HAL
archives-ouvertes.fr

Homomorphic Encryption for Multiplications and Pairing Evaluation

Fabien Laguillaumie, Guilhem Castagnos

▶ **To cite this version:**

Fabien Laguillaumie, Guilhem Castagnos. Homomorphic Encryption for Multiplications and Pairing Evaluation. Ivan Visconti and Roberto De Prisco. Security and Cryptography for Networks - 8th International Conference, SCN 2012, Sep 2012, Amalfi, Italy. 2012, Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings. <hal-00763110>

HAL Id: hal-00763110
https://hal.inria.fr/hal-00763110

Submitted on 11 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Homomorphic Encryption for Multiplications and Pairing Evaluation

Guilhem Castagnos[1] and Fabien Laguillaumie[2]

[1] Institut de Mathématiques de Bordeaux – Université Bordeaux 1
351, cours de la Libération, 33405 Talence cedex, France
guilhem.castagnos@math.u-bordeaux1.fr
[2] Université de Caen Basse-Normandie and CNRS/ENSL/INRIA/UCBL LIP
Laboratoire de l'Informatique du Parallélisme
46 Allée d'Italie, 69364 Lyon, France
fabien.laguillaumie@unicaen.fr

Abstract. We propose a generic approach to design homomorphic encryption schemes, which extends Gjøsteen's framework. From this generic method, we deduce a new homomorphic encryption scheme in a composite-order subgroup of points of an elliptic curve which admits a pairing $e : \mathbf{G} \times \mathbf{G} \to \mathbf{G}_t$. This scheme has some interesting theoretical and practical properties: it allows an arbitrary number of multiplications in the groups $\mathbf{G}$ *and* $\mathbf{G}_t$, *as well as* a pairing evaluation on the underlying plaintexts. We prove the semantic security under chosen plaintext attack of our scheme under a generalized subgroup membership assumption, and we also prove that it *cannot* achieve ind-cca1 security. We eventually propose an original application to shared decryption. On the theoretical side, this scheme is an example of cryptosystem which can be naturally implemented with groups of prime order, as the homomorphic properties require only a *projecting* pairing using Freeman's terminology. However the application to shared decryption also relies on the fact that the pairing is *cancelling* and therefore does not survive this conversion.

1 Introduction

Homomorphic encryption scheme allows one to operate on plaintexts, only from their given ciphertexts. The Elgamal encryption is a classical example of such a homomorphic encryption, since, given two ciphertexts, it is easy to obtain the encryption of the product of the two corresponding plaintexts. This malleability property is of crucial interest since it is the core of many electronic realizations of real-life applications like electronic voting [BFP+01,DJ01], private information retrieval [Lip05], verifiable encryption [FPS00], mix-nets [NSNK06,Jur03], auction protocols [MMO10], *etc.* In most of these cases, there is a need for an *additively* homomorphic encryption, in the sense that it is possible to obtain the encryption of the sum of plaintexts.

Since the introduction of the first probabilistic encryption scheme by Goldwasser and Micali in 1984 [GM84] (where they also formally defined the notion

of semantic security for encryption), many schemes were designed along the same lines, like Benaloh [Ben88], Naccache and Stern [NS98], or Okamoto and Uchiyama [OU98]. These cryptosystems are based on modular arithmetic, and use indeed several quotients of $\mathbf{Z}$, so that their one-wayness relies on the hardness of the factorization of (special form of) RSA modulus and their semantic security on distinguishing some powers. Significant improvements appear in the subsequent scheme designed by Paillier [Pai99] in 1999 which is still very popular. Its semantic security is based on the decisional composite residuosity assumption. Paillier's scheme has then been generalized by Damgård and Jurik [DJ01], allowing one to encrypt larger messages. All these schemes fit Gjøsteen's framework around subgroup membership problems [Gjo04,Gjo05], which encompasses also multiplicative schemes like Elgamal.

Encryption schemes supporting both additive and multiplicative homomorphisms are of course critical for the design of highly functional cryptosystems. A spectacular breakthrough was made by Gentry who proposed the first *fully homomorphic* encryption scheme [Gen09], which allows to compute *arbitrary* functions over encrypted data without the decryption key. Recent works show that efficiency of such systems could become reality (see for instance some solutions based on the (ring) *learning with error problems* [BV11,BGV12]).

On the way towards practical fully homomorphic encryption are schemes that partially support additive and multiplicative homomorphisms, like Boneh, Goh and Nissim's scheme (BGN) [BGN05]. It is based on groups of points of elliptic curves of composite orders which admit a pairing, supports an arbitrary number of additions and only one multiplication. This remains sufficient to make possible the evaluation of a formula in disjunctive normal form where each conjunction has at most 2 literals. In practice, this provides efficient solutions, with quite standard objects, for operations on encrypted data which do not require fully homomorphic schemes, such as search or statistics.

Our Contributions. In this paper, we propose a homomorphic encryption scheme which supports an arbitrary number of group operations and *pairing evaluation* on the underlying plaintexts. We first give a generic construction of a homomorphic scheme which goes a step forward compared to Gjøsteen's framework and extends its properties. We provide an instantiation within groups of composite orders with a pairing which has richer homomorphic properties, and discuss if this instantiation can be moved into a prime-order setting.

One of the features of our new scheme is that it is possible to encrypt *any* element of a subgroup of composite order of the group of points of a pairing-friendly elliptic curve. Moreover, it is publicly possible, given the encryptions of two points, to compute the encryption of the products of these points (if we consider the group of points of the curve as multiplicative). It is as well possible to publicly compute an encryption of the pairing of these two points. To finish, given the encryptions of two pairing evaluations, it is possible to publicly compute an encryption of the product of these values.

Even if the global setting of our scheme (bilinear groups of composite order) is quite similar to the setting of BGN, the malleability properties of our scheme

are indeed very different from the ones of BGN. This comes from the fact that the plaintexts of BGN are *small* integers (or elements of $\mathbf{Z}/2\mathbf{Z}$) encoded in elliptic curve points by exponentiation whereas plaintexts of our scheme are just points.

Quite surprisingly, our system is *not* $\mathsf{ind}-\mathsf{cca1}$ (cf. Prop. 1). This result proves that even with strong assumptions, there exist homomorphic schemes which cannot reach such a level of security. Moreover, the role of the splitting problem in our system makes it possible to provide a natural and original application to shared decryption, that does not rely on traditional secret sharing techniques. Concerning the conversion in the prime-order setting, we are able to benefit from Freeman's transformation (cf. [Fre10,MSF10,SC12]) from pairing-based schemes in composite-order groups into equivalent ones in prime-order groups: Our basic scheme can be directly converted, which gives a more efficient cryptosystem, based on the Decision Linear Problem. However, the nice result on $\mathsf{ind}-\mathsf{cca1}$ security and the application to shared decryption do not survive this conversion. This may give an evidence of the existence of limits to Freeman's transformation.

The paper is organized as follows. In section 2, we give the necessary background to define a homomorphic encryption scheme for multiplications and pairing evaluation. In section 3, we describe a generic construction of a multiplicative homomorphic scheme. This construction gives schemes whose one-wayness is based on a generalization of the splitting problem in finite groups and whose semantic security is based on a generalization of the symmetric subgroup membership problem. These problems have been introduced by Gjøsteen [Gjo04,Gjo05] and our generic construction can be viewed as a generalization of his construction with more than two subgroups. An instantiation of our construction in quotients of $\mathbf{Z}$ can be found in [GBD05]. Section 4 is devoted to an instantiation in bilinear groups of composite order that gives a concrete and efficient homomorphic scheme for multiplications and pairing evaluation. As detailed in that section, it is necessary, contrary to BGN, to use groups whose order is the product of at least *three* prime numbers to get a secure scheme. At the end of this section we give an application to shared decryption. Eventually, we compare our new cryptosystem with existing schemes and discuss the (im)possibility to move our scheme into a prime-order setting.

2 Background

2.1 Encryption Scheme: Definitions

Definition Let $\lambda \in \mathbf{N}$ be a security parameter. An encryption scheme is a triple of algorithms $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$. The probabilistic polynomial-time key generation algorithm KeyGen takes 1^λ as input and returns a pair (pk, sk) of public key and the matching secret key. The probabilistic polynomial-time encryption algorithm $\mathsf{Encrypt}$ takes 1^λ, a public key pk and a message m as inputs, and outputs a ciphertext c. The deterministic polynomial-time decryption algorithm $\mathsf{Decrypt}$ takes 1^λ, a secret key sk and a ciphertext c as inputs and returns either a message m or the symbol $\perp$ which indicates the invalidity

of the ciphertext. The scheme must be *correct*, which means that for all security parameters λ, and for all messages m, if $(pk, sk) \xleftarrow{\$} \mathcal{E}.\mathsf{KeyGen}(1^\lambda)$ then $\mathcal{E}.\mathsf{Decrypt}(1^\lambda, sk, \mathcal{E}.\mathsf{Encrypt}(1^\lambda, pk, m)) = m$ with probability (taken on all internal random coins and random choices) 1.

