Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2007, Article ID 13801, 10 pages
doi:10.1155/2007/13801

Review Article

A Survey of Homomorphic Encryption for Nonspecialists

Caroline Fontaine and Fabien Galand

CNRS/IRISA-TEMICS, Campus de Beaulieu, 35042 Rennes Cedex, France

Correspondence should be addressed to Caroline Fontaine, caroline.fontaine@irisa.fr

Received 30 March 2007; Revised 10 July 2007; Accepted 24 October 2007

Recommended by Stefan Katzenbeisser

Processing encrypted signals requires special properties of the underlying encryption scheme. A possible choice is the use of homomorphic encryption. In this paper, we propose a selection of the most important available solutions, discussing their properties and limitations.

1. INTRODUCTION

The goal of encryption is to ensure confidentiality of data in communication and storage processes. Recently, its use in constrained devices led to consider additional features, such as the ability to delegate computations to untrusted computers. For this purpose, we would like to give the untrusted computer only an encrypted version of the data to process. The computer will perform the computation on this encrypted data, hence without knowing anything on its real value. Finally, it will send back the result, and we will decrypt it. For coherence, the decrypted result has to be equal to the intended computed value if performed on the original data. For this reason, the encryption scheme has to present a particular structure. Rivest et al. proposed in 1978 to solve this issue through *homomorphic encryption* [1]. Unfortunately, Brickell and Yacobi pointed out in [2] some security flaws in the first proposals of Rivest et al. Since this first attempt, a lot of articles have proposed solutions dedicated to numerous application contexts: secret sharing schemes, threshold schemes (see, e.g., [3]), zero-knowledge proofs (see, e.g., [4]), oblivious transfer (see, e.g., [5]), commitment schemes (see, e.g., [3]), anonymity, privacy, electronic voting, electronic auctions, lottery protocols (see, e.g., [6]), protection of mobile agents (see, e.g., [7]), multiparty computation (see, e.g., [3]), mix-nets (see, e.g., [8, 9]), watermarking or fingerprinting protocols (see, e.g., [10–14]), and so forth.

The goal of this article is to provide nonspecialists with a survey of homomorphic encryption techniques. Section 2 recalls some basic concepts of cryptography and presents homomorphic encryption; it is particularly aimed at noncryptographers, providing guidelines about the main characteristics of encryption primitives: algorithms, performance, security. Section 3 provides a survey of homomorphic encryption schemes published so far, and analyses their characteristics.

Most schemes we describe are based on mathematical notions the reader may not be familiar with. In the cases these notions can easily be introduced, we present them briefly. The reader may refer to [15] for more information concerning those we could not introduce properly, or algorithmic problems related to their computation.

Before going deeper in the subject, let us introduce some notation. The integer $\ell(x)$ denotes the number of bits constituting the binary expansion of x. As usual, $\mathbf{Z}_n$ will denote the set of integers modulo n, and $\mathbf{Z}_n^*$ the set of its invertible elements.

2. TOWARDS HOMOMORPHIC ENCRYPTION

2.1. Basics about encryption

In this section, we will recall some important concepts concerning encryption schemes. For more precise information, the reader may refer to [16] or the more recent [17].

Encryption schemes are, first and foremost, designed to preserve confidentiality. According to Kerckoffs' principle (see [18, 19] for the original papers, or any book on cryptography), their security must not rely on the obfuscation of their code, but only on the secrecy of the decryption key. We can distinguish two kinds of encryption schemes: *symmetric* and *asymmetric* ones. We will present them shortly and discuss their performance and security issues.

Symmetric encryption schemes

Here "symmetric" means that encryption and decryption are performed with the same key. Hence, the sender and the receiver have to agree on the key they will use before performing any secure communication. Therefore, it is not possible for two people who never met to use such schemes directly. This also implies to share a different key with every one we want to communicate with. Nevertheless, symmetric schemes present the advantage of being really fast and are used as often as possible. In this category, we can distinguish block ciphers (AES [20, 21])[1] and stream ciphers (One-time pad presented in Figure 1 [22], Snow 2.0 [23]),[2] which are even faster.

Asymmetric encryption schemes

In contrast to the previous family, asymmetric schemes introduce a fundamental difference between the abilities to encrypt and to decrypt. The encryption key is public, as the decryption key remains private. When Bob wants to send an encrypted message to Alice, he uses her *public key* to encrypt the message. Alice will then use her *private key* to decrypt it. Such schemes are more functional than symmetric ones since there is no need for the sender and the receiver to agree on anything before the transaction. Moreover, they often provide more features. These schemes, however, have a big drawback: they are based on nontrivial mathematical computations, and much slower than the symmetric ones. The two most prominent examples, RSA [24] and ElGamal [25], are presented in Figures 2 and 3.

Performance issues

A block cipher like AES is typically 100 times faster than RSA encryption and 2000 times than RSA decryption, with about 60 MB per second on a modest platform. Stream ciphers are even faster, some of them being able to encrypt/decrypt 100 MB per second or more.[3] Thus, while encryption or decryption of the whole content of a DVD will take about a minute with a fast stream cipher, it is simply not realistic to use an asymmetric cipher in practice for such a huge amount of data as it would require hours, or even days, to encrypt or decrypt.

Hence, in practice, it is usual to encrypt the data we want to transmit with an efficient symmetric cipher. To provide the receiver with the secret key needed to recover the data, the sender encrypts this key with an asymmetric cipher. Hence, the asymmetric cipher is used to encrypt only a short data, while the symmetric one is used for the longer one. The sender and the receiver do not need to share anything before performing the encryption/decryption as the symmetric key is transmitted with the help of the public key of the receiver. Proceeding this way, we combine the advantages of both: efficiency of symmetric schemes and functionalities of the asymmetric schemes.

