Compact Identity Based Encryption Based on $n^{th}-$Residuosity Assumption

S. Sree Vivek[1], S. Sharmila Deva Selvi[1], Ramarathnam Venkatesan[2], C. Pandu Rangan[1]

[1]TCS Lab, Dept of CSE, IIT Madras, Chennai, India.
{svivek, sharmila,prangan}@cse.iitm.ac.in
[2]Microsoft Research, One Microsoft Way, Redmond, WA 98052.
venkie@microsoft.com

Abstract. Practical Identity Based Encryption (IBE) schemes use the costly bilinear pairing computation. Clifford Cock proposed an IBE based on quadratic residuosity in 2001 which does not use bilinear pairing but was not efficient in practice, due to the large ciphertext size. In 2007, Boneh et al. proposed the first space efficient IBE that was also based on quadratic residuosity problem. It was an improvement over Cock's scheme but still the time required for encryption was quartic in the security parameter. In this paper, we propose a compact, space and time efficient identity based encryption scheme without pairing, based on a variant of Paillier Cryptosystem and prove it to be CPA secure. We have also proposed a CCA secure scheme based on the basic IBE scheme using the Fujisaki-Okamoto transformation. We have proved both the schemes in the random oracle model.

Keywords: Identity Based Encryption, Paillier Encryption Scheme, Pairing-free, Composite n^{th}-Residuosity.

1 Introduction

The technique of using the identity of a user as the public key was first introduced by Adi Shamir [28], who gave a general overview of Identity Based Cryptosystem (IBC) and proposed a possible construction for Identity Based Signatures (IBS) but left IBE unsolved. Practical IBE appeared in the dawn of the 21^{st} century. Sakai et al. [24, 25], Boneh and Franklin [6], independently proposed two schemes based on bilinear pairing, following them Cock [12] came out with a completely different construct, based on quadratic residuosity problem. The above three results in succession came after a comparatively very long pause of almost two decades from the introduction of IBC. Although, several attempts are made, solutions prior to the year 2000 were fundamentally inefficient (Tanaka [30], Tsujii and Itoh [31], Maurer and Yacobi [22] and Steiner et al. [29]). Improvements to the Boneh-Franklin IBE scheme resulted in new schemes and a few totally new approaches to IBE have evolved since 2001. However, they rely on bilinear pairing, which is a powerful mathematical construct defined over certain algebraic curves, but are costly operations in cryptography.

Bilinear pairing (bilinear map), is a function from two cyclic groups into a third group, the algebraic groups that define its domain and co-domain should have the discrete logarithm and related computational and decisional problems in-feasibly hard. Bilinear pairings have quickly become very important in cryptographic protocols [2, 27, 15]. Identity Based Cryptography and IBE in particular can be regarded as a best application of bilinear pairings in cryptography. There are many practical identity based encryption schemes proved to be secure in the random oracle model [9, 16, 1, 10, 11, 26] and in the standard model [5, 4, 32, 17, 19, 33, 3].

In this regard, Cock's IBE is pairing-free and is based on the standard quadratic residuosity problem modulo an RSA composite N (in the random oracle model). One of the main limitations is that the ciphertext in this system contain two elements in $\mathbb{Z}/N\mathbb{Z}$ [7] for every bit of plaintext. Hence, the encryption of an l-bit message is of size $2l.log_2N$. In contrast, the pairing-based scheme produced a very small ciphertext (roughly, thousandth) for comparable security. A long standing open problem solved by Boneh et al. in [7] is the construction of a space efficient IBE system without pairings. They have used the theory of quadratic forms, in particular, encryption and decryption are based on an effective version of Legendre's famous three squares theorem [7], which can be considered as an extension of Cock's IBE. Encryption time in [7] is still not ideal, in the sense that, encryption time in most practical public key systems such as RSA and existing IBE systems

is cubic in the security parameter but in Boneh's pairing-free system it is quartic in the security parameter per message bit. However, decryption is cubic as in other systems. The bottleneck during encryption is the need to generate primes on the order of N. Barua et al. [18] proposed an improvement for this scheme, which has an efficient encryption algorithm but the ciphertext size blows up to $2\left\lceil\sqrt{l}\right\rceil$ elements of $\mathbb{Z}/N\mathbb{Z}$, which makes it impractical.

A slightly different approach was used by Ding et al. for realizing a pairing free IBE scheme in [13]. They use another party called security mediator (SEM) to share the responsibilities in the decryption process. Since the collusion of SEM with any other user may lead to total break of the scheme and the security of this scheme is not established even for a CPA model (only non-invertibility was proved in [13]), this scheme also remains impractical and weak.

Our Contribution: In view of the state-of-the-art, the construction of an efficient, pairing free identity based encryption scheme turns out to be a problem of significant practical and theoretical interest. In 2008, Boneh et al. [8] proved that it is impossible to base IBE on trapdoor permutations like RSA. They showed that there is no black-box construction of IBE from trapdoor permutations or even chosen ciphertext secure public key encryption. In section 6.1 of [8] they have also commented that their oracle separation also rules out black-box IBE construction from number-theoretic encryption systems such as Paillier system [23]. However, in this paper, we show that with some appropriate modifications to **PE**, it is possible to construct a compact IBE scheme. In sum, our result would expose the richness of mathematical structure underlying Paillier cryptosystem namely, the number theoretic power of using the square of RSA modulus.

2 Preliminaries

We review the hard problems, facts, concepts which are required in various parts of the paper in this section.

Isomorphism in $\mathbb{Z}^*_{n^2}$: The group $\mathbb{Z}^*_{n^2}$ is isomorphic to the direct product of two groups $\mathbb{Z}_n \times \mathbb{Z}^*_n$, where the group $\mathbb{Z}_n$ is additive modulo n and the group $\mathbb{Z}^*_n$ is multiplicative modulo n. Thus there exists an isomorphism of the following form [23]:

$$\psi : \mathbb{Z}_n \times \mathbb{Z}^*_n \stackrel{iso}{\rightarrow} \mathbb{Z}^*_{n^2} \tag{1}$$

This isomorphism can be explicitly specified as follows. We fix a $u \in \mathbb{Z}^*_{n^2}$, whose order is n. Then g may be written as:

$$\psi_u(x, y) = u^x y^n, \; for\, some\, unique\, y \in \mathbb{Z}^*_n \tag{2}$$

When the details of u is irrelevent, we may write $\psi_u(x, y)$ simply as $\psi(x, y)$.
If g is an n^{th} residue modulo n^2 then g is given by:

$$g = y^n = \psi(0, y), \; (For\, some\, unique\, y) \tag{3}$$

Decisional n^{th} Power Residue Problem: Given g, decide if $g = y^n \; mod \; n^2$ for some y. In other words decide whether a given element g is of the form $\psi(0, y)$, for some y.

n^{th} Power Residue Distinguishability: Distinguish the distributions μ_1 and μ_2 where, μ_1 is uniform distribution on $\mathbb{Z}^*_{n^2}$ and μ_2 is the uniform distribution over n^{th} power residues in $\mathbb{Z}^*_{n^2}$. The distinguishability property may be stated in terms of the isomorphism function ψ as follows: Given g, determine whether $g = \psi(x, y)$, for some $x \neq 0$ or not.

Both the problems stated above are assumed to be hard in the sense that any polynomial time randomized algorithm will produce correct answer only with negligible probability.

New Factorization Assumption: Let p and q be two primes with $|p| = |q| = \kappa$, let $n = pq$, let $g \in_R \mathbb{Z}^*_{n^2}$ and $g < n$. Given $\left\langle n, g, g^\lambda \, mod \, n^2 \right\rangle$ where $\lambda = lcm(p-1, q-1)$, factoring n is hard.

Proof Sketch: Let us assume that there exists an adversary $\mathcal{A}$, who is given $\left\langle n, g, g^\lambda \right\rangle$ could factor n. If such an adversary exists, then **MPE** scheme is not secure. Since theorem 1 proves that **MPE** is as secure as the

Decision n^{th}-Residuosity assumption, we assume that the New Factorization Assumption is hard to break by any polynomial time adversary $\mathcal{A}$.

Digital Signature: Let $KeyGen$ be an algorithm that generates a secret signing key sk and a public verification key pk via $\langle sk, pk\rangle = KeyGen(1^{\lambda})$. Then, the value $\sigma = Sign(sk, m)$ is a digital signature for the message m generated by a signing algorithm $Sign$. The signature σ is verified by a public function $Verify(pk, m, \sigma) \rightarrow \{0, 1\}$ in binary corresponds to whether or not σ is a valid signature for m using the signing key sk.

Identity Based Encryption: An identity based encryption scheme consists of four algorithms, namely: $Setup$, $Key\,Extract$, $Encryption$ and $Decryption$:

Setup(κ): This algorithm is executed by the Private Key Generator (PKG), a trusted authority in the system. This algorithm takes the security parameter κ as input and returns the system parameters $params_{IBE}$ and the master private key msk_{IBE}. The system parameters are the description of the message space, and the description of the ciphertext space. The system parameters $params_{IBE}$ is publicly known, while the master private key msk_{IBE} is known only to the PKG.

Key Extract($params_{IBE}, msk_{IBE}, ID_X$): This algorithm is executed by the PKG to generate the private key of a user with identity ID_X. It takes as input the system parameters $params_{IBE}$, the master private key msk_{IBE} and an arbitrary identity $ID_X \in \{0, 1\}^*$ as input, and returns a private key SK_X corresponding to the identity ID_X. Here ID_X is an arbitrary string that will be used as a public key. In sum the Key Extract algorithm extracts a private key from a given identity.

