the notation, hereafter a *set* and a *distribution* will denote a *polynomial size set* and a *polynomial time samplable distribution* respectively, and subindexes will be omitted whenever possible.

Definition 14 (Keypair generator) *Let PK and SK sets be such that PK_ℓ are all disjoint, and assume that the parameter ℓ can be derived from $\mathsf{pk} \in PK_\ell$ by a deterministic* PT *algorithm. In this context, ℓ is called the* security parameter. *Assume also that* pk *can be obtained from $\mathsf{sk} \in SK_\ell$ by a deterministic* PT *algorithm. Let I be a samplable probability distribution over $PK \times SK$. The triple (PK, SK, I) will be called a* keypair generator.

Given a keypair generator, a *set family* X is defined as $\{X_{\mathsf{pk}}\}_{\mathsf{pk}\in PK}$ and a *map family* $f : X \to Z$ is defined as $\{f_{\mathsf{pk}} : X_{\mathsf{pk}} \to Z_{\mathsf{pk}}\}_{\mathsf{pk}\in PK}$. Notice that the elements in PK_ℓ can be interpreted as indexes, although they provide more information. Thereby pk is public and characterizes the sets $X_{\mathsf{pk}}, Z_{\mathsf{pk}}$ as well as the map f_{pk}. On the other hand, sk is kept secret.

Finally we specify how we deal with PPT algorithms whose domain X is a joint probability space $X_1X_2\ldots X_r$ constructed by iteratively joining fibers $X_{j|x_1\ldots x_{j-1}}$ to $X_1\ldots X_{j-1}$, where $x_j \leftarrow X_{j|x_1\ldots x_{j-1}}$ is the conditional distribution of $x_j \in X_{j|x_1\ldots x_{j-1}}$, assuming $X_1 = x_1, \ldots, X_{j-1} = x_{j-1}$. The notation

$$\mathsf{Pr}\left[A(x_1x_2\ldots x_r) = f(x_1x_2\ldots x_r) \mid x_1 \leftarrow X_1, x_2 \leftarrow X_{2|x_1}, \ \ldots\ , x_r \leftarrow X_{r|x_1\ldots x_{r-1}}\right]$$

means the probability of the event $A(x_1x_2\ldots x_r) = f(x_1x_2\ldots x_r)$ if first x_1 is randomly chosen, then x_2, then x_3, ... , and so on.

Definition 15 (TOW function) *Let (PK, SK, I) be a keypair generator, and X, Z be set families. A map family $f : X \to Z$ is called a* Trapdoor One-Way *function (with respect to the keypair generator) if:*

– *there exists a* PT *algorithm that on input (pk, x) outputs $f_{\mathsf{pk}}(x)$ for all $\mathsf{pk} \in PK$, $x \in X_{\mathsf{pk}}$.*

– *there exists a map family* $g : Z \to X,\ g = \{g_{\mathsf{sk}} : Z_{\mathsf{pk}} \to X_{\mathsf{pk}}\}$ *and a* PT *algorithm that on input* $(\mathsf{sk}, f_{\mathsf{pk}}(x))$ *outputs* $g_{\mathsf{sk}}(f_{\mathsf{pk}}(x)) = x$*, for all* $\mathsf{sk} \in SK$*,* $x \in X_{\mathsf{pk}}$*.*

– *for any* PPT *algorithm* $\mathcal{A}^{\mathsf{OW}}$*,*

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

The following definition, based on [Poi00], is somewhat related to the notion of probabilistic one-way encryption. Let (PK, SK, I) a keypair generator. Let X, Y, Z be set families, $f : X \times Y \to Z$ a family of injective maps and $g : Z \to X$ their partial inverses, i.e. $g_{\mathsf{sk}}(f_{\mathsf{pk}}(x, y)) = x$ for all possible pairs $(\mathsf{pk}, \mathsf{sk})$ generated by I and for all $x \in X_{\mathsf{pk}}$ and $y \in Y_{\mathsf{pk}}$.

Definition 16 (TPOW function) *The injective map family,* f*, is called a* Trapdoor Partial One-Way (TPOW) *function (with respect to the keypair generator) if:*

– *there exists a* PT *algorithm that on input* (pk, x, y) *outputs* $f_{\mathsf{pk}}(x, y)$ *for all* $\mathsf{pk} \in PK$*,* $x \in X_{\mathsf{pk}}$ *and* $y \in Y_{\mathsf{pk}}$*.*

– *there exists a* PT *algorithm that on input* $(\mathsf{sk}, f_{\mathsf{pk}}(x, y))$ *outputs* $g_{\mathsf{sk}}(z) = (x, y)$ *for all* $\mathsf{sk} \in SK$ *and for all* $z \in Z_{\mathsf{pk}}$*.*

– *for any* PPT *algorithm* $\mathcal{A}^{\mathsf{POW}}$*,*

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

The remaining definitions are important to state and to deal with *decisional assumptions* (cf. Section 1.4), a crucial building block in modern cryptography.

Definition 17 (Polynomial indistinguishability) *Let* D_1, D_2 *be probability distributions over a set* X*. Then* D_1 *and* D_2 *are* polynomially indistinguishable*, denoted as* $D_1 \approx D_2$*, if for any* PPT *algorithm* $\mathcal{A}$

$$\left|\mathsf{Pr}\left[\mathcal{A}(1^\ell, D_{1,\ell}) = 1\right] - \mathsf{Pr}\left[\mathcal{A}(1^\ell, D_{2,\ell}) = 1\right]\right| \in \mathsf{negl}(\ell).$$

Property 18 *Let* D_1, D_2 *be two families of probability distributions over a set* X *such that* $D_1 \approx D_2$*, and let* $g : X \to Y$ *be a bijection map such that* g *and* g^{-1} *can be computed in probabilistic polynomial time. Then* $D_1 \approx D_2$ *is equivalent to* $g(D_1) \approx g(D_2)$*.*

1.2 Encryption schemes

In this section the definitions of public and private key encryption schemes are given, as well as the definitions of some cryptographic primitives that are needed to describe a well known model for hybrid encryption.

1.2.1 Public and private key encryption schemes

Let us first define an asymmetric encryption scheme using the terminology introduced in the previous section.

Definition 19 (Asymmetric encryption schemes) *Let (PK, SK, I) a keypair generator. An* asymmetric encryption scheme PKE *consists of three algorithms*

$$(\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec}),$$

with sets $\mathcal{M}$, $\mathcal{R}$ and $\mathcal{C}$, with the following properties:

– *The keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\ell)$ are generated by using the sampling algorithm for I.*

– $\mathsf{PKE.Enc}$ *is a* PPT *encryption algorithm which, on inputs a public key $\mathsf{pk} \in PK$ and $m \in \mathcal{M}_{\mathsf{pk}}$, runs on a randomness $r \in \mathcal{R}_{\mathsf{pk}}$ and returns a ciphertext $c \in \mathcal{C}_{\mathsf{pk}}$.*

– $\mathsf{PKE.Dec}$ *is a* PT *deterministic decryption algorithm that, on inputs a secret key $\mathsf{sk} \in SK$, and c, returns a string m* [1]*. We require that if $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{PKE.KeyGen}(1^\ell)$, then*

$$\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m, r)) = m \text{ for all } (m, r) \in \mathcal{M}_{\mathsf{pk}} \times \mathcal{R}_{\mathsf{pk}}.$$

Definition 20 (Symmetric encryption schemes) *A* symmetric encryption scheme SKE *consists of three algorithms* $(\mathsf{SKE.KeyLen}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$*, with the following properties:*

– $\mathsf{SKE.KeyLen}$ *is a* PT *algorithm which on input a security parameter 1^ℓ returns a positive integer $SKE.KeyLen$.*

– $\mathsf{SKE.Enc}$ *is a* PT *encryption algorithm with inputs a symmetric key $\kappa \in \{0,1\}^{SKE.KeyLen}$ and a message $m \in \{0,1\}^*$, that is, an arbitrary length bit string. It outputs a ciphertext $c \in \{0,1\}^*$ with $|c| = |m|$.*

[1] This string can be a special string, meaning there was a failure in the execution of the algorithm. It will be referred to as the reject string or reject symbol.

– $\mathsf{SKE.Dec}$ *is a* PT *encryption algorithm with inputs a symmetric key* $\kappa \in \{0,1\}^{SKE.KeyLen}$ *and a ciphertext* $c \in \{0,1\}^*$*. It outputs a message* m *with* $|m| = |c|$ *or* reject*. We require the following* soundness *condition: for all* ℓ*, for all* $\kappa \in \{0,1\}^{SKE.KeyLen}$ *and for all* $m \in \{0,1\}^*$

$$\mathsf{SKE.Dec}(\kappa, \mathsf{SKE.Enc}(1^\ell, \kappa, m)) = m.$$

1.2.2 Hybrid encryption

We point out that the definition of an asymmetric encryption scheme implies that it has a restricted message space given a security parameter. Moreover, practice shows that asymmetric schemes are often hundreds of times slower than symmetric schemes. Consequently, a useful approach to design an efficient asymmetric scheme is to build a *hybrid encryption scheme* HE, where one uses asymmetric cryptographic techniques to encrypt a session key κ, which is then used to encrypt the actual arbitrary length message using symmetric cryptography.

In the sequel, a model introduced in [CS, Sho04] for designing hybrid schemes is presented. This model is built from two primitives: a *key encapsulation mechanism* KEM and a *data encapsulation mechanism* DEM, which are defined next. A KEM is an asymmetric primitive while a DEM is a symmetric primitive. Roughly speaking, a KEM is a mechanism to encrypt and decrypt random session keys, while a DEM uses this random key to efficiently encrypt and decrypt arbitrary length messages.

Definition 21 (Key encapsulation mechanisms) *Let* (PK, SK, I) *a keypair generator. A* key encapsulation mechanism KEM *consists of three algorithms*

$$(\mathsf{KEM.KeyGen}, \mathsf{KEM.Enc}, \mathsf{KEM.Dec}),$$

along with sets $\mathcal{K}$ *and* $\mathcal{C}$*, with the following properties:*

– *The keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KEM.KeyGen}(1^\ell)$ *are generated by using the sampling algorithm for* I*.*

– $\mathsf{KEM.Enc}$ *is a* PPT *encryption algorithm which, on inputs a public key* $\mathsf{pk} \in PK$*, returns a symmetric-key/encapsulation pair* $(K, C_0) \in \mathcal{K} \times \mathcal{C}$*.*

– $\mathsf{KEM.Dec}$ *is a* PT *deterministic decryption algorithm that, on inputs a secret key* $\mathsf{sk} \in SK$*, and an encapsulation* C*, returns a symmetric key* K *or* reject*. We require a* KEM *to be* sound*, that is, for all* ℓ*, for all* $(\mathsf{pk}, \mathsf{sk}) \in PK_\ell \times SK_\ell$ *and for any output* (K, C_0) *of* $\mathsf{KEM.Enc}$

$$\mathsf{KEM.Dec}(\mathsf{sk}, C_0) = K.$$

A key encapsulation mechanism also specifies a positive integer $KEM.KeyLen$, i.e. the length of the symmetric key in the output by KEM.Enc *and* KEM.Dec.

Definition 22 (Data encapsulation mechanisms) *A* data encapsulation mechanism DEM *consists of three algorithms* $(\mathsf{DEM.KeyLen}, \mathsf{DEM.Enc}, \mathsf{DEM.Dec})$, *with the following properties:*

– $\mathsf{DEM.KeyLen}(1^\ell)$ *specifies a key length $DEM.KeyLen$.*

– DEM.Enc *is a* PT *encryption algorithm which, on inputs a symmetric key*

$$K \in \{0,1\}^{DEM.KeyLen}$$

and a plaintext $m \in \{0,1\}^*$, returns a ciphertext $C_1 \in \{0,1\}^{|m|}$.

– DEM.Dec *is a* PT *deterministic decryption algorithm that, on inputs a symmetric key $K \in \{0,1\}^{DEM.KeyLen}$, and a ciphertext C_1, outputs a plaintext $m \in \{0,1\}^{|C_1|}$ or* reject. *We require the following* soundness *property: for all ℓ, for all $K \in \{0,1\}^{DEM.KeyLen}$ and for any $m \in \{0,1\}^*$*

$$\mathsf{DEM.Dec}(K, \mathsf{DEM.Enc}(K, m)) = m.$$

Definition 23 (Hybrid encryption scheme) *A* hybrid encryption scheme HE *is a family of asymmetric cyphers with $\mathcal{M} = \mathcal{R} = \mathcal{C} = \{0,1\}^*$ parametizered by the following* system parameters*: a key encapsulation mechanism* KEM *and a data encapsulation mechanism* DEM. *Any combination of* KEM *and* DEM *may be used, provided that $KEM.KeyLen = DEM.KeyLen$. The algorithms* $(\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec})$, *are specified as follows:*

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\ell)$	$C \leftarrow \mathsf{HE.Enc}(\mathsf{pk}, m)$	$m \leftarrow \mathsf{HE.Dec}(\mathsf{sk}, C)$
1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KEM.KeyGen}(1^\ell)$	1. $(K, C_0) \leftarrow \mathsf{KEM.Enc}(\mathsf{pk})$ 2. $C_1 \leftarrow \mathsf{DEM.Enc}(K, m)$ 3. Set $C = C_0 \| C_1$ 4. Output C	1. Parse C as (C_0, C_1); output reject otherwise 2. $K \leftarrow \mathsf{KEM.Dec}(\mathsf{sk}, C_0)$ 3. $m \leftarrow \mathsf{DEM.Dec}(K, C_1)$ 4. Output m

A concrete DEM design

In the sequel, a concrete design of a data encapsulation mechanism from a *symmetric encryption scheme* and a *one-time message authenticacion code* (MA) is described. First, the definition of a MAC algorithm is given and then the particular DEM construction is presented. Informally speaking, a MA algorithm allows us to verify the integrity of a given string.

Definition 24 (Message authentication code) *A* one-time message authentication code MA *is a scheme that defines two positive integers* $MA.KeyLen$ *and* $MA.MacLen$*, along with a function* MA.Eval*. This function takes a symmetric key* κ' *and a string* $T \in \{0,1\}^*$ *as inputs, and computes as output a string* MAC *of length* $MA.MacLen$*.*

Definition 25 (DEM2) DEM2 *is a family of data encapsulation mechanisms parametizered by a symmetric encryption scheme* SKE *and a message authenticacion code* MA*. The value* $DEM2.KeyLen$ *is defined as* $SKE.KeyLen + MA.KeyLen$*. The algorithms* (DEM2.Enc, DEM2.Dec)*, are specified in Table 1.1:*

$C_1 \leftarrow \mathsf{DEM2.Enc}(K, m)$	$m \leftarrow \mathsf{DEM2.Dec}(K, C_1)$
1. Parse K as $K = \kappa \|\| \kappa'$, $\|\kappa\| = SKE.KeyLen$ and $\|\kappa'\| = MA.KeyLen$	1. Parse K as $K = \kappa \|\| \kappa'$
2. $c \leftarrow \mathsf{SKE.Enc}(\kappa, m)$	2. If $\|C_1\| < MA.MacLen$ output reject
3. $MAC \leftarrow \mathsf{MA.Eval}(\kappa', c)$	3. Parse C_1 as $C_1 = c \|\| MAC$ where $\|MAC\| = MA.MacLen$
4. Set $C_1 = c \|\| MAC$	4. $MAC' \leftarrow \mathsf{MA.Eval}(\kappa', c)$
5. Output C_1	5. If $MAC \neq MAC'$ output reject
	6. $m \leftarrow \mathsf{SKE.Dec}(\kappa, c)$
	7. Output m

Table 1.1: DEM2 description

As we shall see in Section 1.3.2, the problem of designing data encapsulation mechanisms is essentially solved with this construction.

1.3 Security issues

In this section the most usual security notions for a encryption scheme are presented, as well as the relations among them. For the sake of completeness, two fundamental models arising from the cryptographic practice are covered.

1.3.1 Asymmetric encryption security models

In order to treat the security of a cryptographic scheme rigorously one must specify two things: the *power of the adversary* both in terms of computation (time, memory etc.) and in terms of access to the system, and what *breaking* the cryptosystem means. An encryption scheme access to the system means the type of attack (e.g. chosen plaintext or chosen ciphertext), and breaking the encryption scheme should specify the functionality the adversary has with respect to the plaintext. Examples of defining such functionalities are *one-wayness*, *semantic security* and *indistinguishability of encryptions*. We first formalize the notion of one-wayness for an asymmetric cryptosystem.

Definition 26 (One-way - OW) *Consider the following game that an adversary $\mathcal{A}^{\mathsf{OW}}$ plays against a system, using an asymmetric encryption scheme PKE with security parameter 1^ℓ.*

1. *The system runs $\mathsf{PKE.KeyGen}(1^\ell)$, generating a keypair $(\mathsf{pk}, \mathsf{sk})$ and passes the value pk to the attacker $\mathcal{A}^{\mathsf{OW}}$.*
2. *The system picks a message $m \leftarrow M_{\mathsf{pk}}$ and calculates the challenge ciphertext $c^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m)$. The system then passes $c^\star$ back to the attacker.*
3. *The attacker outputs a guess m' for the message m.*

Let

$$\mathsf{Succ}_{\mathcal{A}^{\mathsf{OW}}}[\mathsf{PKE}, \ell] = \mathsf{Pr}\left[\mathcal{A}^{\mathsf{OW}}(\mathsf{pk}, c^\star) = m \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\ell) \\ m \leftarrow M_{\mathsf{pk}},\ c^\star \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m) \end{array}\right].$$

PKE *is said to be* one-way *if for any* PPT *attacker* $\mathcal{A}^{\mathsf{OW}}$, $\mathsf{Succ}_{\mathcal{A}^{\mathsf{OW}}}[\mathsf{PKE}, \ell] \in \mathsf{negl}(\ell)$.

The rigorous treatment of the security of encryption schemes was initiated in the seminal work of Goldwasser and Micali [GM84], where they introduced two fundamental notions of security, semantic security and indistinguishability of encryptions. *Semantic security* is a computational analogue of Shannon's definition of perfect secrecy [Sha49]. It requires that whatever information about the plaintext that one may compute from the ciphertext and some a-priori information, it can be essentially computed as efficiently from the a-priori information alone (this specific formulation was suggested in [Gol93]). This definition is the most natural, because it directly addresses the user's concerns (i.e., that nothing can be gained by looking at the ciphertext). The formalization we give here resembles [GLN02].

Definition 27 (Semantic security - SS) *Consider the following game that a 2-stage adversary $\mathcal{A}^{\mathsf{SS}} = (\mathcal{A}_1, \mathcal{A}_2)$ plays against a system, using an asymmetric encryption scheme PKE with security parameter 1^ℓ.*

1. *The system runs* $\mathsf{PKE.KeyGen}(1^\ell)$*, generating a keypair* $(\mathsf{pk}, \mathsf{sk})$ *and passes the value* pk *to the attacker* $\mathcal{A}^{\mathsf{SS}}$.

2. *The algorithm* $\mathcal{A}_1$ *generates a* PT *samplable distribution* D_{pk} *over* M_{pk} *and two distinct* PT *computable functions* $h, f : M_{\mathsf{pk}} \to \{0,1\}^*$. h *specifies partial information (i.e. information leakage) regarding the plaintext that is given to the adversary, and* f *specifies some information that the adversary claims to be able to learn.*

3. *The system calculates the challenge ciphertext* $c^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m)$*, where* $m \leftarrow D_{\mathsf{pk}}$*, and returns* $(h(m), c^\star)$ *to the algorithm* $\mathcal{A}_2$*.*

4. *The attacker outputs a guess* $v \in \{0,1\}^*$ *for* $f(m)$*.*

PKE *is said to be* semantically secure *if for any 2-stage* PPT *attacker* $\mathcal{A}^{\mathsf{SS}}$*, there exists a benign 2-stage adversary* $\mathbf{A}^{\mathsf{SS}} = (\mathbf{A}_1, \mathbf{A}_2)$*, which follows the same game except that it is not given the ciphertext, and "performs as well" as the real attacker. That is,*

$$\mathsf{Pr}\left[\mathcal{A}_2(D_{\mathsf{pk}}, h, f, h(m), c^\star) = f(m) \;\middle|\; \begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\ell) \\ (D_{\mathsf{pk}}, h, f) \leftarrow \mathcal{A}_1(\mathsf{pk}) \\ m \leftarrow D_{\mathsf{pk}},\ c^\star \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m)\end{array}\right] <$$

$$\mathsf{Pr}\left[\mathbf{A}_2(D_{\mathsf{pk}}, h, f, h(m)) = f(m) \;\middle|\; \begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\ell) \\ (D_{\mathsf{pk}}, h, f) \leftarrow \mathbf{A}_1(\mathsf{pk}),\ m \leftarrow D_{\mathsf{pk}}\end{array}\right] + \mu(\ell),$$

where $\mu(\ell) \in \mathsf{negl}(\ell)$.

Notice that the benign adversary is given a perfectly secure encryption of the plaintext m, that is, it is being given nothing.

Indistinguishability of encryptions is a technical definition requiring that, for any two messages, it is infeasible to distinguish the encryption of the first message from the encryption of the second message. The importance of the technical definition of indistinguishability of encryptions stems from the fact that it was shown to be equivalent to semantic security in a certain attack scenario (cf. [GM84]), while being easier to work with and reason about.

Definition 28 (Indistinguishability - IND**)** *Consider the following game that a 2-stage adversary* $\mathcal{A}^{\mathsf{IND}} = (\mathcal{A}_1, \mathcal{A}_2)$ *plays against a system, using an asymmetric encryption scheme* PKE *with security parameter* 1^ℓ*.*

1. *The system runs* $\mathsf{PKE.KeyGen}(1^\ell)$*, generating a keypair* $(\mathsf{pk}, \mathsf{sk})$ *and passes the value* pk *to the attacker* $\mathcal{A}^{\mathsf{IND}}$.

2. *The algorithm $\mathcal{A}_1$ generates two distinct messages $m_0, m_1 \in M_{\mathsf{pk}}$ with the same length. Next, it submits m_0, m_1 to the system.*

3. *The system*

   (a) *Chooses a bit b uniformly at random from $\{0,1\}$.*

   (b) *Calculates the challenge ciphertext $c^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m_b)$ and returns this to the algorithm $\mathcal{A}_2$.*

4. *The attacker outputs a guess b' for b. The attacker wins the above game if $b' = b$.*

Let

$$\mathsf{Adv}_{\mathcal{A}^{\mathsf{IND}}}[\mathsf{PKE}, \ell] = 2 \times \Pr\left[\mathcal{A}_2(m_0, m_1, c^\star) = b \;\middle|\; \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\ell) \\ (m_0, m_1) \leftarrow \mathcal{A}_1(\mathsf{pk}) \\ b \leftarrow \{0,1\},\ c^\star = \mathsf{PKE.Enc}(\mathsf{pk}, m_b) \end{array}\right] - 1.$$

PKE *is said to have* indistinguishability of encryptions *if for any 2-stage* PPT *attacker* $\mathcal{A}^{\mathsf{IND}}$, $\mathsf{Adv}_{\mathcal{A}^{\mathsf{IND}}}[\mathsf{PKE}, \ell] \in \mathsf{negl}(\ell)$.

There is a fourth goal for an encryption scheme, called *non-malleability.* It was proposed in [DDN91], and roughly speaking implies that given a ciphertext of a plaintext, any adversary cannot construct another ciphertext whose plaintext is meaningfully related to the initial one. It is not usual to work with this notion when dealing with encryption schemes, and we do not formalize it here.

In the following we describe the most extended attack models that are currently considered for an asymmetric encryption scheme.

Definition 29 (Attack models) *The attack algorithm $\mathcal{A}$ runs in two stages: pre-challenge and post-challenge. Let the attacker have access to an oracle $\mathcal{O}_1$ up until the challenge is issued, and access to the oracle $\mathcal{O}_2$ after this time.*

1. *The attack is said to be a* chosen plaintext attack (CPA) *if the oracles are both trivial, i.e. $\mathcal{O}_1 = \mathcal{O}_2$ and both return* reject *for any input.*

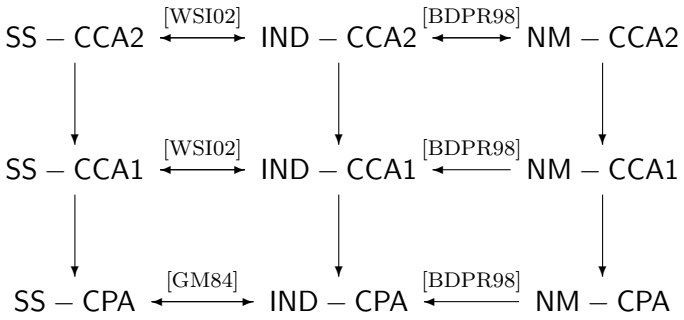
2. *The attack is said to be a* plaintext checking attack (PCA) *if both oracles, when queried with a pair (m, c), return 1 if c encrypts m and 0 otherwise.*

3. *The attack is said to be a* chosen ciphertext attack (CCA1) *or* lunchtime attack *if the oracle $\mathcal{O}_1$ decrypts messages (therefore $\mathcal{O}_1(c) = \mathsf{PKE.Dec}(\mathsf{sk}, c)$) but the oracle $\mathcal{O}_2$ is trivial.*

4. *The attack is said to be an* adaptive chosen ciphertext attack (CCA2) *if both oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ decrypt messages, with the exception that the oracle $\mathcal{O}_2$ returns* reject *if it is queried on the challenge ciphertext $c^\star$.*

When the oracles $\mathcal{O}_1$ and $\mathcal{O}_2$ are not trivial, they are referred to as decryption oracles. *Of course, if any oracle is queried with an invalid input, that is, outside the correct domain, then it is returned* reject.

The formalization of the first attack model was proposed in [GM84], the second in [NY90] and the last one in [RS92]. Combining the goals and the attack models CPA, CCA1, CCA2, we obtain nine security notions. The relations among them have been studied in several works, and those with greater impact are [GM84, BDPR98, WSI02, GLN02]. In the following diagram the implications among them are specified. For instance NM-CPA $\rightarrow$ IND-CPA means that any asymmetric encryption scheme that is NM-CPA is also IND-CPA. Non-trivial implications appear with reference to the work in which they were first shown.



Relations among security notions

The standard security notion for a general purpose public-key cryptosystem is IND-CCA2. The reason for this is that in [BDPR98] it was shown that NM-CCA2 and IND-CCA2 were equivalent security notions. In the works [WSI02, GLN02] it has been shown that this notion is in fact very strong, since it turns out to be equivalent to SS-CCA2, a claim that was implicitly assumed by many cryptographers without proof. Hereafter, IND-CCA2 is referred as IND-CCA as well as *semantic security against adaptive adversaries.*

1.3.2 Security notions for KEM-DEM hybrid encryption

In Section 1.2.2 we presented a model for hybrid encryption, built from the lower level primitives KEM and DEM. In the following, we describe some security requirements for the components in a KEM-DEM hybrid scheme that lead to proving it semantically secure against adaptive adversaries.

Definition 30 (KEM IND-CCA **security)** *An adversary against a key encapsulation mechanism* KEM *in an* adaptive chosen ciphertext attack *is a* PPT *algorithm* $\mathcal{A}^{\mathsf{KEM}}$ *that takes as input a security parameter* 1^ℓ*, and plays the following attack game:*

1. *The adversary queries a* key generation oracle*, which computes*

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KEM.KeyGen}(1^\ell)$$

   and returns pk*.*

2. *The adversary makes a sequence of calls to a* decryption oracle*, submitting encapsulations* C *of its choice, to which the decryption oracle responds with*

$$\mathsf{KEM.Dec}(\mathsf{sk}, C).$$

3. *The adversary queries an* encryption oracle*, which computes:*

$$(K_0, C^*) \leftarrow \mathsf{KEM.Enc}(\mathsf{pk}); \quad K_1 \leftarrow \{0,1\}^{KEM.KeyLen}; \quad b \leftarrow \{0,1\}$$

   and returns the pair (K_b, C^*)*.*

4. *The adversary issues new calls to the decryption oracle, subject only to the restriction that a submitted ciphertext* $C \neq C^*$*.*

5. *The adversary outputs* $b' \in \{0,1\}$*.*

For a PPT *adversary* $\mathcal{A}^{\mathsf{KEM}}$ *we define*

$$\mathsf{Adv}_{\mathcal{A}^{\mathsf{KEM}}}(\ell) := \left|\mathsf{Pr}\left[\mathcal{A}^{\mathsf{KEM}}(1^\ell) = 1 \,\middle|\, b = 0\right] - \mathsf{Pr}\left[\mathcal{A}^{\mathsf{KEM}}(1^\ell) = 1 \,\middle|\, b = 1\right]\right|.$$

We say that a KEM *is* secure against adaptive adversaries *if for all* $\mathcal{A}^{\mathsf{KEM}}$ *the function* $\mathsf{Adv}_{\mathcal{A}^{\mathsf{KEM}}}(\ell)$ *is negligible in* ℓ*.*

There are several transformations for building secure KEMs from well-known cryptographic primitives, as shown in [Den03], but these proofs usually use heuristic arguments. There are few practical KEMs proven secure without heuristic reasonings, and designing new ones is currently a challenging task. The following definitions deal with the symmetric components in an KEM-DEM scheme.