Security issues

Security of encryption schemes was formalized for the first time by Shannon [26]. In his seminal paper, Shannon introduced the notion of *perfect secrecy/unconditional security*, which characterizes encryption schemes for which the knowledge of a ciphertext does not give any information either about the corresponding plaintext or about the key. He proved that the one-time pad is perfectly secure under some conditions, as explained in Figure 1. In fact, no other scheme, neither symmetric nor asymmetric, has been proved unconditionally secure. Hence, if we omit the one-time pad, any encryption scheme's security is evaluated with regard to the computational power of the opponent. In the case of asymmetric schemes, we can rely on their mathematical structure to estimate their security level in a formal way. They are based on some well-identified mathematical problems which are hard to solve in general, but easy to solve for the one who knows the trapdoor, that is, the owner of the keys. Hence, it is easy for the owner of the keys to compute his/her private key, but no one else should be able to do so, as the knowledge of the public key should not endanger the private key. Through reductions, we can compare the security level of these schemes with the difficulty of solving these mathematical problems (factorizing large integers or computing a discrete logarithm in a large group) which are famous for their hardness. Proceeding this way, we obtain an estimate of the security level, which sometimes turns out to be optimistic. This estimation may not be sufficient for several reasons. First, there may be other ways to break the system than solving the reference mathematical problem [27, 28]. Second, most of security proofs are performed in an idealized model called the *random oracle model*, in which involved primitives, for example, hash functions, are considered truly random. This model has allowed the study of the security level for numerous asymmetric ciphers. Recent works show that we are now able to perform proofs in a more realistic model called the *standard model*. From [29] to [30], a lot of papers compared these two models, discussing the gap between them. In parallel with this formal estimation of the security level, an empirical one is performed in any case, and new symmetric and asymmetric schemes are evaluated according to published attacks.

The framework of a security evaluation has been stated by Shannon in 1949 [26]: all the considered messages are encrypted with the same key—so, for the same recipient—and the opponent's challenge is to take an advantage from all his observations to disclose the involved secret/private key.

[1] AES has been standardized; see http://csrc.nist.gov/groups/ST/toolkit/block_ciphers.html for more details.

[2] Snow 2.0 is included in the draft of Norm ISO/IEC 18033-4, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=3997.

[3] See, for example, http://www.ecrypt.eu.org/stream/perf/alpha/benchmarks/snow-2.0 for some benchmark of Snow 2.0, or openssl for AES and RSA.

Usually, to evaluate the attack capacity of the opponent, we distinguish among several contexts [31]: *ciphertext-only attacks* (where the opponent has access only to some ciphertexts), *known-plaintext attacks* (where the opponent has access to some pairs of corresponding plaintext-ciphertexts), *chosen-plaintext attacks* (same as previous, but the opponent can choose the plaintexts and get the corresponding ciphertexts), and *chosen-ciphertext attacks* (the opponent has access to a decryption oracle, behaving as a black-box, that takes a ciphertext and outputs the corresponding plaintext). The first context is the most frequent in real life, and results from eavesdropping the communication channel; it is the worst case for the opponent. The other cases may seem difficult to achieve, and may arise when the opponent has a more powerful position; he may, for example, have stolen some plaintexts, or an encryption engine. The "chosen" ones exist in *adaptive* versions, where the opponent can wait for a computation result before choosing the next input.

How do we choose the right scheme?

The right scheme is the one that fits your constraints in the best way. By constraints, we may understand constraints in time, memory, security, and so forth. The two first criteria are very important in highly constrained architectures, often encountered in very small devices (PDAs, smart cards, RFID tags, etc.). They are also important if we process a huge amount of data, or numerous data at the same time, for example, video streams. Some schemes as AES or RSA are usually chosen because of their reputation, but it is important to note that new schemes are proposed each year. Indeed, it is necessary to keep a diversity in the proposals. First, it is necessary in order to be able to face new kinds of requirements. Second, because of security purpose, having all the schemes relying on the same structure may lead to a disaster in case an attack breaks this structure. Hence, huge international projects have been funded to ask for new proposals, with a fair evaluation to check their advantages and drawbacks, for example, RIPE, NESSIE,[4] and NIST's call for the design of the AES,[5] CRYPTREC,[6] ECRYPT,[7]and so forth.

2.2. Probabilistic encryption

The most well-known cryptosystems are *deterministic*: for a fixed encryption key, a given plaintext will always be encrypted in the same ciphertext. This may lead to some drawbacks. RSA is a good example to illustrate this point:

(i) particular plaintexts may be encrypted in a too much structured way: with RSA, messages 0 and 1 are always encrypted as 0 and 1, respectively;

(ii) it may be easy to compute partial information about the plaintext: with RSA, the ciphertext c leaks one bit of information about the plaintext m, namely, the so-called Jacobi symbol;

(iii) when using a deterministic encryption scheme, it is easy to detect when the same message is sent twice while processed with the same key.

So, in practice, we prefer encryption schemes to be probabilistic. In the case of symmetric schemes, we introduce a random vector in the encryption process (e.g., in the pseudorandom generator for stream ciphers, or in the operating mode for block ciphers), generally called IV. This vector may be public, and transmitted as it is, without being encrypted, but IV must be changed every time we encrypt a message. In the case of asymmetric ciphers, the security analysis is more mathematical, and we want the randomized schemes to remain analyzable in the same way as the deterministic schemes. Some adequate modes have been proposed to randomize already published deterministic schemes, as the Optimal Asymmetric Encryption Padding OAEP for RSA (or any scheme based on a trap-door one-way permutation) [33].[8] Some new schemes, randomized by nature, have also been proposed [25, 34, 35] (see also Figures 3 and 4).

A simple consequence of this requirement to be probabilistic appears in the so-called *expansion*: since for a plaintext we require the existence of several possible ciphertexts, the number of ciphertexts is greater than the number of possible plaintexts. This means the ciphertexts cannot be as short as the plaintexts, they have to be strictly longer. The ratio between the length, in bits, of ciphertexts and plaintexts is called the *expansion*. Of course, this parameter is of practical importance. We will see in the sequel that efficient probabilistic encryption schemes have been proposed with an expansion less than 2 (e.g., Paillier's scheme).

2.3. Homomorphic encryption

We will present in this section the basic definitions related to *homomorphic* encryption. The state of the art will be given in Section 3.

The most common definition is the following. Let $\mathcal{M}$ (resp., $\mathcal{C}$) denote the set of the plaintexts (resp., ciphertexts). An encryption scheme is said to be *homomorphic* if for any given encryption key k the encryption function E satisfies

$$\forall m_1, m_2 \in \mathcal{M}, \quad E(m_1 \odot_{\mathcal{M}} m_2) \longleftarrow E(m_1) \odot_{\mathcal{C}} E(m_2) \qquad (1)$$

for some operators $\odot_{\mathcal{M}}$ in $\mathcal{M}$ and $\odot_{\mathcal{C}}$ in $\mathcal{C}$, where $\leftarrow$ means "can be directly computed from," that is, without any intermediate decryption.

