Sādhanā (2022)47:35
https://doi.org/10.1007/s12046-022-01806-5

A quantum secure ID-based cryptographic encryption based on group rings

GAURAV MITTAL[1,2,*], SUNIL KUMAR[2] and SANDEEP KUMAR[2]

[1]Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, India
[2]Defence Research and Development Organization, Delhi, India
e-mail: gmittal@ma.iitr.ac.in; sunilkumar.hqr@gov.in; sandeepkumar.hqr@gov.in

MS received 11 June 2021; revised 17 December 2021; accepted 5 January 2022

Abstract. Identity-based (ID-based) encryption is a very important cryptographic primitive. It is advantageous over the conventional public key cryptosystems due to direct and easy verification of the public keys. The security of most of the ID-based encryption schemes is based directly or indirectly on solving integer factorization problem, Elliptic curve discrete logarithm problem or discrete logarithm problem. It is known that these well-studied problems are not safe against attacks on a sufficiently large quantum computer. Therefore, in this paper we propose a secure ID-based encryption scheme whose security depends on the newly discovered hard problems in the algebraic structure of group rings. We show that the proposed scheme is IND-ID-CPA secure and safe against the chosen ciphertext attack. Moreover, we also comment on the IND-ID-CCA security of the proposed scheme.

Keywords. Cryptography; ID-based encryption; Group rings.

1. Introduction

The concept of identity-based (ID-based) cryptography was proposed by Shamir way back in 1984 [1]. The main advantage of ID-based cryptography over public key cryptography is that former provides a easy verification of the public keys instead of distribution of public key certificates. To be more precise, in an ID-based encryption scheme, the message can be encrypted by utilizing the identity of the receiver which acts as a public key. Consequently, the sender does not need to first obtain the large public key of receiver in the form of public key certificates and then verify the digital signature on it. This saves time as well as reduces the computational complexity of the system. Shamir [1] suggested to use the unambiguous identity (i.e., email or some other identity) as a public key whereas the corresponding private key is generated through some algorithm by a trusted party known as private key generator (PKG). Therefore, only public parameters of the encryption scheme need to be certified and there is no need of public key certificates.

Some of the first practical ID-based encryption schemes appeared in [2] and [3]. Boneh and Franklin [2] used the concept of Weil pairing [4] that is a category of bilinear maps for the construction of ID-based cryptosystem. The work done in [2] received a lot of attention and as a result, a number of ID-based cryptosystems have been proposed that are based on bilinear mappings [5–7]. Apart from this, several other ID-based encryption schemes have been proposed by various researchers, see e.g. [8–14] and the references therein. In all of these schemes, the public key also consists of some randomly chosen bits along with the identity. The security of all the above-mentioned schemes depend on one of the following problems: integer factorization problem (IFP), elliptic curve discrete logarithm problem (ECDLP), discrete logarithm problem (DLP), combination of integer factorization and discrete logarithm problem. Due to various quantum algorithms [15], it is believed that these problems can be efficiently solved on a sufficiently large quantum computer. For example, it requires 2300 (approx.) qubit quantum computer to break ECDLP based on an elliptic curve over a field with prime of size 256 bits [16], 4099 (approx.) qubit quantum computer to break the RSA-2048 or IFP with primes of size 1024 bits [16, 17]. Therefore, there is a demand of novel ID-based encryption schemes whose security depends on other hard problems that are safe against attacks on quantum computers.

There are several ID-based encryption schemes proposed in the literature that are claimed to be quantum secure. For example, the ID-based encryption scheme based on multivariate cryptography [18], ID-based encryption schemes based on lattices [19, 20], lightweight identity-based encryption based on ring learning with errors (RLWE) [21], identity-based encryption with equality test [22] etc. Basically, these schemes are based on well-studied and hard mathematical problems (see table 5). In this paper, we

*For correspondence
Published online: 12 February 2022

continue in the direction of construction of quantum safe ID-based encryption schemes and consider an algebraic structure (i.e., a set with two binary operations satisfying few properties) that is source of many hard problems known as *group rings* (see [23] for an excellent survey on group rings). We utilize the hard problems of group rings to construct a novel ID-based encryption scheme.

The subject of group rings was introduced in cryptography with hard problems such as discrete logarithm problem in group rings, inverse computation problem etc., by Hurley *et al* in 2011 [24]. We emphasize the fact that before 2011, other papers were not directly based on group rings, see e.g., [25]. We also refer to [26–29] and the references therein for some recent literature on group rings based cryptography. The hard problems mentioned-above in group rings can be utilized to create secure cryptographic primitives, since there is no known algorithm to solve these problems in polynomial time.

Our contribution. As already mentioned, most of the existing (not recently proposed) ID-based encryption schemes detailed above are not immune against attacks on a sufficiently large quantum computer. Therefore, there is an emerging need of new ideas of ID-based encryption schemes that are believed to be immune against any attack on a quantum computer. In this paper, we propose a secure (even in quantum sense) and novel ID-based encryption scheme that is based on the structure of group rings. We carefully discuss the security analysis of our novel scheme in the current scenario. Additionally, in order to show the importance of the structure of group rings, we study the security of our scheme by assuming that ECDLP is solvable. We also discuss a toy example to demonstrate our scheme. Moreover, we talk about the size of parameters required in our scheme as well as the computational cost of the various algorithms of the scheme.

In terms of security, we show the IND-ID-CPA security (see next section for its definition), as well as the security against the brute force and chosen ciphertext attacks. Moreover, we also comment on the IND-ID-CCA security (see subsection 4.4) of our scheme. Hence, our scheme may be a good alternative to the existing ID-based encryption schemes that are not quantum safe.

The rest of this paper is divided in seven sections. The preliminaries required in this paper are introduced in section 2. Our novel ID-based encryption scheme based on group rings and bilinear pairings is proposed in section 3. In section 4, we make efforts to discuss the security of our novel scheme. The section 5 contains discussion on a toy example of our scheme. The computation cost of our scheme as well as discussion on parameters selection is given in section 6. In section 7, we compare the hard problems of group rings with the existing hard problems. Finally, the last section includes some conclusive remarks.

2. Preliminaries

In this section, we discuss some basic definitions that are pre-requisite in our work. This section contains definitions from [2, 4, 23, 29].

Definition 2.1 Group Ring: Let $\mathcal{K}$ be a ring with unity and G be a group. The following set $\mathcal{K}G$ of all finite sums is known as group ring:

$$\mathcal{K}G = \Big\{ \sum_{m=1}^{t} k_m g_m : k_m \in \mathcal{K}, g_m \in G \Big\}.$$

This set is a group as well as ring under the binary operations $+, *$. Here, a binary operation is a calculation that combines two elements to produce another element. For $\alpha, \beta \in \mathcal{K}G$, let the product of α and β be denoted by $\alpha * \beta$.

Definition 2.2 Unit of a group ring: Let $\alpha \in \mathcal{K}G$. The element α is said to be a unit of $\mathcal{K}G$ if it has inverse with respect to multiplication, i.e., there exists some $\alpha^{-1} \in \mathcal{K}G$ fulfilling

$$\alpha * \alpha^{-1} = \alpha^{-1} * \alpha = 1. \tag{1}$$

Here 1 is the unity of $\mathcal{K}G$, i.e.,

$$\alpha * 1 = \alpha = 1 * \alpha$$

for any $\alpha \in \mathcal{K}G$.

Next, we recall some hard problems in group rings.

Definition 2.3 Discrete logarithm problem in group rings (DLPGR): Let $\alpha_1, \alpha_2 \in \mathcal{K}G$ be two elements such that

$$\alpha_2 = \alpha_1^k \tag{2}$$

for some positive integer k. Then DLPGR is the problem of deducing k from the known values of α_1 and α_2.

Definition 2.4 Inverse computation problem in group rings (ICPGR): Let $\alpha \in \mathcal{K}G$ be a unit. Then ICPGR is the problem of deducing inverse α^{-1} of α, given only α such that (1) holds.

Definition 2.5 Decisional Diffie-Hellman problem (DDHP) in group rings: Let $\alpha \in \mathcal{K}G$ be a randomly chosen element of group ring. Then DDHP in group rings states that $\alpha^{rs} \in \mathcal{K}G$ should appear like a random element whenever one knows the values of α^r and α^s for arbitrary and independently chosen r and s.

Next, we recall the definitions of identity-based encryption, IND-ID-CPA security, Elliptic curves and Weil pairing.

Definition 2.6 Identity-Based Encryption: In order to define an identity-based encryption scheme, the following four algorithms are required:

(1) **Setup:** A security parameter s is submitted as an input to this algorithm. In response, it returns the set Prms, i.e., all the system parameters along with the master key. The master key is only known to the *Private Key Generator* (PKG), whereas the set Prms is known publicly.
(2) **Extract:** This algorithm extracts the private key corresponding to an arbitrary ID $\in \{0,1\}^*$, Prms and master key.
(3) **Encrypt:** This algorithm provides the ciphertext C corresponding to a message M encrypted via ID and Prms.
(4) **Decrypt:** This algorithm provides the message M corresponding to a ciphertext C decrypted via private key associated with identity ID.

Definition 2.7 Indistinguishability under chosen plaintext attack (IND-ID-CPA) security: An identity based encryption scheme is IND-ID-CPA secure (or semantically secure) if any polynomially bounded adversary $\mathcal{A}d$ plays the following IND-ID-CPA game with the challenger $\mathcal{C}h$ and has negligible advantage over $\mathcal{C}h$:

(1) **Setup:** $\mathcal{C}h$ obtains the system parameters Prms by submitting the security parameter s to the Setup algorithm.
(2) **Phase-1:** $\mathcal{A}d$ issues the extraction queries $\mathrm{ID}_{i,1\le i\le n}$ (one by one and may be adaptively) for some $m > 0$. $\mathcal{C}h$ runs the extract algorithm to extract the private keys corresponding to $\mathrm{ID}_{i,1\le i\le n}$ and send these to $\mathcal{A}d$.
(3) **Challenge:** Once $\mathcal{A}d$ decides that enough extraction queries are made, it selects a identity ID (additional constraint is that this ID $\neq \mathrm{ID}_i$ for all $1 \le i \le n$) and two messages $\mathcal{M}_0$ and $\mathcal{M}_1$ on which it wish to challenge $\mathcal{C}h$. In response, $\mathcal{C}h$ randomly picks $w \in \{0,1\}$ and encrypts $\mathcal{M}_w$. $\mathcal{C}h$ submits this as the challenged ciphertext to $\mathcal{A}d$.
(4) **Phase-2:** $\mathcal{A}d$ issues some other extraction queries

$$\mathrm{ID}_{i,n+1\le i\le m},$$

(again one by one and may be adaptively) for some $m > 0$ with the condition that

$$\mathrm{ID}_i \neq \mathrm{ID}$$

for each $n+1 \le i \le m$. The response of $\mathcal{C}h$ is similar as in Phase-1.
(5) **Guess:** Finally, $\mathcal{A}d$ picks a $w' \in \{0,1\}$. The scheme is IND-ID-CPA secure if $w' \neq w$. In the above game, $\mathcal{A}d$ is an IND-ID-CPA adversary. We denote the function

$$\left|Pr[w = w'] - \frac{1}{2}\right|$$

as the *advantage* of $\mathcal{A}d$ against the ID-based encryption scheme.

