The Equivalence of Two Quadratic Based IBEs

George Teşeleanu

Simion Stoilow Institute of Mathematics of the Romanian Academy
21 Calea Grivitei, Bucharest, Romania

Abstract. In this short note, we show that two identity-based encryption schemes, introduced by Joye and Zhao *et al.*, which appear different, are essentially the same scheme. The only difference between them is that one prioritizes speed (Zhao *et al.*), while the other prioritizes bandwidth (Joye). We also show how to speed-up Joye's scheme at the cost of adding at most one integer to the public key, thereby achieving a better encryption complexity, while having the same bandwidth requirements.

Keywords: identity based encryption · quadratic residues

1 Introduction

The concept of identity-based encryption (IBE) was first introduced in 1984 when Shamir outlined its fundamental principles [15]. However, he left the challenge of constructing a practical scheme unresolved. In 2001, Boneh and Franklin [3] proposed the first functional IBE scheme using bilinear mappings, while Cocks [8] introduced an alternative approach based on quadratic residues.

Cocks' encryption scheme operates on a bit-by-bit basis, where each encrypted bit is represented as a pair of integers. Decryption involves computing the Jacobi symbol of one of the integers in each pair. Due to this structure, the scheme has a high ciphertext-to-plaintext ratio, making it efficient only for encrypting small messages. A more space-efficient IBE based on quadratic residues was later introduced in [4]. However, this scheme relies on a deterministic time-complexity algorithm quartic in the size of the modulus, thus rendering it impractical.

The challenge of achieving both homomorphic properties and security in IBE has led to various approaches over the years. The first approach was taken by Clear *et al.* [6, 7], who introduced a universally anonymous xor-homomorphic variant of IBE. By utilizing polynomials, they revealed an underlying algebraic structure in the scheme. This structure was subsequently studied and refined by Joye [13], leading to improvements in both encryption speed and ciphertext expansion. Nevertheless, to send one bit Joye's proposal still requires sending two integers, just as Cocks' scheme. Building on [13], Nica and Ţiplea [14] revisited Joye's proposal using insights from an earlier study [16], ultimately providing a more straightforward description of the scheme. Another approach was presented in [17], drawing similarities to the Goldwasser-Micali cryptosystem [12].

Nevertheless, this solution also suffers from a large ciphertext-to-plaintext ratio, requiring four integers to encrypt a single bit.

Along with its initial proposal, Joye [13] also introduced another variant of Cocks' IBE that simplifies the homomorphic process at the cost of sending two additional bits. This is achieved by masking the message through multiplication with the Jacobi symbol of a random element t, and then encapsulating t in a Cocks-type ciphertext. This process is done twice, since we do not know whether $H(id)$ is a modular quadratic residue or not. On the other hand, Zhao *et al.*'s IBE [17] generates a random polynomial of degree 1, squares it to obtain $g(x)$, encodes the message using a Goldwasser-Micali-type method, and finally multiplies the result by $g(x)$. Again, this is done twice.

One of the main differences between the two proposals is that Zhao *et al.*'s scheme requires sending four integers, while Joye's requires sending two integers and two bits. Another difference is that encryption with Zhao *et al.*'s IBE involves at most 10 multiplications, whereas Joye's requires computing two Jacobi symbols and two modular inversions.

Although, at first glance, Zhao *et al.*'s proposal appears unrelated to Joye's variant, we managed to prove that the two are essentially the same scheme, the main difference being the choice of encoding/compression functions. A closer look at the two schemes reveals that Joye prioritized bandwidth efficiency at the cost of encryption speed, whereas Zhao *et al.* did the exact opposite. In the final part of the paper, we present a different trade-off that leads to a faster version of Joye's scheme, while maintaining the same bandwidth requirement.

Structure of the paper. In Section 2, we introduce the building blocks used in the paper. The equivalence between Zhao *et al.*'s IBE and Joye's IBE is provided in Section 3. We conclude in Section 4 by presenting an optimization of Joye's scheme.