Encrypt($params_{IBE}, ID_X, m$): This algorithm is executed by the sender to encrypt the message m. It takes $params_{IBE}$, ID_X and the message m as input and returns the ciphertext C.

Decrypt($params_{IBE}, SK_X, C$): This algorithm is executed by the receiver. It takes $params_{IBE}$, the ciphertext C and a private key SK_X as input. The output of this algorithm is the message m.

EUF-CMA Security Notion: A signature scheme is EUF-CMA (Existentially Forgeable under Chosen Message Attack) secure if no polynomially bounded adversary **A** has a non-negligible advantage against a challenger **C** in the following EUF-CMA game:

KeyGen: **C** runs the key generation algorithm $\langle sk, pk\rangle = KeyGen(1^{\lambda})$ and sends the public verification key pk to **A**.

Training Phase: **A** makes any number of signing queries to **C** by sending message m. **C** sends the signature on m by executing the signing algorithm $\sigma = Sign(sk, m)$ and send σ to **A**

Forgery: **A** forges a new message/signature pair (m^*, σ^*) and sends it to **C**. The challenger accepts the message/signature pair only if σ^* was not the output of the signing oracle for message m^* during the training phase and $Verify(pk, m^*, \sigma^*) \rightarrow 1$.

The signature scheme is said to be EUF-CMA secure if for all probabilistic polynomial time adversary **A**, there exists a negligible function ϵ such that:

$$Adv_{\mathbf{A}}(\lambda) < \epsilon(\lambda)$$

CPA Security Notion: An identity based encryption scheme is semantically secure against chosen plaintext attack (IND-CPA) if no polynomially bounded adversary **A** has a non-negligible advantage against the challenger **C** in the following IND-CPA game:

Setup: **C** takes a security parameter κ and runs the Setup algorithm. **C** gives **A** the system parameters $params_{IBE}$ and keeps the master key private.

Phase 1: The adversary is allowed to query the following oracle:

- *Key Extract Oracle(ID_X):* **C** responds by running the $Key\,Extract$ algorithm to generate the private key SK_X corresponding to the identity ID_X. **C** sends SK_X to **A**.

Challenge: Once the Phase 1 is over (which is decided by **A**), **A** gives to **C** two plaintexts m_0, m_1 of equal length and an identity ID_T on which **A** wishes to be challenged. The only constraint is that the private Key Extract oracle should not have been queried with ID_T as input during Phase 1. **C** picks a random bit $b \in_R \{0,1\}$ and sets $C^* = Encryption(params_{IBE}, ID_T, m_b)$. **C** sends C^* as the challenge to **A**.

Phase 2: **A** issues more queries as in Phase 1.

Guess: Finally, **A** outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

The adversary $\mathbf{A}'s$ advantage in attacking the identity based encryption scheme is the following function of the security parameter κ

$$Adv_{\mathbf{A}}(\kappa) = |Pr[b = b'] - \frac{1}{2}|$$

CCA Security Notion: An identity based encryption scheme is semantically secure against adaptive chosen ciphertext attack (IND-CCA) if no polynomially bounded adversary **A** has a non-negligible advantage against the challenger **C** in the following IND-CCA game:

Setup: **C** takes a security parameter κ and runs the Setup algorithm. **C** gives **A** the system parameters $params_{IBE}$ and keeps the master key private.

Phase 1: The adversary issues queries to the following oracles:

- *Key Extract Oracle(ID_X):* **C** responds by running the *Key Extract* algorithm to generate the private key SK_X corresponding to the identity ID_X. **C** sends SK_X to **A**.
- *Decryption Oracle(ID_X, C):* **C** responds by sending the resulting plaintext m to **A**. These queries are asked adaptively, that is, each query may depend on the replies to previous queries.

Challenge: Once the Phase 1 is over, **A** outputs two plaintexts m_0, m_1 of equal length and an identity ID_T on which **A** wishes to be challenged. The only constraint is that the private Key Extract oracle should not have been queried with ID_T as input during Phase 1. **C** picks a random bit $b \in_R \{0,1\}$ and sets $C^* = Encryption(params_{IBE}, ID_T, m_b)$. **C** sends C^* as the challenge to **A**.

Phase 2: **A** issues more queries as in Phase 1. These queries may be asked adaptively as in Phase 1. The restrictions on **A** are that $\mathcal{A}$ should not query the decryption oracle with C^* as input and **A** should not query the private key corresponding to ID_T.

Guess: Finally, **A** outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

The adversary $\mathbf{A}'s$ advantage in attacking the identity based encryption scheme is the following function of the security parameter κ

$$Adv_{\mathbf{A}}(\kappa) = |Pr[b = b'] - \frac{1}{2}|$$

The Paillier Encryption Scheme (PE): Here we describe the basic Paillier encryption scheme. The security of the **PE** scheme is based on the composite n^{th}-residuosity class problem. This scheme was proved to be IND-CPA secure in the standard model. More details on the scheme can be found in [23].

Key generation(κ): Choose two large prime numbers p and q randomly and independently of each other such that $gcd(pq, (p-1)(q-1)) = 1$. Compute $n = pq$ and $\lambda = lcm(p-1, q-1)$. Let:

$$\mathbb{G} = \{x : x \in \mathbb{Z}_{n^2} : x^n \, mod \, n^2 = 1\} \tag{4}$$

Choose $g \in_R \mathbb{G}$. L is a function defined as $L(u) = \frac{u-1}{n}$. Here $u \in \mathbb{G}$ and $L : \mathbb{G} \rightarrow \mathbb{Z}$ is computed over integers. The public (encryption) key is $pk = \langle n, g \rangle$. The private (decryption) key is $sk = \langle \lambda, p, q \rangle$.

Encryption(m, pk): Let m be a message to be encrypted, where $m < n$. Choose a random integer r, where $r < n$. Compute the ciphertext as: $C = g^m r^n \mod n^2$.

Decryption(C, sk): To decrypt a ciphertext $C \in \mathbb{Z}^*_{n^2}$, compute

$$m = \frac{L(C^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n \tag{5}$$

3 Modified Paillier Encryption Scheme (MPE)

In this section, we suggest a modification to **PE** and prove its security. The difference between **PE** and **MPE** are:

- In **PE**, $g \in_R \mathbb{G}$ and note that by (5), discrete logarithm in the restricted group $\mathbb{G}$ is easy. In **MPE**, $g \in_R \mathbb{Z}^*_{n^2}$.
- In **MPE**, we provide (g, g^λ) as the public key. We need to show that giving access to the pair (g, g^λ) with $g \in \mathbb{Z}_{n^2}$ does not compromise the security properties. It should be noted that providing (g, g^λ) with $g \in \mathbb{G}$ in **PE** will lead to total break of **PE** because $\frac{L(g^\lambda \mod n^2)}{L(g)} \mod n = \lambda$. However in **MPE**, though we give $\langle g, g^\lambda \rangle$, $g \in \mathbb{Z}^*_{n^2}$ it does not lead to any security violation.

The modified scheme makes use of an RSA Instance Generator algorithm described below.
RSA-INSTANCE-GENERATOR(κ): This algorithm chooses two large prime numbers p and q randomly and independently such that $p' = (p-1)/2$ and $q' = (q-1)/2$ are also primes and $gcd(pq, (p-1)(q-1)) = 1$. Compute $n = pq$ and $\lambda = lcm(p-1, q-1)$. Output $\langle p, q, \lambda, n \rangle$.

λ-RSA-SAMPLER: This algorithm executes $\langle p, q, \lambda, n \rangle \leftarrow$RSA-INSTANCE-GENERATOR($\kappa$) and outputs a tuple $\langle n, g, g^\lambda \rangle$, where $g \in_R \mathbb{Z}^*_{n^2}$. We denote the output distribution by Δ_1. We remark that given only n, without knowing the factors of n, it is infeasible to produce triples of the form $\langle n, g, g^\lambda \rangle$ and hence Δ_1 is not polynomial time sample-able.

FAKE-λ-RSA-SAMPLER: This algorithm executes $\langle p, q, \lambda, n \rangle \leftarrow$RSA-INSTANCE-GENERATOR($\kappa$) and outputs a tuple $\langle n, g, h^\lambda \rangle$, where $g, h \in_R \mathbb{Z}^*_{n^2}$. We denote the output distribution by Δ_2. It should be noted that given only n, without knowing the factors of n, it is easy to sample the triple $\langle n, g, h^\lambda \rangle$. This is performed by just outputting a triple of the form $\langle n, g, 1 + \alpha n \rangle$ for some $\alpha < n$ and hence Δ_2 is polynomial time sample-able.

Note that in **MPE**, to generate the public keys, we need triples that follow the Δ_1 distribution but in the proof we use the triple distributed with respect to Δ_2. Therefore, we have to prove the following lemma.

Lemma 1. *Distinguishing the outputs of* λ-RSA-SAMPLER *and* FAKE-λ-RSA-SAMPLER *is as hard as distinguishing* n^{th} *- Residues from random elements in* $\mathbb{Z}^*_{n^2}$

Proof: We show that if there exists a distinguisher who can distinguish the two tuples $\langle g, g^\lambda \rangle$ and $\langle g, h^\lambda \rangle$ generated by the algorithms λ-RSA-SAMPLER and FAKE-λ-RSA-SAMPLER, then it is possible to construct an algorithm which can solve the n^{th} Power Residue Distinguishability problem.

