Additively Homomorphic Encryption with a Double Decryption Mechanism, Revisited

Andreas Peter[1], Max Kronberg[2], Wilke Trei[2], and Stefan Katzenbeisser[1]

[1] Security Engineering Group
Technische Universität Darmstadt and CASED, Germany
{peter,katzenbeisser}@seceng.informatik.tu-darmstadt.de
[2] Arbeitsgruppe Algebra/Geometrie
Universität Oldenburg, Germany
{m.kronberg,wilke.trei}@uni-oldenburg.de

Abstract. We revisit the notion of additively homomorphic *encryption with a double decryption mechanism* (DD-PKE), which allows for additions in the encrypted domain while having a master decryption procedure that can decrypt all properly formed ciphertexts by using a special master secret. This type of encryption is generally considered as a practical way to enforce access control in hierachical organisations where some form of malleability properties are required. Up to now, only two additively homomorphic DD-PKE schemes have been proposed: CS-Lite by Cramer and Shoup (Eurocrypt 2002), and a variant called *BCP* by Bresson, Catalano and Pointcheval (Asiacrypt 2003).

In this work, we argue that the two existing schemes only provide partial solutions for hierarchical organisations. Essentially, this is due to the fact that the master authority, being in possession of the master secret, has no control on the validity of given ciphertexts. We say that the master is unable to "detect invalid ciphertexts", which limits the employment of such schemes in practice. Therefore, we propose the first additively homomorphic DD-PKE scheme which allows the master to detect invalid ciphertexts. In fact, our scheme has the additional property that the master decryption is independent of the users' public keys. Our solution is based on elliptic curves over rings and we prove it to be semantically secure under a DDH-related assumption. Moreover, we give experimental results on the choice of elliptic curves and their effect on the efficiency of our scheme's setup.

Keywords: Public-Key Cryptography, Homomorphic Encryption, Double Decryption Mechanisms, Elliptic Curves, Factoring.

1 Introduction

We consider a concrete example taken from practice that involves a company having many employees (e.g., an insurance company) with a certain hierarchy among them, and in particular with some master authority (e.g., the head of the company) that sits at the top of this hierarchy. Most of the company's data

D. Gollmann and F.C. Freiling (Eds.): ISC 2012, LNCS 7483, pp. 242–257, 2012.

is stored on some central servers where hierachical access control is enforced by using encryption. But it happens occasionally that some employees leave the company or new people are being employed, and so every employee should get her own public and corresponding private keys. In this scenario, the company should be concerned with the following challenges:

- To avoid expensive key management, employees should be able to generate their own key pairs without getting in touch with the master authority.
- If an employee leaves the company or loses her keys (this concerns both the public and the private key), the master authority still wants to be able to recover all data. Hence, the master authority needs some master secret (independent of the employees' individual private keys) that allows to decrypt *any* data stored on the company's servers. Moreover, the master authority should be able to check whether a ciphertext has been encrypted under a given employee's public key. This is relevant, for instance, in the following case: Assume an unavailable employee (for whatever reason, maybe due to quitting) left some important data on the server, e.g., an encryption of an important decision (1 for 'yes' and 0 for 'no'). The master authority needs to know this decision, but at the same time needs to verify whether it was encrypted by the respective employee, i.e., under her public key. In fact, an encryption under the wrong employee's public key might lead the master to a wrong decision.
- Additionally, in practice there is often the requirement that the used cryptosystem has a certain malleability property or is even homomorphic.

The just described scenario is a typical application (cf. [16]) of so-called additively homomorphic *encryption schemes with a double decryption mechanism* (DD-PKE) which combine all the above properties in just one cryptosystem. Roughly speaking, such schemes have two independent, additively homomorphic decryption procedures. Now, because solutions to the described scenario are most wanted in practice, one would expect the existence of many cryptosystems of this type. But in fact, there exist only two such schemes, namely CS-Lite by Cramer and Shoup [8] and a variant called *BCP* by Bresson, Catalano and Pointcheval [7]. Looking at these two schemes in detail, one notices two major weaknesses:

1. In the BCP cryptosystem, in order for the master authority to decrypt a given ciphertext, it has to know the employee's public key under which it was created. This fact contradicts to the requirement that the company does not want to do any complex key management (and in fact simply does not see the public keys in general).
2. Furthermore, both cryptosystems have the drawback that the master authority is unable to check whether a given ciphertext was encrypted under a given public key. This also contradicts the requirements of the above scenario. We note here that the authors of [7] left such "ciphertext validity checks" of the master authority as an open question.

In this work, we propose the *first* additively homomorphic DD-PKE scheme that avoids both just mentioned drawbacks: It is *User-Independent* (i.e., the master decryption procedure is independent of the public keys of the employees/users) and it allows the master to *detect invalid ciphertexts* (i.e., given a ciphertext and a user's public key the master can check whether the ciphertext was encrypted under the given public key).

Our solution is based on elliptic curves over rings $\mathbb{Z}_{N^2}$ where $N = pq$ is some RSA-modulus, and we prove its semantic security under a Decisional Diffie-Hellman (DDH) related assumption on such curves. Finally, we discuss different possible choices of elliptic curves in the setup of our cryptosystem. Since these choices might have an effect on the security of our scheme, we also consider randomly chosen curves (which we require to have an order with at least two large prime factors). For this, one has to rely on a conjecture by Galbraith and McKee [14] about the likelyhood of hitting on such curves. Therefore, we made a substantial number of experiments to get an idea on the efficiency of our setup algorithm for randomly chosen curves. Since there are only a few experimental results on this matter in the literatue, our results might be of independent interest.

Related Work. Since the first efficient, additively homomorphic encryption scheme was proposed by Paillier [24] a lot of follow-up papers appeared in this area (see [10] for a survey). In particular, there were many approaches to construct such schemes by using elliptic curves (see Galbraith's elliptic-curve-based Paillier scheme [13] and the references therein). Another important paper in this context is by Armknecht, Katzenbeisser and Peter [2] who give an easy to use abstract framework and security characterization of such schemes. While we are only interested in additively homomorphic encryption (i.e., it is possible to evaluate the addition of plaintexts over their encryptions without knowledge of the private key), much attention is recently being devoted to the topic of fully homomorphic encryption [17,6], which allows for the evaluation of any circuit over encrypted data without being able to decrypt.

