Journal of Universal Computer Science, vol. 15, no. 3 (2009), 506-522
submitted: 17/11/08, accepted: 28/1/09, appeared: 1/2/09 © J.UCS

Advances in Homomorphic Cryptosystems

Mufutau Akinwande
(Lagos State University, Lagos, Nigeria
mboakinwande@yahoo.com)

Abstract: During the last few years homomorphic encryption techniques have been studied extensively since they have become more and more important in many different cryptographic protocols such as voting protocols, lottery protocols, anonymity, privacy, and electronic auctions.

This paper critically summarizes the current state-of-art of homomorphic cryptosystems. It recalls the basic ideas, discusses their parameters, performances and security issues. And, finally we present their capabilities in the future applications.

Keywords: Probability encryption, Homomorphic encryption, Cryptosystems
Categories: E.3

1 Introduction

The demand for privacy of digital data and of more complex structures like algorithms has become stronger during the last few years. This goes hand in hand with growth of communication networks like the Internet and a vastly growing number of electronic devices. On the one hand these devices enable a great variety of attacks on digital goods and on the other hand they are vulnerable to attacks such as the manipulation or destruction of data and the theft of sensitive information. For storing and reading data securely there exist several possibilities to guarantee privacy such as data encryption and tamper resistant hardware [Idowu et al, 05]. The problem becomes more complex when asking for the possibility to compute (publicly) with private data or to modify functions or algorithms in such a way that they are still executable while their privacy is ensured. This is where homomorphic cryptosystems can be used since they enable computations with encrypted data.

[Rivest et al, 78] were the first to solve this issue through *homomorphic encryption*. Unfortunately, [Brickell and Yacobi, 87] pointed out in 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 such as secret sharing schemes, threshold schemes, zero-knowledge proofs, oblivious transfer, commitment schemes, anonymity, privacy, electronic voting, electronic auctions, lottery protocols, protection of mobile agents, multiparty computation, mix-nets, watermarking or fingerprinting protocols [Rappe, 04].

Furthermore, the question rose again in 1991 when [Feigenbaum and Merritt, 91] asked: “Is there an encryption function E() such that both E(x+y) and E(xy) are easy to compute from E(x) and E(y)?” They were asking explicitly for so called algebraically homomorphic encryption techniques. Unfortunately, there has been little progress made in determining whether such encryption techniques exist that are efficient and secure, although it is one of the crucial open problems in cryptography.

We would examine and illuminate homomorphic cryptosystems in three steps ("what", "how", "why") that reflect the guidelines about the main characteristics of encryption primitives: algorithms, performance, and security.

2 Basic concepts

2.1 Homomorphic encryption

We will present in this section the basic definitions related to *homomorphic* encryption while the current trend will be given in [Section 3].

2.1.1 Definition

Let *M* (or *C*) denote the set of the plaintexts (or ciphertexts, respectively). An encryption scheme is said to be *homomorphic* if for any given encryption key *k* the encryption function *E* satisfies

$$\forall\, m_1, m_2 \in M, \qquad E(m_1 \circ_M m_2) = E(m_1) \circ_C E(m_2)$$

for some operators $\circ_M$ in *M* and $\circ_C$ in *C*, where $=$ means "can be directly computed from," that is, without any intermediate decryption [Fontaine and Galand, 07].

Informally speaking, homomorphic cryptosystem is a cryptosystem with the additional property that there exists an efficient algorithm to compute an encryption of the sum or the product, of two messages given the public key and the encryptions of the messages but not the messages themselves.

If *M* (or *C*) is an additive (semi-) group then the scheme is called *additively homomorphic* and the algorithm is called Add. Otherwise the scheme is called *multiplicatively homomorphic* and the algorithm is called Mult.

2.1.2 Remarks

- Note that for a homomorphic encryption scheme to be efficient it is crucial to make sure that the size of the ciphertexts remains polynomially bounded in the security parameter during repeated computations.
- The security aspects, definitions, and models of homomorphic cryptosystems are the same as for other cryptosystems.