Security Requirements. The *total break* of an encryption scheme is declared if an attacker can recover the secret key from (at least) the public key. Therefore any probabilistic polynomial-time Turing machine $\mathcal{A}$ (the *attacker*) must have a success in recovering the public key arbitrarily small, where the *success* is defined, for an integer λ, as:

$$\mathbf{Succ}^{\mathsf{tb}}_{\mathcal{E}}(\mathcal{A}) = \Pr\big[(pk, sk) \xleftarrow{\$} \mathcal{E}.\mathsf{KeyGen}(1^\lambda) : \mathcal{A}(pk) = sk\big].$$

A stronger security notion expected from an encryption scheme is the *one-wayness*, which means that, given only the public data, an adversary cannot recover the message corresponding to a given ciphertext. More precisely, if we denote by $\mathcal{M}$ the set of plaintexts, any probabilistic polynomial-time Turing machine $\mathcal{A}$ has a success in inverting the encryption algorithm arbitrarily small, where the *success* is defined, for an integer λ, as $\mathbf{Succ}^{\mathsf{ow}}_{\mathcal{E}}(\mathcal{A})$ equals to

$$\Pr\big[(pk, sk) \xleftarrow{\$} \mathcal{E}.\mathsf{KeyGen}(1^\lambda), m \xleftarrow{\$} \mathcal{M} : \mathcal{A}(pk, \mathcal{E}.\mathsf{Encrypt}(1^\lambda, pk, m)) = m\big].$$

Note that the previous definition supposes that the attacker has no more information than the public key : the attacker is said to do a *chosen-plaintext attack* (since he can produce the ciphertext of messages of his choice). If he has access to a decryption oracle, the attack is said be a *chosen-ciphertext attack*.

An encryption scheme must indeed reach a stronger notion of security : it must have *semantic security* (a.k.a. *indistinguishability*). This means that an attacker is computationally unable to distinguish between two messages, chosen by himself, which one has been encrypted, with a probability significantly better than one half. The *indistinguishability game* is formally defined as:

Experiment $\mathbf{Exp}^{\mathsf{ind}-\mathsf{atk}}_{\mathcal{E}}(\mathcal{A})$

$(pk, sk) \xleftarrow{\$} \mathcal{E}.\mathsf{KeyGen}(1^\lambda)$
$(m_0, m_1, s) \xleftarrow{\$} \mathcal{A}_1^{\mathcal{O}_1}(pk)$
$b^\star \xleftarrow{\$} \{0, 1\}$
$c^\star \xleftarrow{\$} \mathcal{E}.\mathsf{Encrypt}(1^\lambda, pk, m_{b^\star})$
$b \xleftarrow{\$} \mathcal{A}_2^{\mathcal{O}_2}(s, c^\star)$
if $b = b^\star$ then return 1
else return 0

with

- $\mathsf{atk} = \mathsf{cpa}$ and
  - $\mathcal{O}_1 = \emptyset$
  - $\mathcal{O}_2 = \emptyset$
- $\mathsf{atk} = \mathsf{cca1}$ and
  - $\mathcal{O}_1 = \mathcal{E}.\mathsf{Decrypt}(1^\lambda, sk, \cdot)$
  - $\mathcal{O}_2 = \emptyset$
- $\mathsf{atk} = \mathsf{cca2}$ and
  - $\mathcal{O}_1 = \mathcal{E}.\mathsf{Decrypt}(1^\lambda, sk, \cdot)$
  - $\mathcal{O}_2 = \mathcal{E}.\mathsf{Decrypt}(1^\lambda, sk, \cdot)$

where the adversary $\mathcal{A}$ is modeled as a 2-stage probabilistic polynomial-time Turing machine $(\mathcal{A}_1, \mathcal{A}_2)$. In the CCA2 game, a natural restriction is imposed

to $\mathcal{A}_2$ which is not allowed to query $\mathcal{O}_2$ on $c^\star$. The *advantage* of the attacker is then defined as

$$\mathbf{Adv}_{\mathcal{E}}^{\mathsf{ind-atk}}(\mathcal{A}) = \left| \Pr\Big(\mathbf{Exp}_{\mathcal{E}}^{\mathsf{ind-atk}}(\mathcal{A}) = 1\Big) - \frac{1}{2} \right|.$$

It is well known that encryption schemes which enjoy homomorphic properties, cannot achieve the highest level of security (namely IND-CCA2 security), but can still achieve IND-CCA1 security (see for instance [APK10]).

2.2 Homomorphic Encryption for Multiplications and Pairing Evaluation

In order to describe more precisely our new encryption scheme with its features, we will use the following less general definition of encryption schemes but more adapted to our setting.

First of all, the set of plaintexts will be composed of two distinct *multiplicative groups* $(\mathbf{M}, \times_{\mathbf{M}})$ and $(\mathbf{M_t}, \times_{\mathbf{M_t}})$. Similarly, the set of ciphertexts is composed of two distinct sets $\mathbf{C}$ and $\mathbf{C_t}$ corresponding respectively to encryptions of elements of $\mathbf{M}$ and $\mathbf{M_t}$. Moreover, a particular characteristic of our encryption scheme is that there is a function e (a *pairing*) mapping elements from $\mathbf{M} \times \mathbf{M}$ onto elements of $\mathbf{M_t}$.

Definition 1. *Let $\lambda \in \mathbf{N}$ be a security parameter. An* homomorphic encryption scheme for multiplications and pairing evaluation *is composed of the following algorithms:*

- KeyGen *is a probabilistic algorithm which takes as input 1^λ and outputs the keys pair (pk, sk) of public and secret key respectively, the groups of plaintexts $\mathbf{M}$ and $\mathbf{M_t}$, the sets of ciphertexts $\mathbf{C}$ and $\mathbf{C_t}$ and the pairing $e : \mathbf{M} \times \mathbf{M} \to \mathbf{M_t}$. The description of the groups $\mathbf{M}, \mathbf{M_t}, \mathbf{C}, \mathbf{C_t}$ and of the pairing e will be common parameters for each of the following algorithms;*
- Encrypt *is a probabilistic algorithm which takes as inputs 1^λ, the public key pk and a plaintext m. If $m \in \mathbf{M}$ it outputs a ciphertext $c \in \mathbf{C}$ else if $m \in \mathbf{M_t}$ it outputs a ciphertext $c \in \mathbf{C_t}$;*
- Decrypt *is a deterministic algorithm which takes as inputs 1^λ, the secret key sk and a ciphertext c. It outputs either a plaintext m (in $\mathbf{M}$ if $c \in \mathbf{C}$ and in $\mathbf{M_t}$ if $c \in \mathbf{C_t}$) or $\perp$;*
- EvalMul *is a probabilistic algorithm which takes as inputs 1^λ, the public key pk and two ciphertexts c and c' of unknown plaintexts m and m' of the same group. If c and c' are elements of $\mathbf{C}$, it outputs an element $c'' \in \mathbf{C}$ which is a random encryption[3] of $m \times_{\mathbf{M}} m'$; else if c and c' are elements of $\mathbf{C_t}$ it outputs a random encryption $c'' \in \mathbf{C_t}$ of $m \times_{\mathbf{M_t}} m'$;*

[3] By *random encryption*, we mean that the distribution of the outputs c'' of EvalMul is the same as the distribution of the encryption algorithm on inputs $m \times_{\mathbf{M}} m'$.

– $\mathsf{EvalPair}$ *is a probabilistic algorithm which takes as inputs* 1^λ*, a public key* pk*, and two ciphertexts* c *and* c' *of* $\mathbf{C}$ *of unknown plaintexts* m *and* m' *of* $\mathbf{M}$*. It outputs a random encryption* $c'' \in \mathbf{C_t}$ *of* $e(m, m') \in \mathbf{M_t}$*.*

These algorithms must verify the different correctness *properties, defined as follows. For all* $\lambda \in \mathbf{N}$*,*

$$\Pr\big[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda), m \xleftarrow{\$} \mathbf{M} \cup \mathbf{M_t}, c \xleftarrow{\$} \mathsf{Encrypt}(1^\lambda, pk, m) : \mathsf{Decrypt}(1^\lambda, sk, c) = m\big] = 1.$$

$$\Pr\big[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda), m \xleftarrow{\$} \mathbf{M}, m' \xleftarrow{\$} \mathbf{M}, c \xleftarrow{\$} \mathsf{Encrypt}(1^\lambda, pk, m), c' \xleftarrow{\$} \mathsf{Encrypt}(1^\lambda, pk, m'), c'' \xleftarrow{\$} \mathsf{EvalMul}(1^\lambda, c, c', pk) : \mathsf{Decrypt}(1^k, sk, c'') = m \times_{\mathbf{M}} m'\big] = 1$$

$$\Pr\big[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda), m \xleftarrow{\$} \mathbf{M_t}, m' \xleftarrow{\$} \mathbf{M_t}, c \xleftarrow{\$} \mathsf{Encrypt}(1^\lambda, pk, m), c' \xleftarrow{\$} \mathsf{Encrypt}(1^\lambda, pk, m'), c'' \xleftarrow{\$} \mathsf{EvalMul}(1^\lambda, c, c', pk) : \mathsf{Decrypt}(1^k, sk, c'') = m \times_{\mathbf{M_t}} m'\big] = 1$$

and for pairing evaluation:

$$\Pr\big[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda), m \xleftarrow{\$} \mathbf{M}, m' \xleftarrow{\$} \mathbf{M}, c \xleftarrow{\$} \mathsf{Encrypt}(1^\lambda, pk, m), c' \xleftarrow{\$} \mathsf{Encrypt}(1^\lambda, pk, m'), c'' \xleftarrow{\$} \mathsf{EvalPair}(1^\lambda, c, c', pk) : \mathsf{Decrypt}(1^k, sk, c'') = e(m, m)'\big] = 1$$