If $(\mathcal{M}, \odot_{\mathcal{M}})$ and $(\mathcal{C}, \odot_{\mathcal{C}})$ are groups, we have a *group homomorphism*. We say a scheme is *additively homomorphic* if we consider addition operators, and *multiplicatively homomorphic* if we consider multiplication operators.

A lot of such homomorphic schemes have been published that have been widely used in many applications. Note that

[4] see http://www.cryptonessie.org.

[5] see http://csrc.nist.gov and http://csrc.nist.gov/CryptoToolkit/aes.

[6] see http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html.

[7] see http://www.ecrypt.eu.org.

[8] Note that there are a lot of more recent papers proposing variants or improvements of OAEP, but it is not our purpose here.

Prerequisite: Alice and Bob share a secret random keystream, say a binary one.
Goal: Alice can send an encrypted message to Bob, and Bob can send an encrypted message to Alice.
Principle: To encrypt a message, Alice (resp., Bob) XORs the plaintext and the keystream. To decrypt the received message, Bob (resp., Alice) applies XOR on the ciphertext and the keystream.
Security: This scheme has been showed to be unconditionally secure by Shannon [26] if and only if the keystream is truly random, has the same length as the plaintext, and is used only once. Thus, this scheme is used only for very critical situations for which these constraints may be managed, as the red phone used by the USA and the USSR [32, pp. 715-716]. What we may use more commonly is a similar scheme, where the keystream is generated by a pseudorandom generator, initialized by the secret key shared by Alice and Bob. A lot of such stream ciphers has been proposed, and their security remains only empirical. Snow 2.0 is one of these.

Figure 1: One-time pad—1917(used)/1926 (published [22]). Note that this scheme may be transposed in any group $(G, +)$ other than $(\{0, 1\}, \text{XOR})$, encryption being related to addition of the keystream, while decryption consists in subtracting the keystream.

Prerequisite: Alice computed a (public, private) key: an integer $n = pq$, where p and q are well chosen large prime numbers, an integer e such that $\gcd(e, \phi(n)) = 1$, and an integer d which is the inverse of e modulo $\phi(n)$, that is, $ed \equiv 1 \bmod \phi(n)$; $\phi(n)$ denotes the Euler function, $\phi(n) = \phi(pq) = (p-1)(q-1)$. Alice's public key is (n, e), and her private key is d; p and q have also to be kept secret, but are no more needed to process the data, they were only useful for Alice to compute d from e.
Goal: Anyone can send an encrypted message to Alice.
Principle: To send an encrypted version of the message m to Alice, Bob computes $c = m^e \mod n$. To get back to the plaintext, Alice computes $c^d \mod n$ which, according to Euler's theorem, is precisely equal to m.
Security: It is clear that if an opponent may factor n and recover p and q, he will be able to compute $\phi(n)$, then d, and will be able to decrypt Alice messages. So, the RSA problem (accessing m while given c) is weaker than the factorization problem. It is not known whether the two problems are equivalent or not.

Figure 2: RSA—1978 [24].

in some contexts it may be of great interest to have this property not only for one operator but for two at the same time. Hence, we are also interested in the design of *ring/algebraic homomorphisms*. Such schemes would satisfy a relation of the form

$$\begin{aligned} \forall m_1, m_2 \in \mathcal{M}, \quad E(m_1 +_{\mathcal{M}} m_2) &\longleftarrow E(m_1) +_{\mathcal{C}} E(m_2), \\ E(m_1 \times_{\mathcal{M}} m_2) &\longleftarrow E(m_1) \times_{\mathcal{C}} E(m_2). \end{aligned} \tag{2}$$

As it will be further discussed, no convincing algebraic homomorphic encryption scheme has been found yet, and their design remains an open problem.

Less formally, these definitions mean that, for a fixed key k, it is equivalent to perform operations on the plaintexts before encryption, or on the corresponding ciphertexts after encryption. So we require a kind of commutativity between encryption and some data processing operations.

Of course, the schemes we will consider in the following have to be probabilistic ciphers, and we may consider E to behave in a probabilistic way in the above definitions.

2.4. New security considerations

Probabilistic encryption was introduced with a clear purpose: security. This requires to properly define different security levels. *Semantic security* was introduced in [34], at the same time as probabilistic encryption, in order to define what could be a strong security level, unavailable without probabilistic encryption. Roughly, a probabilistic encryption is *semantically secure* if the knowledge of a ciphertext does not provide any useful information on the plaintext to some hypothetical adversary having only a reasonably restricted computational power. More formally, for any function f and any plaintext m, and with only polynomial resources (that is, with algorithms which time/space complexities vary as a polynomial function of the size of the inputs), the probability to guess $f(m)$ (knowing f but not m) does not increase if the adversary knows a ciphertext corresponding to m. This might be thought of as a kind of perfect secrecy in the case when we only have polynomial resources.

Together with this strong requirement, the notion of *polynomial security* was defined: the adversary chooses two plaintexts, and we choose secretly at random one plaintext and provide to the adversary a corresponding ciphertext. The adversary, still with polynomial resources, must guess which plaintext we chose. If the best he can do is to achieve a probability $1/2 + \varepsilon$ of success, the encryption is said to be *polynomially secure*. Polynomial security is now known as the *indistinguishability of encryptions* following the terminology and definitions of Goldreich [36].

Quite amazingly, Goldwasser and Micali proved the equivalence between polynomial security and semantic security [34]; Goldreich extended these notions [36] preserving the equivalence. With this equivalence, it is easy to state that a deterministic asymmetric encryption scheme cannot be semantically secure since it cannot be indistinguishable: the adversary knows the encryption function, and thus can compute the single ciphertext corresponding to each plaintext.

Prerequisite: Alice generated a (public, private) key: she first chose a large prime integer p, a generating element g of the cyclic group $\mathbf{Z}_p^*$, and considered $q = p - 1$, the order of the group; building her public key, she picked at random $a \in \mathbf{Z}_q$ and computed $y_A = g^a$ in $\mathbf{Z}_p^*$, her public key being then (g, q, y_A); her private key is a.
Goal: Anyone can send an encrypted message to Alice.
Principle: To send an encrypted version of the message m to Alice, Bob picks at random $k \in \mathbf{Z}_q$, computes $(c_1, c_2) = (g^k, my_A^k)$ in $\mathbf{Z}_p^*$. To get back to the plaintext, Alice computes $c_2(c_1^a)^{-1}$ in $\mathbf{Z}_p^*$, which is precisely equal to m.
Security: The security of this scheme is related to the Diffie-Hellman problem: if we can solve it, then we can break ElGamal encryption. It is not known whether the two problems are equivalent or not. This scheme is IND-CPA.