If the encryption algorithm *E* gets as additional input a uniform random number r of a set Z, the encryption scheme is called *probabilistic* otherwise it is called *deterministic*. Hence if a cryptosystem is probabilistic there belong several different ciphertexts to one message depending on the random number $r \in Z$. But note that as before the decryption algorithm remains deterministic, i.e. there is just one message belonging to a given ciphertext as illustrated in the example (2.1.3). Furthermore, in a probabilistic, homomorphic cryptosystems the encryption algorithm should be probabilistic too to hide the input ciphertexts.

2.1.3 Example

We now give an example of a deterministic, multiplicatively homomorphic scheme and an example for a probabilistic, additively homomorphic scheme.

1. The classical RSA scheme [Rivest et al, 78] is an example of a deterministic, multiplicatively homomorphic cryptosystem on $M = (Z/NZ, \times)$ where N is the product of two large primes. As ciphertext space we have $C = (Z/NZ, \times)$ and as key space we have $K = \{(k_e, k_d) = ((N,e),d) | N = pq,\ ed \equiv 1 \bmod \varphi(N)\}$. The encryption of a message $m \in M$ is defined as $E_{k_e}(m) := m^e \bmod N$ ' and for decryption of a ciphertext $E_{k_e}(m) := c \in C$ we compute $D_{k_e,k_d}(c) := c^d \bmod N = m \bmod N$. Obviously, the encryption of the product of the two messages can be efficiently computed by multiplying the corresponding ciphertexts, i.e.,

$$E_{k_e}(m_1 \times m_2) = (m_1 m_2)^e \bmod N = (m_1{}^e \bmod N)(m_2{}^e \bmod N)$$
$$= E_{k_e}(m_1) \times E_{k_e}(m_2)$$

where $m_1, m_2 \in M$. Hence, the algorithm Mult can easily be implemented as $\mathrm{Mult}(E_{k_e}(m_1), E_{k_e}(m_2)) := E_{k_e}(m_1) \times E_{k_e}(m_2)$.

Usually in the RSA scheme as well as in most schemes based on the difficulty of factoring the security parameter is the bit length of N. For instance 1024 is a common security parameter.

2. The Goldwasser-Micali scheme, proposed in [Goldwasser and Micali, 84] is an example of a probabilistic, additively homomorphic cryptosystem on $M = (Z/2Z, +)$ with $C = Z = (Z/NZ)^*$ where $N = pq$ is the product of two large primes.

We have

$$K = \left\{ (k_e, k_d) = ((N,a),(p,q)) \middle| N = pq, a \in (Z/NZ)^* : \left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1 \right\}.$$

since this scheme is probabilistic, the encryption algorithm gets as additional input a random value $r \in Z$. We define $E_{k_e}(m,r) := a^m r^2 \bmod N$ and

$$D_{(k_e,k_d)}(c) := \begin{cases} 0, & \text{if } c \text{ is a square} \\ 1, & \text{otherwise} \end{cases}$$

It holds that

$$E_{k_e}(m_1, r_1) \times E_{k_e}(m_2, r_2) = E_{k_e}(m_1 + m_2, r_1 r_2).$$

Thus the algorithm Add can be efficiently implemented e.g. as

$$\begin{aligned}\mathrm{Add}(E_{k_e}(m_1, r_1), E_{k_e}(m_2, r_2), r_3) &= E_{k_e}(m_1, r_1) \times E_{k_e}(m_2, r_2) \times r_3^2 \bmod N \\ &= E_{k_e}(m_1, r_1) \times E_{k_e}(m_2, r_2) \times E_{k_e}(0, r_3) \\ &= E_{k_e}(m_1 + m_2, r_1 r_2 r_3)\end{aligned}$$

where $m_1, m_2 \in M$ and $r_1, r_2, r_3 \in Z$.

Note that as already mentioned this algorithm should be probabilistic, i.e., it obtains a random number r_3 as additional input.