At that point, it is important to keep in mind that in our scheme, a first level of plaintexts will lie in the group $(\mathbf{M}, \times_{\mathbf{M}})$ and their corresponding ciphertexts will lie in the set $\mathbf{C}$. Once $\mathsf{EvalPair}$ is evaluated on two such ciphertexts, the result is an encryption of the pairing of the original first level plaintexts from $\mathbf{M}$ and so lies in $\mathbf{C_t}$: this gives a second level of ciphertexts, corresponding to the second level of plaintexts $\mathbf{M_t}$. Since the homomorphic property will also apply on the second level, it is possible to obtain the encryption of products of such pairings. This is why our scheme is homomorphic for the two multiplications $\times_{\mathbf{M}}$ *and* $\times_{\mathbf{M_t}}$ *and* for the pairing evaluation.

Another important remark is that the scheme can not be semantically secure for the whole message set: The first stage adversary of the indistinguishability game can pick one plaintext in $\mathbf{M}$ and the other one in $\mathbf{M_t}$. Then the second stage adversary will observe if the challenge ciphertext is in $\mathbf{C}$ or $\mathbf{C_t}$. The semantic security of the scheme will rather hold for plaintexts of $\mathbf{M}$ and for plaintexts of $\mathbf{M_t}$ separately.

3 General Setting

In this section, we first give a natural generic construction of an homomorphic scheme on which our instance of an homomorphic encryption scheme for multiplications and pairing evaluation will be based. This construction is quite

natural but the algorithmic problem on which relies the one wayness of the scheme is *not*. That's why we give in Subsection 3.3 a particular setting of this construction for which the one wayness of the scheme is related to a classical splitting problem. This construction generalizes the scheme from [GBD05] in an abstract group with more than 2 subgroups. This generalization actually allows the design of richer cryptosystems: indeed, the scheme from [GBD05] does not support bilinear groups (see Subsection 4.2), whereas it is possible to implement our framework with such specific groups, which leads to an encryption scheme which is more versatile. In the next section, we show how to apply this construction to pairing-friendly elliptic curves to get the homomorphic encryption scheme for multiplications and pairing evaluation.

3.1 A Generic Construction

Let $\lambda \in \mathbf{N}$ be a security parameter and k be a fixed integer. Let $\mathbf{G}$ be a finite Abelian *multiplicative* group and for $i \in \{1, \ldots, k\}$, $\mathbf{H}_i$ is a subgroup of $\mathbf{G}$ of order denoted by $|\mathbf{H}_i|$. We impose that the orders of the subgroups $\mathbf{H}_1, \ldots, \mathbf{H}_k$ are k distinct integers of λ bits such that $\gcd(|\mathbf{H}_1|, \ldots, |\mathbf{H}_k|) = 1$. We denote $(u_1, \ldots, u_k)$ the integers such that $\sum_{i=1}^{k} u_i |\mathbf{H}_i| = 1$. We call $\mathsf{Bézout}$ the algorithm which computes these k values from the orders $|\mathbf{H}_1|, \ldots, |\mathbf{H}_k|$.

In the following, whenever a group appears in the input or output of an algorithm, it means that an efficient way to compute the group law is known and that we can sample random elements of this group. For example, the groups are cyclic and a generator is given.

We denote as $\mathsf{GroupsGen}$ the probabilistic algorithm that takes as input 1^λ and outputs the tuple $(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, |\mathbf{H}_1|, \ldots, |\mathbf{H}_k|)$. The public key pk consists of the groups $\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k$ whereas the private key sk will consist of their orders and the Bézout coefficients. More precisely, the key generation algorithm is as follows:

Algorithm $\mathsf{KeyGen}(1^\lambda)$

$(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, |\mathbf{H}_1|, \ldots, |\mathbf{H}_k|) \xleftarrow{\$} \mathsf{GroupsGen}(1^\lambda)$
$(u_1, \ldots, u_k) \leftarrow \mathsf{Bézout}(|\mathbf{H}_1|, \ldots, |\mathbf{H}_k|)$
$pk \leftarrow (\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k)$
$sk \leftarrow (|\mathbf{H}_1|, \ldots, |\mathbf{H}_k|, u_1, \ldots, u_k)$
$\mathsf{return}\ (pk, sk)$

The encryption algorithm will use the homomorphism $\Pi : \mathbf{G} \to \mathbf{G}/\mathbf{H}_1 \times \cdots \times \mathbf{G}/\mathbf{H}_k$. This homomorphism is the Cartesian product of the surjective homomorphisms $\pi_i : \mathbf{G} \to \mathbf{G}/\mathbf{H}_i$ for $i = 1, \ldots, k$. The set of plaintexts is defined to be $\mathbf{G}$. Let m be an element of $\mathbf{G}$: It is encrypted as a random representative of the k-tuple of classes $\Pi(m) = (m\mathbf{H}_1, \ldots, m\mathbf{H}_k) \in \mathbf{G}/\mathbf{H}_1 \times \cdots \times \mathbf{G}/\mathbf{H}_k$. For example, when generators $(h_1, \ldots, h_k)$ of $(\mathbf{H}_1, \ldots, \mathbf{H}_k)$ are publicly known, an encryption of m consists therefore of $(mh_1^{r_1}, \ldots, mh_k^{r_k})$ for random $r_1, \ldots, r_k \in$

$\{1, \ldots, |\mathbf{G}|\}$. To decrypt $C = (c_1, \ldots, c_k) \in \mathbf{G}^k$, one computes $\prod_{i=1}^{k} c_i^{u_i|H_i|}$. If C is an encryption of m, then$\prod_{i=1}^{k} c_i^{u_i|\mathbf{H}_i|} = m^{\sum_{i=1}^{k} u_i|\mathbf{H}_i|} = m$, and the encryption scheme is correct.

More formally, the encryption and decryption algorithms are described bellow. It is easy to see that this gives an homomorphic scheme : if C_1 (resp. C_2) is an encryption of m_1 (resp. m_2) then C_1C_2 (with the component-wise multiplication) is an encryption of m_1m_2 that can be randomized by a multiplication by a random element of $(\mathbf{H}_1, \ldots, \mathbf{H}_k)$.

Algorithm $\mathsf{Encrypt}(1^k, pk, m)$
$(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k) \leftarrow pk$
$C \xleftarrow{\$} \Pi(m)$
return C

Algorithm $\mathsf{Decrypt}(1^k, sk, C)$
$(c_1, \ldots, c_k) \leftarrow C$
$(|\mathbf{H}_1|, \ldots, |\mathbf{H}_k|, u_1, \ldots, u_k) \leftarrow sk$
$m \leftarrow \prod_{i=1}^{k} c_i^{u_i|\mathbf{H}_i|}$
return m

3.2 Security of the Generic Construction

The *total break* under a *chosen plaintext attack* of the scheme presented in the previous subsection is equivalent to the following problem: given $\mathbf{G}$ and k of its subgroups $\mathbf{H}_1, \ldots, \mathbf{H}_k$, find the orders of $\mathbf{H}_1, \ldots, \mathbf{H}_k$. This is a standard *order-finding problem* which can be solved with standard algorithms for computing discrete logarithms. These algorithms are of complexity either exponential or sub-exponential in the security parameter, depending on context (when the discrete logarithm is supposed to be hard). If the order of $\mathbf{G}$ is given, the total break is equivalent to the factorization of this number, which is at least a λ bit integer (note that not the whole factorization of $|\mathbf{G}|$ might be found). The best algorithms for factoring have a sub-exponential complexity.

The *one wayness* of the scheme under a *chosen plaintext attack* is equivalent to the difficulty of the following problem: Given a random representative of the image $\Pi(m) \in \mathbf{G}/\mathbf{H}_1 \times \cdots \times \mathbf{G}/\mathbf{H}_k$, recover $m \in \mathbf{G}$. In the next subsection, we give a specific setting where this problem is equivalent to a more common problem, namely the *splitting problem* [Gjo05].