Besides the great many of works on homomorphic encryption, there are several constructions of (non-homomorphic) DD-PKE schemes [16,27]. In this regard, we note that although identity-based encryption [5,25] is related to DD-PKE, therein the master secret is essential in order to generate the users' private keys (in DD-PKE only some publicly known master information is needed, so there is no interaction between the users and the master).

Finally, we mention the only two existing schemes [8] and [7] which are *both* additively homomorphic *and* have a double decryption mechanism.

2 Preliminaries

Notation. We write $x \longleftarrow X$ if X is a random variable or distribution and x is to be chosen randomly from X according to its distribution. In the case where X is solely a set, $x \overset{U}{\longleftarrow} X$ denotes that x is chosen uniformly at random from X. For an algorithm $\mathcal{A}$ we write $x \longleftarrow \mathcal{A}(y)$ if $\mathcal{A}$ outputs x on fixed input y according to

$\mathcal{A}$'s distribution. Sometimes, we need to specify the randomness of a probabilistic algorithm $\mathcal{A}$ explicitly. To this end, we interpret $\mathcal{A}$ as a deterministic algorithm $\mathcal{A}(y, r)$, which has access to random values r.

By a *description* of a finite set X we mean an efficient sampling algorithm (according to some distribution) for the set X. If X is a group, a *description* of X additionally includes the neutral element and a set of efficient algorithms that allow us to perform the usual group operation on X and the inversion of group elements. We abuse notation and write X both for the description and for the set itself. If a description of X is given, we denote sampling from X according to the distribution given by the sampling algorithm of the description by $x \longleftarrow X$.

If $f : X \to Y$ is a mapping between two sets X and Y, we write $\mathrm{dom}(f) = X$ for the *domain* of f and $\mathrm{im}(f)$ for its *image*. In addition, we write $f|_S$ for the *restriction* of f to a subset $S \subseteq X$, i.e. $f|_S : S \to Y$ with $f|_S(s) := f(s)$ for all $s \in S$. If X and Y are groups (additively written), and f is a group homomorphism, we write $\ker(f) := \{x \in X \mid f(x) = 0\}$ for the *kernel* of f. If f is surjective, we write $f^{-1}(y) := \{x \in X \mid f(x) = y\}$ for the *preimage* of y under f for $y \in Y$. Surjective group homomorphisms are also called *group epimorphisms.*

We recall that a public-key encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of a probabilistic polynomial time (PPT) key generation algorithm KeyGen which generates a pair $(\mathsf{pk}, \mathsf{sk})$ of corresponding public and private keys for a given security parameter κ, a PPT encryption algorithm Enc and a deterministic PT decryption algorithm Dec with the usual correctness condition. We denote the set of plaintexts by $\mathcal{P}$, the set of ciphertexts by $\widehat{\mathcal{C}}$, and the *set of all encryptions* (i.e., outputs of the encryption algorithm) by $\mathcal{C}$.

Elliptic Curves over Rings. In this section, we want to recall some facts about elliptic curves over rings. To this end, let R be a commutative unital ring with R^* denoting its group of units. We say that for $a, b \in R$ the equation

$$E : y^2 z = x^3 + axz^2 + bz^3 \tag{1}$$

defines an *elliptic curve E over R* if the *discriminant* $\Delta := 16(4a^3 + 27b^2)$ is a unit in R, i.e., $\Delta \in R^*$. For all triples $(x, y, z) \in R^3$ that satisfy (1), we say that (x, y, z) is *equivalent to* (x', y', z') if there exists $\nu \in R^*$ such that $\nu x = x', \nu y = y'$ and $\nu z = z'$. Indeed this defines an equivalence relation (denoted by $\sim$) on all such triples and we denote equivalence classes by $(x : y : z)$. This relation allows us to define the *set of R-valued points* of E (denoted by $E(R)$) as the set of all equivalence classes $(x : y : z)$ with $x, y, z \in R$ satisfying (1) such that the *ideal I generated by x, y, z* is R, i.e., $I := \{rx + sy + tz \mid r, s, t \in R\} = R$.

It can be shown (see [21, Section 3]) that the usual chord and tangent process on elliptic curves over fields (cf. [26, Chapter III]) yields a group law on $E(R)$ with identity element $\mathcal{O} := (0 : 1 : 0)$ *if* R has the property that every projective R-module of rank one is free. For our work, it suffices to consider this case, since we will only work over finite rings which have this property. Therefore, we will from now on restrict our attention to finite rings R.

It should be noted that there are explicit and efficient formulae to perform the group law on $E(R)$ which we do not want to recall here due to space limitations (instead we refer to [21,13]). Furthermore, we recall that the Chinese Remainder Theorem on $\mathbb{Z}_N$ (where $N = pq$ is some RSA-modulus) implies natural reduction maps from $E(\mathbb{Z}_N)$ to $E(\mathbb{Z}_p)$ and $E(\mathbb{Z}_q)$. It follows that $E(\mathbb{Z}_N) \cong E(\mathbb{Z}_p) \times E(\mathbb{Z}_q)$ (see [13]).

There are a few other facts in the case where $R = \mathbb{Z}_{N^2}$ for some RSA-modulus $N = pq$, that are of particular interest to us, which follow from the p-adic theory of elliptic curves, and we refer the reader to [13] and [26] for details:

1. $\#E(\mathbb{Z}_{N^2}) = N\#E(\mathbb{Z}_N) = N\#E(\mathbb{Z}_p)\#E(\mathbb{Z}_q)$.
2. $P_i := (Ni : 1 : 0) \in E(\mathbb{Z}_{N^2})$ with $mP_i = P_{mi}$ for all $m \in \mathbb{Z}_N$.
3. $NP_1 = \mathcal{O}$.