Since $g, h \in_R \mathbb{Z}^*_{n^2}$, for some y_1 and y_2, we have $g = u^{x_1} y_1^n$ and $h = u^{x_2} y_2^n$ from (2). We can rewrite the tuples $\langle g, g^\lambda \rangle$ and $\langle g, h^\lambda \rangle$ as:

$$\langle g, g^\lambda \rangle = \langle u^{x_1} y_1^n, (u^{x_1} y_1^n)^\lambda \rangle = \langle u^{x_1} y_1^n, u^{x_1 \lambda} \rangle \tag{6}$$

and

$$\langle g, h^\lambda \rangle = \langle u^{x_1} y_1^n, (u^{x_2} y_2^n)^\lambda \rangle = \langle u^{x_1} y_1^n, u^{x_2 \lambda} \rangle \tag{7}$$

Consider that there is an algorithm **C** which generates samples of the two distributions Δ_1 and Δ_2 using samplers S_1 and S_2. Let $\{0,1\} \leftarrow \mathbf{A}(x, y)$ be an adversarial algorithm which is capable of distinguishing the tuples given in equations (6) and (7) following the distributions Δ_1 and Δ_2 respectively. Therefore we can write,

$$Pr[(\mathbf{A}(x,y) = 1) | (x,y) \leftarrow \Delta_i] = P_i \,,\ 1 \le i \le 2 \tag{8}$$

$$Adv(\mathbf{A}(x,y)) = |P_1 - P_2| \ge \epsilon \tag{9}$$

Now, we can construct an algorithm $\{0,1\} \leftarrow \mathbf{B}(\delta)$ that uses **A** as an oracle to distinguish an element δ, as a random element in $\mathbb{Z}^*_{n^2}$ or a n^{th} power residue in $\mathbb{Z}^*_{n^2}$, i.e, distinguish whether δ follows the distribution μ_1 or μ_2. Therefore we can write,

$$Pr[(\mathbf{B}(\delta) = 1) | \delta \leftarrow \mu_i] = P'_i,\ 1 \le i \le 2 \tag{10}$$

$$Adv(\mathbf{B}(\delta)) = |P'_1 - P'_2| \ge \epsilon \tag{11}$$

B receives a tuple $\langle n, g, g^\lambda \rangle \leftarrow \lambda$-RSA-Sampler, which follows distribution Δ_1, computes a new tuple $\langle g\delta, g^\lambda \rangle$ and sends it to **A** and returns the value $\mathbf{A}(g\delta, g^\lambda)$

Since $\langle g, g^\lambda \rangle$ follows the distribution Δ_1, $\langle g\delta, g^\lambda \rangle = \langle u^{x_1} y_1^n \delta, u^{x_1 \lambda} \rangle$. The output of **B** is correct with advantage ϵ due to the argument that follows:

- If δ follows the distribution μ_1 then, $\delta = u^{x'} y'^n$ (from (2)). Therefore, $\mathbf{A}(u^{x_1} y_1^n \delta, u^{x_1 \lambda}) = \mathbf{A}(u^{x_1} y_1^n u^{x'} y'^n, u^{x_1 \lambda}) = \mathbf{A}(u^{x_1 + x'}(y_1 y')^n, u^{x_1 \lambda})$, which clearly follows the distribution Δ_2.
- If δ follows the distribution μ_2 then, $\delta = y'^n$ (from (3)). Therefore, $\mathbf{A}(u^{x_1} y_1^n \delta, u^{x_1 \lambda}) = \mathbf{A}(u^{x_1} y_1^n y'^n, u^{x_1 \lambda}) = \mathbf{A}(u^{x_1}(y_1 y')^n, u^{x_1 \lambda})$, which clearly follows the distribution Δ_1.

This shows that distinguishing the outputs of λ-RSA-Sampler and Fake-λ-RSA-Sampler is as hard as distinguishing n^{th} - Residues from random elements in $\mathbb{Z}^*_{n^2}$. ■

3.1 The MPE Scheme

Key generation(κ): This algorithm executes $\langle n, g, g^\lambda \rangle \leftarrow \lambda$-RSA Sampler . The public (encryption) key is $pk = \langle n, g, g^\lambda \rangle$. The private (decryption) key is $sk = \langle \lambda, p, q \rangle$.

Encryption(m, pk): Let m be a message to be encrypted, where $m < n$. Choose $r \in_R \mathbb{Z}_n$ and compute the ciphertext $C = g^m g^{rn} \mod n^2$.

Decryption(C, sk): The decryption is same as in **PE** (Refer (5)).

Lemma 2. *The decryption algorithm of the modified encryption scheme works for a well formed ciphertext.*

Proof: The analysis in the proof of this lemma is similar to PE. In the modified scheme, we have $g \in_R \mathbb{Z}^*_{n^2}$. From (2), we can write $g = \psi(x, y) = u^x y^n$. From the encryption process of the modified scheme, we have

$C = g^m g^{rn} = (u^x y^n)^m (u^x y^n)^{rn} = u^{xm} u^{rxn} y^{n(m+rn)}$

The decryption of the ciphertext C is performed as in (5).

$$\frac{L(C^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n = \frac{L((u^{xm} u^{xrn} y^{n(m+rn)})^\lambda \mod n^2)}{L((u^x y^n)^\lambda \mod n^2)} \mod n$$
$$= \frac{L(u^{mx\lambda})}{L(u^{x\lambda})} \mod n, \text{ (since } y^{n\lambda} = u^{n\lambda} = 1 \mod n^2)$$

Since $u^{x\lambda}$ is an element of order n in $\mathbb{Z}^*_{n^2}$ and $(p-1)|\lambda$ and $(q-1)|\lambda$, due to Fermat's Little Theorem we have $u^{x\lambda} = 1 \, mod \, p$ and $u^{x\lambda} = 1 \, mod \, q$. Due to Chinese Reminder Theorem (CRT), we have $u^{x\lambda} = 1 \, mod \, n$. Therefore, $u^{x\lambda} = 1 + \beta n$ for some unknown β. Hence we have $u^{x\lambda} = 1 + \beta n$ and $(u^{x\lambda})^m = (1 + \beta n)^m$. Using Binomial Theorem, we have $u^{x\lambda m} = 1 + m\beta n \, mod \, n^2$. Hence,

$$\left[\frac{L(u^{xm\lambda} \, mod \, n^2)}{L(u^{x\lambda} \, mod \, n^2)}\right] \, mod \, n = \left[\frac{m\beta}{\beta}\right] \, mod \, n = m \, mod \, n, \left(since \, L(u) = \frac{u-1}{n}\right)$$

This proves that the decryption of a well-formed ciphertext outputs the correct message, when $m < n$. ■

3.2 Security Proof

Theorem 1. *The **MPE** scheme is IND-CPA secure.*

Proof: We prove that **MPE** scheme is IND-CPA secure. If there exists an adversarial algorithm who can distinguish the ciphertext of **MPE** scheme, then we show that it is possible to construct another algorithm that can distinguish the n^{th} power residue with the same advantage.

In order to show this we construct a fake **MPE** scheme (FMPE). The parameters of **FMPE** are constructed as follows:

Key generation(κ): This algorithm executes $\langle n, g, h^\lambda \rangle \leftarrow$ FAKE-λ-RSA Sampler . The public (encryption) key is $pk = \langle n, g, h^\lambda \rangle$. The private (decryption) key is $sk = \langle \lambda, p, q \rangle$. The encryption and decryption algorithm are same as **MPE**.

Lemma 3. *The public key of **MPE** $pk_{MPE} = \langle n, g, g^{\lambda} \rangle$ and **FMPE** $pk_{FMPE} = \langle n, g, h^{\lambda} \rangle$ are indistinguishable.*

Proof: The proof is similar to lemma 1. □

Lemma 4. *The ciphertexts of **MPE** and **FMPE** are indistinguishable.*

Proof: Let us consider that there are two worlds, one is the **MPE** world and the other is the **FMPE** world. Let **C** be an algorithm which takes a set of transcripts $\langle m_i, C_i \rangle$ for $i = 1$ to k as input, where m_i, C_i are the set of message/ciphertext pairs and distinguishes a fresh pair (m^*, C^*) as whether it is generated by **MPE** or **FMPE** . It should be noted that g^{λ} and h^{λ} are identical and are also not used for encrypting the messages in either **MPE** or **FMPE** respectively. Let $\pi_2(\Delta)$ represent the second component of the distribution Δ, we can write:

$$\pi_2(\Delta_1) \approx \pi_2(\Delta_2) \tag{12}$$

Where $\approx$ represents identical distribution. If **C** is capable of distinguishing (m^*, C^*) then it is possible to construct another algorithm **D** who can make use of **C** and distinguish the outputs of λ-RSA-Sampler and Fake-λ-RSA-Sampler which is as hard as distinguishing n^{th} - Residues from random elements in $\mathbb{Z}^*_{n^2}$ due to lemma 1.

□

Consider that there exists an adversarial algorithm **A** who is capable of distinguishing the ciphertext of **MPE**. We construct another algorithm **B** which is capable of deciding whether an element g_1 is n^{th} residue modulo n^2 (i.e. $g_1 = \psi(0, y)$ for some y or not). The construction follows:

B sets the public key $pk = pk_{FMPE} = \langle n, g, h^{\lambda} \rangle$ and gives them to **A**. Note that the pk given to **A** is the public key of **FMPE**, but **A** cannot distinguish pk from pk_{MPE} due to lemma 3.