A public-key homomorphic encryption scheme on a (semi-) ring $(M, +, \times)$ can be defined in an analogous way. Such schemes consist of two algorithms Add and Mult for the homomorphic property instead of one algorithm, i.e., it is additively and multiplicatively homomorphic at the same time. Such schemes are called *algebraically homomorphic.*

A lot of such homomorphic schemes have been published that have been widely used in many applications. 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 distributivity between encryption and some data processing operations.

The schemes we will consider in the next section have to be probabilistic ciphers, and we may consider *E* to behave in a probabilistic way.

2.2 Security Issues

A minimal requirement for an encryption scheme is that it must be impossible to retrieve an encrypted plaintext for anybody not knowing the decryption key. However, this condition may be too weak - in some applications even partial information gained from the plaintext could endanger security. This is why we demand it to be “infeasible to learn *anything* about the plaintext from the ciphertext” or, in other words, “whatever an eavesdropper can compute about the cleartext given the ciphertext, he can also compute without the ciphertext” [Goldwasser and Micali, 84].

Probabilistic encryption was introduced with a clear purpose: security. This requires to properly define different security levels. *Semantic security* was introduced in [Goldwasser and Micali, 82], 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: if no passive adversary can, in expected polynomial time, select two plaintext messages m_1 and m_2 and then correctly distinguish between encryptions of m_1 and m_2 with probability significantly greater than ½, thus 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, 93].

Quite amazingly, [Goldwasser and Micali, 82] proved the equivalence between polynomial security and semantic security; [Goldreich, 93] extended these notions 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.

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*)). If we relying on the different contexts, 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 [Fontaine and Galand, 07].

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* [Dolev et al, 00].

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, knowing *c*, we can compute $c' = c \circ_C c$ and deduce, by the homomorphic property, that c' is a ciphertext of $m' = m \circ_M m$. According to the previous remark on adaptive chosen-ciphertext indistinguishability, a homomorphic encryption has no access to the strongest security requirement. The highest security level it can reach is IND-CPA.

We now point out some security considerations about deterministic homomorphic encryption. First, it was proved that a deterministic homomorphic encryption for which the operation $\circ$ is a simple addition is insecure [Fontaine and Galand, 07].

Second, [Boneh and Lipton, 96] showed that any deterministic algebraically homomorphic cryptosystem can be broken in subexponential time. We 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 [Idowu et al, 05].

3 Current Trends in Homomorphic Encryption

First, we 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).

Furthermore its probabilistic variants, obtained through Optimal Asymmetric Encryption Padding, 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. Comparing it with the original ElGamal, this variant also involves an element *G* (*G* may be equal to *g*) that generates $(Z_q, +)$ with respect to the addition operation. To send an encrypted version of the message *m* to Alice, Bob picks at random $k \in Z_q$ 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 proposed by [Goldwasser and Micali, 82], to the famous Paillier's encryption scheme [Paillier, 99] 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.

3.1 Goldwasser-Micali (GM) Scheme

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 $r \in \mathrm{Z}_n^*$, and computes

$$c = g^b r^2 \bmod n$$

(where 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 $(-1)^b$.
Note that the scheme encrypts 1 bit of information, while its output is usually 1024 bits long!

Security:
This scheme was the first system based upon the concept of probabilistic encryption and furthermore the first system proven to be semantically secure (assuming the intractability of the quadratic residuosity problem).
It is, nevertheless, not a practicable scheme since in general, one plaintext-bit is expanded into n bits of ciphertext.

3.2 Damgard-Jurik Scheme

We now give an encryption scheme which is probabilistic, additively and scalar homomorphic cryptosystem. It was published by Damgard and Jurik [Damgard and Jurik, 03]. The scheme works as follows:

Key generation:

- Choose an RSA modulus $N = pq = (2p'+1)(2q'+1)$ with primes p, p', q, q'.
- Select an element $g \in Q_N$ where Q_N denotes the group of all squares in $(Z/NZ)^*$. Choose $\alpha \in Z/\tau Z$, where $\tau = p'q' = |Q_N|$.
- Compute $h = g^\alpha \bmod N$.