Finally, we state that if the factorization of N is not known, the *Decisional Diffie-Hellman Problem* is believed to be hard for elliptic curves over $\mathbb{Z}_{N^2}$. It is defined as follows: Given a random point Q of large order k (meaning that k has about the same size as N), points rQ, sQ and tQ $(r, s, t \in \mathbb{Z}_k)$, it is computationally infeasible to decide whether $t = rs \bmod k$ or not. We denote this problem by $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$. We stress that even if the factorization of N is known, $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$ is still believed to be hard for *randomly* chosen elliptic curves over $\mathbb{Z}_{N^2}$. We will see later that, as in the case where N is prime, if the factorization of N is known, then $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$ can be solved efficiently for *pairing-friendly* curves (here, we mean curves where the reduced Weil or Tate pairing over $\mathbb{Z}_p$ and $\mathbb{Z}_q$ can be efficiently computed). For a detailed discussion on pairing-friendly elliptic curves over fields, we refer to [11], and to [15] when working over rings.

3 (User-Independent) Double Decryption

We start by recalling what it means for an encryption scheme to have a double decryption mechanism. We do this along the lines of Galindo and Herranz's work [16].

Definition 1. *A public key encryption scheme* with a double decryption mechanism (DD-PKE) *is a tuple* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{mDec})$ *of PPT algorithms such that*

Setup: $\mathsf{Setup}(\kappa)$ *takes a security parameter* κ *as input and outputs a tuple* $(\mathsf{PP}, \mathsf{MK})$ *where* PP *contains the public system parameters (particularly includes descriptions of the plaintext space* $\mathcal{P}$ *and the ciphertext space* $\widehat{\mathcal{C}}$*), and* MK *is the master secret key which is only known to the master entity.*

Key Generation: $\mathsf{KeyGen}(\mathsf{PP})$ *takes the system's public parameters* PP *as input and outputs a pair of public/private keys* $(\mathsf{pk}, \mathsf{sk})$ *to a user.*

Encryption: $\mathsf{Enc}_{(\mathsf{PP},\mathsf{pk})}(m)$ *takes the public parameters* PP*, a user's public key* pk *and a message* $m \in \mathcal{P}$ *as input and outputs a ciphertext* $c \in \mathcal{C}$*.*

User Decryption: $\mathsf{Dec}_{(\mathsf{PP},\mathsf{sk})}(c)$ *takes the public parameters* PP*, a user's secret key* sk *and a ciphertext* $c \in \widehat{\mathcal{C}}$ *as input and outputs either a plaintext* $m \in \mathcal{P}$ *or the special symbol* $\perp$*.*

Master Decryption: $\mathsf{mDec}_{(\mathsf{PP},\mathsf{MK},\mathsf{pk})}(c)$ *takes the public parameters* PP*, the master secret key* MK*, a user's public key* pk *and a ciphertext* $c \in \widehat{\mathcal{C}}$ *as input and outputs either a plaintext* $m \in \mathcal{P}$ *or the symbol* $\perp$*.*

For such schemes, we require the usual correctness condition in public key encryption schemes both for the user decryption and the master decryption. It should be noted that by combining the system's public parameters in the user's public keys, we can think of a DD-PKE scheme as being a usual encryption scheme that additionally has a master decryption procedure (that uses the master secret key). Also, we stress that the notion of semantic security is exactly the same as that for usual public-key encryption schemes. Furthermore, it is noteworthy that the key generation algorithm KeyGen does *not* get the master secret MK as input.

Next, we introduce the notion of *User Independence* in the context of such DD-PKE schemes, which basically means that the master entity can decrypt any given ciphertext even without knowing the corresponding receiver (i.e., the user's public key under which it has been encrypted). In other words this means that the master decryption is independent of the users.

Definition 2. *A DD-PKE scheme is* user-independent *(UI-DD-PKE) if the master decryption does* not *get the user's public key as an input, i.e., it only gets the system's public parameters, the master secret and a ciphertext as input.*

4 An Additively Homomorphic UI-DD-PKE Scheme

We introduce a new public key cryptosystem with a simple structure that combines a couple of unique properties in a single scheme. Due to its many properties, we will restrict our attention to the scheme's formal definition and proof of correctness in this section, and deal with its properties in the next section. The semantic security of the scheme will be proven in Section 6. In order to formally define our cryptosystem, we need the following two facts:

Proposition 1. *If* $N = pq$ *is some RSA-modulus, i.e.,* p *and* q *are primes of about the same bit length* κ*, then there is an efficient construction of elliptic curves* $E : y^2z = x^3 + axz^2 + bz^3$ *over* $\mathbb{Z}_{N^2}$ *such that* $M := \mathrm{lcm}(\#E(\mathbb{Z}_p), \#E(\mathbb{Z}_q))$ *has at least two large (of about the same size as* p *and* q*) prime factors.*

Proof. There are three different methods to construct such elliptic curves, which have direct influence on the system's efficiency and applicability. We therefore put the proof of this proposition in a section on its own (see Section 7). □

Lemma 1. *As in Proposition 1, let* $M \in \mathbb{N}$ *have at least two large prime factors (of about* κ *bits). If* $\pi(M)$ *denotes the product of all small prime factors (including multiples) of* M*, then*

$$\Pr_{s \xleftarrow{U} \Pi(M)} [\gcd(s, M) \neq 1] \quad \textit{is negligible in } \kappa,$$

where $\Pi(M) := \{s \in \mathbb{Z}_{N^2} \setminus \{0\} \mid \gcd(s, \pi(M)) = 1\}$*.*

Proof. Let $L(M) = \prod_{i=1}^{r} p_i^{\nu_i}$ be the product ($r \geq 2$) of all large (of about κ bits) prime factors in M, i.e., $M = \pi(M) \cdot L(M)$. By definition, we have that $\Pr[\gcd(s, M) \neq 1 \text{ for } s \in \Pi(M)] = \Pr[s \in \mathbb{Z}_{L(M)} \setminus \mathbb{Z}^*_{L(M)}]$. But if φ denotes Euler's totient function, we have

$$\#\mathbb{Z}^*_{L(M)} = \varphi(L(M)) = \prod_{i=1}^{r}(p_i - 1)p_i^{\nu_i - 1}, \text{ and hence } \frac{\#\mathbb{Z}^*_{L(M)}}{\#\mathbb{Z}_{L(M)}} = \prod_{i=1}^{r}\left(1 - \frac{1}{p_i}\right).$$

However, the fractions $\frac{1}{p_i}$ are negligible in κ, and so the product of all these is negligibly close to 1. Therefore, we have

$$\Pr[s \in \mathbb{Z}_{L(M)} \setminus \mathbb{Z}^*_{L(M)}] = 1 - \frac{\#\mathbb{Z}^*_{L(M)}}{\#\mathbb{Z}_{L(M)}} = 1 - (1 - \mathtt{negl}(\kappa)) = \mathtt{negl}(\kappa),$$

where $\mathtt{negl}(\kappa)$ denotes a negligible function in κ. □

Definition 3 (The Cryptosystem).

