Towards a DL-based Additively Homomorphic Encryption Scheme

[Published in J. Garay Ed., *Information Security Conference - ISC 2007*, vol. 4779 of *Lecture Notes in Computer Science*, pp. 362–375, Springer-Verlag, 2007.]

Guilhem Castagnos[1] and Benoît Chevallier-Mames[2]

[1] DMI-XLIM,
Université de Limoges,
123, Avenue Albert-Thomas
87060 Limoges Cedex, France
`guilhem.castagnos@unilim.fr`

[2] Gemalto, Security Labs,
La Vigie, Avenue du Jujubier, ZI Athélia IV,
F-13705 La Ciotat Cedex, France
`benoit.chevallier-mames@gemplus.com`

Abstract. ElGamal scheme has been the first encryption scheme based on discrete logarithm. One of its main advantage is that it is simple, natural and efficient, but also that its security is clearly understood. However, one of its — often forgotten — disadvantages is that this scheme requires the encoding of messages into group elements, in order to be semantically secure. Unfortunately, this need prevents the scheme to be fully practical.
In this paper, we propose a new way to deal with the problem of message encoding, which offers several advantages though some disadvantages. Our scheme is based on a quite simple combination of the standard ElGamal scheme with a message encoding inspired by the Naccache-Stern cryptosystem. We consider our solution as a new step towards the open problem of designing a discrete-logarithm based encryption scheme with the property of being additively homomorphic. Unfortunately, our construction is still not a complete solution. We hope however that it might give clues for a possible full solution.

Keywords: ElGamal encryption scheme, Naccache-Stern cryptosystem, DL-based homomorphic scheme, standard model, public-key cryptography.

1 Introduction

Since the discovery of public-key cryptography by Diffie and Hellman [DH76], several encryption schemes have been proposed, but very few of them had real impact on the academic community or industry. Clearly, it is commonly agreed that RSA [RSA78] and ElGamal [ElG85] are of this kind.

More precisely, ElGamal scheme has been the first encryption scheme based on discrete logarithm. One of its main advantage is that it is simple, natural and efficient, but also that its chosen-plaintext security is clearly understood: under the so-called CDH assumption, the one-wayness is ensured; under the so-called DDH assumption, the scheme is semantically secure. However, one of its often forgotten disadvantages is that this scheme requires the encoding of messages into group elements, to ensure indistinguishability. Unfortunately, this need prevents the ElGamal scheme to be fully practical, and its homomorphic properties to be really used.

To get rid off this problem, either the so-called hashed-ElGamal is preferred (in which case, the security is only ensured in the random oracle model), or the construction is totally modified. Note that the Cramer-Shoup encryption scheme (cf. [CS98]), whose IND-CCA proof is valid in the standard model, also requires this encoding.

On the contrary to the problem of designing additive homomorphic encryption schemes based on factorization, which has already been efficiently solved by Paillier [Pai99], after other less-efficient constructions such as Goldwasser-Micali [GM84] or Okamoto-Uchiyama [OU98], no practical homomorphic DL-based primitive is currently known. One would note that the DL-type encoding-free scheme proposed by Chevallier-Mames, Paillier and Pointcheval [CPP06] offers a very-weak kind of malleability, in the sense that one can add a plaintext to a ciphertext without decrypting.

These malleability properties (which also include self-blinding, *i. e.*, the ability of "re-randomizing" a ciphertext) are of great use in certain applications such as e-votes or banking. For example the system of Paillier, and its generalization proposed by Damgård and Jurik [DJ01], have been used to design electronic vote systems [BFP$^+$01,Jur03], for Private Information Retrieval [Lip05], or for building Mix-nets [NSNK06,Jur03].

Our contribution. In this paper, we propose a new solution to the encoding problem, with a security (for chosen-plaintext attacks) in the standard model, under classical assumptions (namely, the DDH assumption). As we explain later, our scheme offers several advantages, though some disadvantages.

Roughly, our scheme is a simple and natural combination of ElGamal with a message encoding inspired by the Naccache-Stern [NS97] cryptosystem. In this regard, and even if our solution has been designed and studied mainly because of the encoding difficulties, our scheme can also be seen as a modification of the original Naccache-Stern construction in order to achieve a certain proof of security: indeed, nothing is really known about the plain Naccache-Stern scheme.

Last but not least, our solution might be seen as a new step towards the design of an additive homomorphic encryption scheme based on the discrete logarithm problem: under conditions that will be detailed, addition of ciphertexts is possible. Moreover, as we do not change the construction of ElGamal, our solution still offers full self-blinding. We hope that our construction might give clues for a possible future full solution.