The public key is *a: = (N, g, h)* and the secret key is α.

Encryption:
To encrypt a message *m*, choose an integer *s > 0* so that $m \in Z/N^s Z$, and choose a random $r \in Z/nZ$, where $n = 4^{\log_2 N}$. The ciphertext is

$$E_a(m, r) = (g^r \bmod N, (h^r \bmod N)^{N^s} (N+1)^m \bmod N^{s+1}) =: (G, H).$$

Let L_s denote a function with

$$L_s((N+1)^m \bmod N^{s+1}) = m \bmod N^s.$$

An algorithm that computes this function, i.e., that calculates the discrete logarithm with respect to the element *(N + 1)* is described in [Damgard and Jurik, 01].

Decryption:
Given a ciphertext $c = (G, H) = E_a(m, r), s$ can be deduced from the length of c or it is attached to the encryption. Then the message m can be found as

$$\begin{aligned} m &= L_s(H(G^{\alpha} \bmod N)^{-N^s}) \\ &= L_s((g^{\alpha r} \bmod N)^{N^s}(N+1)^m(g^{r\alpha} \bmod N)^{-N^s}) \\ &= L_s((N+1)^m \bmod N^{s+1}). \end{aligned}$$

Homomorphic Property:
This scheme is additively homomorphic since given $E_a(m_1, r_1)$ and $E_a(m_2, r_2)$ we can compute

$$\begin{aligned} E_a(m_1 + m_2, r_1 + r_2) &= (g^{r_1+r_2} \bmod N, (h^{r_1+r_2} \bmod N)^{N^s}(N+1)^{m_1+m_2} \bmod N^{s+1}) \\ &= E_a(m_1, r_1) \times E_a(m_2, r_2) \end{aligned}$$

Hence the algorithm Add can efficiently be implemented by multiplying the input ciphertexts and applying a blinding algorithm.

3.3 Paillier's Scheme

The Paillier scheme [Paillier, 99] is an example of a very efficient, probabilistic, additively and scalar homomorphic encryption scheme based on arithmetics in the ring of integers modulo N^2 where N is the product of two large primes. It was published in 1999 and analysed and extended by several authors such as [Damgard and Jurik, 01]. One of these extensions is the elliptic curve Paillier scheme (ECPS) which was recently published by Galbraith, S. [Galbraith, 02]. The ECPS is a generalization of Paillier's encryption scheme from the integers modulo a square to elliptic curves over rings.

Paillier himself tried to generalize his scheme to the elliptic curve setting by using anomalous elliptic curves over rings, but Galbraith found security flaws in this generalization [Galbraith, 02] whereas the ECPS can be proven semantically secure relative to a new defined problem. In the same way as Damgard and Jurik managed to generalize the original Paillier scheme to higher moduli to enable a wider application scope Galbraith developed a generalization of the elliptic curve Paillier scheme [Galbraith, 02].

The Paillier scheme, the new elliptic curve version by Galbraith as well as his further generalization are examples for the probabilistic, additively homomorphic cryptosystems, which are also scalar homomorphic.

The performances of the ECPS and of its generalization are by far slower than the original Paillier scheme together with the generalization of Damgard and Jurik since they operate on elliptic curves modulo large numbers. Hence, the elliptic curve version is mainly of theoretical interest.

One interesting point is that the elliptic curve version is based on a slightly different assumption than Paillier's original version. This assumption may also hold even if the original Paillier assumption were broken.

3.3.1 Elliptic Curve Paillier Scheme

We will now summarize the elliptic curve Paillier scheme and its generalization as illustrated by Galbraith. It is a natural generalization of Paillier's probabilistic, homomorphic public key cryptosystem [Paillier, 99] to elliptic curves over rings.