Setup: $\mathsf{Setup}(\kappa)$ *computes an RSA-modulus* $N = pq$ *where* p *and* q *are primes of about the same bit length* κ *and constructs an elliptic curve* $E : y^2z = x^3 + axz^2 + bz^3$ *over* $\mathbb{Z}_{N^2}$ *such that* E *has the properties as described in Proposition 1. Furthermore, it chooses a point* $Q = (x : y : z) \in E(\mathbb{Z}_{N^2})$ *whose order divides* $M = lcm(\#E(\mathbb{Z}_p), \#E(\mathbb{Z}_q))$.[1]
It outputs the public parameters $\mathsf{PP} := (N, \pi(M), a, b, Q)$ *and the master secret key* $\mathsf{MK} := M$. *The plaintext space is* $\mathcal{P} = \mathbb{Z}_N$ *and the ciphertext space is* $\widehat{\mathcal{C}} = \langle Q \rangle \times \langle Q, P_1 \rangle$.

Key Generation: $\mathsf{KeyGen}(\mathsf{PP})$ *chooses* $s \in \mathbb{Z}^*_M$ *at random and computes* $R := sQ$. *This can be done by sampling* $s \in \Pi(M)$ *(which is possible as* $\pi(M)$ *is included in* PP*), since then* $s \in \mathbb{Z}^*_M$ *holds with overwhelming probability by Lemma 1.*[2] *It outputs the user's public key* $\mathsf{pk} := R$ *and secret key* $\mathsf{sk} := s$.

Encryption: $\mathsf{Enc}_{(\mathsf{PP},\mathsf{pk})}(m)$ *chooses a random value* $r \in \mathbb{Z}_{N^2}$ *and computes the ciphertext* (A, B) *as*

$$A := rQ \text{ and } B := rR + P_m.^{3}$$

User Decryption: $\mathsf{Dec}_{(\mathsf{PP},\mathsf{sk})}(A, B)$ *outputs*

$$m = \frac{x(B - sA)}{N}.$$

Master Decryption: $\mathsf{mDec}_{(\mathsf{PP},\mathsf{MK})}(A, B)$ *outputs*

$$m = \frac{x(MB)}{N} M^{-1} \bmod N.$$

[1] This can be done by taking a random point $Q' = (x' : y' : z') \in E(\mathbb{Z}_{N^2})$ and setting $Q := NQ'$. See also Section 7.

[2] We note that by using Hasse's bound on $\#E(\mathbb{Z}_p)$ and $\#E(\mathbb{Z}_q)$, we have $M \leq \#E(\mathbb{Z}_p)\#E(\mathbb{Z}_q) \leq N^2$.

[3] We note that if we forget about the first component A of our ciphertexts, then the encryption looks very similar to Galbraith's elliptic-curve-based Paillier scheme [13].

Concerning the *correctness* of both decryption procedures, we see that

$$\mathsf{Dec}_{(\mathsf{PP},\mathsf{sk})}(\mathsf{Enc}_{(\mathsf{PP},\mathsf{pk})}(m)) = \frac{x(rR + P_m - srQ)}{N} = m$$

and

$$\mathsf{mDec}_{(\mathsf{PP},\mathsf{MK})}(\mathsf{Enc}_{(\mathsf{PP},\mathsf{pk})}(m)) = \frac{x(M(rR + P_m))}{N} M^{-1} \bmod N = m$$

by using the fact that $\mathrm{ord}(Q)$ divides M, so $MR = sMQ = \mathcal{O}$.

Remark 1. 1. We stress that the knowledge of M is polynomial-time equivalent to the knowledge of the factorization of N (cf. [23, Theorem 10]). Therefore, it is computationally infeasible to compute the master secret key MK from the public parameters PP.

2. It is also computationally infeasible to compute the user's secret key from its public key under the assumption that the *Discrete Logarithm Problem* (DLP) is hard in $E(\mathbb{Z}_{N^2})$.
3. Without knowledge of the factorization of N it is computationally infeasible to find a point Q' on the curve (that differs from linear combinations of the publicly known points Q, R and P_1), because one would need to solve polynomial equations in $\mathbb{Z}_{N^2}$.
4. We notice that the users' public keys are not needed in the master decryption algorithm, and so we have successfully defined a UI-DD-PKE scheme.
5. Finally, we note that the master decryption never fails on a given ciphertext $c \in \widehat{\mathcal{C}}$, and so it always outputs a message $m \in \mathcal{P}$. This is different for the user decryption. It will output $\perp$ if $x(B - sA)$ is not divisible by N. We will show in the next section (Property 2) that this happens if and only if the given ciphertext is invalid, which users can efficiently detect.

5 Properties of the Cryptosystem

We start with two properties of the cryptosystem that are independent of the choice of elliptic curves in the setup algorithm as long as these curves satisfy the properties of Proposition 1.

Property 1. The cryptosystem is additively homomorphic, i.e.,

$$\mathsf{Dec}_{(\mathsf{PP},sk)}(\mathsf{Enc}_{(\mathsf{PP},\mathsf{pk})}(m_1) + \mathsf{Enc}_{(\mathsf{PP},\mathsf{pk})}(m_2)) = m_1 + m_2.$$

Together with item 4 of Remark 1 this means that the scheme is an additively homomorphic UI-DD-PKE scheme.