Outlines. Our paper is organized as follows. In the second part, we remind the background needed for this work, notably definitions of encryption schemes and of their security notions. Then, we describe the ElGamal encryption schemes, and why encoding is needed. In the same section, we also expose solutions that were already proposed to deal with this inherent problem. Fourth part is the main part of our paper: it describes our new encoding for the ElGamal scheme and details its features, efficiency and malleability properties. Finally, we conclude our paper, by opening new problems that are consequences of our work.

2 Preliminaries

In this section, we briefly remind the background regarding public key encryption.

2.1 Public-Key Encryption

We describe a public-key encryption scheme $\mathcal{E}$ as four probabilistic algorithms, $\mathcal{E} = (\textsc{Set}_{\mathcal{E}}, \textsc{Gen}_{\mathcal{E}}, \textsc{Encrypt}, \textsc{Decrypt})$:

Setup. Given a security parameter k, $\textsc{Set}_{\mathcal{E}}(1^k)$ produces some common parameters params, used by the three others algorithms.

Key Generation. Given a security parameter k, $\textsc{Gen}_{\mathcal{E}}(1^k)$ produces a pair $(\mathsf{pk}, \mathsf{sk})$ of public and private keys.

Encryption. Given a message m and a public key pk, $\textsc{Encrypt}_{\mathsf{pk}}(m)$ produces a ciphertext c. As for security reasons, the procedure is typically probabilistic, we write $c = \textsc{Encrypt}_{\mathsf{pk}}(m, r)$ where r denotes the randomness used by $\textsc{Encrypt}$.

Decryption. Given a ciphertext c and a private key sk, $\textsc{Decrypt}_{\mathsf{sk}}(c)$ returns a plaintext m or a special symbol $\perp$ if the ciphertext is invalid.

We will say that a public-key encryption scheme $\mathcal{E}$ is *additively homomorphic* if, given two ciphertexts $c_1 = \textsc{Encrypt}_{\mathsf{pk}}(m_1, r_1)$ and $c_2 = \textsc{Encrypt}_{\mathsf{pk}}(m_2, r_2)$ of unknown plaintexts m_1 and m_2, one can publicly compute a valid ciphertext c_3 of message $m_1 + m_2$.

Moreover, we will say that $\mathcal{E}$ allows *self-blinding*, if given c, an encryption of some (unknown) message m, it is possible to generate efficiently another unlinkable encryption c' of m.

2.2 Security Notions for Encryption Schemes

One-Wayness. The most important security notion that one would expect from an encryption scheme to fulfil is the property of *one-wayness* (OW): an attacker should not be able to recover the plaintext matching a given ciphertext. We capture this notion more formally by saying that for any adversary $\mathcal{A}$, succeeding in inverting the effects of ENCRYPT on a ciphertext c should occur with negligible probability. $\mathcal{A}$ is said to (k, ε, τ)-break OW when

$$\mathsf{Succ}_{\mathcal{E}}^{\mathsf{OW}}(\mathcal{A}) = \Pr_{m,r}[(\mathsf{pk}, \mathsf{sk}) \leftarrow \textsc{Gen}_{\mathcal{E}}(1^k) \ : \ \mathcal{A}(\mathsf{pk}, \textsc{Encrypt}_{\mathsf{pk}}(m, r)) = m] \geq \varepsilon \ ,$$

where the probability is taken over the random coins of the experiment and the ones of the adversary, and $\mathcal{A}$ halts after τ elementary steps. An encryption scheme is said to be one-way if no probabilistic algorithm (k, ε, τ)-breaks OW for $\tau \leq \mathsf{poly}(k)$ and $\varepsilon \geq 1/\mathsf{poly}(k)$.

Semantic Security. The notion of *semantic security* (IND) [GM84], as known as *indistinguishability of encryptions* captures a stronger notion of privacy. Here, the attacker should not learn any information whatsoever about a plaintext given its encryption. The adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is said to (k, ε, τ)-break IND when

$$\mathsf{Adv}_{\mathcal{E}}^{\mathsf{IND}}(\mathcal{A}) = 2 \times \Pr_{b,r}\left[\begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \textsc{Gen}_{\mathcal{E}}(1^k), (m_0, m_1, s) \leftarrow \mathcal{A}_1(\mathsf{pk}), \\ c = \textsc{Encrypt}_{\mathsf{pk}}(m_b, r) \ : \ \mathcal{A}_2(m_0, m_1, s, c) = b\end{array}\right] - 1 \geq \varepsilon,$$

where again the probability is taken over the random coins of the experiment as well as the ones the adversary. $\mathcal{A}$ must run in at most τ steps and it is imposed that $|m_0| = |m_1|$. An encryption scheme is said to be semantically secure or indistinguishable if no probabilistic algorithm can (k, ε, τ)-break IND for $\tau \leq \mathsf{poly}(k)$ and $\varepsilon \geq 1/\mathsf{poly}(k)$.