Definition 31 (SKE **passive security)** *An adversary against a symmetric scheme* SKE *in a* passive attack *is a* PPT *algorithm* $\mathcal{A}^{\mathsf{SKE}}$ *that takes as input a security parameter* 1^ℓ*, and plays the following attack game:*

1. *The algorithm $\mathcal{A}^{\mathsf{SKE}}$ generates two distinct messages m_0, m_1 with the same length. Next, it submits m_0, m_1 to an* encryption oracle.

2. *The encryption oracle generates a random key κ of length* $\mathsf{SKE.KeyLen}(1^\ell)$*, along with a random $b \in \{0,1\}$, and encrypts the message m_b using the key κ. The adversary is given the resulting ciphertext c^*.*

3. *The adversary outputs $b' \in \{0,1\}$.*

For a PPT *adversary $\mathcal{A}^{\mathsf{SKE}}$ we define*

$$\mathsf{Adv}_{\mathcal{A}^{\mathsf{SKE}}}(\ell) := \left|\mathsf{Pr}\left[\mathcal{A}^{\mathsf{SKE}}(1^\ell) = 1 \,\middle|\, b = 0\right] - \mathsf{Pr}\left[\mathcal{A}^{\mathsf{SKE}}(1^\ell) = 1 \,\middle|\, b = 1\right]\right|.$$

We say that a SKE *is* secure against passive adversaries *if for all $\mathcal{A}^{\mathsf{SKE}}$ the function $\mathsf{Adv}_{\mathcal{A}^{\mathsf{SKE}}}(\ell)$ is negligible in ℓ.*

Definition 32 (MA **security)** *A* one-message attack *adversary against a message authentication code* MA *is a* PPT *algorithm $\mathcal{A}^{\mathsf{MA}}$ that takes as input a security parameter 1^ℓ, and plays the following attack game:*

1. *The algorithm $\mathcal{A}^{\mathsf{MA}}$ chooses a bit string T, and submits it to an* authentication oracle.

2. *This oracle generates a random key κ' of length $MA.KeyLen$, computes*

$$\mathsf{MA.Eval}(\kappa', T)$$

   and returns the corresponding MAC value to the adversary.

3. *The adversary outputs a list $((T_1, MAC_1), \ldots, (T_k, MAC_k))$ of pairs of bit strings.*

$\mathcal{A}^{\mathsf{MA}}$ has produced a forgery *if for some $1 \leq i \leq k$, we have $MAC_i \neq MAC$ and $\mathsf{MA.Eval}(\kappa', T_i) = MAC_i$. For a* PPT *adversary $\mathcal{A}^{\mathsf{MA}}$ we define $\mathsf{Adv}_{\mathcal{A}^{\mathsf{MA}}}(\ell)$ as the probability that $\mathcal{A}^{\mathsf{MA}}$ produces a forgery in the above game. We say that a* MA *is* secure against one-message attacks *if for all $\mathcal{A}^{\mathsf{MA}}$ the function $\mathsf{Adv}_{\mathcal{A}^{\mathsf{MA}}}(\ell)$ is negligible in ℓ.*

Definition 33 (DEM **adaptive security)** *An adversary against a data encapsulation mechanism* DEM *in an* adaptive attack *is a* PPT *algorithm $\mathcal{A}^{\mathsf{DEM}}$ that takes as input a security parameter 1^ℓ, and plays the following attack game:*

1. *The algorithm $\mathcal{A}^{\mathsf{DEM}}$ generates two distinct messages m_0, m_1 with the same length. Next, it submits m_0, m_1 to an* encryption oracle.

2. *The encryption oracle generates a random key K of length* $\mathsf{SKE.KeyLen}(1^\ell)$, *along with a random $b \in \{0,1\}$, and encrypts the message m_b using the key K. The adversary is given the resulting ciphertext c^*.*

3. *The adversary makes a sequence of calls to a* decryption oracle, *submitting ciphertexts c of its choice, to which the decryption oracle responds with* $\mathsf{DEM.Dec}(K, c)$.

4. *The adversary outputs $b' \in \{0,1\}$.*

For a PPT *adversary* $\mathcal{A}^{\mathsf{DEM}}$ *we define*

$$\mathsf{Adv}_{\mathcal{A}^{\mathsf{DEM}}}(\ell) := \left|\mathsf{Pr}\left[\mathcal{A}^{\mathsf{DEM}}(1^\ell) = 1 \,\middle|\, b = 0\right] - \mathsf{Pr}\left[\mathcal{A}^{\mathsf{DEM}}(1^\ell) = 1 \,\middle|\, b = 1\right]\right|.$$

We say that a DEM *is* secure against adaptive adversaries *if for all* $\mathcal{A}^{\mathsf{DEM}}$ *the function* $\mathsf{Adv}_{\mathcal{A}^{\mathsf{DEM}}}(\ell)$ *is negligible in* ℓ.

Theorem 34 *Let* SKE *be secure against passive attacks and* MA *be secure against one-message attacks. Then* DEM2 *is a data encapsulation mechanism secure against adaptive adversaries.*

Proof: See Theorem 4 in [CS]. ∎

In contrast to KEMs, the problem of designing secure DEMs is solved by the latter theorem, since there are known ways to efficiently build secure SKEs and MAs using well-known cryptographic techniques (cf. [CS, Sho04]).

Theorem 35 *Let* SKE *be secure against passive attacks,* MA *be secure against one-message attacks and* KEM *be secure against adaptive adversaries. Let* $\mathsf{DEM2} = (\mathsf{SKE}, \mathsf{MA})$ *be the data encapsution mechanism obtained from Table 1.1. Then* $\mathsf{HE} = (\mathsf{KEM}, \mathsf{DEM2})$ *is semantically secure against adaptive adversaries.*

Proof: See Theorem 5 in [CS]. ∎

With this result, building secure hybrid encryption schemes is reduced to designing secure key encapsulation mechanisms.

1.3.3 Beyond the standard model

Random Oracle Model

The tools to design and analyze encryption schemes presented so far are known as the *standard model* for provable security. As explained in the introduction, designing and proving secure practical asymmetric cryptosystems, as well as key encapsulation mechanisms, has been shown to be quite a difficult task. With the exception of

the works [CS98, CS02] by Cramer and Shoup, there has been little progress in this area. For this reason, the idealised model of computation called *Random Oracle Model* (ROM) was proposed by Bellare and Rogaway in [BR93], in which giving security proofs is far easier than in the standard model. This model is discussed in the sequel.

A *random oracle* can be viewed as a special type of random process or random sequence. The random oracle is defined through an idealised functionality that is closely related to the random oracle simulations often used in the proofs of security. The following definition states what it is meant in this work by a random oracle.

Definition 36 (Random oracle) *Let A be a samplable 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 $\{0,1\}^\star$ can be viewed as an ordered set. A* random oracle *over A is an oracle that answers queries exactly as if the random function G is 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 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 is then stored in $\mathcal{T}_G$. Schematically,

$G(s)$
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

Then, in a security proof in the ROM, the adversary is given oracle access to one or more random functions. This means it is given a *black-box* acces, i.e. it does not evaluate the random function directly. In order to formalize internal random coins of the random functions involved, a step $G \leftarrow \mathcal{R}(A)$ must be added for each random function at the beginning of the attack game.

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

[CGH98, GT03]) 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 [BBP04].

Side channel attacks

In complexity-theoretic cryptography, an adversary may attack a cryptographic algorithm essentially only by exchanging messages with it. The adversary is given some access to explicit inputs and/or outputs of the algorithm and is even given knowledge of its internal code, except for the secret key. However, in the real world, computations are physical processes, so an adversary may exploit the information leakage inherent to the physical execution of an algorithm. Such attacks are commonly called *side-channel attacks*. Since side-channel attacks are outside complexity-theoretic cryptography, its security tools do not protect against such attacks. In fact, there have been found realistic powerful attacks against widely used complexity-theoretic cryptographic schemes (eg. [Koc96, KJJ99]). The development of a theoretical framework including security against physical attacks is currently a challenging task in the cryptographic community, and initial steps have been taken in some works, such as [MR04, GLM+04]. In this work we only make an indirect use of these ideas.

1.4 Trusted mathematical assumptions and concrete security

In complexity-theoretic cryptography protocols, security is in the last instance based on the existence of some problems that are conjectured to be unsolvable in probabilistic polynomial time. They are the *atomic primitives* in designing any protocol. Some of these primitives are derived from mathematical objects, some others arise from the cryptographic practice. For instance, the hardness of factoring belongs to the former category, while assuming that using a block cipher with a random seed produces a pseudorandom sequence belongs to the latter.

In the sequel we will focus on number theoretic hard problems, all of them well-known and recently proposed. It is also discussed why they are assumed to be hard, showing the best algorithms known to attack them. This leads to present the study of concrete security, which is a crucial step when evaluating the security of any cryptographic protocol used in practice. This security estimation enables us to relate the expected time for solving the underlying problems and the key size in the scheme.

The hardness assumptions dealt with are divided into three groups, namely, *computational*, *decisional* and *gap* assumptions. Roughly speaking, a computational assumption is related to the hardness of computing a solution of a problem. A decisional problem is related to the hardness of deciding if an instance of a problem has solution, but it is not required to find any solution. In gap problems, one tries to compute a solution of a problem with the help of an oracle that solves the related decisional problem. In every case, the concept of a *keypair generator* (see Definition 14) will play a fundamental role in the description of these problems. Regarding its origin, the mathematical assumptions used in this work can be divided into two different families: *factoring* and *discrete logarithm* based assumptions. In the following, we assume the reader to be familiar with arithmetic in rings and finite fields.

1.4.1 Factoring based assumptions

The schemes based on factoring use arithmetic in $\mathbb{Z}_n$, where n is some composite number. In almost all cases the ability to factor the number n, also called *modulus*, implies the ability to solve the hard problem. The description of a factoring-based assumption strongly depends on the probability distribution induced over $PK \times SK$ by the keypair generator. Let us first define the key sets for most of the factoring-based primitives studied in this work. We shall define two keypair sets, the first one, written as $PK^{\mathsf{FAC}} \times SK^{\mathsf{FAC}}$, will be used for *factoring-like* schemes, and the second one, written as $PK^{\mathsf{RSA}} \times SK^{\mathsf{RSA}}$, for *RSA-like* schemes. Although most of the objects we will define are treated as mathematical entities, they must be representable as bit strings to fit in our definition of PPT algorithms. The representations proposed in [IEE99] are valid for our purposes.

Given a security parameter 1^ℓ, we denote by $\mathcal{PRIMES}(\ell)$ the set of primes with length ℓ in its binary representation, and $\lambda(n) = \mathrm{lcm}(p-1, q-1)$ when $n = pq$. Such integers are usually called RSA modulus. Then, on the one hand,

$$PK_\ell^{\mathsf{FAC}} = \{n \mid n = pq;\ p, q \in \mathcal{PRIMES}(\ell)\} \quad \text{and}$$
$$SK_\ell^{\mathsf{FAC}} = \{(n, p, q) \mid p, q \in \mathcal{PRIMES}(\ell),\ n = pq\};$$

while on the other hand,

$$PK_\ell^{\mathsf{RSA}} = \{(n, e) \mid n = pq;\ p, q \in \mathcal{PRIMES}(\ell),\ \gcd(e, \lambda(n)) = 1\}$$

and

$$SK_\ell^{\mathsf{RSA}} = \left\{ (n, e, p, q, d) \;\middle|\; \begin{array}{c} p, q \in \mathcal{PRIMES}(\ell),\ n = pq, \\ \gcd(e, \lambda(n)) = 1,\ ed \equiv 1 \bmod \lambda(n) \end{array} \right\}.$$

Before describing the usual distributions over these sets, we first note that a distribution over a factoring keypair set can be transformed into a distribution over a RSA keypair

set, and vice versa. In the first direction, we should choose a suitable e or d once the values (n, p, q) are given; in the second direction, we simply take the values (n, p, q) from the tuple (n, e, p, q, d). In the following, some distributions over these two keypair sets are described, which will be denoted by I^{FAC} when they are factoring-like, and by I^{RSA} in the RSA case. We simplify the notation by writing only the output over SK, since by definition there exists a PT algorithm that retrieves pk from sk. Moreover, in this case this algorithm is completely trivial.

Definition 37 (Factoring-like distributions) *The distributions $I^{\mathsf{FAC}}_{\mathsf{clas}}, I^{\mathsf{FAC}}_{\mathsf{Blum}}, I^{\mathsf{FAC}}_{\mathsf{strong}}$ over $PK^{\mathsf{FAC}} \times SK^{\mathsf{FAC}}$, named as* classical, Blum *and* strong-primes factoring distributions *respectively, are defined as follows:*

- $I^{\mathsf{FAC}}_{\mathsf{clas}}$ *on input a security parameter 1^ℓ outputs (n, p, q) s.t.*

$$p, q \leftarrow PRIMES(\ell/2),\ n = pq.$$

- $I^{\mathsf{FAC}}_{\mathsf{Blum}}$ *on input a security parameter 1^ℓ outputs (n, p, q) s.t.*

$$p, q \leftarrow PRIMES(\ell/2),\ where\ p, q \equiv 3 \bmod 4,\ n = pq.$$

- $I^{\mathsf{FAC}}_{\mathsf{strong}}$ *on input a security parameter 1^ℓ outputs (n, p, q) s.t.*

$$p, q \leftarrow PRIMES(\ell/2) \text{ where } p, q \text{ are strong primes, i.e.}$$
$$p = 2p' + 1 \text{ and } q = 2q' + 1 \text{ with } p', q' \text{ prime numbers and } n = pq.$$

There are several ways to implement algorithms with output close to the distributions defined above. We refer to Annex A in [IEE99] for examples of standardized concrete implementations.

Definition 38 (RSA-like distributions) *The distributions $I^{\mathsf{RSA}}_{\mathsf{clas}}$, $I^{\mathsf{RSA}}_{\mathsf{prac}}$ and $I^{\mathsf{RSA}}_{\mathsf{Blum}}$ over $PK^{\mathsf{RSA}} \times SK^{\mathsf{RSA}}$, named as* classical, practical *and* Blum RSA distributions *respectively, are defined as follows:*

- $I^{\mathsf{RSA}}_{\mathsf{clas}}$ *on input a security parameter 1^ℓ outputs (n, p, q, e, d) s.t.*

$$p, q \leftarrow PRIMES(\ell/2),\ e \leftarrow \mathbb{Z}_{\lambda(n)},\ ed \equiv 1 \bmod \lambda(n).$$

- $I^{\mathsf{RSA}}_{\mathsf{prac}}$ *on input a security parameter 1^ℓ outputs (n, p, q, e, d) s.t. e is chosen from a given set of 'small' prime integers, usually Fermat primes $\{3, 5, 7, 17, 257, 65537\}$. Then $p, q \leftarrow PRIMES(\ell/2)$ s.t. $\gcd(e, p-1) = \gcd(e, q-1) = 1$, $n = pq$ and $ed \equiv 1 \bmod \lambda(n)$.*

- $I^{\mathsf{RSA}}_{\mathsf{Blum}}$ *has the same output than* $I^{\mathsf{RSA}}_{\mathsf{prac}}$ *except that* $p \equiv q \equiv 3 \bmod 4$.

We point out that cryptographers using the *classical RSA distribution* in proving the security of RSA-like schemes is a common situation, but they are actually thinking of the *practical RSA distribution* for the implementation step. Therefore, if their schemes are used in practice, the security is in fact related to the practical version, although unstated. From a theoretical point of view, $I^{\mathsf{RSA}}_{\mathsf{clas}}$ is preferable to $I^{\mathsf{RSA}}_{\mathsf{prac}}$. In the first case, the parameters p, q and e, d are chosen almost independently, while in the second case the primes are restricted to those p, q such that the public exponent e is coprime to $p-1$ and $q-1$, and then some randomness is lost with respect to the first option. In contrast, $I^{\mathsf{RSA}}_{\mathsf{prac}}$ is better for practical purposes, since using small size exponents leads to very efficient encryptions. In fact, the RSA keypair generation algorithm proposed in the standard [IEE99] uses the distribution $I^{\mathsf{RSA}}_{\mathsf{prac}}$. Both distributions will be considered hereafter, but preference will be given to $I^{\mathsf{RSA}}_{\mathsf{prac}}$ because of its significance in real life implementations. Consequently, our approach is different from the usual expressions found in the literature.

Computational factoring-based assumptions

Informally speaking, the general formulation of a computational factoring assumption states that it is hard to factor the modulus, or to solve the RSA problem, induced by a certain distribution over the corresponding keypair sets.

Assumption 39 (Factoring assumption) *For every* PPT *algorithm* $\mathcal{A}$

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n) = (p, q) \mid (n, p, q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{clas}}(1^\ell)\right] \in \mathsf{negl}(\ell)\,.$$

where the probability is computed with respect to distribution $I^{\mathsf{FAC}}_{\mathsf{clas}}$ *and the coin tosses of* $\mathcal{A}$.

Assumption 40 (Factoring Blum-RSA assumption) *For every* PPT *algorithm* $\mathcal{A}$

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n) = (p, q) \mid (n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{Blum}}(1^\ell)\right] \in \mathsf{negl}(\ell)\,.$$

where the probability is computed with respect to distribution $I^{\mathsf{RSA}}_{\mathsf{Blum}}$ *and the coin tosses of* $\mathcal{A}$.

Assumption 41 (RSA assumption) *For every* PPT *algorithm* $\mathcal{A}$

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n, e, y) = x \;\middle|\; \begin{array}{c}(n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{clas}}(1^\ell) \\ x \leftarrow \mathbb{Z}^*_n,\ y = x^e\end{array}\right] \in \mathsf{negl}(\ell)\,,$$

where the probability is computed with respect to distribution $I^{\mathsf{RSA}}_{\mathsf{clas}}$ *and the coin tosses of* $\mathcal{A}$.

Assumption 42 (Practical RSA assumption) *For every* PPT *algorithm* $\mathcal{A}$

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n, e, y) = x \;\middle|\; \begin{array}{c}(n,p,q,e,d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{prac}}(1^\ell) \\ x \leftarrow \mathbb{Z}^*_n,\ y = x^e\end{array}\right] \in \mathsf{negl}(\ell)\,.$$

where the probability is computed with respect to distribution $I^{\mathsf{RSA}}_{\mathsf{prac}}$ *and the coin tosses of* $\mathcal{A}$*.*

Remark 43 *In the same way, one would define the* Blum and strong *factoring assumptions using the distributions* $I^{\mathsf{FAC}}_{\mathsf{Blum}}$ *and* $I^{\mathsf{FAC}}_{\mathsf{strong}}$ *respectively.*

Decisional factoring-based assumptions

The following assumptions are described by means of a certain keypair distribution, but it is possible to use any of the factoring-based distributions defined before. We have chosen the most useful distributions in the rest of this work.

Assumption 44 (QR assumption) *The probability distributions* $D_{1,n}, D_{2,n}$ *induced by the following random variables* X_1, X_2 *over* $\mathbb{Z}^*_n$ *are polynomially indistinguishable:*

$$\begin{aligned} X_1 &= (n, y) \quad \text{where} \quad (n,p,q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{clas}}(1^\ell),\ x \leftarrow \mathbb{Z}^*_n,\ y = x^2 \bmod n\,, \\ X_2 &= (n, y) \quad \text{where} \quad (n,p,q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{clas}}(1^\ell),\ y \leftarrow \mathbb{Z}^*_n \text{ s.t. } \left(\frac{y}{n}\right) = 1. \end{aligned}$$

where $\left(\frac{\cdot}{n}\right)$ *is the Jacobi symbol. This assumption is called* Quadratic Residuosity (QR) *assumption.*

Assumption 45 (BQR assumption) *The probability distributions* $D_{1,n}, D_{2,n}$ *induced by the following random variables* X_1, X_2 *over* $\mathbb{Z}^*_n$ *are polynomially indistinguishable:*

$$\begin{aligned} X_1 &= (n, y) \quad \text{where} \quad (n,p,q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{Blum}}(1^\ell),\ x \leftarrow \mathbb{Z}^*_n,\ y = x^2 \bmod n\,, \\ X_2 &= (n, y) \quad \text{where} \quad (n,p,q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{Blum}}(1^\ell),\ y \leftarrow \mathbb{Z}^*_n \text{ s.t. } \left(\frac{y}{n}\right) = 1. \end{aligned}$$

where $\left(\frac{\cdot}{n}\right)$ *is the Jacobi symbol. We call this assumption* Blum Quadratic Residuosity (BQR) *assumption.*

Assumption 46 (DSeR assumption) *The probability distributions* $D_{1,n}, D_{2,n}$ *induced by the following random variables* X_1, X_2 *over* $\mathbb{Z}^*_{n^2}$ *are polynomially indistinguishable:*

$$\begin{aligned} X_1 &= (n, e, y) \quad \text{where} \quad (n,p,q,e,d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{prac}}(1^\ell),\ x \leftarrow \mathbb{Z}^*_n,\ y = x^e \bmod n^2\,, \\ X_2 &= (n, e, y) \quad \text{where} \quad (n,p,q,e,d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{prac}}(1^\ell),\ y \leftarrow \mathbb{Z}^*_{n^2}. \end{aligned}$$

This assumption is called Decisional Small e-Residues (DSeR) *assumption, and it was first introduced in* [CGHN01].

Assumption 47 (DS2eR assumption) *The probability distributions $D_{1,n}, D_{2,n}$ induced by the following random variables X_1, X_2 over Q_{n^2} are polynomially indistinguishable:*

$$X_1 = (n, e, y) \quad \text{where} \quad (n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{BlumPrac}}(1^\ell),\ x \leftarrow Q_n,\ y = x^{2e} \bmod n^2\,,$$
$$X_1 = (n, e, y) \quad \text{where} \quad (n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{BlumPrac}}(1^\ell),\ y \leftarrow Q_{n^2}\,,$$

where Q_m denotes the set of quadratic residues modulo an integer m. This assumption is called Decisional Small $2e$-Residues (DS$2e$R) *assumption, and it was first introduced in* [GMMV02].

Gap factoring-based assumptions

Roughly speaking, the Gap-Rabin assumption states that no PPT algorithm can solve the factoring problem, even with access to an oracle that solves the QR problem.

Assumption 48 (Gap-Rabin assumption) *For every* PPT *algorithm $\mathcal{A}$*

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n) = (p, q) \mid (n, p, q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{clas}}(1^\ell)\right] \in \mathsf{negl}(\ell)\,.$$

even with access to an oracle that deterministically solves the QR *problem, where the probability is computed with respect to distribution $I^{\mathsf{FAC}}_{\mathsf{clas}}$ and the coin tosses of $\mathcal{A}$.*

1.4.2 Discrete logarithm based assumptions

Problems related to the discrete logarithm are usually phrased in terms of a cyclic group $G = \langle g \rangle$ where g has prime order p. Usually, these groups are obtained by taking a large enough prime-order subgroup of $\mathbb{F}_q^*$, or a prime-order subgroup of an elliptic curve defined over $\mathbb{F}_q$. Technically, since all groups of prime order are isomorphic, the hardness of the problem depends not upon the group itself but upon the representation of the group. For example, the discrete logarithm problem is thought to be hard in elliptic curve subgroups, but is definitely easy in the group of integers modulo p.

To describe the keypair sets of discrete logarithm problems we will use the notion of *group scheme*. A group scheme is a set $\mathbf{G} = \{\mathbf{G}_\ell\}$ of *group descriptions*. A group description $(\mathcal{G}, G, g, p) \in \mathbf{G}_\ell$ specifies a finite abelian group $\mathcal{G}$, along with a prime-order subgroup G, a generator g of G and the order p of G, and there exists a polynomial $t \in \mathsf{poly}(\ell)$ such that both the length of the strings representing the elements in $\mathcal{G}$ and the bit length of p are bounded by $t(\ell)$. Then, keypair sets for discrete logarithm schemes in their most basic and usual version are

$$PK^{\mathbf{G}}_\ell = \{(\mathcal{G}, G, g, p, h) \mid (\mathcal{G}, G, g, p) \in \mathbf{G}_\ell,\ h \in G\} \quad \text{and}$$
$$SK^{\mathbf{G}}_\ell = \{(\mathcal{G}, G, g, p, h, u) \mid (\mathcal{G}, G, g, p) \in \mathbf{G}_\ell,\ h \in G,\ u \in \mathbb{Z}_p^* \text{ s.t. } h = g^u\}.$$

More generally, the elements in $PK_\ell^{\mathbf{G}}$ are of the form $(\mathcal{G}, G, g, p, h_1, \ldots, h_l)$, where $h_1, \ldots, h_l \in G$; while the elements in $SK_\ell^{\mathbf{G}}$ have the form

$$(\mathcal{G}, G, g, p, h_1, \ldots, h_l, u_1, \ldots, u_l),$$

where $u_1, \ldots, u_l \in \mathbb{Z}_p^*$ and $u_i = \log_g h_i$. However, intractability assumptions will be stated in terms of the basic version. We will present in the sequel two examples of cryptographic group schemes: finite fields and elliptic curves. After that, some usual distributions over these sets are described. For notational purposes, $|G|$ denotes the cardinality of a group G.

Discrete logarithm groups and distributions

Definition 49 (Finite field groups) *A* finite field group scheme $\mathbf{FF} = \{\mathbf{FF}_\ell\}$ *consists of the following group descriptions:*

- $\mathcal{G}$ *is a finite field* $\mathbb{F}_q$ *with* $|q| = \ell$.
- G *is a cyclic subgroup of* $\mathcal{G}$ *with prime order* p.
- g *is a generator of* G.
- $u \in \mathbb{Z}_p^*$ *and* $h = g^u \in G$.

Before describing elliptic curve group schemes, we recall some basic facts about elliptic curves that must be known in order to understand the rest of this section. There are nice introductions to elliptic curves from a cryptographic point of view, for instance [Men93, BSS99], while a standard reference with a mathematical treatment is [Sil86].

Assume that $\mathbb{F}_q$ has characteristic greater than 3. An *elliptic curve* E over $\mathbb{F}_q$ is defined by an equation $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_q$ and $4a^3 + 27b \neq 0$. For finite fields with characteristic 2 or 3, the equation defining an elliptic curve takes different forms [Men93]. The *group of points* of an elliptic curve E is the set of all solutions $(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$, where $\overline{\mathbb{F}}_q$ is the algebraic closure of $\mathbb{F}_q$, together with a special point $\mathcal{O}$ called *the point at infinity*. E is an abelian group with the point $\mathcal{O}$ acting as its identity element, and it is often written with additive notation. The addition formulas for characteristic greater than 3 are as follows: let $P = (x, y) \in E$, then $-P = (x, -y)$. If $Q = (x_2, y_2) \in E$, $Q \neq -P$, then $P + Q = (x_3, y_3)$, where

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = \lambda(x_1 - x_3) - y_1,$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if} P \neq Q \\ \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if} P = Q. \end{cases}$$

The group of $\mathbb{F}_q$ *rational points* of E, denoted by $E(\mathbb{F}_q)$, consists of the points in E having both coordinates in $\mathbb{F}_q$ plus the point $\mathcal{O}$, and it is also an abelian group. A well-known theorem of Hasse states that the cardinality of $E(\mathbb{F}_q)$ is $|E(\mathbb{F}_q)| = q+1-t$, where $-2\sqrt{q} \leq t \leq 2\sqrt{q}$. The curve E is said to be *supersingular* if $t^2 = 0, q, 2q, 3q, 4q$; otherwise the curve is *non-supersingular*. A result by Waterhouse [Wat69] states that if q is a prime, then for each t satisfying $-2\sqrt{q} \leq t \leq 2\sqrt{q}$ there exists at least one elliptic curve E over $\mathbb{F}_q$ with $|E(\mathbb{F}_q)| = q + 1 - t$. There exists a similar result when q is a power of 2. We can now state the definition of an elliptic curve group scheme.

Definition 50 (Elliptic curve groups) *An* elliptic curve group scheme $\mathbf{EC} = \{\mathbf{EC}_\ell\}$ *consists of the following group descriptions:*

- *$\mathcal{G}$ is the group of $\mathbb{F}_q$ rational points of an elliptic curve E defined over $\mathbb{F}_q$.*
- *G is a cyclic subroup of $\mathcal{G}$ with prime order p such that $|p| = \ell$.*
- *P is a generator of G.*
- *$u \in \mathbb{Z}_p^*$ and $Q = uP \in G$.*

We point out a difference in the definition of finite field and elliptic curve group descriptions: in the former the security parameter sets the binary length representation of the elements in the group $\mathcal{G}$, while in the latter ℓ sets the bits length of the cardinality of the subgroup G. This difference will be explained when we study the computational effort to solve discrete logarithms in each group.

Regarding the finite field case, we consider two possible distributions: in the first place, a distribution that samples finite fields $\mathbb{F}_q$ with q of any form. In the second place, a distribution only considering $\mathbb{F}_q$ for strong primes q. Although we do not use these distributions in this work, we have decided to put them here for the sake of completeness. As in the factoring case, only the output over $SK^{\mathbf{FF}}$ is written.