Definition 2.8 Elliptic curves: For a prime p, let $\mathbb{F}_p$ denote a finite field, $\mathcal{A}, \mathcal{B} \in \mathbb{F}_p$ and O be a special point at infinity. The following $E(\mathbb{F}_p)$ is known as elliptic curve:

$$E(\mathbb{F}_p) = \left\{(x,y) \in (\mathbb{F}_p)^2 \mid y^2 = x^3 + \mathcal{A}x + \mathcal{B}, \; 4\mathcal{A}^3 + 27\mathcal{B}^2 \neq 0\right\} \cup \{\mathrm{O}\}.$$

Here, $(\mathbb{F}_p)^2 = \mathbb{F}_p \times \mathbb{F}_p$, i.e., it contains tuples of the form (x, y), where both x and y are elements of $\mathbb{F}_p$.

Definition 2.9 Weil pairing: Let G_1 be a subgroup of $E(\mathbb{F}_p)$ for some prime p and G_2 be a group and M be a positive integer. The bilinear mapping $W_M : G_1 \times G_1 \to G_2$ that satisfies

$$W_M(g_1, g_2)^M = 1$$

for $g_1, g_2 \in G_1$ is known as Weil pairing.

3. Identity-based encryption scheme

In this section, we present our identity-based encryption scheme, called IBES by describing the following four algorithms:

Parameters-setup, Extract, Encrypt and Decrypt. Let s be the security parameter associated with the Parameters-setup algorithm. Let $\mathcal{G}en$ be a parameter generator of IBES (defined in (i) below).

Parameters-setup: Given s, the algorithm runs as follows:

(i) Run $\mathcal{G}en$ on input s. $\mathcal{G}en$ generates three groups G_1, G_2 and G_3, where

$$|G_1| = |G_2| = q, \text{ for some prime } q \text{ and } |G_3| \ge \ell_1,$$

where the message space M is such that $M = \{0,1\}^{\ell_1}$ for some positive integer ℓ_1. Let P_1 be generator of G_1,

$$\widehat{e} : G_1 \times G_1 \to G_2$$

be a bilinear map (e.g. Weil pairing) and

$$G_3 = \{g_i \mid 1 \le i \le |G_3|\}$$

be arbitrary but fixed ordering of elements of the group G_3. The ciphertext space is $G_1^* \times (\mathbb{F}_2 G_3)^2$, i.e., ciphertext C corresponding to a message $\mathcal{M}$ consists of three elements, one from G_1^* (the non-zero elements of the group G_1), and two from group ring $\mathbb{F}_2 G_3$. The group ring $\mathbb{F}_2 G_3$ contains elements that are linear combination of the elements of G_3, where the coefficients are taken from $\mathbb{F}_2$. For example, if

$$G_3 = \{g_1, g_2, g_3\},$$

then

$$\mathbb{F}_2 G_3 = \{0, g_1, g_2, g_3, g_1 + g_2, g_1 + g_3, g_2 + g_3, g_1 + g_2 + g_3\}.$$

(ii) Construct a unit u of the group ring $\mathbb{F}_2 G_3$ of large order. This can be done by generating the known type units and take their product or taking the power of a known type unit. Here, by a known type unit, we mean to say a unit that can be generated with the help of a certain formula, e.g. Bicyclic units, Bass cyclic units etc., [23]. The order, $\mathrm{Ord}(u)$, of the unit u is not public but the inverse of u is public. Here, by definition, $\mathrm{Ord}(u)$ is the smallest positive integer η such that $u^\eta = 1$.

(iii) Pick a random $k_1 \in \mathbb{F}_q^*$ and a tuple

$$A = (y_1, y_2, \cdots, y_{\ell_2})$$

for some positive integer ℓ_2, where $y_i \in \mathbb{F}_{\mathrm{Ord(u)}}$ for each i. Here, $\mathbb{F}_{\mathrm{Ord(u)}}$ is a group of integers under addition modulo $\mathrm{Ord}(u)$, i.e.,

$$\mathbb{F}_{\mathrm{Ord(u)}} = \{0, 1, 2, \cdots, \mathrm{Ord}(u) - 1\}.$$

(iv) Use A to compute

$$B = (u^{y_1}, u^{y_2}, \cdots, u^{y_{\ell_2}}) \in (\mathbb{F}_2 G_3)^{\ell_2}. \tag{3}$$

(v) Choose three cryptographic hash functions

$$H_1 : \{0,1\}^* \to G_1, H_2 : G_2^* \to \{0,1\}^{\ell_1}$$

and

$$H_3 : \{0,1\}^* \to \{0,1\}^{\ell_2},$$

where $\{0,1\}^*$ is the set of all the arbitrary binary sequences of finite length. The system parameters are

$$\mathrm{Prms} = \langle q, G_i, H_i, \widehat{e}, \ell_1, \ell_2, P_1, u, B, P_1^{\mathrm{Pub}}, \mathbb{F}_2 G_3 \rangle, \tag{4}$$

where $i = 1, 2, 3$. The master key is the tuple (k_1, A) and P_1^{Pub} is defined in (9).

Extract: This algorithm computes the following: Let ID represent the binary representation of an identity. For any $\mathrm{ID} \in \{0,1\}^*$, compute

$$h_1 = H_1(\mathrm{ID}) \quad \text{and} \quad h_3 = H_3(\mathrm{ID}). \tag{5}$$

Let

$$h_3 = (h_3^{(1)}, \cdots, h_3^{(\ell_2)}). \tag{6}$$

Compute

$$a_{\mathrm{ID}} = \sum_{i=1}^{\ell_2} h_3^{(i)} y_i \bmod (\mathrm{Ord}(u)). \tag{7}$$

Therefore, the extracted private key is

$$\mathrm{ID}^{\mathrm{Pri}} = (\mathrm{ID}^{\mathrm{Pri}(1)}, \mathrm{ID}^{\mathrm{Pri}(2)}) = (a_{\mathrm{ID}}, k_1 h_1). \tag{8}$$

The corresponding public key is

$$P_1^{\mathrm{Pub}} = (b_{\mathrm{ID}}, P_2 = k_1 P_1), \tag{9}$$

where

$$b_{\mathrm{ID}} = \prod_{i=1}^{\ell_2} u^{y_i h_3^{(i)}}. \tag{10}$$

Encrypt: The following algorithm is used to encrypt the message $\mathcal{M}$ having same size as that of h_2: Compute h_1 as in (5) and choose two random integers e_1 and e_2 in $\mathbb{F}_q^*$. Further, use the second coordinate P_2 of P_1^{Pub} given in (9) to compute

$$h_2 = H_2(\widehat{e}(h_1, P_2)^{e_1}). \tag{11}$$

Next, use (11) to compute

$$\mathcal{M} \oplus h_2 = (z_1, \cdots, z_{\ell_1}),$$

and express

$$\mathcal{M} \oplus h_2 = \sum_{i=1}^{\ell_1} z_i g_i \tag{12}$$

as an element of group ring $\mathbb{F}_2 G_3$. Finally, utilize the first coordinate b_{ID} of P_1^{Pub} and (12) to compute the ciphertext

$$C = (C_1, C_2, C_3) = (e_1 P_1, u^{-e_2}, (\mathcal{M} \oplus h_2) * (b_{\mathrm{ID}})^{e_2}). \tag{13}$$

Decrypt: After obtaining C, the following algorithm can be employed to decrypt the ciphertext: Utilize (7) and (13) to compute

$$\mathcal{M}' = C_3 * (C_2)^{a_{\mathrm{ID}}} \tag{14}$$

and

$$\widehat{e}(k_1 h_1, C_1) = \widehat{e}(k_1 h_1, e_1 P_1) = \widehat{e}(h_1, P_1)^{k_1 e_1}$$

$$= \widehat{e}(h_1, k_1 P_1)^{e_1} = \widehat{e}(h_1, P_2)^{e_1}.$$

Note that $\mathcal{M}'$ is of the form

$$\mathcal{M}' = \sum_{i=1}^{\ell_1} z'_i g_i.$$

Therefore, we express $\mathcal{M}'$ as an element in $\{0,1\}^{\ell_1}$ by extracting the coefficients z'_i in the sum $\sum_{i=1}^{\ell_1} z'_i g_i$, i.e.,

$$\mathcal{M}' = (z'_1, \cdots, z'_{\ell_1}) \tag{15}$$

and use this along with (5), (13) to compute

$$\mathcal{M} = \mathcal{M}' \oplus H_2(\widehat{e}(k_1 h_1, C_1)). \tag{16}$$

The required message is given by (16).

Correctness: Next, we verify the correctness of decrypt algorithm: From (10), we note that

$$b_{\mathrm{ID}} = u^{(\sum_{i=1}^{\ell_2} y_i h_3^{(i)})} = u^{a_{\mathrm{ID}}}, \tag{17}$$

since $u^{\mathrm{Ord(u)}} = e$. Further, using (14) and (17), we deduce

$$\begin{aligned}\mathcal{M}' &= C_3 * (C_2)^{a_{\mathrm{ID}}} \\ &= \left((\mathcal{M} \oplus h_2) * (b_{\mathrm{ID}})^{e_2}\right) * (u^{-e_2})^{a_{\mathrm{ID}}} \\ &= (\mathcal{M} \oplus h_2) * \left((u^{a_{\mathrm{ID}}})^{e_2}\right) * (u^{-e_2})^{a_{\mathrm{ID}}}\right) \\ &= \mathcal{M} \oplus h_2. \end{aligned} \tag{18}$$

Moreover, $H_2(\widehat{e}(k_1 h_1, C_1))$ computed during decryption is same as h_2 computed during encryption in (12). Consequently, (16) and (18) imply that

$$\begin{aligned}&\mathcal{M}' \oplus H_2(\widehat{e}(k_1 h_1, C_1)) \\ &\quad = \mathcal{M} \oplus h_2 \oplus h_2 = \mathcal{M}.\end{aligned}$$

Thus, we have completed the description of IBES.

