Fourth Conference on Algebraic Geometry, Number Theory, Coding Theory and Cryptography
(21–24 november 2000, Tokyo, Japan).

About Generic Conversions from any Weakly Secure Encryption Scheme into a Chosen-Ciphertext Secure Scheme

David Pointcheval

LIENS – CNRS, École Normale Supérieure, 45 rue d'Ulm, 75230 Paris Cedex 05, France.
David.Pointcheval@ens.fr, http://www.di.ens.fr/~pointche.

Abstract. Since the appearance of public-key cryptography in the seminal Diffie-Hellman paper, many schemes have been proposed, but many have been broken. Indeed, for many people, the simple fact that a cryptographic algorithm withstands cryptanalytic attacks for several years is considered as a kind of validation. But some schemes took a long time before being widely studied, and maybe thereafter being broken.
A much more convincing line of research has tried to provide "provable" security for cryptographic protocols, in a complexity theory sense: if one can break the cryptographic protocol, one can efficiently solve the underlying problem.
Unfortunately, very few practical schemes can be proven in this so-called "standard model" because such a security level rarely meets with efficiency. A convenient way to achieve some kind of validation of efficient schemes has been to identify some concrete cryptographic objects with ideal random ones: hash functions are considered as behaving like random functions, in the so-called "random oracle model", and groups are used as black-box groups, in which one has to ask for additions to get new elements, in the so-called "generic model".
In this paper we present some generic designs for asymmetric encryption with provable security in the random oracle model.

Keywords: Cryptography, Public-Key Encryption, Provable Security, Random Oracle Model, Discrete Logarithm, Diffie-Hellman Problems, Elliptic Curves

1 Introduction

1.1 Motivation

Since the beginning of public-key cryptography, with the seminal Diffie-Hellman paper [17], many suitable problems for cryptography have been proposed (*e.g.* one-way —possibly trapdoor— functions) and many cryptographic schemes have been designed, together with more or less heuristic proofs of their security relative to the intractability of these problems (namely from the number theory, such as the integer factorization, RSA [51], the discrete logarithm [20] and the Diffie-Hellman [17] problems, or from the complexity theory with some $\mathcal{NP}$-complete problems, such as the knapsack [15] problem or the decoding problem of random linear codes [35]). However, most of those schemes have thereafter been broken.

The simple fact that a cryptographic algorithm withstands cryptanalytic attacks for several years is often considered as a kind of validation procedure, but some schemes take a long time before being broken. The best example is certainly the Chor-Rivest cryptosystem [15, 34, 63], based on the knapsack problem, which took more than 10 years to be totally broken, whereas before this last attack it was believed to be very hard, since all the classical attacks against knapsack instances, such as LLL [33], have failed because of the high density of the involved instances. With this example, but also many others, the lack of attacks at some time should not be considered as a validation of the proposal.

1.2 Provable Security and Exact Security

A completely different paradigm is provided by the concept of "provable" security. A significant line of research has tried to provide proofs in the framework of complexity theory: the proofs provide reductions from a well-studied problem (RSA or the discrete logarithm) to an attack against a cryptographic protocol. Firstly, people just tried to produce polynomial reductions, in an asymptotic way [26, 25, 38, 49]. However, such a result has no practical impact on the real security. Indeed, even with a polynomial reduction, one may be able to break the cryptographic protocol within few hours, whereas the reduction just leads to an algorithm against the underlying problem which requires many years. Therefore, those reductions only prove the security when very huge (and thus unpractical) parameters are used.

For few years, more efficient reductions have been expected, under the denominations of either "exact security" [9] or "concrete security" [43], which provide practical security results. The ideal situation is reached when one manages to prove that from an attack, one can describe an algorithm against the underlying problem, with almost the same success probability within almost the same amount of time.

Unfortunately, in many cases, provable security is at the cost of an important loss in terms of efficiency for the cryptographic protocol. Thus some models have been proposed, trying to deal with the security of efficient schemes: some concrete objects are identified with ideal (or black-box) ones.

For example, it is by now usual to identify hash functions with ideal random functions, in the so-called "random oracle model" formalized by Bellare and Rogaway [7]. More recently, an other kind of idealization has been introduced in cryptography, the black-box group [39, 59]: a new element necessarily comes from the addition (or the subtraction) of two already known elements. It is the by now called "generic model". A recent work [55] even uses both models together to provide some new validations.

1.3 Outline of the Paper

In the next section, we discuss about the correctness of both ideal models, the random oracle model and the generic model. Then, we precise the problems on which DL-based schemes can rely. In the following section, we review the formalism of public-key encryption schemes, with the security notions and some examples. Then, we present some generic conversions which apply to any weakly secure scheme to make a strongly secure one, with a sketch of the security analysis.

2 The Ideal Models

As said above, efficiency rarely meets with provable security. More precisely, none of the most efficient schemes in their category has been proven secure in the standard model. However, some of them admit security validations under ideal model assumptions: the random oracle model and/or the generic model.

2.1 The Random Oracle Model

Many cryptographic schemes use a hash function H (such as MD5 [50] or the American standards SHA-1 [41], SHA-256, SHA-384 and SHA-512 [42]). This use of hash functions was originally motivated by the wish to sign long messages with a single short signature. In order to achieve *non-repudiation*, a minimal requirement on the hash

function is the impossibility for the signer to find two different messages providing the same hash value. This property is called *collision-resistance*.

It was later realized that hash functions were an essential ingredient for the security of, first, signature schemes, and then of most of the cryptographic schemes. In order to obtain security arguments, while keeping the efficiency of the designs that use hash functions, few authors suggested using the hypothesis that H behaves like a random function. First, Fiat and Shamir [21] applied it heuristically to provide a signature scheme "as secure as" factorization. Then, Bellare and Rogaway [7, 8] formalized this concept in many fields of cryptography: signature and public-key encryption.

In this model, the so-called "random oracle model", the hash function can be seen as an oracle which produces a truly random value for each new query. Of course, if the same query is asked twice, identical answers are obtained. This is precisely the context of relativized complexity theory with "oracles," hence the name.

2.2 The Generic Model

Another model has been more recently defined, by now known as the "generic model". This model, used first by Naechev [39], focuses on adversaries that do not exploit any special property of the encodings of group elements other than the fact that each group element is encoded by a unique string. Typically, algorithms like Pollard's [48] and Shanks' [56] algorithms are under the scope of this formalism, while index-calculus methods [32, 27] do not fall in this category. Therefore, generic algorithms just use the group-operation (*e.g.* the addition) as a black-box: any new element necessarily comes from the addition (or the subtraction) of two already known elements. More recently, Shoup [59] gave lower bounds for generic algorithms against the discrete logarithm, the computational Diffie-Hellman problem [17] and the decisional version [12]. He therefore provided a lower bound for any "generic adversary" against the Schnorr's identification scheme [53].