2 Preliminaries

Notations. Throughout the paper, λ denotes a security parameter. The action of selecting a random element x from a sample space X is denoted by $x \xleftarrow{\$} X$, while $x \leftarrow y$ represents the assignment of value y to variable x.

The Jacobi symbol of an integer a modulo an integer n will be represented by $J_n(a)$. We consider the sets QR_n and QNR_n of quadratic and, respectively, non-quadratic residues modulo an integer n. J_n denotes the sets of integers modulo n with Jacobi symbol 1.

2.1 Identity-Based Encryption

An IBE scheme [2] comprises four probabilistic polynomial-time (PPT) algorithms, denoted as *Setup*, *KeyGen*, *Enc*, and *Dec*. The first algorithm takes the security parameter as input and produces the master secret key along with the

system's public parameters as output. The subsequent algorithm takes an identity id, the master secret key, the public parameters as input, and yields a private key associated with id as output. The third algorithm, labeled *Enc*, accepts a message m, an identity id, and the public parameters as input, encrypting m using a key derived from id to produce the ciphertext c. The final algorithm, *Dec*, decrypts the ciphertext c using the private key associated with id, yielding the original message m.

Zhao *et al.*'s IBE scheme. An IBE scheme based on the QR assumption[1] was introduced in [17]. This scheme utilizes polynomials modulo n, where the prime numbers p and q satisfy $p \equiv -q \bmod 4$. Later, [9] extended and generalized the scheme to support arbitrary choices of p and q. Here, we present the scheme's description as provided in [9].

Setup(λ): Given a security parameter λ, generate two primes $p, q > 2^\lambda$ and compute their product $n = pq$. Randomly generate two integers $u, y \in \mathbb{Z}_n$ such that $J_p(u) = J_q(u) = -1$ and $J_p(y) = -J_q(y)$. The public parameters are $pp = \{n, u, y, H\}$, where $H : \{0,1\}^* \to J_n$ is a cryptographic hash function. The master secret key is $msk = \{p, q\}$.

KeyGen(pp, msk, id): Let $R = H(id)$. If $R \in QR_n$, then compute $r \equiv R^{1/2} \bmod n$. Otherwise, computes $r = (uR)^{1/2} \bmod n$. The private key is r.

Enc(pp, id, m): On inputting pp, an identity id and a message $m \in \{0,1\}$, compute the hash value $R = H(id)$ and randomly choose two polynomials $f(x), \overline{f}(x)$ of degree 1 from $\mathbb{Z}_n[x]$. Also, calculate

$$g(x) = f(x)^2 \bmod (x^2 - R) \quad \text{and} \quad \overline{g}(x) = \overline{f}(x)^2 \bmod (x^2 - uR).$$

Return the ciphertext $C = (y^m \cdot g(x), y^m \cdot \overline{g}(x))$.

Dec(pp, r, C): On input pp, a secret key r and a ciphertext $C = (c(x), \overline{c}(x))$, compute

$$m' = \begin{cases} J_n(c(r)) & \text{if } r^2 \equiv H(id) \bmod n; \\ J_n(\overline{c}(r)) & \text{otherwise,} \end{cases}$$

and recover $m = (1 - m')/2$.

Joye's IBE scheme. A variant of Cocks' IBE that inherently supports a homomorphic operation was introduced in [13]. The scheme's security also relies on the QR assumption. We further present the IBE scheme described in [13].

Setup(λ): Given a security parameter λ, generate two primes $p, q > 2^\lambda$ and compute their product $n = pq$. Randomly generate an integer $u \in J_n \setminus QR_n$. The public parameters are $pp = \{n, u, H\}$, where $H : \{0,1\}^* \to J_n$ is a cryptographic hash function. The master secret key is $msk = \{p, q\}$.

[1] This assumption states that an adversary trying to decide if a random element is from $J_n \setminus QR_n$ or QR_n has a negligible success probability.