We refer to table 1 in order to show the differences between IBES and the conventional ID based PKC of Boneh *et al* [2].

4. Security

In order to study the security of IBES, let us first define a related scheme that is not an identity based scheme. This scheme is a public key encryption scheme and we name it as PES. PES is described via the following three algorithms: Key-generation, Encrypt and Decrypt.

Key-generation: Given s, the algorithm runs as follows.

(i) The first four steps of this algorithm are same as the steps (*i*)-(*iv*) of Parameters-setup algorithm of IBES.
(ii) Choose a cryptographic hash function $H_2 : G_2^* \to \{0,1\}^{\ell_1}$.
(iii) Pick two random $h_1 \in G_1^*$ and $h_3 \in \{0,1\}^{\ell_2}$ and let h_3 be given by (6).
(iv) Compute a_{ID} as in (7) using h_3 mentioned in (*iii*). Therefore, the private key is $\mathrm{ID}^{\mathrm{Pri}}$ and the corresponding public key P_1^{Pub} are same as in (8) and (9).

The public parameters along with public keys are

$$\begin{aligned}\mathrm{PubPrms} = \langle &q, G_i, h_1, H_2, h_3, \widehat{e}, \ell_1, \ell_2, \\ &P_1, u, B,\ P_1^{\mathrm{Pub}}, \mathbb{F}_2 G_3 \rangle,\end{aligned} \tag{19}$$

where $i = 1, 2, 3$.

Encrypt and Decrypt: These algorithms are exactly similar to Encrypt and Decrypt algorithms described for IBES.

Therefore, with this, we have completed the description of PES. It is worth to mention that the PES is a public key encryption scheme that emerges from IBES. The main difference between the two is that h_1 and h_3, given by (5), in IBES correspond to an ID, whereas h_1 and h_3 in PES are randomly picked from G_1^* and $\{0,1\}^{\ell_2}$. No ID is used in PES to generate any public or private key.

Next, in the following proposition, we show that IND-ID-CPA attack on IBES implies the IND-CPA attack on

Table 1. Conventional Scheme of Boneh *et al* [2] and IBES.

Algorithms	Conventional ID based scheme	IBES
Parameters	q , G_1 , G_2 ,H_1 , H_2 , Master key k_1 , P_1 bilinear pairing , ℓ_1	q , G_1 , G_2 , G_3 ,H_1 , H_2 , H_3 ,Master key (k_1, A) , P_1 bilinear pairing , ℓ_1 , ℓ_2
Extract	for any ID compute H_1(ID) private key is $k_1 H_1$(ID) public key is $k_1 P_1$	for any ID, compute public and private keys as in (8) and (9)
Encrypt	pick a random e_1 compute $h_2 = H_2(\widehat{e}(h_1, k_1 P_1)^{e_1})$ for a message $\mathcal{M}$, ciphertext is ($e_1 P_1$, $\mathcal{M} \oplus h_2$)	pick two random e_1, e_2 compute h_2 as in (11) for a message $\mathcal{M}$, ciphertext is given by (13)
Decrypt	let $\mathcal{M}' = \mathcal{M} \oplus h_2$ plaintext is given by $\mathcal{M}' \oplus H_2(\widehat{e}(k_1 h_1, e_1 P_1))$	compute $\mathcal{M}'$ as in (14) plaintext is given by (16)

PES, i.e., the extraction inquiries of the private key are of no help to the adversary. The proof is largely motivated by the works of [30] and [2].

Proposition 4.1 *Let the hash functions H_1 and H_3 mentioned in the scheme IBES be random oracles. Let $\mathcal{A}_{\text{ID}}$ be an IND-ID-CPA adversary and it has advantage $\epsilon(s)$ against IBES. Then there is a IND-CPA adversary $\mathcal{A}$ and that adversary has advantage*

$$\epsilon^*(s) \geq \left(\frac{q_M}{1+q_M}\right)^{q_M} \left(\frac{1}{1+q_M}\right) \epsilon(s) \tag{20}$$

against PES, provided there exists a simulator $\mathcal{B}$ that provides a part a_{ID} of the private key (for any ID not equal to challenged ID) and $\mathcal{B}$ can respond to extraction query at most q_M times, where $q_M < \ell_2$.

Proof: We need to show the existence of a IND-CPA adversary $\mathcal{A}$ that utilizes $\mathcal{A}_{\text{ID}}$ to have an advantage $\epsilon^*(s)$ against PES. By definition, the game between $\mathcal{A}$ and challenger starts with the challenger provide random public parameters PubPrms as in (19) of PES, to $\mathcal{A}$. The corresponding private key is given by (8). The challenger provides PubPrms to $\mathcal{A}$. By definition, $\mathcal{A}$ outputs messages $\mathcal{M}_0$ and $\mathcal{M}_1$. In response, challenger picks a random $v \in \{0,1\}$ and encrypts $\mathcal{M}_v$ using PubPrms and provides the ciphertext to $\mathcal{A}$. In response, let $v' \in \{0,1\}$ be the output of $\mathcal{A}$, i.e., v' is guess for v by $\mathcal{A}$.

Next, Algorithm $\mathcal{A}$ uses $\mathcal{A}_{\text{ID}}$ in an IND-ID-CPA game in the following manner: $\mathcal{A}$ gives the system parameters Prms (4) of IBES to $\mathcal{A}_{\text{ID}}$. Here

$$q, G_{i,1\leq i\leq 3}, H_2, \widehat{e}, \ell_1, \ell_2, P_1, u, B, P_1^{\text{Pub}}, \mathbb{F}_2 G_3$$

are taken from (19), and H_1 and H_3 are random oracles controlled by $\mathcal{A}$ through the following algorithm:

H_1 and H_3-queries: We assume that $\mathcal{A}_{\text{ID}}$ can query H_1 and H_3 at any time. In order to answer these queries, $\mathcal{A}$ creates two lists

$$\text{list}_1 = \{\langle ID_r, h_{1r}, b_r, c_r\rangle \mid r \in I\},$$

and

$$\text{list}_3 = \{\langle ID_r, h_{3r}, c_r\rangle \mid r \in I\}$$

of tuples for some index set I. In list_1, corresponding to a $r \in I$, ID_r is the identity, h_{1r} is the hash value $H_1(ID_r)$ if $ID_r \in \text{list}_1 \cap \text{list}_3$ and it is an element of G_1^* if $ID_r \notin \text{list}_1 \cap \text{list}_3$, b_r is a random element of $\mathbb{F}_q^*$ and c_r is either 0 or 1. Similarly, in list_3, corresponding to a $r \in I$, ID_r is the identity, h_{3r} is the hash value $H_3(ID_r)$ if $ID_r \in \text{list}_1 \cap \text{list}_3$ and it is an element of $\{0,1\}^{\ell_2}$ if $ID_r \notin \text{list}_1 \cap \text{list}_3$, and c_r is same as in list_1. We note that corresponding to a ID_r, first and last elements of both the tuples in lists are same. Initially,

$$\text{list}_1 = \text{list}_3 = \emptyset.$$

When $\mathcal{A}_{\text{ID}}$ query both H_1 and H_3 for a particular identity ID_r, $\mathcal{A}$ responds in the following manner:

(1) If $ID_r \in \text{list}_1 \cap \text{list}_3$, then $\mathcal{A}$ responds with

$$h_{1r} = H_1(ID_r), \quad \text{and} \quad h_{3r} = H_3(ID_r).$$

(2) Otherwise, $\mathcal{A}$ generates a random coin $\mathcal{C} \in \{0,1\}$. Let

$$\Pr[\mathcal{C} = 0] = \eta \tag{21}$$

for some $\eta > 0$. This η will be determined later on.

(3) $\mathcal{A}$ picks two random $b_r \in \mathbb{F}_q^*$ and $d_r \in \{0,1\}^{\ell_2}$. If $\mathcal{C} = 0$, compute

$$h_{1r} = b_r P_1 \in G_1^*.$$

If $\mathcal{C} = 1$, compute

$$h_{1r} = b_r h_1 \in G_1^*.$$

(4) $\mathcal{A}$ adds tuples

$$\langle ID_r, h_{1r}, b_r, \mathcal{C}\rangle \quad \text{and} \quad \langle ID_r, h_{3r} = d_r, \mathcal{C}\rangle \tag{22}$$

to lists list_1, list_3, respectively, and responds to $\mathcal{A}_{\text{ID}}$ with

$$H_1(ID_r) = h_{1r} \quad \text{and} \quad H_3(ID_r) = d_r. \tag{23}$$

Clearly, h_{1r} and d_r are uniform in G_1^* and $\{0,1\}^{\ell_2}$, respectively and independent of the current view of $\mathcal{A}_{\text{ID}}$ as desired.

Phase-1: Let $\mathcal{A}_{\text{ID}}$ issue the private key extraction query corresponding to ID_r. $\mathcal{A}$ answers this query as follows:

(1) $\mathcal{A}$ runs the algorithm H_1 and H_3-queries detailed above to obtain $H_1(ID_r)$ and $H_3(ID_r)$ as in (23) and the corresponding tuples mentioned in (22) in the lists list_1 and list_3. If $\mathcal{C} = 1$, then $\mathcal{A}$ terminates by reporting failure and the attack on PES failed.

(2) Otherwise, $\mathcal{C} = 0$. Hence

$$h_{1r} = b_r P_1 \text{ and } h_{3r} = d_r = (d_r^{(1)}, \cdots, d_r^{(\ell_2)}).$$

Using $\mathcal{B}$, $\mathcal{A}$ defines

$$\text{ID}_r^{\text{Pri}} = (a_{\text{ID}_r}, b_r P_2),$$

where

$$a_{\text{ID}_r} = \sum_{i=1}^{\ell_2} d_r^{(i)} y_i \bmod (\text{Ord}(u)).$$

Clearly,

$$b_r P_2 = b_r k_1 P_1 = k_1 h_{1r}$$

that means $\mathrm{ID}_r^{\mathrm{Pri}}$ is the private key corresponding to public key ID_r. Further, $\mathcal{A}$ provides $\mathrm{ID}_r^{\mathrm{Pri}}$ to $\mathcal{A}_{\mathrm{ID}}$.