2.3 Discussion

About the random oracle model, no one has ever been able to provide a convincing contradiction to this model, but just a theoretical counter-example [13] which uses a classical diagonalization technique on clearly wrong designs for practical purpose! Therefore, this model has been strongly accepted by the community, and is considered as a good one, in which proofs of security give a good taste of the security level. Even if it does not provide a formal proof of security (as in the standard model), it is argued that proofs in this model ensure security of the overall design of the scheme provided that the hash function has no weakness. Furthermore, assuming the tamper-resistance of some devices, such as smart cards, the random oracle model is equivalent to the standard model, simply requiring the existence of pseudo-random functions [37]. As a consequence, almost all the standard organizations introduce designs provably secure in that model, thanks to the security validation of very efficient protocols.

On the other hand, generic adversaries are not so realistic, because of their strong restrictiveness. Indeed, some non-generic algorithms exist for a long time against the discrete logarithm problem in the multiplicative groups of finite fields, such as the index calculus [32, 27]. However this generic model is still considered valid for elliptic curves [29] and hyper-elliptic curves [30, 31] settings. Non-generic algorithms appeared in some particular cases, such as the anomalous elliptic curves [52], the super-singular curves, where the discrete logarithm problem can be reduced to the finite field setting,

because of the Frobenius map which has a trace zero [36], as well as the curves of trace one [61] and more recently when many automorphisms exist on the curve [19]. However these curves were already advised not to be used in practice, but random curves. Anyway, this model seems a somewhat stronger model than the random oracle model.

3 The Intractability Assumptions

There are two major families in the number theory based public-key cryptography:

1. the schemes based on the integer factorization, and on the RSA problem;
2. the schemes based on the discrete logarithm problem, and on the Diffie-Hellman problems, in any "suitable" group. The first groups in use were cyclic subgroups of $\mathbb{Z}_p^\star$. But many schemes are now applied to cyclic subgroups of an elliptic curve, or of the Jacobian of an hyper-elliptic curve, with namely the so-called ECDSA [3], the DSA [40] on elliptic curves.

3.1 Integer Factorization and the RSA Problem

A first classical problem which is widely believed to be intractable is the integer factorization: it is clear that from two large primes p and q, it is easy to compute the product $N = pq$. However the inverting problem, recovering p and q from N, is hard to solve. Indeed, the Number Field Sieve technique [32] which the best known method is super-polynomial in the size of N. And it has been recently used to establish the record [14] of factoring a 512-bit number within three months.

A related problem is the well-known RSA problem on which was based the first public-key cryptosystem, the RSA function [51], proposed by Rivest, Shamir and Adleman in 1978: Given a composite modulus, $N = pq$, product of two large primes, and an exponent e, relatively prime to $\varphi(N)$, for any $x \in \mathbb{Z}_N^\star$, it is easy to compute $y = x^e \bmod N$. But for any $y \in \mathbb{Z}_N^\star$, it is difficult to compute the e-th root x, which satisfies $x^e = y \bmod N$, unless one knows the factorization of N, or equivalently $d = e^{-1} \bmod \varphi(N)$, since $x = y^d \bmod N$.

3.2 The Discrete Logarithm and the Diffie-Hellman Problems

The second family regroups DL-based problems in any suitable group. The setting is quite general: one is given

- a cyclic group $(\mathcal{G}, +)$ of order q, denoted additively;
- a generator $\mathbf{g}$.

We note in bold (such as $\mathbf{g}$) any element of the group $\mathcal{G}$, to distinguish it from a scalar $x \in \mathbb{Z}_q$. But such a $\mathbf{g}$ could be an element in $\mathbb{Z}_p^\star$ or a point of an elliptic curve, according to the setting. Above, we talked about a "suitable" group $\mathcal{G}$. In such a group, some of the following problems have to be hard to solve.

- the **Discrete Logarithm** problem (**DL**): given $\mathbf{y} \in \mathcal{G}$, compute $x \in \mathbb{Z}_q$ such that $\mathbf{y} = x \cdot \mathbf{g}$, then one denotes $x = \log_{\mathbf{g}} \mathbf{y}$.
- the **Computational Diffie-Hellman** problem (**CDH**): given two elements in the group $\mathcal{G}$, $\mathbf{a} = a \cdot \mathbf{g}$ and $\mathbf{b} = b \cdot \mathbf{g}$, compute $\mathbf{c} = ab \cdot \mathbf{g}$. Then one denotes $\mathbf{c} = \mathbf{DH}(\mathbf{a}, \mathbf{b})$.

– the **Decisional Diffie-Hellman** Problem (**DDH**): given three elements in the group $\mathcal{G}$, $\mathbf{a} = a \cdot \mathbf{g}$, $\mathbf{b} = b \cdot \mathbf{g}$ and $\mathbf{c} = c \cdot \mathbf{g}$, decide whether $\mathbf{c} = \mathbf{DH}(\mathbf{a}, \mathbf{b})$ (or equivalently, whether $c = ab \bmod q$).

It is clear that they are sorted from the strongest problem to the weakest one. Furthermore, one may remark that they all are "random self-reducible", which means that any instance can be reduced to a uniformly distributed instance: there are only average cases. Thus, the ability to solve a problem for a non-negligible fraction of instances in polynomial time is equivalent to solve any instance in polynomial expected time.

Very recently, Tatsuaki Okamoto and the author [45] defined a new variant of the Diffie-Hellman problem, which we called the *Gap Diffie-Hellman Problem*, where one wants to solve the **CDH** problem with an access to a **DDH** oracle.

Definition 1 (the Gap Diffie-Hellman Problem (GDH)). Given two elements in the group $\mathcal{G}$, $\mathbf{a} = a \cdot \mathbf{g}$ and $\mathbf{b} = b \cdot \mathbf{g}$, compute $\mathbf{c} = ab \cdot \mathbf{g}$, with the access to a Decisional Diffie-Hellman oracle.

One may easily remark the following properties about above problems. Indeed, using the oracle notation from the complexity theory, one can see that $\mathbf{GDH} = \mathbf{CDH}^{\mathbf{DDH}}$.

Property 2.

$$\mathbf{DL} \geq \mathbf{CDH} \geq \mathbf{DDH}$$

$$\mathbf{DDH} \text{ easy} \iff \mathbf{GDH} = \mathbf{CDH}$$

$$\mathbf{GDH} \text{ easy} \iff \mathbf{CDH} = \mathbf{DDH}$$

For signature applications, one only requires groups where the DL problem is hard, whereas encryption needs trapdoor problems and therefore requires groups where some of the **DH**'s problems are also hard to solve.

However, the **CDH** problem is usually believed to be much stronger than the **DDH** problem, which means that the **GDH** problem is difficult. This was the motivation of our work on new encryption schemes based on the **GDH** problem [44] (see section 4.4.2).