Proof. Let $m_1, m_2 \in \mathbb{Z}_N$ be two plaintexts encrypted as (A_1, B_1) and (A_2, B_2), respectively. Then $(A, B) := (A_1, B_1) + (A_2, B_2)$ is a ciphertext of $m := m_1 + m_2$ since

$$\begin{aligned}\frac{x(B - sA)}{N} &= \frac{x(r_1 R + P_{m_1} + r_2 R + P_{m_2} - sr_1 Q - sr_2 Q)}{N} \\ &= \frac{x(P_{m_1+m_2})}{N} = \frac{(m_1 + m_2)N}{N} = m_1 + m_2.\end{aligned}$$

□

Property 2. Users can detect invalid ciphertexts.

Proof. By definition (see also item 3 of Remark 1), a ciphertext c is of the form $(A, B) = (rQ, tQ + P_m) \in \langle Q \rangle \times \langle Q, P_1 \rangle$ (recall that $mP_1 = P_m$ and $\mathrm{ord}(P_1) \mid N$; cf. Section 2). If s denotes a user's private key, we know that a ciphertext c is valid if and only if $t = rs \bmod \mathrm{ord}(Q)$, which in turn is equivalent to saying that $B - sA = P_m$ (recall that $P_m \notin \langle Q \rangle$ for all $0 \neq m \in \mathbb{Z}_N$). □

There are a couple of interesting properties of the cryptosystem that depend on the actual choice of the elliptic curve in the setup algorithm. We start with the detection of invalid ciphertexts for the master entity.

Property 3. If $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$ is hard in $E(\mathbb{Z}_{N^2})$ (even when the factorization of N is known), then the master entity, when given a user's public key, *cannot* decide whether a given ciphertext is a valid encryption under this public key or not.

Proof. Assume that the master can detect invalid ciphertexts which are, by definition, of the form $(A, B) = (rQ, tQ + P_m) \in \langle Q \rangle \times \langle Q, P_1 \rangle$. Then we can use this detection algorithm to solve $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$ as follows: Given a $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$-tuple (Q, rQ, sQ, tQ), we just check the ciphertext $(A, B) = (rQ, tQ)$ for validity under the public key sQ. Clearly, we have:

$$(A, B) \text{ is valid } \iff (Q, rQ, sQ, tQ) \text{ is a valid } \mathsf{DDH}_{\mathbb{Z}_{N^2}}\text{-tuple.}$$

□

As explained in the Introduction, there are applications where the master entity should be able to check ciphertexts for validity as well. This is where pairings come into play. We will see that our cryptosystem is actually a nice application of *hidden pairings* – a notion introduced by Dent and Galbraith [9]. Therein, they present an identification scheme as a cryptographic application which was the only interesting application known until now. Unfortunately, since our scheme uses elliptic curves with certain properties in a non-black-box way, we cannot use the construction of a "hidden pairing"-group of [9] directly, but need to construct our own. Our construction is given in the following result that we prove in Section 7:

Lemma 2. *There is an efficient construction of an elliptic curve E over $\mathbb{Z}_{N^2}$ with properties as in Proposition 1 together with a point $Q \in E(\mathbb{Z}_{N^2})$ of large order dividing M such that*

1. *if Q_1 and Q_2 denote the natural reductions of Q to $E(\mathbb{Z}_p)$ and $E(\mathbb{Z}_q)$, respectively, we have that $ord(Q) = lcm(ord(Q_1), ord(Q_2))$*
2. *we can efficiently compute the 'reduced' Tate pairings τ_p and τ_q on E over $\mathbb{Z}_p$ and $\mathbb{Z}_q$, respectively.*

Since the Tate pairings τ_p and τ_q can only be computed if the factorization of N is known, they are called *hidden pairings*. Concerning the security of elliptic curves with properties as in the Lemma, we refer the reader to [15] and [9]. This Lemma has an interesting consequence on our cryptosystem:

Property 4. Let Q be a point on an elliptic curve E over $\mathbb{Z}_{N^2}$ as in Lemma 2. If our cryptosystem uses E and Q in its public parameters, then the master entity can detect invalid ciphertexts under a given user's public key.

Proof. Let $R = sQ$ be a user's public key and let $(A, B) = (rQ, tQ + P_m) \in \langle Q \rangle \times \langle Q, P_1 \rangle$ be a ciphertext. In order to check the validity of (A, B) under R, the master entity first uses the master secret M to compute the plaintext m (by using mDec). Since the master knows the factorization of N, it can now compute the reductions modulo p of Q, R, A and $T := B - P_m = tQ$ in $E(\mathbb{Z}_p)$ which we denote by Q_1, R_1, A_1 and T_1, respectively. Additionally, let $Q_2, R_2, A_2, T_2 \in E(\mathbb{Z}_q)$ be the respective reductions modulo q. Since the master can efficiently compute the 'reduced' Tate pairings τ_p and τ_q, respectively, it can check whether (Q_1, R_1, A_1, T_1) and (Q_2, R_2, A_2, T_2) are valid DDH-tuples in $E(\mathbb{Z}_p)$ and $E(\mathbb{Z}_q)$, respectively, in the usual way (see [12] and [22]). We have the relation that (Q_1, R_1, A_1, T_1) is a valid DDH-tuple in $E(\mathbb{Z}_p)$ if and only if $t = sr \bmod \operatorname{ord}(Q_1)$. An analogous relation holds for the prime q. Together, the Chinese Remainder Theorem over $\mathbb{Z}_{\operatorname{ord}(Q)}$ yields that (Q_1, R_1, A_1, T_1) and (Q_2, R_2, A_2, T_2) are valid DDH-tuples over their respective prime fields if and only if $t = rs \bmod \operatorname{ord}(Q)$ which in turn holds if and only if (A, B) is a valid ciphertext under R. □

6 Semantic Security

Considering the fact that our cryptosystem is an additive variant of the El-Gamal cryptosystem, it is rather obvious that it is semantically secure under the $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$-assumption. Therefore, and due to space limitations, we only give a proof sketch of this fact here: Proving semantic security of additively homomorphic cryptosystems boils down to proving that a random encryption is computationally indistinguishable from an encryption of 0 (e.g., Armknecht et al. [2]). In our cryptosystem, a random encryption has the form $(rQ, rR + P_m)$ with randomness $r \in \mathbb{Z}_{N^2}$ and random message $m \in \mathbb{Z}_N$. An encryption of 0, on the other hand, has the form (rQ, rR) for randomness $r \in \mathbb{Z}_{N^2}$. Now, if we write $X = rQ$ and $S = rR + P_m$ for randomness $r \in \mathbb{Z}_{N^2}$ and random message $m \in \mathbb{Z}_N$, we see that semantic security states: Given points $X, R \in \langle Q \rangle$ and given a random point S, decide whether $\log_Q(S) = \log_Q(X) \log_Q(R)$. This problem is the $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$-problem, except that S is chosen from a larger group (and not only from $\langle Q \rangle$). However, $\mathsf{DDH}_{\mathbb{Z}_{N^2}}$ reduces to this more general problem.