2.3 Computational Assumptions

We now briefly recall the definition of the discrete-log and related problems needed for the sake of this work. In what follows, $\mathbb{G}$ denotes an abelian finite group (denoted multiplicatively), described by a generator g and its prime order q.

Definition 1 (Discrete Logarithm – DL). *Given $g^x \in \mathbb{G}$ where $x \leftarrow \mathbb{Z}_q$, compute x.*

Definition 2 (Computational Diffie-Hellman – CDH). *Given $g^x \in \mathbb{G}$ and $g^y \in \mathbb{G}$ for $x, y \leftarrow \mathbb{Z}_q$, compute $g^{xy} \in \mathbb{G}$.*

Definition 3 (Decision Diffie-Hellman – DDH). *Let us consider the two distributions $D = (g^x, g^y, g^{xy})$ and $R = (g^x, g^y, g^z)$ for randomly distributed $x, y, z \leftarrow \mathbb{Z}_q$. Distinguish D from R.*

It is easily seen that DDH $\Leftarrow$ CDH $\Leftarrow$ DL where $\Leftarrow$ denotes polynomial reductions. In most cryptographic applications, the structure of the group $\mathbb{G}$ is chosen in such a way that these three computational problems seem intractable. A typical example is to choose $\mathbb{G} \subseteq \mathbb{Z}_p^*$ where q divides $(p-1)$ where classically, p is a 1024-bit prime and q a 160-bit prime. Another widely used family of groups is elliptic curves over large prime fields [Mil85,Kob87].

3 The ElGamal Cryptosystem

ElGamal encryption was introduced around twenty years ago [ElG85]. It requires a cryptographic group $\mathbb{G} = \langle g \rangle$ of order q. In the ElGamal scheme, one generates a public-private key pair by randomly selecting $x \leftarrow \mathbb{Z}_q$ and computing $y = g^x$. The public key is then y while the private key is x. In order to encrypt a message m, one randomly selects $r \leftarrow \mathbb{Z}_q$ and computes $u = g^r$ and $v = y^r \cdot m$. The ciphertext is $c = (u, v)$. Using the private key x, the ciphertext $c = (u, v)$ can be decrypted as $m = v \cdot u^{-x}$.

It is well-known that, for security reasons, one has first to use groups $\mathbb{G}$ of prime order, and second to define $\mathcal{M}$ as included in the group $\mathbb{G}$. Under these assumptions, it has been shown that ElGamal encryption is IND-CPA under the DDH assumption.

Unfortunately, the two above constraints make the ElGamal encryption scheme less practical than it appears at first sight. Indeed, before encryption takes place, the message must be *encoded* into a group element, and this group encoding must be efficiently invertible in order to allow the original message to be recovered during the decryption process. Such an encoding may be time consuming, and may also partially or totally destroy the inherent homomorphic property of the system. Also, using a group encoding remains incompatible with the optimization which consists in working in a small subgroup, $\mathbb{G}$, of $\mathbb{Z}_p^\star$ of prime order q where q is a 160-bit prime, a setting in which group exponentiations are much faster. Indeed, the only known way to preserve the homomorphic property is to use a morphism from $\mathbb{Z}_q$ to $\mathbb{G}$, such as $m \mapsto g^m$, as an encoding function. Unfortunately, this leads to an inefficient decryption process, as one has to compute a discrete logarithm to reverse the encoding (see the computation of the tally in [CGS97] where an e-vote system is built from ElGamal).

3.1 The ElGamal Cryptosystem

With specifications due to the previously explained details, ElGamal encryption scheme is defined as follows [ElG85].

ElGamal

Setup: Let p and q be two large primes so that q divides $(p-1)$. Let $\mathbb{G}$ be the subgroup of $\mathbb{Z}_p^\star$ of order q, and g be a generator of $\mathbb{G}$. Let Ω be an (bijective) encoding map from $\mathbb{Z}_q$ onto $\mathbb{G}$.

Key generation: The private key is $x \leftarrow \mathbb{Z}_q$. The corresponding public key is $y = g^x$.