Key generation:
To generate a key

- Compute a modulus $N = pq$ as a product of two primes $p, q > 3$.
- Choose a random elliptic curve $E : y^2z = x^3 + axz^2 + bz^3$ over Z / NZ, i.e. $\gcd(N, 6(4a^3 + 27b^2)) = 1$.

Let $M = \left|E(F_p)\right| \cdot \left|E(F_q)\right|$ be the order of $E(Z / NZ)$. Then knowledge of M is polynomial-time equivalent to knowledge of the factorization of $N = pq$ (see [Galbraith, 02]). Furthermore, if *p, q* are known then *M* can be computed in polynomial time using the Schoof-Atkin-Elkies algorithm (see e.g. [Blake et al, 99]).

- Choose a point $Q = (x : y : z)$ with ord$(Q)|M$ in $E(Z / N^2Z)$. Since we have $|E(Z / NZ)| = MN$, this point can be found by taking a random point $Q' = (x' : y' : z') \in E(Z / N^2Z)$ and setting $Q = NQ'$.
  Let $P_1 := (N : 1 : 0) \in E(Z / N^2Z)$.

The public key consists of the modulus *N* (and hence the point P_1), the coefficients *(a, b)* of the elliptic curve, and the point *Q*. The secret key is the order *M* of the group $E(Z / NZ)$.

Observe that $mP_1 = P_m = (mN : 1 : 0)$ for $0 \leq m < N$. Since

$$(m + N)P_1 = ((m + N)N : 1 : 0) = (mN : 1 : 0) \in E(Z / N^2Z)$$

we can also define $mP_1 = P_m$ for $m \in Z / N^2Z$. This is also valid for the generalizations given in the following sections.

Encryption:
To encrypt a message $m \in Z / NZ$ choose a random integer $1 \leq r < N$ and compute the point $C = rQ + mP_1 = rQ + P_m$. The ciphertext is the point $C \in E(Z / N^2Z)$.

Decryption:
To decrypt the ciphertext *C* use the secret key *M* to compute

$$MC = r(MQ) + MP_m = P_{mM} = (mMN : 1 : 0).$$

Given the x-coordinate mMN interpreted in Z we can divide by N to obtain $mM \in Z / NZ$ and then multiply by the inverse of $M \bmod N$ to recover the message $m \in Z / NZ$. Observe that we have $Z / NZ \cong E_1(Z / N^2 Z)$.

Homomorphic Property:
This scheme is additively homomorphic since given encryptions $C_1 = r_1 Q + m_1 P_1$ of m_1 and $C_2 = r_2 Q + m_2 P_1$ of m_2 an encryption $(C_1 + C_2) = (r_1 + r_2)Q + (m_1 + m_2)P_1$ of $(m_1 + m_2)$ can be computed just by adding the ciphertexts C_1 and C_2. Hence, we can define the algorithm *Add* as

$$Add(E(m_1), E(m_2)) := E(m_1) + E(m_2) = E(m_1 + m_2)$$

or as

$$Add(E(m_1), E(m_2)) := E(m_1) + E(m_2) + r'Q = E(m_1 + m_2)$$

with $1 \leq r' < N$ in order to blind the result. Since the message set $\Omega = Z / NZ$ the cryptosystem is also a scalar homomorphic and the algorithm Mixed-Mult can be implemented using repeatedly the algorithm *Add* and some blinding algorithm [Rappe, 04].

3.3.2 Generalization of the Elliptic Curve Paillier Scheme

In a similar way as Damgard and Jurik [Damgard and Jurik, 01] have given a generalization of the original Paillier scheme to make it more interesting for applications Galbraith generalized the ECPS [Galbraith, 02]. His generalization for the elliptic curve case will be presented here. The generalization use higher powers of N and have certain advantages as can be seen in [Damgard and Jurik, 01]. So, instead of considering the ciphertext group $E(Z / N^2 Z)$ we now consider elliptic curve over $E(Z / N^{s+1} Z)$ for $s > 0$. In this process we have to take care of subtleties relating to the formal group, see [Galbraith, 02].