In practice, from an adversary's point of view the situation is even worse, since without knowledge of the factorization of N it is extremely hard to find a point Q' on the curve at all (that differs from linear combinations of the publicly known points Q, R and P_1), because one would need to solve polynomial equations in $\mathbb{Z}_{N^2}$. It should be mentioned though that the security highly depends on the order of the point Q. Therefore, one should always take great care in the setup of the cryptosystem that the point Q really has large order (of about the same size as the prime factors of N).

Finally, we note that concerning the size of the security parameter of our scheme, we need to ensure that the bit length of the primes p and q is roughly

512 (at least). This yields a 1024 bit RSA-modulus and so we can assume that factoring such a large number is indeed hard in practice. Since solving discrete logarithms on elliptic curves over prime fields is assumed hard if the bit length of the order of the underlying prime field is about 180, having 512 bits here makes it reasonable to assume that the DLP is indeed hard on our chosen curves. Such a parameter setting is similar to the settings of [13] and [9], where it is argued that one can assume a high level of security while having efficient group operations on the curve at the same time.

7 Concrete Setup of the System's Parameters

The basic goal of this section is to prove Proposition 1 and Lemma 2, i.e., to give efficient constructions of elliptic curves with the properties as described in the respective claim. Since curves satisfying Lemma 2 will also satisfy Proposition 1, we start with the latter (Method 1) and then look at which of these curve additionally satisfy the Lemma (Methods 2 and 3).

Method 1: Random Curves. Given a security parameter κ (which in practice will be of size 512), the fundamental idea is to choose two distinct, random primes p and q of about κ bits (so $N = pq$ is our RSA-modulus) together with two random elliptic curves E_1 and E_2 over $\mathbb{Z}_p$ and $\mathbb{Z}_q$, respectively. We require that both $E_1(\mathbb{Z}_p)$ and $E_2(\mathbb{Z}_q)$ have at least one large prime factor (of about κ bits) – so we discard all curves not having this property and repeat choosing random curves until we find two suitable elliptic curves. Then, by using standard techniques (i.e., considering E_1 and E_2 over $\mathbb{Z}_{p^2}$ and $\mathbb{Z}_{q^2}$, respectively (cf. Lemma 3), and then using the Chinese Remainder Theorem), we construct an elliptic curve E over $\mathbb{Z}_{N^2}$ such that $M := \operatorname{lcm}(\#E(\mathbb{Z}_p), \#E(\mathbb{Z}_q))$ has at least two large prime factors.

We remark that concerning the security of our cryptosystem, this way of constructing the elliptic curves prevents an attacker to exploit any particular structure of the used elliptic curve.

Likelyhood of hitting on such curves. One problem with this approach concerns the likelyhood of hitting on such curves by random sampling given a prime p. Since there is no final answer to this question in theory, we have to rely on a conjecture by Galbraith and McKee [14]: First, let us only consider elliptic curves E with prime order. It is conjectured that

$$\Pr[\#E(\mathbb{Z}_p) \text{ is prime}] \text{ is asymptotic to } c_p \frac{1}{\log p} \text{ as } p \to \infty,$$

where

$$c_p = \frac{2}{3} \prod_{l>2} \left(1 - \frac{1}{(l-1)^2}\right) \prod_{2<l|p-1} \left(1 + \frac{1}{(l+1)(l-2)}\right)$$

and the probability is over all random primes p and $(a,b) \xleftarrow{U} \mathbb{Z}_p^2 \setminus \{(a,b) \in \mathbb{Z}_p^2 \mid 4a^3 + 27b^2 = 0\}$. We ran some numerical tests ourselves (see Table 1) which confirm the conjecture in practice.

Table 1. Numerical probability of hitting on a curve with prime order

Bit length of p	64	128	192	256
$\Pr[\#E(\mathbb{Z}_p)$ is prime$]$	1.17 %	0.58 %	0.38 %	0.27 %

As we have discussed before, we actually do not need the curve to have a large prime order, but only a nearly prime order. Therefore, we can optimize our search for elliptic curves by using a result by Lenstra [20] that small prime factors appear with a high probability. The idea is to fix a set S of small primes and allow $\#E(\mathbb{Z}_p)$ to be divisible by powers of $s \in S$. This increases the probability to hit on a curve with a large prime dividing the order by a huge factor (e.g., for orders of the form 2^k·prime this factor is about 3, while for orders of the form $2^k \cdot 3^l$·prime the factor is about 5.5 in our numerical results). This was also conjectured by Galbraith and McKee in [14] and our numerical tests give evidence for this conjecture (cf. Table 2).

Table 2. Numerical probability of hitting on a curve with nearly prime order

Bit length of p	64	128	192	256
$\Pr[\#E(\mathbb{Z}_p) = 2^k \cdot \text{prime}]$	3.61 %	1.78 %	1.28 %	0.90 %
$\Pr[\#E(\mathbb{Z}_p) = 2^k \cdot 3^l \cdot \text{prime}]$	6.58 %	3.06 %	2.23 %	1.45 %

Concerning the efficiency of constructing curves as in Proposition 1, our experiments show that for an RSA-modulus of 512 bits (i.e., two primes of about 256 bits) it takes roughly 15 minutes using MAGMA on a single core of an Intel Xeon running at 2.5 GHz. For an 1024 bit RSA-modulus, it takes approximately 13 hours per curve. Allowing primes of up to three dividing the group order, we were able to generate five pairs of elliptic curves in approximately two days, while allowing prime factors of up to 13, this time halves (cf. Table 3). Since the Setup algorithm of our cryptosystem needs to be run only once, such an efficiency is reasonable in practice.