Encryption: To encrypt a message $m \in \mathbb{Z}_q$, one encodes m by computing $\omega = \Omega(m)$, randomly selects $r \leftarrow \mathbb{Z}_q$ and computes $(u, v) = (g^r, y^r \cdot \omega)$. The ciphertext is $c = (u, v)$.

Decryption: To decrypt a ciphertext $c = (u, v)$, one computes $\omega = v \cdot u^{-x}$ and recovers the original plaintext $m = \Omega^{-1}(\omega)$, if ω is in the set of all the possible encodings. On the contrary, if ω is not a valid encoding, the decryption process returns a special symbol $\perp$.

This cryptosystem is known to be one-way under the CDH assumption, and indistinguishability holds under the DDH assumption. These security notions are reached in the context of chosen-plaintext attacks, in the standard model.

3.2 The Hash-ElGamal Cryptosystem

In order to overcome the issue of group encoding, a hash variant of ElGamal encryption was suggested.

Hash-ElGamal

Setup: Let p and q be two large primes so that q divides $(p-1)$. Let $\mathbb{G}$ be the subgroup of order q of $\mathbb{Z}_p^\star$, and g be a generator of $\mathbb{G}$. Let $\mathcal{H} : \mathbb{G} \rightarrow \{0,1\}^{\ell_m}$ be a hash function.

Key generation: The private key is again $x \leftarrow \mathbb{Z}_q$. The corresponding public key is $y = g^x$.

Encryption: To encrypt a message $m \in \{0,1\}^{\ell_m}$, one randomly selects $r \leftarrow \mathbb{Z}_q$ and computes $(u,v) = (g^r, \mathcal{H}(y^r) \oplus m)$. The ciphertext is $c = (u,v)$.

Decryption: To decrypt a ciphertext $c = (u,v)$, one computes $m = \mathcal{H}(u^x) \oplus v$.

This cryptosystem features one-wayness and indistinguishability under chosen plaintext-attacks under the sole CDH assumption. The security proof, however, stands in the random oracle model [FS86,BR93]. Alternatively, under the DDH assumption, one can apply a randomness extractor in place of the random oracle, in order to generate a truly random mask. Unfortunately, this is much less efficient [CFGP06].

Note however that the use of the hash function destroys the homomorphic property of the scheme.

3.3 Encoding-Free ElGamal Encryption

In 2006, Chevallier-Mames, Paillier and Pointcheval proposed an ElGamal variant, using a new encoding-free technique [CPP06]. Their cryptosystem enjoys performances similar to plain ElGamal but does not require group encoding, nor randomness extractors. Furthermore, the security holds in the standard model, but under new intractability assumptions (namely, *Class Diffie-Hellman problems*), the computational of which is shown to be equivalent to CDH problem.

More precisely, their scheme uses the so-called class of an element of subgroup of $\mathbb{Z}_{p^2}$ of order pq, where p and q are two large primes, so that q divides $p-1$. This class is defined as follows. Let $\mathcal{L}$ be described as $\mathcal{L}(w) = (w^q - 1 \bmod p^2)/p$. Let g be a generator of the subgroup of order q of $\mathbb{Z}_p^*$. The class of $w \in \mathbb{Z}_{p^2}$ of order pq is by definition $[\![w]\!] = \mathcal{L}(w) \cdot \mathcal{L}(g)^{-1} \bmod p$. The encoding-free ElGamal encryption (in its additive variant) is then described as follows.

Encoding-free ElGamal

Setup: Let p and q be two large primes, so that q divides $p-1$. Let g be a generator of the subgroup of order q of $\mathbb{Z}_p^*$.

Key generation: The private key is a random number $x \in \mathbb{Z}_q$. The corresponding public key is $y = g^x \bmod p$.

Encryption: To encrypt a message $m \in \mathbb{Z}_q$, one picks a random $r \in \mathbb{Z}_q$ and computes $u = g^r \bmod p$ and $v = [\![y^r \bmod p]\!] + m \bmod p$. The ciphertext is $c = (u, v)$.

Decryption: To decrypt a ciphertext $c = (u, v)$, one simply computes $m = v - [\![u^x \bmod p]\!] \bmod p$.

We refer to the original paper to show that the security in the standard model under chosen-plaintext attacks is based on the CDH assumption for one-wayness, and on the assumption that the so-called Decision Class Diffie-Hellman is hard for indistinguishability.

Note that this scheme offers a very-weak kind of malleability, in the sense that one can add a plaintext to a ciphertext without decrypting.