Challenge: Once $\mathcal{A}_{\mathrm{ID}}$ concludes that Phase-1 is finished, it submits $\mathcal{M}_0$ and $\mathcal{M}_1$ and an identity $\mathrm{ID}_{\mathrm{ch}}$ on which it wants to play the game. $\mathcal{A}$ does the following:

(1) $\mathcal{A}$ submits messages $\mathcal{M}_0$ and $\mathcal{M}_1$ to its challenger which replies with PES ciphertext

$$C = (C_1', C_2', C_3') \tag{24}$$

corresponding to plaintext $\mathcal{M}_w$ for $w \in \{0, 1\}$ randomly chosen.

(2) After obtaining $\mathcal{M}_w$, $\mathcal{A}$ runs the algorithm H_1 and H_3-queries. In response, $\mathcal{A}$ obtains h_1 and d such that

$$H_1(\mathrm{ID}_{\mathrm{ch}}) = h_1 \quad \text{and} \quad H_3(\mathrm{ID}_{\mathrm{ch}}) = d. \tag{25}$$

Further, let

$$\langle \mathrm{ID}_{\mathrm{ch}}, h_1', b, \mathcal{C} \rangle \text{ and } \langle \mathrm{ID}_{\mathrm{ch}}, h_3 = d, \mathcal{C} \rangle$$

be two tuples. If $\mathcal{C} = 0$, $\mathcal{A}$ terminates by reporting failure. Consequently, attack on PES has failed.

(3) Otherwise, we know $\mathcal{C} = 1$. Therefore,

$$h_1' = bh_1 \quad \text{and} \quad h_3 = d. \tag{26}$$

Clearly, for C in (24), $C_1' \in G_1^*$. As $b \in \mathbb{F}_q^*$, b^{-1} exists. Consequently, $\mathcal{A}$ sets

$$C' = (b^{-1}C_1', C_2', C_3') \tag{27}$$

and submit this to $\mathcal{A}_{\mathrm{ID}}$ as a challenge. We claim that C' in (27) is an encryption of $\mathcal{M}_w$ under the scheme IBES and public key $\mathrm{ID}_{\mathrm{ch}}$. To see this, note that $H_1(\mathrm{ID}_{\mathrm{ch}})$ and $H_3(\mathrm{ID}_{\mathrm{ch}})$ are same as in (25), where

$$d = (d^{(1)}, \cdots, d^{(\ell_2)}).$$

The private key is

$$\mathrm{ID}_{\mathrm{ch}}^{\mathrm{Pri}} = (\mathrm{ID}_{\mathrm{ch}}^{\mathrm{Pri}(1)}, \mathrm{ID}_{\mathrm{ch}}^{\mathrm{Pri}(2)}) = (a_{\mathrm{ID}_{\mathrm{ch}}}, bP_2 = k_1 h_1'). \tag{28}$$

Moreover, using (28), we note that

$$\widehat{e}(b^{-1}C_1, \mathrm{ID}_{\mathrm{ch}}^{\mathrm{Pri}(2)}) = \widehat{e}(b^{-1}C_1, k_1 h_1'). \tag{29}$$

We use the property of bilinear mapping in (29) to deduce that

$$\begin{aligned} \widehat{e}(b^{-1}C_1, \mathrm{ID}_{\mathrm{ch}}^{\mathrm{Pri}(2)}) &= \widehat{e}(C_1, k_1 b^{-1} h_1') \\ &= \widehat{e}(C_1, k_1 h_1) \\ &= \widehat{e}(b^{-1}C_1, \mathrm{ID}^{\mathrm{Pri}(2)}), \end{aligned}$$

where we have utilized (26) in deducing the above equality. Therefore, the decryption of C' given by (27) in the scheme IBES using the private key $\mathrm{ID}_{\mathrm{ch}}^{\mathrm{Pri}}$ is same as decryption of C given by (24) in the scheme PES with the private key $\mathrm{ID}^{\mathrm{Pri}}$.

Phase-2: Similar to Phase-1, during this phase, $\mathcal{A}$ answers to extraction query (for private key), where the new ID's are different from the $\mathrm{ID}_{\mathrm{ch}}^{\mathrm{Pri}}$.

Guess: $\mathcal{A}_{\mathrm{ID}}$ guesses w' for w. Consequently, $\mathcal{A}$ guesses w' for w.

Claim: We claim that if $\mathcal{A}$ never aborts while simulation, then $\mathcal{A}_{\mathrm{ID}}$ is of same opinion as in the real attack. Moreover, $|\Pr[w = w'] - \frac{1}{2}| \geq \epsilon$, if $\mathcal{A}$ does not abort (this probability is calculated on the random bits w and w' employed by $\mathcal{A}_{\mathrm{ID}}$, $\mathcal{A}$ together with the challenger). To see this, note that the outputs of the algorithm H_1 and H_3-queries are random and therefore, these are same as in the real attack. Moreover, all the outputs of extract algorithm are valid. Finally, C' submitted to $\mathcal{A}_{\mathrm{ID}}$ is the IBES encryption of $\mathcal{M}_w$ for randomly chosen $w \in \{0, 1\}$. Consequently, by definition of $\mathcal{A}_{\mathrm{ID}}$, claim holds.

Next, we explain why $q_M < \ell_2$. Note that if $q_M \geq \ell_2$, then as assumed in the statement of Proposition 4.1, $\mathcal{B}$ can respond to extraction query ℓ_2 times. With each query, $\mathcal{A}$ gets a new

$$a_{\mathrm{ID}_r} = \sum_{i=1}^{\ell_2} d_r^{(i)} y_i \bmod (\mathrm{Ord}(u)),$$

where

$$h_3^r = d_r = (d_r^{(1)}, \cdots, d_r^{(\ell_2)}).$$

Consequently, in order to deduce

$$A = (y_1, \cdots, y_{\ell_2}),$$

$\mathcal{A}$ could create a $\ell_2 \times \ell_2$ linear system $\mathcal{L}$ of ℓ_2 equations that can be solved efficiently. However, this system may not be solvable always as it depends on the dimension, i.e., $\mathrm{Rank}(\mathcal{L})$ of the linear system. Here, $\mathrm{Rank}(\mathcal{L})$ is the maximum number of linearly independent columns of $\mathcal{L}$. In order to protect $\mathcal{A}$ from getting some part of the master key, we restrict $q_m < \ell_2$.

It remains to find out the advantage $\epsilon^*(s)$ of $\mathcal{A}$. To see this, we deduce the probability that during simulation, $\mathcal{A}$ aborts. As $\mathcal{A}_{\mathrm{ID}}$ makes q_M extraction queries, the probability of $\mathcal{A}$ not reporting the failure in the Challenge step is η^{q_M}. This is because $\mathcal{A}$ reports failure when $\mathcal{C}_r = 1$ and we assume that (21) holds.

Moreover, the probability that $\mathcal{A}$ does not report failure in the challenge step is $1 - \eta$. This is because in this step, if $\mathcal{C} = 0$, then failure occurs and we assume that (21) holds. Consequently, during simulation, the total probability of not reporting failure by $\mathcal{A}$ is

$$\psi(\eta) := \eta^{q_M}(1 - \eta).$$

In order to deduce the maximum value of $\psi(\eta)$, we see that

$$\psi'(\eta) = q_M \eta^{q_M - 1} - (q_M + 1)\eta^{q_M} = 0$$

$$\Rightarrow \eta' = \frac{q_M}{q_M + 1}.$$

Therefore, the maximum probability becomes

$$\psi(\eta') = \left(\frac{q_M}{1 + q_M}\right)^{q_M} \left(\frac{1}{1 + q_M}\right).$$

This means that the advantage of $\mathcal{A}$ is

$$\epsilon^*(s) \geq \epsilon(s)\psi(\eta'),$$

i.e., (20) holds. This completes the proof.

4.1 *IND-ID-CPA security of IBES*

The proposition 4.1 assures that the IND-ID-CPA security of the scheme IBES depends on the IND-CPA security of the scheme PES. Consequently, we check for the IND-CPA security of the scheme PES. We emphasize the fact that we study the security of IBES scheme by assuming that ECDLP is solvable. This is because of the following:

(i) It is known that a sufficiently large quantum computer can break ECDLP (e.g., it is proved that a quantum computer having roughly 2300 qubits can easily break the current cryptographically secure implementations of elliptic curves (e.g., curve 25519 proposed by Bernstein [15]) [31].
(ii) If ECDLP is strong, then the IND-CPA security of the scheme PES can be studied on the similar lines by using the idea of [2] and Proposition 4.1.

Therefore, it is worth studying the IND-CPA security of PES scheme by assuming that ECDLP is solvable. We study the same in the next theorem.

Theorem 4.1 Suppose that ECDLP is solvable. Then, even if there exists a polynomial time algorithm $\mathcal{B}$ that solves inverse computation problem in group rings (ICPGR), the scheme IBES is IND-ID-CPA secure, provided DDHP is a hard problem in group rings.

Proof It is clear from the Proposition 4.1 that IBES is IND-ID-CPA secure, if PES is IND-CPA secure. Consequently, we study the IND-CPA security of the scheme PES. By definition, the challenger provides public key along with the public parameters PubPrms as in (19) to an adversary $\mathcal{A}d$. In response, $\mathcal{A}d$ submits two messages $\mathcal{M}_0$ and $\mathcal{M}_1$ to the challenger. The challenger picks a random $v \in \{0, 1\}$ and encrypts $\mathcal{M}_v$ using PubPrms and deduces the ciphertext C by using (13) and provides it to $\mathcal{A}d$. Here, h_2 is given by (12).