Note: For a given $\langle n, g \rangle$ finding g^{λ} is hard without the knowledge of λ however finding h^{λ}, for some h can be done indirectly by choosing an $\alpha < n$ and computing $1 + \alpha n$. Note that $(1 + \alpha n) = h^{\lambda}$ for some h but we do not know h. Thus a pair of the form $\langle g, h^{\lambda} \rangle$ is generated by computing $\langle g, 1 + \alpha n \rangle$, for some $\alpha < n$.

A gives two messages m_0 and m_1 of equal length to **B**. **B** chooses $b \in_R \{0, 1\}$ and constructs $C^* = g^{m_b} g_1 \ mod \ n^2$ and sends it to **A**.

Now, there are two cases to be addressed:

- Case 1: The case where $g_1 \in_R \mathbb{Z}^*_{n^2}$. In this case $g_1 = \psi(x, y)$. Therefore, $C^* = \psi(m_b + x, y)$.
- Case 2: The case where g_1 is an n^{th} residue mod n^2. In this case $g_1 = \psi(0, y)$ and hence $C^* = \psi(m_b, y)$.

$$|Pr\left[\mathbf{A}(\psi(m_b + x), y)\right] - Pr\left[\mathbf{A}(\psi(m_b), y)\right]| \leq \epsilon \tag{13}$$

A sends b' as an educated guess for bit b to **B**. If **A** outputs $b = b'$ with a non-negligible probability ϵ over guessing, then **B** outputs that g_1 is an n^{th} residue modulo n^2 with the same probability. Else outputs g_1 is a random element in $\mathbb{Z}^*_{n^2}$. ■

4 A New Digital Signature Scheme

An IBE requires a digital signature scheme for extracting the secret keys for the users. The secret key of an IBE scheme could be viewed as a digital signature on the idebtity of the user using the master private key of the PKG. In this section, we propose a novel digital signature scheme whose security is based on the New Factoring Assumption presented in section 2.

4.1 The Scheme

KeyGen(κ): This algorithm is executed by the signer to generate the private key for signing and public key for verification. Let k be an integer such that the hash functions **G** and **H** defined below are collision resistant with k bit output and performing 2^k computations is hard in polynomial time. Typical value of $k = 160$. The signer performs the following to generate the signing/verification key pair:

- Choose two primes p and q with $|p| = |q| = \kappa$
- Compute $n = pq$ and n^2
- Choose $g \in_R \mathbb{Z}^*_{n^2}$ and $g < n$
- Compute $\lambda = lcm(p-1, q-1)$, $\phi(n) = (p-1)(q-1)$ and $\phi(n^2) = n(p-1)(q-1)$
- Choose two cryptographic hash function $\mathbf{G} : \{0,1\}^* \rightarrow \{0,1\}^k$ and $\mathbf{H} : \{0,1\}^* \rightarrow \{0,1\}^k$
- Choose $x_i, y_i \in_R \{0,1\}^k$, for $i = 0$ to k
- For $i = 0$ to k compute
  - $X_i = g^{x_i} \ mod \ n^2$ and
  - $Y_i = g^{y_i} \ mod \ n^2$

The public key $pk = \langle g, n, n^2, X_0, X_1, \ldots, X_k, Y_0, Y_1, \ldots, Y_k, \mathbf{G}, \mathbf{H} \rangle$. The private key is $sk = \langle p, q, \lambda, \phi(n) \rangle$.
Sign(κ, m, sk): To sign the message m using the private key sk, the signer performs the following:

- Choose $r_A, r_{A1}, r_{A2}, s_{A1}, s_{A2} \in_R \mathbb{Z}^*_n$
- Compute $G_m = \mathbf{G}(m) = g_1 g_2 \ldots g_k$, where $g_1 g_2 \ldots g_k$ is the binary of G_m and $H_m = \mathbf{H}(m) = h_1 h_2 \ldots h_k$, where $h_1 h_2 \ldots h_k$ is the binary of H_m. Let $I_m = \{i | 1 \leq i \leq k, g_i = 1\}$ and $J_m = \{i | 1 \leq i \leq k, h_i = 1\}$
- Compute $x_m = x_0 + \sum_{i \in I_m} x_i$ and $y_m = y_0 + \sum_{i \in J_m} y_i$
- Compute

$$d_{m1} = r_m \lambda + r_{m1} x_m + r_{m2} y_A \, mod \, \phi(n^2) \tag{14}$$

$$d_{m2} = r_{m1} + s_{m1} \lambda \, mod \, \phi(n^2) \tag{15}$$

$$d_{m3} = r_{m2} + s_{m2} \lambda \, mod \, \phi(n^2) \tag{16}$$

$$D_{m4} = (g^{r_m} g^{-x_m s_{m1}} g^{-y_m s_{m2}})^{\lambda} \, mod \, n^2 \tag{17}$$

The signature on m using the private key sk is $\sigma = \langle d_{m1}, d_{m2}, d_{m3}, D_{m4} \rangle$

Verify(κ, m, σ, pk): The signature $\sigma = \langle d_{m1}, d_{m2}, d_{m3}, D_{m4} \rangle$ on message m is verified using the public key pk by computing

$$X_m = \left[X_0 . \prod_{i \in I_m} X_i \right] \ mod \, n^2 \tag{18}$$

$$Y_m = \left[Y_0 . \prod_{i \in J_m} Y_i \right] \ mod \, n^2 \tag{19}$$

and checking whether:

$$D_{m4} \stackrel{?}{=} \left[\frac{g^{d_{m1}}}{X_m^{d_{m2}} Y_m^{d_{m3}}} \right] \ mod \ n^2 \tag{20}$$

$$(D_{m4})^n \ mod \ n^2 \stackrel{?}{=} 1 \tag{21}$$

4.2 Security Proof

Theorem 2. *The proposed signature scheme is EUF-CMA secure in the random oracle model, if the New Factoring Assumption holds.*

Proof: We show that if there exists an adversary who can break the EUF-CMA security of the signature scheme, then it is possible to break the New Factoring Assumption with almost the same probability as the attacker forging a signature.

KeyGen: **C** obtains the hard problem instance $\langle n, g, g^{\lambda} \rangle$ as input and sets the public key of the user in the following way:

- **C** takes n and g from the hard problem instance.

- Chooses $x_i, y_i \in_R \{0,1\}^k$, for $i = 0$ to k
- For $i = 0$ to k compute:
  - $X_i = g^{x_i} \ mod \ n^2$ and
  - $Y_i = g^{y_i} \ mod \ n^2$
- Chooses two cryptographic hash functions **G** and **H**
- **C** sets the public key $pk = \langle g, n, n^2, X_0, \ldots, X_k, Y_0, \ldots, Y_k, \mathbf{G}, \mathbf{H} \rangle$
- **C** uses the g^λ value obtained from the hard problem instance for the training phase.

Training Phase: **A** makes any number of signing queries to **C** by sending message m. **C** generates σ in the following way and sends it to **A**:

- Choose $d_{m1}, d_{m2}, d_{m3} \in_R |\phi(n)|$.
- Compute $G_m = \mathbf{G}(m) = g_1 g_2 \ldots g_k$, where $g_1 g_2 \ldots g_k$ is the binary of G_m and $H_m = \mathbf{H}(m) = h_1 h_2 \ldots h_k$, where $h_1 h_2 \ldots h_k$ is the binary of H_m. Let $I_m = \{i | 1 \leq i \leq k, g_i = 1\}$ and $J_m = \{i | 1 \leq i \leq k, h_i = 1\}$
- Compute $x_m = x_0 + \sum_{i \in I_m} x_i$ and $y_m = y_0 + \sum_{i \in J_m} y_i$
- Compute $D_{m4} = \left[\frac{(g^\lambda)^{d_{m1}}}{(g^\lambda)^{x_m d_{A2}} (g^\lambda)^{y_m d_{A3}}} \right] \ mod \ n^2$

Forgery: **A** forges a new message/signature pair (m^*, σ^*) and sends it to **C**. The forgery is of the following form:

$d^*_{m1} = r_m \lambda + r_{m1} x_m + r_{m2} y_A \, mod \, \phi(n^2)$
$d^*_{m2} = r_{m1} + s_{m1} \lambda \, mod \, \phi(n^2)$
$d^*_{m3} = r_{m2} + s_{m2} \lambda \, mod \, \phi(n^2)$
$D^*_{m4} = (g^{r_m} g^{-x_m s_{m1}} g^{-y_m s_{m2}})^\lambda \, mod \, n^2$

C performs the following to obtain the solution to the hard problem.

- Compute $D^* = \left[\frac{(g^\lambda)^{d^*_{m1}}}{(g^\lambda)^{x_m d^*_{A2}} (g^\lambda)^{y_m d^*_{A3}}} \right] \ mod \ n^2$
- Computes $\lambda = \left[\frac{L(D^*)}{L(D^*_{m4})} \right] \ mod \ n$

5 The New IBE Scheme

In this section, we propose the new basic IBE scheme. The Key Extract algorithm is a novel contribution. Here the encryption algorithm is based on **PE**. We use a Waters type hash function [32] to extract the private key of the users. The scheme is proved to be secure by showing a polynomial time reduction to **MPE**. In the modified version, $g^\lambda \, mod \, n^2$ is also made public along with g. We have supported this modification with a proof showing that revealing $g^\lambda \, mod \, n^2$ does not affect the security of **MPE**. It should be noticed that g^λ is not used in the scheme for any public computation but it plays a crucial role in the security proof.