Table 3. Numerical results for the runtime of the Setup algorithm for $\log p = 512$ and 5 keys generated, where $S := \{\text{prime } \mathfrak{p} \mid \mathfrak{p} \text{ is allowed to divide } \#E(\mathbb{Z}_p)\}$

	$S = \{2,3\}$	$S = \{2,3,5\}$	$S = \{2,3,5,7,11\}$	$S = \{2,3,5,7,11,13\}$
Time	2d 6h 4m 20s	23h 15m 10s	2d 5h 49m 41s	1d 4h 43m 17s
Tested Curves	1298	552	1266	687

Performing our cryptosystem's setup. Recall that for a high level of security it is not enough to find suitable elliptic curves, we should also choose the point $Q \in E(\mathbb{Z}_{N^2})$ in the Setup algorithm to be of large order dividing $M = \operatorname{lcm}(\#E(\mathbb{Z}_p), \#E(\mathbb{Z}_q))$ (cf. Section 6). The following two lemmata can be used in order to do this:

Lemma 3. *For a prime $p > 3$ and an elliptic curve E over $\mathbb{Z}_p$, we can efficiently construct an elliptic curve E' over $\mathbb{Z}_{p^2}$ such that $E'(\mathbb{Z}_{p^2})$ has order $\#E(\mathbb{Z}_p) \cdot p$ and the reduction from $\mathbb{Z}_{p^2}$ to $\mathbb{Z}_p$ induces a group homomorphism from $E'(\mathbb{Z}_{p^2})$ to $E(\mathbb{Z}_p)$.*

Proof. Let E be given by the short Weierstrass equation $y^2z = x^3 + axz^2 + bz^3$ over $\mathbb{Z}_p$. The existence of E' such that it reduces to E is simple, because it is sufficient to define E' by the same Weierstrass equation. Since the discriminant of E is invertible modulo p it also is modulo p^2, thus E' is an elliptic curve. Due to the geometric definition of the elliptic curve group law, the existence of the induced group homomorphism is obvious. It is left to be proven that this homomorphism is surjective.

Fix any finite point $P = (x_0 : y_0 : 1) \in E(\mathbb{Z}_p)$, then y_0 is a solution to the polynomial equation $0 = y^2 - (x_0^3 + ax_0 + b)$. In the case $y_0 \not\equiv 0 \pmod{p}$ there is a unique integer $0 \leq k < p$ such that $(y_0 + kp)^2 - (x_0^3 + ax_0 + b) \equiv 0 \pmod{p^2}$ by Hensel's lifting lemma. The new point $(x_0 : y_0 + kp : 1)$ obviously reduces to the initial point. In the case $y_0 \equiv 0 \pmod{p}$ we know that x_0 has been a solution for $0 = x^3 + ax + b \pmod{p}$. Since this polynomial cannot have any double roots we can find a solution $x_0 + kp$ for the same equation modulo p^2. This proves surjectivity.

To compute the order of $E'(\mathbb{Z}_{p^2})$ it is sufficient to compute the kernel of the reduction to $E(\mathbb{Z}_p)$. Obviously there are exactly p points $(kp : 1 : 0)$ for $0 \leq k < p$ on $E'(\mathbb{Z}_{p^2})$ that reduce to the point at infinity on $E(\mathbb{Z}_p)$. Thus $\#E'(\mathbb{Z}_{p^2}) = \#E(\mathbb{Z}_p) \cdot p$ holds due to the homomorphism theorem. □

Lemma 4. *Let $p > 3$ be a prime, E be an elliptic curve defined over $\mathbb{Z}_p$ and $P \in E(\mathbb{Z}_p)$ a point with $\gcd(ord(P), p) = 1$. Then the curve $E'(\mathbb{Z}_{p^2})$ constructed as in Lemma 3 contains a point P' of order $ord(P)$.*

Proof. Let Q' be any preimage of P under the reduction map (Q' can be constructed following the proof of Lemma 3). By the homomorphism theorem, we have $\operatorname{ord}(P) \mid \operatorname{ord}(Q')$. Multiplying both points with p permutes the subgroup generated by P on $E(\mathbb{Z}_p)$ and $P' := pQ'$ has order $\operatorname{ord}(P') = \operatorname{ord}(pP) = \operatorname{ord}(P)$ since the order of P is coprime to p. □

Now, the construction of a point $Q \in E(\mathbb{Z}_{N^2})$ with large order dividing M, where E is a random curve such that M has at least two large prime factors (as in Proposition 1), works as follows:

1. Choose a random RSA-modulus $N = pq$ and random elliptic curves $E_1(\mathbb{Z}_p)$, $E_2(\mathbb{Z}_p)$ with nearly prime order as described before.
2. Pick points $P_1 \in E_1(\mathbb{Z}_p)$ and $P_2 \in E_2(\mathbb{Z}_q)$ of high order coprime to p and q.

3. Apply Lemma 3 and 4 to construct elliptic curves E'_1 and E'_2 with points $P'_1 \in E'_1(\mathbb{Z}_{p^2})$ and $P'_2 \in E'_2(\mathbb{Z}_{q^2})$ of high order.
4. Use the Chinese Remainder Theorem to merge E'_1 and E'_2 to a single curve E defined over the ring $\mathbb{Z}_{N^2}$. The lift Q of P'_1 and P'_2 will have order $\mathrm{lcm}(\mathrm{ord}(P_1), \mathrm{ord}(P_2))$ and is the point used for the public parameters PP.