By assumption, ECDLP is solvable which means that $\mathcal{A}d$ can calculate e_1 as well as h_2, since k_1 is also known to adversary. Suppose that $\mathcal{A}d$ guesses $v = 0$. Consequently, $\mathcal{A}d$ calculates

$$\mathcal{M}_v \oplus h_2.$$

After this, $\mathcal{A}d$ can win the game in the following two situations:

(i) Suppose $\mathcal{A}d$ solves DDHP in group rings, i.e., given u^{e_2} (this can be calculated by submitting u^{-e_2} to $\mathcal{B}$) and

$$b_{\mathrm{ID}} = \prod_{i=1}^{\ell_2} u^{y_i h_3^{(i)}} = u^{a_{\mathrm{ID}}},$$

if $\mathcal{A}d$ calculates $(u^{a_{\mathrm{ID}}})^{e_2}$, then $\mathcal{A}d$ can also calculate

$$\mathcal{M}'_v = (\mathcal{M}_v \oplus h_2) * (b_{\mathrm{ID}})^{e_2}. \tag{30}$$

So, if $\mathcal{A}d$ solves the DDHP in group rings, then PES is not IND-CPA secure and $\mathcal{A}d$ wins the game.
(ii) Suppose $\mathcal{M}_v \oplus h_2$ is invertible, then using $\mathcal{B}$, $\mathcal{A}d$ calculates

$$(\mathcal{M}_v \oplus h_2)^{-1} * \left((\mathcal{M}_v \oplus h_2) * (b_{\mathrm{ID}})^{e_2}\right)$$

$$= \left((\mathcal{M}_v \oplus h_2)^{-1} * (\mathcal{M}_v \oplus h_2)\right) * (b_{\mathrm{ID}})^{e_2} = (b_{\mathrm{ID}})^{e_2}. \tag{31}$$

This, as in (*i*) above, again implies that if $\mathcal{A}d$ solves the DDHP of (*i*), then the scheme is not IND-CPA secure.

In all, we have shown that if DDHP is a hard problem in group rings, then IBES is IND-CPA secure. This discussion is summarized in table 2.

4.1.1 *Hardness of DDHP in group rings*

Let $\alpha \in \mathcal{K}G$ be a randomly chosen element of group ring. Then, as mentioned in Definition 2.5, DDHP in group rings states that $\alpha^{rs} \in \mathcal{K}G$ should appear like a random element whenever we know the values of α^r and α^s for arbitrary and independently chosen r and s.

Informally, it means that if one know the values of α^r and α^s, then it should not be possible to calculate the value of α^{rs}. However, we know that if DLPGR (see Definition 2.3) is solvable, then DDHP is solvable. This is because, one can obtain r and s from α^r and α^s by utilizing the algorithm that solves DLPGR. After knowing r and s, it is straightforward to calculate α^{rs}. Consequently, in the following subsection, we study of hardness of DLPGR in order to study the hardness of DDHP.

4.1.2 *Hardness of DLPGR*

We recall from Definition 2.3 that DLPGR is the problem of deducing k from the known values of α_1 and α_2 fulfilling (2) for some positive integer k and $\alpha_1, \alpha_2 \in \mathcal{K}G$. Let number of elements in G be m and

Table 2. IND-ID-CPA security.

Challenger $\mathcal{C}h$	Adversary $\mathcal{A}d$
Provides public key along with PubPrms as in (19) to $\mathcal{A}d$	
	Submits two messages $\mathcal{M}_0$ and $\mathcal{M}_1$ to $\mathcal{C}h$
Picks a random $v \in \{0,1\}$ and encrypts $\mathcal{M}_v$ and deduces C using (13), and submits C to $\mathcal{A}d$	
	Wins the game if $\mathcal{A}d$ calculates (30) or (31), i.e., if DDHP is solvable.

$$\alpha_1 = \sum_{j=1}^{m} k_j g_j \text{ with } \mathrm{Ord}(\alpha_1) = n.$$

Here, $\mathrm{Ord}(\alpha)$ means the order of α. Since (2) holds, via brute force attack, k can be deduced in at most n multiplications of α_1 with itself.

To this end, next, we deduce the bit complexities of DLPGR via brute force. We observe that the computation of α_1^2 requires $O(2m^2)$ multiplications (i.e., m^2 ring multiplications and m^2 group multiplications). Therefore, DLPGR can be solved in at most $O(m^2 n)$ multiplications. In the simplest case when $\mathcal{K} = \mathbb{F}_2$ (finite field of order 2) and $G = \mathbb{F}_m$ (cyclic group of order m), where m is a ℓ-bit number, we see that DLPGR can be solved in $\mathcal{O} = O(m^2 n \ell^2)$ bit operations. If n has size r bits, i.e., $n \approx 2^r$, then $\mathcal{O} = O(m^2 2^r \ell^2)$. This is an exponential time since the input size is in r bits (order of unit is input). However, due to standard collision algorithms, such as *Shanks babystep-giantstep algorithm* [4], one can reduce the number of bit operations needed to solve DLPGR to $\mathcal{O} = O(m^2 2^{\frac{r}{2}} \ell^2)$ bit operations which is again an exponential time. This is because a general collision algorithm tends to reduce the operations by square root times.

Further, to the best of our knowledge, there are no index calculus type algorithms to solve DLPGR. One of the main reasons behind this is that the best known sub-exponential algorithm, i.e., *index calculus algorithm* [4] in group can not be directly adapted to the settings of group rings, since the latter has a complex structure than that of former. Therefore, we can say that currently there is no known polynomial or sub-exponential time algorithm to solve DLPGR. The size of the parameters related to group rings required in our scheme are mentioned in the following table 3.

Table 3. Size of parameters and the expected security.

Parameters Size	Expected security
$\lvert G\rvert = m \geq 2^7$, $\mathrm{Ord}(\alpha_1) \geq 2^{225}$	128 bits
$\lvert G\rvert = m \geq 2^8$, $\mathrm{Ord}(\alpha_1) \geq 2^{485}$	256 bits

It is known that the current secure implementations of DLP require a prime of size 2048 bits in order to obtain security equivalent to 112 bits. Consequently, it is worth to mention that the size of parameters required in our scheme related to group rings are quite small as compare to DLP in groups. We refer to [32] for constructions of group rings that can contain arbitrary units of large order.

4.2 *Hardness of directly deducing private keys from public keys*

In our IBES scheme, corresponding to a ID, we recall from (8) and (9) that private key is

$$\mathrm{ID}^{\mathrm{Pri}} = (\mathrm{ID}^{\mathrm{Pri}(1)}, \mathrm{ID}^{\mathrm{Pri}(2)}) = (a_{\mathrm{ID}}, k_1 h_1)$$

for a_{ID} defined in (7), and the corresponding public key is

$$P_1^{\mathrm{Pub}} = (b_{\mathrm{ID}}, P_2 = k_1 P_1),$$

where b_{ID} is defined in (10). The hardness of deducing k_1 from the known values of P_1 and P_2 is well studied [4]. The important requirement in our scheme is that an adversary must not able to compute a_{ID} from

$$b_{\mathrm{ID}} = \prod_{i=1}^{\ell_2} u^{y_i h_3^{(i)}} = u^{\sum_{i=1}^{\ell_2} y_i h_3^{(i)}} = u^{a_{\mathrm{ID}}}.$$

Clearly, in order to deduce a_{ID} from b_{ID}, an adversary needs to solve DLPGR which is a hard problem in group rings as discussed in subsection 4.1.2. We want to mention that the hardness of DLPGR in group rings is not only due to the fact that there is currently no polynomial time algorithm to solve it, but also that it is well studied by the mathematically community. We made the last statement because DLPGR is a generalization of DLP in groups to DLP in group rings, and there is no known polynomial time quantum algorithm to solve DLP in non-abelian groups [15]. Therefore, it is computationally infeasible for a polynomially bound adversary to deduce the private keys from the public keys in our scheme.

4.3 *Security against Chosen Ciphertext Attack (CCA)*

In this subsection, we show that the scheme IBES is secure against CCA, provided ICPGR is a hard problem. Again, we emphasize the fact that we study the security of IBES scheme by assuming that ECDLP is solvable for the same reason mentioned in subsection 4.1.

Theorem 4.2 *Suppose that an adversary $\mathcal{A}d$ knows an oracle $\mathcal{O}$ that provides the messages corresponding to arbitrary ciphertexts encrypted using authenticated public keys of IBES scheme. Then $\mathcal{A}d$ can solve ICPGR.*

Proof Suppose that $\mathcal{A}d$ consults $\mathcal{O}$ and sends Prms as in (4) to $\mathcal{O}$. Here, the public key corresponding to identity ID is

$$P_1^{\text{Pub}} = (b_{\text{ID}}, P_2 = k_1 P_1),$$

where b_{ID} is defined in (10). After getting the call from $\mathcal{O}$ to submit the ciphertext, $\mathcal{A}d$ selects a random e_1 along with $e_2 = 1$ and $C_3 = I$ to calculate

$$C = (e_1 P_1, u^{-1}, I),$$

and submits C to $\mathcal{O}$. We recall that the legitimate C is defined in (13). Here, e_1 can be arbitrarily chosen by adversary, since ECDLP is solvable. Also, note that $e_1 = 1$ can be taken by $\mathcal{A}d$ as u^{-1} is publicly known.

In response, as per the decrypt algorithm of IBES, $\mathcal{O}$ returns

$$(C_3 * (C_2)^{a_{\text{ID}}})' \oplus H_2(\widehat{e}(k_1 h_1, C_1)),$$

where $(C_3 * (C_2)^{a_{\text{ID}}})'$ is the representation of element $C_3 * (C_2)^{a_{\text{ID}}}$ of the group ring in the space $\{0,1\}^{\ell_2}$. Further, as we have assumed that ECDLP is solvable, $\mathcal{A}d$ calculates $H_2(\widehat{e}(k_1 h_1, C_1))$. Consequently, $\mathcal{A}d$ possesses the knowledge of element

$$C_3 * (C_2)^{a_{\text{ID}}} = I * (u^{-1})^{a_{\text{ID}}} = u^{-a_{\text{ID}}}$$

of the group ring, i.e., $\mathcal{A}d$ possesses the knowledge of inverse of $u^{a_{\text{ID}}}$.

However, if $\mathcal{A}d$ is smart enough and accepts only those ciphertexts for which

$$C_3 \neq I,$$

even then $\mathcal{A}d$ can manipulate $\mathcal{O}$ in the following manner. $\mathcal{A}d$ submits

$$C = (C_1, C_2, C_3)$$

to $\mathcal{O}$, where e_1 can be arbitrary chosen, $e_2 = 1$ and $C_3 = v$, where v is a unit of $\mathbb{F}_2 G_3$ whose inverse is known to $\mathcal{A}d$. For example, $\mathcal{A}d$ can select $v = u^2$.