Definition 51 (Finite field distributions) *The distributions $I^{\mathbf{FF}}_{\mathsf{clas}}$ and $I^{\mathbf{FF}}_{\mathsf{strong}}$ over $PK^{\mathbf{FF}} \times SK^{\mathbf{FF}}$, named as* classical *and* strong-primes finite field distributions*, respectively, are defined as follows:*

- *$I^{\mathbf{FF}}_{\mathsf{clas}}$ on input a security parameter 1^ℓ outputs $(\mathbb{F}_q, G, g, p, h, u)$ where $p, q \leftarrow PRIMES(\ell)$ s.t. $q \equiv 1 \bmod p$, $g \leftarrow \mathbb{F}_q$ generates a p order subgroup G, $u \leftarrow \mathbb{Z}_p^*$ and $h = g^u \in G$.*
- *$I^{\mathbf{FF}}_{\mathsf{strong}}$ on input a security parameter 1^ℓ outputs $(\mathbb{F}_q, G, g, p, h, u)$ such that $q \leftarrow PRIMES(\ell)$ is a strong prime, i.e. $q = 2p + 1$, G is the set of quadratic residues modulo q, $g \leftarrow G$ is a generator of G, $u \leftarrow \mathbb{Z}_p^*$ and $h = g^u \in G$.*

The algorithms described in Annexes A.16.1 and A.16.3 in [IEE99] have outputs that fit into the distribution $I^{\mathbf{FF}}_{\mathsf{clas}}$.

With respect to elliptic curve group distributions, there are a number of ways to generate elliptic curve group descriptions. These include:

- Selecting an appropriate finite field $\mathbb{F}_q$. Then randomly select an elliptic curve over the field (that is, randomly chosing the coeficients in its defining equation), count the number of points on the curve using Schoof's algorithm [Sch95], check whether the number of points p is nearly prime with $|p| \geq \ell$, and repeat until appropriate parameters are found.
- Selecting an appropriate field $\mathbb{F}_q$, then select an appropriate group order p, and generate a curve over the field with this number of points using techniques based on *complex multiplication* [AM93, LZ94].
- Selecting an elliptic curve from a special family of curves holding special properties (whose order is easily computable, with bilinear maps, etc.).

The first method, known as *randomly generation of elliptic curve domain parameters*, is the most conservative choice because it offers a probabilistic guarantee against future attacks exploiting special properties. However, existing implementations of the point counting algorithm by Schoof are less efficient than implementations of the other parameter selection methods. The second method, known as *complex multiplication generation of elliptic curve domain parameters*, generates parameters more efficiently than the first method. The third method, known as *generating elliptic curve domain parameters from a special family*, generates elliptic curve groups with particular properties, which may be efficient generation and efficient group operations, as in the Standards for Efficient Cryptography [Cer00a], or the existence of additional mathematical objects, such as the Weil and Tate pairings for cryptographic purposes [Jou02]. However, despite their efficiency or special properties benefits, the second and third methods should be used with some caution because they produce parameters which may be susceptible to future special-purpose attacks. In the next definition we introduce distributions associated with the first and second generation methods. As usual, only the output over $SK^{\mathbf{EC}}$ is written.

Definition 52 (Elliptic curve distributions) *The distributions $I^{\mathbf{EC}}_{\mathsf{rand}}$ and $I^{\mathbf{EC}}_{\mathsf{CM}}$ over $PK^{\mathbf{EC}} \times SK^{\mathbf{EC}}$, named as* random *and* complex multiplication elliptic curve distributions, *respectively, are defined as follows:*

- *$I^{\mathbf{EC}}_{\mathsf{rand}}$ on input a security parameter 1^ℓ outputs (E, G, P, p, Q, u) where E is an elliptic curve over $\mathbb{F}_q$, p is a prime s.t. $|p| \geq \ell$, $P \leftarrow E(\mathbb{F}_q)$ generates the p order subgroup G, $u \leftarrow \mathbb{Z}^*_p$ and $Q = uP \in G$. These parameters are obtained by the random generation method.*

- $I_{\mathsf{CM}}^{\mathbf{EC}}$ *on input a security parameter* 1^ℓ *outputs* (E, G, P, p, Q, u) *where* E *is an elliptic curve over* $\mathbb{F}_q$*,* p *is a prime s.t.* $|p| \geq \ell$*,* $P \leftarrow E(\mathbb{F}_q)$ *generates the* p *order subgroup* G*,* $u \leftarrow \mathbb{Z}_p^*$ *and* $Q = uP \in G$*. These parameters are obtained by the complex multiplication generation method.*

Standardized algorithms for distribution $I_{\mathsf{rand}}^{\mathbf{EC}}$ can be found in [X9.99, X9.01], while for $I_{\mathsf{CM}}^{\mathbf{EC}}$ can be found in [IEE99]. Concrete elliptic curve group descriptions with distribution $I_{\mathsf{rand}}^{\mathbf{EC}}$ and with Koblitz curves [Kob92] are suggested in [Cer00b].

Computational discrete logarithm based assumptions

Informally speaking, the general formulation of a computational discrete logarithm assumption states that it is hard to solve the discrete logarithm or the Computational Diffie-Hellman problem, induced by a certain distribution over finite field or elliptic curve keypair sets. In the subsequent definitions, all probabilities are computed with respect to a distribution $I^{\mathbf{G}}$ and the coin tosses of a PPT algorithm $\mathcal{A}$.

Assumption 53 (DL assumptions) *For every* PPT *algorithm* $\mathcal{A}$

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, \mathcal{G}, G, g, p, h) = u \mid (\mathcal{G}, G, g, p, h, u) \leftarrow I^{\mathbf{G}}(1^\ell)\right] \in \mathsf{negl}(\ell)\,.$$

It is called Discrete Logarithm (DL) *assumption if* $\mathbf{G} = \mathbf{FF}$*, and* Elliptic Curve Discrete Logarithm (ECDL) *assumption if* $\mathbf{G} = \mathbf{EC}$.

Assumption 54 (CDH assumptions) *For every* PPT *algorithm* $\mathcal{A}$

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, \mathcal{G}, G, g, p, g^u, g^v) = g^{uv} \mid (\mathcal{G}, G, g, p, h, u) \leftarrow I^{\mathbf{G}}(1^\ell),\ v \leftarrow \mathbb{Z}_p^*\right] \in \mathsf{negl}(\ell)\,.$$

It is called Computational Diffie-Hellman (CDH) *assumption if* $\mathbf{G} = \mathbf{FF}$*, and* Elliptic Curve Computational Diffie Hellman (ECDH) *assumption if* $\mathbf{G} = \mathbf{EC}$.

Decisional discrete logarithm based assumptions

Assumption 55 (DDH assumptions) *The probability distributions* $D_{1,\ell}, D_{2,\ell}$ *induced by the following random variables* X_1, X_2 *are polynomially indistinguishable:*

$$\begin{aligned} X_1 &= (\mathcal{G}, G, g, p, g^u, g^v, g^{uv}) \text{ where } (\mathcal{G}, G, g, p, h, u) \leftarrow I^{\mathbf{G}}(1^\ell),\ v \leftarrow \mathbb{Z}_p^*\,, \\ X_2 &= (\mathcal{G}, G, g, p, g^u, g^v, g^w) \text{ where } (\mathcal{G}, G, g, p, h, u) \leftarrow I^{\mathbf{G}}(1^\ell),\ v, w \leftarrow \mathbb{Z}_p^*\,, \end{aligned}$$

This assumption is called Decisional Diffie-Hellman (DDH) *assumption if* $\mathbf{G} = \mathbf{FF}$*, and* Elliptic Curve Decisional Diffie Hellman (ECDDH) *assumption if* $\mathbf{G} = \mathbf{EC}$.

Gap assumptions

Assumption 56 (Gap-CDH assumption) *For every* PPT *algorithm* $\mathcal{A}$

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, \mathcal{G}, G, g, p, g^u, g^v) = g^{uv} \mid (\mathcal{G}, G, g, p, h, u) \leftarrow I^{\mathbf{G}}(1^\ell),\ v \leftarrow \mathbb{Z}_p^*\right] \in \mathsf{negl}(\ell)\,.$$

even with access to an oracle that deterministically solves the DDH *problem in* $\mathbf{G}$*. This assumption is called* Gap-CDH *assumption if* $\mathbf{G} = \mathbf{FF}$*, and* Gap-ECDH *assumption if* $\mathbf{G} = \mathbf{EC}$.

1.4.3 Hardness of factoring and discrete logarithm problems

The problems on which the security of our schemes is based have been already formalized. They are *conjectured* to be unsolvable by PPT algorithms, and it is not known how to prove that these problems are actually hard. Indeed, this issue is related to one of the most famous open problems in complexity theory and mathematics, namely if P $\neq$ NP (cf. [Gol01]). The best we can do is to estimate the hardness of these problems by studying the computational complexity of the fastest attacks known against them.

Attacks against factoring and RSA problems

The fastest factoring algorithm published today is the General Number Field Sieve (NFS), a variation of the original algorithm invented by John Pollard in 1988. NFS can handle numbers of arbitrary form, including RSA moduli. On heuristic grounds the NFS can be expected to require time proportional to

$$L[n] := e^{(1.9229+o(1))\ln^{1/3} n \ln^{2/3}(\ln n)}$$

to factor an RSA modulus n, where the $o(1)$ term goes to zero as n goes to infinity. This run time is called *subexponential* in the input size n because as n goes to infinity it is less than n^c for any constant $c > 0$. The storage requirements of the NFS are proportional to $\sqrt{L[n]}$.

With respect to solving RSA, the best known way to attack it, except for some particular cases (see [Bon99]), is to factor the modulus n, so it is usual to assume that solving RSA requires roughly the same computational effort as factoring the modulus. Nevertheless, there is evidence that breaking RSA cannot be equivalent to factoring, as shown in [BV98].

Regarding decisional QR assumptions, the only way known to solve them is answering the related computational problem, that is, computing square roots modulo n. But this problem is equivalent to factoring, so for practical purposes the computational

complexity of QR problems is set to be similar to factoring n. However, since there is no known reduction from factoring to solving QR, it is reasonable to trust the hardness of the Gap-Rabin problem, but this problem has still not been studied in depth.

In the case of DSeR and DS2eR assumptions, the fastest attack known is to solve its computational version, that is, we answer these problems by finding a solution of the equations $x^e = c \bmod n^2$ with $c \in \mathbb{Z}^*_{n^2}$, or $x^{2e} = c \bmod n^2$, with $c \in Q_{n^2}$ respectively. So we are confronted with the problem of finding small solutions of low degree polynomials. The best option is to apply the following result due to Coppersmith [Cop96]:

Theorem 57 *Let N be an integer and let $f(x) \in \mathbb{Z}_N[x]$ be a monic polynomial of degree d. Then there is an efficient algorithm to find all $x_0 \in \mathbb{Z}$ such that $f(x_0) = 0 \bmod N$ and $|x_0| < N^{1/d}$.*

In our case, given the equations $c = x^e \bmod n^2$ or $c = x^{2e} \bmod n^2$, we must find a root $x < n$. Coppersmith's result ensures this is efficiently computable (i.e. in polynomial time) for all $|x| < n^{2/e}$ and $|x| < n^{2/2e} = n^{1/e}$ respectively. For all values x greater than this bound, there is at present no polynomial algorithm solving this problem when the factorization of n is unknown, despite much research in this topic (for instance in [HG99, BD99, BDHG99, HG01]). Then for any $e \geq 5$ and $e \geq 3$, respectively, the assumptions seem to be valid, with hardness depending on the size of the exponent e.

Attacks against discrete logarithm problems

The discrete logarithm related to a group description $(\mathbb{F}_q, G, g, p)$ can be solved by computing discrete logarithms in $\mathbb{F}_q$ or in the subgroup G. In the first case, the best algorithm known is a variation of NFS, referred as DLNFS, which finds a discrete logarithm in $\mathbb{F}_q$ in expected time proportional to $L[p]$. Furthermore, the subgroup discrete logarithm can also be attacked by Pollard's ρ method [Pol78]. This method can be applied to any group, as long as the group elements allow a unique representation, and the group law can be applied efficiently, which is the case. The expected running time of Pollard's ρ method is exponential in the size of q, namely $1.25\sqrt{q}$ multiplications in $\mathbb{F}_q$, and its storage requirements are very small. Then, solving DL requires subexponential time in $|q|$ and exponential time in $|p|$, and then $|p|$ can be much smaller than $|q|$.

In the case of CDH, it is widely believed that it is equivalent to DL. Maurer and Wolf have shown in several works explicit reductions from DL to CDH for many groups, as can be seen in their survey [MW00]. Another point supporting this claim is that no group for which CDH problem is substantially easier than DL problem has been exhibited up to now. However there are still many groups for which this equivalence is still not proven. For these reasons, CDH and DL are assumed to have similar complexity.

The DDH problem appears to be easier than the CDH problem in general. For instance, consider a subgroup G with order $|G| = 2p$ where p is a large prime, and for which CDH is hard. Then, with probability $3/4$, the correct DDH tuple can be recognized. The reason is that from g^a, one can determine $a \bmod 2$ by computing $(g^a)^p$ (see [MW00] for more details). Generally, the CDH problem can be hard in a group G if $|G|$ contains *at least one large* prime factor, whereas the DDH problem can only be hard if $|G|$ is *free of small prime factors*. There is no known reduction from CDH to DDH but, as in the previous cases, they are assumed to have the same computational complexity provided that DDH is believed hard to solve.

Attacks against elliptic curve discrete logarithm problems

The best algorithm to date for solving the ECDL problem related to a group description (E, G, P, p), is an improved version of the Pollar ρ-method, which takes about $\sqrt{\pi p}/2$ elliptic curve additions. Thus, solving ECDL requires time exponential in $|p|$. For this reason ECDL-based schemes need smaller key sizes than factoring, RSA and DL based schemes for the same level of security.

However, care must be taken when dealing with special families of elliptic curves. Indeed there exist certain elliptic curves where it is possible to solve ECDL in subexponential time and even in polynomial time: supersingular and anomalous curves. An elliptic curve E over $\mathbb{F}_q$ is said to be *prime-field-anomalous* if $|E(\mathbb{F}_q)| = q$. Semaev [Sem98], Smart [Sma99] and Satoh and Araki [SA98] independently showed how to compute efficiently an isomorphism between $E(\mathbb{F}_q)$, where E is a prime-field-anomalous curve, and the additive group of $\mathbb{F}_q$. This gives a PT algorithm for ECDL in $E(\mathbb{F}_q)$. The attack does not appear to extend to any other class of elliptic curves. Such curves are then useless for cryptographic purposes.

Menezes, Okamoto and Vanstone [MOV93] used the Weil pairing on an elliptic curve E to embed the group $E(\mathbb{F}_q)$ in the multiplicative group of the field $\mathbb{F}_{q^k}$ for some integer k, called the *embedding degree*. This reduces ECDL in $E(\mathbb{F}_q)$ to DL in $\mathbb{F}^*_{q^k}$. A necessary condition for G to be embedded in $\mathbb{F}^*_{q^k}$ is that p divides $q^k - 1$. Now in $\mathbb{F}^*_{q^k}$ we can use the DLNFS method with subexponential running time $L[q^k]$. For the case when q is a power of 2 or when q is prime and $k = 1$, there exist algorithms computing DL with that computational complexity. No algorithm with running time $L[q^k]$ is known when q is odd and $k > 1$, but we adopt the supposition that this time estimate is the complexity of the discrete logarithm problem in $\mathbb{F}^*_{q^k}$ for all q and $k \geq 1$, as suggested in [KMV00].

Note that k must be less than $\log^2 q$, since otherwise the DLNFS algorithm for $\mathbb{F}^*_{q^k}$ will take exponential time in $|q|$. This is the case of supersingular curves, which are known to have $k \leq 6$. For these curves, the MOV reduction gives a subexponential-

time algorithm for the ECDL. In contrast, for most randomly generated elliptic curves (for instance, with distributions $I^{\mathbf{EC}}_{\mathsf{rand}}$ or $I^{\mathbf{EC}}_{\mathsf{CM}}$) it turns out that $k > \log^2 q$, so MOV reduction does not threaten the security of ECDL in this case.

In order to prevent MOV reduction attacks, a good cryptographic practice when selecting an elliptic curve is to check that p does not divide $q^k - 1$ for any $1 \leq k \leq C$, which implies that the embedding degree is greater than C. This checking is done, for instance, in [IEE99, X9.99], and setting $k = 20$ is considered enough. Despite this, curves with relatively small embedding degree k, for example $6 \leq k \leq 20$ are being considered for cryptographic applications using efficiently computable non-degenerate bilinear maps [Jou00, DBS04]. In this case, one must guarantee the infeasibility of ECDL and DL in G and $\mathbb{F}^*_{q^k}$ respectively.

With respect to ECDH and ECDL, the situation is similar to the finite field case. There is even more evidence about the hardness of ECDH, since Boneh and Lipton proved in [BL96] that if ECDL cannot be solved in subexponential time, then neither can ECDH.

Things are quite different for ECDDH. It has been recently published [JN03] that the hardness of ECDDH is a much stronger hypothesis than the hardness of the regular CDH problem, describing reasonably looking cryptographic groups where ECDDH is easy while ECDH is presumably hard. These groups are derived from the special family of elliptic curves with bilinear maps. In contrast, ECDDH is believed to be hard over randomly generated elliptic curves, where as usual it is assumed that ECDDH and ECDL have similar complexity, even though no reduction has actually been presented.

1.4.4 Concrete security

As explained in the introduction, security proving works via a reduction: if we were able to defeat a particular security level in an encryption scheme, then we would be able to solve a conjectured unsolvable mathematical problem. But these equivalences hold asymptotically, that is, they guarantee that for a sufficiently large security parameter 1^ℓ we would have a low probability $\varepsilon(\ell)$ when trying to break the protocol. However, when a cryptographic scheme is used in practice, a fixed value ℓ_0 for the security parameter is used, so we would have to *quantify* the security for this particular value. This is the idea behind concrete security.

At first glance, one must consider the computational effort that the fastest known method needs to invest to solve a particular hard problem with a concrete security parameter. Since this information has been provided in the previous section, one could try to derive particular values for the security parameter from these theoretical estimates. However, this approximation is still incomplete, since a theoretically feasible attack could be impractical or even infeasible in the real world, because of its memory storage requirements or the resources needed to design a machine efficiently implementing it.

The most satisfactory work dealing with this problem is [LV01] by Lenstra and Verheul. They take into account not only cryptanalysis matters, but also the expected change in computational resources available to attackers and its relation with the life span of the key; economic considerations, etc.

To be consistent with the time units commonly used in the literature, we use the sentence *a problem $\mathcal{P}$ has a 2^t security level* to say that an attacker against $\mathcal{P}$, running in time less than 2^t 3-DES encryptions[2], has a "negligible" success probability. In Table 1.2 we present some of the results obtained by Lenstra and Verheul. They must be read in the following way: the first line in the table means that until the year 2012 a computational effort equivalent to 2^{80} 3-DES encryptions is assumed to be infeasible; and that either factoring an RSA modulus with bit-length in the interval [1120,1464], or computing the DL in a [1120,1464] bit-length finite field along with a 139 bit-length subgroup, or solving ECDL in a elliptic curve subgroup with a length between 149 and 165 bits, requires a computational effort equivalent to 2^{80} 3-DES encryptions. We stress that this in indeed the current demmanded security level.

Year	Symmetric key size	Asymmetric key size and field size	FF subgroup size	Elliptic curve key size
2012	80	[1120,1464]	139	[149,165]
2026	90	[1792,2236]	160	[170,205]
2040	101	[2656,3214]	179	[191,244]
2050	109	[3392,4047]	193	[206,272]

Table 1.2: Cryptographic key sizes

Another issue of interest for the study of concrete security is the efficiency of a reduction. This is defined as the relationship between an *attacker* who breaks the cryptosystem with probability at least ε in time t, doing less than q_D calls to a decryption oracle, and less than q_H calls to an oracle for a hash function; and the implied (t', ε') *solver* against the corresponding trusted cryptographic assumption. Such an attacker is referred to as a $(t, \varepsilon, q_D, q_H)$ attacker for short. Then, the security reduction is *tight* if $\frac{t'}{\varepsilon'} \approx \frac{t}{\varepsilon}$, and *not tight* if $\frac{t'}{\varepsilon'} > q_D \frac{t}{\varepsilon}$. It is also stated that a scheme is *very tight* if $\varepsilon \approx \varepsilon'$ and t' is equal to t plus a linear quantity in the number of oracle calls. The tighter the reduction is, the smaller the gap between the computational efforts needed to break the scheme and to solve the underlying problem. This has a great impact on the efficiency of the scheme, since a tight security reduction allows us to use smaller security parameters. For instance, if the security reduction to factoring of some cryptographic protocol is very tight, then the use of a 1464-bit length RSA modulus makes the protocol secure against adversaries with running time bounded by 2^{80} 3-DES encryptions. In contrast, if

[2] 3-DES is a very popular symmetric encryption scheme.

the reduction is not tight, the key length must be notably increased. This is the case for the RSA-OAEP scheme reduction in [FOPS01], in which $t' \approx t + q_H^2 \cdot \ell^3$ and $\varepsilon' \approx \varepsilon^2$, and then an RSA modulus with more than 4000-bit long is needed to attain the same security level [Poi02].

Chapter 2

Semantically Secure Encryption Schemes against Passive Adversaries

In this chapter, new schemes with semantic security against passive adversaries in the standard model are presented and analysed. Both schemes base their security on factoring related hard problems and have a fast encryption.

2.1 Rabin-Paillier Encryption Scheme

Designing practical IND-CCA schemes in the standard model is quite a difficult task. The most appealing approach used up to now is found in [CS98] and [CS02]. In this setting, the security of the proposed IND-CCA schemes is only based on number-theoretic decisional assumptions. The technique used in [CS98] and [CS02] is to improve existing IND-CPA schemes under appropiate and widely accepted decisional assumptions, obtaining IND-CCA schemes based on the same assumptions and without significantly degrading their efficiency. There exist three different realizations in this setting, which are based on the Decisional Diffie-Hellman, Decision Composite Residuosity [Pai99] and the classical Quadratic Residuosity assumptions respectively. It would be of great interest to construct IND-CCA schemes from the RSA and Rabin-Williams primitives in this model. A decisional assumption's candidate for the RSA scheme was proposed in the modification of Paillier scheme [CGHN01], and the proof of the equivalence between the one-wayness of this scheme and the RSA scheme was presented shortly after in [CNS02]. The validity of this new assumption deserves further study and developing an IND-CCA scheme from it remains an open problem. As far as we know, no decisional number-theoretic problem for the Rabin-Williams primitive has been proposed.

Our results

In first place, we construct a new trapdoor permutation based on factoring, which has interest on its own. Trapdoor permutations play an important role in cryptography. Many theoretic schemes use this object as a building block, in such a way that any trapdoor permutation can be easily transformed into IND-CCA ciphering (although very impractical), signature, or authentication schemes, for instance. Despite this fact, few candidate trapdoor permutations are known, and fewer that are as secure as factoring (cf. [PG97]). The new trapdoor permutation is obtained from a modification of RSA-Paillier's trapdoor permutation [CGHN01], which is reminiscent from the modifications applied by Rabin [Rab79] and Williams [Wil80] to RSA cryptosystem. Then, using this new function as a primitive, we design a new cryptosystem which is one-way under the Blum-RSA factoring assumption, and IND-CPA under the Decisional Small $2e$-Residues (DS$2e$R) assumption. We call it *Rabin-Paillier* scheme. We summarize hereafter the main features of the proposed scheme:

- We take profit of the nice characteristics of Rabin schemes and overcome their drawbacks, by using the Rabin-Williams function to hide the randomness. More precisely, the encryption of a message $m \in \mathbb{Z}_n$ with randomness $r \in Q_n$ is defined as $E(r, m) = r^{2e} + mn \bmod n^2$, where e is an integer of small size.

- It is remarkable that the scheme allows to encrypt arbitrary messages with a very simple procedure, that does not depend further on the form of the message to be enciphered, which was the case for the previous Rabin based schemes. Besides, the efficiency is similar to that of plain RSA.

- The scheme is IND-CPA under the Decisional Small 2e-Residues assumption (DS$2e$R).

Although the scheme is obtained by a simple modification of the RSA-Paillier scheme, this modification deeply influences the underlying mathematical structure. This was in turn the case of RSA-Paillier scheme with respect to the original Paillier scheme [Pai99]. The main difference is that the one-wayness of the new scheme is equivalent to factoring, and independent of the size of the exponent e. This is also the case for the Rabin-Williams primitive, a fact that has raised some doubt on the usefulness of taking $e > 1$. We show that using such an exponent value is meaningful since its size plays a crucial role in the hardness of our new decisional assumption. We are not aware of any previous result showing an utility for $e > 1$ in Rabin-Williams function.

We can compare our scheme with the Okamoto-Uchiyama's scheme (OU) [OU98], which we briefly recall in the following. For a security parameter 1^ℓ, the public key is

(n, g, h, ℓ), where $n = p^2q$ with $p, q \leftarrow PRIMES(\ell)$; $g \in \mathbb{Z}_n^*$ is such that the order of g^{p-1} in $\mathbb{Z}_{p^2}^*$ is p, and $h = g^n \bmod n$. Then, the encryption function takes a message $0 < m < 2^{\ell-1}$, randomness $r \leftarrow \mathbb{Z}_n$ and computes the ciphertext $C = g^m h^r \bmod n$. Finally, the decryption algorithm computes

$$m = \frac{L(C^{p-1} \bmod p^2)}{L(g^{p-1} \bmod p^2)} \bmod p, \quad \text{where} \quad L(x) = \frac{x-1}{p}.$$

The one-wayness of OU is equivalent to factoring $n = p^2q$, whereas in our case is equivalent to factoring $n = pq$, which is the classical factoring assumption. Our scheme is drastically more efficient in ciphering, since OU presents an encryption cost proportional to the lenght of the modulus n. For instance, an exponent value $e = 17$ can be safely used in our scheme, and then only 6 multiplications modulo n^2 are needed to encrypt $m \in \mathbb{Z}_n$. Besides, our scheme presents an expansion factor 2, while OU's expansion factor is 3. However, OU scheme is homomorphic, and more efficient in decryption than ours.

It makes also sense to compare our scheme with the efficient Blum-Goldwasser (BG) IND-CPA scheme [BG85]. Its semantic security is based on the Blum factoring assumption, that is, a computational assumption. Roughly one hundred squares modulo n are needed to encrypt $m \in \mathbb{Z}_n$ and the expansion factor is 2 (see Section 8.7.2 in [MvOV97]). Its decryption time is similar to RSA. Thus, our scheme is roughly ten times faster in encryption than BG scheme while presenting the same expansion factor, but 2 times slower in decryption.

The main drawback of our scheme is that, as well as in the previous schemes with one-wayness equivalent to factoring, there exist a chosen ciphertext attack that completely breaks the scheme. It remains an open problem to further study the validity of the DS2eR assumption and to modify our scheme to achieve IND-CCA security under the DS2eR assumption in the standard model.

2.1.1 Some previous schemes and related trapdoor permutations

In this section, we briefly recall some previous schemes and related trapdoor permutations, from which we will derive the new trapdoor permutation based on factoring, and the scheme we propose. We denote by $\mathrm{RSA}[n, e]$ the RSA function with public exponent e, i.e. $\mathrm{RSA}[n, e](x) = x^e \bmod n$, $x \in \mathbb{Z}_n^*$.

Rabin function

Let p, q be two different primes with equal length, $n = pq$. Rabin proposed in [Rab79] a provably secure cryptosystem based on the modular squaring function

$$\begin{array}{rcl} \mathbb{Z}_n^* & \longrightarrow & Q_n \\ x & \longmapsto & x^2 \bmod n\,. \end{array}$$

It is well known that modular squaring is a trapdoor one-way function assuming that factorisation of large numbers is infeasible. However, modular squaring is a 4 to 1 function, so a ciphertext is not uniquely decrypted. In order to avoid this drawback and to speed up the decryption algorithm (i.e. the computation of square roots modulo n), the following proposal by Blum and Williams can be considered:

Blum-Williams function

Let p, q be (different) primes with equal length, $p \equiv q \equiv 3 \bmod 4$, $n = pq$. The squaring function restricted to Q_n, i.e.

$$\begin{array}{rcl} \mathcal{G}_n : Q_n & \longrightarrow & Q_n \\ x & \longmapsto & x^2 \bmod n \end{array}$$

is a trapdoor one-way permutation if factoring large numbers is infeasible. Then, if we restrict the set of messages to Q_n, a ciphertext will be uniquely decrypted. However, this is not suitable for real applications, since it does not allow to encrypt arbitrary messages. To decrypt $c \in Q_n$ one has to compute $\mathcal{G}_n^{-1}(c)$, i.e. the element $s \in Q_n$ such that $s^2 = c \bmod n$. Let us briefly recall how to make this computation (see [Til99] for a nice account on this). Assume that we know the factorisation of $n = pq$, where $p \equiv q \equiv 3 \bmod 4$. We first compute the numbers $f = c^{\frac{p+1}{4}} \bmod p$ and $g = c^{\frac{q+1}{4}} \bmod q$, which are the square roots of c modulo p and modulo q that are quadratic residues to their respective modulus. Then, by using the Chinese Remainder Theorem, we obtain an $s \in Q_n$ such that $s^2 = c \bmod n$.

Rabin-Williams function

Let p, q be (different) primes with equal length, $p \equiv q \equiv 3 \bmod 4$, $n = pq$ and e a public RSA exponent (i.e. an integer such that $\gcd(e, \lambda(n)) = 1$, where λ denotes Carmichael's function). The map

$$\begin{array}{rcl} \mathcal{W}_e : Q_n & \longrightarrow & Q_n \\ x & \longmapsto & x^{2e} \bmod n \end{array}$$

is also a trapdoor one-way permutation, assuming that factoring large numbers is infeasible, since a perfect reduction to the Blum-Williams function inversion problem can be done as follows. Given $c = \mathcal{G}_n(x) = x^2 \bmod n$, x can be retrieved from $c^e \bmod n = x^{2e} \bmod n$ by inverting the Rabin-Williams function with some non-negligible probability.

