Anonymous IBE from Quadratic Residuosity with Improved Performance[⋆]

Michael Clear[⋆⋆], Hitesh Tewari, Ciarán McGoldrick

School of Computer Science and Statistics,
Trinity College Dublin

Keywords: Identity Based Encryption, Anonymous IBE, Cocks Scheme, Quadratic Residuosity

Abstract. Identity Based Encryption (IBE) has been constructed from bilinear pairings, lattices and quadratic residuosity. The latter is an attractive basis for an IBE owing to the fact that it is a well-understood hard problem from number theory. Cocks constructed the first such scheme, and subsequent improvements have been made to achieve anonymity and improve space efficiency. However, the anonymous variants of Cocks' scheme thus far are all less efficient than the original. In this paper, we present a new universally-anonymous IBE scheme based on the quadratic residuosity problem. Our scheme has better performance than the universally anonymous scheme from Ateniese and Gasti (CT-RSA 2009) at the expense of more ciphertext expansion. Another contribution of this paper is a modification to a variant of the space-efficient scheme by Boneh, Gentry and Hamburg (FOCS 07) that results in an IND-ID-CPA secure IBE scheme with comparable efficiency to Cocks, but with reduced ciphertext expansion.

1 Introduction

Identity-Based Encryption (IBE) is centered around the notion that a user's public key can be efficiently derived from an identity string and system-wide public parameters. The public parameters are chosen by a Trusted Authority (TA) along with a master secret key, which is used to extract secret keys for user identities. IBE was first proposed by Shamir [2]. The first secure IBE schemes were presented by Cocks [3] (based on the quadratic residuosity problem), and Boneh and Franklin [4] (based on bilinear pairings). More recently, there have been IBE constructions based on worst-case lattice problems [5,6]. Ciphertext expansion in Cocks' scheme is large, which has hindered its practicality. Nevertheless, it is notable as being one of the few known IBE constructions based on number-theoretic assumptions. The quadratic residuosity problem on which it is based has been well studied, and is held to be a hard problem. Since it relies on such a standard assumption, Cocks' scheme has been subject to research efforts to derive more powerful primitives such as anonymous IBE or Public-key Encryption with Keyword Search (PEKS) [7]. It is known that Cocks' scheme is not anonymous.

The notion of anonymity stems from that of key privacy put forward by Bellare et al. [8]. An IBE scheme is said to be anonymous if an adversary cannot distinguish which identity was used to create a ciphertext, even if the adversary gets to choose a pair of identities to distinguish between. Anonymous IBE is a useful primitive because it can be used to facilitate searching on encrypted data, to allow anonymous broadcasts to be made in a network, and to act as a countermeasure against traffic analysis. A multitude of anonymous IBEs have been constructed based on both pairings and lattices including [4,6,7,9].

Anonymous variants of Cocks' IBE scheme whose security relies on the quadratic residuosity assumption have already been proposed in the literature [10–12]. The most efficient in terms of ciphertext size is due to Boneh, Gentry and Hamburg [11]. However, encryption time in their scheme is quartic in the security parameter, and thus has poor performance. The PEKS scheme in [10] performs better but still requires many Jacobi symbol computations when used as an anonymous IBE. The most time-efficient anonymous IBE to date was presented at CT-RSA 2009 by Ateniese and Gasti [12]. Their construction has similarly-sized ciphertexts to Cocks' original scheme while there is a drop of approximately 30% in performance compared to Cocks according to our experimental results (for a 1024-bit modulus used

[⋆] This is an extended version of a paper that appeared at Africacrypt 2014 [1].

[⋆⋆] The author's work is funded by the Irish Research Council EMBARK Initiative.

to encrypt a 128-bit symmetric key; note that IBE is typically used as part of a KEM-DEM). While this is still practical, it is desirable to obtain an anonymous IBE from quadratic residuosity whose performance is on par with the original Cocks scheme, especially for time-critical applications.