4 Main Scheme

This section is the core of our paper. Our goal is to propose a new variant of ElGamal that enjoys both useful malleability properties and a security proof in the standard model (under chosen-plaintext attacks), relatively to a well known assumption. A way to do that is to keep intact the construction of ElGamal and to design a new message encoding that allows some malleability.

First, we remind the Naccache-Stern encryption scheme, whose construction inspired our encoding. Second, we describe our new message encoding for the ElGamal cryptosystem and detail its features. Third, we do exhibit arguments of security for the scheme derived from the combination of ElGamal and our message encoding, and we finally conclude by showing its interesting malleability properties.

4.1 The Naccache-Stern Cryptosystem

The Naccache-Stern cryptosystem is very special in the world of asymmetric cryptography. Indeed, it uses a special trapdoor (a kind of multiplicative knapsack) for deciphering, which makes it unique. The Naccache-Stern scheme [NS97][1] can be described as follows.

[1] We refer the reader to the original paper [NS97] for details on how ℓ_i and p_i are set, in order to achieve optimal efficiency.

Naccache-Stern

Key generation: Let p be a strong prime. For a parameter n, the key generation algorithm searches for n primes p_i and n valuations ℓ_i, such that $\prod_{i=1}^{n} {p_i}^{\ell_i - 1} < p$.

The private key is a random number $x \in \mathbb{Z}_{p-1}^*$. The corresponding public key is the set of elements c_i's defined as $c_i = {p_i}^{1/x \bmod (p-1)} \bmod p$. Then, one defines the set $\mathsf{NS} \in \mathbb{Z}_p^*$ as $\mathsf{NS} = \{\prod_{i=1}^{n} {p_i}^{m_i}, \text{for } m_i \in [0, \ell_i - 1]\}$. For security, the p_i's and NS might be kept private.

Encryption: To encrypt a message $m = \{m_i\}$ with $m_i \in [0, \ell_i - 1]$, one simply computes $w = \prod_{i=1}^{n} {c_i}^{m_i} \bmod p$. The ciphertext is w. Of course, to achieve indistinguishability, some randomization is included in a pre-step (for example, via a padding of the message with random).

Decryption: To decrypt a ciphertext w, one computes $t = w^x \bmod p$. Then, if $t \in \mathsf{NS}$, one simply recovers the message $m = \{m_i\}$ by decomposing t into the base of primes p_i, which is simple as p_i's are small and known. On the contrary, if $t \notin \mathsf{NS}$, the decryption returns a special symbol $\perp$.

Even if the Naccache-Stern scheme is based on the classical DL problem, its special type of trapdoor makes that there does not exist real proof of security for this scheme.

4.2 Our Scheme

As previously said, our scheme is made of the (almost natural but never proposed at our knowledge) composition of ElGamal and a message encoding inspired by the Naccache-Stern cryptosystem (in its optimal bandwidth variant[2]).

Main points of design. First, as we want to mix Naccache-Stern and ElGamal schemes, we have to include the p_i's in the subgroup $\mathbb{G} \subset \mathbb{Z}_p^*$ of order q. Unfortunately, taking (relatively) small subgroup order is in contradiction with expecting large bandwidth, as the smaller is q, the more negligible is the probability that a small prime is in $\mathbb{G}$. Thus, we have to take a maximal order for q, *i.e.*, we have to use a strong prime p.

Second, once we have mixed the ElGamal and Naccache-Stern cryptosystems (see below for further details), we see it is no more needed to scramble elements of NS (which will be used to encode the messages in $\mathbb{G}$), as done in Naccache-Stern scheme. Therefore, we need no more to compute and publish large public key set $\{c_i\}$, but rather to take common parameters p_i for all users.

[2] If, however, the reader prefers to first look at our description with the simplest variant of Naccache-Stern scheme, it suffices to suppose that $\ell_i = 2$ for all i.

Description. We describe our proposal for a new encoding for the standard ElGamal cryptosystem (see Subsection 3.1) according to the previously explained points.

Our new encoding

Setup: Let $p = 2q + 1$ be a strong prime. Let g be a generator of the subgroup $\mathbb{G}$ of $\mathbb{Z}_p^*$ of order q. For a certain parameter n, the setup algorithm searches for n primes p_i and n valuations ℓ_i, so that p_i's are of order q into $\mathbb{Z}_p^*$ and such that $\prod_{i=1}^{n} {p_i}^{\ell_i - 1} < p$.