Concerning *the indistinguishability* under a *chosen plaintext attack*, we define the following problem, which is generally called a *subgroup membership problem*. In this specific form it is a direct generalization of the *symmetric subgroup membership problem* (cf. [Gjo04,Gjo05]), where $k = 2$, $\mathbf{H}_1 \cap \mathbf{H}_2 = \{1\}$ and $\mathbf{G} = \mathbf{H}_1\mathbf{H}_2$.

Definition 2 (Generalized Symmetric Subgroup Membership Problem). *The* generalized symmetric subgroup membership problem *(*GSSMP*) consists, given the tuple* $(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k)$ *as input, in distinguishing the two distributions* $\mathbf{G} \times \cdots \times \mathbf{G}$ *and* $\mathbf{H}_1 \times \cdots \times \mathbf{H}_k$*. More formally, let us consider the following random experiment:*

Experiment $\mathbf{Exp}^{\mathsf{GSSMP}}_{\mathsf{GroupsGen}}(\mathcal{A})$

$(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, |\mathbf{H}_1|, \ldots, |\mathbf{H}_k|) \xleftarrow{\$} \mathsf{GroupsGen}(1^\lambda)$
$b^\star \xleftarrow{\$} \{0,1\}$
if $b^\star = 0$ then $X \xleftarrow{\$} \mathbf{G} \times \cdots \times \mathbf{G}$
  else $X \xleftarrow{\$} \mathbf{H}_1 \times \cdots \times \mathbf{H}_k$
$b \leftarrow \mathcal{A}(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, X)$
if $b = b^\star$ then return 1
  else return 0

The advantage *of $\mathcal{A}$ in solving the generalized symmetric subgroup membership problem is*

$$\mathbf{Adv}^{\mathsf{GSSMP}}_{\mathsf{GroupsGen}}(\mathcal{A}) = \left| \Pr[\mathbf{Exp}^{\mathsf{GSSMP}}_{\mathsf{GroupsGen}}(\mathcal{A}) = 1] - \frac{1}{2} \right|.$$

Theorem 1 ($\mathsf{ind}-\mathsf{cpa}$)**.** *Let k be an integer. If there exists an attacker against the indistinguishability of the generic encryption scheme of subsection 3.1 with parameter k in a chosen plaintext attack with security parameter λ, running time τ and advantage ε, then there exists an algorithm for the generalized symmetric subgroup membership problem with the same security parameter, advantage $\varepsilon/2$ and running time $\tau + T_{k\text{-}Mul}$ where $T_{k\text{-}Mul}$ is the time to perform k multiplications in $\mathbf{G}$.*

Proof. Suppose that $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is an $\mathsf{ind}-\mathsf{cpa}$ attacker against the generic encryption scheme, denoted by $\mathcal{E}$. The following distinguisher $\mathcal{D}$ will break a challenge of the form $(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, X)$ for the GSSMP thanks to its oracle access to $\mathcal{A}$.

Distinguisher $\mathcal{D}(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, X)$

$(x_1, \ldots, x_k) \leftarrow X$
$(m_0,\, m_1,\, s) \leftarrow \mathcal{A}_1(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k)$
$b^\star \xleftarrow{\$} \{0,\, 1\}$,
$C \leftarrow (m_{b^\star} x_1, \ldots, m_{b^\star} x_k)$
$b \leftarrow \mathcal{A}_2(s,\, C)$
if $b^\star = b$ then return 1
  else return 0

If $X \xleftarrow{\$} \mathbf{H}_1 \times \cdots \times \mathbf{H}_k$, then C is a correct encryption of $m_{b^\star}$ and $\mathcal{D}$ outputs 1 if and only $\mathcal{A}_2$ has correctly guessed the value of $b^\star$. Therefore

$$\Pr\big[\mathbf{Exp}^{\mathsf{GSSMP}}_{\mathsf{GroupsGen}}(\mathcal{D}) = 1 \mid X \xleftarrow{\$} \mathbf{H}_1 \times \cdots \times \mathbf{H}_k\big] = \Pr\big(\mathbf{Exp}^{\mathsf{ind}-\mathsf{cpa}}_{\mathcal{E}}(\mathcal{A}) = 1\big).$$

If $X \xleftarrow{\$} \mathbf{G} \times \cdots \times \mathbf{G}$, then C is independent of $b^\star$, so $\mathcal{A}_2$ has no advantage in guessing the right value of this bit, and $\mathcal{D}$ outputs 1 with probability 1/2. Therefore,

$$\Pr\left[\mathbf{Exp}^{\mathsf{GSSMP}}_{\mathsf{GroupsGen}}(\mathcal{D}) = 1\right] = \frac{1}{2}\left(\Pr\left(\mathbf{Exp}^{\mathsf{ind-cpa}}_{\mathcal{E}}(\mathcal{A}) = 1\right) + \frac{1}{2}\right),$$

and

$$\mathbf{Adv}^{\mathsf{GSSMP}}_{\mathsf{GroupsGen}}(\mathcal{D}) = \frac{1}{2}\,\mathbf{Adv}^{\mathsf{ind-cpa}}_{\mathsf{Scheme}}(\mathcal{A}).$$

□

Remark that conversely, given a distinguisher for the GSSMP, it is trivial to build an attacker for the semantic security. As a result, the two problems are polynomially equivalent.

3.3 A Particular Setting

A particular specialization of the generic construction of subsection 3.1, is when there exists subgroups $\mathbf{G}_1, \ldots, \mathbf{G}_k$ of $\mathbf{G}$ such that $\mathbf{G} = \prod_{i=1}^{k} \mathbf{G}_i$ and $\mathbf{G}_i \cap \mathbf{G}_j = \{1\}$ if $i \neq j$. We suppose that $|\mathbf{G}_1|, \ldots, |\mathbf{G}_k|$ are k distinct primes of $\lambda/(k-1)$ bits. In this case, we define the subgroups $\mathbf{H}_i$ as $\mathbf{H}_i = \prod_{j \neq i} \mathbf{G}_j$ for $i \in \{1, \ldots, k\}$.

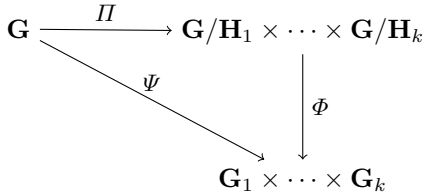
We denote as $\mathsf{GroupsGen}'$ the algorithm that takes as input 1^λ and outputs the tuple

$$(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, |\mathbf{H}_1|, \ldots, |\mathbf{H}_k|, \mathbf{G}_1, \ldots, \mathbf{G}_k).$$

We still suppose that there exists a public method to sample random elements of $\mathbf{G}$ and of the subgroups $\mathbf{H}_1, \ldots, \mathbf{H}_k$. However, it is not necessary that anyone can sample elements of the subgroups $\mathbf{G}_1, \ldots, \mathbf{G}_k$ (as we shall see in subsection 4.2, such an implementation of the construction with elliptic curves equipped with pairings, actually leads to an insecure scheme). The encryption scheme is defined in the same way as in subsection 3.1. Only the construction of the subgroups $\mathbf{H}_1, \ldots, \mathbf{H}_k$ differs (with $\mathsf{GroupsGen}'$ instead of $\mathsf{GroupsGen}$).

For each $i \in \{1, \ldots, k\}$, $\mathbf{G}/\mathbf{H}_i$ is isomorphic to $\mathbf{G}_i$. We denote as ϕ_i this isomorphism and as Φ the Cartesian product of the ϕ_i for $i \in \{1, \ldots, k\}$. This map Φ is an isomorphism between $\mathbf{G}/\mathbf{H}_1 \times \cdots \times \mathbf{G}/\mathbf{H}_k$ and $\mathbf{G}_1 \times \cdots \times \mathbf{G}_k$.

We have the following commutative diagram where each map is an isomorphism:



Let m be an element of $\mathbf{G}$, then there is a unique decomposition of m as a $k-$tuple $(m_1, \ldots, m_k) \in \mathbf{G}_1 \times \cdots \times \mathbf{G}_k$ such that $m = \prod_{i=1}^{k} m_i$. The map Ψ corresponds to this decomposition, and Ψ^{-1} is the computation of the product $\prod_{i=1}^{k} m_i$.