5.1 The Scheme

Setup(κ): This algorithm is executed by the PKG to generate the public system parameters and the master private key of the IBE scheme. Let k be an integer such that the hash function **G** and **H** defined below are collision resistant with k bit output and performing 2^k computations is hard in polynomial time. Typical value of $k = 160$. PKG performs the following to generate the public parameters and the master private key of the system:

- Choose two primes p and q with $|p| = |q| = \kappa$
- Compute $n = pq$ and n^2
- Choose $g \in_R \mathbb{Z}^*_{n^2}$ and $g < n$
- Compute $\lambda = lcm(p-1, q-1)$, $\phi(n) = (p-1)(q-1)$ and $\phi(n^2) = n(p-1)(q-1)$

- Choose two cryptographic hash function $\mathbf{G} : \{0,1\}^* \rightarrow \{0,1\}^k$ and $\mathbf{H} : \{0,1\}^* \rightarrow \{0,1\}^k$
- Choose $x_i, y_i \in_R \{0,1\}^k$, for $i = 0$ to k
- For $i = 0$ to k compute
  - $X_i = g^{x_i} \ mod \ n^2$ and
  - $Y_i = g^{y_i} \ mod \ n^2$

The public system parameter $params_{IBE} = \langle g, n, n^2, X_0, X_1, \ldots, X_k, Y_0, Y_1, \ldots, Y_k, \mathbf{G}, \mathbf{H} \rangle$. The master private key is $msk_{IBE} = \langle p, q, \lambda, \phi(n) \rangle$.

Key Extract(ID_A): The user A submits his identity ID_A to the PKG and requests for the corresponding private key. The PKG performs the following to provide the private key to the user:

- Choose $r_A, r_{A1}, r_{A2}, s_{A1}, s_{A2} \in_R \mathbb{Z}_n^*$
- Compute $G_A = \mathbf{G}(ID_A) = g_1 g_2 \ldots g_k$, where $g_1 g_2 \ldots g_k$ is the binary of G_A and $H_A = \mathbf{H}(ID_A) = h_1 h_2 \ldots h_k$, where $h_1 h_2 \ldots h_k$ is the binary of H_A. Let $I_A = \{i | 1 \leq i \leq k, g_i = 1\}$ and $J_A = \{i | 1 \leq i \leq k, h_i = 1\}$
- Compute $x_A = x_0 + \sum_{i \in I_A} x_i$ and $y_A = y_0 + \sum_{i \in J_A} y_i$
- Compute

$$d_{A1} = r_A \lambda + r_{A1} x_A + r_{A2} y_A \, mod \, \phi(n^2) \tag{22}$$

$$d_{A2} = r_{A1} + s_{A1} \lambda \, mod \, \phi(n^2) \tag{23}$$

$$d_{A3} = r_{A2} + s_{A2} \lambda \, mod \, \phi(n^2) \tag{24}$$

$$D_{A4} = (g^{r_A} g^{-x_A s_{A1}} g^{-y_A s_{A2}})^{\lambda} \, mod \, n^2 \tag{25}$$

The private key of the user with identity ID_A, $SK_A = \langle d_{A1}, d_{A2}, d_{A3}, D_{A4} \rangle$ is sent to the user through a secure channel. To verify the correctness of the private key SK_A, compute

$$X_A = \left[X_0 . \prod_{i \in I_A} X_i \right] \, mod \, n^2 \tag{26}$$

$$Y_A = \left[Y_0 . \prod_{i \in J_A} Y_i \right] \, mod \, n^2 \tag{27}$$

Check whether:

$$D_{A4} \stackrel{?}{=} \left[\frac{g^{d_{A1}}}{X_A^{d_{A2}} Y_A^{d_{A3}}} \right] \, mod \ n^2 \tag{28}$$

$$(D_{A4})^n \, mod \ n^2 \stackrel{?}{=} 1 \tag{29}$$

Lemma 5. *If the private key SK_A is correctly generated as per the key extract algorithm, the values in SK_A satisfies (28)*

We show that the above equality holds if the private key is correctly generated as per the key extract algorithm.

$$\begin{aligned}
RHS &= \left[\frac{g^{d_{A1}}}{X_A^{d_{A2}} Y_A^{d_{A3}}} \right] \, mod \ n^2 = \left[\frac{g^{r_A \lambda + r_{A1} x_A + r_{A2} y_A}}{X_A^{r_{A1} + s_{A1} \lambda} Y_A^{r_{A2} + s_{A2} \lambda}} \right] \, mod \ n^2 \\
&= \left[\frac{g^{r_A \lambda + r_{A1} x_A + r_{A2} y_A}}{(g^{x_A})^{r_{A1} + s_{A1} \lambda} (g^{y_A})^{r_{A2} + s_{A2} \lambda}} \right] \, mod \ n^2 \\
&= \left[\frac{g^{r_A \lambda} g^{r_{A1} x_A} g^{r_{A2} y_A}}{(g^{r_{A1} x_A})(g^{x_A s_{A1} \lambda})(g^{r_{A2} y_A})(g^{y_A s_{A2} \lambda})} \right] \, mod \ n^2 \\
&= \left[\frac{g^{r_A \lambda}}{(g^{x_A s_{A1} \lambda})(g^{y_A s_{A2} \lambda})} \right] \, mod \ n^2 \\
&= g^{r_A \lambda} g^{-x_A s_{A1} \lambda} g^{-y_A s_{A2} \lambda} \, mod \ n^2 \\
&= (g^{r_A} g^{-x_A s_{A1}} g^{-y_A s_{A2}})^{\lambda} \, mod \ n^2 \\
&= D_{A4} = LHS
\end{aligned}$$

■

Encryption(m, ID_A)**:** In order to encrypt a message m $(m < n)$ with the public key (identity) ID_A of the receiver A, the sender performs the following:

- Choose $r \in_R \mathbb{Z}_n$
- Compute $H_A = \mathbf{H}(ID_A)$
- Compute $X_A = \left[X_0 . \prod_{i \in I_A} X_i\right] \ mod\ n^2$ and $Y_A = \left[Y_0 . \prod_{i \in J_A} Y_i\right] \ mod\ n^2$
- Compute

$$C_1 = g^m g^{rn} \, mod \, n^2 \tag{30}$$
$$C_2 = X_A^m X_A^{rn} \, mod \, n^2 \tag{31}$$
$$C_3 = Y_A^m Y_A^{rn} \, mod \, n^2 \tag{32}$$

The ciphertext is $C = \langle C_1, C_2, C_3 \rangle$.

Decryption(C, SK_A)**:** Let $L(x)$ be defined as $L(x) = \left[\frac{x-1}{n}\right]$. The receiver with identity ID_A and private key $SK_A = \langle d_{A1}, d_{A2}, d_{A3}, D_{A4} \rangle$ decrypts the ciphertext C by performing the following:

- Compute $\hat{C} = \left(\frac{C_1^{d_{A1}}}{C_2^{d_{A2}} C_3^{d_{A3}}}\right) mod\, n^2$
- Compute $m = \left[\frac{L\left(\hat{C}\right)}{L(D_{A4})}\right] mod\ n$

Lemma 6. *The Decryption Algorithm outputs the correct message for a well-formed ciphertext.*

Proof: From equations (22) and (30) we have,

$$C_1^{d_{A1}} \, mod \, n^2 = (g^m g^{rn})^{r_A \lambda + x_A r_{A1} + y_A r_{A2}} \, mod \, n^2 \tag{33}$$

From equations (23), (24), (31) and (32) we have,

$$\begin{aligned} C_2^{d_{A2}} C_3^{d_{A3}} \, mod \, n^2 &= (X_A^m X_A^{rn})^{(r_{A1} + s_{A1}\lambda)} (Y_A^m Y_A^{rn})^{(r_{A2} + s_{A2}\lambda)} \, mod \, n^2 \\ &= (g^{x_A m} g^{x_A rn})^{(r_{A1} + s_{A1}\lambda)} (g^{y_A m} g^{y_A rn})^{(r_{A2} + s_{A2}\lambda)} \, mod \, n^2 \\ &= (g^{x_A m} g^{x_A rn})^{r_{A1}} (g^{x_A m} g^{x_A rn})^{s_{A1}\lambda} (g^{y_A m} g^{y_A rn})^{r_{A2}} \\ &\quad (g^{y_A m} g^{y_A rn})^{s_{A2}\lambda} \, mod \, n^2 \end{aligned}$$

Therefore,

$$\begin{aligned} C_2^{d_{A2}} C_3^{d_{A3}} \, mod \, n^2 = &(g^{x_A m} g^{x_A rn})^{r_{A1}} (g^{x_A m} g^{x_A rn})^{s_{A1}\lambda} (g^{y_A m} g^{y_A rn})^{r_{A2}} \\ &(g^{y_A m} g^{y_A rn})^{s_{A2}\lambda} \, mod \, n^2 \end{aligned} \tag{34}$$