1.1 Universal Anonymity

Ateniese and Gasti's scheme also enjoys the property of universal anonymization, first introduced at Asiacrypt 2005 by Hayashi and Tanaka [13]. This property allows any party to anonymize a ciphertext without access to the secret key of the recipient. An illustrative application involves disparate systems distinguished by whether they need to know the intended recipient of encrypted data. Regulations may stipulate that some systems learn the recipient's identity. At some suitable point prior to sending the encrypted data to less trusted systems, the encrypted data can be anonymized by any party without knowledge of the secret key.

1.2 Contributions

We present a new universally anonymous IBE from quadratic residuosity whose performance closely matches that of the original Cocks scheme. Our work builds upon techniques presented in [14], especially the homomorphic property identified therein, to construct a universally anonymous variant of Cocks' scheme that achieves better performance than [12]. Unfortunately, the size of ciphertexts in our scheme is double that of Cocks, and almost double that of [12]. However, we obtain anonymity using a different approach which we believe to be conceptually simpler. We prove this system ANON-IND-ID-CPA secure in the random oracle model and provide both an analytical and experimental comparison between our approach and that of [12].

Another contribution of this paper is a security assessment of a scheme by Jhanwar and Barua [15], which in turn is a variant of the non-anonymous IBE system from [11]. We consider an alternative parameter setting that gives us IND-ID-CPA security. The resulting scheme outperforms the original Cocks scheme, and is slightly more space-efficient. Although the same ideas do not readily allow us to construct an anonymous IBE, the performance benefits provide good motivation for pursuing this in future work. Performance measurements from this scheme are reported along with the others in Section 4. However, due to space constraints, the details of this scheme are deferred to Appendix C.

1.3 Overview of Main Construction

As pointed out in previous works, the main obstacle to achieving anonymity for variants of Cocks' scheme is a property that is unconditionally satisfied for ciphertexts produced under a certain identity id. This property holds with probability negligibly close to 1/2 with respect to any other identity id'. Thus, it is possible for an adversary to readily distinguish the recipient's identity by checking whether this property holds.

We provide an informal description here to highlight the intuition behind our approach. Let $N = pq$ be an integer where p and q are prime. Let $x \in \mathbb{Z}$. We write $\left(\dfrac{x}{N}\right)$ to denote the Jacobi symbol of x mod N.

As in [12], we let $H : \{0,1\}^* \to \mathbb{Z}_N^*[+1]$ be a full-domain hash. A message bit is mapped to an element of $\{-1, 1\}$ via a mapping $\nu : \{0,1\} \to \{-1,1\}$ with $\nu(0) = 1$ and $\nu(1) = -1$.

An overview of the Cocks scheme is as follows. The Trusted Authority (TA) generates two large primes p and q, which constitute the master secret key. It outputs the public parameters $N = pq$. For any identity id, the public key corresponding to that identity is computed as $a = H(\mathsf{id})$. It will be shown later that given p and q, it is easy to derive an integer $r \in \mathbb{Z}_N$ with

$$r^2 \equiv a \mod N \quad \text{or} \quad r^2 \equiv -a \mod N.$$

Such an r is a secret key for identity id. Now encryption of a message $m \in \{0,1\}$ under identity id is straightforward: an encryptor samples two integers $t_1, t_2 \in \mathbb{Z}_N^*$ uniformly at random subject to the condition that

$$\left(\frac{t_1}{N}\right) = \left(\frac{t_2}{N}\right) = \nu(m).$$

It then computes a ciphertext $(c := t_1 + at_1^{-1}, d := t_2 - at_2^{-1})$. Decryption is also simple: set $e := c$ if $r^2 \equiv a \mod N$; otherwise set $e := d$. Then we decrypt by computing $\nu^{-1}\left(\left(\frac{e+2r}{N}\right)\right)$. However, to simplify the description, we will focus our attention on the first component of a ciphertext, namely c. In fact, the properties that we will consider concerning such c with respect to a hold analogously for d with respect to $-a$.