Remark 1. Decrypting a ciphertext $C = (c_1, \ldots, c_k)$ associated to the plaintext m is closely related to the decomposition of Ψ as it corresponds to the computation of $\Psi^{-1} \circ \Phi$. More precisely, let us fix $i \in \{1, \ldots, k\}$ and let us consider a representative $c_i = mh_i \in \mathbf{G}$ of $\pi_i(m)$ with $h_i \in \mathbf{H}_i$. Remember that we have $\sum_{j=1}^{k} u_j |\mathbf{H}_j| = 1$. Modulo $|\mathbf{G}_i|$ this sum gives $u_i |\mathbf{H}_i| = 1$ as $|\mathbf{G}_i|$ divides all $|\mathbf{H}_j|$ with $j \neq i$. As a consequence, if $(m_1, \ldots, m_k) = \Psi(m)$, then $m_j^{u_i|\mathbf{H}_i|} = 1$ if $j \neq i$ and $m_i^{u_i|\mathbf{H}_i|} = m_i$. The decryption $\prod_{i=1}^{k} c_i^{u_i|\mathbf{H}_i|}$ gives

$$\prod_{i=1}^{k} c_i^{u_i|\mathbf{H}_i|} = \prod_{i=1}^{k} (mh_i)^{u_i|\mathbf{H}_i|} = \prod_{i=1}^{k} (m_1 m_2 \ldots m_k)^{u_i|\mathbf{H}_i|} = \prod_{i=1}^{k} m_i = m.$$

To sum up, the decryption process corresponds to the computation of $(m_1, \ldots, m_k)$ with Φ and making their product with ψ^{-1}.

In this special setting, breaking the one wayness of the encryption scheme is equivalent to solving a direct generalization of a well known problem, the splitting problem defined in (cf. [Gjo04,Gjo05]) where $k = 2$.

Definition 3 (Splitting Problem). *The* splitting problem *consists, given the tuple* $(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k)$ *and* $m \in \mathbf{G}$*, in finding* $(m_1, \ldots, m_k) \in \mathbf{G}_1 \times \cdots \times \mathbf{G}_k$ *such that* $m = \prod_{i=1}^{k} m_i$*. More formally, let us consider the following random experiment:*

Experiment $\mathbf{Exp}^{\mathsf{SP}}_{\mathsf{GroupsGen}'}(\mathcal{A})$

$(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, |\mathbf{H}_1|, \ldots, |\mathbf{H}_k|, \mathbf{G}_1, \ldots, \mathbf{G}_k) \leftarrow \mathsf{GroupsGen}'(1^\lambda)$
$m \xleftarrow{\$} \mathbf{G}$
$(m_1, \ldots, m_k) \leftarrow \mathcal{A}(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, m)$
if $\forall i \in \{1, \ldots, k\}, m_i \in \mathbf{G}_i$ and $\prod_{i=1}^{k} m_i = m$ then return *1*
  else return 0

The success *of* $\mathcal{A}$ *in solving the splitting problem is*

$$\mathbf{Succ}^{\mathsf{SP}}_{\mathsf{GroupsGen}'}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\mathsf{SP}}_{\mathsf{GroupsGen}'}(\mathcal{A}) = 1\right].$$

Theorem 2 (One-Wayness-CPA). *If there exists an attacker against the one-wayness under a chosen plaintext attack of the encryption scheme of subsection 3.3 with security parameter* λ*, running time* τ *and success* ε*, then there exists an algorithm for the splitting problem with the same security parameter,*

success ε^k *and running time* $\tau + (k+1)T_{k\text{-}Mul} + T_{k\text{-}Inv} + (k+1)T_{k\text{-}Rand}$ *where* $T_{k\text{-}Mul}$ *(resp.* $T_{k\text{-}Inv}$*) is the time to perform a multiplication (resp. an inversion) in* $\mathbf{G} \times \cdots \times \mathbf{G}$*, and* $T_{k\text{-}Rand}$ *the time to sample a random element of* $\mathbf{H}_1 \times \cdots \times \mathbf{H}_k$*.*

Proof. Let us denote $\mathcal{E}'$ the encryption scheme of this subsection and suppose that there is an attacker $\mathcal{A}$ which succeeds in breaking the one-wayness of the scheme with probability $\varepsilon = \mathbf{Succ}^{\mathsf{ow}}_{\mathcal{E}'}(\mathcal{A})$ and running time τ. We show that this attacker can be used to design a successful algorithm $\mathcal{B}$ which solves the Splitting Problem.

The challenge of $\mathcal{B}$ consists of $(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k, m)$. Let us denote $\Psi(m) = (m_1, \ldots, m_k)$, the solution that $\mathcal{B}$ is looking for.

The algorithm $\mathcal{B}$ first retrieves m_1 thanks to its oracle $\mathcal{A}$. Let $(h_1, \ldots, h_k)$ be a random element of $\mathbf{H}_1 \times \cdots \times \mathbf{H}_k$ and f another random element of $\mathbf{H}_1$. $\mathcal{B}$ builds the ciphertext $C = (mh_1, h_2 f, \ldots, h_k f)$. Denote $(1, f_2, \ldots, f_k) = \Psi(f)$. It is easy to see that C is a random encryption of $m_1 f_2 f_3 \ldots f_k = m_1 f$ where f is known by $\mathcal{B}$. As a result, $\mathcal{B}$ forward the public key $(\mathbf{G}, \mathbf{H}_1, \ldots, \mathbf{H}_k)$ and the ciphertext C to $\mathcal{A}$, and gets m_1 with probability ε. Iterating this procedure, $\mathcal{B}$ outputs $(m_1, \ldots, m_k)$ with probability ε^k, k calls to $\mathcal{A}$, $k+1$ samples of random elements of $\mathbf{H}_1 \times \cdots \times \mathbf{H}_k$ and $(k+1)$ multiplications and one inversion in $\mathbf{G} \times \cdots \times \mathbf{G}$. □

Again, there is an equivalence between the two problems. Let us denote $C = (c_1, c_2, \ldots, c_k)$ an encryption of m where $c_i = mh_i$, with $h_i \in \mathbf{H}_i$ for all $i \in \{1, \ldots, k\}$ and $(m_1, m_2, \ldots, m_k) = \Psi(m)$. For $i \in \{1, \ldots, k\}, \Psi(c_i) = \Psi(m)\Psi(h_i)$ and

$$\Psi(h_i) = (h_{i,1}, \ldots, h_{i,i-1}, 1, h_{i,i+1}, \ldots, h_{i,k})$$

due to the construction of $\mathbf{H}_i$. As a result, an oracle for the Splitting Problem called on the input c_i gives m_i in the i-th coordinate. With k calls to the oracle, one can retrieve $m = m_1 m_2 \ldots m_k$ and break the one wayness of the encryption scheme.

3.4 Known Implementations of the Construction

Let $p = 2n + 1$, $n = q_1 q_2$ where p, q_1, q_2 are distinct primes. The particular setting described in the previous subsection was used in [GBD05] with $\mathbf{G}$ the cyclic subgroup of the multiplicative group $(\mathbf{Z}/p\mathbf{Z})^*$ of order n and $k = 2$. The subgroup $\mathbf{H}_1 = \mathbf{G}_2$ (resp. $\mathbf{H}_2 = \mathbf{G}_1$) is the cyclic subgroup of order q_2 (resp. of order q_1). In this work the Splitting Problem was named Projection Problem. This scheme was generalized in an abstract group $\mathbf{G}$ still with $k = 2$ in [Bro07]. Our construction can thus be viewed as a generalization of this last work with $k \geq 2$. Other schemes based on the Symmetric Subgroup Membership Problem and the Splitting Problem are implementations of this construction, such as the scheme of [Gjo05].

4 A Concrete Homomorphic Scheme for Multiplications and Pairing Evaluation

In this section, we consider the construction of subsection 3.3 in a context of pairing-friendly elliptic curves. This means that there exists a non-degenerate efficiently computable bilinear map $e : \mathbf{G} \times \mathbf{G} \to \mathbf{G_t}$, where $\mathbf{G}_t$ is a group isomorphic to $\mathbf{G}$ called the target group. In this case, $\mathbf{G}$ is essentially a group of points of an elliptic curve. We will then enjoy a double homomorphic property: The homomorphy for the group of points of the elliptic curve and the homomorphy in the target group of the pairing. As a result we will get a secure scheme satisfying Definition 1, which is more versatile than existing schemes.

4.1 Implementation of the Generic Construction with Bilinear Groups with Composite Orders

As in the generic construction, let k be a fixed integer and $\lambda \in \mathbf{N}$ be a security parameter. Let $q_1, \ldots, q_k$ be k distinct prime integers of λ bits and $n = \prod_{i=1}^{k} q_i$ be the product of these primes. The integer ℓ is defined as the smallest integer such that $p = \ell n - 1$ is prime and $p \equiv 2 \pmod 3$. The following construction of a bilinear group with composite order has been initially proposed in [BGN05] with $k = 2$.

Let us consider the supersingular elliptic curve of equation $y^2 = x^3+1$ defined over $\mathbf{F}_p$. The $\mathbf{F}_p$-rational points of this curve form a group of cardinality $p+1 = \ell n$ and we denote by $\mathbf{G}$ its subgroup of order n. Let $\mathbf{G_t}$ be the subgroup of $(\mathbf{F}_{p^2})^\star$ of order n. Finally, let $e : \mathbf{G} \times \mathbf{G} \to \mathbf{G_t}$ be the modified Weil Pairing as defined in [BF03,Mil04]. In [BRS11], a method with ordinary curves and embedding degree 1 is also proposed which is quite equivalent in terms of efficiency: For the supersingular curve construction, $\rho := \log p / \log n \approx 1$ (ℓ is less than 10 bits in practice, for a 1500 bits n) and the embedding degree is 2. In [BRS11], the curves constructed with embedding degree 1 have $\rho \approx 2$. So both constructions are close to the minimum $\rho \times \kappa = 2$ where κ is the embedding degree.