Encoding $\Omega(m)$ of a message m: A message m is a $n-$tuple of integers, $m = (m_1, \ldots, m_n)$ such that $0 \leq m_i < l_i$ for all $i = 1, \ldots, n$. The encoding of m is $\Omega(m) = \prod_{i=1}^{n} {p_i}^{m_i}$. We denote NS the image of Ω, *i. e.*, the set of all the possible encodings.

Decoding of an element t of $\mathsf{NS} \subset \mathbb{G}$: To decode, one decomposes t into the base of primes p_i, $t = \prod_{i=1}^{n} {p_i}^{m_i}$ and outputs $m = (m_1, \ldots, m_n)$.

Efficiency. On the one hand, the encoding process is composed of at most n multiplications and n small exponentiations in $\mathbb{Z}$; on the other hand, the decryption is made of a factorization of a simple instance, which is done in practise by some trial divisions by the p_i's. Hence, in term of speed, the cost of the encoding process is negligible compared to the cost of the ElGamal encryption and decryption steps.

Parameter size. In the Naccache-Stern framework, the public key is made of some secret powers of small primes; our scheme also needs a large set of elements of $\mathbb{Z}_p^*$ (namely, the p_i's), but a definitive advantage over the Naccache-Stern scheme is that (i) these elements can be shared by users instead of being dedicated to a given person, and (ii) these elements are small[3], while c_i's in Naccache-Stern are full-size elements of $\mathbb{Z}_p^*$. These two advantages allow much more practical public-key infrastructures, almost as efficient as those of others ElGamal schemes.

The impact of the maximal order. However, to be fair, one disadvantage of our construction over others ElGamal-based scheme is that we limit the order of the subgroup (that is q) to be maximal, which makes both private key larger

[3] Typically, one finds the p_i's in the $2n$ first primes, the factor 2 coming from the condition that the chosen primes must be of order q in $\mathbb{Z}_p^*$.

and exponentiation longer[4]. We however may consider this as a price to pay to achieve a Naccache-Stern type construction with a provable security.

Encryption bandwidth. By experimentation, we can estimate the size of the messages that we could encrypt. Typically, using $\ell_i = 2$ for every i, we may be able to generate p of 1024 bits and p_i's, such that $n \approx 117$, meaning that we would encrypt message of almost 117 bits. For more general case, the point is to optimize $\prod_i \ell_i$ (whose logarithm is the bitsize of messages one can encrypt), under the condition that $\sum_i (\ell_i - 1)\log_2(p_i)$ is limited by the bitsize of p.

Variant. To work in the subgroup of order q, *i. e.*, the subgroup of squares of $\mathbb{Z}_p^*$, we can use another trick instead of using p_i's of order q. As p is a strong prime, $p \equiv 3 \pmod 4$, so given an element x of $\mathbb{Z}_p^*$, either x or $-x$ is a square. As a consequence, in the setup of our system, we can use all small primes, relaxing the condition that p_i must be a square, *i. e.*, of order q. However, we restrict the l_i's such that $\prod_{i=1}^n {p_i}^{\ell_i - 1} < p/2$. In the encoding process, given a message $m = (m_1, \ldots, m_n)$ with $0 \leq m_i < l_i$, we compute $w' = \prod_{i=1}^n {p_i}^{m_i}$ and set $w = w'$ (resp. $w = p - w'$), according to w is a square (resp. a non-square). To decode t, one factors t (resp. $p - t$) if $t < p/2$ (resp. if $t > p/2$).

This variant reduces the public key size (instead of giving all the p_i's we can make n public and the p_i's will be the n first primes), and gives a better bandwidth (if we use $\ell_i = 2$ for every i and if p is a 1024 bit prime, we can encrypt a message of 131 bits as the product of the first 131 primes is smaller than 2^{1022}). The encoding cost is similar to the one of the main scheme (to see efficiently if w is a square, one can pre-compute the Legendre symbols of the p_i's). The decoding process has the same complexity.

Security. We finish this comparison by a major advantage of our scheme (notably over the encoding-free ElGamal scheme based on Class Diffie-Hellman problems): the security of our scheme is based on a classical assumption, namely DDH, as shown in the next section.

4.3 Security Analysis

Oppositely to the typical encryption proofs, we start with indistinguishability, as it appears surprisingly simpler than the analysis of the one-wayness of our scheme.

Theorem 1. *The composition of our new encoding and the* ElGamal *scheme is semantically secure against chosen plaintext attacks, under the* DDH *assumption.*

[4] Indeed, the order has been chosen maximal in order to make that almost half small primes are of order q in $\mathbb{Z}_p^*$. However, if the speed is more important than the scheme bandwidth, it should be possible to take shorter orders q at the price of larger p_i's and so less bandwidth.