Figure 3: ElGamal—1985 [25].

But with asymmetric encryption schemes, the adversary knows the whole encryption material E involving both the encryption function and the encryption key. Thus, he can compute any pair $(m, E(m))$. Naor and Yung [37] and Rackoff and Simon [38] introduced different abilities, relying on the different contexts we discussed above. From the weakest to the strongest, we have the chosen-plaintext, nonadaptive chosen ciphertext and the strongest is the adaptive chosen ciphertext. This leads to the IND-CPA, IND-CCA1, and IND-CCA2 notions in the literature. IND stands for indistinguishability whereas CPA and CCA are acronyms for chosen plaintext attack and chosen-ciphertext attack. Finally, CCA1 refers to nonadaptive attacks, and CCA2 to adaptive ones. Considering the previous remarks on the ability for anyone to encrypt while using asymmetric schemes, the adversary has always the chosen-plaintext ability.

Another security requirement termed *nonmalleability* has also been introduced to complete the analysis. Given a ciphertext $c = E(m)$, it should be hard for an opponent to produce a ciphertext c' such that the corresponding plaintext m', that is not necessary known to the opponent, has some known relation with m. This notion was formalized differently by Dolev et al. [39, 40], and by Bellare et al. [41], both approaches being proved equivalent by Bellare and Sahai [42].

We will not detail the relations between all these different notions and the interested reader can refer to [41–43] for a comprehensive treatment. Basically, the adaptive chosen-ciphertext indistinguishability IND-CCA2 is the strongest requirement for an encryption; in particular, it implies nonmalleability.

It should be emphasized that a homomorphic encryption cannot have the nonmalleability property. With the notation of Section 2.3, knowing c, we can compute $c' = c \odot_{\mathcal{C}} c$ and deduce, by the homomorphic property, that c' is a ciphertext of $m' = m \odot_{\mathcal{M}} m$. According to the previous remark on adaptive chosen-ciphertext indistinguishability, an homomorphic encryption has no access to the strongest security requirement. The highest security level it can reach is IND-CPA.

To conclude this section on security, and for the sake of completeness, we point out some security considerations about deterministic homomorphic encryption. First, it was proved that a deterministic homomorphic encryption for which the operation $\odot$ is a simple addition is insecure [44]. Second, Boneh and Lipton showed in 1996 that any deterministic algebraically homomorphic cryptosystem can be broken in subexponential time [45]. Note that this last point does not mean that deterministic algebraically homomorphic cryptosystems are insecure, but that one can find the plaintext from a ciphertext in a subexponential time (which is still too long to be practicable). For example, we know that the security of RSA encryption depends on factorization algorithms and we know subexponential factorization algorithm. Nevertheless, RSA is still considered strong enough.

3. HOMOMORPHIC ENCRYPTION: STATE OF THE ART

First of all, let us recall that both RSA and ElGamal encryption schemes are multiplicatively homomorphic. The problem is that the original RSA being deterministic, it cannot achieve a security level of IND-CPA (which is the highest security level for homomorphic schemes, see Section 2.4). Furthermore its probabilistic variants, obtained through OAEP/OAEP+, are no more homomorphic. In contrast to RSA, ElGamal offers the best security level for a homomorphic encryption scheme, as it has been shown to be IND-CPA. Moreover, it is interesting to notice that an additively homomorphic variant of ElGamal has also been proposed [48]. Comparing it with the original ElGamal, this variant also involves an element G (G may be equal to g) that generates $(\mathbf{Z}_q, +)$ with respect to the addition operation. To send an encrypted version of the message m to Alice, Bob picks at random $k \in \mathbf{Z}_q$ and computes $(c_1, c_2) = (g^k, G^m y_A^k)$. To get back the plaintext, Alice computes $c_2(c_1^a)^{-1}$, which is equal to G^m; then, she has to compute m in a second step. Note that this last decryption step is hard to achieve and that there is no other choice for Alice than to use brute force search to get back m from G^m. It is also well known that ElGamal's construction works for any family of groups for which the discrete logarithm problem is considered intractable. For example, it may be derived in the setup employing elliptic curves. Hence, ElGamal and its variants are known to be really interesting candidates for realistic homomorphic encryption schemes.

We will now describe another important family of homomorphic encryption schemes, ranging from the first probabilistic system[9] proposed by Goldwasser and Micali in 1982

[9] To be more precise, the first published probabilistic public-key encryption scheme is due to McEliece [49], and the first to add the homomorphic property is due to Goldwasser-Micali.

Prerequisite: Alice computed a (public, private) key: she first chose $n = pq$, p and q being large prime numbers, and g a quadratic nonresidue modulo n whose Jacobi symbol is 1; her public key is composed of n and g, and her private key is the factorization of n.

Goal: Anyone can send an encrypted message to Alice.

Principle: To encrypt a bit b, Bob picks at random an integer $r \in \mathbf{Z}_n^*$, and computes $c = g^b r^2 \bmod n$ (remark that c is a quadratic residue if and only if $b = 0$). To get back to the plaintext, Alice determines if c is a quadratic residue or not. To do so, she uses the property that the Jacobi symbol (c/p) is equal to $(-1)^b$. Please, note that the scheme encrypts 1 bit of information, while its output is usually 1024 bits long!

Security: This scheme is the first one that was proved semantically secure against a passive adversary (under computational assumption).

Figure 4: Goldwasser-Micali—1982 [34, 46].

Prerequisite: Alice computed a (public, private) key: she first chose an integer $n = pq$, p and q being two large prime numbers and n satisfying gcd $(n, \phi(n)) = 1$, and considered the group $G = \mathbf{Z}_{n^2}^*$ of order k. She also considered $g \in G$ of order n. Her public key is composed of n and g, and here private key consists in the factors of n.

Goal: Anyone can send a message to Alice.