In response, $\mathcal{O}$ returns

$$(C_3 * (C_2)^{a_{\text{ID}}})' \oplus H_2(\widehat{e}(k_1 h_1, C_1)).$$

Further, as ECDLP is solvable, $\mathcal{A}d$ calculates $H_2(\widehat{e}(k_1 h_1, C_1))$. Consequently, $\mathcal{A}d$ obtains

$$C_3 * (C_2)^{a_{\text{ID}}} = v * (u^{-1})^{a_{\text{ID}}}.$$

Moreover, using the knowledge of v^{-1}, $\mathcal{A}d$ can deduce $u^{-a_{\text{ID}}}$, i.e., inverse of $u^{a_{\text{ID}}}$.

In both the situations, we have shown that $\mathcal{A}d$ can deduce $u^{-a_{\text{ID}}}$. Further, as $u^{a_{\text{ID}}}$ can be arbitrarily chosen by adversary, i.e., it can be any element whose inverse is needed by $\mathcal{A}d$, we conclude that $\mathcal{A}d$ can solve ICPGR using $\mathcal{O}$. This completes the proof.

Due to Theorem 4.2, it is important to study the hardness of ICPGR in arbitrary group rings. We do the same in the following subsection.

4.3.1 *Hardness of ICPGR*

Let the number of elements in G be m and

$$\alpha = \sum_{j=1}^{m} k_j g_j. \tag{32}$$

Then, as mentioned in Definition 2.4, ICPGR demands to find a

$$\alpha^{-1} = \sum_{t=1}^{m} k_t' g_t$$

such that (1) holds. This means

$$\left(\sum_{j=1}^{m} k_j g_j\right) * \left(\sum_{t=1}^{m} k_t' g_t\right) = \sum_{j,t=1}^{m} (k_j k_t')(g_j g_t) = 1. \tag{33}$$

There is no general formula to compute the inverse of an arbitrary element, provided inverse exists, due to involvement of algebraic structure. However, one can always utilize brute force attack. Since $k_j' \in \mathbb{F}_2$, $|G| = m$, and $|\mathbb{F}_2 G| = 2^m$, (33) can always be solved in roughly $O(2^m)$ operations by substituting every element of the group ring. Further, due to the following collision algorithm, one can reduce the number of operations roughly to $O(2^{\frac{m}{2}})$.

Algorithm 1 **Collision algorithm for ICPGR**

- INPUT: a unit α of the group ring.
- OUTPUT: inverse α^{-1} of α.
- Steps: The basic idea is to create two lists and then adopt the following simple procedure to find the collision.

(1) We create lists $\mathcal{L}_1$ and $\mathcal{L}_2$ of elements of group rings, where

$$\mathcal{L}_1 = \{(\alpha_1, \alpha * \alpha_1) \in (\mathbb{F}_2 G)^2 : \alpha_1 = \sum_{j=1}^{[m/2]} k^{(j)} g_j\},$$

$$\mathcal{L}_2 = \{(\alpha_2, 1 - (\alpha * \alpha_2)) \in (\mathbb{F}_2 G)^2 : \alpha_2 = \sum_{j=[m/2]+1}^{m} k^{(j)} g_j\}.$$

Note that both the lists $\mathcal{L}_1$ and $\mathcal{L}_2$ roughly contain $2^{[m/2]}$ elements.

(2) Find a match of second coordinates in the lists that is guaranteed via the following proof of correctness. Let the second coordinates of elements

$$(\alpha_1', \alpha * \alpha_1') \quad \text{and} \quad (\alpha_2', (1 - \alpha * \alpha_2'))$$

of the lists $\mathcal{L}_1$ and $\mathcal{L}_2$ match. Then $\alpha_1' + \alpha_2'$ is the required inverse of α.

Proof of correctness: Using (33), we can write

$$\begin{aligned} 1 = \alpha * \alpha^{-1} &= \left(\sum_{j=1}^{m} k_j g_j\right) * \left(\sum_{t=1}^{m} k_t' g_t\right) \\ &= \left(\sum_{j=1}^{m} k_j g_j\right) * \left(\sum_{t=1}^{[m/2]} k_t' g_t + \sum_{t=[m/2]+1}^{m} k_t' g_t\right) \\ &= \left(\sum_{j=1}^{m} k_j g_j\right) * \left(\sum_{t=1}^{[m/2]} k_t' g_t\right) \\ &\quad + \left(\sum_{j=1}^{m} k_j g_j\right) * \left(\sum_{t=[m/2]+1}^{m} k_t' g_t\right). \end{aligned}$$

This further means that

$$\begin{aligned} \alpha * \alpha_1 &= \left(\sum_{j=1}^{m} k_j g_j\right) * \left(\sum_{t=1}^{[m/2]} k_t' g_t\right) \\ &= 1 - \left(\sum_{j=1}^{m} k_j g_j\right) * \left(\sum_{t=[m/2]+1}^{m} k_t' g_t\right) \\ &= 1 - \alpha * \alpha_2, \end{aligned} \tag{34}$$

where

$$\alpha_1 = \sum_{t=1}^{[m/2]} k_t' g_t, \quad \text{and} \quad \alpha_2 = \sum_{t=[m/2]+1}^{m} k_t' g_t.$$

A match in the second coordinates of the lists $\mathcal{L}_1$ and $\mathcal{L}_2$ is guaranteed by (34).

To this end, next, we discuss the bit operations complexity of ICPGR due to Algorithm 1. The main task is that of creating lists $\mathcal{L}_1$ and $\mathcal{L}_2$ that require $O(2^{m/2})$ operations. Note that both the lists involves multiplication of elements of group ring in the second coordinate. As the number of elements in G are m and α is given by (32), creating one entry in both the lists $\mathcal{L}_1$ and $\mathcal{L}_2$ require $O(m^2)$ multiplications. This means we need to perform $O(2^{m/2}m^2)$ multiplications in order to create two lists $\mathcal{L}_1$ and $\mathcal{L}_2$.

In the simplest case, when $\mathcal{K} = \mathbb{F}_2$ and $G = \mathbb{F}_m$ for a ℓ-bit integer m, we see that ICPGR can be solved in $O(2^{m/2}m^2\ell^2)$ bit operations. Further, using standard sorting and searching algorithms [4], a match in two lists of cardinality 2^m can be found in $O(\log(2^{m/2}))$ steps. Therefore, Algorithm 1 takes around $O(2^{m/2}m^2\ell^2 \log(2^{m/2}))$ bit operations which is a great saving than the naive brute force algorithm that takes $O(2^m m^2 \ell^2)$ bit operations.

From this discussion, we conclude that ICPGR is safe against brute force attack or attack via collision algorithm, provided we wisely choose the size of the group. The expected security provided by ICPGR is presented in table 4.

4.4 *IND-ID-CCA security*

The indistinguishability under (non-adaptive) chosen ciphertext attack (IND-ID-CCA) security is an extension of IND-ID-CPA security. The additional advantage to adversary in case of IND-ID-CCA game is that in the **Phase-2** of Definition 2.7, $\mathcal{A}d$ can also make decryption queries of ciphertexts C_i corresponding to identities ID_i such that for any i, C_i is not equal to the challenged ciphertext corresponding to challenged identity ID.

Due to discussion in subsections 4.1.1 and 4.1.2, we can see that DDHP is a hard problem in group rings. Consequently, Theorem 4.1 implies that our novel IBES scheme is IND-ID-CPA secure. Further, on utilizing the Fujisaki-Okamoto transformation [33], under a random oracle model, our IBES scheme can be converted into a scheme that is IND-ID-CCA secure.

Moreover, due to a fine-grained and modular toolkit of transformations provided by Hofheinz *et al* [34], our IBES scheme can be converted into a scheme that is IND-ID-CCA secure under a quantum random oracle model. Consequently, we can obtain the guaranteed security in a post-quantum setting. In all, we can convert any IND-CPA secure scheme to a IND-CCA scheme by the careful utilization of various available transformations.

Table 4. Security provided by ICPGR.

$\lvert G\rvert = m$	security (atleast)
$m \geq 200$ or 8 bits	112 bits
$m \geq 450$ or 9 bits	256 bits

5. Example

In this section, we discuss a toy example to demonstrate our scheme IBES. We consider the elliptic curve

$$E(\mathbb{F}_{1009}) = \{(x, y) \mid x^3 + 37x = y^2\} \tag{35}$$

over the finite field $\mathbb{F}_{1009}$. Let $\widehat{e}$ be the Weil pairing. One can efficiently compute the Weil pairing by using the Miller's algorithm [35]. Let G_1 be subgroup of $E(\mathbb{F}_{1009})$ generated by $P_1 = (49, 20)$. It can be verified that P_1 is a point on elliptic curve given by (35). Let

$$\ell_1 = 3 \quad \text{and} \quad G_3 = Q_8,$$

i.e., Quaternion group of order 8 and G_2 be a multiplicative group generated by 105 modulo 1009. The presentation of G_3 is as follows:

$$G_3 := \langle a, b \mid a^4 = b^4 = 1, a^2 = b^2, ba = a^{-1}b \rangle.$$

Let $u = 1 + a + b$. It can be verified that u is a unit of $\mathbb{F}_2 Q_8$ [36] with inverse

$$u^{-1} = 1 + ab + ab^2 + b^3 + ab^3.$$

Let

$$k_1 = 1, \quad \ell_2 = 2, \quad \text{and} \quad A = (2, 3). \tag{36}$$

This means

$$B = (u^2, u^3)$$

$$= (1 + ab + ab^3, 1 + ab + ab^2 + b^3 + ab^3). \tag{37}$$

One can use the software [37] to compute the calculations in group rings. Let the hash functions be

$$H_1 : \{0,1\}^* \to G_1, H_2 : G_2^* \to \{0,1\}^3$$

and

$$H_3 : \{0,1\}^* \to \{0,1\}^2.$$

We can take the standard hash function, e.g., SHA-256 and truncate the 256 bit hash value to obtain the desired number of bits. Let ID to be used be gmittal@ma.iitr.ac.in. We take the ASCII value of each symbol of ID and then take the binary representation of each ASCII value to write ID as an element of $\{0,1\}^*$. Let

$$H_1(ID) = (8, 703), \quad H_3(ID) = (1, 0). \tag{38}$$

Then using (38), (36) and (7), we have

$$a_{\text{ID}} = \sum_{i=1}^{2} h_3^{(i)} y_i = 2. \tag{39}$$

The extracted private key is

$$\text{ID}^{\text{Pri}} = (2, (8, 703)).$$

The corresponding public key using (36) is

$$P_1^{\text{Pub}} = (u^2, (49, 20)), \tag{40}$$

where u^2 is same as in (37).