This theorem follows from the semantic security of the standard ElGamal scheme as we only specify the encoding process of messages in elements of $\mathbb{G}$.

□

We emphasize that this semantic security is interesting not only for our scheme, but also as our modification can be seen as a way to achieve a certain security for a Naccache-Stern type cryptosystem (while the original Naccache-Stern scheme has no known proof of security).

One-wayness. The one-wayness of the construction is not as simple to characterize. At least, due to relations with semantic security, we know it is at least as difficult as DDH to invert the scheme. In addition, one might have the intuition that the scheme is as hard as the CDH to solve, but in the following, we *almost* infirm this, by showing that the natural reduction one could think about is inefficient.

Let a CDH challenge $(g, s = g^a \bmod p, g^x \bmod p)$ on a group $\mathbb{G} = \langle g \rangle$ of order $q = (p-1)/2$ in $\mathbb{Z}_p^*$, described by (g, p, q), and assume an access to an attacker $\mathcal{A}$ against the one-wayness of the scheme corresponding to the composition of our new encoding and ElGamal.

One could give (s, r) as a ciphertext to the attacker (for a random $r \in \mathbb{G}$), which would return a plaintext m if $(rg^{-ax} \bmod p) \in \mathsf{NS}$. However, very few elements of $\mathbb{G}$ are in NS[5], and so the probability of this reduction is very small. Anyway, if such m was returned, one could simply re-encode the returned m by computing $\prod_{i=1}^{n} {p_i}^{m_i} \bmod p$, and divides r by this quantity, in order to get g^{ax}, the answer of the given challenge.

As a conclusion, we only claim that the OW-CPA security of the scheme is at least as difficult as the DDH problem to solve (thanks to the indistinguishability proof), even it might be possible one could find a better proof, based on a weaker assumption.

4.4 Towards a DL-based Homomorphic Scheme

As we said earlier, additively homomorphic encryption primitives are wanted objects, as they have many applications in protocols. However, today, only schemes based on factorization are known (*e. g.*, Paillier [Pai99], or its predecessors such as Goldwasser-Micali [GM84] and Okamoto-Uchiyama [OU98]).

Remarkably, the encoding-free ElGamal variant of Chevallier-Mames, Paillier and Pointcheval offers a weak kind of malleability, for the first time for a DL-based primitive. As we explain in this section, our scheme goes further, as it allows full *self-blinding* of ciphertexts and a certain malleability on ciphertexts.

Self-blinding. The self-blinding property is very useful, as it allows to re-randomize ciphertexts, which is a key feature for certain applications. In our

[5] In fact, $\prod_{i=1}^{i=n} \ell_i$ over the q elements of $\mathbb{G}$.

scheme, as we keep intact the structure of $\mathsf{ElGamal}$, we can still re-randomize a ciphertext (u, v) by picking a random $r \in \mathbb{Z}_q$, then forming the new ciphertext (u', v') of the same plaintext, with $u' = u \cdot g^r \bmod p$ and $v' = v \cdot y^r \bmod p$.

Adding ciphertexts (under restrictions). Let two valid ciphers (u, v) and (u', v'), ciphering respectively two messages $m = (m_1, \ldots, m_n)$ and $m' = (m'_1, \ldots, m'_n)$. Then, $(u'' = u \cdot u' \bmod p, v'' = v \cdot v' \bmod p)$, is a valid cipher of $m'' = (m_1 + m'_1, \ldots, m_n + m'_n)$ as long as one has

$$\forall\, i \in \mathbb{N},\ \ 1 \leq i \leq n,\ \ m_i + {m'}_i < \ell_i. \tag{†}$$

This means that it is possible to add ciphertexts, if plaintexts were previously selected in such way the condition (†) is always fulfilled.

For example, one might set a voting scheme sketched as follows. The voters would either vote for *yes* (1) or *no* (0) to some terrible question. During the setting of the election, each voter would be assigned an index i she must use (that is, for any $1 \leq i \leq n$, ℓ_i voters would use the same p_i to encrypt their vote $m_i \in \{0, 1\}$). Then, all the votes[6] of the $\sum_i \ell_i$ voters could be "aggregated" by multiplying the ciphertexts, then decrypted by some well-know techniques of threshold encryption (in [Ped91], for instance, one can find the description of a robust threshold variant of $\mathsf{ElGamal}$ that can be applied to our scheme). This would give at the end the sum of the votes, and so the result of the ballot. In practice, if p is a 1024 bits prime, if $n = 1$ and $p_1 = 2$, we can manage until 1024 voters with our scheme, which is a satisfying number as in a real life scenario, there are around a thousand registered voters by polling stations.