KeyGen(pp, msk, id): Let $R = H(id)$. If $R \in QR_n$, then compute $r \equiv R^{1/2} \bmod n$. Otherwise, computes $r = (uR)^{1/2} \bmod n$. The private key is r.

Enc(pp, id, m): On inputting pp, an identity id and a message $m \in \{-1, 1\}$, compute the hash value $R = H(id)$ and randomly choose two values $t_1, t_2 \xleftarrow{\$} \mathbb{Z}_n$. Also, calculate

$$\varepsilon_1 = m \cdot J_n(t_1) \quad \text{and} \quad c_1 = t_1 + \frac{R}{t_1} \bmod n,$$
$$\varepsilon_2 = m \cdot J_n(t_2) \quad \text{and} \quad c_2 = t_2 + \frac{uR}{t_2} \bmod n.$$

Return the ciphertext $C = (\varepsilon_1, c_1, \varepsilon_2, c_2)$.

Dec(pp, r, C): On input pp, a secret key r and a ciphertext C, compute

$$m = \begin{cases} \varepsilon_1 \cdot J_n(c_1 + 2r) & \text{if } r^2 \equiv H(id) \bmod n; \\ \varepsilon_2 \cdot J_n(c_2 + 2r) & \text{otherwise.} \end{cases}$$

3 The IBEs Equivalence

We start this section by formalizing the notion of equivalence between two encryption schemes. We first note that the encryption and decryption functions for an IBE can also be defined as efficient functions $Enc : \mathcal{ID} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ and $Dec : \mathcal{SK} \times \mathcal{C} \to \mathcal{M}$, where $\mathcal{ID}, \mathcal{M}, \mathcal{R}, \mathcal{C}$ and $\mathcal{SK}$ are the identities, message, randomness, ciphertext and secret keys spaces. Note that for simplicity, we made the public parameters implicit. For example, in the case of Zhao *et al.*'s IBE we have $Enc(id, m, (f(x), \overline{f}(x)) = (y^m \cdot g(x), y^m \cdot \overline{g}(x))$ and $Dec(r, (c(x), \overline{c}(x))) = m$.

Definition 1. *Let two IBE schemes $\mathcal{IBE} = (Setup, KeyGen, Enc, Dec)$ and $\mathcal{IBE}' = (Setup', KeyGen', Enc', Dec')$ be given, where $Enc : \mathcal{ID} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ and $Enc' : \mathcal{ID} \times \mathcal{M}' \times \mathcal{R} \to \mathcal{C}'$. We say that $\mathcal{IBE}$ implies $\mathcal{IBE}'$ if $Setup = Setup'$, $KeyGen = KeyGen'$ and there exists an efficient transformation $F : \mathcal{R} \times \mathcal{C} \to \mathcal{C}'$ such that for all $id \in \mathcal{ID}$, $m \in \mathcal{M}$, and $r \in \mathcal{R}$, letting $c = Enc(id, m, r)$ and $c' = F(r, c)$, it holds that $Dec'(sk, c') = m$, where sk is the secret key corresponding to id. We say that $\mathcal{IBE}$ and $\mathcal{IBE}'$ are* **functional equivalent** *if each implies the other.*

Intuitively, Definition 1 states that we can transform a valid ciphertext generated according to the specifications of the first scheme into another valid ciphertext that can be decrypted using the decryption specifications of the second scheme. On the technical side, note that we treat the randomness space as a random number generator that is queried until all the required random numbers are obtained. Also, for simplicity, we made the public parameters and the identity implicit for F.

Remark 1. Note that the IBE schemes of Zhao *et al.* and Joye are both semantically secure, provide anonymity, and support homomorphic properties. Therefore, in our analysis, we focus solely on showing functional equivalence between the two constructions.

We further analyze Zhao *et al.*'s and Joye's IBE schemes. Handling the ciphertexts as two separate components - one associated with the public key R and the other with uR - can be cumbersome. Without loss of generality, when analyzing the mathematical structure of the two IBEs, we assume that R is a square and we work only with its corresponding ciphertext. Note that the other case can be obtained by replacing R with uR.