RSA-Paillier function

Catalano et al. proposed in [CGHN01] a mix of Paillier's scheme [Pai99] with RSA scheme, in order to obtain an IND-CPA cryptosystem in the standard model with efficiency similar to that of RSA cryptosystem. It is based on the permutation

$$\begin{aligned} \mathcal{E}_e : \mathbb{Z}_n^* \times \mathbb{Z}_n &\longrightarrow \mathbb{Z}_{n^2}^* \\ (r, m) &\longmapsto r^e(1+mn) \bmod n^2, \end{aligned}$$

where p, q are distinct primes with the same length, $n = pq$, and $e \in \mathbb{Z}_n$ is such that $\gcd(e, \lambda(n^2)) = 1$. The encryption scheme $\mathcal{E}_e(r, m)$ with randomness $r \in \mathbb{Z}_n^*$ is semantically secure under the Decisional Small e-Residues assumption.

Sakurai and Takagi claimed in [ST02] that deciphering RSA-Paillier scheme with public exponent e is actually equivalent to inverting the original RSA$[n, e]$ function. However, Catalano, Nguyen and Stern found a flaw in the proof by Takagi and Sakurai, and they proposed in [CNS02] an alternative proof of the claim in [ST02]. Therefore, RSA-Paillier scheme is a practical semantically secure RSA-type scheme in the standard model.

2.1.2 New trapdoor permutation based on factoring

In this section we present a new length-preserving trapdoor permutation based on factoring, i.e. a length-preserving bijection that is one-way assuming that factoring large integers is hard. It is worthwhile to remark that as well as ours, all previous trapdoor permutations provably secure are based on the factoring problem [PG97]. In this context, we say a cryptographic scheme is provably secure if it is proven to be as secure as the underlying primitive problems (i.e., discrete logarithm or factoring problems). To the best of our knowledge, only two length-preserving provably secure trapdoor permutations exist, namely, the Rabin-Williams permutation, and another one proposed by Gong and Harn in [GH99].

A new trapdoor permutation

Let $(n, p, q, e, d) \leftarrow I_{\mathsf{Blum}}^{\mathsf{RSA}}(1^\ell)$ for some security parameter 1^ℓ, and

$$\mathcal{F}_e : Q_n \times \mathbb{Z}_n \longrightarrow Q_{n^2}$$

$$(r, m) \longmapsto r^{2e} + mn \bmod n^2.$$

Proposition 58 *$\mathcal{F}_e$ is a well-defined length-preserving bijection.*

Proof: From the Hensel-lifting, the set of quadratic residues modulo n^2 can be alternatively defined as $Q_{n^2} = \{x + yn \mid x \in Q_n, y \in \mathbb{Z}_n\}$. Then if $c = \mathcal{F}_e(r, m) = r^{2e} + mn \bmod n^2$, with $r \in Q_n$, $m \in \mathbb{Z}_n$, it is obvious that $c \bmod n = r^{2e} \bmod n \in Q_n$, which implies that $\mathcal{F}_e$ is well-defined.

To prove that $\mathcal{F}_e$ is bijective it suffices to show that it is injective, because, from the alternative definition of Q_{n^2}, we deduce that the sets $Q_n \times \mathbb{Z}_n$ and Q_{n^2} have the same number of elements. Let us suppose that $\mathcal{F}_e(r_0, m_0) = \mathcal{F}_e(r_1, m_1)$. Then $r_0^{2e} = r_1^{2e} \bmod n$, and since squaring and computing e-th powers modulo n, with $\gcd(e, \lambda(n)) = 1$, are bijections over Q_n, we conclude that $r_0 = r_1 \bmod n$. This implies $m_0 n = m_1 n \bmod n^2$, so $m_0 = m_1 \bmod n$.

Finally, $\mathcal{F}_e$ is length-preserving, since the natural bit representation of an arbitrary element either in $Q_n \times \mathbb{Z}_n$ or in Q_{n^2} has length $2\lceil \log_2 n \rceil$. ∎

In the sequel we prove that inverting $\mathcal{F}_e$ is as difficult as factoring the modulus n.

Assumption 59 *$\mathcal{G}_n$ is* Trapdoor One-Way *with respect to the keypair generator* $(PK^{\mathsf{RSA}}, SK^{\mathsf{RSA}}, I^{\mathsf{RSA}}_{\mathsf{Blum}})$, *that is, for every* PPT *algorithm $\mathcal{A}$,*

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n, c) = r \mid (n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{Blum}},\ r \leftarrow Q_n,\ c = r^2 \bmod n\right] \in \mathsf{negl}(\ell).$$

Proposition 60 *$\mathcal{G}_n$ is* TOW *if and only if the* Blum-RSA factoring *assumption holds.*

Proof:

($\Rightarrow$) Let us suppose that the Blum-RSA factoring assumption does not hold. Then there exists a *PPT* algorithm that factors $n = pq$ with a non-negligible probability ε. Given $c \in Q_n$ and knowing p and q, compute $x_p = x^{\frac{p+1}{4}} \bmod p$ and $x_q = x^{\frac{q+1}{4}} \bmod q$. Then, using the Chinese Remainder Theorem, $x \in \mathbb{Z}_n^*$ is obtained such that $x^2 = c \bmod n$ and $x \in Q_n$ with probability ε.

($\Leftarrow$) Let us assume that $\mathcal{G}_n$ is not TOW. Then, there exists a *PPT* algorithm $\mathcal{A}$ such that given $c \leftarrow Q_n$ returns $x \in Q_n$, such that $x^2 = c \bmod n$, with probability ε. We first randomly choose $\bar{x} \in \mathbb{Z}_n^*$ such that $(\frac{\bar{x}}{n}) = -1$, and compute $c = \bar{x}^2 \bmod n$, which is uniformly distributed in Q_n. Let $\mathcal{A}(n, c) = x \in Q_n$. We claim that $\gcd(n, x - \bar{x}) = p$ or q. Notice that $\bar{x}$ satisfies (I) $\{(\frac{\bar{x}}{p}) = 1$ and $(\frac{\bar{x}}{q}) = -1\}$ or (II) $\{(\frac{\bar{x}}{p}) = -1$ and $(\frac{\bar{x}}{q}) = 1\}$.

Since $(\frac{x}{p}) = (\frac{x}{q}) = 1$, and $(\frac{-1}{p}) = (\frac{-1}{q}) = -1$, then in case (I) $\bar{x} = x \bmod p$ and $\bar{x} = -x \bmod q$, while in (II) $\bar{x} = -x \bmod p$ and $\bar{x} = x \bmod q$. Therefore the claim holds, and n is factored with probability ε. ∎

Assumption 61 *$\mathcal{F}_e$ is* TOW *with respect to the keypair generator $(PK^{\mathsf{RSA}}, SK^{\mathsf{RSA}}, I^{\mathsf{RSA}}_{\mathsf{Blum}})$, that is, for every* PPT *algorithm $\mathcal{A}$,*

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n, e, c) = (r, m) \;\middle|\; \begin{array}{l}(n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{Blum}} \\ r \leftarrow Q_n,\ m \leftarrow \mathbb{Z}_n,\ c = \mathcal{F}_e(r, m)\end{array}\right] \in \mathsf{negl}(\ell).$$

Proposition 62 *$\mathcal{F}_e$ is a* TOW *permutation with respect to the keypair generator $(PK^{\mathsf{RSA}}, SK^{\mathsf{RSA}}, I^{\mathsf{RSA}}_{\mathsf{Blum}})$, if and only if the* Blum-RSA factoring *assumption holds.*

Proof:

(⇒) Let us suppose that the Blum-RSA factoring assumption does not hold. Then there exists a polynomial time algorithm that factors $n = pq$ with a non-negligible probability ε. Knowing p and q, one can compute $d \in \mathbb{Z}_n^*$ s.t. $de \equiv 1 \bmod \lambda(n)$, since $\gcd(e, \lambda(n)) = 1$. For any $c \in Q_{n^2}$ we can also compute $r = \mathcal{G}_n^{-1}(c^d \bmod n) = \mathcal{G}_n^{-1}(r^2 \bmod n)$, and $m \in \mathbb{Z}_n$ from the equality $mn = c - r^{2e} \bmod n^2$. These values are such that $\mathcal{F}_e(r, m) = c$, so we can invert $\mathcal{F}_e$ on $c \leftarrow Q_{n^2}$ with non-negligible success probability ε, which implies that $\mathcal{F}_e$ is not one-way.

(⇐) Let us suppose that $\mathcal{F}_e$ is not one-way. Recall that e is a prime Fermat number (or has been chosen from some particular known set) and that $\gcd(e, \lambda(n)) = 1$. The goal is to show that a *PPT* algorithm that inverts $\mathcal{F}_e$ on a random input can be transformed into another algorithm that inverts Blum-Williams permutation $\mathcal{G}_n$. Assume then we are given a security parameter 1^ℓ, an integer n and $c \in Q_n$ with the distributions described in assumption 59. Let $c' = c^e + mn \bmod n^2$, where $m \leftarrow \mathbb{Z}_n$. Then, since c was uniformly chosen in Q_n and the map

$$\begin{array}{rcl} Q_n \times \mathbb{Z}_n & \longrightarrow & Q_{n^2} \\ (c, m) & \longmapsto & c^e + mn \bmod n^2 \end{array}$$

is a bijection, we deduce that c' is uniformly distributed in Q_{n^2}. Let $(r, m') = \mathcal{A}(1^\ell, n, e, c')$, where $\mathcal{A}$ is the algorithm that inverts $\mathcal{F}_e$ on a random input with a non-negligible probability ε. If $\mathcal{A}$ gives the correct answer, then $c^e + mn = r^{2e} + m'n \bmod n^2$. Reducing this equality modulo n, we have $r^{2e} = c^e \bmod n$, which is equivalent to $c = r^2 \bmod n$, since $\gcd(e, \lambda(n)) = 1$. Then $\mathcal{G}_n^{-1}(c) = r$ with probability ε. Applying Proposition 60 n is factored with probability ε. ∎

2.1.3 Rabin-Paillier scheme

Using the permutation $\mathcal{F}_e$ as a primitive, we are able to develop the following encryption scheme, which we call Rabin-Paillier scheme.

Key generation. Given a security parameter 1^ℓ, $(\mathsf{pk}, \mathsf{sk}) \leftarrow I^{\mathsf{RSA}}_{\mathsf{Blum}}(1^\ell)$, that is, $\mathsf{pk} = (n, e)$ and $\mathsf{sk} = (n, p, q, e, d)$. Let us observe that the integer e satisfy $\gcd(e, \lambda(n)) = 1$. In order to be able to prove that one-wayness is equivalent to factoring, as well as semantic security, we need in addition that $\gcd(e, n) = 1$ and $e > 2$ respectively. Since $\lambda(n^2) = n\lambda(n)$, the first condition implies $\gcd(e, \lambda(n^2)) = 1$. Notice that for real security parameters these additional requirements are trivially satisfied in the distribution $I^{\mathsf{RSA}}_{\mathsf{Blum}}$ as long as e is a small prime number.

Encryption. To encrypt a message $m \in \mathbb{Z}_n$ we compute $c = \mathcal{F}_e(r, m)$, where r is randomly chosen in Q_n. The choice of the randomness in Q_n can be done, for instance, by selecting $s \leftarrow \mathbb{Z}^*_n$ at random, and computing $r = s^2 \bmod n$.

Decryption. To recover the message m from $c = \mathcal{F}_e(r, m)$, the randomness r is computed firstly, and, afterwards, m is easily obtained from $mn = c - r^{2e} \bmod n^2$. To obtain r from c, we compute $t = \mathrm{RSA}[n, e]^{-1}(c \bmod n) = c^d \bmod n$, and then $r = \mathcal{G}_n^{-1}(t)$, computed as explained in section 2.1.1.

2.1.4 Security analysis

In this section we discuss the security properties of the encryption scheme, namely, its one-wayness and semantic security against passive adversaries. We show the scheme is OW under the Blum-RSA factoring assumption and IND-CPA under the DS2eR assumption.

One-wayness

In order to study the one-wayness of the scheme, we introduce a new computational problem which is closely related. Afterwards, we prove that the new computational problem is intractable if and only if the factoring problem is intractable. In fact, the new problem is the natural extension to our case of the questions dealt with in [ST02] and [CNS02].

In [CNS02], given an RSA modulus n and a public exponent e relatively prime to $\lambda(n)$, the following function from $\mathbb{Z}^*_n$ to $\mathbb{Z}^*_{n^l}$, for $l > 1$, is defined:

$$\textbf{Hensel-RSA}[n, e, l](r^e \bmod n) = r^e \bmod n^l \,,$$

and it is proven that the hardness of computing such a function is equivalent to the RSA assumption as stated in definition 41. With some slight modifications, the arguments in [CNS02] can be applied to our encryption scheme. Let us consider the **Hensel-Rabin-Williams** function from Q_n to Q_{n^l} defined as

$$\mathbf{Hensel\text{-}RW}[n, e, l](r^{2e} \bmod n) = r^{2e} \bmod n^l \,,$$

where $r \in Q_n$. The following proposition can then be stated

Proposition 63 *Computing* $\mathbf{Hensel\text{-}RW}[n, e, 2]$ *with respect to the keypair generator* $(PK^{\mathsf{RSA}}, SK^{\mathsf{RSA}}, I^{\mathsf{RSA}}_{\mathsf{Blum}})$*, on a random element* $w \in Q_n$*, is hard if and only if the function* $\mathcal{W}_e$ *is* TOW *with respect to* $(PK^{\mathsf{RSA}}, SK^{\mathsf{RSA}}, I^{\mathsf{RSA}}_{\mathsf{Blum}})$.

Proof:

($\Rightarrow$) If $\mathcal{W}_e$ is not one-way, then r can be computed from $r^{2e} \bmod n$ with non-negligible probability and therefore $\mathbf{Hensel\text{-}RW}[n, e, 2](r^{2e})$ is trivially computed.

($\Leftarrow$) The adversary, who wants to invert Rabin-Williams function on a random input $r^{2e} \bmod n$, calls an oracle twice for the $\mathbf{Hensel\text{-}RW}[n, e, 2]$ on inputs r^{2e} and $r^{2e}a^{2e}$, where a is randomly chosen in Q_n. Assuming that ε is the probability that the oracle gives the right answer, the adversary knows $r^{2e} \bmod n^2$ and $\mu^{2e} \bmod n^2$, where $\mu = ar \bmod n$, with probability ε^2. Then, it follows that there exists $z \in \mathbb{Z}_n$ such that

$$ar = \mu(1 + zn) \bmod n^2. \tag{2.1}$$

Raising this equality to the power $2e$ we obtain the equation

$$a^{2e}r^{2e} = \mu^{2e}(1 + 2ezn) \bmod n^2,$$

from which z can be computed, since the rest of values involved are known. The last step is the computation of r and μ from equation (2.1). This can be done by using lattice reduction techniques (see [CNS02] for further details). Therefore, $\mathcal{W}_e$ is inverted with probability ε^2. ∎

The following lemma states the relation between computing $\mathbf{Hensel\text{-}RW}[n, e, 2]$ function and the one-wayness of our scheme.

Lemma 64 *The encryption scheme described in Section 2.1.3 is* OW *if and only if computing* $\mathbf{Hensel\text{-}RW}[n, e, 2]$ *with respect to* $(PK^{\mathsf{RSA}}, SK^{\mathsf{RSA}}, I^{\mathsf{RSA}}_{\mathsf{Blum}})$ *on a random input is hard.*

Proof:

($\Rightarrow$) For a random ciphertext $c \leftarrow Q_{n^2}$, the message m is easily recovered from the **Hensel-Rabin-Williams** oracle since $mn = c - \mathbf{Hensel\text{-}RW}[n, e, 2](c \bmod n)$.

($\Leftarrow$) To compute $\mathbf{Hensel\text{-}RW}[n, e, 2]$ on $c_0 \leftarrow Q_n$, it suffices to choose $m_0 \leftarrow \mathbb{Z}_n$, and submit $c_0 + m_0 n$ to the adversary that is able to invert the proposed cryptosystem with a non-negligible probability ε. (Note that m_0 is intended to match the exact probability distribution needed for the query to the adversary.) Since there exist unique $r \in Q_n$ and $m \in \mathbb{Z}_n$ such that $c_0 + m_0 n = r^{2e} + mn \bmod n^2$, the adversary answers m with probability ε. Then, $\mathbf{Hensel\text{-}RW}[n, e, 2](c_0) = c_0 + (m_0 - m)n \bmod n^2$. ■

The above arguments lead to the following theorem:

Theorem 65 *The encryption scheme described in Section 2.1.3 is* OW *if and only if the* Blum-RSA factoring *assumption holds. Moreover, an adversary against* OW *with success probability ε can be transformed into a factoring algorithm with success probability ε^2.*

Proof: From Lemma 64 and Proposition 63, one-wayness of our scheme is equivalent to one-wayness of the Rabin-Williams function, that is in turn equivalent to the Blum-RSA factoring assumption. ■

Tightness improvement

Kurosawa and Takagi presented in [KT03] an improvement on the reduction we found between the one-wayness of our scheme and factoring. They provide a very tight reduction, that is, an algorithm breaking one-wayness with success probability ε leads to a factoring algorithm with the same success probability. Their reduction is quite simple and doesn't make use of lattice reduction techniques. They prefer to base the one-wayness of our scheme on the classical factoring assumption, instead of the Blum-RSA factoring assumption we use. But there is a price to pay in efficiency in their choice: a prime encryption exponent $e > \sqrt{n}$ is needed to prove their result. In this case, Rabin-Paillier scheme doesn't present anymore better efficiency in encryption than OU scheme. See [KT03] for more details and Section 1.4.1 for our discussion on factoring assumptions.

At this point, we have to notice that, as the previous schemes with one-wayness based on factoring, there exists a chosen ciphertext attack that completely breaks our cryptosystem. The reason for this is that a decryption oracle $\mathcal{O}_D$ for our scheme can be exploited to compute the $\mathbf{Hensel\text{-}RW}[n, e, 2]$ function. Indeed, let $s \leftarrow Q_n$, $m \leftarrow \mathbb{Z}_n$ and $c = s + mn \bmod n^2$. Therefore, if $m' = \mathcal{O}_D(c)$, then $\mathbf{Hensel\text{-}RW}[n, e, 2](s) = c - m'n \bmod n^2$. Finally, by applying Proposition 63 computing $\mathbf{Hensel\text{-}RW}[n, e, 2]$ is equivalent to factoring.

Semantic security

Let us recall the DS2eR assumption.

DS2eR assumption. The probability distributions $D_{1,n}, D_{2,n}$ induced by the following random variables X_1, X_2 over Q_{n^2} are polynomially indistinguishable:

$$X_1 = (n, e, y) \quad \text{where} \quad (n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{BlumPrac}}(1^\ell),\ x \leftarrow Q_n,\ y = x^{2e} \bmod n^2\,,$$
$$X_2 = (n, e, y) \quad \text{where} \quad (n, p, q, e, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{BlumPrac}}(1^\ell),\ y \leftarrow Q_{n^2}\,.$$

Proposition 66 *The encryption scheme described in Section 2.1.3 is semantically secure if and only if* DS2eR *assumption holds.*

Proof: Semantic security is equivalent to indistinguishability of encryptions, that is, for all $m_0 \in \mathbb{Z}_n$, the distributions

$$\begin{aligned} D_0 &= (n, e,\ r^{2e} + m_0 n \mod n^2) \text{ where } r \leftarrow Q_n \quad \text{and} \\ D &= (n, e,\ r^{2e} + mn \mod n^2) \text{ where } r \leftarrow Q_n,\ m \leftarrow \mathbb{Z}_n \end{aligned}$$

are polynomially indistinguishable. It is easy to see that the map

$$\begin{aligned} Q_{n^2} &\longrightarrow Q_{n^2} \\ c &\longmapsto c - m_0 n \bmod n^2 \end{aligned}$$

is a polynomial time bijection. Then, applying Property 18, $D_0 \approx D$ is equivalent to

$$(n, e, r^{2e} \bmod n^2) \approx (n, e, r^{2e} + m'n \mod n^2),\ \text{where } r \leftarrow Q_n, m' \leftarrow \mathbb{Z}_n.$$

Note that the distribution on the left side is $D_{1,n}$.
Besides, since $r^{2e} + m'n \mod n^2 = \mathcal{F}_e(r, m')$, and $\mathcal{F}_e$ is a bijection, then D and $D_{2,n}$ are identically distributed. ∎

Finally, since $e \geq 3$ in our encryption scheme, the assumption DS2eR seems to be valid (cf. our discussion in Section 1.4.3).

2.2 Lifted-Rabin Elliptic Curve Encryption Scheme

Our aim is to design an elliptic curve cryptosystem with provably secure one-wayness (in the sense of 2.1.2), with semantic security against passive adversaries and with fast encryption in the standard model. The scheme uses arithmetic modulo n^2 and merges ideas from Paillier and Rabin related schemes. As a result, we first describe two new length-preserving trapdoor permutations equivalent to factoring. The one-wayness of the scheme is equivalent to factoring and the semantic security is proved under a reasonable decisional assumption.

2.2.1 Some results about elliptic curves

In the sequel we summarize some results about elliptic curves defined over the finite field $\mathbb{Z}_p$, and over the rings $\mathbb{Z}_{p^2}$ and $\mathbb{Z}_{n^2}$, where n is an RSA modulus. Since in this section we mainly deal with elliptic curves over rings, the notation is slightly changed from the one introduced in Section 1.4.2 for finite fields.

Definition 67 *Let $p > 3$ be a prime. An elliptic curve over the finite field $\mathbb{Z}_p$, denoted by $E_{a,b}(\mathbb{Z}_p)$, is the set of points $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ such that $y^2 = x^3 + ax + b \bmod p$, where $a, b \in \mathbb{Z}_p$, and $\gcd(4a^3 + 27b^2, p) = 1$, together with a point $\mathcal{O}$ called* the point at infinity.

Recall that the set $E_{a,b}(\mathbb{Z}_p)$ is a group, with the usual tangent-and-chord operation described in 1.4.2. We denote by $|E_{a,b}(\mathbb{Z}_p)|$ the number of elements of the group $E_{a,b}(\mathbb{Z}_p)$. Elliptic curves can also be defined on the projective plane $\mathbb{P}^2(\mathbb{Z}_p)$ as the set of points $(x : y : z)$ satisfying $y^2z = x^3 + axz^2 + bz^3 \bmod p$, and $\gcd(x, y, z, p) = 1$. In particular, the point $(0 : 1 : 0)$ corresponds to the point at infinity $\mathcal{O}$. Following [Gal02], this definition can be extended to the ring $\mathbb{Z}_{p^2}$. The natural map $\pi_p : E_{a,b}(\mathbb{Z}_{p^2}) \to E_{a,b}(\mathbb{Z}_p)$ that reduces coordinates modulo p, is a surjective group morphism whose kernel is the set $\{O_m = (mp : 1 : 0) \mid m \in \mathbb{Z}_p\}$, called the set of points at infinity.

Via the Chinese Remainder Theorem (CRT), $E_{a,b}(\mathbb{Z}_{n^2})$ can be defined as a group isomorphic to $E_{a,b}(\mathbb{Z}_{p^2}) \times E_{a,b}(\mathbb{Z}_{q^2})$ where $n = pq$, and p, q are different odd primes. In the same way, $E_{a,b}(\mathbb{Z}_n)$ can be defined as a group isomorphic to $E_{a,b}(\mathbb{Z}_p) \times E_{a,b}(\mathbb{Z}_q)$. The natural group morphism from $E_{a,b}(\mathbb{Z}_{n^2})$ to $E_{a,b}(\mathbb{Z}_n)$ will be denoted as π_n. Points on curves $E_{a,b}(\mathbb{Z}_{n^2})$ can be classified in three types:

- Points at infinity: $O_m = (mn : 1 : 0)$, $m \in \mathbb{Z}_n$, (the kernel of π_n)
- Affine points: $(x, y) = (x : y : 1) \in E_{a,b}(\mathbb{Z}_{n^2})$.
- Semi-infinite points: $(x : y : z) \in E_{a,b}(\mathbb{Z}_{n^2})$, with $\gcd(z, n) = p$ or q.

Point addition on $E_{a,b}(\mathbb{Z}_p)$ and $E_{a,b}(\mathbb{Z}_q)$ can be transferred to $E_{a,b}(\mathbb{Z}_n)$ using CRT, but then the factorization of n should be provided. However, the usual tangent-and-chord formulas allows us to perform addition of affine points on $E_{a,b}(\mathbb{Z}_n)$, without knowledge of the factorisation of n. In particular, the formula to double an affine point is the following:

$$\boxed{2\#(x,y) = (\lambda^2 - 2x, -\lambda^3 + 3x\lambda - y), \quad \text{where} \quad \lambda = (3x^2 + a)(2y)^{-1}.}$$

To deal with points at infinity, the following addition formulas are used:

$$\boxed{\begin{array}{l} O_m + O_{m'} = O_{m+m'}. \\ (x,y) + O_m = (x - 2ymn, y - (3x^2 + a)mn). \end{array}}$$

2.2.2 Some previous elliptic curve based schemes

Galbraith proposes in [Gal02] an elliptic curve scheme based on the one-way trapdoor function

$$\begin{array}{rcl} \mathcal{X}_Q : \mathbb{Z}_n \times \mathbb{Z}_n & \longrightarrow & E_{a,b}(\mathbb{Z}_{n^2}) \\ (r,m) & \longmapsto & r\#Q + O_m \end{array}$$

where $Q \in E_{a,b}(\mathbb{Z}_{n^2})$ is a fixed point whose order is a big-enough factor of $|E_{a,b}(\mathbb{Z}_n)|$. The semantic security of the scheme $C = \mathcal{X}_Q(r,m)$ is related to the following decisional problem: given an RSA modulus n, an elliptic curve $E_{a,b}(\mathbb{Z}_{n^2})$, a point $Q \in E_{a,b}(\mathbb{Z}_{n^2})$ whose order is a divisor of $|E_{a,b}(\mathbb{Z}_n)|$, and a random point $S \in E_{a,b}(\mathbb{Z}_{n^2})$, determine whether S lies on the subgroup generated by Q. The scheme is only of theoretical interest, since it presents a high computational cost, both in key generation and decryption. Moreover, Galbraith's scheme involves the computation of the multiple $r\#Q$, where r has roughly the same length as n.

Koyama *et al.* propose in [KMOV91] a deterministic elliptic curve RSA based scheme. They use supersingular elliptic curves of type $E_n(0,b)$, and thus avoid the problem of computing $|E_{a,b}(\mathbb{Z}_n)|$, because $|E_n(0,b)| = (p+1)(q+1)$ when $p \equiv q \equiv 2 \bmod 3$. To encrypt a message $m = (x,y) \in \mathbb{Z}_n \times \mathbb{Z}_n$, the following trapdoor one-way function is used:

$$\begin{array}{rcl} \mathrm{KMOV}[n,e] : \mathbb{Z}_n \times \mathbb{Z}_n & \longrightarrow & \mathbb{Z}_n \times \mathbb{Z}_n \\ (x,y) & \longmapsto & e\#(x,y), \end{array}$$

where $e\#(x,y)$ stands for the e-multiple of (x,y) computed on the elliptic curve $E_n(0,b)$, where $b = y^2 - x^3 \bmod n$. Let us observe that the elliptic curve used to perform the computation is determined by the message point. Although it is required that $b \in \mathbb{Z}_n^*$ in

order to be able to perform the computation, this condition is fulfilled with overwhelming probability. For e such that $\gcd(e,(p+1)(q+1))=1$, the trapdoor is

$$d = e^{-1} \bmod \operatorname{lcm}(p+1, q+1),$$

since $d\#(e\#(x,y)) = (x,y)$ on $E_n(0,b)$.

A probabilistic version of KMOV scheme has been proposed in [GMMV03b]. Basically, this scheme is a lifted version of KMOV that works on supersingular elliptic curves over $\mathbb{Z}_{n^2}$. For small values of e, KMOV$[n,e]$ as well as its lifted version are significantly more efficient than Galbraith's scheme, as shown in [GMMV03b].

The optimal efficiency would be achieved using $e=2$, but in this case the map KMOV$[n,2]$ is not bijective (some points have 4 pre-images, others have none). Next we show that this inconvenience can be avoided by restricting the set of points, and using an RSA modulus $n=pq$ such that $p \equiv q \equiv 5 \bmod 12$. In this way, a new trapdoor permutation equivalent to factoring is obtained.

2.2.3 New trapdoor permutations

In the following, the well-known Blum-Williams trapdoor permutation is adapted to the elliptic curve setting.

Point-doubling trapdoor permutation.
As in KMOV scheme, only supersingular curves $E_n(0,b)$ will be considered. Thus, $p \equiv q \equiv 2 \bmod 3$. A new restriction on the prime factors of n must be introduced, in order to avoid the existence of points of order 4.