By using the malleability property, we could also design a multi-candidate election system with n candidates and k voters as follows: a voter would vote for the i^{th} candidate, with $1 \leq i \leq n$, by encrypting p_i. By multiplying all the ciphertexts, the election manager would get an encryption of $\prod_i p_i^{k_i}$ where the k_i's are the number of votes for the i^{th} candidate, respectively. With the setting $k < \log_{p_n}(p)$ (where we supposed that p_n is the larger prime) and $\mathsf{NS} = \{\prod_{i=1}^{n} {p_i}^{m_i},$ for m_i so that $\sum_i m_i \leq k\}$, this encryption is valid, *i. e.*, can be decrypted. If $n = 5$ and p is a 4096 bit prime such that $p_5 = 17$ (that is, one finds a strong prime $p = 2q + 1$ such that 5 of the 7 first primes are of order q), one can still have a thousand voters. With the variant of our encoding, we can use all the primes but with the restriction $k < \log_{p_n}(p/2)$. If $n = 7$ and p is a 4096 bit prime, with this variant, we have $p_7 = 17$ (the seventh prime) and can still have a thousand voters.

We do agree that our scheme is still not the panacea for complete additive homomorphy, but at least, we believe that the full self-blinding of the scheme as well as its restricted additive property might be of interest. Clearly, devising a full and complete DL-based additive scheme is still an open problem.

[6] That would have been proved to be a correct encryption of a message in the valid set.

5 Conclusion

In this paper, we have proposed another way to deal with the problem of message encoding, which is often a forgotten drawback and bottleneck of ElGamal encryption type cryptosystems. Our scheme offers several advantages though some disadvantages: notably, our scheme is based on the classical DDH assumption in the standard model (for the chosen-plaintext scenario), while previous solutions were mainly based on new defined problems or random oracles.

We also showed how our scheme possesses some additive homomorphy, which was an open problem for a long time for DL-type primitives. Notably, we have shown how it allows full self-blinding and a restricted additivity of the ciphertexts.

Open problems let by this work are of several kinds: firstly, we still consider the search of a full additively homomorphic DL-based scheme as interesting and challenging; secondly, we think it might be possible to achieve a better proof of our scheme in the OW-CPA scenario; lastly, it should be possible to adapt our technique to mix Naccache-Stern and Cramer-Shoup encryption schemes, in order (maybe) to obtain a scheme without encoding that would be secure in the standard model against chosen ciphertext attacks.

Acknowledgement. The work described in this paper has been supported in part by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT.

References

[BFP+01] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *Proc. of PODC' 01*, 2001.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security – ACM CCS 1993*, pages 62–73. ACM Press, 1993.

[CFGP06] Olivier Chevassut, Pierre-Alain Fouque, Pierrick Gaudry, and David Pointcheval. The twist-augmented technique for key exchange. In *Public Key Cryptography – PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 410–426. Springer-Verlag, 2006.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *Advances in Cryptology – EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 103–118. Springer-Verlag, 1997.

[CPP06] Benoît Chevallier-Mames, Pascal Paillier, and David Pointcheval. Encoding-free ElGamal encryption without random oracles. In *Public Key Cryptography – PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 91–104. Springer-Verlag, 2006.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology – CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer-Verlag, 1998.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DJ01] Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Public Key Cryptography – PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136. Springer-Verlag, 2001.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology – CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–184. Springer-Verlag, 1986.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[Jur03] Mads Jurik. *Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols*. PhD thesis, Aarhus University, 2003.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[Lip05] Helger Lipmaa. An Oblivious Transfer Protocol with Log-Squared Communication. In *The 8th Information Security Conference (ISC'05)*, volume 3650 of *Lecture Notes in Computer Science*, pages 314–328. Springer-Verlag, 2005.

[Mil85] Victor S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology – CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer-Verlag, 1985.

[NS97] David Naccache and Jacques Stern. A new public-key cryptosystem. In *Advances in Cryptology – EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 27–36. Springer-Verlag, 1997.

[NSNK06] Lan Nguyen, Rei Safavi-Naini, and Kaoru Kurosawa. Verifiable shuffles: a formal model and a Paillier-based three-round construction with provable security. *Int. J. Inf. Secur.*, 5(4):241–255, 2006.

[OU98] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 308–318. Springer-Verlag, 1998.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 1999.

[Ped91] Torben P. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *Proc. of Eurocrypt' 91*, pages 522–526, 1991.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.