Principle: To encrypt a message $m \in \mathbf{Z}_n$, Bob picks at random an integer $r \in \mathbf{Z}_n^*$, and computes $c = g^m r^n \bmod n^2$. To get back to the plaintext, Alice computes the discrete logarithm of $c^{\lambda(n)} \bmod n^2$, obtaining $m\lambda(n) \in \mathbf{Z}_n$, where $\lambda(n)$ denotes the Carmichael function. Now, since gcd $(\lambda(n), n) = 1$, Alice easily computes $\lambda(n)^{-1} \bmod n$ and gets m.

Security: This scheme is IND-CPA.

Figure 5: Paillier—1999 [47].

[34, 46] (described in Figure 4), to the famous Paillier's encryption scheme [47] (described in Figure 5) and its improvements. Paillier's scheme and its variants are famous for their efficiency, but also because, as ElGamal, they achieve the highest security level for homomorphic encryption schemes. We will not discuss their mathematical considerations in detail, but will summarize their important parameters and properties.

(i) We begin with the rather simple scheme of Goldwasser-Micali [34, 46]. Besides some historical importance, this scheme had an important impact on later proposals. Several other schemes, that will be presented below, were obtained as generalizations of this one. For these reasons, we provide a detailed description in Figure 4. Here, as for RSA, we use computations modulo $n = pq$, a product of two large primes. Encryption is simple, with a product and a square, whereas decryption is heavier, with an exponentiation. Nevertheless, this step can be done in $\mathcal{O}(\ell(p)^2)$. Unfortunately, this scheme presents a strong drawback since its input consists of a single bit. First, this implies that encrypting k bits leads to a cost of $\mathcal{O}(k\cdot\ell(p)^2)$. This is not very efficient even if it is considered as practical. The second consequence concerns the expansion: a single bit of plaintext is encrypted in an integer modulo n, that is, $\ell(n)$ bits. Thus, the expansion is really huge. This is the main drawback of this scheme.

Before continuing our review, let us present the Goldwasser-Micali (GM) scheme from another point of view. This is required to understand how it has been generalized. The basic principle of GM is to partition a well-chosen subset of integers modulo n into two secret parts: M_0 and M_1. Then, encryption selects a random element of M_b to encrypt b, and decryption allows to know in which part the randomly selected element lies. The core point lies in the way to choose the subset, and to partition it into M_0 and M_1. GM uses group theory to achieve the following: the subset is the group G of invertible integers modulo n with a Jacobi symbol, with respect to n, equal to 1. The partition is generated by another group $H \subset G$, composed of the elements that are invertible modulo n with a Jacobi symbol, with respect to a fixed factor of n, equal to 1; with these settings, it is possible to split G into two parts: H and $G \setminus H$.

The generalizations of Goldwasser-Micali play with these two groups; they try to find two groups G and H such that G can be split into more than $k = 2$ parts.

(ii) Benaloh [50] is a generalization of GM, that enables to manage inputs of $\ell(k)$ bits, k being a prime satisfying some particular constraints. Encryption is similar as in the previous scheme (encrypting a message $m \in \{0, \ldots, k-1\}$ means picking an integer $r \in \mathbf{Z}_n^*$ and computing $c = g^m r^k \bmod n$) but decryption is more complex. The input and output sizes being, respectively, of $\ell(k)$ and $\ell(n)$ bits, the expansion is equal to $\ell(n)/\ell(k)$. This is better than in the GM case. Moreover, the encryption cost is not too high. Nevertheless, the decryption cost is estimated to be $\mathcal{O}(\sqrt{k}\ell(k))$ for precomputation, and the same for each dynamical decryption. This implies that k has to be taken quite small, which limits the gain obtained on the expansion.

(iii) Naccache-Stern [51] is an improvement of Benaloh's scheme. Considering a parameter k that can be greater than before, it leads to a smaller expansion. Note that the constraints on k are slightly different. The encryption step is precisely the same as in Benaloh's scheme, but the decryption is different. To summarize, the expansion is still equal to $\ell(n)/\ell(k)$, but the decryption cost is lower: $\mathcal{O}(\ell(n)^5 \log(\ell(n)))$, and the authors claim it is reasonable to choose the parameters as to get an expansion equal to 4.

(iv) In order to improve previous schemes, Okamoto and Uchiyama decided to change the base group G [52]. Considering $n = p^2q$, p and q still being two large primes, and the group $G = \mathbf{Z}_{p^2}^*$, they achieve $k = p$. Thus, the expansion is equal to 3. As Paillier's scheme is an improvement of this one and will be fully described below, we will not discuss its description in detail. Its advantage lies in the proof that its security is equivalent to the factorization of n. Unfortunately, a chosen-ciphertext attack has been proposed leading to this factorization. This scheme was used to design the EPOC systems [53], currently submitted for the supplement P1363a to the IEEE Standard Specifications for Public-Key Cryptography (IEEE P1363). Note that earlier versions of EPOC were subject to security flaws as pointed out in [54], due to a bad use of the scheme.

(v) One of the most well-known homomorphic encryption schemes is due to Paillier [47], and is described in Figure 5. It is an improvement of the previous one, that decreases the expansion from 3 to 2. Paillier came back to $n = pq$, with $\gcd(n, \phi(n)) = 1$, but considered the group $G = \mathbf{Z}_{n^2}^*$, and a proper choice of H led him to $k = \ell(n)$. The encryption cost is not too high. Decryption needs one exponentiation modulo n^2 to the power $\lambda(n)$, and a multiplication modulo n. Paillier showed in his paper how to manage decryption efficiently through the Chinese Remainder Theorem. With smaller expansion and lower cost compared with the previous ones, this scheme is really attractive. In 2002, Cramer and Shoup proposed a general approach to gain security against adaptive chosen-ciphertext attacks for certain cryptosystems with some particular algebraic properties [55]. Applying it to Paillier's original scheme, they proposed a stronger variant. Bresson et al. proposed in [56] a slightly different version that may be more accurate for some applications.