3.3 Proofs of Security

Until 1996, no practical DL-based cryptographic scheme has ever been formally studied, but just heuristically. And surprisingly, at Eurocrypt '96, two opposite studies have been driven on the El Gamal signature [20], the first DL-based signature designed in 1985 and depicted on Figure 1. Whereas existential forgeries were known for that scheme, it was believed to prevent universal forgeries. The first analysis, from Daniel Bleichenbacher [11], showed such a universal forgery when the generator g is

– **Initialization** g a generator of $\mathbb{Z}_p^\star$, where p is a large prime secret key $x \in \mathbb{Z}_{p-1}^\star$ public key $y = g^x \bmod p$	– **Signature** – $K \in \mathbb{Z}_{p-1}^\star$ and $r = g^K \bmod p$ – $s = (m - xr)/K \bmod p - 1$ – **Verification** $g^m \stackrel{?}{=} y^r r^s \bmod p$

Fig. 1. The El Gamal Signature Scheme.

not properly chosen. The second one, from Jacques Stern and the author [47], proved the security, against existential forgeries under adaptively chosen-message attacks, of a slight variant with a randomly chosen generator g. The slight variant simply replaces the message m by $H(m, r)$ in the computation, while one uses a hash function H that behaves like a random oracle. It is amazing to remark that the Bleichenbacher's attack also applies on our variant. Therefore, according to the initialization, our variant could be a very strong signature scheme or become a very weak one!

About encryption, the first efficient construction with a formal proof of security has been proposed by Bellare and Rogaway [8] in 1994. This construction, the so-called OAEP, was claimed to apply to any trapdoor one-way permutation, as described on Figure 2. However, in the late November 2000, this claim appeared to be wrong [58].

– **Initialization** f a trapdoor one-way permutation over the space $\{0,1\}^k$, g is the inverse, thanks to the trapdoor $k = n + k_0 + k_1$ secret key g public key f	– **Encryption** of $m \in \{0,1\}^n$ – $r \in \{0,1\}^{k_0}$ – $s = m\|0^{k_1} \oplus G(r)$ and $t = r \oplus H(s)$ – $c = f(s\|t)$ – **Decryption** of $c \in \{0,1\}^k$ – $(s,t) = g(c)$ – $r = H(s) \oplus t$ and $M = s \oplus G(r)$ – if $M = m\|0^{k_1}$ for some m, returns m, otherwise, returns Reject.

Fig. 2. The OAEP Construction.

Anyway Eiichiro Fujisaki, Tatsuaki Okamoto, Jacques Stern and the author immediately provided a new proof [24], but then for trapdoor *partially* one-way permutations, a stronger requirement about the permutation. In other words, the OAEP construction is still secure if recovering more than half of the bits of the pre-image by f is intractable. Anyway, RSA is the sole trapdoor permutation, and thus the sole practical application of the OAEP conversion. And for RSA, the partial one-wayness is equivalent to the one-wayness, thanks to the random self-reducibility. Therefore, the RSA–OAEP is still secure relatively to the RSA problem.

Anyway, both examples show that a proof has to be performed in details. Furthermore, the conclusions have to be strictly followed by implementators, otherwise the concrete implementation of a secure scheme can be very weak.

4 Public-Key Encryption

The aim of a public-key encryption is to allow anybody who knows the public key of Alice to send her a message that she will be the only one able to recover it, thanks to her private key.

4.1 Definitions

A public-key encryption scheme is defined by the three following algorithms:

– The *key generation algorithm* G. On input 1^k, where k is the security parameter, the algorithm G produces a pair $(\mathsf{k_p}, \mathsf{k_s})$ of matching public and secret keys. Algorithm G is probabilistic.

– The *encryption algorithm* $\mathcal{E}$. Given a message m and a public key $\mathsf{k_p}$, $\mathcal{E}$ produces a ciphertext c of m. This algorithm may be probabilistic.
– The *decryption algorithm* $\mathcal{D}$. Given a ciphertext c and the secret key $\mathsf{k_s}$, $\mathcal{D}$ gives back the plaintext m. This algorithm is necessarily deterministic.

4.2 Security Notions

As for signature schemes, the goals of the adversary may be various. The first common security notion that one would like for an encryption scheme is the *one-wayness*: with just public data, an attacker cannot get back the whole plaintext of a given ciphertext. More formally, this means that for any adversary $\mathcal{A}$, her success in inverting $\mathcal{E}$ without the secret key should be negligible over the probability space $\mathcal{M} \times \Omega$, where $\mathcal{M}$ is the message space and Ω is the space of the random coins r used for the encryption scheme, and the internal random coins of the adversary:

$$\mathsf{Succ}_{\mathcal{A}} = \Pr_{m,r}[(\mathsf{k_p}, \mathsf{k_s}) \leftarrow G(1^k) : \mathcal{A}(\mathsf{k_p}, \mathcal{E}(\mathsf{k_p}, m; r)) = m].$$

However, many applications require more from an encryption scheme, namely the *semantic security* (a.k.a. *polynomial security/indistinguishability of encryptions* [25]): if the attacker has some information about the plaintext, for example that it is either "yes" or "no" to a crucial query, any adversary should not learn more with the view of the ciphertext. This security notion requires the computational impossibility to distinguish between two messages, chosen by the adversary, which one has been encrypted, with a probability significantly better than one half: her advantage $\mathsf{Adv}_{\mathcal{A}}$, as defined below where the adversary $\mathcal{A}$ is seen as a 2-stage Turing machine $(\mathcal{A}_1, \mathcal{A}_2)$, should be negligible.

$$\mathsf{Adv}_{\mathcal{A}} = 2 \times \Pr_{b,r}\left[\begin{array}{l}(\mathsf{k_p}, \mathsf{k_s}) \leftarrow G(1^k) \\ (m_0, m_1, s) \leftarrow \mathcal{A}_1(\mathsf{k_p}) \\ c = \mathcal{E}(\mathsf{k_p}, m_b; r)\end{array} : \mathcal{A}_2(m_0, m_1, s, c) = b\right] - 1.$$

Another notion has been thereafter defined, the so-called *non-malleability* [18], but this notion is equivalent to the above one in some specific scenarios [10]. Moreover, it is equivalent to the semantic security [6] in the most interesting scenario. Therefore, we will just focus on the one-wayness and the semantic security.

On the other hand, an attacker can play many kinds of attacks: she may just have access to public data, and then encrypt any plaintext of her choice (*chosen-plaintext attacks*) or moreover query the decryption algorithm on any ciphertext of her choice, except the challenge ciphertext (*adaptively/non-adaptively chosen-ciphertext attacks* [38, 49]). Recently, an intermediate attack has been described where the adversary has access to an oracle which, on input a pair (m, c), answers whether c encrypts the message m. This attack has been named the *Plaintext-Checking Attack* [44]:

Definition 3 (Plaintext-Checking Attack (PCA)). In the Plaintext-Checking scenario, the adversary may ask any pair (m, c) of her choice to a Plaintext-Checking Oracle that answers whether c encrypts m or not.

Furthermore, multi-user scenarios can be considered where related messages are encrypted under different keys to be sent to many people (*e.g.* broadcast of encrypted data). This may provide many useful data for an adversary. For example, RSA is well-known to be weak in such a scenario [28, 57], namely with a small encryption

exponent, using the Chinese Remainderings Theorem. But recent results prove that semantically secure schemes, in the classical sense as described above, remain secure in multi-user scenarios [5, 4], whatever the kind of attacks.