Before proceeding with our analysis, we observe that the $Setup$ algorithm for Zhao *et al.*'s scheme also works for Joye's IBE. Conversely, Joye's $Setup$ must be augmented with a value y such that $J_p(y) = -J_q(y)$. Therefore, the two algorithms are equivalent for all practical purposes. Furthermore, the $KeyGen$ algorithms are identical. In the remainder of this work, we therefore implicitly assume that both the $Setup$ and $KeyGen$ are identical for the two encryption schemes.

Lemma 1. *Let $Enc(id, m, f(x)) = y^m \cdot g(x)$ be the encryption function of Zhao* et al.*'s IBE. Then there exist a functional equivalent IBE with the encryption function $Enc'(id, m, (a, t)) = y^m \cdot g'(x)$, where $g'(x) = a^2 t \cdot (2x + t + Rt^{-1})$ and $a, t \xleftarrow{\$} \mathbb{Z}_n$.*

Proof. According to Zhao *et al.*'s encryption algorithm, we first choose random polynomial $f(x) = ax + b$, and thus

$$g(x) = (ax + b)^2 = 2abx + b^2 + a^2 R \bmod (x^2 - R).$$

This leads to the ciphertext

$$\begin{aligned} y^m \cdot g(x) &= y^m \cdot ab \cdot (2x + (ab)^{-1} \cdot (b^2 + a^2 R)) \\ &= y^m \cdot ab \cdot (2x + a^{-1}b + ab^{-1}R). \end{aligned}$$

Note that with non-negligible probability integers a and b are invertible[2]. We further denote by $t = a^{-1}b$, and thus we have

$$y^m \cdot g(x) = y^m \cdot a^2 t \cdot (2x + t + Rt^{-1}) = y^m \cdot g'(x).$$

Therefore, we can define $Enc'(id, m, (a, t)) = y^m \cdot g'(x)$, while the decryption algorithm Dec' is identical to Dec. Let $c(x) = y^m \cdot g(x)$ and $c'(x) = y^m \cdot g'(x)$. It is easy to see that $F(f(x), c(x)) = c(x)$ and $F'((a, t), c'(x)) = c'(x)$ produce valid ciphertexts that can be decrypted to m using Dec. □

Lemma 2. *Let $Enc'(id, m, (a, t)) = y^m \cdot g'(x)$ be the encryption function of the IBE from Lemma 1. Then there exists a functional equivalent IBE with the encryption function $Enc''(id, m, (a, t)) = (y^m \cdot a^2 t, t + Rt^{-1})$.*

Proof. Let $g'(x) = a'x + b'$. We can now modify the information transmitted to the recipient. More specifically, rather than sending $(c, d) = (y^m a', y^m b')$ we can send $(c', d') = (y^m \cdot a^2 t, t + Rt^{-1})$. This changes the way we decipher. In the first

[2] otherwise, we can factor n

case we recover m by computing $Dec(r, cx+d) = J_n(cr+d)$, while in the second case by calculating

$$Dec''(r, (c', d')) = J_n(c' \cdot (2r + d')) = J_n(c') \cdot J_n(2r + d').$$

Remark that $d' = t + Rt^{-1}$ is identical to the second part of Joye's ciphertext and to decrypt we have to compute $J_n(2r+d')$ just as in Joye's IBE. Let $c'(x) = y^m \cdot g'(x)$. Note that, in this case, the transformations are $F((a,t), c'(x)) = (c'(x) \cdot (2x + t + Rt^{-1})^{-1}, t + Rt^{-1})$ and $F'((a,t), (c', d')) = c' \cdot (2x + d')$. □