Observation 68 *If $p \equiv 5 \bmod 12$, then $|E_p(0,b)| \equiv 2 \bmod 4$, and consequently there are no points of order 4 on $E_p(0,b)$. Moreover, there is a unique point of order 2, $(\eta,0)$, where η is the unique cubic root of $-b$. This implies that given a point $P \in E_p(0,b)$, the equation $2\#\bar{P} = 2\#P$ has exactly two solutions: $\bar{P} = P$ and $\bar{P} = P + (\eta,0)$, since the order of the point $\bar{P} - P$ divides 2.*

Now, the elliptic curve analogous to the set of quadratic residues is defined.

Definition 69 *For $n=pq$, and $p \equiv q \equiv 5 \bmod 12$, let*

$$D_n = \{2\#(x,y) \in \mathbb{Z}_n \times \mathbb{Z}_n \mid x \in \mathbb{Z}_n,\ y \in \mathbb{Z}_n^*,\ y^2 - x^3 \in \mathbb{Z}_n^*\},$$

where the double $2\#(x,y)$ is computed on the curve $E_n(0,b)$, with $b = y^2 - x^3$.

We say that $(x, y) \in \mathbb{Z}_n \times \mathbb{Z}_n^*$ is a *double* if belongs to D_n. We will also consider the sets D_p and D_q defined in the same way as D_n but using modulo p and q instead of n. From the CRT, it is clear that $D_n = D_p \times D_q$.

Lemma 70 *If $(u, v) \in D_n$, then $v \in \mathbb{Z}_n^*$.*

Proof: Let $Q = (u, v) \in D_n$. Then, there exists a point $P = (x, y)$ on the same curve such that $Q = 2\#P$ and $y \in \mathbb{Z}_n^*$. Let us suppose that $v = 0 \bmod p$. This implies that $2\#\pi_p(Q) = O$ and then $4\#\pi_p(P) = O$. Since there are no points of order 4 on $E_p(0, b)$, we can assure that $2\#\pi_p(P) = O$. Therefore, $y \equiv 0 \bmod p$, and we obtain a contradiction. ∎

Lemma 71 *$|D_p| = \frac{(p-1)^2}{2}$ and $|D_n| = \frac{(p-1)^2(q-1)^2}{4}$.*

Proof: Let $Q \in E_p(0, b) \cap D_p$, where $b \in \mathbb{Z}_p^*$. From observation 68 it is clear that the equation $2\#P = Q$ has exactly two solutions $P, \bar{P} \in E_p(0, b)$. Since there are $p - 1$ affine points $P = (x, y)$ on $E_p(0, b)$ with $y \in \mathbb{Z}_p^*$, then $|E_p(0, b) \cap D_p| = \frac{p-1}{2}$. By considering the $p - 1$ possible values for b, we obtain the claimed result $|D_p| = \frac{(p-1)^2}{2}$. Finally, $|D_n| = \frac{(p-1)^2(q-1)^2}{4}$ comes from $D_n = D_p \times D_q$. ∎

Proposition 72 *Let $n = pq$, with $p \equiv q \equiv 5 \bmod 12$. Then, the following map is a bijection:*

$$\begin{aligned} \Delta_n : D_n &\longrightarrow D_n \\ (x, y) &\longmapsto 2\#(x, y) \end{aligned}$$

Proof: It suffices to show that Δ_n is injective. Δ_n is well-defined by the definition of D_n and lemma 70. In order to prove that Δ_n is injective, let us consider Q_1 and Q_2 in D_n such that $2\#Q_1 = 2\#Q_2$. On the one hand, this implies that there exist P_1 and P_2 such that $Q_1 = 2\#P_1$ and $Q_2 = 2\#P_2$. On the other hand, P_1, P_2, Q_1 and Q_2 lie on the same curve and $2\#(Q_2 - Q_1) = O$. Thus, $4\#(P_2 - P_1) = O$ which implies $2\#(P_2 - P_1) = O$, since there are no points of order 4 in $E_n(0, b)$, and therefore $Q_1 = Q_2$. ∎

We point out that Δ_n is an elliptic analogous of Blum-Williams function. Before studying the one-wayness of Δ_n we define both the keypair generator and the computational assumption involved.

Definition 73 *Let $(n, p, q) \leftarrow I_{\mathsf{spec}}^{\mathsf{FAC}}(1^\ell)$, where $p, q \leftarrow PRIMES(\ell/2)$ such that $p \equiv q \equiv 5 \bmod 12$ and $n = pq$. The* special congruence factoring assumption *states that for any* PPT *algorithm $\mathcal{A}$*

$$\mathsf{Pr}\left[\mathcal{A}(1^\ell, n) = (p, q) \mid (n, p, q) \leftarrow I_{\mathsf{spec}}^{\mathsf{FAC}}(1^\ell)\right] \in \mathsf{negl}(\ell)\,.$$

where the probability is computed with respect to distribution $I_{\mathsf{spec}}^{\mathsf{FAC}}$ and the coin tosses of $\mathcal{A}$.

Proposition 74 *Δ_n is a length-preserving* TOW *permutation with respect to $I^{\mathsf{FAC}}_{\mathsf{spec}}$ if and only if the* special congruence factoring *assumption holds.*

Proof:

($\Rightarrow$) Let us see how to invert Δ_n efficiently on a point $Q \in D_n$, given the trapdoor information p and q. Since Δ_n is a bijection, there exists a point $P \in D_n$ such that $Q = 2\#P$, but also exist another point $R \in D_n$ such that $P = 2\#R$, that is $Q = 4\#R$. Let us consider two points $T_p = \frac{p+3}{4}\#\pi_p(Q)$ and $T_q = \frac{q+3}{4}\#\pi_q(Q)$. Then, $T_p = (p+3)\#\pi_p(R) = 2\#\pi_p(R) = \pi_p(P)$ and $T_q = (q+3)\#\pi_q(R) = 2\#\pi_q(R) = \pi_q(P)$. Thus, the preimage P of Q can easily be computed from T_p and T_q by the CRT. In fact, a point-halving procedure that works in a more general case can be found in [KMOV91].

($\Leftarrow$) Now we show a reduction from the one-wayness of Δ_n to the problem of factoring n. To do this, take a random pair $\bar{P} \leftarrow (\bar{x}, \bar{y}) \in \mathbb{Z}_n \times \mathbb{Z}_n^*$ and compute $Q = 2\#\bar{P}$, which is uniformly distributed on D_n. Observe that $\pi_q(\bar{P}) \in D_q$ and $\pi_p(\bar{P}) \notin D_p$ with probability $1/4$. Let us consider that this is in fact the case. Since $Q \in D_n$, there exists a point $P = (x, y) \in D_n$ such that $Q = 2\#P$. Let us consider an algorithm $\mathcal{A}$ such that on input (n, Q) returns P with probability ε. If $\mathcal{A}$ succeeds then $2\#\bar{P} = 2\#P$. We can assure now that $\pi_q(\bar{P}) = \pi_q(P)$ and $\bar{x} \neq x \bmod p$ (note that, if $\bar{x} = x \bmod p$, then $\pi_p(\bar{P}) = \pm\pi_p(P)$ and $\pi_p(\bar{P}) \in D_p$, which is a contradiction). Finally, $\gcd(\bar{x} - x, n) = p$. By considering the opposite case, $\pi_p(\bar{P}) \in D_p$ but $\pi_q(\bar{P}) \notin D_q$, it is straightforward to show that this procedure gives a non-trivial factor of n with probability $\varepsilon/2$. ∎

Lifted trapdoor bijection.

Next, a lifted version of the map Δ_n is presented. The technique used here is somewhat related to the one used in [GMMV03b]. The following useful property allows us to lift a point $P_0 \in E_n(0, b_0)$ to a special point P on each curve $E_{n^2}(0, b)$ such that $b \equiv b_0 \bmod n$.

Property 75 *Let $b \in \mathbb{Z}_{n^2}^*$ and $P = (x_0, y_0) \in E_n(0, b \bmod n)$, with $y_0 \in \mathbb{Z}_n^*$. Then, there exists a unique point $(x_0, y) \in E_{n^2}(0, b)$ such that $y \equiv y_0 \bmod n$.*

Proof: Let $y = y_0 + \gamma n \in \mathbb{Z}_{n^2}^*$, where $\gamma \in \mathbb{Z}_n$. Then, (x_0, y) belongs to $E_{n^2}(0, b)$ if and only if

$$\gamma = \frac{x_0^3 - y_0^2 + b}{n}(2y_0)^{-1} \bmod n.$$

∎

Let $n = pq$, with $p \equiv q \equiv 5 \bmod 12$, and let us consider the following sets:

$$\Omega_n = \{(x, y) \in \mathbb{Z}_{n^2} \times \mathbb{Z}_{n^2}^* \mid \pi_n(x, y) \in D_n\}, \quad \omega_n = \{(x, y) \in \Omega_n \mid x < n\},$$

and the function

$$\begin{aligned} \psi_n : \omega_n \times \mathbb{Z}_n &\longrightarrow \Omega_n \\ (x, y, m) &\longrightarrow 2\#P + O_m \end{aligned}$$

where $P = (x, y)$, and the double as well as the addition are performed on $E_{n^2}(0, b)$, with $b = y^2 - x^3 \bmod n^2$.

Lemma 76 *If $p \equiv q \equiv 5 \bmod 12$, then the map ψ_n is well defined and bijective.*

Proof: The map ψ_n is well-defined since $\psi_n(x, y, m)$ is always in Ω_n. This is implied by the definition of Ω_n, since $\psi_n(x, y, m) \in \Omega_n$ if and only if $\pi_n(\psi_n(x, y, m)) \in D_n$. As $(x, y) \in \omega_m$, $\pi_n(x, y) \in D_n$ and then $\pi_n(\psi_n(x, y, m)) = \pi_n(2\#(x, y)) = 2\#\pi_n(x, y) \in D_n$.

In order to show that ψ_n is injective, let us suppose $\psi_n(x, y, m) = \psi_n(x', y', m')$ for some $(x, y), (x', y') \in \omega_n$ and $m, m' \in \mathbb{Z}_n$. Reducing this equality modulo n, we obtain $2\#\pi_n(x, y) = 2\#\pi_n(x', y')$. By the injectivity of Δ_n and from the fact that $\pi_n(x, y)$ and $\pi_n(x, y)$ are points in D_n we deduce $\pi_n(x, y) = \pi_n(x', y')$.

Now, taking into account that (x, y), (x', y') belong to the same curve $E_{n^2}(0, b)$, and that $0 \leq x, x' < n$, we use Property 75 to deduce $(x, y) = (x', y')$. From this, it is easy to see that $O_m = O_{m'}$, so $m = m'$.

Finally, let us show that ψ_n is surjective. Let $C = (u, v) \in \Omega_n$ and $b = v^2 - u^3 \bmod n^2$. Then there exists $P_0 = (x_0, y_0) \in D_n$ such that $\pi_n(u, v) = 2\#P_0$. Let $P = (x_0, y)$ be the point on $E_{n^2}(0, b)$ given in Property 75. Clearly, $P \in \omega_n$ and $2\#P - C$ is a point at infinity, say O_m. Then, $C = \psi_n(x_0, y, m)$. ∎

Proposition 77 *ψ_n is a length-preserving* TOW *permutation with respect to $I^{\mathsf{FAC}}_{\mathsf{spec}}$ if and only if the* special congruence factoring *assumption holds.*

Proof:

(⇒) Let us see, given the trapdoor information, p and q, how to invert ψ_n efficiently on a point $C = (u, v) \in \Omega_n$. Let $b = v^2 - u^3 \bmod n^2$. Compute $Q_0 = \pi_n(C)$, that is a point in D_n, and let $P_0 \in D_n$ such that $Q_0 = 2\#P_0$. The point P_0 can be efficiently computed by using the procedure for inverting Δ_n described in the proof of Proposition 74. Then, let $P = (x, y) \in E_{n^2}(0, b)$ the point given in Property 75 computed from P_0. Clearly, $P \in \omega_n$ and $C - 2\#P$ is a point at infinity, say O_m. Then, $C = \psi_n(x, y, m)$.

(⇐) Now we show a reduction from the problem of factoring n to the one-wayness of ψ_n. As in the proof of Proposition 74, take a random pair $(\bar{x}, \bar{y}) \leftarrow \mathbb{Z}_n \times \mathbb{Z}_n^*$ and compute $Q_0 = (u_0, v_0) = 2\#(\bar{x}, \bar{y})$. Now randomly lift Q_0 obtaining $C = (u_0 +$

$\mu n, v_0 + \nu n)$, where μ and ν are randomly selected in $\mathbb{Z}_n$. Note that C is uniformly distributed on Ω_n. Let us consider an algorithm $\mathcal{A}$ such that on input (n, C) returns $P = (x, y) \in \omega_n$ and $m \in \mathbb{Z}_n$ such that $C = \psi_n(x, y, m)$, with probability ε. If $\mathcal{A}$ succeeds, then $\Delta_n(\pi_n(x, y)) = 2\#\pi_n(x, y) = \pi_n(C) = Q_0$. Thus, by following the same steps as in the proof of Proposition 74, a nontrivial factor of n is found with probability $\varepsilon/2$. ∎

2.2.4 Lifted-Rabin Elliptic Curve Scheme

Based on the previous TOW permutation, we present an elliptic curve cryptosystem (ECC) over the ring $\mathbb{Z}_{n^2}$ which is semantically secure against passive adversaries under a new decisional assumption, and has the fastest encryption and the strongest one-way security among the known ECC, in the standard model.

Key generation. Given a security parameter 1^ℓ, let $(\mathsf{pk}, \mathsf{sk}) \leftarrow I^{\mathsf{FAC}}_{\mathsf{spec}}(1^\ell)$, that is, $\mathsf{pk} = (n)$ and $\mathsf{sk} = (n, p, q)$ with $p \equiv q \equiv 5 \bmod 12$.

Encryption. To encrypt a message $m \in \mathbb{Z}_n$ we choose at random $z \leftarrow \mathbb{Z}_n$ and $t \leftarrow \mathbb{Z}_n^*$ and let $b_0 = t^2 - z^3 \in \mathbb{Z}_n^*$. This choice determines an elliptic curve $E_n(0, b_0)$ and a point $Q = (z, t)$ on it. Let $P_0 = (x_0, y_0) = 2\#Q$ and γ chosen at random in $\mathbb{Z}_n$, and compute $y = y_0 + \gamma n$. Then $P = (x_0, y)$ is a random point in ω_n. The encryption of the message $m \in \mathbb{Z}_n$ is $C = \psi_n(x_0, y, m)$.

Decryption. To recover the message m from the ciphertext $C = (u, v) = \psi_n(x, y, m)$, the randomness $(x, y) \in \omega_n$ is firstly computed and, afterwards, m is easily obtained from $O_m = C - 2\#(x, y)$. We recall the steps to obtain (x, y) from C. Let us compute $\pi_n(x, y)$ by inverting Δ_n on $\pi_n(C)$ (using the CRT). Next, compute $(x, y) \in E_{n^2}(0, b)$, where $b = v^2 - u^3 \bmod n^2$, by using Property 75.

In the following, the security of this scheme is analyzed.

One-wayness

The following lemma enables us to compute, with overwhelming probability, a rational function of the coordinates of a point $P_0 \in D_n$, given two special lifted points Q_1 and Q_2 such that $\pi_n(Q_1) = \pi_n(Q_2) = 2\#P_0$.

Lemma 78 *Let $Q_1 = (u_1, v_1) = 2\#P_1$ and $Q_2 = (u_2, v_2) = 2\#P_2$ where P_1 and P_2 are different points in ω_n such that $\pi_n(P_1) = \pi_n(P_2)$. Let $b_1 = v_1^2 - u_1^3 \bmod n^2$ and $b_2 = v_2^2 - u_2^3 \bmod n^2$. Let $(x_0, y_0) = \pi_n(P_1)$. Then*

$$9\alpha \left(\frac{x_0}{y_0}\right)^4 = -4\beta \bmod n$$

where $\alpha = (b_2 - b_1)/n$ *and* $\beta = (u_2 - u_1)/n$.

Proof: Since $P_1, P_2 \in \omega_n$, we can write $P_1 = (x_0, y_1)$ and $P_2 = (x_0, y_2)$, where $y_1 \equiv y_2 \equiv y_0 \bmod n$ and $x_0 < n$. Observe that both points lie on different curves. Indeed, Q_1 and P_1 are in $E_n(0, b_1)$ while Q_2 and P_2 are in $E_n(0, b_2)$. Since $b_1 \equiv b_2 \bmod n$, $\alpha = (b_2 - b_1)/n$ is well defined.

By using the doubling formula, we obtain

$$\begin{aligned} u_1 &= \left(\frac{3x_0^2}{2y_1}\right)^2 - 2x_0 = \frac{9x_0^4}{4(x_0^3 + b_1)} - 2x_0 \bmod n^2 \\ u_2 &= \left(\frac{3x_0^2}{2y_2}\right)^2 - 2x_0 = \frac{9x_0^4}{4(x_0^3 + b_2)} - 2x_0 \bmod n^2 \end{aligned}$$

and then,

$$u_2 - u_1 = \frac{9x_0^4}{4(x_0^3 + b_2)} - \frac{9x_0^4}{4(x_0^3 + b_1)} = \frac{9x_0^4(b_1 - b_2)}{4(x_0^3 + b_1)(x_0^3 + b_2)} = -\frac{9}{4}\frac{x_0^4}{y_1^2 y_2^2}\alpha n \bmod n^2$$

so

$$\beta = \frac{u_2 - u_1}{n} = -\frac{9}{4}\left(\frac{x_0}{y_0}\right)^4 \alpha \bmod n$$

■

Note that if Q_1 and Q_2 are chosen at random (but satisfying the conditions in Lemma 78), then $\alpha \in \mathbb{Z}_n^*$ with overwhelming probability.

From this lemma, given a random modulus n, we can exploit an adversary $\mathcal{A}$ against the one-wayness of the proposed scheme to build two such points Q_1 and Q_2, and to derive efficiently a nontrivial factor of n.

Proposition 79 *The one-wayness of the proposed scheme is equivalent to the* special congruence factoring *assumption.*

Proof: Let $\mathcal{A}$ be an adversary trying to break the one-wayness of the proposed cryptosystem. Let us consider the following probability

$$\mathsf{Succ}_{\mathcal{A}}^{\mathsf{OW}}(\ell) = \mathsf{Pr}\left[\mathcal{A}(n, \psi_n(x, y, m)) = m \mid (n, p, q) \leftarrow I_{\mathsf{spec}}^{\mathsf{FAC}}(1^\ell);\ (x, y) \leftarrow \omega_n;\ m \leftarrow \mathbb{Z}_n\right]$$

The following algorithm $\mathcal{B}$ can be used to obtain a nontrivial factor of n.

$\mathcal{B}(n)$
1 $\bar{x}_0 \leftarrow \mathbb{Z}_n;\ \bar{y}_0 \leftarrow \mathbb{Z}_n;\ b_0 = \bar{y}_0^2 - \bar{x}_0^3 \bmod n$

2 if $\gcd(\bar{y}_0, n) \neq 1$ return $\gcd(\bar{y}_0, n)$
3 if $\gcd(b_0, n) \neq 1$ return $\gcd(b_0, n)$
4 $(u_0, v_0) = 2\#(\bar{x}_0, \bar{y}_0)$, computed in $E_n(0, b_0)$
5 $\gamma_1 \leftarrow \mathbb{Z}_n;\ \delta_1 \leftarrow \mathbb{Z}_n;\ C_1 = (u_0 + \gamma_1 n, v_0 + \delta_1 n)$
6 $m_1 = \mathcal{A}(n, C_1);\ (u_1, v_1) = C_1 - O_{m_1}$
7 $\gamma_2 \leftarrow \mathbb{Z}_n;\ \delta_2 \leftarrow \mathbb{Z}_n;\ C_2 = (u_0 + \gamma_2 n, v_0 + \delta_2 n)$
8 $m_2 = \mathcal{A}(n, C_2);\ (u_2, v_2) = C_2 - O_{m_2}$
9 $\alpha = (v_2^2 - u_2^3 - v_1^2 + u_1^3)/n$
10 if $\gcd(\alpha, n) \neq 1$ return $\gcd(\alpha, n)$
11 $\beta = (u_2 - u_1)/n$
12 return $\gcd\left(\frac{\bar{x}_0^4}{\bar{y}_0^4} + \frac{4\beta}{9\alpha}, n\right)$

At steps 1 to 4 of the algorithm, a random point $Q_0 = (u_0, v_0) \in D_n$ is built. Next, points $Q_1 = (u_1, v_1)$ and $Q_2 = (u_2, v_2)$ are built by calling $\mathcal{A}$ twice using two randomly lifted points C_1 and C_2 coming from the same point Q_0.

If $\mathcal{A}$ succeeds in the first call, at step 6, then Q_1 can be written as $Q_1 = 2\#P_1$ where $P_1 \in \omega_n$. This is a consequence of the bijectivity of ψ_n, since $C_1 \in \Omega_n$, and then there exists a unique $P_1 \in \omega_n$ and a unique $m_1 \in \mathbb{Z}_n$ such that $C_1 = \psi_n(P_1, m_1)$. The same occurs with $Q_2 = 2\#P_2$, if $\mathcal{A}$ succeeds in the second call.

Let us consider the case that $\mathcal{A}$ succeeds in both calls. Note that $Q_0 = \pi_n(C_1) = \pi_n(C_2)$ and $Q_0 = 2\#\pi_n(P_1) = 2\#\pi_n(P_2)$. But there is only one point in D_n whose double is Q_0. Thus, $\pi_n(P_1) = \pi_n(P_2)$. Let $P_0 = (x_0, y_0) = \pi_n(P_1) = \pi_n(P_2)$. Since Q_1 and Q_2 fulfil the conditions in the previous lemma, then

$$\left(\frac{x_0}{y_0}\right)^4 = -\frac{4\beta}{9\alpha} \bmod n$$

if $\alpha \in \mathbb{Z}_n^*$.

On the other hand, $Q_0 = 2\#(\bar{x}_0, \bar{y}_0) = 2\#P_0$. Observe that $P_0 \in D_n$ but $\bar{P}_0 = (\bar{x}_0, \bar{y}_0)$ is chosen at random. By using the Chinese Reminder Theorem, $\pi_p(\bar{P}_0) = \pi_p(P_0)$ with probability $1/2$, and independently $\pi_q(\bar{P}_0) = \pi_q(P_0)$ with probability $1/2$. So, with probability $1/4$, $\pi_q(\bar{P}_0) = \pi_q(P_0)$ but $\pi_p(\bar{P}_0) \neq \pi_p(P_0)$. The last inequality implies $\bar{x}_0 \neq x_0 \bmod p$. To see this, let us suppose that $\bar{x}_0 = x_0 \bmod p$. Then, $\pi_p(\bar{P}_0) = -\pi_p(P_0)$. From $2\#\bar{P}_0 = 2\#P_0$ we deduce that $4\#\pi_p(\bar{P}_0) = O$. Since there are no points with order 4 on $E_p(0, b_0 \bmod p)$ then $2\#\pi_p(\bar{P}_0) = O$ and consequently $\bar{y}_0 \equiv 0 \bmod p$. But, this is not possible due to step 2 in the algorithm.

Except for a negligible fraction of the values of $(\bar{x}_0, \bar{y}_0)$, it can also be shown that[1]

$$\left(\frac{\bar{x}_0}{\bar{y}_0}\right)^4 \neq \left(\frac{x_0}{y_0}\right)^4 \bmod p.$$

Then, by using Lemma 78,

$$\gcd\left(\frac{\bar{x}_0^4}{\bar{y}_0^4} + \frac{4\beta}{9\alpha}, n\right) = p.$$

By considering the other case, $\pi_p(\bar{P}_0) = \pi_p(P_0)$ but $\pi_q(\bar{P}_0) \neq \pi_q(P_0)$, the previous $\gcd$ expression leads to the other nontrivial factor of n.

Finally, except for a negligible function of ℓ (due to the technical steps 2, 3 and 10, and the anomalous values of $(\bar{x}_0, \bar{y}_0)$) the success probability

$$\mathsf{Succ}_{\mathcal{B}}^{\mathsf{FACT}}(\ell) = \mathsf{Pr}\left[\mathcal{B}(n) \in \{p, q\} \mid (n, p, q) \leftarrow I_{\mathsf{spec}}^{\mathsf{FAC}}(1^\ell)\right]$$

is one half the probability that $\mathcal{A}$ is successful in both calls. Notice that these two calls are not independent, since they share the same values of n and Q_0. However, by using Lemma 13 in Section 1.1.2 with algorithm $\mathcal{A}$, predicate $P =$ "$\mathcal{A}$ succeeds" and map $f(n, C) = (n, \pi_n(C))$, the following inequality is obtained

$$\mathsf{Succ}_{\mathcal{B}}^{\mathsf{FACT}}(\ell) \geq \frac{1}{2}\left(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{OW}}(\ell)\right)^2.$$

■

Semantic security

The scheme is semantically secure under the following assumption:

Decisional Small-x Double assumption (DSD assumption).
The following probability distributions are polynomially indistinguishable

$$\begin{aligned} D_{\text{double}} &= (n, 2\#(x, y)) \quad \text{where } (n, p, q) \leftarrow I_{\mathsf{spec}}^{\mathsf{FAC}}(1^\ell),\ (x, y) \leftarrow \omega_n \\ D_{\text{random}} &= (n, (x', y')) \quad \text{where } (n, p, q) \leftarrow I_{\mathsf{spec}}^{\mathsf{FAC}}(1^\ell),\ (x', y') \leftarrow \Omega_n. \end{aligned}$$

Proposition 80 *The proposed scheme is semantically secure if and only if the DSM assumption holds.*

[1] The exception are points $(\bar{x}_0, \bar{y}_0)$ such that $\bar{x}_0 \bmod p$ is a root of a certain polynomial of degree 8. However, by making some cumbersome calculations, it can be shown that if $p \equiv 1 \bmod 8$ then there are no exceptional points, otherwise, i.e. $p \equiv 5 \bmod 8$, there are only $p - 1$ exceptional points (modulo p), that is, only a fraction $1/p$. (See appendix A for details.)

Proof: Semantic security is equivalent to indistinguishability of encryptions, so we have to prove that for all $m_0 \in \mathbb{Z}_n$, the distributions

$$\begin{aligned} D_0 &= (n, \psi_n(x, y, m_0)) \quad \text{where } (n, p, q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{spec}}(1^\ell),\ (x, y) \leftarrow \omega_n\,, \quad \text{and} \\ D &= (n, \psi_n(x, y, m)) \quad \text{where } (n, p, q) \leftarrow I^{\mathsf{FAC}}_{\mathsf{spec}}(1^\ell),\ (x, y) \leftarrow \omega_n,\ m \leftarrow \mathbb{Z}_n. \end{aligned}$$

are polynomially indistinguishable. From the definition of sum of an affine point and a point at infinity given at the end of Section 2.2.1, it is easy to see that the map

$$\begin{aligned} \Omega_n &\longrightarrow \Omega_n \\ P &\longmapsto P - O_{m_0} \end{aligned}$$

is a polynomial time bijection. Then, $D_0 \approx D$ is equivalent to

$$(n, 2\#(x, y)) \approx (n, 2\#(x, y) + O_{m'}), \quad \text{with } (x, y) \leftarrow \omega_n,\ m' \leftarrow \mathbb{Z}_n\,.$$

Note that the distribution on the left side is D_{double}. Besides, since $2\#(x, y) + O_{m'} = \psi_n(x, y, m')$, and ψ_n is a bijection, then D and D_{random} are identical distributions. ■

Hardness of the Small-x Double Problems

In this subsection we argue why one should be confident about the hardness of the new decisional problem presented in this paper.

According to the formula for computing the double of a point on an elliptic curve $E_{n^2}(0, b)$ (see end of Section 2.2.1), given $(u, v) = 2\#(x_1, y_1)$, x_1 is a root of the univariate polynomial $R(x) = x^4 + 4x^3u - 8bx + 4bu \in \mathbb{Z}_{n^2}[x]$. Then, DSD assumption is related to the difficulty of deciding if the polynomial $R(x)$ has a root smaller than n.

Similarly, the semantic security of other related cryptosystems (such as RSA and Rabin-Paillier schemes) is related to the difficulty of deciding if a certain polynomial has a root smaller than n. The best known way to attack the above decisional problems is to solve their computational versions by using Coppersmith's algorithm as stated in Theorem 57. This result ensures that one can efficiently compute all roots x_1 of a polynomial $P(x) \in \mathbb{Z}_K[x]$ with degree d such that $|x_1| < K^{1/d}$. Up to now, no improvement on this bound has been made. The result by Coppersmith implies we can only find the roots $|x_1| < (n^2)^{1/4} = n^{1/2}$ of the polynomial $R(x)$, which does not affect the validity of DSD assumption.

2.2.5 Efficiency analysis

Now we study the encryption cost of our scheme. Since operations modulo a large number are involved, we neglect the cost of performing additions, multiplications and

divisions by small integers. We will express the cost in terms of multiplications $\bmod n$, because modular inverses can be computed within a constant number of modular multiplications.

Generating $(x, y) \in \omega_n$: 5 multiplications modulo n, 1 inverse modulo n, and 1 n-length integer multiplication.
Computing $2\#(x, y)$: 5 multiplications modulo n^2, 1 inverse modulo n^2.
Adding O_m : 3 multiplications modulo n, 2 n-length integer multiplication.