A general study of these security notions and attacks has been driven in [6], we therefore refer the reader to this paper for more details. However, we can just review the scenarios we will be interested in in the following:

- one-wayness under chosen-plaintext attacks (OW-CPA) – where the adversary wants to recover the whole plaintext from just the ciphertext and the public key. This is the weakest scenario.
- one-wayness under plaintext-checking attacks (OW-PCA) – where the adversary wants to recover the whole plaintext with an access to a Plaintext-Checking Oracle that answers, for any pair (m, c), whether c encrypts m or not. This is also a very weak scenario.
- semantic security under adaptively chosen-ciphertext attacks (IND-CCA) – where the adversary just wants to distinguish which plaintext, between two messages of her choice, has been encrypted, while she can ask any query she wants to a decryption oracle (except the challenge ciphertext). This is the strongest scenario, and thus our goal when we design a cryptosystem.

4.3 Some Examples

The RSA Encryption. As already said, in 1978, Rivest–Shamir–Adleman [51] proposed a public-key encryption, thanks to the "trapdoor one-way permutation" property of the RSA function: the generation algorithm produces a large composite number $N = pq$, a public key e, and a secret key d such that $e \cdot d = 1 \bmod \varphi(N)$. The encryption of a message m, encoded as an element in $\mathbb{Z}_N^\star$, is simply $c = m^e \bmod N$. This ciphertext can be easily decrypted thanks to the knowledge of d, $m = c^d \bmod N$. Clearly, this encryption is OW-CPA, relative to the RSA problem. The determinism makes a PCA-oracle unuseful (its answer would be unrelevant), therefore the RSA-encryption scheme is OW-PCA relative to the RSA problem.

However, since it is deterministic, it cannot be semantically secure: given the encryption c of either m_0 or m_1, the adversary simply computes $c' = m_0^e \bmod N$ and checks whether $c' = c$. Furthermore, as said above, with a small exponent e (*e.g.* $e = 3$), any security vanishes under a multi-user attack: given $c_1 = m^3 \bmod N_1$, $c_2 = m^3 \bmod N_2$ and $c_3 = m^3 \bmod N_3$, one can easily compute $m^3 \bmod N_1 N_2 N_3$ thanks to the Chinese Remainderings Theorem, which is exactly m^3 in $\mathbb{Z}$ and therefore leads to an easy recovery of m.

The El Gamal Encryption. In 1985, El Gamal [20] designed a public-key encryption scheme based on the Diffie-Hellman key exchange protocol [17]: given a cyclic group $\mathcal{G}$ of order q and a generator $\mathbf{g}$, the generation algorithm produces a random element $x \in \mathbb{Z}_q^\star$ as secret key, and a public key $\mathbf{y} = x \cdot \mathbf{g}$. The encryption of a message m, encoded as an element $\mathbf{m}$ in $\mathcal{G}$, is a pair $(\mathbf{c} = a \cdot \mathbf{g}, \mathbf{d} = a \cdot \mathbf{y} + \mathbf{m})$. This ciphertext can be easily decrypted thanks to the knowledge of x, since $a \cdot \mathbf{y} = x \cdot \mathbf{c}$, $\mathbf{m} = \mathbf{d} - x \cdot \mathbf{c}$. This encryption scheme is well-known to be OW-CPA relative to the Computational Diffie-Hellman problem. It is also semantically secure (under chosen-plaintext attacks) relative to the Decisional Diffie-Hellman problem [62]. About OW-PCA, it relies on the new Gap Diffie-Hellman problem [45].

4.4 Secure Designs

As we have seen above, the expected security level is IND-CCA, whereas the RSA encryption just reaches OW-CPA under the RSA assumption, and the El Gamal encryption achieves IND-CPA under the **DDH** assumption. Can we achieve IND-CCA for practical encryption schemes?

In 1994, Bellare and Rogaway [8] proposed a generic conversion, the "Optimal Asymmetric Encryption Padding" (OAEP), which was claimed to apply to any trapdoor one-way permutation, such as RSA, in the random oracle model. Eventually, it has just been shown to apply to trapdoor partially one-way permutations [58, 24]. Anyway, there was already no hope to use it with any DL-based primitive, because of the "permutation" requirement.

Hopefully, first Fujisaki and Okamoto [22] proposed a generic conversion from any IND-CPA scheme into an IND-CCA one. While applying this conversion to the above El Gamal encryption, one obtains an IND-CCA encryption scheme relative to the **DDH** problem, in the random oracle model. After, independently, Fujisaki and Okamoto [23] and the author [46] proposed better generic conversions since they apply to any OW-CPA scheme to make it into an IND-CCA one, under the same assumption.

This high security level is just at the cost of two more hashings for the new encryption, as well as two more hashings and one re-encryption for the new decryption process. This re-encryption cost is the main drawback of these conversions. Therefore, with Tatsuaki Okamoto, we tried and succeeded in providing a both secure and efficient conversion [44]: REACT, for "Rapid Enhanced-security Asymmetric Cryptosystem Transform". This latter conversion is indeed very efficient in many senses

- the computational overhead is just the cost of two hashings for both encryption and decryption
- if one can break IND-CCA of the resulting scheme with an expected time T, one can break OW-PCA of the basic scheme within almost the same amount of time.

Anyway, both above conversions apply to any OW-CPA (or OW-PCA resp.) public-key encryption scheme. And the El Gamal encryption is a very nice candidate, under the Diffie-Hellman problems (the **CDH** problem, or the **GDH** problem resp.). Let us describe both generic conversions [46, 44] on any encryption scheme $\mathcal{S} = (G, \mathcal{E}, \mathcal{D})$

$$\mathcal{E} : \mathcal{PK} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C} \qquad \mathcal{D} : \mathcal{SK} \times \mathcal{C} \to \mathcal{M},$$

where $\mathcal{PK}$ and $\mathcal{SK}$ are the sets of the public and secret keys, $\mathcal{M}$ is the message-space, $\mathcal{C}$ is the ciphertext-space and $\mathcal{R}$ is the random-space. One should remark that $\mathcal{R}$ may be small and even empty, with a deterministic encryption scheme, such as RSA. But in many other cases, such as the El Gamal encryption, it is as large as $\mathcal{M}$. Anyway, as we will see later, if required, it can be made as huge as necessary.