Key generation:
To generate a key

- Choose a modulus $N = pq$ as a product of two primes greater than 3 and choose $s > 0$. (Thus the message set will be group $Z / N^s Z$.)
- Choose a random elliptic curve $E : y^2 z = x^3 + axz^2 + bz^3$ over Z / NZ, i.e., $\gcd(N, 6(4a^3 + 27b^2)) = 1$. Let $M = |E(F_p)| \cdot |E(F_q)|$.
- Choose a point $Q = (x : y : z)$ with $\text{ord}(Q) \mid M$ in $E(Z / N^{s+1} Z)$. This point Q can be found by taking a random point $Q' = (x' : y' : z') \in E(Z / N^{s+1} Z)$ and setting $Q = N^s Q'$. Note that $| E(Z / N^{s+1} Z) | = MN^s$.

> Now, let $P_1 := (N:1:w(N)) = (N:1:N^3 + aN^7 + \cdots) \in E(Z/N^{s+1}Z)$. We take terms in the z-coordinate until the degree is greater than *s + 1*. It can be shown that P_1 has order N^s, i.e., $N^s P_1 = \mathrm{O}$.

The public key consists of *N, s,* the coefficients *(a, b)* of the elliptic curve, and the point Q. The corresponding secret key is the order M of the group $E(Z/NZ)$.

Encryption:
To encrypt a message $m \in Z/N^sZ$ choose a random integer $1 \le r < N^s$ and compute the point $C = rQ + mP_1$. The ciphertext is the point $C \in E(Z/N^{s+1}Z)$.

Decryption:
To recover the message $m \in Z/N^sZ$ compute

$$MC = r(MQ) + mMP_1 = mMP_1 := m'P_1 = (m'N + \cdots : 1 : (m'N)^3 + \cdots).$$

Then $m' \in Z/N^sZ$ can be computed iteratively and after multiplying the result by $M^{-1} \bmod N^s$ we obtain the message as $m = m'M^{-1} \bmod N^s$. We observe that owing to the fact that for *s > 2* the map φ is not a group isomorphism but induces only the group isomorphism from

$$N^j(Z/N^sZ)/N^{j+1}(Z/N^sZ) \text{ to } E_j(Z/N^sZ)/E_{j+1}(Z/N^sZ).$$

The iteration is as follows: We write $m' = \sum_i m_i' N^i$ in terms of its base-*N* representation. Let the point $m'P_1 = (x:y:z)$ be given. The x-coordinate of this point equals $\sum_i m_i' N^i \cdot N + \cdots = m_0'N + m_1'N^2 + \cdots + \cdots$. We can determine the value of m_0' as $m_0' = \frac{x}{N} \bmod N$. We can then subtract $m_0'P_1$ from $m'P_1$ to obtain a new point $(x:y:z)$. From this point we can recover $m_1' = \frac{x}{N^2} \bmod N$ and the process is iterated.

Observe that for *s = 1* we obtain the basic elliptic curve Paillier scheme.

Obviously this generalization has the same homomorphic properties as the basic scheme. This time its semantic security is based on the assumed hardness of the following assumption, which we call the *generalized elliptic curve Paillier assumption:*
Given a point $Q \in E(Z/N^{s+1}Z)$ of order dividing $|E(Z/NZ)|$ where N is the product of two large primes and given a random point $C \in E(Z/N^{s+1}Z)$ determine whether C lies in the subgroup generated by Q.

We would now analyze the important parameters and properties of the above schemes.

❖ We begin with the rather simple scheme of GM [Goldwasser and Micali, 82]. Besides some historical importance, this scheme had an important impact on later proposals. Several other schemes, which will be presented below, were obtained as generalizations of this one. 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 $\mathrm{O}(l(p)^2)$.

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 $\mathrm{O}(k \cdot l(p)^2)$. 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, $l(n)$ bits. Thus, the expansion is really huge. This is the main drawback of this scheme.

Now we present the GM scheme from another point of view in order 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 knowing 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 \ H*.

The generalizations of GM 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.