From equations (33) and (34) we have,

$$\begin{aligned} \hat{C} &= \left(\frac{C_1^{d_{A1}}}{C_2^{d_{A2}} C_3^{d_{A3}}}\right) mod \, n^2 \\ &= \frac{(g^m g^{rn})^{r_A \lambda + x_A r_{A1} + y_A r_{A2}}}{(g^{x_A m} g^{x_A rn})^{r_{A1}} (g^{x_A m} g^{x_A rn})^{s_{A1}\lambda} (g^{y_A m} g^{y_A rn})^{r_{A2}} (g^{y_A m} g^{y_A rn})^{s_{A2}\lambda}} \, mod \, n^2 \\ &= \frac{(g^m g^{rn})^{r_A \lambda} (g^m g^{rn})^{x_A r_{A1}} (g^m g^{rn})^{y_A r_{A2}}}{(g^m g^{rn})^{x_A r_{A1}} (g^{x_A m} g^{x_A rn})^{s_{A1}\lambda} (g^m g^{rn})^{y_A r_{A2}} (g^{y_A m} g^{y_A rn})^{s_{A2}\lambda}} \, mod \, n^2 \\ &= (g^m g^{rn})^{r_A \lambda} (g^{x_A m} g^{x_A rn})^{-s_{A1}\lambda} (g^{y_A m} g^{y_A rn})^{-s_{A2}\lambda} \, mod \, n^2 \\ &= (g^m g^{rn})^{r_A \lambda - x_A s_{A1} \lambda - y_A s_{A2} \lambda} \, mod \, n^2 \\ &= (g^m)^{r_A \lambda - x_A s_{A1} \lambda - y_A s_{A2} \lambda} (g^{rn})^{r_A \lambda - x_A s_{A1} \lambda - y_A s_{A2} \lambda} \, mod \, n^2 \\ &= (g^m)^{r_A \lambda - x_A s_{A1} \lambda - y_A s_{A2} \lambda} (g^{rn})^{\lambda (r_A - x_A s_{A1} - y_A s_{A2})} \, mod \, n^2 \\ &= (g^m)^{r_A \lambda - x_A s_{A1} \lambda - y_A s_{A2} \lambda} \, mod \, n^2, \text{ Since } (g^{rn})^{\lambda} = 1 \ mod \ n^2 \\ &= (g^{m r_A} g^{-m x_A s_{A1}} g^{-m y_A s_{A2}})^{\lambda} \, mod \, n^2 \\ &= (g^{r_A} g^{-x_A s_{A1}} g^{-y_A s_{A2}})^{m\lambda} \, mod \, n^2 \end{aligned}$$

Thus, we have

$$\hat{C} = (g^{r_A} g^{-x_A s_{A1}} g^{-y_A s_{A2}})^{m\lambda} \, mod \, n^2 \tag{35}$$

The private key component D_{A4} can be written as $D_{A4} = (g^{r_A} g^{-x_A s_{A1}} g^{-y_A s_{A2}})^{\lambda} \; mod \; n^2 = h^{\lambda} \, mod \, n^2$, where $h = (g^{r_A} g^{-x_A s_{A1}} g^{-y_A s_{A2}}) \in \mathbb{Z}^*_{n^2}$. Hence, we can write $\hat{C} = h^{m\lambda}$, where $h^{\lambda} \; mod \; n^2$ is an element of order n in $\mathbb{Z}^*_{n^2}$.

Now, $\left[\dfrac{L\left(\hat{C}\right)}{L(D_{A4})}\right] \; mod \; n = \left[\dfrac{L(h^{m\lambda})}{L(h^{\lambda})}\right] \; mod \; n.$

Since h^{λ} is an element of order n in $\mathbb{Z}^*_{n^2}$ and $(p-1)|\lambda$ and $(q-1)|\lambda$, due to Fermat's Little Theorem we have $h^{\lambda} = 1 \, mod \, p$ and $h^{\lambda} = 1 \, mod \, q$. Due to Chinese Reminder Theorem (CRT), we have $h^{\lambda} = 1 \, mod \, n$. Therefore, $h^{\lambda} = 1 + \beta n$ for some unknown β. Hence we have $h^{\lambda} = 1 + \beta n$ and $(h^{\lambda})^m = (1 + \beta n)^m$. Using Binomial Theorem, we have $h^{\lambda m} = 1 + m\beta n \, mod \, n^2$. Hence,

$\left[\dfrac{L(h^{x\lambda} \, mod \, n^2)}{L(h^{\lambda} \, mod \, n^2)}\right] \; mod \; n = \left[\dfrac{m\beta}{\beta}\right] \; mod \; n = m \; mod \; n, \; \left(since \, L(u) = \dfrac{u-1}{n}\right)$

This proves that the decryption of a well-formed ciphertext outputs the correct message when $m < n$. ■

5.2 Security Proof

Theorem 3. *The proposed IBE scheme is IND-CPA secure in the random oracle model, if **MPE** is IND-CPA secure.*

Proof: We show that if there exists an adversary who can break the IND-CPA security of our IBE scheme, then it is possible to construct another adversary who interacts with the former one and breaks the IND-CPA security of **MPE**.

- Let **A** be the adversary of the IBE scheme.
- Let **C** be the challenger of the IBE scheme and also act as an adversary to the underlying **MPE**.
- Let **B** be the challenger to **MPE**.

The figure shows the interaction between **A**, **B** and **C**. **B** generates the system parameters of **MPE** and challenges **C** to break the IND-CPA security of the scheme. **C** in turn makes use of **A**, who claims to be capable of breaking the IND-CPA security of the IBE scheme to do so. The details of the game follows.

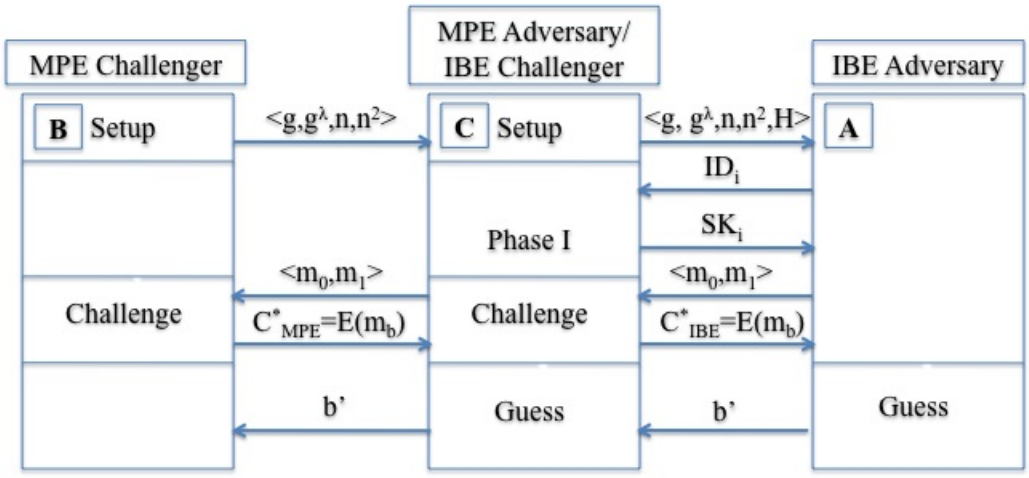

Fig. 1. Reduction

Setup: B gives the public key $pk_{\mathbf{MPE}} = \langle n, n^2, Enc_{\mathbf{MPE}}, g, g^{\lambda}\rangle$ of **MPE** to **C**. Since $|p| = |q| = \kappa$, $|n| = 2\kappa$, $|n^2| = 4\kappa$, $|\phi(n)| = |(p-1)(q-1)| = 2\kappa$ and $|\phi(n^2)| = |n\phi(n)| = 4\kappa$.

C generates the public parameters $params_{IBE}$ of the IBE scheme as follows:

- Compute $l = 2q_E$, where q_E is the maximum number of *Key Extract* oracle queries. Let $kl < n$ for a sufficiently large k (k corresponds to the size of the output of the function **H**).
- Choose $x_0 \in_R \{-kl, \ldots, 0\}$.
- Choose $\hat{x}_i, \in_R \{0, \ldots, l\}$, for $i = 1, \ldots, k$.
- Choose $\bar{x}_i \in_R \{0,1\}^k$, for $i = 1, \ldots, k$.
- Choose $y_i \in_R \{0,1\}^k$, for $i = 1, \ldots, k$.
  - Compute $X_i = g^{\lambda\hat{x}_i} g^{-\bar{x}_i}$ and $Y_i = g^{y_i}$. Thus, $x_i = \lambda\hat{x}_i - \bar{x}_i$ (implicitly).
  - Define functions $\hat{F}(m) \stackrel{Def}{=} x_0 + \sum_{i=1}^{k} m_i\hat{x}_i$ and $\hat{G}(m) \stackrel{Def}{=} \sum_{i=1}^{k} m_i\bar{x}_i$, where m_i represents the i^{th} bit of the input m.

C provides $params_{IBE} = \langle n, n^2, g, X_i, Y_i, \mathbf{G}, \mathbf{H}\rangle$, for $i = 0$ to k, to **A** and provides access to the key extract oracle.

Phase I: In this phase, **C** provides oracle access to the key extract algorithm of the IBE scheme to **A**. The description of the Key Extract oracle follows:

Key Extract(ID_A) Oracle: **A** gives the identity (ID_A) of a user as input to this oracle and **C** gives the private key corresponding to the identity ID_A to **A**. In order to respond to this query consistently, **A** performs the following:

- Compute $H_A = \mathbf{H}(ID_A) = h_1 h_2 \ldots h_k$, where h_i's are bits in the binary representation of H_A and $I_A = \{i | 1 \leq i \leq k, h_i = 1\}$. Let $|I_A|$ (the total number of h_i's that are 1) be denoted by α_A.
- Compute $\hat{x}_A = \sum_{i \in I_A} \hat{x}_i$, $\bar{x}_A = \sum_{i \in I_A} \bar{x}_i$ and $y_A = \sum_{i \in I_A} y_i$, where all computations are performed over integers.
- If $\hat{F}(H_A) = 0$ then *abort*.
- Else,
  - Choose $d_{A1}, d_{A2}, d_{A3} \in_R \{0,1\}^{4\kappa}$.
  - Find integer solutions for the following set of linear equations for the unknown values r, s and t.

$$d_{A1} = y_A s - \bar{x}_A r \tag{36}$$

$$d_{A2} = y_A t + r \tag{37}$$

$$d_{A3} = \bar{x}_A t + s \tag{38}$$

  - Compute

$$D_{A4} = g^{-\lambda\hat{x}_A y_A t} g^{-\lambda\hat{x}_A r} \, mod \, n^2 \tag{39}$$

- If $\hat{F}(H_A) = 0$ then *abort*.
- Else, output $\langle d_{a1}, d_{A2}, d_{A3}, D_{A4}\rangle$ as the private key corresponding to ID_A.