4.4.1 The First Conversion. In the conversion [46], we need, as for OAEP [8], two functions, a hash function H and a generator function G, both assumed to be ideal random functions [7],

$$H : \{0,1\}^k \to \mathcal{R} \qquad G : \mathcal{M} \to \{0,1\}^k,$$

where k is a security parameter. The cryptosystem is depicted on Figure 3, with $k = \ell + \kappa$, where ℓ and κ denote the length of the messages to be encrypted and the error-parameter respectively: the new message-space $\mathcal{M}'$ can be identified to $\{0,1\}^\ell$, and

G': **Key Generation**
$(\mathsf{k_p}, \mathsf{k_s}) \leftarrow G(1^k)$ $\longrightarrow (\mathsf{k_p}, \mathsf{k_s})$
$\mathcal{E}'$: **Encryption of** $\mathbf{m} \in \mathcal{M}' = \{\mathbf{0},\mathbf{1}\}^\ell \rightarrow (\mathbf{a},\mathbf{b})$
$r \in \mathcal{M}$ and $s \in \{0,1\}^\kappa$ are randomly chosen $a = \mathcal{E}(\mathsf{k_p}, r; H(m\|s))$ $\quad$ $b = (m\|s) \oplus G(r)$ $\longrightarrow (a,b)$ is the ciphertext
$\mathcal{D}'$: **Decryption of** $(\mathbf{a},\mathbf{b})$
Given $a \in \mathcal{C}$ and $b \in \{0,1\}^k$ $r = \mathcal{D}(\mathsf{k_s}, a)$ $\quad$ $M = b \oplus G(r)$ if $a = \mathcal{E}(\mathsf{k_p}, r; H(M)) \longrightarrow m = [M]_\ell$ is the plaintext (otherwise, "Reject: invalid ciphertext")

Fig. 3. Generic Converted Encryption Scheme $\mathcal{S}'$

this conversion decreases the success probability, in the reduction from the underlying problem to an attack, by a value negligible in κ. The notation $[M]_\ell$ id used for the truncation of the bit-string M to its ℓ most significant bits.

About the new scheme $\mathcal{S}'$, one can show that, under some assumptions about the original encryption scheme $\mathcal{S}$, an attacker against semantic security under an adaptively chosen-ciphertext attack can be used to efficiently invert $\mathcal{E}$: in other words, OW-CPA of $\mathcal{S}$ implies IND-CCA of $\mathcal{S}'$, in the random oracle model. Let us formally prove this important result, which first relies on the following lemma.

Lemma 4. *Let $\mathcal{A}$ be an attacker against the semantic security of $\mathcal{S}'$ in a chosen-plaintext scenario. If we denote by ε the advantage of this attacker after q_H queries to the random oracle H, one can design an algorithm $\mathcal{B}$ that outputs, for any given c, a set S of values such that the plaintext of c is in S with probability greater than $\varepsilon/2 - q_H/2^\kappa$.*

To understand this lemma, one just has to remark that the adversary has to have asked either $H(m\|s)$ or $G(r)$ to get any information about the plaintext m. However, the probability to have guessed the correct s after q_H queries to H is less than $q_H/2^\kappa$. If one outputs (α, β), as the encryption of either m_0 or m_1, where α is the value c on which one wants to invert $\mathcal{E}$ and β a random bit-string, then the probability that the adversary has asked for $G(\rho)$ is greater than $\varepsilon/2 - q_H/2^\kappa$ (on all the random coins and on the pair (α, β) itself, and thus on c.) In that case, the plaintext ρ of α (and thus of c) is in the list of queries asked to G.

However, if one wants to deal with the chosen-ciphertext attacks, one has to simulate the decryption oracle. But this can be easily done by someone who has access to the list of the query-answer pairs from G and H: let (a, b) be a ciphertext asked by the adversary to the decryption oracle. One takes all the queries r asked to G whose answer has been g_r and checks if the value $M = b \oplus g_r$ has been asked to H. To a positive answer, one finally checks whether $a = \mathcal{E}(\mathsf{k_p}, r; H(M))$ or not. Therefore, a really encrypted message will be accepted and correctly decrypted, but the adversary may have sent a valid ciphertext (a, b), by chance: with an a that is really equal to $\mathcal{E}(\mathsf{k_p}, r; H(G(r) \oplus b))$.

As shown by recent works [58, 24], a complete proof has to be driven. Therefore, let us use greek letters for the challenge ciphertext (α, β): ρ, σ for the implicitly random

values. If one assumes $\mathcal{E}(\mathsf{k_p}, \cdot; \cdot)$ to be an injective map for any key $\mathsf{k_p}$ (on the input space $\mathcal{M} \times \mathcal{R}$), then the preimage of α is a pair (ρ, τ). Eventually, we denote by μ the padded plaintext, and thus $\mu = m_b \| \sigma$. Let us be given a ciphertext (a, b) to decrypt, with the associated random r, s, as well as (r, t) for the pre-image of a, and M the padded plaintext. We have several events to define:

- AskG and AskH the events that ρ, and μ, have been asked to G, and H respectively;
- AskR and AskM the events that r, and M, have been asked to G, and H respectively;
- BadR and BadM the events that $r = \rho$, and $M = \mu$ respectively.

First, as already remarked, an accepted and decrypted ciphertext by above simulation is necessarily correct. Therefore, an incorrect decryption (a failure, denoted by the event F) may just come from a rejection of a valid ciphertext. Let us evaluate the probability p of this event, provided that $\neg\mathsf{AskG}$: ρ has not been asked to G (the preimage ρ of α by $\mathcal{E}$ is not in the list of the queries asked to G.) Then we use the notation pr for probabilities provided that $\neg\mathsf{AskG}$, and let us split the failure event according to AskR:

$$p = \mathsf{pr}[\mathsf{F} \wedge \mathsf{AskR}] + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR}].$$

Before any further analysis, let us remark that some events cannot happen at the same time: BadR and BadM together would necessarily imply that the message and the random values are the same as for the challenge ciphertext, which would lead to the same ciphertext. However it is prohibited to the adversary to ask this challenge ciphertext to the decryption oracle. Thus

$$\mathsf{BadR} \Rightarrow \neg\mathsf{BadM} \text{ and } \mathsf{BadM} \Rightarrow \neg\mathsf{BadR}.$$

Let us go back to the analysis of the probability p. The former event can be split according to AskM. But as we have already remarked, event $\mathsf{AskR} \wedge \mathsf{AskM}$ necessarily implies a correct decryption. Therefore

$$\begin{aligned}
\mathsf{pr}[\mathsf{F} \wedge \mathsf{AskR}] &= \mathsf{pr}[\mathsf{F} \wedge \mathsf{AskR} \wedge \neg\mathsf{AskM}] \\
&= \mathsf{pr}[\mathsf{F} \wedge \mathsf{AskR} \wedge \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] + \mathsf{pr}[\mathsf{F} \wedge \mathsf{AskR} \wedge \neg\mathsf{AskM} \wedge \mathsf{BadM}] \\
&\leq \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] + \mathsf{pr}[\mathsf{BadM} \wedge \neg\mathsf{BadR}] \\
&\leq \mathsf{pr}[\mathsf{F} \mid \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] + \Pr[\mathsf{BadM} \mid \neg\mathsf{BadR} \wedge \neg\mathsf{AskG}] \\
&\leq \Pr[H(M) = t \mid \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] + \Pr[G(r) = \mu \oplus b \mid \neg\mathsf{AskG} \wedge \neg\mathsf{BadR}] \\
&\leq 1/\#\mathcal{R} + 1/2^k
\end{aligned}$$