❖ [Benaloh, 88] is a generalization of GM, that enables to manage inputs of $l(k)$ bits, *k* being a prime satisfying some particular constraints. Encryption is similar as in the previous scheme (encrypting a message $m \in \{0,...,k-1\}$ means picking an integer $r \in \mathrm{Z}_n^*$ and computing $c = g^m r^k \bmod n$) but decryption is more complex. The input and output sizes being, respectively, of $l(k)$ and $l(n)$ bits, the expansion is equal to $l(n)/l(k)$. This is better than in the GM case.

Moreover, the encryption cost is not too high. Nevertheless, the decryption cost is estimated to be $\mathrm{O}(\sqrt{k}l(k))$ for precomputation, and the same for each dynamical decryption, i.e. *k* has to be taken quite small, to limits the gain obtained on the expansion.

- [Naccache and Stern, 98] 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 $l(n)/l(k)$, but the decryption cost is lower: $\mathrm{O}(l(n)^5 \log(l(n)))$, and the authors claim it is reasonable to choose the parameters as to get an expansion equal to 4.

- In order to improve previous schemes,[Okamoto and Uchiyama, 98] decided to change the base group *G*. Considering $n = p^2q$, *p* and *q* still being two large primes, and the group $G = Z^*_{p^2}$, 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 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, due to a bad use of the scheme [Okamoto et al, 00].

- One of the most well-known homomorphic encryption schemes is

due to [Paillier, 99], and is described earlier. 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, $\varphi(n)$) = 1, but considered the group $G = Z^*_{n^2}$, and a proper choice of subgroup *H* led him to $k = l(n)$.

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.

[Cramer and Shoup, 02] proposed a general approach to gain security against adaptive chosen-ciphertext attacks for certain cryptosystems with some particular algebraic properties. Applying it to Paillier's original scheme, they proposed a stronger variant. [Bresson et al., 03] proposed a slightly different version that may be more accurate for some applications.

- [Damgard and Jurik, 01] proposed a generalization of Paillier's

scheme to groups of the form $Z^*_{n^{s+1}}$ 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 $m \in Z_n$, one picks $r \in Z_n^*$ at random and computes $g^m r^{n^s} \in Z_{n^{s+1}}$.

The authors showed 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 $s(s + 1)(s + 2)/6$. The same ratio for the decryption step equals $(s + 1)(s + 2)/6$.

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 $l(n)$ bits, running Paillier's scheme k times is less costly than running Damgard-Jurik's scheme once.

❖ [Galbraith, 02] proposed an adaptation of the previous scheme in
the context of elliptic curves. Its expansion is equal to 3. The ratio of the encryption (respectively, decryption) cost of this scheme in the case $s = 1$ over Paillier's can be estimated to be about 7 (respectively, 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 Damgard-Jurik's scheme, the higher the s is, the stronger the scheme is.

❖ [Castagnos, 07] explored 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).

❖ Finally, we would mention the ElGamal-Paillier amalgam, as proposed by [Damgard and Jurik, 03], which merges Paillier and the additively homomorphic variant of ElGamal. 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 Damgard-Jurik's variant to the original Paillier. To encrypt a message $m \in Z_n$, Bob picks at random an integer k, and computes

$$(c_1, c_2) = (g^k \bmod n, (1+n)^m (y_A^k \bmod n)^n \bmod n^2).$$

4 Conclusions

We observe that Paillier scheme is always better than Damgard-Jurik because the latter is always slow and as *s* increases; it gets worse, much worse. It also has complicated discrete logarithm function.

Although, there is no surprise that RSA is the overall fastest, but Paillier scheme fastest probabilistic homomorphic scheme is faster than RSA in decryption because of finding r. Thus, since Paillier is faster with the same advantages, it is a much better choice.

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.

Since homomorphic cryptosystems present a promising new direction for research, we would like to mention a few research directions and challenges.

First, 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 [Boneh and Franklin, 01]. 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, 01] to obtain a secure homomorphic ID-based scheme.