Correctness of the Key Extract Oracle: According to (28), a valid private key should satisfy: $D_{A4} \stackrel{?}{=} \left[\frac{g^{d_{A1}}}{X_A^{d_{A2}} Y_A^{d_{A3}}}\right]$.

The simulated private key also passes this verification as shown below:

$$\begin{aligned} RHS &= \left[\frac{g^{d_{A1}}}{X_A^{d_{A2}} Y_A^{d_{A3}}}\right] = \frac{g^{(y_A s - \bar{x}_A r)}}{(g^{x_A})^{(y_A t + r)} (g^{y_A})^{(\bar{x}_A t + s)}} \\ &= \frac{g^{(y_A s - \bar{x}_A r)}}{(g^{\lambda\hat{x}_A - \bar{x}_A})^{(y_A t + r)} (g^{y_A})^{(\bar{x}_A t + s)}} (\text{Since } x_A = \lambda\hat{x}_A - \bar{x}_A) \\ &= \frac{g^{y_A s} g^{-\bar{x}_A r}}{g^{\lambda\hat{x}_A y_A t} g^{-\bar{x}_A y_A t} g^{\lambda\hat{x}_A r} g^{-\bar{x}_A r} g^{\bar{x}_A y_A t} g^{y_A s}} \\ &= g^{-\lambda\hat{x}_A y_A t} g^{-\lambda\hat{x}_A r} = D_{A4} = LHS \end{aligned}$$

Challenge: At the end of Phase I, **A** gives two messages m_0 and m_1 of equal length to **C**. It also gives an identity ID_T as the target identity on which **A** would like to be challenged. **C** submits the messages m_0 and m_1 received from **A** to **B**. **B** chooses a random bit $b \in_R \{0,1\}$ computes $C^*_{\mathbf{MPE}} = g^{m_b} r^n \bmod n^2$ and gives it back to **C**. **C** computes the challenge ciphertext with respect to the target identity (ID_T for which $\hat{F}(H_T) = 0$) as follows:

- Set $C_1^* = C^*$
- Compute $C_2^* = (C^*)^{x_T} \, mod \, n^2$. It should be noted that **C** will be able to compute x_T, because when $\hat{F}(H_T) = 0$, $x_T = \sum_{i \in I_A} \bar{x}_T$, where **C** knows all $\bar{x}_i$'s. Further note that if $\hat{F}(H_T) \neq 0$, $x_T = \lambda \sum_{i \in I_A} \hat{x}_T - \sum_{i \in I_A} \bar{x}_T$ but **C** does not know λ and cannot compute x_T.
- Compute $C_3^* = (C^*)^{y_T} \, mod \, n^2$

C gives $C^*_{IBE} = \langle C_1^*, C_2^*, C_3^* \rangle$ to **A** as the challenge ciphertext of the IBE scheme.

Guess: In this phase, **A** outputs b' to **C** as the guess for the IBE scheme. Now, **C** outputs b' to **B** as the guess to **MPE**. ■

6 CCA Security in the Random Oracle Model

We now propose a CCA secure IBE scheme based on the basic CPA secure IBE scheme presented in the previous section. The scheme is proved to be secure in the random oracle model by showing a polynomial time reduction to **MPE**. We make use of the well known Fujisaki-Okamoto transformation to achieve CCA security [14].

6.1 The Scheme

Setup: This algorithm is similar to that of the setup algorithm in section 5.1. Apart from the parameters of the CPA secure scheme, this scheme has two more hash functions, defined as, $\mathbf{H}_1 : \{0,1\}^{|m|} \times \{0,1\}^k \rightarrow \{0,1\}^k$ and $\mathbf{H}_2 : \{0,1\}^k \times \mathbb{Z}^*_{n^2} \times \mathbb{Z}^*_{n^2} \times \mathbb{Z}^*_{n^2} \rightarrow \{0,1\}^{|m|}$.

The public parameters $params_{IBE} = \langle g, n, n^2, X_0, X_1, \ldots, X_k, Y_0, Y_1, \ldots, Y_k, \mathbf{H}, \mathbf{H}_1, \mathbf{H}_2 \rangle$. The master private key is $msk_{IBE} = \langle p, q, \lambda, \phi(n) \rangle$.

Key Extract(ID_A): This algorithm is similar to that of the key extract algorithm in section 5.1. The private key of the user with identity ID_A, $SK_A = \langle d_{A1}, d_{A2}, d_{A3}, D_{A4} \rangle$.

Encryption(m, ID_A): In order to encrypt a message m ($m < n$) with the public key (identity) ID_A, perform the following:

- Choose $s \in_R \{0,1\}^k$
- Compute $r = \mathbf{H}_1(m \| s)$
- Compute $H_A = \mathbf{H}(ID_A) = h_1 h_2 \ldots h_k$, where $h_1 h_2 \ldots h_k$ is the binary of H_A. Let $I_A = \{i | 1 \leq i \leq k, h_i = 1\}$
- Compute $X_A = \left[X_0 \prod_{i \in I_A} X_i \right] \, mod \, n^2$ and $Y_A = \left[Y_0 \prod_{i \in I_A} Y_i \right] \, mod \, n^2$
- Compute

$$C_1 = g^s g^{rn} \, mod \, n^2 \tag{40}$$

$$C_2 = X_A^s X_A^{rn} \, mod \, n^2 \tag{41}$$

$$C_3 = Y_A^s Y_A^{rn} \, mod \, n^2 \tag{42}$$

$$C_4 = m \oplus \mathbf{H}_2(s, C_1, C_2, C_3) \, mod \, n^2 \tag{43}$$

The ciphertext is $C = \langle C_1, C_2, C_3, C_4 \rangle$.

Decryption(C, SK_A): The receiver with identity ID_A and private key $SK_A = \langle d_{A1}, d_{A2}, d_{A3}, D_{A4} \rangle$ decrypts the ciphertext C by performing the following decryption process:

- Compute $\hat{C} = \left(\frac{C_1^{d_{A1}}}{C_2^{d_{A2}} C_3^{d_{A3}}} \right) \, mod \, n^2$
- Compute $s' = \left[\frac{L\left(\hat{C}\right)}{L(D_{A3})} \right] \, mod \, n$
- Compute $m' = C_4 \oplus \mathbf{H}_2(s', C_1, C_2, C_3)$
- Check whether:

$$C_1 \stackrel{?}{=} g^{s'} g^{\mathbf{H}_1(m'\|s')n} \, mod \, n^2 \tag{44}$$

$$C_2 \stackrel{?}{=} X_A^{s'} X_A^{\mathbf{H}_1(m'\|s')n} \, mod \, n^2 \tag{45}$$

$$C_3 \stackrel{?}{=} Y_A^{s'} Y_A^{\mathbf{H}_1(m'\|s')n} \, mod \, n^2 \tag{46}$$

6.2 Security Proof

Theorem 4. *The identity based encryption scheme is IND-CCA secure in the random oracle model if **MPE** is IND-CPA secure.*

Proof: Similar to the proof of the IND-CPA secure scheme, we show that, if there exists an adversary who can break the IND-CCA security of our IBE scheme, then it is possible to construct another adversary who interacts with the former one and breaks the IND-CPA security of **MPE**. Let **A** be the adversary of the IBE scheme. Let **C** be the challenger to the IBE and adversary to the underlying Paillier cryptosystem. Let **B** be the challenger to the Paillier Cryptosystem.

Setup: The setup phase is similar to that of the setup in theorem 3. The hash functions $\mathbf{H}, \mathbf{H}_1$ and $\mathbf{H}_2$ are defined as random oracles $\mathcal{O}_{\mathbf{H}}(.)$, $\mathcal{O}_{\mathbf{H}_1}(.)$ and $\mathcal{O}_{\mathbf{H}_2}(.)$.

C gives **A** the public parameters $params_{IBE} = \langle n, n^2, g, X_i, Y_i \rangle$ and oracle access to $\mathcal{O}_{\mathbf{H}}(.)$, $\mathcal{O}_{\mathbf{H}_1}(.)$, $\mathcal{O}_{\mathbf{H}_2}(.)$ oracles.

Phase I: In this phase, **C** provides oracle access to $\mathcal{O}_{\mathbf{H}}(.)$, $\mathcal{O}_{\mathbf{H}_1}(.)$, $\mathcal{O}_{\mathbf{H}_2}(.)$ oracles, the key extract oracle and the decryption oracle. **C** maintains three lists $L_{\mathbf{H}}$, $L_{\mathbf{H}_1}$ and $L_{\mathbf{H}_2}$ to maintain consistency in responding to the queries. The description of these oracles follow:

$\mathcal{O}_{\mathbf{H}}(ID_i \in \{0,1\}^*)$**:** If a tuple of the form $\langle ID_i, H_i \rangle$ exists in the list $L_{\mathbf{H}}$ then return H_i. If a tuple of this form does not exist then, choose $H_i \in_R \{0,1\}^k$, store the tuple $\langle ID_i, H_i \rangle$ in the list $L_{\mathbf{H}}$ and return H_i to **A**.