Now, let us consider the second part which we can split according to BadR:

$$\mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR}] = \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \mathsf{BadR}] + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR}],$$

where the former event can furthermore be split according to AskM:

$$\begin{aligned}
\mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \mathsf{BadR}] &= \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \mathsf{BadR} \wedge \mathsf{AskM}] \\
&\quad + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \mathsf{BadR} \wedge \neg\mathsf{AskM}] \\
&\leq \mathsf{pr}[\mathsf{AskM} \wedge \neg\mathsf{AskR} \wedge \mathsf{BadR}] + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskM} \wedge \mathsf{BadR}] \\
&\leq \mathsf{pr}[\mathsf{AskM} \mid \neg\mathsf{AskR} \wedge \mathsf{BadR}] + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] \\
&\leq \Pr[\mathsf{AskM} \mid \mathsf{BadR} \wedge \neg\mathsf{AskG}] + \mathsf{pr}[\mathsf{F} \mid \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] \\
&\leq q_H/2^\kappa + 1/\#\mathcal{R}.
\end{aligned}$$

About the latter, we can also split it according to AskM: $\mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR}]$

$$
\begin{aligned}
&= \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR} \wedge \mathsf{AskM}] + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR} \wedge \neg\mathsf{AskM}] \\
&\leq \mathsf{pr}[\mathsf{AskM} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR}] + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR} \wedge \neg\mathsf{AskM} \wedge \mathsf{BadM}] \\
&\quad + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR} \wedge \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] \\
&\leq \mathsf{pr}[\mathsf{AskM} \mid \neg\mathsf{AskR} \wedge \neg\mathsf{BadR}] \\
&\quad + \mathsf{pr}[\mathsf{BadM} \wedge \neg\mathsf{AskR} \wedge \neg\mathsf{BadR}] + \mathsf{pr}[\mathsf{F} \wedge \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] \\
&\leq \mathsf{pr}[\mathsf{AskM} \mid \neg\mathsf{AskR} \wedge \neg\mathsf{BadR}] \\
&\quad + \mathsf{pr}[\mathsf{BadM} \mid \neg\mathsf{AskR} \wedge \neg\mathsf{BadR}] + \mathsf{pr}[\mathsf{F} \mid \neg\mathsf{AskM} \wedge \neg\mathsf{BadM}] \\
&\leq q_H/2^k + 1/2^k + 1/\#\mathcal{R}.
\end{aligned}
$$

Therefore,

$$p \leq \frac{3}{\#\mathcal{R}} + \frac{q_H + 2}{2^k} + \frac{q_H}{2^\kappa}.$$

This even provides a plaintext-extractor [8]. Then combining both results, one obtains

$$\Pr[\mathsf{AskG}] \geq \frac{\varepsilon}{2} - \frac{q_H}{2^\kappa} - q_D \left(\frac{3}{\#\mathcal{R}} + \frac{q_H + 2}{2^k} + \frac{q_H}{2^\kappa} \right) - \Pr[\mathsf{AskG}].$$

That allows us to state the following result, which encompasses chosen-ciphertext attacks, under the assumption that for any public key $\mathsf{k_p} \in \mathcal{PK}$, the encryption function $\mathcal{E}_{\mathsf{k_p}} : \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ is an injection.

Theorem 5. *Let $\mathcal{A}$ be an attacker against the semantic security of $\mathcal{S}'$ in a chosen-ciphertext scenario. If we denote by ε the advantage of this attacker after q_H queries to the random oracle H and q_D queries to the decryption oracle, one can design an algorithm $\mathcal{B}$ that outputs, for any given c, a set S of values such that the plaintext of c is in S with probability greater than*

$$\frac{\varepsilon}{4} - \frac{1}{2} \times \left(\frac{(q_D + 1)q_H}{2^\kappa} + \frac{q_D(q_H + 2)}{2^k} + \frac{3q_D}{\#\mathcal{R}} \right).$$

Remark 6. One may remark that if $\mathcal{R}$ is too small, which can even be empty in the case of a fully trapdoor function, above result is meaningless. However, one can easily extend $\mathcal{R}$: *e.g.* $\mathcal{E}_2(\mathsf{k_p}, m; r_1 \| r_2) \leftarrow \mathcal{E}_1(\mathsf{k_p}, m; r_1) \| r_2$.

In general, randomly choosing an element in the set S which is the set of the queries asked to G, one can invert $\mathcal{E}$ with success probability greater than

$$\frac{1}{4 \cdot q_G} \times \left(\varepsilon - \left(\frac{2(q_D + 1)q_H}{2^\kappa} + \frac{2q_D(q_H + 2)}{2^k} + \frac{6q_D}{\#\mathcal{R}} \right) \right),$$

where q_H, q_G and q_D denote the number of queries asked to H, G and $\mathcal{D}$ respectively. But such a reduction can be improved in the case of a random self-reducible problem, such as the Diffie-Hellman problem, using the Shoup's theorem [59] about the faulty Diffie-Hellman oracles. It can also be improved under a gap-problem assumption [45], and thus relative to the OW-PCA of the underlying cryptosystem $\mathcal{S}$.

4.4.2 A Rapid Enhanced-security Asymmetric Cryptosystem Transform

In the second REACT conversion [44], we also need two hash function G and H, both assumed again to behave like random functions [7],

$$G : \mathcal{M} \to \{0,1\}^{\ell} \qquad H : \{0,1\}^{\star} \to \{0,1\}^{\kappa},$$

where κ is the security parameter, while ℓ denotes the size of the messages to encrypt. The REACT conversion is depicted on Figure 4.

G': **Key Generation**
$(\mathsf{k_p}, \mathsf{k_s}) \leftarrow G(1^k)$ $\longrightarrow (\mathsf{k_p}, \mathsf{k_s})$
$\mathcal{E}'$: **Encryption of** $\mathbf{m} \in \mathcal{M}' = \{\mathbf{0},\mathbf{1}\}^{\ell} \to (\mathbf{a},\mathbf{b},\mathbf{c})$
$R \in \mathcal{M}$ and $r \in \mathcal{R}$ are randomly chosen $a = \mathcal{E}(\mathsf{k_p}, R; r) \qquad b = m \oplus G(R) \qquad c = H(R, m, a, b)$ $\longrightarrow (a, b, c)$ is the ciphertext
$\mathcal{D}'$: **Decryption of** $(\mathbf{a},\mathbf{b},\mathbf{c})$
Given $a \in \mathcal{C}$, $b \in \{0,1\}^{\ell}$ and $c \in \{0,1\}^{\kappa}$ $R = \mathcal{D}(\mathsf{k_s}, a) \qquad m = b \oplus G(R)$ if $c = H(R, m, a, b)$ and $R \in \mathcal{M} \longrightarrow m$ is the plaintext (otherwise, "Reject: invalid ciphertext")

Fig. 4. Rapid Enhanced-security Asymmetric Cryptosystem Transform $\mathcal{S}'$

About this new scheme $\mathcal{S}'$, one can show that, an attacker against semantic security under an adaptively chosen-ciphertext attack can be used to efficiently invert $\mathcal{E}$, while just asking for the validity of plaintext-ciphertext relations: in other words, OW-PCA of $\mathcal{S}$ implies IND-CCA of $\mathcal{S}'$, in the random oracle model [7]. Based on this new PCA scenario [44] and on the gap-problems [45], one can claim the following security result.