We point out that $a^{-1} \bmod n^2$ can be obtained by computing $a^{-1} \bmod n$ and then performing two multiplications modulo n^2. Let c be the number of multiplications modulo n needed to compute $a^{-1} \bmod n$. Since the cost of multiplying two numbers mod n^2 is roughly the cost of 4 multiplications modulo n, we deduce that $a^{-1} \bmod n^2$ can be computed in $8 + c$ multiplications modulo n. Practical implementations suggest that the value $c = 8$ can be taken (see [Bre98]). Then, since the n-length integer multiplication cost is bounded by the cost of a multiplication modulo n, the encryption cost of our scheme is 55 multiplications modulo n.

Next, we will compare the efficiency of our scheme with the well-known El Gamal ECC scheme. We assume that El Gamal ECC is performed over $\mathbb{F}_q$, where q is 160 bits long, and our scheme is performed over $\mathbb{Z}^*_{n^2}$, where n is 1024 bits long, which still are the more usual values. We will express both encryption costs in terms of multiplications modulo n.

In El Gamal ECC the most time consuming operation is the computation of two multiples $r\#P$ and $ra\#P$, where r is a random integer whose size is roughly the same as the modulus p, and a is a fixed integer. Then, using the $double\ and\ add$ algorithm, the computation of these two multiples requires on average k additions of points and $2k$ doublings, where k is the bit length of r. Assuming that a point addition or doubling requires about 12 modular multiplications, then El Gamal ECC would take approximately $3 \cdot 160 \cdot 12$ multiplications modulo q. Since the time needed to perform a modular multiplication is quadratic in the size of the modulus, the ratio between the time of a multiplication modulo q and a multiplication modulo n is $\frac{160^2}{1024^2}$. It follows that the encryption time of El Gamal ECC would be equivalent to 159 multiplications modulo n, which is almost three times the encryption cost of our scheme.

Thus our cryptosystem is the provably secure IND-CPA ECC in the standard model with the fastest encryption procedure to the best of our knowledge. In fact, not even El Gamal ECC is provably secure, since its one-wayness is equivalent to solving ECDH, but not to solving ECDL.

The key generation of the proposed cryptosystem is faster than generating an RSA key, since only the modulus is needed. Regarding decryption, the main cost is due to the

computation of $\frac{p+3}{4}\#P \in E_p(0,b)$, and $\frac{q+3}{4}\#P \in E_q(0,b)$, from $P \in E_n(0,b)$ which is almost the same as in the other existing ECC over $\mathbb{Z}_{n^2}$. Therefore, the decryption procedure has a very high computational cost compared to El Gamal ECC, so it is unlikely that our scheme could compete with EL Gamal ECC from a global point of view.

Chapter 3

Semantically Secure Encryption Schemes against Adaptive Adversaries

In this chapter we revisit some of the most relevant asymmetric schemes with semantic security against adaptive adversaries appeared in the literature. All of them use the Random Oracle heuristic, except for the ACE key encapsulation mechanism, which is based on [CS98].

In the first place, we identify some ambiguities in the security proof of the popular generic conversion by Fujisaki and Okamoto [FO99], from which false conclusions can be drawn. In doing so, we continue with the careful revision of the provable security techniques initiated by Shoup in [Sho01], where he questioned some properties of the OAEP scheme [BR95] which were accepted without proof. We modify the Fujisaki-Okamoto transformation to remove the ambiguities detected, and to prove that the resulting conversion is secure in the Random Oracle Model (ROM). The security proof is phrased using current widely accepted proof techniques.

In the second place, we re-evaluate the elliptic curve based KEMs presented to become standards, which are called ACE-KEM, ECIES-KEM and PSEC-KEM. We analyse both their security properties and performance when elliptic curves with efficiently computable bilinear maps (hereafter referred as *pairing curves*) are used. It turns out that these KEMs present a very tight security reduction to the ECDH problem over pairing curves in the ROM; moreover, one can even relate their security to the ECDL problem in certain pairing curves with a small security loss. The key point is that the ECDDH decisional problem is solvable in these groups. It is also shown that ECIES-KEM arises as the best option among these KEMs when pairing curves are used. This is remarkable, since NESSIE [Nes03] didn't select ECIES-KEM as a candidate to standardization.

3.1 Fujisaki-Okamoto Hybrid Encryption Revisited

Regarding IND-CCA public key encryption schemes, several powerful generic constructions have been designed [FO99, Poi00, OP01b, CHJ+02a], providing practical IND-CCA schemes by combining asymmetric and symmetric schemes, with weak security properties, in the idealized Random Oracle Model. The proposal by Fujisaki and Okamoto is by far the most known conversion.

Among these constructions, [OP01b, CHJ+02a] present a better security reduction than [FO99, Poi00]. This is mainly due to the use of the *plaintext checking oracle* introduced in [OP01a], which enables to use the concept of indistinguishability against plaintext checking attacks IND-PCA (see Definition 28). The disadvantage 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 even more subtle than it was expected. The first example of this fact was the claim by Shoup [Sho01] against the widely believed IND-CCA security of OAEP when applied to a trapdoor permutation. From this and other findings (see [Ste03] for a nice account), we are aware that there are ambiguities and misconceptions in the security model, which can lead to false claims.

We aim at revisiting 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 [OU98], known as EPOC-2 [EPO], has found practical attacks that lead to a total break [JQY01, Den02a, ST02]. The most serious flaw was found in [JQY01], where the secret key was recovered in the IND-CCA game itself. The authors of [JQY01] 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 especification of EPOC-2, an attacker could obtain vital information about the system from those ciphertexts. The other attacks mentioned above ([Den02a, ST02]), belong to the *side-channel attacks* category. They 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 EPOC authors in [JQY01] about FO conversion. Then we show that some ambiguities still remain in the proof of security, with the outcome that the security result claimed in [FO99] 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.

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.

3.1.1 Easy verifiable functions

Let X, Y, Z be set families, $f : X \times Y \to Z$ a Trapdoor Partial One-Way (TPOW) function with respect to a keypair generator I and g its partial inverse (see Definition 16). Then, a probabilistic one-way cryptosystem $\mathsf{PKE}^f = (\mathsf{PKE.KeyGen}^f, \mathsf{PKE.Enc}^f, \mathsf{PKE.Dec}^f)$ is obtained from f in the following way: the keys

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}^f(1^\ell)$$

are generated by using the sampling algorithm for I; the ciphertext for a message $x \in X_{\mathsf{pk}}$ with randomness $y \leftarrow Y_{\mathsf{pk}}$ is $c = \mathsf{PKE.Enc}^f(\mathsf{pk}, x) = f_{\mathsf{pk}}(x, y)$ and a valid ciphertext $z \in Z_{\mathsf{pk}}$ is decrypted by means of $\mathsf{PKE.Dec}^f(\mathsf{sk}, c) = g_{\mathsf{sk}}(c)$. Note that we are implicitly assuming that Y is samplable.

New kinds of attacks and computational problems have been introduced and several applications found in the context of probabilistic cryptosystems (cf [OP01a, OP01b]). 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 81 *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* $\mathsf{pk} \in PK$*,* $x \in X_{\mathsf{pk}}$ *and* $z \in Z_{\mathsf{pk}}$*,* $\mathcal{O}_{\mathsf{PC}}$ *answers 1 if there exists* $y \in Y_{\mathsf{pk}}$ *such that* $f_{\mathsf{pk}}(x, y) = z$*, 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 82 *A* TPOW *function family f is* Partial One-Way against Plaintext Checking Attacks (TPOW-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 [Poi00], 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 83 *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 84 *If the map family f is easy verifiable, then it is* TPOW-PCA.

3.1.2 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 $f_{\mathsf{pk}}(x, y) = (\tilde{f}_{\mathsf{pk}}(x), y)$, 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 TPOW-PCA on a gap problem, which is a stronger assumption (cf [OP01a, OP01b]).

A non-trivial example of an easy verifiable function is the RSA-Paillier trapdoor permutation defined in [CGHN01]. A generalization of that function is presented below.

Easy verifiable function families from RSA-Paillier

Let $(n, e, p, q, d) \leftarrow I^{\mathsf{RSA}}_{\mathsf{prac}}(1^\ell)$. For any integer $r > 1$ with size polynomial in ℓ, consider the subset $\Omega_{n,r} \subset \mathbb{Z}_{nr}$ defined as $\Omega_{n,r} = \mathbb{Z}^\star_n + n\mathbb{Z}_r$. Then, the function family

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

turns out to be a trapdoor permutation family, for $\mathsf{pk} = (n, r, e)$ and $\mathsf{sk} = (p, q, r, d)$, 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. Indeed, 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 obtain $x_0^e = x_1^e \bmod n$, 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.

Proposition 85 *The partial one-wayness of the bijection family $f_{n,r,e}$ is tightly equivalent to the* practical RSA *assumption.*

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.

$$\mathsf{Pr}\left[\mathcal{A}(n, r, e, x^e + ny \bmod nr) = x \mid (n, e, p, q, d) \leftarrow I_{\mathsf{prac}}^{\mathsf{RSA}}(1^\ell);\ x \leftarrow \mathbb{Z}_n^\star;\ y \leftarrow \mathbb{Z}_r\right] = \varepsilon$$

The following PPT algorithm, $\mathcal{B}$, can be used to invert the $\mathrm{RSA}[n, e]$ 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, $\mathsf{Pr}\left[\mathcal{B}(n, e, x^e \bmod n) = x \mid (n, e, p, q, d) \leftarrow I_{\mathsf{prac}}^{\mathsf{RSA}}(1^\ell);\ x \leftarrow \mathbb{Z}_n^\star\right] \geq \varepsilon$.

$\Leftarrow$) Trivial. ∎

Proposition 86 *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. ∎

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 [Men93].

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

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

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

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

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

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 $\mathsf{pk} = (E, G, P, p, W)$ and $\mathsf{sk} = (E, G, P, p, W, s)$.

The function f_W is clearly injective. To show the bijectivity of f_W it suffices to compute the preimage (x, y) of any $(Q, r) \in (G \setminus \{\mathcal{O}\}) \times \mathbb{F}_q^\star$ 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 $(\mathsf{sk}, \mathsf{pk}) \leftarrow I^{\mathbf{EC}}_{\mathsf{BilMap}}$ be the probability distribution on $PK^{\mathbf{EC}} \times SK^{\mathbf{EC}}$ induced by the algorithm generating elliptic curve group descriptions with the following properties:

- q is a prime (power) with length polynomial in ℓ, such that the characteristic of $\mathbb{F}_q$ is different from 2 and 3.
- $a, b \in \mathbb{F}_q$ such that b is not a quadratic residue in $\mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0$.
- q is a prime with length ℓ and P is a point of order p on the elliptic curve $E(\mathbb{F}_q)$, defined by the Weierstrass equation $y^2 = x^3 + ax + b$.

- There is a unique subgroup G of order p in $E(\mathbb{F}_q)$, 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'.

Proposition 87 *The partial one-wayness of the bijection family f_W is tightly equivalent to the hardness of the* ECDH *problem with respect to the probability distribution $I^{\mathbf{EC}}_{\mathsf{BilMap}}$.*

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

$$\mathsf{Pr}\left[\mathcal{A}(E,G,P,W,yP,(yW)_x x) = x \;\middle|\; \begin{array}{l}(E,G,P,p,W,s) \leftarrow I^{\mathbf{EC}}_{\mathsf{BilMap}}(1^\ell) \\ x \leftarrow \mathbb{F}_q^\star;\ y \leftarrow \mathbb{Z}_p^\star\end{array}\right] = \varepsilon$$

The following PPT algorithm $\mathcal{B}$, can be used to solve ECDH 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:

$\mathcal{B}(E,G,P,Q,W)$
1 $r \leftarrow \mathbb{F}_q^\star$
2 $x \leftarrow \mathcal{A}(E,G,P,W,Q,r)$
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 $\mathbb{F}_q$. Then,

$$\mathsf{Pr}\left[\mathcal{B}(E,G,P,yP,W) = yW \mid (E,G,P,p,W,s) \leftarrow I^{\mathbf{EC}}_{\mathsf{BilMap}}(1^\ell);\ y \leftarrow \mathbb{Z}_p^\star\right] \geq \varepsilon,$$

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 88 *The bijection family f_W is easy verifiable.*

Proof: The plaintext checking algorithm works as follows. On input (E, G, P, p, W, x, Q, r), firstly verify if $x, r \in \mathbb{F}_q^\star$ and $Q \in G$, i.e. pqQ 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 $y \in \mathbb{Z}_p^\star$ 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 $T = -yW$. ∎

3.1.3 Symmetric encryption

In the sequel, we introduce a definition for symmetric key encryption different from that given in Section 1.2.1. Our aim is to be as close as possible to the tools used in [FO99]. The main difference is that in this new definition the scheme has a restricted message space, and that a certain relation is required between encryptions keys and pairs of plaintext-ciphertext.

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 a 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 [FO99], if any couple of PPT algorithms $\mathcal{A}^{\mathsf{IND-SYM}}(\mathcal{E}^{sym}) = (\mathcal{A}_1, \mathcal{A}_2)$ (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 $\mathcal{A}^{\mathsf{IND-SYM}}(\mathcal{E}^{sym})$,

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

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

Note that this is a weak concept of security, which considers a passive adversary, but it is all we require to build a hybrid cryptosystem.

3.1.4 Revisiting Fujisaki-Okamoto hybrid scheme

The transformation introduced in [FO99] from weak symmetric and asymmetric schemes into an IND-CCA hybrid encryption scheme is revisited below.

The original construction

Let $\mathsf{PKE}^f = (\mathsf{PKE.KeyGen}^f, \mathsf{PKE.Enc}^f, \mathsf{PKE.Dec}^f)$ 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 $\mathsf{HE}^{\mathbf{FO}} = (\mathsf{HE.KeyGen}^{\mathbf{FO}}, \mathsf{HE.Enc}^{\mathbf{FO}}, \mathsf{HE.Dec}^{\mathbf{FO}})$, proposed by Fujisaki and Okamoto, works as follows.

Key generation. The public and secret keys are generated as in $\mathsf{PKE.KeyGen}^f$.

Encryption. The ciphertext for a message $m \in M_\ell$ is $c = (f_{\mathsf{pk}}(x, y), \mathsf{Enc}^{sym}_{G(x)}(m))$, 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 $x = g_{\mathsf{sk}}(c_1)$. Then, compute $m = \mathsf{Dec}^{sym}_{G(x)}(c_2)$ and return m if $c_1 = f_{\mathsf{pk}}(x, H(x, m))$. Otherwise, return reject. If it is not possible to compute $g_{\mathsf{sk}}(c_1)$ or $\mathsf{Dec}^{sym}_{G(x)}(c_2)$, return reject.

Let $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})[T, \varepsilon, q_G, q_H, q_D]$ 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 $\mathcal{D}_{\mathsf{sk}}$ and $\mathcal{D}_{\mathsf{sk},c^\star}$. When queried with a ciphertext c, the first decription oracle answers $\mathsf{HE.Dec}(sk, c)$. The only difference between $\mathcal{D}_{sk}$ and $\mathcal{D}_{sk,c^\star}$ is that the second oracle rejects the query $c^\star$, answering reject. Then, the result claimed in [FO99] can be reformulated in the following way:

Theorem 89 *If there exists an adversary $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})[T, \varepsilon, q_G, q_H, q_D]$, then there exist an adversary $\mathcal{A}^{\mathsf{POW}}$ against the* TPOW *of f in time T_1 with success probability ε_1 and an adversary $\mathcal{A}^{\mathsf{IND-SYM}}(\mathcal{E}^{sym})$ against the* IND-SYM *security of $\mathcal{E}^{sym}$ in time T_2*

with advantage ε_2 such that

$$\varepsilon \leq (2(q_G + q_H)\varepsilon_1 + \varepsilon_2 + 1)\left(1 - 2\varepsilon_1 - 2\varepsilon_2 - \frac{1}{|Y|} - \frac{1}{|M|}\right)^{-q_D} - 1$$

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 [FO01] this result for the particular case of the Okamoto-Uchiyama scheme [OU98] (known as EPOC-2) and claimed, without proof, that a tight reduction is obtained for trivial easy verifiable primitives, in our terminology.

Identifying dangerous ambiguities

However, as pointed out in the introduction, several attacks against EPOC-2 have been found [JQY01, Den02a, ST02]. Despite the changes introduced in FO conversion after [JQY01], 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 $X_{\mathsf{pk}} \subset \overline{X}_{\mathsf{pk}}$ such that $f_{\mathsf{pk}}(x, y)$ is computable in polynomial time for any $x \in \overline{X}_{\mathsf{pk}}$ and $y \in Y_{\mathsf{pk}}$. Then, some badly generated ciphertexts $c = (f_{\mathsf{pk}}(x, H(x, m)), \mathsf{Enc}^{sym}_{G(x)}(m))$ for $x \in \overline{X}_{\mathsf{pk}} \setminus X_{\mathsf{pk}}$ may be accepted. This was the case for Okamoto-Uchiyama function in the original EPOC-2, where $\overline{X}_{\mathsf{pk}} = \mathbb{Z}_{2^{\ell}+1}$ and $X_{\mathsf{pk}} = \mathbb{Z}_{2^{\ell}}$, for $2^{\ell} < p < 2^{\ell} + 1$. This information was used in [JQY01] to obtain the secret value p.

As Fujisaki and Okamoto proposed later in [FO01], this attack is avoided if all ciphertexts (c_1, c_2) such that $g_{\mathsf{sk}}(c_1) \notin X_{\mathsf{pk}}$ 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 $\mathcal{D}_{\mathsf{sk}}$ 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 the 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 permutation (i.e. squaring quadratic residues modulo $n = pq$, $p \equiv q \equiv 3 \bmod 4$), where $X_{\mathsf{pk}} = Q_n$ and $\overline{X}_{\mathsf{pk}} = Q_n \cup -Q_n$. Rejection of all ciphertexts (c_1, c_2) such that $g_{\mathsf{sk}}(c_1) \notin X_{\mathsf{pk}}$ 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.

3.1.5 The new proposal

From the above discussion, we know that although it is necessary to check if $g_{\mathsf{sk}}(c_1) \in X_{\mathsf{pk}}$ 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 [Den02a, ST02], 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 [OP01b, CHJ+02a, CHJ+02b] 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 $\mathsf{PKE}^f = (\mathsf{PKE.KeyGen}^f, \mathsf{PKE.Enc}^f, \mathsf{PKE.Dec}^f)$ 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 Section 1.3.3. We believe it is not realistic to restrict the inputs of the random functions, as suggested in [FO99], 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 $Z_{\mathsf{pk}} \subseteq \overline{Z}_{\mathsf{pk}}$, and that the partial inverse of f_{pk} can also be computed (in polynomial time) on elements of the extended set $\overline{Z}_{\mathsf{pk}}$.

The proposed hybrid cryptosystem, $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec})$, 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 $\mathsf{HE.Dec}$. Thus, when a ciphertext is rejected, the adversary will know the reason, depending on the output, and will not be able to mount 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.

$\mathsf{HE.Dec}(\mathsf{sk}, c)$
1 if $c \notin \overline{Z}_{\mathsf{pk}} \times M_\ell$; return reject_1; endif
2 $(c_1, c_2) = c$
3 $x \leftarrow g_{\mathsf{sk}}(c_1)$
4 $m \leftarrow \mathsf{Dec}^{sym}_{G(x)}(c_2)$

5 $y \leftarrow H(x, m)$
6 if $x \notin X_{\mathsf{pk}}$ or $f_{\mathsf{pk}}(x, y) \neq c_1$; return reject$_2$; 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 90 *If there exists an adversary $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})[T, \varepsilon, q_G, q_H, q_D]$ 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 partial one-wayness of f with success probability ε_1 and an adversary $\mathcal{A}^{\mathsf{IND-SYM}}(\mathcal{E}^{sym})$ that in time T breaks* IND-SYM *security of $\mathcal{E}^{sym}$ with advantage ε_2 such that*

$$\varepsilon \leq \varepsilon_1 + 3\varepsilon_2 + \frac{2q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{2q_D}{|Y| - q_D}$$

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 Section 3.1.6. ∎

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

$$\varepsilon \leq \varepsilon_1 + 3\varepsilon_2 + \frac{2q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{2q_D}{|Y| - q_D}$$

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, $T[\mathcal{V}] = 1$. ■

Particular cases

Both in the case of the trivial construction of partial one-way function families and in the non-trivial family defined in Section 3.1.2, 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 $\mathsf{pk} \in \mathsf{pk}$, $x \in X_{\mathsf{pk}}$ and $z \in \overline{Z}_{\mathsf{pk}}$, $\mathcal{V}(\mathsf{pk}, x, z) = 1$ if and only if $\tilde{f}_{\mathsf{pk}}(x) = \tilde{\pi}_{\mathsf{pk}}(z)$. Notice that this property implies the injectivity of $\tilde{f}$. It is shown in the appendix that

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

which provides a *very-tight* security reduction.

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

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

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,

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

3.1.6 Security proof

Let $\mathcal{A}(\mathsf{HE})[T, \varepsilon, q_G, q_H, q_D] = (\mathcal{A}_1, \mathcal{A}_2)$ be the adversary aiming to attack the IND-CCA security of the hybrid encryption scheme, $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec})$ described in Section 3.1.5.

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 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ 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 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$. 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 an event across different games.

Lemma 92 *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*

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

Since we consider different probability spaces, this lemma is a generalization of the lemma used by Shoup in [Sho01].

Game0. The IND-CCA attack. There are some minor differences between Game0 and the standard IND-CCA game, described in Section 1.3, but they do not modify any probability.

$\mathsf{Game0}()$

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

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

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

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

For the sake of the readability of the rest in 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

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

Game1. In this game, the queries made by $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ 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 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ by the above functions.

Game1()
1 $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\ell)$; $G \leftarrow \mathcal{R}(K_\ell)$; $H \leftarrow \mathcal{R}(Y_{\mathsf{pk}})$
2 $b \leftarrow \{0,1\}$; $x^\star \leftarrow X_{\mathsf{pk}}$
• 3 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G1, H1, \mathcal{D}_{\mathsf{sk}}}(\mathsf{pk})$
4 $y^\star \leftarrow H(x^\star, m_b)$; $c^\star \leftarrow \left(f_{\mathsf{pk}}(x^\star, y^\star), \mathsf{Enc}^{sym}_{G(x^\star)}(m_b)\right)$
• 5 $b' \leftarrow \mathcal{A}_2^{G1, H1, \mathcal{D}_{\mathsf{sk}, c^\star}}(s, c^\star)$

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

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

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 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ to oracle $G1$ to the execution point. Now in this game, all ciphertexts (c_1, c_2) submitted to the decryption oracle such that $g_{\mathsf{sk}}(c_1) \notin X_{\mathsf{pk}} \cap \mathcal{Q}_G$ are rejected by returning reject_2, even when some of them may be valid ciphertexts.

$\mathcal{D}2_{\mathsf{sk}}(c)$
1 if $c \notin \overline{Z}_{\mathsf{pk}} \times M_\ell$; return reject_1; endif
2 $(c_1, c_2) = c$
3 $x \leftarrow g_{\mathsf{sk}}(c_1)$
• 4 if $x \notin X_{\mathsf{pk}}$ or $x \notin \mathcal{Q}_G$; return reject_2; endif

5 $m \leftarrow \mathsf{Dec}^{sym}_{G(x)}(c_2)$
6 $y \leftarrow H(x, m)$
7 if $f_{\mathsf{pk}}(x, y) \neq c_1$; return reject$_2$; endif
8 return m

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

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 $\mathcal{D}2_{\mathsf{sk}}$. Before F occurs, both games are indentical. Then, by Lemma 92,

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

From these inequalities, it can be easily shown that

$$\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})\right] \leq \mathsf{Pr}_2[\mathsf{S}_1] + |\mathsf{Pr}_2[\mathsf{S}_{01}] - \mathsf{Pr}_2[\mathsf{S}_{00}]| + 2\mathsf{Pr}[\mathsf{F}]$$

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

Lemma 93

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

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 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ 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 $\mathsf{View} = (\mathsf{pk}, \mathcal{T}_G, \mathcal{T}_H, \mathcal{T}_{\mathcal{D}})$, where $\mathcal{T}_{\mathcal{O}}$ denotes the sequence of all queries made by $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ 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 $\mathcal{D}2_{\mathsf{sk}}(\bar{c}) \neq \mathcal{D}_{\mathsf{sk}}(\bar{c})$; that is, $\mathcal{D}2_{\mathsf{sk}}$ rejects $\bar{c}$ (returning reject_2) while $\mathcal{D}_{\mathsf{sk}}$ accepts it. This means that $\bar{c} = (f_{\mathsf{pk}}(\bar{x}, \bar{y})), \bar{c}_2)$, where $\bar{x} \in X_{\mathsf{pk}} \setminus \mathcal{Q}_G$, $\bar{y} \in Y_{\mathsf{pk}}$, $\bar{c}_2 \in M_\ell$, 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 $\mathcal{D}_{\mathsf{sk}}$ to the queries $(f_{\mathsf{pk}}(\bar{x}, y), c_2)$ for some $y \in Y_{\mathsf{pk}}$ and $c_2 \in M_\ell$. Notice that any queried ciphertext $c \notin Z_{\mathsf{pk}} \times M_\ell$ is rejected by $\mathcal{D}_{\mathsf{sk}}$, 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 $c^{(j)} = (f_{\mathsf{pk}}(\bar{x}, y_j), c_2^{(j)})$ for $j = 1, \ldots, k-1$. Since $\bar{x} \notin \mathcal{Q}_G$, then $\mathcal{D}_{\mathsf{sk}}(c^{(j)}) = \mathcal{D}2_{\mathsf{sk}}(c^{(j)}) = \mathsf{reject}_2$, 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 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$. 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 $(\bar{x}, \mathsf{Dec}^{sym}_g(\bar{c}_2)) \notin \mathcal{Q}_H$, the variable $H(\bar{x}, \mathsf{Dec}^{sym}_g(\bar{c}_2))$ is uniformly distributed over a set of at least $|Y_{\mathsf{pk}}| - (k-1)$ elements, because if $\bar{c}_2 = c_2^{(j)}$, then the value y_j is forbidden. Consequently,

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

and summing up, we obtain

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

If $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ is in the 'guessing' stage, then $c^\star$ holds valuable information. In fact, $\mathsf{View} = (\mathsf{pk}, \mathcal{T}_G, \mathcal{T}_H, \mathcal{T}_{\mathcal{D}}, c^\star)$, 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 $H(\bar{x}, \mathsf{Dec}^{sym}_{G(\bar{x})}(\bar{c}_2))$ is conditioned by the equations $y_j \neq H(x^\star, \mathsf{Dec}^{sym}_{G(x^\star)}(c_2^{(j)}))$, for $j = 1 \ldots, k-1$, $y^\star = H(x^\star, m_b)$ and $m_b = \mathsf{Dec}^{sym}_{G(x^\star)}(c_2^\star)$.

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

Finally,

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

■

Game2′. In this game, oracles G and H are simulated by using tables $\mathcal{T}_G$ and $\mathcal{T}_H$, as described in Section 1.3.3. 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 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$, since in Game2 $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ neither $x^\star$ is queried to G nor $(x^\star, m)$ to H, for any m. (Note that, at step 6 of $\mathcal{D}2_{\mathsf{sk}}$, $x \neq x^\star$ since $x^\star \notin \mathcal{T}_{G1}$.)

$\mathsf{Game2}'()$

- 1 $\mathcal{T}_G \leftarrow \mathsf{empty}\,;\ \mathcal{T}_H \leftarrow \mathsf{empty}$
- 2 $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\ell)$
- 3 $b \leftarrow \{0,1\};\ x^\star \leftarrow X_{\mathsf{pk}}$
- 4 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G2', H2', \mathcal{D}2'_{\mathsf{sk}}}(\mathsf{pk})$
- 5 $g^\star \leftarrow K_\ell;\ y^\star \leftarrow Y_{\mathsf{pk}};\ c^\star \leftarrow \left(f_{\mathsf{pk}}(x^\star, y^\star), \mathsf{Enc}^{sym}_{g^\star}(m_b)\right)$
- 6 $b' \leftarrow \mathcal{A}_2^{G2', H2', \mathcal{D}2'_{\mathsf{sk}, c^\star}}(s, c^\star)$

$\mathcal{D}2'_{\mathsf{sk}}(c)$

- 1 if $c \notin \overline{Z}_{\mathsf{pk}} \times M_\ell$; return reject_1; endif
- 2 $(c_1, c_2) = c$
- 3 $x \leftarrow g_{\mathsf{sk}}(c_1)$
- 4 if $x \notin X_{\mathsf{pk}}$ or x not in $\mathcal{T}_G$; return reject_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 $f_{\mathsf{pk}}(x, y) \neq c_1$; return reject_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 $r \leftarrow Y_{\mathsf{pk}}$
- 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 $\mathcal{A}^{\mathsf{IND-SYM}}(\mathcal{E}^{sym}) = (\mathcal{A}_1^{sym}, \mathcal{A}_2^{sym})$ who tries to break the IND-SYM security of $\mathcal{E}^{sym}$ (see 3.1.3).