Lemma 3. *Let $Enc''(id, m, (a,t)) = (y^m \cdot a^2 t, t + Rt^{-1})$ be the encryption function of the IBE from Lemma 2. Then there exist a functional equivalent IBE with the encryption function $Enc'''(id, m, t) = (y^m \cdot t, t + Rt^{-1})$.*

Proof. We can further simplify the scheme from Lemma 2 by noticing that

$$J_n(c') = J_n(y^m \cdot a^2 t) = J_n(y^m \cdot t).$$

Thus, we can just send $c'' = y^m \cdot t$ instead of c'. In this case $Enc'''(id, m, t) = (y^m \cdot t, t + Rt^{-1})$ and $Dec'''(r, (y^m \cdot t, t + Rt^{-1})) = J_n(y^m \cdot t) \cdot J_n(2r + d')$, where $d' = t + Rt^{-1}$. We define $F((a,t), (c', d')) = (c', d')$ and $F'(t, (c'', d')) = (c'', d')$. □

Proposition 1. *Zhao* et al.*'s IBE is functionally equivalent to Joye's IBE.*

Proof. To see that the IBE schemes of Zhao *et al.* and Joye differ only in their choice of encoding/compression functions, we make use of the following observation

$$J_n(y^m \cdot t) = J_n(y^m) \cdot J_n(t) = m' \cdot J_n(t), \tag{1}$$

where $m' = 1 - 2m$. Note that $J_n(2r + d') = J_n(t)$. Therefore, using Lemmas 1 to 3 and Equation (1) there are two alternative ways to sending the same message

Zhao *et al.*: Send $c' = y^m \cdot t$ and $d' = t + Rt^{-1}$, and recover $m = (1 - m')/2$ by computing

$$m' = J_n(y^m \cdot t) \cdot J_n(2r + d').$$

Joye: Send $\varepsilon = m' \cdot J_n(t)$ and $d' = t + Rt^{-1}$, and recover m' by computing

$$m' = \varepsilon \cdot J_n(2r + d').$$

Therefore, the equivalence transformations are $F(t, (c', d')) = (J_n(c'), d')$ and $F'(t, (\varepsilon, d')) = (y^\delta \cdot t, d')$, where $\delta = (1 - \varepsilon \cdot J_n(t))/2$. □

Discussion. From a bandwidth perspective, Joye's scheme is more efficient, since it transmits $2\log_2(n)+2$ bits, while Zhao *et al.*'s scheme $4\log_2(n)$. Note that the schemes' bandwidth can be improved to $2\log_2(n)$ [13] and $4\log_2(n)-4$ [10].

Let $M(\cdot)$ denote the complexity of multiplication. According to [5,11], computing a Jacobi symbol or a modular inverse can be reduced to a form of the binary *gcd* algorithm. As a result, both computations have a complexity of $\mathcal{O}(M(\mu)\log\mu)$ for μ-bit integers [5,11].

Hence, from a computational point of view, we can easily see that Zhao *et al.*'s IBE (13 multiplication - $\mathcal{O}(13M(2\lambda))$) is faster than Joye's IBE (two Jacobi symbols and two inverse computations - $\mathcal{O}(4\log(2\lambda)M(2\lambda))$). Therefore, we can view each scheme as a trade-off between bandwidth (Joye's IBE) and encryption speed (Zhao *et al.*'s IBE).

4 A Different Trade-Off Between Speed and Bandwidth

To speed up Joye's scheme, we include an element $e \in \mathbb{Z}_n \setminus J_n$ into the public key. Note that the t values can be interpreted as a Goldwasser-Micali ciphertext [12].

$Enc_{opt}(pp, id, m)$: On inputting pp, an identity id and a message $m \in \{-1, 1\}$, compute the hash value $R = H(id)$. Randomly choose two values $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_n$ and two bits $b_1, b_2 \xleftarrow{\$} \{0,1\}$. Set $b'_i = 1 - 2b_i$ and $t_i \equiv e^{b_i} x_i^2 \bmod n$ for $i \in \{1,2\}$. Also, calculate

$$\varepsilon_1 = m \cdot b'_1 \quad \text{and} \quad c_1 = t_1 + \frac{R}{t_1} \bmod n,$$
$$\varepsilon_2 = m \cdot b'_2 \quad \text{and} \quad c_2 = t_2 + \frac{uR}{t_2} \bmod n.$$

Return the ciphertext $C = (\varepsilon_1, c_1, \varepsilon_2, c_2)$.