Theorem 7. *Let $\mathcal{A}$ be an attacker against the semantic security of $\mathcal{S}'$ in a chosen-ciphertext scenario. If we denote by ε the advantage of this adversary after q_D, q_G and q_H queries to the decryption oracle and to the random oracles G and H respectively, one can design an algorithm $\mathcal{B}$ that outputs, for any given C, the plaintext of C, after less than $q_G + q_H$ queries to the Plaintext-Checking Oracle with probability greater than $\varepsilon/2 - q_D/2^{\kappa}$.*

As for the previous conversion, that security result comes from two distincts remarks:

- the adversary has to have asked either $G(R)$ or $H(R, m, a, b)$ to get any information about m. Which means that for a given $C = \mathcal{E}(\mathsf{k_p}, R; r)$, R is in the list of queries asked to G or to H. Simply asking for the $q_G + q_H$ candidates to the Plaintext-Checking Oracle, one can output the right one. Then, with probability $\varepsilon/2$, one inverts $\mathcal{E}$, after $(q_G + q_H)$ queries to the Plaintext-Checking Oracle.
- However, in the chosen-ciphertext scenario, the adversary may ask queries to the decryption oracle. We have to simulate it. To any query (a, b, c) asked by the adversary to the decryption oracle, one looks at all the pairs (R, m) such that

$\mathcal{G}$ a group of order q $\mathbf{g}$ a generator of $\mathcal{G}$ $G : \mathcal{G} \to \{0,1\}^\ell$ and $H : \{0,1\}^\star \to \{0,1\}^\kappa$ $(E, D) : \{0,1\}^\ell \times \{0,1\}^\star \to \{0,1\}^\star$ a symmetric encryption scheme
Secret Key: $x \in \mathbb{Z}_q$ Public Key: $\mathbf{y} = x \cdot \mathbf{g}$
Encryption
$\mathbf{r} \in_R \mathcal{G}$ and $r \in \mathbb{Z}_q$ $\mathcal{E}(\mathbf{y}, m; \mathbf{r} \Vert r) = \begin{cases} \mathbf{a} = r \cdot \mathbf{g} & \mathbf{b} = r \cdot \mathbf{y} + \mathbf{r} \\ b = E(G(\mathbf{r}), m) & c = H(\mathbf{r}, m, \mathbf{a}, b) \end{cases}$
Decryption
$\mathcal{D}(x, (\mathbf{a}, \mathbf{b}, b, c)) = \begin{cases} \mathbf{r} = \mathbf{b} - x \cdot \mathbf{a} & t = D(G(\mathbf{r}), b) \\ \text{if } c = H(\mathbf{r}, m, \mathbf{a}, b) & \text{then } m = t \end{cases}$

Fig. 5. The REACT-EG Encryption Scheme

(R, m, a, b) has been asked to the random oracle H. For any such R, one asks to the Plaintext-Checking Oracle whether a is a ciphertext of R (remark that it does not make more queries to the Plaintext-Checking Oracle, since it has already been taken in account above). Then it computes $K = G(R)$, maybe using a simulation of G if the query R has never been asked. If $b = K \oplus m$ then one outputs m as the plaintext of the triple (a, b, c). Therefore, any correctly computed ciphertext is decrypted by the simulator. But if the adversary has not asked $H(R, m, a, b)$ the probability that the ciphertext is valid, and thus the decryption not correctly simulated, is less than $1/2^\kappa$, if the adversary correctly guesses the value of $H(m, R, a, b)$.

4.5 Hybrid Conversions

In both conversions, one can improve efficiency. Indeed, in both cases, we have computed some $b = m \oplus K$, where K can be seen as a session key used in a one-time pad encryption scheme. The one-time pad is well-known to be a perfect encryption scheme, but in those conversions, it could be replaced by any symmetric encryption scheme that is just semantically secure (under no plaintext nor ciphertext attacks). Therefore, plaintexts of any size could be encrypted using those conversions.

4.6 The REACT-EG Encryption Scheme

4.6.1 Description. If one applies that latter REACT conversion to the famous El Gamal encryption scheme [20], one gets the scheme presented on Figure 5, where (E, D) is any symmetric encryption scheme whose keys are ℓ-bit strings. This hybrid scheme achieves the following security properties.

Theorem 8 (The REACT-EG Encryption Scheme). *Let $\mathcal{A}$ be an IND-CCA against the REACT-EG Encryption Scheme running within time bound T. If we denote by ε the advantage of this adversary after q_D, q_G and q_H queries to the decryption oracle and the hash functions G and H respectively, one can design an algorithm $\mathcal{B}$ that, within time $T' \leq T + (q_G + q_H) \times \mathcal{O}(1)$, for any $\nu < \varepsilon$,*

– *either breaks the* ***GDH*** *Problem after less than* $q_G + q_H$ *queries to the* ***DDH****-oracle, where*

$$\delta = \frac{\varepsilon - \nu}{2} - \frac{q_D}{2^\kappa},$$

– *or gets an advantage greater than* ν *in breaking the semantic security of the symmetric encryption scheme* (E, D)*.*

In practice, one would choose a group $\mathcal{G}$ of order q on 160 bits, with a security parameter $\kappa = 80$. If we consider that the adversary cannot make more than 2^{64} calls to oracles, and even less to the decryption oracle $\mathcal{D}$, then a successful attack in expected time T would lead to an algorithm that solves the Gap Diffie-Hellman problem within similar time, unless the symmetric encryption scheme is weak. That is optimal!

4.6.2 Comparaison with Previous DH-Based Proposals. We focused on this application for REACT, because many variants have been proposed for the El Gamal encryption. However REACT provides the most efficient one. Indeed, many El Gamal variants have already been proposed in the past, trying to deal with chosen-ciphertext attacks. First, at PKC '98, Tsiounis and Yung [62] proposed a variant secure against chosen-ciphertext attacks, in the random oracle model. However, it was furthermore based on both the Decisional Diffie-Hellman problem and an assumption about the unforgeability of Schnorr signatures [54], which can only be proven in the generic model [55]. Thereafter Shoup and Gennaro [60] proposed a new variant provably secure against chosen-ciphertext attacks in the random oracle model, under the sole assumption of the Decisional Diffie-Hellman problem. For both of them, efficiency was a serious drawback: twice, or even more, slower than our proposals.

The most famous is of course the Cramer-Shoup variant [16], which is provably secure in the standard model. However, the security relies on the Decisional Diffie-Hellman problem and is rather slow, still more than twice as slow as ours.

Before this candidate, DHAES [1] was the most efficient El Gamal-like public-key encryption scheme which reached IND-CCA security. However, this security, whereas it does not require the random oracle model [7], is based on a non-standard assumption, the Oracle Diffie-Hellman Assumption [2], which is somewhat as strong as the random oracle model, and furthermore requires a MAC.

Consequently, the REACT-EG variant is among the most efficient, since it is almost as efficient as the original El Gamal scheme. Indeed, it just performs two more hashings. Furthermore, it is semantically secure against adaptively chosen-ciphertext attacks under the sole assumption of the Gap Diffie-Hellman problem in the random oracle model.