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 [Fontaine and Galand, 07].

As per algebraic homomorphy, designing algebraically homomorphic encryption schemes is a real challenge today. No satisfactory solution has been proposed so far, and, as [Boneh and Lipton, 96] conjectured that any algebraically homomorphic encryption would prove to be insecure; the question of their existence and design is still open.

Acknowledgements

The author thanks an anonymous reviewer whose comments and suggestions helped improving the paper.

References

[Benaloh, 88] Benaloh, J.: *Verifiable secret-ballot elections,* Ph.D. thesis, Yale University, Department of Computer Science, New Haven, USA, 1988.

[Blake et al, 99] Blake, I., Seroussi, G. and Smart, N.: Elliptic Curves in Cryptography, London Mathematical Society, Lecture Note Series 265, Cambridge University Press, 1999.

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

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

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

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

[Castagnos, 07] Castagnos, G.: An efficient probabilistic public-key cryptosystem over quadratic fields quotients, *Finite Fields and Their Applications*, http://www.unilim.fr/pagesperso/guilhem.castagnos/.

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

[Damgard and Jurik, 01] Damgard, I. and Jurik, M.: A generalization, a simplification and some applications of Pailliers probabilistic public-key system, *4th International Workshop on Practice and Theory in Public-Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, 2001, pp. 119–136, Springer, New York, USA.

[Damgard and Jurik, 03] Damgard, I. and Jurik, M.: A Length-Flexible Threshold Cryptosystem with Applications, *In Proceedings of the 8th Australasian Conference on Information Security and Privacy (ACISP 2003)*, LNCS 2727, Springer, New York, USA.

[Dolev et al, 00] Dolev, D., Dwork, C. and Naor, M.: Non-malleable cryptography, *SIAM Journal of Computing*, 30(2), 2000, pp. 391 - 437.

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

[Feigenbaum and Merritt, 91] Feigenbaum, J. and Merritt, M.: Open Questions, Talk Abstracts, and Summary of Discussions, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2, 1991, pp. 1-45.

[Fontaine and Galand, 07] Fontaine, C. and Galand, F.: A Survey of Homomorphic Encryption for Nonspecialists" *EURASIP Journal on Information Security, 2007*.

[Galbraith, 02] Galbraith, S.: Elliptic curve paillier schemes, *Journal of Cryptology*, 15(2), 2002, pp. 129–138.

[Goldreich, 93] Goldreich, O.: A uniform complexity treatment of encryption and zero-knowledge," *Journal of Cryptology*, 6(1), 1993, pp. 21 - 53.

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

[Goldwasser and Micali, 84] Goldwasser, S. and Micali, S.: Probabilistic encryption, *Journal of Computer and System Sciences*, 28(2), 1984, pp. 270 - 299.

[Idowu et al, 05] Idowu, B. A., Olaitan, H. M. and Akinwande, M. B. O.: Modern Digital Cryptosystems and their Security Capabilities, *Nigerian Journal of Science*, 39, 2005, pp.137-143.

[Naccache and Stern, 98] Naccache, D. and Stern, J.: A new public-key cryptosystem based on higher residues, *Proceedings of the 5th ACM Conference on Computer and Communications Security*, 1998, pp. 59–66, San Francisco, California, USA.

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

[Okamoto et al, 00] Okamoto, T., Uchiyama, S. and Fujisaki, E.: Epoc: efficient probabilistic public key encryption, Technical Report Proposal to IEEE P1363a, http://grouper.ieee.org/groups/1363/P1363a/draft.html

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

[Rappe, 04] Rappe, D.: Homomorphic Cryptosystems and their Applications, Ph.D. thesis, University of Dortmund, Dortmund, Germany, 2004,
http://www.rappe.de/doerte/Diss.pdf.

[Rivest et al, 78] Rivest, R., Adleman, L. and Dertouzos, M.: "On data banks and privacy homomorphisms". *Foundations of Secure Computation*, 1978, pp. 169 - 177, Academic Press.