Next, we encrypt a message $\mathcal{M} = (1, 1, 0)$. Let

$$e_1 = 1, \quad \text{and} \quad e_2 = 2. \tag{41}$$

Accordingly, it follows from (11) and [38, Example 6.2] that

$$h_2 = H_2(\widehat{e}((8, 703), (49, 20))) = H_2(105).$$

Let H_2 be such that

$$H_2(105) = (0, 1, 1).$$

Consequently, we have

$$\mathcal{M} \oplus h_2 = (1, 1, 0) \oplus (0, 1, 1) = (1, 0, 1).$$

We write $\mathcal{M} \oplus h_2$ as an element of group ring $\mathbb{F}_2 G_3$ as

$$\mathcal{M} \oplus h_2 = 1 + 0(a) + a^2. \tag{42}$$

Finally, we utilize (40), (41) and (42) to compute the ciphertext

$$C = ((49, 20), u^{-2}, (1 + a^2) * (u^2)^2)$$

$$= ((49, 20), 1 + ab + ab^3, 1 + a^2), \tag{43}$$

where we have used the fact that $u^4 = 1$.

Next, we decrypt the ciphertext C. We use (39) and (43) to compute

$$\mathcal{M}' = (1 + a^2) * (1 + ab + ab^3)^2$$

$$= (1 + a^2), \tag{44}$$

and

$$\widehat{e}((8, 703), (49, 120)) = \widehat{e}(h_1, P_2)^{e_1}.$$

We express $\mathcal{M}'$ in (44) as an element in $\{0,1\}^{\ell_1}$, i.e.,

$$\mathcal{M}' = (1, 0, 1)$$

and use this to compute the decrypted ciphertext as

$$(1,0,1) \oplus H_2(105) = (1,1,0).$$

This completes the decryption procedure.

6. Computational cost and parameters size

In this section, we discuss the computational cost of encrypting and decrypting a message through IBES and the size of parameters required to securely implement IBES. It is known that computational cost of an algorithm is directly proportional to the number of operations performed by the algorithm. Therefore, in order to find the cost, we calculate the number of operations required in each of the four algorithms of IBES in the following subsection:

6.1 *Cost of Parameters-setup algorithm*

In this algorithm, we need to compute B in (3), which involves calculation of u^{y_i} for each $1 \le i \le \ell_2$. We can use square and multiply algorithm [4] to compute u^{y_i} in $O(\log_2(y_i))$ multiplications. Therefore, the computation of B requires $O(\ell_2 \log_2(y_i))$ multiplications. Let T_m be the cost of a single multiplication of two elements of a group ring. Consequently, the total cost of this algorithm is $O(\ell_2 \log_2(y_i) T_m)$.

6.2 *Cost of extract algorithm*

In this algorithm, we need to compute h_1 and h_3 given in (4). Following [39], let T_{h_1} and T_{h_3} be the cost of hash functions H_1 and H_3 for computing h_1 and h_3 in (4). The computation of a_{ID} in (5) requires at most ℓ_2 modular additions. Further, one can compute $k_1 h_1$ given by (7) in at most $O(\log_2(k_1))$ additions on a group G_1 by using the standard double and add algorithm [4]. Let T_g be the cost of a single addition on a group G_1 and T_d be the cost of a modular addition. Therefore, the cost of extracting private key in (7) is $O(\log_2(k_1) T_g + \ell_2 T_d + T_{h_1} + T_{h_3})$.

Next, we deduce the cost of computing a public key given by (8). We note from (5) that

$$b_{\mathrm{ID}} = u^{a_{\mathrm{ID}}}.$$

As mentioned in subsection 6.1, we can use square and multiply algorithm to compute b_{ID} in $O(\log_2(a_{\mathrm{ID}}))$ multiplications. Also, the computation of $k_1 P_1$ can be done in $O(\log_2(k_1))$ additions on an elliptic curve. Let T_e be the cost of a single addition on an elliptic curve. Therefore, the cost of computing the public key is $O(\log_2(a_{\mathrm{ID}}) T_m + \log_2(k_1) T_e)$.

6.3 *Cost of encrypt algorithm*

In this algorithm, the first step is to compute h_2 in (10). The costs of computing h_1 and $k_1 P_1$ are already mentioned in subsection 6.2. Let T_B be the cost of applying the bilinear pairing $\widehat{e}$ on a pair of elements. For example, one can use Weil pairing that is a well known Bilinear pairing. Moreover, using Miller algorithm [35], one can compute Weil pairing in polynomial time. Further, Let T_{h_2} be the cost of Hash function H_2 and T'_m be the cost of a single multiplication of two elements of group G_2. Since, $\widehat{e}(h_1, k_1 P_1)$ is an element of G_2, the total cost of computing h_2 is $O(T_{h_2} + \log_2(e_1) T'_m)$. The computation of $\mathcal{M} + h_2$ needs ℓ_1 XOR operations. Let T_r be the cost of representing $\mathcal{M} + h_2$ in the form (11). Therefore, the final cost of computing ciphertext is

$$O(T_B + T_{h_2} + \log_2(e_1) T'_m + \log_2(k_1) T_e + T_r + (2\log_2(e_2) + 1) T_m),$$

where $\log_2(e_1) T'_m$ is the cost of computing $e_1 P_1$, $2\log_2(e_2) T_m$ is the cost of computing u^{-e_2} and $b_{\mathrm{ID}}^{e_2}$, $T_{h_2} + \log_2(e_1) T'_m$ is the cost of computing h_2 and T_m is the cost of multiplying $\mathcal{M} \oplus h_2$ with $b_{\mathrm{ID}}^{e_2}$. Here, we have neglected the cost of XOR operations in comparison to addition operation on an elliptic curve.

6.4 *Cost of Decrypt algorithm*

In this algorithm, the first step is to compute $\mathcal{M}'$ in (13). The cost of computing this is $\log_2(a_{\mathrm{ID}}) T_m + T_m$. The cost of computing $\widehat{e}(k_1 h_1, C_1)$ is T_B. Let T'_r be the cost of expressing $\mathcal{M}'$ (13) in the form (15). Therefore, the total cost of decryption is

$$O((1 + \log_2(a_{\mathrm{ID}})) T_m + T_B + T'_r),$$

where we have neglected the cost of XOR operations.

6.5 *Parameters selection*

In this subsection, we discuss how to select various parameters as well as hash functions required in IBES. The current secure implementation of elliptic curve cryptography requires selection of an elliptic curve over a prime field, where the size of prime is 256 bits. Therefore, we suggest to use a prime

$$p = 2^{255} - 19 \tag{45}$$

proposed by Bernstein [40]. To be more precise, one may consider the following elliptic curve

$$y^2 = x^3 + 486662x^2 + x,$$

which is a Montgomery curve [41], over the field $\mathbb{F}_p$ with p

as in (45). By considering the base point $x = 9$, we can generate a cyclic subgroup $\mathcal{S}$ whose order is the prime

$$2^{252} + 27742317777372353535851937790883648493. \tag{46}$$

We refer to [41] for more details on this elliptic curve. One may take G_1 and G_2 as $\mathcal{S}$. This choice is made due to the large size of subgroup $\mathcal{S}$. Further, one may use Weil pairing or Tate pairing as bilinear pairing, since both can be efficiently computed [4]. Also, we suggest

$$\ell_1 \geq 128, \quad \ell_2 \geq 224.$$

The choice of $\ell_1 \geq 128$ ensures that our scheme becomes a 128 bit (atleast) block cipher. The choice of $\ell_2 \geq 224$ ensures that one can use standard available hash functions that produce output of 224 bits, e.g. SHA-2 (secure hash algorithm) [4]. Accordingly $|G_3| \geq 128$ and the message space consists of all the binary strings of length 128 bits. The order of units needed in our scheme for securely implementing IBES is already discussed in section 4.

One of the most important requirement in IBES is that of hash functions H_1, H_2 and H_3. The main problem is that how one can construct the required hash functions. This is discussed in the following:

(1) Since H_3 maps an arbitrary but finite length input to a ℓ_2-bit output, we can use SHA-2 or SHA-3 as H_3. This is the reason of choosing $\ell_2 \geq 224$ as the outputs of SHA-2 or SHA-3 are of size 224, 256, 384 and 512 bits.

(2) We note that $H_1 : \{0,1\}^* \to G_1$. It follows from (46) that the size of G_1 is a 253-bit number. We also know that the generator of $\mathcal{S}$ is $x = 9$. In order to construct H_1, we use SHA-2 or SHA-3 with output 256 bits and take the 253 MSBs of the 256 bit output, by truncating (Tru) it. The final hash function is

$$H_1(\mathrm{ID}) = 9^{\mathrm{Tru}(\mathrm{SHA}-2(\mathrm{ID}))} \mod p,$$

where p is same as in (45). We have chosen 9 in above map as $x = 9$ is the generator of $\mathcal{S}$. The exponentiation with respect to 9 of the truncated hash value ensures that H_1 is a function that maps an arbitrary string to an element of G_1.

(3) We require $H_2 : G_2^* \to \{0,1\}^{\ell_1}$. Similar to point (2), for any $\alpha \in G_2^*$, we take

$$H_2(\alpha) = \mathrm{Tru}(\mathrm{SHA}\text{-}2(\beta)),$$

where $\beta = \mathrm{Bin}(\alpha)$, i.e., β is a binary representation of α and the function Tru truncates 256 bit values to $\ell_1 \geq 128$ MSBs. This completes the description of hash functions as well as parameters size for IBES.

7. Comparison with other post quantum cryptographic primitives

In this section, we discuss about the various Post quantum cryptographic primitives and compare the group rings based hard problems with other hard problems. The area of post quantum cryptography is divided in five categories, i.e., Lattice based, Multivariate equations based, Code based, Supersingular elliptic curve based and Hash based [15, 42]. We discuss the hard problems in the above-mentioned five areas in the following table 5.