(vi) Damgård and Jurik proposed in [57] a generalization of Paillier's scheme to groups of the form $\mathbf{Z}_{n^{s+1}}^*$ with $s > 0$. The larger the s is, the smaller the expansion is. Moreover, this scheme leads to a lot of applications. For example, we can mention the adaptation of the size of the plaintexts, the use of threshold cryptography, electronic voting, and so forth. To encrypt a message $m \in \mathbf{Z}_n$, one picks $r \in \mathbf{Z}_n^*$ at random and computes $g^m r^{n^s} \in \mathbf{Z}_{n^{s+1}}$. The authors show that if one can break the scheme for a given value $s = \sigma$, then one can break it for $s = \sigma - 1$. They also show that the semantic security of this scheme is equivalent to that of Paillier. To summarize, the expansion is of $1 + 1/s$, and hence can be close to 1 if s is sufficiently large. The ratio of the encryption cost of this scheme over Paillier's can be estimated to be $(1/6)s(s+1)(s+2)$. The same ratio for the decryption step equals $(1/6)(s+1)(s+2)$. Note that even if this scheme is better than Paillier's according to its lower expansion, it remains more costly. Moreover, if we want to encrypt or decrypt k blocks of $\ell(n)$ bits, running Paillier's scheme k times is less costly than running Damgård-Jurik's scheme once.

(vii) Galbraith proposed in [58] an adaptation of the previous scheme in the context of elliptic curves. Its expansion is equal to 3. The ratio of the encryption (resp., decryption) cost of this scheme in the case $s = 1$ over Paillier's can be estimated to be about 7 (resp., 14). But, in contrast to the previous scheme, the larger the s is, the more the cost may decrease. Moreover, as in the case of Damgård-Jurik's scheme, the higher the s is, the stronger the scheme is.

(viii) Castagnos explored in [59, 60][10] another improvement direction considering quadratic fields quotients. We have the same kind of structure regarding n^{s+1} as before, but in another context. To summarize, the expansion is 3 and the ratio of the encryption/decryption cost of this scheme in the case $s = 1$ over Paillier's can be estimated to be about 2 (plus 2 computations of Legendre symbols for the decryption step).

(x) To close the survey of this family of schemes, let us mention the ElGamal-Paillier amalgam, which merges Paillier and the additively homomorphic variant of ElGamal. More precisely, it is based on Damgård-Jurik's (presented above) and Cramer-Shoup's [55] analyses and variants of Paillier's scheme, and was proposed by [9]. The goal was to gain the advantages of both schemes while minimizing their drawbacks. Preserving the notation of both ElGamal and Paillier schemes, we will describe the encryption in the particular case $s = 1$, which leads Damgård-Jurik's variant to the original Paillier. To encrypt a message $m \in \mathbf{Z}_n$, Bob picks at random an integer k, and computes $(c_1, c_2) = (g^k \bmod n, (1+n)^m (y_A^k \bmod n)^n \bmod n^2)$.

Now that we have reviewed the two most famous families of homomorphic encryption schemes, we would like to mention a few research directions and challenges.

First, as we mentioned in Section 2.1, it is important to have different kinds of schemes, because of applications and security purposes. One direction to design homomorphic schemes that are not directly related to the same mathematical problems as ElGamal or Paillier (and variants) is to consider the recent papers dealing with Weil pairing. As this new direction is more and more promising in the design of asymmetric schemes, the investigation in the particular case of homomorphic ciphers is of interest. ElGamal may not be directly used in the Weil pairing setup as the mathematical problem it is based on becomes easy to manage. One more promising direction is the use of the pairing-based scheme proposed by Boneh and Franklin [61] to obtain a secure homomorphic ID-based scheme (see directions in [62] for the ability of such schemes to provide interesting new features).

A second interesting research direction lies in the area of symmetric encryption. As all the homomorphic encryption schemes we mentioned so far are asymmetric, they are not as fast as symmetric ones could be. But, homomorphy is easier to manage when mathematical operators are involved in the encryption process, which is not usually the case in symmetric schemes. Very few symmetric homomorphic schemes have been proposed, most of them being broken ([63] broken in [64, 65], [66] broken in [67]). Nevertheless, it may

[10] This scheme is mentioned in the conclusion of [59], and more deeply presented in [60], unfortunately in French.

be of interest to consider a simple generalization of the one-time pad, where bits are replaced by integers modulo n, as introduced by [68]. In terms of security, it has exactly the same properties than the one-time pad, that is, perfect secrecy if and only if the keystream is truly random, of same length as the plaintext, and is used only once. Here again, this is overwhelming and the keystream could be generated by a well-chosen pseudorandom generator (e.g., as Snow 2.0), decreasing security from unconditional to computational. Note that this scheme's homomorphy is a little bit fuzzy, as we have for any pair of encryption keys (k_1, k_2)

$$\forall m_1, m_2 \in \mathcal{M}, \quad E_{k_1+k_2}(m_1 + m_2) \longleftarrow E_{k_1}(m_1) + E_{k_2}(m_2). \tag{3}$$

This is the only example of a symmetric homomorphic encryption that has not been cracked.

As per algebraic homomorphy, designing algebraically homomorphic encryption schemes is a real challenge today. There has been only a few ones proposed: by Fellows and Koblitz [69] (which cannot be considered as secure nor efficient [70]), by Domingo-Ferrer [63, 66] (which has been broken [64, 65, 67]), and construction studies of Rappe et al. [3]. No satisfactory solution has been proposed so far, and, as Boneh and Lipton conjectured that any algebraically homomorphic encryption would prove to be insecure [45], the question of their existence and design is still open.

4. CONCLUSION

We presented in this paper a state of the art on homomorphic encryption schemes discussing their parameters, performances and security issues. As we saw, these schemes are not well suited for every use, and their characteristics must be taken into account. Nowadays, such schemes are studied in wide application contexts, but the research is still challenging in the cryptographic community to design more powerful/secure schemes. Their use in the signal processing community is quite new, and we hope this paper will serve as a guide for understanding their specificities, advantages and limits.

ACKNOWLEDGMENTS

The authors are indebted to the referees for their fruitful comments concerning this manuscript, and to Fabien Laguillaumie and Guilhem Castagnos for discussions about the recent improvements in the field. They also thank all the people who took the time to read this manuscript and share their thoughts about it. Dr. C. Fontaine is supported (in part) by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

REFERENCES

[1] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," in *Foundations of Secure Computation*, pp. 169–177, Academic Press, 1978.

[2] E. Brickell and Y. Yacobi, "On privacy homomorphisms," in *Advances in Cryptology (EUROCRYPT '87)*, vol. 304 of *Lecture Notes in Computer Science*, pp. 117–126, Springer, New York, NY, USA, 1987.

[3] D. Rappe, *Homomorphic cryptosystems and their applications*, Ph.D. thesis, University of Dortmund, Dortmund, Germany, 2004, http://www.rappe.de/doerte/Diss.pdf.