$\mathcal{O}_{\mathbf{H}_1}(m||s)$**:** To respond to this query, **C** checks whether a tuple of the form $\langle m||s, h_1 \rangle$ exists in the list $L_{\mathbf{H}_1}$. If a tuple of this form exists, **C** returns the corresponding h_1, else chooses $h_1 \in_R \{0,1\}^k$, adds the tuple $\langle m||s, h_1 \rangle$ to the list $L_{\mathbf{H}_1}$ and returns h_1 to **A**.

$\mathcal{O}_{\mathbf{H}_2}(s, C_1, C_2, C_3)$**:** To respond to this query, **C** checks whether a tuple of the form $\langle s, C_1, C_2, C_3, h_2 \rangle$ exists in the list $L_{\mathbf{H}_2}$. If a tuple of this form exists, **C** returns the corresponding h_2, else chooses $h_2 \in_R \{0,1\}^{|m|}$, adds the tuple $\langle s, C_1, C_2, C_3, h_2 \rangle$ to the list $L_{\mathbf{H}_2}$ and returns h_2 to **A**.

Key Extract(ID_A) Oracle: The description of this oracle is same as the key extract oracle in theorem 3.

$\mathcal{O}_{Decryption}(C, ID_i)$**:** **C** performs the following to decrypt the ciphertext $C = \langle C_1, C_2, C_3, C_4 \rangle$:

If $\hat{F}(H_i) \neq 0$ then **C** performs the decryption as per the decryption algorithm. If $\hat{F}(H_i) = 0$, **C** performs the following:

- Retrieve the tuple of the form $\langle s, C_1, C_2, C_3, h_2 \rangle$ from list $L_{\mathbf{H}_2}$. If a tuple does not exist, then return $\perp$ (to represent invalid ciphertext). If it exists then choose the corresponding s.
- Check whether a tuple of the form $\langle m||s, h_1 \rangle$ exist in list $L_{\mathbf{H}_1}$. If a tuple does not exist, then return $\perp$, else retrieve the corresponding m and h_1, and perform the checks defined in equations (44), (45) and (46).
- If all the checks hold good then return the m retrieved in the previous step as the decryption of C.

Challenge: At the end of the Phase I, **A** gives two messages m_0 and m_1 of equal length to **C** and an identity ID_T, **C** aborts the game if $\hat{F}(H_T) \neq 0$. **C** chooses $s_0, s_1 \in_R \{0,1\}^k$ and gives them to **B**. **B** chooses a random bit $b \in_R \{0,1\}$, computes $C^*_{\mathbf{MPE}} = g^{s_b} g^{rn} \bmod n^2$ and gives it back to **C**. **C** computes the challenge ciphertext with respect to the target identity (ID_T for which $\hat{F}(H_T) = 0$, where $H_T = \mathbf{H}(ID_T)$) as follows:

- Set $C^*_1 = C^*_{\mathbf{MPE}}$
- Compute $C^*_2 = (C^*_{\mathbf{MPE}})^{x_T} \, mod \, n^2$
- Compute $C^*_3 = (C^*_{\mathbf{MPE}})^{y_T} \, mod \, n^2$
- Choose a random bit $b \in_R \{0,1\}$. Add the tuple $\langle m_0 \| s_0, - \rangle$, $\langle m_0 \| s_1, - \rangle$, $\langle m_1 \| s_0, - \rangle$ and $\langle m_1 \| s_1, - \rangle$ to the $L_{\mathbf{H}_1}$ list.
- Choose $h^*_2 \in_R \{0,1\}^{|m|}$ and add the tuple $\langle -, C^*_1, C^*_2, C^*_3, h^*_2 \rangle$ to the $L_{\mathbf{H}_2}$ list. The first entry is marked with $'-'$ because, **C** does not know whether s_0 or s_1 is encrypted in $C^*_{\mathbf{MPE}}$
- Compute $C^*_4 = m_b \oplus h^*_2$

C gives $C^*_{IBE} = \langle C^*_1, C^*_2, C^*_3, C^*_4 \rangle$ to **A** as the challenge ciphertext of the IBE scheme.

Phase II: **A** performs the second phase of interaction, where it makes polynomial number of queries to the oracles provided by **C** with the following condition:

- **A** should not query the $\mathcal{O}_{Decryption}$ oracle with C^*_{IBE} as input.
- **A** should not query the private key SK_T corresponding to the identity ID_T.

All the oracle simulations are similar to Phase I.

Guess: In this phase, **A** outputs b' to **C** as the guess for the IBE scheme. Now, **C** performs the following to output a guess for the bit b.

- Find out the tuple of the form $\langle s_b, C^*_1, C^*_2, C^*_3, h^*_2 \rangle$ from the list $L_{\mathbf{H}_2}$.
- Check whether a query with input $m_{b'} \| s_b$ or $m_{\bar{b'}} \| s_b$ was asked to the $\mathcal{O}_{\mathbf{H}_1}$ oracle during the Phase II.
- If s_b appears in list $L_{\mathbf{H}_2}$ and a query was made as mentioned in the previous step then output b as the guess for **MPE**. ■

7 Conclusion

In this paper, we have shown the existence of practical, space and time efficient identity based encryption without pairing which does not involve a security mediator. The IBE scheme is based on the Paillier encryption scheme and hence theoretically based on the Composite $n^{th}-$Residuosity Assumption. The key generation is the novelty of the scheme and the encryption is based on the newly proposed modified Paillier encryption scheme (**MPE**). We have proposed two IBE schemes where the first scheme is CPA secure and the second one is CCA secure. We have proved both the schemes in the random oracle model.

References

1. Nuttapong Attrapadung, Jun Furukawa, Takeshi Gomi, Goichiro Hanaoka, Hideki Imai, and Rui Zhang. Efficient identity-based encryption with tight security reduction. In *Cryptology and Network Security, 5th International Conference*, volume 4301 of *Lecture Notes in Computer Science*, pages 19–36. Springer, 2006.
2. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography, 12th International Workshop, SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005.
3. Mihir Bellare, Brent Waters, and Scott Yilek. Identity-based encryption secure against selective opening attack. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*, volume 6597 of *Lecture Notes in Computer Science*, pages 235–252. Springer, 2011.
4. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.
5. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, 2004.

6. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
7. Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 647–657. IEEE Computer Society, 2007.
8. Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008*, pages 283–292. IEEE Computer Society, 2008.
9. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, vol 20(no 3):265–294, 2007.
10. Liqun Chen and Zhaohui Cheng. Security proof of sakai-kasahara's identity-based encryption scheme. In *Cryptography and Coding, 10th IMA International Conference*, volume 3796 of *Lecture Notes in Computer Science*, pages 442–459. Springer, 2005.
11. Liqun Chen, Zhaohui Cheng, John Malone-Lee, and Nigel P. Smart. An efficient id-kem based on the sakai-kasahara key construction. Cryptology ePrint Archive, Report 2005/224, 2005.
12. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding, 8th IMA International Conference*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.
13. Xuhua Ding and Gene Tsudik. Simple identity-based cryptography with mediated rsa. In *Topics in Cryptology - CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 193–210. Springer, 2003.
14. Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC 1999*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 1999.
15. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, vol 156(no 16), 2008.
16. David Galindo. Boneh-franklin identity based encryption revisited. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 791–802. Springer, 2005.
17. Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, 2006.
18. Mahabir Prasad Jhanwar and Rana Barua. A variant of boneh-gentry-hamburg's pairing-free identity based encryption scheme. In *Information Security and Cryptology, 4th International Conference, Inscrypt 2008*, volume 5487 of *Lecture Notes in Computer Science*, pages 314–331. Springer, 2008.
19. Eike Kiltz and David Galindo. Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. *Theoretical Computer Science*, vol 410(no 4-49), 2009.
20. J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal. Assoc. Comp. Mach*, vol 32(no 1):229–246, 1985.
21. A. K. Lenstra, H. W. Lenstra, and L. Lovasz. Factoring polynomials with rational coefficients. *Math. Ann.*, vol 261:515–534, 1982.
22. Ueli M. Maurer and Yacov Yacobi. A non-interactive public-key distribution system. *Design Codes Cryptography*, Vol 9(no 3):305–316, 1996.
23. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.
24. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Symposium on Cryptography and Information Security, Okinawa, Japan*, 2000.
25. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing over elliptic curve (in japanese). In *Symposium on Cryptography and Information Security, Oiso, Japan*, 2001.
26. Ryuichi SAKAI and Masao KASAHARA. Id based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003.
27. Michael Scott. Computing the tate pairing. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2005.
28. Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
29. Michael Steiner, Gene Tsudik, and Michael Waidner. Diffie-hellman key distribution extended to group communication. In *ACM Conference on Computer and Communications Security*, pages 31–37, 1996.
30. Hatsukazu Tanaka. A realization scheme for the identity-based cryptosystem. In *Advances in Cryptology - CRYPTO 1987*, volume 293 of *Lecture Notes in Computer Science*, pages 340–349. Springer, 1987.
31. S. Tsujii and T. Itoh. An id-based cryptosystem based on the discrete logarithm problem. *IEEE Journal on Selected Areas in Communications*, vol 7(no 4):467–473, 1989.

32. Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.
33. Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.