The problems such as SVP, CVP, solving systems of multivariate polynomial equations and decoding a general linear code are NP-hard. The problem of constructing an isogeny is hard but no security reduction to a NP-hard problem is known. Further, the problem of deducing a collision is computationally secure, but again, no security reduction to a NP-hard problem is known. We refer to [4, 15] for more details on these problems.

The National Institute of Standards and Technology (NIS- T), USA is also organizing a competition to standarize quantum safe cryptographic primitives [42] that includes encryption schemes, digital signatures etc. The finalists of this competition are given in table 6.

The security of all of these primitives is based on a NP-hard problem as mentioned in table 5. Talking about the identity based encryption schemes, there are several schemes that are proved to be quantum safe. For example, the scheme of Kundu *et al* [18] based on multivariate cryptography, lightweight identity-based encryption based on ring learning with errors (RLWE) [21], identity-based encryption with equality test [22], the schemes [19, 20] based on lattices. We also refer to references in [18] for several other identity based encryption schemes based on multivariate cryptography.

To this end, we compare the hard problems in group rings with the existing hard problems as mentioned in table 5. The problems such as DLPGR and ICPGR are recently discovered hard problems [24], whereas the hard problems in lattices and coding theory etc., are well studied. However, the hardness of DLPGR and ICPGR can be

Table 5. PQC areas and their hard problems.

Type	Hard problem
Lattice based	Shortest vector problem (SVP) Closest vector problem (CVP)
Multivariate equations based	Solving multivariate equations
Code based	Decoding a general linear code
Supersingular elliptic curve based	Construction of an isogeny
Hash based	Find collision

Table 6. NIST PQC finalists.

Type	Schemes
Lattice based	Crystals-Kyber, SABER, FALCON, NTRU Crystals-Dilithium
Code based	Classic McEliece
Multivariate equations based	Rainbow

comparable to that of isogeny problem in supersingular elliptic curve cryptography. This is because, isogeny is a well studied hard problem but it is not proven to be NP-hard or NP-complete. The advantage of utilizing hard problems such as DLPGR and ICPGR is that they provide a security equivalent to 112 bits or more with the small size parameters as compare to other group based schemes. Moreover, currently, the area of group rings based cryptography is receiving a lot of attention of the various researchers (see [26–29] etc.). It is worth to mention that it is an open problem to deduce whether or not the problems DLPGR and ICPGR are NP-hard.

8. Conclusion

We have proposed a ID-based scheme based on the algebraic structure of group rings. We have shown the IND-ID-CPA security of the scheme by utilizing the hardness of hard problems in group rings. Further, we have also discussed about the various transformations that can turn our IND-ID-CPA secure scheme to a IND-ID-CCA secure, even in a post-quantum setting. Therefore, how to securely apply these transformations is an important future work in this direction. We want to emphasize the fact that our scheme requires considerably small sizes of the parameters (see Section 4) in order to provide the same level of security as provided by the current secure implementations of ECDLP and RSA (or DLP). For example, the order of groups should be atleast 256 and 3072 bits for ECDLP and RSA, respectively in order to obtain the security equivalent to 128 bits. Due to the small size of the involved parameters and strong security, we conclude that our proposed scheme can be employed in the near future in place of the existing schemes.

Acknowledgements

The authors are very thankful to both the Associate Editor and Anonymous Reviewer for their careful reading of the manuscript, valuable comments and suggestions that have immensely helped us in improving this work.

Declarations

Conflict of interest The author(s) declare(s) that there is no conflict of interest.

References

[1] Shamir A 1984 Identity-based cryptosystems and signature schemes. *Proc. of CRYPTO'84, Lecture Notes in Comput. Sci., Springer, Verlag.* 196: 47–53
[2] Boneh D and Franklin M 2003 Identity based encryption from the Weil pairing. *SIAM J. Comput.* 32: 586–615
[3] Cocks C 2001 An identity based encryption scheme based on quadratic residues. *International Conference on Cryptography and Coding (Proceedings of IMA), Lecture Notes in Comput. Sci., Springer-Verlag.* 2260: 360–363
[4] Hoffstein J, Pipher J and Silverman J 2008 *An introduction of mathematical cryptography*. New York: Springer
[5] Boneh D and Boyen X 2004 Secure identity based encryption without random oracles. *Advances in Cryptology, CRYPTO 2004, Lecture Notes in Computer Science, Springer, Verlag*. 3152: 443–459
[6] Boneh D and Boyen X 2004 Efficient selective-id secure identity based encryption without random oracles. *Advances in Cryptology, EUROCRYPT 2004, Lecture Notes in Computer Science, Springer, Verlag*. 3027: 223–238
[7] Waters B 2005 Efficient identity-based encryption without random oracles. *Advances in Cryptology, CRYPTO 2005, Lecture Notes in Comput. Sci., Springer, Verlag.* 3494: 114–127
[8] Gangishetti R, Gorantla M, Das M and Saxena A 2007 Threshold key issuing in identity-based cryptosystems. *Comput. Stand. Interfaces.* 29: 260–264
[9] Kiltz E and Vahlis Y 2008 CCA2 secure IBE: standard model efficiency through authenticated symmetric encryption. *Lecture Notes in Computer Science, Springer-Verlag*. 4964: 221–239
[10] Lee W and Liao K 2004 Constructing identity-based cryptosystems for discrete logarithm based cryptosystems. *J. Netw. Comput. Appl.* 22: 191-199
[11] Meshram C, Meshram S and Zhang M 2012 An ID-based cryptographic mechanisms based on GDLP and IFP. *Inf. Process. Lett.* 112: 753–758
[12] Meshram C and Meshram S 2013 An identity-based cryptographic model for discrete logarithm and integer factoring based cryptosystem. *Inf. Process. Lett.* 113: 375–380
[13] Meshram C 2015 An efficient ID-based cryptographic encryption based on discrete logarithm problem and integer factorization problem. *Inf. Process. Lett.* 115: 351–358
[14] Sun J, Zhang C, Zhang Y and Fang Y 2010 An identity-based security system foruser privacy in vehicular ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.* 27: 1227–1239
[15] Bernstein D, Buchmann J and Dahmen E 2009 *Post quantum cryptography*. Berlin: Springer
[16] National Academies of Sciences, Engineering, and Medicine, 2019 Quantum Computing: Progress and Prospects. E. Grumbling and M. Horowitz (Eds.). *Washington, D.C.: The National Academies Press*

[17] Zhang L, Miranskyy A and Rjaibi W 2019 Quantum Advantage and Y2K Bug: Comparison, *arxiv.org/pdf/1907.10454.pdf*

[18] Kundu N, Dey K, Stănicŭ P, Debnath S and Pal S 2021 Post-Quantum secure identity-based encryption from multivariate public key cryptography. *Lecture Notes in Electrical Engineering, Springer, Singapore*. vol. 744, pp. 139–149

[19] Ducas L, Lyubashevsky V and Prest T 2014 Efficient identity-based encryption over NTRU lattices. *Lecture Notes in Computer Science*. 8874: 22–42

[20] Zhandry M 2012 Secure identity-based encryption in the quantum random oracle model. *CRYPTO 2012, Lecture Notes in Computer Science, Springer Berlin.* 7417: 758–775

[21] Guneysu T and Oder T 2017 Towards lightweight Identity-Based encryption for the post-quantum-secure Internet of Things. 18*th International Symposium on Quality Electronic Design (ISQED), IEEE*

[22] Susilo W, Duong D and Le H 2020 Efficient post-quantum identity-based encryption with equality test. *IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*

[23] Milies C and Sehgal S 2002 *An Introduction to group rings*. Netherlands: Springer

[24] Hurley B and Hurley T 2011 Group ring cryptography. *Int. J. Pure Appl. Math.* 69: 67–86

[25] Rososhek S 2008 Cryptosystems in automorphism groups of group rings of Abelian groups. *J. Math. Sci. (N.Y.)*. 154: 386–391

[26] Goel N, Gupta I and Dubey M 2016 Undeniable signature scheme based over group ring. *AAECC*. 27: 523–535

[27] Gupta I, Pandey A and Dubey M 2019 A key exchange protocol using matrices over group ring. *Asian-European J. Math.* 12: 1950075

[28] Inam S and Ali R 2018 A new ElGamal-like cryptosystem based on matrices over groupring. *Neural Comput. and Applic.* 29: 1279–1283

[29] Mittal G, Kumar S, Narain S and Kumar S 2021 Group rings based public key cryptosystems. *J. Discret. Math. Sci. Cryptogr.* online first, https://doi.org/10.1080/09720529.2020.1796868

[30] Coron J 2000 On the exact security of Full-domain-Hash. *Advances in cryptology-Eurocrypt* 2000, *Lecture Notes in Comput. Sci., Springer-Verlag*. 1880: 229–235

[31] Proos J and Zalka C 2003 Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Info. Comput.* 3: 317-344

[32] Dietzel C and Mittal G 2021 Summands of finite group algebras. *Czech. Math. J.* 71: 1011–1014

[33] Fujisaki E and Okamoto T 1999 Secure integration of asymmetric and symmetric encryption schemes. *Advances in cryptology-Crypto* 99, *Lecture Notes in Comput. Sci., Springer-Verlag*. 1666: 537–554

[34] Hofheinz D, Hövelmanns K and Kiltz E 2017 A modular analysis of the Fujisaki-Okamoto transformation. *Theory of Cryptography, Kalai Y., Reyzin L. (eds). TCC 2017, Springer: Cham.* 10677: 341–371

[35] Miller V 1986 Short Programs for functions on Curves. *crypto.stanford.edu/miller/miller.pdf*

[36] Sharma R and Yadav P 2008 The unit group of Z_2Q_8. *Algebras Groups and Geometries.* 24: 425–430

[37] GAP *Groups, Algorithms, Programming*. https://www.gap-system.org.

[38] Aftuck A 2011 *The Weil pairing on elliptic curves and its cryptographic applications*. Graduate Thesis and Dissertations, UNF, Jacksonville

[39] Meshram C, Tseng Y-M, Lee C-C and Meshram S 2017 An IND-ID-CPA Secure ID-based cryptographic protocol using GDLP and IFP. *Informatica*. 28: 471–484

[40] Bernstein D *How do I use Curve25519 in my own software?*. cr.yp.to/ecdh.html

[41] *Curve25519.* https://en.wikipedia.org/wiki/Curve25519.

[42] *Post-Quantum Cryptography* https://csrc.nist.gov/projects/post-quantum-cryptography