5 Conclusion

In this paper, we reviewed the security notions for encryption, as well as some security proofs. Furthermore, we have seen many generic conversions which apply to weakly secure encryption schemes to make strongly secure ones. The last one, REACT, is the most efficient and the most general one. Applied to the El Gamal encryption scheme, it leads to the most efficient DL-based cryptosystem, based on the Gap Diffie-Hellman problem.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. Submission to IEEE P1363a. September 1998.
   Available from `http://grouper.ieee.org/groups/1363/`.
2. M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *RSA '2001*, LNCS. Springer-Verlag, Berlin, 2001.
3. American National Standards Institute. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm. ANSI X9.62-1998. January 1999.
4. O. Baudron, D. Pointcheval, and J. Stern. Extended Notions of Security for Multicast Public Key Cryptosystems. In *Proc. of the 27th ICALP*, LNCS 1853, pages 499–511. Springer-Verlag, Berlin, 2000.
5. M. Bellare, A. Boldyreva, and S. Micali. Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements. In *Eurocrypt '2000*, LNCS 1807, pages 259–274. Springer-Verlag, Berlin, 2000.
6. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.
7. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.
8. M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.
9. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.
10. M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, Berlin, 1999.
11. D. Bleichenbacher. Generating El Gamal Signatures without Knowing the Secret Key. In *Eurocrypt '96*, LNCS 1070, pages 10–18. Springer-Verlag, Berlin, 1996.
12. S. A. Brands. An Efficient Off-Line Electronic Cash System Based on the Representation Problem. Technical Report CS-R9323, CWI, Amsterdam, 1993.
13. R. Canetti, O. Goldreich, and S. Halevi. The Random Oracles Methodology, Revisited. In *Proc. of the 30th STOC*, pages 209–218. ACM Press, New York, 1998.
14. S. Cavallar, B. Dodson, A. K. Lenstra, W. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, C. Putnam, C. Putnam, and P. Zimmermann. Factorization of a 512-bit RSA Modulus. In *Eurocrypt '2000*, LNCS 1807, pages 1–18. Springer-Verlag, Berlin, 2000.
15. B. Chor and R. L. Rivest. A Knapsack Type Public Key Cryptosystem based on Arithmetic in Finite Fields. In *Crypto '84*, LNCS 196, pages 54–65. Springer-Verlag, Berlin, 1985.
16. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.
17. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.
18. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.
19. I. Duursma, P. Gaudry, and F. Morain. Speeding Up the Discrete Log Computation on Curves with Automorphisms. In *Asiacrypt '99*, LNCS 1716. Springer-Verlag, Berlin, 1999.
20. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
21. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. In *Crypto '86*, LNCS 263, pages 186–194. Springer-Verlag, Berlin, 1987.
22. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.
23. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.
24. E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA–OAEP is Still Alive. Cryptology ePrint Archive 2000/061. November 2000. Available from `http://eprint.iacr.org/`.
25. S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
26. S. Goldwasser, S. Micali, and R. Rivest. A "Paradoxical" Solution to the Signature Problem. In *Proc. of the 25th FOCS*, pages 441–448. IEEE, New York, 1984.

27. D. M. Gordon. Discrete Logarithms in GF(p) Using the Number Field Sieve. *SIAM Journal of Discrete Mathematics*, 6(1):124–138, February 1993.
28. J. Håstad. Solving Simultaneous Modular Equations of Low Degree. *SIAM Journal of Computing*, 17:336–341, 1988.
29. N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.
30. N. Koblitz. A Family of Jacobians Suitable for Discrete Log Cryptosystems. In *Crypto '88*, LNCS 403, pages 94–99. Springer-Verlag, Berlin, 1989.
31. N. Koblitz. Hyperelliptic Cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.
32. A. Lenstra and H. Lenstra. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.
33. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
34. H.W. Lenstra. On the Chor-Rivest Knapsack Cryptosystem. *Journal of Cryptology*, 3:149–155, 1991.
35. R. J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *DSN progress report*, 42-44:114–116, 1978. Jet Propulsion Laboratories, CALTECH.
36. A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. *IEEE Transactions on Information Theory*, 39:1639–1646, 1993.
37. D. M'Raïhi, D. Naccache, D. Pointcheval, and S. Vaudenay. Computational Alternatives to Random Number Generators. In *Fifth Annual Workshop on Selected Areas in Cryptography (SAC '98)*, LNCS 1556, pages 72–80. Springer-Verlag, Berlin, 1998.
38. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.
39. V. I. Nechaev. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
40. NIST. Digital Signature Standard (DSS). Federal Information Processing Standards PUBlication 186, November 1994.
41. NIST. Secure Hash Standard (SHS). Federal Information Processing Standards PUBlication 180–1, April 1995.
42. NIST. Secure Hash Algorithm 256/384/512. October 2000.
43. K. Ohta and T. Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. In *Crypto '98*, LNCS 1462, pages 354–369. Springer-Verlag, Berlin, 1998.
44. T. Okamoto and D. Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In *RSA '2001*, LNCS. Springer-Verlag, Berlin, 2001.
45. T. Okamoto and D. Pointcheval. The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes. In *PKC '2001*, LNCS. Springer-Verlag, Berlin, 2001.
46. D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. In *PKC '2000*, LNCS 1751, pages 129–146. Springer-Verlag, Berlin, 2000.
47. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt '96*, LNCS 1070, pages 387–398. Springer-Verlag, Berlin, 1996.
48. J. M. Pollard. Monte Carlo Methods for Index Computation (mod p). *Mathematics of Computation*, 32(143):918–924, July 1978.
49. C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.
50. R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, The Internet Engineering Task Force, April 1992.
51. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
52. T. Satoh and K. Araki. Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves. *Comment. Math. Helv.*, 47(1):81–92, 1998.
53. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Crypto '89*, LNCS 435, pages 235–251. Springer-Verlag, Berlin, 1990.
54. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
55. C. P. Schnorr and M. Jakobsson. Security of Signed ElGamal Encryption. In *Asiacrypt '2000*, LNCS 1976, pages 458–469. Springer-Verlag, Berlin, 2000.
56. D. Shanks. Class Number, a Theory of Factorization, and Genera. In *Proceedings of the Symposium on Pure Mathematics*, volume 20, pages 415–440. AMS, 1971.
57. H. Shimizu. On the Improvement of the Håstad Bound. In *1996 IEICE Fall Conference*, Volume A-162, 1996. In Japanese.

58. V. Shoup. OAEP Reconsidered. Cryptology ePrint Archive 2000/060. November 2000. Available from `http://eprint.iacr.org/`.
59. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt '97*, LNCS 1233, pages 256–266. Springer-Verlag, Berlin, 1997.
60. V. Shoup and R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *Eurocrypt '98*, LNCS 1403, pages 1–16. Springer-Verlag, Berlin, 1998.
61. N. Smart. The Discrete Logarithm Problem on Elliptic Curves of Trace One. *Journal of Cryptology*, 12(3):193–196, 1999.
62. Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, Berlin, 1998.
63. S. Vaudenay. Cryptanalysis of the Chor-Rivest Scheme. In *Crypto '98*, LNCS 1462, pages 243–256. Springer-Verlag, Berlin, 1998.