Our analysis is similar to the one presented in [10]. For simplicity, we further assume that $p \bmod 8 \leq q \bmod 8$. Thus, we can select e as follows

$$e = \begin{cases} -1 & p \equiv -q \bmod 4, \\ 2 & p \equiv 1 \bmod 8 \text{ and } q \equiv 5 \bmod 8, \\ 2 & p \equiv 3 \bmod 8 \text{ and } q \equiv 7 \bmod 8, \\ \bar{e} & \text{otherwise}, \end{cases}$$

where $\bar{e} \xleftarrow{\$} \mathbb{Z}_n \setminus J_n$. Depending on the choice of parameters, when $b = 1$, computing the t values is equivalent to

$$t = \begin{cases} n - x^2 & e = -1, \\ x^2 + x^2 & e = 2, \\ \bar{e}x^2 & \text{otherwise}, \end{cases}$$

which leads us to the following complexities

$$\begin{cases} \mathcal{O}(M(2\lambda)) & e = -1, \\ \mathcal{O}(M(2\lambda)) & e = 2, \\ \mathcal{O}(2M(2\lambda)) & \text{otherwise.} \end{cases}$$

When $b = 0$, the complexity of computing a t value is $\mathcal{O}(M(2\lambda))$. Therefore, the complexity of this optimization is dominated by the two modular inverses (*i.e.* $\mathcal{O}(2\log(2\lambda)M(2\lambda))$. Hence, we reduced Joye's complexity by roughly half while maintaining the same ciphertext expansion.

The condition $p \equiv q \bmod 4$ implies that $J_p(-1) = J_q(-1)$, and therefore $J_n(-1) = 1$. As a result, the ciphertext space can be reduced from $\{1, \ldots, n-1\}$ to $\{1, \ldots, (n-1)/2\}$. Using this restriction, the method presented in [13] can be used to reduce the bandwidth requirement by computing

$$\begin{aligned} c'_1 &= t_1 + \frac{R}{t_1} \bmod n \quad \text{and} \quad c'_2 = t_2 + \frac{uR}{t_2} \bmod n \\ c_1 &= \min(c'_1, n - c'_1) \quad \text{and} \quad c_2 = \min(c'_2, n - c'_2), \end{aligned}$$

and returning the ciphertext $C = (\varepsilon_1, c_1, \varepsilon_2, c_2)$.

For completeness we provide in Table 1 a summary of the complexities and bandwidth requirements for the IBEs presented in this paper. Note that $\varepsilon = 1$ when $e = 2$ and $\varepsilon = 2$ when $e \neq -1, 2$.

IBE	Bandwidth		Complexity
	any p, q	$p \equiv q \bmod 4$	
Zhao *et al.*'s IBE [17]	$4\log_2(n)$ [17]	$4\log_2(n) - 4$ [10]	$\mathcal{O}(13M(2\lambda))$
Joye IBE [13]	$2\log_2(n) + 2$ [13]	$2\log_2(n)$ [13]	$\mathcal{O}(4\log(2\lambda)M(2\lambda))$
Our Proposal	$2\log_2(n) + 2$	$2\log_2(n)$	$\mathcal{O}(2(\log(2\lambda) + \varepsilon)M(2\lambda))$

Table 1. Complexities and bandwidth requirements.

In order to evaluate the practical efficiency gains of Enc_{opt} over Enc, we implemented the computation of t_1, t_2, ϵ_1, and ϵ_2 in Sage 10.7. The experiments were executed on a standard laptop running Ubuntu 25.10, equipped with an Intel i7-1165G7 CPU @2.80 GHz and 16 GB of RAM. To obtain the average running time[3] in miliseconds, we ran the code for 100 iterations using 3072/7068/15360-bit[4] inputs.