[4] R. Cramer and I. Damgård, "Zero-knowledge for finite field arthmetic, or: can zeroknowledge be for free?" in *Advances in Cryptology (CRYPTO '98)*, vol. 1462 of *Lecture Notes in Computer Science*, pp. 424–441, Springer, New York, NY, USA, 1998.

[5] H. Lipmaa, "Verifiable homomorphic oblivious transfer and private equality test," in *Advances in Cryptology (ASIACRYPT '03)*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 416–433, Springer, New York, NY, USA, 2003.

[6] P.-A. Fouque, G. Poupard, and J. Stern, "Sharing decryption in the context of voting or lotteries," in *Proceedings of the 4th International Conference on Financial Cryptography*, vol. 1962 of *Lecture Notes in Computer Science*, pp. 90–104, Anguilla, British West Indies, 2000.

[7] T. Sander and C. Tschudin, "Protecting mobile agents against malicious hosts," in *Mobile Agents and Security*, vol. 1419 of *Lecture Notes in Computer Science*, pp. 44–60, Springer, New York, NY, USA, 1998.

[8] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal re-encryption for mixnets," in *Proceedings of the RSA Conference Cryptographers (Track '04)*, vol. 2964 of *Lecture Notes in Computer Science*, pp. 163–178, San Francisco, Calif, USA, 2004.

[9] I. Damgård and M. Jurik, "A length-flexible threshold cryptosystem with applications," in *Proceedings of the 8th Australian Conference on Information Security and Privacy (ACISP '03)*, vol. 2727 of *Lecture Notes in Computer Science*, Wollongong, Australia, 2003.

[10] A. Adelsbach, S. Katzenbeisser, and A. Sadeghi, "Cryptology meets watermarking: detecting watermarks with minimal or zero-knowledge disclosures," in *Proceedings of the European Signal Processing Conference (EUSIPCO '02)*, Toulouse, France, September 2002.

[11] B. Pfitzmann and W. Waidner, "Anonymous fingerprinting," in *Advances in Cryptology (EUROCRYPT '97)*, vol. 1233 of *Lecture Notes in Computer Science*, pp. 88–102, Springer, New York, NY, USA, 1997.

[12] N. Memon and P. Wong, "A buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 643–649, 2001.

[13] C.-L. Lei, P.-L. Yu, P.-L. Tsai, and M.-H. Chan, "An efficient and anonymous buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 13, no. 12, pp. 1618–1626, 2004.

[14] M. Kuribayashi and H. Tanaka, "Fingerprinting protocol for images based on aditive homomorphic property," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2129–2139, 2005.

[15] V. Shoup, *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press, 2005, http://www.shoup.net/ntb/.

[16] A. Menezes, P. Van Orschot, and S. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997, http://www.cacr.math.uwaterloo.ca/hac/.

[17] H. Van Tilborg, Ed., *Encyclopedia of Cryptography and Security*, Springer, New York, NY, USA, 2005.

[18] A. Kerckhoffs, "La cryptographie militaire (part i)," *Journal des Sciences Militaires*, vol. 9, no. 1, pp. 5–38, 1883.

[19] A. Kerckhoffs, "La cryptographie militaire (part ii)," *Journal des Sciences Militaires*, vol. 9, no. 2, pp. 161–191, 1883.

[20] J. Daemen and V. Rijmen, "The block cipher RIJNDAEL," in *(CARDIS '98)*, vol. 1820 of *Lecture Notes in Computer Science*, pp. 247–256, Springer, New York, NY, USA, 2000.

[21] J. Daemen and V. Rijmen, "The design of Rijndael," in *AES—the Advanced Encryption Standard*, Informtion Security and Cryptography, Springer, New York, NY, USA, 2002.

[22] G. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *Journal of the American Institute of Electrical Engineers*, vol. 45, pp. 109–115, 1926.

[23] P. Ekdahl and T. Johansson, "A new version of the stream cipher SNOW," in *Selected Areas in Cryptography (SAC '02)*, vol. 2595 of *Lecture Notes in Computer Science*, pp. 47–61, Springer, New York, NY, USA, 2002.

[24] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[25] T. ElGamal, "A prublic key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology (CRYPTO '84)*, vol. 196 of *Lecture Notes in Computer Science*, pp. 10–18, Springer, New York, NY, USA, 1985.

[26] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[27] M. Ajtai and C. Dwork, "A public key cryptosystem with worst-case/average-case equivalence," in *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC '97)*, pp. 284–293, 1997.

[28] P. Nguyen and J. Stern, "Cryptanalysis of the Ajtai-Dwork cryptosystem," in *Advances in Cryptology (CRYPTO '98)*, vol. 1462 of *Lecture Notes in Computer Science*, pp. 223–242, Springer, New York, NY, USA, 1999.

[29] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle model, revisited," in *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC '98)*, pp. 209–218, Berkeley, Calif, USA, 1998.

[30] P. Paillier, "Impossibility proofs for RSA signatures in the standard model," in *Proceedings of the RSA Conference 2007, Cryptographers' (Track)*, vol. 4377 of *Lecture Notes in Computer Science*, pp. 31–48, San Fancisco, Calif, USA, 2007.

[31] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[32] D. Kahn, *The Codebreakers: The Story of Secret Writing*, Macmillan, New York, NY, USA, 1967.

[33] M. Bellare and P. Rogaway, "Optimal asymmetric encryption—how to encrypt with RSA," in *Advances in Cryptology (EUROCRYPT '94)*, vol. 950 of *Lecture Notes in Computer Science*, pp. 92–111, Springer, New York, NY, USA, 1995.

[34] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the 14th ACM Symposium on the Theory of Computing (STOC '82)*, pp. 365–377, New York, NY, USA, 1982.

[35] M. Blum and S. Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information," in *Advances in Cryptology (EUROCRYPT '84)*, vol. 196 of *Lecture Notes in Computer Science*, pp. 289–299, Springer, New York, NY, USA, 1985.

[36] O. Goldreich, "A uniform complexity treatment of encryption and zero-knowledge," *Journal of Cryptology*, vol. 6, no. 1, pp. 21–53, 1993.

[37] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," in *Proceedings of the 22nd ACM Annual Symposium on the Theory of Computing (STOC '90)*, pp. 427–437, Baltimore, Md, USA, 1990.