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
2 $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{HE.KeyGen}(1^\ell)$
3 $b \leftarrow \{0, 1\}$; $x^\star \leftarrow X_{\mathsf{pk}}$
4 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G3, H3, \mathcal{D}3_{\mathsf{sk}}}(\mathsf{pk})$
5 $m \leftarrow M_\ell$
6 $\sigma = (\mathcal{T}_G, \mathcal{T}_H, \mathsf{pk}, \mathsf{sk}, b, x^\star, s)$
7 return (m_b, m, σ)

$\mathcal{A}_2^{sym}(\sigma, \kappa^\star)$

1 $(\mathcal{T}_G, \mathcal{T}_H, \mathsf{pk}, \mathsf{sk}, b, x^\star, s) = \sigma$
2 $y^\star \leftarrow Y_{\mathsf{pk}};\ c^\star \leftarrow (f_{\mathsf{pk}}(x^\star, y^\star), \kappa^\star)$
3 $b' \leftarrow \mathcal{A}_2^{G3, H3, \mathcal{D}3_{\mathsf{sk}, c^\star}}(s, c^\star)$
4 if $b' = b$
5 $\quad \beta' \leftarrow 0$
6 else
7 $\quad \beta' \leftarrow 1$
8 endif
9 $\beta'' \leftarrow 0$

$G3(x)$

1 if x in $\mathcal{T}_G$; return $\mathcal{T}_G(x)$; endif
2 if $x = x^\star$
• 3 $\quad \beta' \leftarrow \{0, 1\}$
• 4 $\quad \beta'' \leftarrow 1$
5 $\quad$ exit game
6 endif
7 $r \leftarrow K_\ell$
8 insert (x, r) in table $\mathcal{T}_G$
9 return r

$H3(x, m)$

1 if (x, m) in $\mathcal{T}_H$; return $\mathcal{T}_H(x, m)$; endif
2 if $x = x^\star$
• 3 $\quad \beta' \leftarrow \{0, 1\}$
• 4 $\quad \beta'' \leftarrow 1$
5 $\quad$ exit game
6 endif
7 $r \leftarrow Y_{\mathsf{pk}}$
8 insert $((x, m), r)$ in table $\mathcal{T}_H$
9 return r

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

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

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

If $\beta = 1$, the value of m_b is used nowhere in the game. So, the view of $\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})$ is independent of b and $\mathsf{Pr}_3[\beta' = 1 \mid \beta = 1 \wedge \neg \mathsf{S}_1] = \mathsf{Pr}_3[b' \neq b \mid \beta = 1 \wedge \neg \mathsf{S}_1] = \frac{1}{2}$. Moreover, $\mathsf{Pr}_3[\beta' = 1 \mid \beta = 1 \wedge \mathsf{S}_1] = \frac{1}{2}$, and then $\mathsf{Pr}_3[\beta' = 1 \mid \beta = 1] = \frac{1}{2}$.

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

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

Then, combining the preceeding expressions:

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

If β'' is used instead of β', then

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

Finally,

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

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'.

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

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

1 $b \leftarrow \{0,1\}$
2 $m \leftarrow M_\ell;\ g^\star \leftarrow K_\ell;\ c^\star \leftarrow (z, \mathsf{Enc}_{g^\star}^{sym}(m))$
3 $\mathcal{T}_G \leftarrow$ empty ; $\mathcal{T}_H \leftarrow$ empty
4 $(m_0, m_1, s) \leftarrow \mathcal{A}_1^{G4,H4,\mathcal{D}4_{\mathsf{pk}}}(\mathsf{pk})$
5 $b' \leftarrow \mathcal{A}_2^{G4,H4,\mathcal{D}4_{\mathsf{pk},c^\star}}(s, c^\star)$
• 6 $x' \leftarrow X_{\mathsf{pk}}$

$G4(x)$

1 if x in $\mathcal{T}_G$; return $\mathcal{T}_G(x)$; endif
• 2 if $x \in X_{\mathsf{pk}}$ and $\mathcal{V}(\mathsf{pk}, x, z) = 1$
• 3 $x' \leftarrow x$
4 exit game
5 endif
6 $r \leftarrow K_\ell$
7 insert (x, r) in table $\mathcal{T}_G$
8 return r

$H4(x, m)$

1 if (x, m) in $\mathcal{T}_H$; return $\mathcal{T}_H(x, m)$; endif
• 2 if $x \in X_{\mathsf{pk}}$ and $\mathcal{V}(\mathsf{pk}, x, z) = 1$
• 3 $x' \leftarrow x$
4 exit game
5 endif
6 $r \leftarrow Y_{\mathsf{pk}}$
7 insert $((x, m), r)$ in table $\mathcal{T}_H$
8 return r

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

1 if $c \notin \overline{Z}_{\mathsf{pk}} \times M_\ell$; return reject$_1$; endif
2 $(c_1, c_2) = c$
• 3 foreach x in $\mathcal{T}_G$
• 4 if $x \in X_{\mathsf{pk}}$ and $\mathcal{V}(\mathsf{pk}, x, c_1) = 1$
5 $g \leftarrow \mathcal{T}_G(x)$
6 $m \leftarrow \mathsf{Dec}_g^{sym}(c_2)$
7 $y \leftarrow H4(x, m)$
8 if $f_{\mathsf{pk}}(x, y) \neq c_1$; return reject$_2$; endif

```
 9          return m
10        endif
• 11    endforeach
12      return reject_2
```

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

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

and, from the above results,

$$\begin{aligned}\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-CCA}}(\mathsf{HE})\right] \leq \mathsf{Succ}\left[\mathcal{A}^{\mathsf{POW}}\right] + 2\mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}(\mathcal{E}^{sym})\right] + \\ + \mathsf{Adv}\left[\mathcal{A}^{\mathsf{IND-SYM}}(\mathcal{E}^{sym})\right]' + \frac{2q_D q_H \gamma}{|K| - q_D q_H \gamma} + \frac{2q_D}{|Y| - q_D}\end{aligned}$$

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 $c \in \overline{Z}_{\mathsf{pk}} \times M_\ell$ and $x \in X_{\mathsf{pk}}$ is negligible.