The results are provided in Table 2. Please take note that the percentages represent the time improvement relative to the original version. We can clearly see that our proposal significantly reduces encryption time by at least 35%.

[3] Measured using Sage's `time` function.
[4] According to NIST [1] this offers an 128/192/256-bit security level.

Length	Original	Proposal		
		$e = -1$	$e = 2$	$e = \bar{e}$
3072 bits	0.07354	0.04737 35.58%	0.04534 38.34%	0.04481 39.06%
7068 bits	0.18587	0.07423 60.06%	0.07338 60.52%	0.07273 60.87%
15360 bits	0.41556	0.13888 66.58%	0.13675 67.09%	0.13724 66.97%

Table 2. Running times.

References

1. Barker, E.: NIST SP800-57 Recommendation for Key Management, Part 1: General. Tech. rep., NIST (2016)
2. Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 506–522. Springer (2004)
3. Boneh, D., Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. In: CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139, pp. 213–229. Springer (2001)
4. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient Identity Based Encryption Without Pairings. In: FOCS 2007. pp. 647–657. IEEE Computer Society (2007)
5. Brent, R.P., Zimmermann, P.: An $O(M(n)\log n)$ Algorithm for the Jacobi Symbol. In: ANTS-IX. Lecture Notes in Computer Science, vol. 6197, pp. 83–95. Springer (2010)
6. Clear, M., Hughes, A., Tewari, H.: Homomorphic Encryption with Access Policies: Characterization and New Constructions. In: AFRICACRYPT 2013. Lecture Notes in Computer Science, vol. 7918, pp. 61–87. Springer (2013)
7. Clear, M., Tewari, H., McGoldrick, C.: Anonymous IBE from Quadratic Residuosity with Improved Performance. In: AFRICACRYPT 2014. Lecture Notes in Computer Science, vol. 8469, pp. 377–397. Springer (2014)
8. Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: IMACC 2001. Lecture Notes in Computer Science, vol. 2260, pp. 360–363. Springer (2001)
9. Cotan, P., Teşeleanu, G.: Generalized Galbraith's Test: Characterization and Applications to Anonymous IBE Schemes. Mathematics **9**(11), 1184 (2021)
10. Cotan, P., Teşeleanu, G.: Elementary Remarks on Some Quadratic Based Identity Based Encryption Schemes. In: SecITC 2023. pp. 26–34. Lecture Notes in Computer Science, Springer (2023)
11. Crandall, R.E., Pomerance, C.: Prime Numbers: A Computational Perspective. Springer (2005)
12. Goldwasser, S., Micali, S.: Probabilistic Encryption. Journal of Computer and System Sciences **28**, 270–299 (1984)
13. Joye, M.: Identity-Based Cryptosystems and Quadratic Residuosity. In: PKC 2016. Lecture Notes in Computer Science, vol. 9614, pp. 225–254. Springer (2016)
14. Nica, A.M., Ţiplea, F.L.: On Anonymization of Cocks' Identity-based Encryption Scheme. Computer Science Journal of Moldova **81**(3), 283–298 (2019)

15. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: CRYPTO 1984. Lecture Notes in Computer Science, vol. 196, pp. 47–53. Springer (1985)
16. Ţiplea, F.L., Iftene, S., Teşeleanu, G., Nica, A.M.: On the Distribution of Quadratic Residues and Non-Residues Modulo Composite Integers and Applications to Cryptography. Applied Mathematics and Computation **372**, 124993–124990 (2020)
17. Zhao, X., Cao, Z., Dong, X., Zheng, J.: Anonymous IBE from Quadratic Residuosity with Fast Encryption. In: ISC 2020. Lecture Notes in Computer Science, vol. 12472, pp. 3–19. Springer (2020)