As in the construction of subsection 3.3, we denote by $\mathbf{G}_i$ the subgroup of $\mathbf{G}$ of order q_i, for all integers $i \in \{1, \ldots, k\}$ and the subgroups $\mathbf{H}_i$ are again defined as $\mathbf{H}_i = \prod_{\substack{j=1 \\ j \neq i}}^{k} \mathbf{G}_j$. With these groups, one can apply the construction of subsection 3.3 to get an homomorphic encryption scheme in $\mathbf{G}$. Moreover, we can define the corresponding subgroups in $\mathbf{G}_t$ and we will get another homomorphic encryption scheme in $\mathbf{G}_t$. With the pairing e, we get an homomorphic encryption scheme for multiplications and pairing evaluation.

We denote as $\mathcal{BG}$ the algorithm which takes as input 1^λ and k and outputs the tuple

$$(\mathbf{G}, \mathbf{G}_t, e, \mathbf{H}_1, \ldots, \mathbf{H}_k, \mathbf{G}_1, \ldots, \mathbf{G}_k, q_1, \ldots, q_k).$$

4.2 Insecure Instantiation with $k = 2$

If one chooses $k = 2$, then $\mathbf{H}_2 = \mathbf{G}_1$ is of order q_1 and $\mathbf{H}_1 = \mathbf{G}_2$ is of order q_2. In this case, the corresponding encryption scheme in $\mathbf{G}_t$ is a direct generaliza-

tion of the [GBD05] scheme in $\mathbf{F}_{p^2}$. Unfortunately, in this case, the Generalized Symmetric Subgroup Membership Problem of Definition 2 is tractable and the encryption scheme is therefore not semantically secure. Indeed, as we want to be able to sample random elements of $\mathbf{H}_1$ and $\mathbf{H}_2$ then generators h_1 of order q_2 and h_2 of order q_1, must be public. In that case, we can easily recognize elements of $\mathbf{H}_1 \times \mathbf{H}_2$ thanks to the pairing e: Let $(x_1, x_2) \in \mathbf{G} \times \mathbf{G}$, then

$$(x_1, x_2) \in \mathbf{H}_1 \times \mathbf{H}_2 \Longleftrightarrow e(x_1, h_2) = 1 \text{ and } e(x_2, h_1) = 1.$$

To see that fact, let g be a generator of $\mathbf{G}$ and let us write $h_2 = g^{rq_2}$ for some r prime to q_1 and $x_1 = g^{r'}$ for some integer r'. Then x_1 is an element of $\mathbf{H}_1$ if and only if q_1 divides r', if and only if $e(x_1, h_2) = e(g, g)^{rr'q_2} = 1$. The criterion for $x_2 \in \mathbf{H}_2$ holds by symmetry.

In the BGN scheme (cf. [BGN05]), a composite bilinear group with $k = 2$ is actually used. However, in that particular scheme, only a random generator of the subgroup $\mathbf{G}_1$ is given in the public key which makes the previous attack unfeasible. As a result, only messages modulo $\mathbf{G}_1$ can be encrypted. This is not a problem since in the BGN cryptosystem, only *small* plaintext messages m of $\mathbf{N}$ are encoded with the exponentiation $g \mapsto g^m$; the decryption can then be performed by the computation of a small discrete logarithm in basis g modulo $\mathbf{G}_1$. In our scheme, we want to encrypt *any* element of $\mathbf{G}$, that is why we also need to publish a generator of $\mathbf{G}_2$ and this attack is then possible. Therefore we need at least $k = 3$ to get a secure scheme.

4.3 Description of our Scheme with $k = 3$

As previously said, to design a secure instantiation from our methodology, we need to use the bilinear groups with composite-order generator $\mathcal{BG}$ with k *at least equals to 3*. For simplicity, we expose our scheme with $k = 3$. This means that the integer n is the product of three primes $n = q_1q_2q_3$. We suppose also that h_i are random generators of the groups $\mathbf{H}_i$ of orders n/q_i for $i = 1, 2, 3$. They can be produced by taking a generator g of $\mathbf{G}$ and setting $h_i = g^{\alpha_i q_i}$, for random α_i prime to n.

Note that $e(g, g)$ generates the group $\mathbf{G_t}$ and $e(g, h_i)$ generates the subgroup of $\mathbf{G_t}$ of order n/q_i. We can therefore apply the generic construction in $\mathbf{G}$ and $\mathbf{G_t}$: to encrypt of elements of $\mathbf{G_t}$, instead of multiplying the message by a random power of h_i, one has to multiply by a random power of $e(g, h_i)$.

This gives an homomorphic scheme for multiplications and pairing evaluation with $\mathbf{M} = \mathbf{G}$, $\mathbf{M_t} = \mathbf{G_t}$, $\mathbf{C} = \mathbf{G}^3$ and $\mathbf{C_t} = \mathbf{G_t}^3$. This scheme is presented in Figure 1.

Correctness of Decryption and Homomorphic Properties The correctness of the decryption algorithm follows from the generic construction. The homomorphic property of $\mathsf{EvalMul}$ for both multiplication in $\mathbf{G}$ and $\mathbf{G_t}$ can be checked easily. Concerning the pairing evaluation, for $i = 1, 2, 3$, we have

Algorithm $\mathsf{KeyGen}(1^\lambda)$

$(\mathbf{G}, \mathbf{G}_t, e, \mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, q_1, q_2, q_3)$
$\quad \xleftarrow{\$} \mathcal{BG}(1^\lambda, k = 3)$
$g \xleftarrow{\$} \mathbf{G}$ of order n ; $g_t \leftarrow e(g, g)$
for i from 1 to 3 do
$\quad h_i \xleftarrow{\$} \mathbf{H}_i$ of order n/q_i
$\quad h_{t_i} \leftarrow e(g, h_i)$
$(u, v, w) \leftarrow \mathsf{Bézout}(q_2q_3, q_1q_3, q_1q_2)$
$n \leftarrow q_1q_2q_3$
$pk \leftarrow (g, h_1, h_2, h_3, g_t, h_{t_1}, h_{t_2}, h_{t_3}, n, \mathbf{G}, \mathbf{G_t}, e)$
$sk \leftarrow pk \cup (q_1, q_2, q_3, u, v, w)$
return (pk, sk)

Algorithm $\mathsf{Encrypt}(1^\lambda, pk, m)$

if $m \in \mathbf{G}$ then
$\quad$ for i from 1 to 3 do
$\quad\quad r_i \xleftarrow{\$} \{1, \ldots, n\}$
$\quad\quad c_i \leftarrow mh_i^{r_i}$
$\quad C \leftarrow (c_1, c_2, c_3)$
else
$\quad$ for i from 1 to 3 do
$\quad\quad r_i \xleftarrow{\$} \{1, \ldots, n\}$
$\quad\quad c_i \leftarrow mh_{t_i}^{r_i}$
$\quad C \leftarrow (c_1, c_2, c_3)$
return C

Algorithm $\mathsf{Decrypt}(1^\lambda, sk, C)$

$(c_1, c_2, c_3) \leftarrow C$
$m \leftarrow c_1^{uq_2q_3} \times c_2^{vq_1q_3} \times c_3^{wq_1q_2}$
return m

Algorithm $\mathsf{EvalPair}(1^k, pk, C, C')$

$(c_1, c_2, c_3) \leftarrow C$
$(c'_1, c'_2, c'_3) \leftarrow C'$
for i from 1 to 3 do
$\quad r_i \xleftarrow{\$} \{1, \ldots, n\}$
$\quad c''_i \leftarrow e(c_i, c'_i)h_{t_i}^{r_i}$
return (c''_1, c''_2, c''_3)

Algorithm $\mathsf{EvalMul}(1^\lambda, pk, C, C')$

$(c_1, c_2, c_3) \leftarrow C$
$(c'_1, c'_2, c'_3) \leftarrow C'$
if $C \in \mathbf{G}^3$ then
$\quad$ for i from 1 to 3 do
$\quad\quad r_i \xleftarrow{\$} \{1, \ldots, n\}$
$\quad\quad c''_i \leftarrow c_ic'_ih_i^{r_i}$
else
$\quad$ for i from 1 to 3 do
$\quad\quad r_i \xleftarrow{\$} \{1, \ldots, n\}$
$\quad\quad c''_i \leftarrow c_ic'_ih_{t_i}^{r_i}$
return (c''_1, c''_2, c''_3)

Fig. 1. Our new homomorphic encryption for multiplications and pairing evaluation

$$e(c_i, c_i') = e(mh_i^{r_i}, m'h_i^{r_i'}) = e(m, m')\underbrace{e(h_i^{r_i}, m')e(m, h_i^{r_i'})e(h_i^{r_i}, h_i^{r_i'})}_{\text{of order } n/q_i}$$

and the element $e(h_i^{r_i}, m')e(m, h_i^{r_i'})e(h_i^{r_i}, h_1^{r_i'})$ lies in the subgroup of $\mathbf{G_t}$ of order n/q_i, therefore $e(c_i, c_i')$ is the i-th part of an encryption of $e(m, m')$.