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

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

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

Particular cases

Both in the case of the trivial construction of easy verifiable functions, and in the non-trivial family in Section 3.1.2, the algorithm $\mathcal{D}4_{\mathsf{pk}}$ 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 $x \in X_{\mathsf{pk}}$ to $G4$.

```
G̃4(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 $r \leftarrow K_\ell$
7 insert (x, r) in table $\mathcal{T}_G$
• 8 if $x \in X_{\mathsf{pk}}$
• 9 insert $(\tilde{f}_{\mathsf{pk}}(x), (x, r))$ in table $\tilde{\mathcal{T}}_G$
• 10 endif
11 return r

$\tilde{\mathcal{D}}4_{\mathsf{pk}}(c)$
1 if $c \notin \overline{Z}_{\mathsf{pk}} \times M_\ell$; return reject_1; endif
2 $(c_1, c_2) = c$
• 3 $\tilde{c}_1 \leftarrow \tilde{\pi}_{\mathsf{pk}}(c_1)$
• 4 if $\tilde{c}_1$ in $\tilde{\mathcal{T}}_G$
• 5 $(x, g) \leftarrow \tilde{\mathcal{T}}_G(\tilde{c}_1)$
6 $m \leftarrow \mathsf{Dec}_g^{sym}(c_2)$
7 $y \leftarrow H4(x, m)$
8 if $f_{\mathsf{pk}}(x, y) \neq c_1$; return reject_2; endif
9 return m
10 endif
11 return reject_2

The plaintext checking algorithm call to $\mathcal{V}(\mathsf{pk}, x, z)$ is replaced by the condition $\tilde{f}_{\mathsf{pk}}(x) = \tilde{\pi}_{\mathsf{pk}}(z)$, afterwards $\tilde{\pi}_{\mathsf{pk}}(z)$ 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 $\tilde{\mathcal{D}}4_{\mathsf{pk}}$ and step 9 in $\tilde{G}4$ can be neglected. Then,

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

3.2 Evaluating Elliptic Curve based KEMs in the light of Pairings

Several efforts have been made recently to put forward a set of cryptographic primitives for public key encryption, suitable to be standardized. In two of them (in the first place the NESSIE project [Nes03], already finished, and in the second place ISO/IEC 18033 [Sho04]), the KEM-DEM methodology by Cramer and Shoup for hybrid encryption has been included. Let us recall (cf. Section 1.3.2) that within this methodology, the problem of designing IND-CCA hybrid schemes is reduced to designing IND-CCA KEMs. Three elliptic curve-based KEMs have been considered so far, namely, ACE-KEM, ECIES-KEM and PSEC-KEM. Their security relies on different problems related to the discrete logarithm on elliptic curves. PSEC-KEM and ECIES-KEM use the Random Oracle (RO) heuristic [BR93] in their security proofs, while ACE-KEM is proven secure in the standard model but based on the decisional assumption ECDDH. They were first proposed as KEMs in [Sho01], the ISO standard draft for public key encryption edited by Victor Shoup, while in their original form they were submitted by IBM, Certicom and NTT corporations, respectively.

In [Jou00] a special family of curves, namely, elliptic curves with a non-trivial bilinear map were found a positive application in cryptography, designing a one-round tripartite Diffie-Hellmann protocol. A breakthrough in this constructive direction was made in [BF01], presenting the most complete and practical identity-based encryption scheme to the date. Since then, pairings have been found a lot of applications in cryptography (see [DBS04] for a comprehensive account), and its study has become an active research area.

Our contribution

We revisit the security proof of the elliptic curve-based KEMs when they are performed over pairing curves. As a result, we show that all these KEMs can be proven secure in the RO heuristic with respect to the ECDH assumption in a pairing curve, and with a very tight reduction, improving then the concrete security claimed over a random curve. It is worth pointing out that although the schemes are implemented over a pairing curve, and we use efficient pairing computations to obtain the concrete security, no pairing computation is involved in a real implementation. The crucial point is that ECDDH problem is solvable in these groups.

Since ECIES-KEM has the best perfomance, we conclude ECIES-KEM is preferable among the others if pairing curves are used. This is noticeable, since when using a randomly generated curve an opposite result is obtained. In fact, ECIES-KEM has not been selected in NESSIE, while ACE-KEM and PSEC-KEM have been positively

evaluated.

On the other hand, using [Mau94] there are elliptic curves where ECDL can be reduced to ECDH. Then, it is possible to give an exact security result relating the IND-CCA security of these KEMs to the ECDL problem. Moreover, they are closely related due to small security losses in the reduction. Finally, we provide some examples of pairing curves where the schemes can be implemented.

3.2.1 Security properties of existing elliptic curve based KEMs

In the following, we summarize the security properties of the KEMs discussed, as well as their performance and the evaluation presented in the NESSIE project. We begin with a schematic description of these algorithms. A description (E, P, p), where $(E, G, P, p, Q, u) \leftarrow I^{\mathbf{EC}}_{\mathsf{rand}}$, is added to the security parameter in the input to the key generation algorithm in these KEMs.

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(E, P, p, 1^\ell)$	$(K, C) \leftarrow \mathsf{Enc}(\mathsf{pk})$	$K \leftarrow \mathsf{Dec}(C, \mathsf{sk})$
1. $w, x, y, z \leftarrow \mathbb{Z}_p^*$	1. $r \leftarrow \mathbb{Z}_p^*$	1. Parse C as (C_1, C_2, C_3)
2. $W := wP$, $X := xP$, $Y := yP$, $Z := zP$	2. $C_1 := rP$	2. $\alpha := Hash(C_1 \| C_2)$
3. $\mathsf{pk} := (E, P, p, W, X, Y, Z, \ell)$	3. $C_2 := rW$	3. $t := x + y\alpha$
4. $\mathsf{sk} := (w, x, y, z, \mathsf{pk})$	4. $Q := rZ$	4. If $C_2 \neq wC_1$, output reject and halt
5. Output $(\mathsf{pk}, \mathsf{sk})$	5. $\alpha := Hash(C_1 \| C_2)$	5. If $C_3 \neq tC_1$, output reject and halt
	6. $C_3 := rX + \alpha rY$	6. $Q := zC_1$
	7. $C := (C_1, C_2, C_3)$	7. $K := KDF(C_1 \| Q)$
	8. $K := KDF(C_1 \| Q)$	8. Output K
	9. Output (K, C)	

Description of ACE-KEM

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(E, P, p, 1^\ell)$	$(K, C) \leftarrow \mathsf{Enc}(\mathsf{pk})$	$K \leftarrow \mathsf{Dec}(C, \mathsf{sk})$
1. $s \leftarrow \mathbb{Z}_p^*$	1. $r \leftarrow \mathbb{Z}_p^*$	1. $Q := sC$
2. $W := sP$	2. $C := rP$	2. If $Q = \mathcal{O}$ output reject and halt
3. $\mathsf{pk} := (E, P, p, W, \ell)$	3. Set x the x-coordinate of rW	3. Set x x-coordinate of rW
4. $\mathsf{sk} := (s, \mathsf{pk})$	4. $K = KDF(C \| x)$	4. $K = KDF(C \| x)$
5. Output $(\mathsf{pk}, \mathsf{sk})$	5. Output (K, C)	5. Output K

Description of ECIES-KEM

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(E, P, p, 1^\ell)$	$(K, C) \leftarrow \mathsf{Enc}(\mathsf{pk})$	$K \leftarrow \mathsf{Dec}(C, \mathsf{sk})$
1. $s \leftarrow \mathbb{Z}_p^*$	1. $r \leftarrow \{0,1\}^\ell$	1. Parse C as (C_1, C_2)
2. $W := sP$	2. $H := KDF(0_{32}\|\|r)$	2. $Q := sC_1$
3. $\mathsf{pk} := (E, P, p, W, \ell)$	3. Parse H as $t\|\|K$	3. $r := C_2 \oplus KDF(1\|\|C_1\|\|Q)$
4. $\mathsf{sk} := (s, \mathsf{pk})$	4. $\alpha := t \bmod p$	4. $H := KDF(0\|\|r)$
5. Output $(\mathsf{pk}, \mathsf{sk})$	5. $Q := \alpha W$	5. Parse H as $t\|\|K$
	6. $C_1 := \alpha P$	6. $\alpha := t \bmod p$
	7. $C_2 := r \oplus KDF(1_{32}\|\|C_1\|\|Q)$	7. If $C_1 \neq \alpha P$, output reject and halt
	8. $C := (C_1, C_2)$	8. Output K
	9. Output (K, C)	

Description of PSEC-KEM

A so-called *key derivation function* KDF has been used in these KEMs. This function can be considered as a hash function for our purposes (for further details see [Sho04]). In Table 3.1 we summarize the exact security results known for the KEMs we are interested in, along with the reference where these results come from. In these expressions, q_K denotes the number of queries made to the KDF oracle, L_G is the time needed to check a Diffie-Hellman triple in G, and SR_q is the time needed to compute a square root modulo q. We point out that in the ECIES-KEM security reduction claimed in [Den02b], the authors do not take into account the time to compute a square root in $\mathbb{F}_q$, which is needed in order to obtain the two points in $E(\mathbb{F}_q)$ that have a given x-coordinate.

Scheme	Assumption	Reduction	Random Oracle	Reference
ACE-KEM	ECDDH	very tight	No	[CS]
	Gap-ECDH	$\varepsilon' \approx \varepsilon$ $t' \approx t + q_K(2L_G + SR_q)$	Yes	[Sho01] [Den02b]
	ECDH	Not tight	Yes	[Sho00]
ECIES-KEM	Gap-ECDH	$\varepsilon' \approx \varepsilon$ $t' \approx t + q_K(2L_G + SR_q)$	Yes	[Den02b]
PSEC-KEM	ECDH	$\varepsilon' \approx \frac{1}{q_D + q_K}\varepsilon$ $t' \approx t$	Yes	[Sho01]

Table 3.1: IND-CCA KEMs concrete security over a random curve

As we can see, ACE-KEM offers several possible concrete security estimates, depending on which problem its security is based. In the case of the NESSIE evaluation the emphasis is put on the ECDDH problem, since the claimed security is achieved in the

standard model. On the other hand, ECIES-KEM presents a very tight reduction to the gap-ECDH problem, while PSEC-KEM has a not tight reduction to the ECDH problem. Both schemes are analysed with the RO heuristic. In Table 3.2 we have the parameter lengths in bytes for a 2^{80} IND-CCA security bound in each scheme. To compute them, it is assumed that ECDDH, Gap-ECDH and ECDH problems have comparable security to the ECDL problem in a random curve. Although this is widely believed, we emphasize that these *are extra assumptions*. Both ACE-KEM and ECIES-KEM use a group G with a $p \approx 2^{160}$ cardinality, while PSEC needs $p \approx 2^{280}$. This important difference arises from the not tight reduction in the security proof of PSEC-KEM. We notice that NESSIE parameter length estimation for PSEC is not exact (it is stated that a 160-bit prime is enough), and we argue it in Section 3.2.5.

Scheme	Operations in Enc	Operations in Dec	(K, C) length 16-Byte Keys	Public key length	Secret key length
ACE-KEM	5	3	76	80	80
ECIES-KEM	**2**	**1**	**36**	**20**	**20**
PSEC-KEM	2	2	67	35	35

Table 3.2: Performance features over random curves (byte lengths using a point compression technique)

In terms of performance, ECIES is clearly the best option. Not only does it present the smallest computation time, but also the smallest parameter length. However, since its security is based on a quite new assumption, NESSIE refused to propose standardizing ECIES-KEM, while accepted ACE-KEM and PSEC-KEM, since these schemes base its security on well studied assumptions. In the next section we argue that, if pairing curves are used, this conclusion is no longer valid. Moreover, we provide evidence that in this case ECIES-KEM arises as the best candidate.

3.2.2 Security analysis over pairing curves

Let $E(\mathbb{F}_q)$ be the group of points of an elliptic curve over the prime finite field $\mathbb{F}_q$. Let $G = \langle P \rangle$ be a cyclic subgroup of $E(\mathbb{F}_q)$ with p elements, where p is a large prime. Let $\mathbb{G}$ be a cyclic group with p elements. We say that E is a *pairing curve over $\mathbb{F}_q$ with respect to G* if there exists a map $e : G \times G \to \mathbb{G}$ with the following properties:

1. Bilinear: that is, $e(uP, vQ) = e(P, Q)^{uv}$, for all $P, Q \in G$ and all $u, v \in \mathbb{Z}$.

2. Non-degenerate: The map does not send all pairs in $G \times G$ to the identity in $\mathbb{G}$.

3. Computable: There is an efficient algorithm to compute $e(P,Q)$ for any $P,Q \in G$.

We call them pairing curves because the usual way to implement the map e is using the Weil or Tate pairings [Men93]. In this case, the group $\mathbb{G}$ is the multiplicative group of a certain finite extension $\mathbb{F}_{q^k}$. The number k is called the *embedding degree* and is the smallest positive integer such that $p|(q^k-1)$.

Let $(E,G,P,p,Q,s) \leftarrow I^{\mathbf{EC}}_{\mathsf{pairing}}(1^\ell)$ be a PPT algorithm sampling pairing curves and providing a PPT algorithm computing the map e. We denote with the superindex pairing the elliptic curve hard problems defined in Section 1.4.2 with respect to $I^{\mathbf{EC}}_{\mathsf{pairing}}$. For instance, $\mathrm{ECDL}^{\mathsf{pairing}}$ stands for the elliptic curve discrete logarithm with respect to a keypair generator $I^{\mathbf{EC}}_{\mathsf{pairing}}$. In contrast, the absence of superindex denotes the parameters haven generated at random (c.f. Section 1.4.2).

With such a map, the $\mathrm{ECDDH}^{\mathsf{pairing}}$ problem is solvable in G. The non-degeneracy property of the Weil and Tate pairings implies that $e(P,P)$ is p-th root of unity, and then (P,uP,vP,wP) is a valid Diffie-Hellman quadruple if and only if $e(uP,vP) = e(P,wP)$. It turns out then that the Gap-$\mathrm{ECDH}^{\mathsf{pairing}}$ and $\mathrm{ECDH}^{\mathsf{pairing}}$ problems are polynomial time equivalent in G, since there exists a polynomial time algorithm replacing the $\mathrm{ECDDH}^{\mathsf{pairing}}$ oracle solver. We use this fact positively to tightly relate the security of ACE-KEM, ECIES-KEM and PSEC-KEM to the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem.

Another consequence of the map e is that solving $\mathrm{ECDL}^{\mathsf{pairing}}$ problem in G can be transformed into solving the DL problem over the finite field $\mathbb{F}_{q^k}$, which can be computed using an index calculus algorithm running in subexponential time. This has been applied to attack the ECDL problem over supersingular curves in [MOV93, FR94]. We should take this into account when computing secure key sizes for each scheme.

Revisiting concrete security with respect to ECDH

We already know that in pairing curves Gap-$\mathrm{ECDH}^{\mathsf{pairing}}$ and $\mathrm{ECDH}^{\mathsf{pairing}}$ problems are equivalent. According to the results summarized in Table 3.1, this implies that ACE-KEM and ECIES-KEM are straightforward secure with respect to the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem. Indeed, they present a very tight reduction to the $\mathrm{ECDH}^{\mathsf{pairing}}$, and the concrete security estimation is obtained by replacing L_G by doubling the time needed to compute the map e, which will be denoted by T_e.

In the case of the PSEC-KEM security proof in [Sho01], the solver of the ECDH problem makes use of a (t,q_D,q_K,ε) adversary against the IND-CCA security of PSEC-KEM to generate a list of q_D+q_K elements containing the solution uvP to the instance (P,uP,vP) with probability roughly ε. Since the ECDDH problem is assumed to be intractable, we were forced to output an element of the list chosen uniformly at random, so the probability was decreased by a factor q_D+q_K. The reduction was then not tight.

Since in a pairing curve $\mathrm{ECDDH}^{\mathsf{pairing}}$ is efficiently solvable, we can find the correct value abP by testing the entries on the list, obtaining thus a solver of $\mathrm{ECDH}^{\mathsf{pairing}}$ with probability roughly ε within time $t + 2(q_K + q_D)T_e$. Therefore, PSEC-KEM presents a very tight security reduction, allowing the use of shorter ciphertexts for the same level of security in a random curve, as we shall see below. In Table 3.3 these concrete security results are summarized.

Scheme	Assumption	Reduction
ACE-KEM	CDH	$\varepsilon' \approx \varepsilon$ $t' \approx t + q_K(4T_e + SR_q)$
ECIES-KEM	CDH	$\varepsilon' \approx \varepsilon$ $t' \approx t + q_K(4T_e + SR_q)$
PSEC-KEM	CDH	$\varepsilon' \approx \varepsilon$ $t' \approx t + 2(q_K + q_D)T_e$

Table 3.3: Security results over a pairing curve

Hardness of the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem

When working with pairing curves we are dealing with a special family of elliptic curves, and then "some randomness" is lost with respect to the original random parameters generation algorithm in these KEMs. Therefore, the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem could be easier to solve than the ECDH problem. The following question then arises: Must we trust the hardness of the the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem? We answer this question positively from two points of view. On the one hand, we take into account the current status of pairing curves in cryptography research. As the survey [DBS04] shows, they are being intensively applied by the cryptographic community to design new appealing protocols. The new problems arising in these protocols can be reduced to $\mathrm{ECDH}^{\mathsf{pairing}}$. Consequently, the trustness on these new assumptions implies trustness on the $\mathrm{ECDH}^{\mathsf{pairing}}$ assumption.

On the other hand, using a technique due to Maurer [Mau94], it is possible to generate certain pairing elliptic curves with a cyclic group G for which ECDH and ECDL problems are equivalent. The basic idea is to transport computing ECDL in G to computing ECDL in an auxiliary group whose number of points has a suitable smoothness bound B. In the latter, the computation of DL can be carried out with a generic algorithm on subgroups of small size. The running time of this reduction is $\mathcal{O}(B \cdot (\log(p)^2)$ group operations in G and field operations in $\mathbb{F}_p$ and $\mathcal{O}(\log(p)^3)$ calls to the ECDH solver for G [MW00]. Since no attacks (different from Pollard ρ method) against $\mathrm{ECDL}^{\mathsf{pairing}}$

with a suitable embedding degree have been found, this theoretical equivalence gives a good indication of the hardness of the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem.

3.2.3 Efficiency analysis over pairing curves

Computing the security parameter

Let us assume that the IND-CCA security of any of these KEMs is $(t, q_D, q_K, \varepsilon)$-broken by some adversary $\mathcal{A}$. Since this adversary can be run repeatedly (with the same input and indepedent internal coin tosses), the expected time to distinguish a real encapsulation from random with advantage roughly 1 is t/ε. Thus, the security parameter of the scheme is $n_{\mathrm{KEM}} = \log(t/\varepsilon) = n + m$, where $n = \log t$ and $m = \log(1/\varepsilon)$.

Usually, $q_D \leq 2^{30}$ (that is, up to one billion decryption queries are allowed), and $q_K \leq t = 2^{55}$. We also consider that evaluating a KDF function is a unit operation (that is, takes the same time as a 3-DES encryption). Using Miller's algorithm, computing a pairing in $E(\mathbb{F}_q)$ with embedding degree k can be done in $\mathcal{O}(k \log q)$ multiplications in $\mathbb{F}_q$ (cf. [Men93]), while computing a square root modulo q takes at most $\mathcal{O}(\log^2 q)$ multiplications in $\mathbb{F}_q$ (cf. [Coh93]). Assuming that a multiplication in $\mathbb{F}_q$ takes 10 times longer than one hash query, and that $k \approx 10$, we obtain

$$t'_{\mathrm{ECIES}} \approx t + 2^{55} \cdot 10^2 \cdot (4 \log q + \log^2 q) \approx 2^n + 2^{62} \cdot \log^2 q$$

for ACE and ECIES-KEM, and

$$t'_{\mathrm{PSEC}} \approx t + 2^{56} \cdot 10^2 \cdot \log q \approx 2^n + 2^{63} \cdot \log q$$

for PSEC-KEM. In the following, we compute the exact security only for $t = 2^{80}$ for ECIES-KEM and PSEC-KEM, since the result for ACE-KEM is equal to the former. Setting $m = 0$ and $n = 80$, we obtain $n_{\mathrm{ECIES}} = 80$ (respectively $n_{\mathrm{PSEC}} = 80$), that is, a 2^{80} security level in each scheme. Let us compute the minimal parameter length to obtain this security level. An advantage roughly 1 in the IND-CCA game implies that the solver computes CDH succesfully with probability roughly 1 in time t'_{ECIES} (respectively t'_{PSEC}). Assuming that $|q| \approx 200$, then

$$t'_{\mathrm{ECIES}} \approx 2^{80} + 2^{62} \cdot 2^{15} \quad \text{and} \quad t'_{\mathrm{PSEC}} \approx 2^{80} + 2^{63} \cdot 2^8.$$

Both reductions are pretty meaningful and then, to get a 2^{80} security level on any of these KEMs a group G is enough with at least a 2^{80} security of the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem. If we make the *additional assumption* that the $\mathrm{ECDH}^{\mathsf{pairing}}$ and $\mathrm{ECDL}^{\mathsf{pairing}}$ problems have comparable security, then we need a group G with $149 \leq |p| \leq 165$ following [LV01]. In the case of PSEC, this is a great improvement compared to a length of roughly 280 bits needed over a random curve.

Furthermore, using a technique due to Maurer [Mau94], one can build certain pairing elliptic curves with a cyclic group G for which ECDH and ECDL problems are equivalent. As was claimed in the previous section, the running time of this reduction is $\mathcal{O}(B \cdot (\log(p)^2)$ group operations in G and field operations in $\mathbb{F}_p$ and $\mathcal{O}(\log(p)^3)$ calls to the CDH solver for G [MW00]. Since in our case the computation of $\mathrm{ECDH}^{\mathsf{pairing}}$ instances is by far the most expensive operation, the reduction to the $\mathrm{ECDL}^{\mathsf{pairing}}$ problem can be carried out with a 2^{23} factor decrease in security for all three schemes, therefore with a total time of $2^{80} \cdot 2^{23}$. Due to this somewhat small factor, the security of the scheme and the $\mathrm{ECDL}^{\mathsf{pairing}}$ problem are tightly related. This allows us to conclude that all three KEMs achieve provable security in the RO model, with the 2^{80} IND-CCA bound, in a group G with a 2^{103} security of the $\mathrm{ECDL}^{\mathsf{pairing}}$ problem, provided that the DL to CDH reduction of [Mau94] holds for this group.

Curves	Related Problem	Assumptions	Minimal security level
Pairing curves	$\mathrm{ECDH}^{\mathsf{pairing}}$	RO	2^{80}
Maurer pairing curves	$\mathrm{ECDL}^{\mathsf{pairing}}$	RO	2^{103}

Table 3.4: Discrete log KEMs for the 2^{80} security bound

Performance

It is now time to study the performance of each scheme over pairing curves. Since all three security reductions are very tight, we have seen that a 2^{80} IND-CCA security is achieved under a 2^{80} security level for the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem. Assuming that $\mathrm{ECDH}^{\mathsf{pairing}}$ and $\mathrm{ECDL}^{\mathsf{pairing}}$ problems have comparable security, and that the embedding degree is large enough to keep the DL infeasibility in $\mathbb{F}_{q^k}$ (in which case, the best attack known is to use the Pollard ρ method in G), a pairing curve $E(\mathbb{F}_q)$ with a group G with $|p| \approx 160$ is needed. However, as explained in the next section, the state of the art in pairing curves prevents us from claiming that $|p| \approx |q|$, but $|p| \leq |q| \leq 2\,|p|$. The performance comparison among the three KEMs will be stated then in bit units and in terms of the size of q. The results are presented in Table 3.5.

In the following section we present some pairing curves where $|p| \approx |q| \approx 160$. With these values, the performance features of ACE-KEM and ECIES-KEM are equivalent to those of Table 3.2, while in PSEC-KEM (K, C) length is reduced from 67 to 56 bytes, and public/secret keys are reduced from 35 to 20 bytes, thus obtaining a great improvement. From these values, we easily see that ECIES-KEM presents the best performance in every feature. Since all three KEMs base their security on the same problem, that is, the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem, we conclude that ECIES-KEM should be considered the best

Scheme	Operations in Enc	Operations in Dec	(K, C) length 16-Byte Keys	Public key length	Secret key length
ACE-KEM	5	3	$128 + 3\,\lvert q\rvert$	$4\,\lvert q\rvert$	$4\,\lvert q\rvert$
ECIES-KEM	**2**	**1**	$\mathbf{128 + \lvert q\rvert}$	$\mathbf{\lvert q\rvert}$	$\mathbf{\lvert q\rvert}$
PSEC-KEM	2	2	$128 + 2\lvert q\rvert$	$\lvert q\rvert$	$\lvert q\rvert$

Table 3.5: Performance features over pairing curves with 2^{80} security (bit lengths using a point compression technique)

option among these KEMs over pairing curves.

3.2.4 Examples of pairing curves

In the following we propose some curves in which the schemes can be performed, and we also discuss why they are suitable. Our aim is to give some examples of pairing curves $E(\mathbb{F}_q)$ to perform the schemes and where the $\mathrm{ECDH}^{\mathsf{pairing}}$ problem is assumed to be hard. We start by describing the conditions that a candidate curve must hold. In the first place, we want pairing curves with small embedding degree k, in order to obtain an efficient pairing computation. However, we cannot use too small embedding degrees: we must take into account that the field $\mathbb{F}_{q^k}$ has to be large enough to fit into the required security level. In our case, we are looking for a 2^{80} security level of the DL problem, which corresponds to $1024 \leq |q^k| \leq 1464$, according to the estimates by Lenstra and Verheul [LV01] and the parameters used nowadays.

Unfortunately, curves with small embedding degree are extremely rare, as shown in [BK98]. An exception are supersingular elliptic curves [Men93], which have $k \leq 6$. However, inasmuch as we are looking for small security parameters, only supersingular elliptic curves with $k = 6$ can fit our purpose. Nevertheless, it is not easy to generate such curves over prime finite fields, and the popular constructions use the field $\mathbb{F}_{3^m}$ (cf. [Gal01]).

Following [Gal04], an algorithm for generating curves with arbitrary k and with a large prime factor p of any form is proposed in [CP01]. Although it solves the embedding degree problem, it has the drawback that it produces curves with $q > p^2$. For instance, this means that for $k = 10$ and $|p| \approx 160$, the algorithm returns a curve $E(\mathbb{F}_q)$ with $|q| > 320$. It is an active area of research to obtain pairing curves in which $|q| \approx |p|$ and $k \geq 7$. First steps in this direction have been taken, for instance, in [DEM02, BW03, SB04]. From [SB04] we take two curves with $k = 6$; from the indications in Section 4.1 in [BW03] and from [Gal01] we derive three curves with $k \geq 7$, which can be used to implement the schemes. These curves are presented in Table 3.6.

$$E_{a,b}(\mathbb{F}_q) \ : \ y^2 = x^3 + ax + b; \quad |G| = p$$

k	6
q	801819385093403524905014779542892948310645897957 **(160 bits)**
a	-3
b	237567233982590907166836683655522398804119025399
p	801819385093403524905015674986573529844218487823 **(160 bits)**
k	6
q	469124930958906667660271791980080506853880359236358999638 9 **(192 bits)**
a	-3
b	311201765051646778586510196202962102273165873896518652743 3
p	234562465479453333830135895994234557291821573739852909483 7 **(192 bits)**
k	12
q	92023287709027882526875031742688685992195575554407985826771/ 85608987307 **(233 bits)**
a	92023287709027882526875031742688684330660552969612105131558931 23/ 268268
b	16615350225787512515295967795006976 1
p	91343854374875651026643947426601579968226918401 **(157 bits)**
k	10
q	21359906007365701929042154038677772262650043848653969045852/ 74435305514681762435224264786397102081 **(320 bits)**
a	70368760954882
b	29230057138066426933401941627939586556508189491 20
p	24519952037889827157137792820712629242745475072115343361 **(185 bits)**
k	6
q	3^{163} **(259 bits)**
a	-1
b	-1
p	58988115142665874085422772558073634885064063229737341 40/ 91790995505756623268837 **(259 bits)**

Table 3.6: Curves to implement KEMs equivalent to ECDH$^{\text{pairing}}$

A major breakthrough in the efficient implementation of these KEMs would be to find methods to generate pairing curves with embedding degree at least 7 and $|q| \approx |p|$. In this case, using ECIES-KEM over pairing curves should be suitable not only for pairing cryptographic environments, but also for medium level security settings with constrained computing and memory capabilities. This is likely the case for many embedded systems, such as smart cards.

3.2.5 PSEC parameter length over a random curve

In the sequel we fix the NESSIE parameter length estimation for PSEC. We use the notation introduced in Section 3.2.3. Let us assume the IND-CCA security of PSEC-

KEM is $(t, q_D, q_K, \varepsilon)$-broken by some adversary $\mathcal{A}$. Then the security parameter of the scheme is $n = \log(t/\varepsilon) = n + m$, where $n = \log t$ and $m = \log(1/\varepsilon)$, and $q_D \leq 2^{30}$, $q_K \leq 2^{60}$. The concrete security reduction for PSEC-KEM over a random curve is $t' \approx t$ and $\varepsilon' \approx \frac{\varepsilon}{q_D + q_K}$. Setting $m = 0$ and $n = 80$ (that is, a 2^{80} security level in the scheme), we obtain

$$t' \approx t = 2^{80} \quad \text{and} \quad \varepsilon' \approx 1/2^{60} = 2^{-60}.$$

From the last expression, an advantage roughly 1 in the IND-CCA game implies that the solver computes ECDH succesfully with probability roughly 2^{-60} in time $t' = 2^{80}$. However, an algorithm solving ECDH with probability roughly 1 is needed to find the parameter length. Running this algorithm with independent internal coin tosses 2^{60} times and returning the most frequent answer, ECDH is solved with probability roughly 1. The computational effort needed to do this is $2^{60} \cdot 2^{80} = 2^{140}$. Assuming that ECDH and ECDL problems have equivalent hardness over a random elliptic curve, we conclude that PSEC-KEM needs a subgroup G with $|p| \approx 280$, since the best attack known is using the Pollard ρ method.

Appendix A

In this appendix, we compute the number of points $(\bar{x}, \bar{y}) \in \mathbb{Z}_p \times \mathbb{Z}_p^*$ such that $\bar{x} \neq x$ and

$$\left(\frac{\bar{x}}{\bar{y}}\right)^4 = \left(\frac{x}{y}\right)^4$$

where $(x, y) \in D_p$ is the unique point such that $2\#(x, y) = 2\#(\bar{x}, \bar{y})$.

From observation 68, $(\bar{x}, \bar{y}) = (x, y) + (\eta, 0)$ and by the addition formula

$$\bar{x} = \left(\frac{y}{x-\eta}\right)^2 - x - \eta = \frac{x^3 - \eta^3}{(x-\eta)^2} - x - \eta = \frac{x^2 + \eta x + \eta^2}{x-\eta} - x - \eta = \eta\frac{x+2\eta}{x-\eta}$$

and

$$\bar{y} = \frac{y}{x-\eta}(\eta - \bar{x}) = \frac{\eta y}{x-\eta}\left(1 - \frac{x+2\eta}{x-\eta}\right) = -\frac{3\eta^2 y}{(x-\eta)^2}$$

Then, dividing both equations

$$\frac{\bar{x}}{\bar{y}} = -\frac{(x+2\eta)(x-\eta)}{3\eta y} = \rho\frac{x}{y}$$

where ρ is a fourth root of unity. This equation is equivalent to $(x+2\eta)(x-\eta) = -3\rho\eta x$, that means x is a root of the polynomial equation $(x+2\eta)^4(x-\eta)^4 = 81\eta^4 x^4$. So, there are at most 8 different values of x, given η. Moreover, there are at most 16 points $(\bar{x}, \bar{y})$ in each curve $E_p(0, b)$ satisfying the conditions at the beginning of this appendix. Finally, the probability that one of these points is guessed at random is at most $16/p$.

A tighter bound for this probability can be obtained if the second order equation $(x + 2\eta)(x - \eta) = -3\rho\eta x$ is discussed for each value of ρ. Let $t = x/\eta$. The equation can be rewriten as $(t + 2)(t - 1) = -3\rho t$, and also as $t^2 + (1 + 3\rho)t - 2 = 0$. The discriminant of the equation is $\Delta = (1 + 3\rho)^2 + 8 = 9\rho^2 + 6\rho + 9$.

Since $p \equiv 1 \bmod 4$, $(\frac{-1}{p}) = 1$ and there are 4 different values of ρ: 1, -1 and the two square roots of -1. Moreover, since $p \equiv 5 \bmod 12$ then $(\frac{3}{p}) = -1$ and $(\frac{2}{p}) = 1$ if and only if $p \equiv 1 \bmod 8$.

Taking this into account, if $\rho = 1$ then $\Delta = 24$, that is a quadratic residue only if $p \equiv 5 \bmod 8$. If $\rho = -1$ then $\Delta = 12$ that is not a quadratic residue. Finally, if $\rho^2 = -1$ then $\Delta = 6\rho$. But

$$(\tfrac{\rho}{p}) = \rho^{\frac{p-1}{2}} = (-1)^{\frac{p-1}{4}} \bmod p$$

that is equal to 1 if and only if $p \equiv 1 \bmod 8$. This implies that 2ρ is always a quadratic residue, so 6ρ never is.

Summing up previous stuff, the only values of t come up when $p \equiv 5 \bmod 8$ and $\rho = 1$. This two values are $t = -(2 \pm \sqrt{6})$. Now, $x = \eta t$ and $y^2 = x^3 - \eta^3 = \eta^3(t^3 - 1)$. From that, for each value of t, only $\frac{p-1}{2}$ values of η lead to existing values of y. It is easy to see that there are exactly $2(p-1)$ points (x, y), but only $p - 1$ are in D_p.

This last step follows from a symmetry argument. In all equations, (x, y) and $(\bar{x}, \bar{y})$ play a symmetric role, since $(\bar{x}, \bar{y}) = (x, y) + (\eta, 0)$ is equivalent to $(x, y) = (\bar{x}, \bar{y}) + (\eta, 0)$. But $(x, y) \in D_p$ and $(\bar{x}, \bar{y}) \notin D_p$. Thus, only half of the solutions found correspond to values of (x, y), and the other half correspond to $(\bar{x}, \bar{y})$.

Bibliography

[AM93] A.O.L. Atkin and F. Morain. Elliptic curves and primality proving. *Mathematics of Computation*, 61:29–67, 1993.

[BBP04] M. Bellare, A. Boldyreva and A. Palacio. An uninstantiable Random-Oracle-Model scheme for a hybrid-encryption problem. In *Advances in Cryptology – EUROCRYPT ' 2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 449–461, 2004.

[BD99] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $n^{0.292}$. In *Advances in Cryptology – EUROCRYPT ' 1999*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 1–11, 1999.

[BDHG99] D. Boneh, G. Durfee and N.A. Howgrave-Graham. Factoring $n = p^r q$ for large r. In *Advances in Cryptology – CRYPTO ' 1999*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 326–337, 1999.

[BDPR98] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO 1998*, vol. 1462 of *Lecture Notes in Computer Science*, pp. 26–45, 1998.

[Bel98] M. Bellare. Practice-oriented provable-security. In *1st. International Workshop on Information Security (ISW 97)*, vol. 1396 of *Lecture Notes in Computer Science*, pp. 221–231, 1998.

[BF01] D. Boneh and M. Franklin. Identity-Based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO ' 01*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, 2001.

[BG85] M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In *Advances in Cryptology – CRYPTO ' 1984*, vol. 196 of *Lecture Notes in Computer Science*, pp. 289–302, 1985.

[BK98] R. Balasubramanian and N. Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *Journal of Cryptology*, 11(2):141–145, 1998.

[BL96] D. Boneh and R.J. Lipton. Algorithms for black-box fields and their application to cryptography (extended abstract). In *Advances in Cryptology – CRYPTO ' 1996*, vol. 1109 of *Lecture Notes in Computer Science*, pp. 283–297, 1996.

[Ble98] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *Advances in Cryptology – CRYPTO 1998*, vol. 1462, pp. 1–12, 1998.

[Bon99] D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices*, 46(2):203–213, 1999.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM CCS*, pp. 62–73. ACM Press, 1993.

[BR95] M. Bellare and P. Rogaway. Optimal asymmetric encryption – how to encrypt with RSA. In *Advances in Cryptology – EUROCRYPT 1994*, vol. 950 of *Lecture Notes in Computer Science*, pp. 92–111, 1995.

[Bre98] R.P. Brent. Some integer factorization algorithms using elliptic curves. *Australian Computer Science Comunications*, pp. 24–26, 1998. Republished 1998.

[BSS99] I.F. Blake, G. Seroussi and N. Smart. *Elliptic Curves in Cryptography*, vol. 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

[BV98] D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In *Advances in Cryptology – EUROCRYPT ' 1998*, vol. 1233 of *Lecture Notes in Computer Science*, pp. 59–71, 1998.

[BW03] F. Brezing and A. Weng. Elliptic curves suitable for pairing based cryptography. Cryptology ePrint Archive, Report 2003/143, 2003. `http://eprint.iacr.org/`.

[Cer00a] Certicom. SEC1: Elliptic Curve Cryptography, 2000. Standards for Efficient Cryptography Group, September 2000. Available at `www.secg.org`.

[Cer00b] Certicom. SEC2: Recommended Elliptic Curve Domain Parameters, 2000. Standards for Efficient Cryptography Group, September 2000. Available at `www.secg.org`.

[CGH98] R. Canetti, O. Goldreich and S. Halevi. The Random Oracle Methodology, revisited. In *Proceedings of the 30th Annual Symposium on Theory Of Computing (STOC)*, pp. 209–218. ACM Press, 1998.

[CGHN01] D. Catalano, R. Gennaro, N. HowgraveGraham and P. Nguyen. Paillier's cryptosystem revisited. In *Proceedings of the 8th ACM CCS*, pp. 206–214. ACM Press, 2001.

[CHJ99] D. Coppersmith, S. Halevi and C. Jutla. ISO 9796-1 and the New Forgery Strategy. Presented at the Rump session of Crypto' 99, 1999.

[CHJ+02a] J. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval and C. Tymen. GEM: a generic chosen-ciphertext secure encryption method. In *CT-RSA 2002*, vol. 2271 of *Lecture Notes in Computer Science*, pp. 263–276, 2002.

[CHJ+02b] J. Coron, H. Handschuh, M. Joye, P. Paillier, D. Pointcheval and C. Tymen. Optimal chosen-ciphertext secure encryption of arbitrary-length messages. In *PKC 2002*, vol. 2274 of *Lecture Notes in Computer Science*, pp. 17–33, 2002.

[CNS99] J.S. Coron, D. Naccache and J.P. Stern. On the security of RSA padding. In *Advances in Cryptology – CRYPTO 1999*, vol. 1666, pp. 1–19, 1999.

[CNS02] 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*, vol. 2501 of *Lecture Notes in Computer Science*, pp. 299–311, 2002.

[Coh93] H. Cohen. *A Course in Computational Algebraic Number Theory*, vol. 138 of *Graduate Texts in Mathematics*. Springer, 1993.

[Cop96] D. Coppersmith. Finding a small root of a univariate modular equation. In *Advances in Cryptology – EUROCRYPT 1996*, vol. 1070 of *Lecture Notes in Computer Science*, pp. 155–165, 1996.

[CP01] C. Cocks and R.G.E. Pinch. Identiy-based cryptosystems based on the Weil pairing, 2001. Unpublished manuscript.

[CS] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. To appear at *SIAM Journal of Computing*. Available at `www.shoup.net`.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology — CRYPTO 1998*, vol. 1462 of *Lecture Notes in Computer Science*, pp. 13–25, 1998.

[CS02] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology — EUROCRYPT 2002*, vol. 2332 of *Lecture Notes in Computer Science*, pp. 45–64, 2002.

[DBS04] R. Dutta, R. Barua and P. Sarkar. Pairing-based cryptography : A survey. Cryptology ePrint Archive, Report 2004/064, 2004. `http://eprint.iacr.org/`.

[DDN91] D. Dolev, C. Dwork and M. Naor. Non-malleable cryptography. In ACM, editor, *Proceedings of the 23rd ACM Symposium on Theory of Computing*, pp. 542–552. IEEE Computer Society Press, 1991.

[DEM02] R. Dupont, A. Enge and F. Morain. Building curves with arbitrary small mov degree over finite prime fields. Cryptology ePrint Archive, Report 2002/094, 2002. `http://eprint.iacr.org/`.

[Den02a] A. W. Dent. An implementation attack against the EPOC-2 public-key cryptosystem. *ELECTRONICS LETTERS*, 38(9):412–413, 2002.

[Den02b] A.W. Dent. ECIES-KEM vs. PSEC-KEM. Technical Report `NES/DOC/RHU/WP5/028/2`, NESSIE, 2002.

[Den03] A.W. Dent. A designer's guide to KEMs. In *IMA Int. Conf. 2003*, vol. 2898 of *Lecture Notes in Computer Science*, pp. 133–151, 2003.

[DK02] H. Delfs and H. Knebl. *Introduction to Cryptography. Principles and Applications.* Springer-Verlag, 2002.

[EPO] EPOC. *Efficient Probabilistic Public-Key Encryption*. `http://info.isl.ntt.co.jp/epoc/`.

[FO99] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology — CRYPTO 1999*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 537–554, 1999.

[FO01] E. Fujisaki and T. Okamoto. A chosen-cipher secure encryption scheme tightly as secure as factoring. *IEICE Trans. Fundamentals*, E84-9(1):179–187, 2001.

[FOPS01] E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern. RSA-OAEP is secure under the RSA assumption. In *Advances in Cryptology – CRYPTO ' 2001*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 260–274, 2001.

[FR94] G. Frey and H.G. Rück. A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of curves. *Mathematics of Computation*, 62:865–874, 1994.

[Gal01] S. Galbraith. Supersingular curves in cryptography. In *Advances in Cryptology – ASIACRYPT 2001*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 495–513, 2001.

[Gal02] S. Galbraith. Elliptic curve Paillier schemes. *Journal of Cryptology*, 15(2):129–138, 2002.

[Gal04] S. Galbraith. Pairings, 2004. Unpublished manuscript.

[GH99] G. Gong and L. Harn. Public-key cryptosystems based on cubic finite field extensions. *IEEE Transactions on Information Theory*, 45(7):2601–2605, 1999.

[GLM+04] R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali and T. Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In *Theory of Cryptography Conference, TCC 2004*, vol. 2951 of *Lecture Notes in Computer Science*, pp. 258–277, 2004. An on-line version is available at the Cryptology ePrint Archive at `http://eprint.iacr.org/2003/120/`.

[GLN02] O. Goldreich, Y. Lustig and M. Naor. On chosen ciphertext security of multiple encryptions. Cryptology ePrint Archive, Report 2002/089, 2002. `http://eprint.iacr.org/`.

[GM84] S. Golwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[GMMV02] D. Galindo, S. Martín, P. Morillo and J. L. Villar. A practical public key cryptosystem from Paillier and Rabin schemes. In *PKC 2003*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 279–291, 2002.

[GMMV03a] D. Galindo, S. Martín, P. Morillo and J. L. Villar. Easy verifiable primitives and practical public key cryptosystems. In *ISC 2003*, vol. 2851 of *Lecture Notes in Computer Science*, pp. 69–83, 2003.

[GMMV03b] D. Galindo, S. Martín, P. Morillo and J. L. Villar. An efficient semantically secure elliptic curve cryptosystem based on KMOV. In *International Workshop on Coding and Cryptography WCC 2003*, pp. 213–221, 2003.

[GMMV03c] D. Galindo, S. Martín, P. Morillo and J. L. Villar. An IND-CPA cryptosystem from Demytko's primitive. In *2003 IEEE Information Theory Workshop*, pp. 167–170, 2003.

[GMMV04] D. Galindo, S. Martín, P. Morillo and J. L. Villar. Fujisaki-Okamoto hybrid encryption revisited. *International Journal of Information Security*, 2004. Full version of [GMMV03a]. To appear.

[GMTV04] D. Galindo, S. Martín, T. Takagi and J. L. Villar. A provably secure elliptic curve scheme with fast encryption, 2004. Submitted.

[GMV04] D. Galindo, S. Martín and J. L. Villar. Evaluating elliptic curve based KEMs in the light of pairings, 2004. Submitted.

[Gol93] O. Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol01] O. Goldreich. *Foundations of Cryptography - Basic Tools*. Cambridge University Press, 2001.

[GT03] S. Goldwasser and Y. Tauman. On the (in)security of the Fiat-Shamir paradigm. In *FOCS 2003*, IEEE Computer Society, pp. 102–, 2003.

[HG99] N.A. Howgrave-Graham. *Computational Mathematics Inspired by RSA*. PhD thesis, University of Bath, 1999.

[HG01] N.A. Howgrave-Graham. Approximate integer common divisors. In *CaLC*, vol. 2146 of *Lecture Notes in Computer Science*, pp. 51–66, 2001.

[HU79] J. Hopcroft and J. Ullman. *Introduction to automata theory, languages, and computation*. Reading, Massachusetts Addison-Wesley, 1979.

[IEE99] IEEE P1363/D13. Standard specifications for public key cryptography, 1999. Last preliminary draft `http://grouper.ieee.org/groups/1363/P1363/draft.html`. The approved standard is IEEE P1363-2000.

[JN03] A. Joux and K. Nguyen. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–247, 2003.

[Jou00] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *ANTS 2000*, vol. 1838 of *Lecture Notes in Computer Science*, pp. 385–394, 2000.

[Jou02] A. Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In *ANTS 2002*, vol. 2369 of *Lecture Notes in Computer Science*, pp. 20–32, 2002.

[JQY01] M. Joye, J.J. Quisquater and M. Yung. On the power of misbehaving adversaries and security analysis of the original EPOC. In *CT-RSA 2001*, vol. 2020 of *Lecture Notes in Computer Science*, pp. 208–222, 2001.

[KJJ99] P. Kocher, J. Jaffe and B. Jun. Differential power analysis. In *Advances in Cryptology – CRYPTO ' 99*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 399–397, 1999.

[KMOV91] K. Koyama, U.M. Maurer, T. Okamoto and S.A. Vanstone. New public-key schemes based on elliptic curves over the ring $\mathbb{Z}_n$. In *Advances in Cryptology – CRYPTO 1991*, vol. 576 of *Lecture Notes in Computer Science*, pp. 252–266, 1991.

[KMV00] Neal Koblitz, Alfred Menezes and Scott A. Vanstone. The state of Elliptic Curve Cryptography. *Designs, Codes and Cryptography*, (2/3):173–193, 2000.

[Kob92] N. Koblitz. CM-curves with good cryptographic properties. In *Advances in Cryptology – CRYPTO ' 1991*, vol. 576 of *Lecture Notes in Computer Science*, pp. 279–287, 1992.

[Koc96] P.C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. *Lecture Notes in Computer Science*, 1109:104–113, 1996.

[KT03] K. Kurosawa and T. Takagi. Some RSA-based encryption schemes with tight security reduction. In *Advances in Cryptology – ASIACRYPT ' 2003*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 19–36, 2003.

[LV01] A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[LZ94] G.J. Lay and H.G. Zimmer. Constructing elliptic curves with given group order over large finite fields. In *ANTS '94*, vol. 877 of *Lecture Notes in Computer Science*, pp. 250–263, 1994.

[Mau94] U. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In *Advances in Cryptology — CRYPTO '94*, vol. 839 of *Lecture Notes in Computer Science*, pp. 271–281, 1994.

[Men93] A. Menezes. *Elliptic Curve Public Key Cryptosystems*, vol. 234 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1993.

[MOV93] A.J. Menezes, T. Okamoto and S.A. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Transactions on Information Theory*, 39:1639–1646, 1993.

[MR04] S. Micali and L. Reyzin. Physically Observable Cryptography (extended abstract). In *Theory of Cryptography Conference, TCC 2004*, vol. 2951 of *Lecture Notes in Computer Science*, pp. 278–296, 2004.

[MvOV97] A. Menezes, , P. van Oorschot and S. Vanstone. *Handbook of Applied Cryptography*. Discrete Mathematics and its Applications. CRC Press, 1997. Available at `http://www.cacr.math.uwaterloo.ca/hac/`.

[MW00] U. Maurer and S. Wolf. The Diffie-Hellman protocol. *Designs, Codes, and Cryptography*, 19:147–171, 2000.

[Nes03] Nessie. NESSIE security report. version 2.0, 2003. `http://www.cryptonessie.org/`.

[NY90] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attack. In *Proc. of the Twenty-Second Annual ACM Symposium on Theory of Computing*, pp. 427–437. ACM, 1990.

[OP01a] T. Okamoto and D. Pointcheval. The gap-problems: a new class of problems for the security of cryptographic schemes. In *PKC 2001*, vol. 1992 of *Lecture Notes in Computer Science*, pp. 104–118, 2001.

[OP01b] T. Okamoto and D. Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In *CT-RSA 2001*, vol. 2020 of *Lecture Notes in Computer Science*, pp. 159–175, 2001.

[OU98] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology — EUROCRYPT 1998*, vol. 1403 of *Lecture Notes in Computer Science*, pp. 308–318, 1998.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT 1999*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 223–238, 1999.

[PG97] J. Patarin and L. Goubin. Trapdoor one-way permutations and multivariate polynomials ($extended\ version$). In *ICICS 1997*, vol. 1334 of *Lecture Notes in Computer Science*, pp. 356–368, 1997.

[Poi00] D. Pointcheval. Chosen-ciphertext security for any one-way cryptosystem. In *PKC 2000*, vol. 1751 of *Lecture Notes in Computer Science*, pp. 129–146, 2000.

[Poi02] D. Pointcheval. Practical security in public-key cryptography. In *ICISC '01*, vol. 2288 of *Lecture Notes in Computer Science*, pp. 1–17, 2002.

[Pol78] J.M. Pollard. Monte carlo methods for index computation mod p. *Mathematics of Computation*, 32:918–924, 1978.

[Rab79] M.O. Rabin. Digitalized signatures and public key functions as intractable as factorisation. Technical Report 212, MIT Laboratory for Computer Science, 1979.

[RS92] C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology – CRYPTO 1991*, vol. 576 of *Lecture Notes in Computer Science*, pp. 433–444, 1992.

[SA98] T. Satoh and K. Araki. Fermat quotients adn the polynomial time discrete log algorithm for anomalous elliptic curves. *Comm. Math. Univ. Sancti Pauli*, 47:81–92, 1998.

[SB04] M. Scott and P.S.L.M Barreto. Generating more MNT elliptic curves. Cryptology ePrint Archive, Report 2004/058, 2004. `http://eprint.iacr.org/`.

[Sch95] R. Schoof. Counting points on elliptic curves over finite fields. *J. Theorie de Nombres de Bourdeaux*, 7:219–254, 1995.

[Sem98] I. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation*, 67:353–356, 1998.

[Sha49] C.E. Shannon. Communication theory of secrecy systems. *Bell. Sys. Tech. Journal*, 28:656–715, 1949.

[Sho00] V. Shoup. Using hash functions as a hedge against chosen ciphertext attacks. In *Advances in Cryptology – EUROCRYPT ' 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 275–288, 2000.

[Sho01] V. Shoup. A proposal for an ISO standard for public key encryption. Technical Report 2.1, 2001.

[Sho04] V. Shoup. Draft ISO/IEC 18033-2: An emerging standard for public-key encryption. Technical report, ISO/IEC, 2004.

[Sil86] J.H. Silverman. *The arithmetic of elliptic curves*, vol. 106 of *Graduate Texts on Mathematics*. Springer Verlag, 1986.

[Sma99] N. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, 12(3):193–196, 1999.

[ST02] K. Sakurai and T. Takagi. A reject timing attack on an IND-CCA2 public-key cryptosystem. In *ICISC 2002*, vol. 2587 of *Lecture Notes in Computer Science*, pp. 359–373, 2002.

[Ste03] J. Stern. Why provable security matters? In *Advances in Cryptology – EUROCRYPT ' 2003*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 449–461, 2003.

[Til99] H.C.A.van Tilborg. *Fundamentals of Cryptology. A Professional Reference and Interactive Tutorial*, vol. 528. Kluwer Academic Publishers SECS, 1999.

[Wat69] W. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. École Norm. Sup.*, 4(2):521–560, 1969.

[Wil80] H.C Williams. A modification of the RSA public-key encryption procedure. *IEEE Transactions on Information Theory*, 26(6):726–729, 1980.

[WSI02] Y. Watanabe, J. Shikata and H. Imai. Equivalence between semantic security and indistinguishability against chosen ciphertext attacks. In *PKC 2003*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 71–84, 2002.

[X9.99] ANSI X9.62-1998. Public key cryptography for the financial services industry : The elliptic curve digital signature algorithm (ECDSA), 1999. Approved American National Standard.

[X9.01] ANSI X9.63-2001. Public key cryptography for the financial services industry, key agreement and key transport using elliptic curve cryptography, 2001. Working draft.