Method 2: Supersingular Curves. A more efficient way to construct elliptic curves that satisfy Proposition 1 is by using supersingular curves E and particular RSA-moduli $N = pq$. For such curves it is known that over a prime field $\mathbb{Z}_p$ we have $\#E(\mathbb{Z}_p) = p + 1$. We note that the following discussion can be done for arbitrary supersingular elliptic curves, however, we restrict our attention to the following family of curves:

Lemma 5 (see [19]). *Let p be an odd prime with $p \equiv 2 \pmod 3$ and let $0 \neq b \in \mathbb{Z}_p$. Consider the elliptic curve $E : y^2 = x^3 + b$. Then, $E(\mathbb{Z}_p)$ is cyclic and $\#E(\mathbb{Z}_p) = p + 1$.*

So if we start with a *strong* prime p (i.e., $p + 1$ is not smooth) with $p \equiv 2 \pmod 3$ and setting E to be the curve given by the equation $y^2 = x^3 + b$ for some $0 \neq b \in \mathbb{Z}_p$, we ensure a large factor in $\#E(\mathbb{Z}_p) = p + 1$. To construct a strong prime p fulfilling the congruence condition it is possible to take a prime p' of the desired bit length κ such that $p := 6p' - 1$ is also prime.

Now, by using Lemmas 3 and 4, we can construct an elliptic curve together with a point Q of high order suitable for our cryptosystem in exactly the same way as we did in Method 1 (items 2 – 4 in the construction therein). We remark that constructing the elliptic curves in this way gives a very fast and easy setup of our system.

Additional property: Hidden pairing. Since supersingular elliptic curves have an embedding degree of at most $k = 6$, they allow for an efficient evaluation of the 'reduced' Tate pairing, which can then be used to solve DDH-challenges [12,22]. Therefore, our just constructed elliptic curve E over $\mathbb{Z}_{N^2}$ has a hidden pairing [9], and we can efficiently solve DDH if the factorization of N is known. This proves Lemma 2.

Method 3: Complex Multiplication. The CM method [3] allows us to construct elliptic curves E together with primes p and q such that E satisfies Proposition 1. Even more, by using extended algorithms [11], it is possible to construct E over $\mathbb{Z}_{N^2}$ such that it has a small embedding degree over the prime fields $\mathbb{Z}_p$ and $\mathbb{Z}_q$. This yields another construction satisfying Lemma 2.

8 Conclusions

We presented a new additively homomorphic UI-DD-PKE scheme that combines many interesting properties in just one scheme. Most importantly, by choosing the system's parameters (i.e., the elliptic curves) appropriately, our scheme is the first that allows the master entity to check for invalid ciphertexts, additionally to being a UI-DD-PKE scheme. Such a cryptosystem has practical relevance in hierarchical organisations, e.g., in order to reduce key management, or to deal with the problem of key loss. Additionally, the ability to check for invalid ciphertexts might be useful in electronic voting systems where some form of "after the fact" validity checks of votes are required [1]. Finally, we note that due to its ElGamal-like structure, our cryptosystem is likely to be *anonymous* [4] and hence it would be interesting to investigate the effect of the double decryption mechanism on known constructions such as *group encryption* [18]. Further potential future work includes an analysis of the MOV-attack [22] in hidden pairing scenarios, and the possibilities of extracting the randomness used to encrypt a message in our scheme, which would probably yield a practical trapdoor discrete logarithm group.

References

1. Adida, B.: Helios: Web-based open-audit voting. In: USENIX Security Symposium, pp. 335–348. USENIX Association (2008)
2. Armknecht, F., Katzenbeisser, S., Peter, A.: Group homomorphic encryption: characterizations, impossibility results, and applications. Designs, Codes and Cryptography, 1–24, doi:10.1007/s10623-011-9601-2
3. Atkin, A.O.L., Morain, F.: Elliptic curves and primality proving. Math. Comp. 61, 29–68 (1993)
4. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-Privacy in Public-Key Encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001)
5. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)
6. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) lwe. In: FOCS, pp. 97–106. IEEE (2011)
7. Bresson, E., Catalano, D., Pointcheval, D.: A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 37–54. Springer, Heidelberg (2003)
8. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
9. Dent, A.W., Galbraith, S.D.: Hidden Pairings and Trapdoor DDH Groups. In: Hess, F., Pauli, S., Pohst, M. (eds.) ANTS 2006. LNCS, vol. 4076, pp. 436–451. Springer, Heidelberg (2006)
10. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP J. Inf. Secur. 2007, 15:1–15:15 (2007)
11. Freeman, D., Scott, M., Teske, E.: A taxonomy of pairing-friendly elliptic curves. J. Cryptology 23(2), 224–280 (2010)
12. Frey, G., Rück, H.G.: A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Math. Comput. 62, 865–874 (1994)
13. Galbraith, S.D.: Elliptic curve paillier schemes. J. Cryptology 15(2), 129–138 (2002)

14. Galbraith, S.D., McKee, J.F.: The probability that the number of points on an elliptic curve over a finite field is prime. Journal of the LMS 62(03), 671–684 (2000)
15. Galbraith, S.D., McKee, J.F.: Pairings on Elliptic Curves over Finite Commutative Rings. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 392–409. Springer, Heidelberg (2005)
16. Galindo, D., Herranz, J.: On the security of public key cryptosystems with a double decryption mechanism. Inf. Process. Lett. 108(5), 279–283 (2008)
17. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, pp. 169–178. ACM (2009)
18. Kiayias, A., Tsiounis, Y., Yung, M.: Group Encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 181–199. Springer, Heidelberg (2007)
19. Koyama, K., Maurer, U.M., Okamoto, T., Vanstone, S.A.: New Public-Key Schemes Based on Elliptic Curves over the Ring Z_n. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 252–266. Springer, Heidelberg (1992)
20. Lenstra, H.W.: Factoring integers with elliptic curves. Annals of Mathematics, 649–673 (1987)
21. Lenstra, H.W.: Elliptic curves and number theoretic algorithms. In: Proceedings of the International Congress of Mathematicians, pp. 99–120 (1988)
22. Menezes, A., Okamoto, T., Vanstone, S.A.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Trans. Inf. Theory 39(5), 1639–1646 (1993)
23. Okamoto, T., Uchiyama, S.: Security of an Identity-Based Cryptosystem and the Related Reductions. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 546–560. Springer, Heidelberg (1998)
24. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
25. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
26. Silverman, J.H.: The Arithmetic of Elliptic Curves. GTM, vol. 106. Springer (1986)
27. Youn, T.-Y., Park, Y.-H., Kim, C.-H., Lim, J.: An Efficient Public Key Cryptosystem with a Privacy Enhanced Double Decryption Mechanism. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 144–158. Springer, Heidelberg (2006)