Security Results The one-wayness of our scheme against chosen plaintext attacks follows from Theorem 2 if the splitting problem is hard. In $\mathbf{G}$, this means it must be hard to decompose an element m in $m_1, m_2, m_3 \in \mathbf{G}_1 \times \mathbf{G}_2 \times \mathbf{G}_3$ such that $m = m_1 m_2 m_3$. According to Theorem 1, our encryption scheme is semantically secure against chosen plaintext attacks for messages in $\mathbf{G}$ if the generalized symmetric subgroup membership problem with pairing is hard in $\mathbf{G}$, *i.e.*, if it is hard to distinguish elements of $\mathbf{H}_1 \times \mathbf{H}_2 \times \mathbf{H}_3$ in $\mathbf{G} \times \mathbf{G} \times \mathbf{G}$, given generators of $\mathbf{G}, \mathbf{H}_1, \mathbf{H}_2$ and $\mathbf{H}_3$ and a pairing $e : \mathbf{G} \times \mathbf{G} \to \mathbf{G}_t$. Given the pairing e, it is easy to see that this GSSMP problem in $\mathbf{G}$ reduces to the GSSMP problem in $\mathbf{G}_t$. As a consequence, under the assumption that the generalized symmetric subgroup membership problem with pairing is hard in $\mathbf{G}$, our encryption scheme is semantically secure against chosen plaintext attacks for both messages in $\mathbf{G}$ and in $\mathbf{G}_t$. This assumption can be proved to hold in the generic group model if factoring n is hard, following the lines of the proofs of [KSW08, Section A.2] and [JS08, Theorem 4].

Regarding the security against adaptive chosen ciphertexts attacks, the cryptosystem being homomorphic, it cannot be even one-way ($\mathsf{ow} - \mathsf{cca2}$) in this scenario. Little is known on the security of homomorphic schemes in the cca1 scenario without strong assumptions (cf. [BP04,APK10]). Surprisingly for our cryptosystem, we are able to prove that for messages in $\mathbf{G}$, $\mathsf{ind} - \mathsf{cca1}$ security cannot be reached. This result proves that even with strong assumptions, *all* the homomorphic schemes cannot be proved to be $\mathsf{ind} - \mathsf{cca1}$ secure.

Proposition 1. *The new homomorphic encryption for multiplications and pairing evaluation of Figure 1 is not* $\mathsf{ind} - \mathsf{cca1}$ *secure for plaintext messages in* $\mathbf{G}$.

Proof. Before getting its challenge ciphertext in the $\mathsf{ind} - \mathsf{cca1}$ experiment, an adversary can use its decryption oracle to decompose a random $x \in \mathbf{G}$ in $x_1, x_2, x_3 \in \mathbf{G}_1 \times \mathbf{G}_2 \times \mathbf{G}_3$ such that $x = x_1 x_2 x_3$ following the reduction of the proof of Theorem 2. Knowing elements of $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$, the subgroups of order q_1, q_2 and q_3, the adversary can now solve the subgroup membership problem like in the case $k = 2$ (see subsection 4.2). Hence, he can break the indistinguishability of the scheme.

As the scheme is not $\mathsf{ind} - \mathsf{cca1}$ secure in $\mathbf{G}$, from $c = (c_1, c_2, c_3)$ a ciphertext for $m \in \mathbf{G}$, the attacker can get some information on m. For example, the proposition tells us that during a "lunchtime" attack, an attacker can solve the splitting problem and compute elements $x_1, x_2, x_3 \in \mathbf{G}_1 \times \mathbf{G}_2 \times \mathbf{G}_3$. As a result,

he can compute, $e(c_i, x_i) = e(m_i, x_i)$ for $i \in \{1, \ldots, 3\}$. The product of these three pairings evaluations gives $e(m, x)$. If x is a generator, the adversary can further get the pairing evaluation of m with elements of $\mathbf{G}$ of his choice. Note that this lunchtime attack in not a full break, the adversary only gets a piece of information on the plaintext. Moreover this attack does not apply in $\mathbf{G_t}$. Note also that Proposition 1 can be generalized for all k.

4.4 Application to Shared Decryption

Our cryptosystem uses three projections whose kernels are subgroups of coprime orders. This particular setting makes it possible to design an original shared decryption process. Suppose that $c = (c_1, c_2, c_3)$ is an encryption of $m \in \mathbf{G}$. The goal is that three entities $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, cooperate to decrypt c. Moreover, we want to achieve some kind of robustness, *i.e.,* that each entity can check if the other ones give correct results. The protocol is a simple modification of our cryptosystem (see Figure 1) as follows: at the end of the KeyGen algorithm, performed by a trusted dealer, each $\mathcal{A}_i$ is given the public key together with the prime q_i. The Encrypt, EvalMul and EvalPair algorithms remain unchanged. During the new Decrypt algorithm, each entity recovers $m_i := c_i^{u_i(n/q_i)}$ where u_i is the inverse of n/q_i modulo q_i. Then, in a reconstruction phase, each party broadcasts m_i to the others and each party can recover the plaintext message $m = m_1 m_2 m_3$. The correctness of the decryption follows from Remark 1. Moreover, before the reconstruction, each entity $\mathcal{A}_i$ can check the validity of the message sent by the others. Without loss of generality, $\mathcal{A}_1$ can compute a random element $x_2 \in \mathbf{G}_2$ (resp. $x_3 \in \mathbf{G}_3$) by selecting a random power of $h_3^{q_1}$ (resp. of $h_2^{q_1}$). Following the discussion at the end of the previous subsection, $\mathcal{A}_1$ accepts m_2 and m_3 if and only if $e(c_i, x_i)$ equals $e(m_i, x_i)$ for $i \in \{2, 3\}$.

This process can be easily extended to more participants by using our construction with $k > 3$. We note that in this protocol, each $\mathcal{A}_i$ learns a part of the secret key and can break the semantic security of the scheme as he can generate elements of $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ and solve the subgroup membership problem (as in the case $k = 2$). However, we believe that this protocol is of interest because of its simplicity and originality compared to standard secret sharing techniques.

5 Comparison with Other Works and Conclusion

As we saw in subsection 4.2, the BGN scheme from [BGN05] is quite similar to ours but with $k = 2$. In that cryptosystem, only *small* plaintext messages m of $\mathbf{N}$ are encoded with the exponentiation $g \mapsto g^m$. This encoding allows to compute sums of messages by computing product of points and to get products with the pairing evaluations. We can also use this encoding in our cryptosystems to get such homomorphic properties. Contrary to our scheme, in the BGN cryptosystem one cannot get encryption of product of arbitrary points, and one cannot get encryption of pairings and of product of pairings. Thus the properties of our scheme are quite different from the ones of BGN.

In [BWY11,Lew12] a general subgroup decision problem is formulated, unifying several decision assumptions made in bilinear composite groups this past few years in the area of (hierarchical) identity-based encryption. This decision problem is different from GSSMP (see Def. 2): two of the subgroups play a different roles from the others, whereas in the problem we consider the role played by all subgroups $\mathbf{H}_i$ to be the same.

In [Fre10], Freeman provides a framework to translate features of composite-order bilinear groups in the prime-order setting. To this purpose, he defines two kinds of property for pairing: *cancelling* and *projecting*. Projecting intuitively means that the pairing and some projections maps commute. This is the core of our construction: a projection map is used in the decryption algorithm, since a ciphertext is projected in $\mathbf{G}_1 \times \mathbf{G}_2 \times \mathbf{G}_3 \simeq \mathbf{G}/\mathbf{H}_1 \times \mathbf{G}/\mathbf{H}_2 \times \mathbf{G}/\mathbf{H}_3$, and the product of each terms gives the plaintext message (cf. Remark 1). The fact that the projection and the pairing commute ensures that the pairing of two ciphertexts in $\mathbf{G}^3$ decrypts to the pairing of the corresponding plaintexts.