[38] C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Advances in Cryptology (CRYPTO '91)*, vol. 576 of *Lecture Notes in Computer Science*, pp. 433–444, Springer, New York, NY, USA, 1991.

[39] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," in *Proceedings of the 23rd ACM Annual Symposium on the Theory of Computing —(STOC '91)*, pp. 542–552, 1991.

[40] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," *SIAM Journal of Computing*, vol. 30, no. 2, pp. 391–437, 2000.

[41] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Advances in Cryptology (CRYPTO '98)*, vol. 1462 of *Lecture Notes in Computer Science*, pp. 26–45, Springer, New York, NY, USA, 1998.

[42] M. Bellare and A. Sahai, "Non-malleable encryption: equivalence between two notions, and an indistinguishability-based characterization," in *Advances in Cryptology (CRYPTO '99)*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 519–536, Springer, New York, NY, USA, 1999.

[43] Y. Watanabe, J. Shikata, and H. Imai, "Equivalence between semantic security and indistinguishability against chosen ciphertext attacks," in *Public Key Cryptography (PKC '03)*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 71–84, Springer, New York, NY, USA, 2003.

[44] N. Ahituv, Y. Lapid, and S. Neumann, "Processing encrypted data," *Communications of the ACM*, vol. 30, no. 9, pp. 777–780, 1987.

[45] D. Boneh and R. Lipton, "Algorithms for black box fields and their application to cryptography," in *Advances in Cryptology (CRYPTO '96)*, vol. 1109 of *Lecture Notes in Computer Science*, pp. 283–297, Springer, New York, NY, USA, 1996.

[46] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[47] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology (EUROCRYPT '99)*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 223–238, Springer, New York, NY, USA, 1999.

[48] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multiauthority election scheme," in *Advances in Cryptology (EUROCRYPT '97)*, vol. 1233 of *Lecture Notes in Computer Science*, pp. 103–118, Springer, New York, NY, USA, 1997.

[49] R. McEliece, "A public-key cryptosystem based on algebraic coding theory," Dsn progress report, Jet Propulsion Laboratory, 1978.

[50] J. Benaloh, *Verifiable secret-ballot elections*, Ph.D. thesis, Yale University, Department of Computer Science, New Haven, Conn, USA, 1988.

[51] D. Naccache and J. Stern, "A new public-key cryptosystem based on higher residues," in *Proceedings of the 5th ACM Conference on Computer and Communications Security*, pp. 59–66, San Francisco, Calif, USA, November 1998.

[52] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Advances in Cryptology (EUROCRYPT '98)*, vol. 1403 of *Lecture Notes in Computer Science*, pp. 308–318, Springer, New York, NY, USA, 1998.

[53] T. Okamoto, S. Uchiyama, and E. Fujisaki, "Epoc: efficient probabilistic publickey encryption," Tech. Rep., 2000, Proposal to IEEE P1363a, http://grouper.ieee.org/groups/1363/P1363a/draft.html.

[54] M. Joye, J.-J. Quisquater, and M. Yung, "On the power of misbehaving adversaries and security analysis of the original EPOC," in *Topics in Cryptology CT-RSA 2001*, vol. 2020 of *Lecture Notes in Computer Science*, Springer, New York, NY, USA, 2001.

[55] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Advances in Cryptology (EUROCRYPT '02)*, vol. 2332 of *Lecture Notes in Computer Science*, pp. 45–64, Springer, New York, NY, USA, 2002.

[56] E. Bresson, D. Catalano, and D. Pointcheval, "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications," in *Advances in Cryptology (ASIACRYPT '03)*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 37–54, Springer, New York, NY, USA, 2003.

[57] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of Pailliers probabilistic public-key system," in *4th International Workshop on Practice and Theory in Public-Key Cryptography*, vol. 1992 of *Lecture Notes in Computer Science*, pp. 119–136, Springer, New York, NY, USA, 2001.

[58] S. Galbraith, "Elliptic curve paillier schemes," *Journal of Cryptology*, vol. 15, no. 2, pp. 129–138, 2002.

[59] G. Castagnos, "An efficient probabilistic public-key cryptosystem over quadratic fields quotients," 2007, Finite Fields and Their Applications, paper version in press, http://www.unilim.fr/pages perso/guilhem.castagnos/.

[60] G. Castagnos, *Quelques schémas de cryptographie asymétrique probabiliste*, Ph.D. thesis, Université de Limoges, 2006, http://www.unilim.fr/pages perso/guilhem.castagnos/.

[61] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology (CRYPTO '01)*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, Springer, New York, NY, USA, 2001.

[62] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Advances in Cryptology (EUROCRYPT '05)*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 440–456, Springer, New York, NY, USA, 2005.

[63] J. Domingo-Ferrer, "A provably secure additive and multiplicative privacy homomorphism," in *Proceedings of the 5th International Conference on Information Security (ISC '02)*, vol. 2433 of *Lecture Notes in Computer Science*, pp. 471–483, Sao Paulo, Brazil, 2002.

[64] D. Wagner, "Cryptanalysis of an algebraic privacy homomorphism," in *Proceedings of the 6th International Conference on Information Security (ISC '03)*, vol. 2851 of *Lecture Notes in Computer Science*, Bristol, UK, 2003.

[65] F. Bao, "Cryptanalysis of a provable secure additive and multiplicative privacy homomorphism," in *International Workshop on Coding and Cryptograhy (WCC '03)*, pp. 43–49, Versailles, France, 2003.

[66] J. Domingo-Ferrer, "A new privacy homomorphism and applications," *Information Processing Letters*, vol. 60, no. 5, pp. 277–282, 1996.

[67] J. Cheon, W.-H. Kim, and H. Nam, "Known-plaintext cryptanalysis of the domingo-ferrer algebraic privacy homomorphism scheme," *Information Processing Letters*, vol. 97, no. 3, pp. 118–123, 2006.

[68] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *ACM/IEEE Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous '05)*, pp. 109–117, 2005.

[69] M. Fellows and N. Koblitz, "Combinatorial cryptosystems galore!," in *Contemporary Mathematics*, vol. 168 of *Finite Fields: Theory, Applications, and Algorithms, FQ2*, pp. 51–61, 1993.

[70] L. Ly, *A public-key cryptosystem based on Polly Cracker*, Ph.D. thesis, Ruhr-Universität Bochum, Bochum, Germany, 2002.