Our cryptosystem can thus be adapted in the prime-order setting following Freeman's construction of a projecting pairing to convert the BGN cryptosystem. For example, we can obtain a cryptosystem satisfying Definition 1 as follows: Let $e : G \times G \to G_t$ be a symmetric pairing where G and G_t are groups of prime order q. Freeman's framework (cf. [Fre10, subsection 3.1]) allows to construct a subgroup $\mathbf{H}$ of $\mathbf{G} = G^3$, a pairing $\hat{e} : \mathbf{G} \times \mathbf{G} \to G_t^9$ and a subgroup $\mathbf{H}_t$ of $\mathbf{G}_t := G_t^9$ such that there exits maps $\pi_1 : \mathbf{G} \to \mathbf{G}$ and $\pi_t : \mathbf{G}_t \to \mathbf{G}_t$ with $\mathbf{H} \subset \ker \pi_1$, $\mathbf{H}_t \subset \ker \pi_t$ and $\hat{e}(\pi_1(x), \pi_1(y)) = \pi_t(\hat{e}(x, y))$, for all $(x, y) \in \mathbf{G}^2$. The public key consists of $\mathbf{G}, \mathbf{H}, \mathbf{G}_t$ and $\mathbf{H}_t$. The private key is the maps (π_1, π_t). To encrypt $m \in G$, one computes $c = (m, m, m)h$ where h is a random element of $\mathbf{H}$. Decryption of c is done by applying π_1, which gives $\pi_1((m, m, m))$. From that, m is recovered as the first element is a power of m, m^s where s is an explicit non zero element of $\mathbf{F}_q$. Decryption in $\mathbf{G}_t$ is carried out in the same way with the map π_t. The scheme is homomorphic for multiplication and for pairing evaluation thanks to the projecting property.

As for the BGN cryptosystem, this conversion gives a more efficient scheme in terms of key size and computation cost. The $\mathsf{ind-cpa}$ security of the converted scheme relies on the Decision Linear Problem.

Our framework also uses a pairing with the cancelling property since we have a decomposition $\mathbf{G} = \mathbf{G}_1\mathbf{G}_2\mathbf{G}_3$ such that $e(g_i, g_j) = 1$ if $g_i \in \mathbf{G}_i$ and $g_j \in \mathbf{G}_j$ with $i \neq j$. This cancelling property is needed for the proof of the result on $\mathsf{ind-cca1}$ security of Proposition 1. Moreover, this property and the relation with the splitting problem is also the core of our application to shared decryption. These properties do not remain after the conversion.

In [MSF10,SC12], the problem of the transposition of *all* cryptosystems using composite-order bilinear groups in prime-order groups is discussed. In [SC12] a prime-order construction with both cancelling and projecting properties is given, together with a new security proof of the blind signature scheme of [MSF10] in the prime-order setting, which was believed impossible to get outside composite bilinear group.

We leave as open the problem of proving that the additional properties of our cryptosystem, which need particular projecting *and* cancelling maps, can or can not be instantiated in prime-order groups with a direct approach. An impossible result would answer the open problem left in [SC12].

References

[APK10] F. Armknecht, A. Peter and S. Katzenbeisser. *Group Homomorphic Encryption: Characterizations, Impossibility Results, and Applications.* To appear in Des. Codes Cryptography. Available as IACR e-print 2010/501, http://eprint.iacr.org/2010/501, (2010)

[BDPR98] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. *Relations Among Notions of Security for Public-Key Encryption Schemes.* Proc. of Crypto'98, Springer LNCS Vol. 1462, 26–45 (1998)

[Ben88] J. C. Benaloh. *Verifiable Secret-Ballot Elections.* PhD thesis, Yale University (1988)

[BF03] D. Boneh and M. K. Franklin. *Identity-Based Encryption from the Weil Pairing.* SIAM J. Comput, 32(3), 586–615 (2003)

[BFP+01] O. Baudron, P.-A. Fouque, D. Pointcheval, G. Poupard and J. Stern. *Practical Multi-Candidate Election System.* Proc. of PODC'01, 274–283 (2001)

[BGN05] D. Boneh, E.-J. Goh and K. Nissim. *Evaluating 2-DNF Formulas on Ciphertexts.* Proc. of TCC'05, Springer LNCS Vol. 3378, 325–341 (2005)

[BP04] M. Bellare and A. Palacio. *Towards Plaintext-Aware Public-Key Encryption without Random Oracles.* Proc. of Asiacrypt'04, Springer LNCS Vol. 3329, 37–52 (2004)

[Bro07] J. Brown. *Secure Public-Key Encryption from Factorisation-related problem.* PhD Thesis, Queensland University of Technology (2007)

[BRS11] D. Boneh, K. Rubin and A. Silverberg. *Finding composite order ordinary elliptic curves using the Cocks-Pinch method.* Journal of Number Theory, 131(5), 832–841, (2011)

[BWY11] M. Bellare, B. Waters and S. Yilek. *Identity-Based Encryption Secure Against Selective Opening Attack.* Proc. of TCC'11, Springer LNCS Vol. 6597, 235–252 (2011)

[BGV12] Z. Brakerski, C. Gentry and V. Vaikuntanathan. *Fully Homomorphic Encryption without Bootstrapping.* To appear in Proc. of Innovations in Theoretical Computer Science (ITCS) 2012

[BV11] Z. Brakerski and V. Vaikuntanathan. *Efficient Fully Homomorphic Encryption from (Standard) LWE.* Proc. of FOCS 2011, IEEE, 97–106 (2011)

[DJ01] I. Damgård and M. J. Jurik. *A Generalisation, a Simplification and some Applications of Paillier's Probabilistic Public-Key System.* Proc. of PKC'01, Springer LNCS Vol. 1992, 119–136 (2001)

[Fre10] D. M. Freeman. *Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups.* Proc. of Eurocrypt'10, Springer LNCS Vol. 6110, 44–61, (2010)

[FPS00] P.-A. Fouque, G. Poupard and J. Stern. *Sharing Decryption in the Context of Voting or Lotteries.* Proc. of Financial Crypto'00, Springer LNCS Vol. 1962, 90–104 (2000)

[GBD05] J. M. González Nieto, C. Boyd and E. Dawson. *A Public Key Cryptosystem Based on a Subgroup Membership Problem.* Des. Codes Cryptography, 36(3), 301–316 (2005)

[Gen09] C. Gentry. *Fully homomorphic encryption using ideal lattices*. Proc. of STOC 2009, ACM, 169–178 (2009)

[Gjo04] K. Gjøsteen. *Subgroup membership problems and public key cryptography*. PhD Thesis, Norwegian University of Science and Technology (2004)

[Gjo05] K. Gjøsteen. *Symmetric Subgroup Membership Problems*. Proc. of PKC'05, Springer LNCS Vol. 3386, 104–119 (2005)

[GM84] S. Goldwasser and S. Micali. *Probabilistic Encryption*. JCSS, 28(2), 270–299 (1984)

[JS08] T. Jager and J. Schwenk. *The Generic Hardness of Subset Membership Problems under the Factoring Assumption*. IACR e-print 2008/482, http://eprint.iacr.org/2008/482, (2008)

[KSW08] J. Katz, A. Sahai and B. Waters. *Predicate encryption supporting disjunctions, polynomial equations, and inner products*. Proc. of Eurocrypt'08, Springer LNCS Vol. 4965, 146–162, (2008)

[Jur03] M. Jurik. *Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols*, PhD thesis, Århus University (2003)

[Lip05] H. Lipmaa. *An Oblivious Transfer Protocol with Log-Squared Communication*, Proc. of ISC'05, Springer LNCS Vol. 3650, 314–328 (2005)

[Lew12] A. Lewko. *Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting*. To appear in Proc. of Eurocrypt 2012, Available as IACR e-print 2011/490, http://eprint.iacr.org/2011/490.pdf (2012)

[Mil04] V. S. Miller. *The Weil Pairing, and Its Efficient Calculation*. J. Cryptology, 17(4), 235–261 (2004)

[MMO10] T. Mitsunaga, Y. Manabe and T. Okamoto. *Efficient Secure Auction Protocols Based on the Boneh-Goh-Nissim Encryption*. Proc. of IWSEC 2010, Springer LNCS Vol. 6434, 149–163 (2010)

[MSF10] S. Meiklejohn, H. Shacham and D. M. Freeman. *Limitations on Transformations from Composite-Order to Prime-Order Groups: The Case of Round-Optimal Blind Signatures*. Proc. of Asiacrypt 2010, Springer LNCS Vol. 6477, 519–538 (2010)

[NS98] D. Naccache and J. Stern. *A New Public Key Cryptosystem Based on Higher Residues*. Proc. of CCS'98, 546–560 (1998)

[NSNK06] L. Nguyen, R. Safavi-Naini and K. 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] T. Okamoto and S. Uchiyama. *A New Public-Key Cryptosystem as Secure as Factoring*. Proc. of Eurocrypt'98, Springer LNCS Vol. 1403, 308–318 (1998)

[Pai99] P. Paillier. *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*. Proc. of Eurocrypt'99, Springer LNCS Vol. 1592, 223–238 (1999)

[SC12] J. H. Seo and J. H. Cheon. *Beyond the Limitation of Prime-Order Bilinear Groups, and Round Optimal Blind Signatures*. Proc. of TCC'12, Springer LNCS Vol. 7194, 133–150 (2012).