It was observed by Galbraith* that for any integer c generated as above, it is an invariant that

$$\left(\frac{c^2 - 4a}{N}\right) = 1.$$

We expect this to hold with probability negligibly close to 1/2 for random a. Hence, an adversary has a non-negligible advantage attacking anonymity. In the XOR-homomorphic variant from [14], the integer c is replaced by a polynomial $c(x) = c_1x + c_0$ in the quotient ring $R_a = \mathbb{Z}_N[x]/(x^2 - a)$. We can generalize the above test for polynomials in R_a. Define

$$\mathsf{GT}(a, c(x), N) = \left(\frac{c_0^2 - c_1^2 a}{N}\right).$$

Now we define two subsets $G_a = \{c(x) \in R_a : \mathsf{GT}(a, c(x), N) = 1\}$ and $\bar{G_a} = \{c(x) \in R_a : \mathsf{GT}(a, c(x), N) = -1\}$ of R_a. In addition, the set of legally generated ciphertext polynomials (i.e. those in the image of the encryption algorithm) is denoted by the set S_a. It is shown in [14] that $S_a \underset{C}{\approx} G_a$ (computationally indistinguishable) even given access to the secret key r. It is also shown that G_a is a multiplicative group in R_a and S_a is a subgroup of G_a.

The main idea behind our construction is to allow anonymized ciphertexts to be elements of $\bar{G_a}$ half of the time and G_a the other half. Therefore, the adversary cannot use Galbraith's test to distinguish identities. The main problem however is that we don't know what a "ciphertext" in $\bar{G_a}$ decrypts to without knowing the secret key. We can show that a random element in $\bar{G_a}$ can be sampled by multiplying any fixed element in $g(x) \in \bar{G_a}$ by a uniformly random element of G_a. Our idea is to derive this fixed element $g(x)$ from the user's identity using a hash function (modelled as a random oracle in the security proofs), and then multiply it by an encryption of the desired message, which lies in S_a. Since S_a and G_a are computationally indistinguishable, the resultant element $c'(x)$ is also computationally indistinguishable from a random element in $\bar{G_a}$. It can also be shown that the homomorphic property holds even between polynomials in $\bar{G_a}$ and G_a. Therefore, $c'(x)$ is an encryption of the desired message XORed with whatever $g(x)$ decrypts to. Since the decryptor can determine what $g(x)$ decrypts to, she can recover the message.

1.4 Related Work

Di Crescenzo and Saraswat [10] constructed an anonymous variant of Cocks' scheme. In fact their construction is an instance of Public-Key Encryption with Keyword Search (PEKS), a primitive introduced in [7] which allows a sender to encrypt a message with a set of hidden keywords such that a decryptor can only determine whether a specific keyword W appears in the ciphertext if she holds a secret key for W (the secret keys are computed by the TA). The scheme from [10] requires $4k$ elements of $\mathbb{Z}_N$ where k is the length of keywords represented as binary strings. Also, encryption requires $4k$ Jacobi symbol evaluations. PEKS captures anonymous IBE as a special case. Two keywords $W_{\mathsf{id}}^{(0)}$ and $W_{\mathsf{id}}^{(1)}$ representing the messages 0 and 1 respectively are associated with each identity id. Accordingly, secret keys for $W_{\mathsf{id}}^{(0)}$ and $W_{\mathsf{id}}^{(1)}$ constitute a secret key for identity id.

Boneh, Gentry and Hamburg (BGH) [11] constructed the first space-efficient variant of the Cocks scheme. The size of ciphertexts using their anonymous scheme is quite practical; an ℓ-bit message requires a ciphertext whose size is $\log_2 N + \ell + 1$ bits, which contrasts with $2\ell \cdot \log_2 N$ bits in Cocks. However, encryption in their scheme is time-consuming. Encryption time is dominated by the generation of $\ell + 1$ primes which are needed to help satisfy $\ell + 1$ equations of the form $Rx^2 + Sy^2 \equiv 1 \mod N$. It is reported in [11] that a 1024-bit prime generation takes 123.6 ms on a 2.015 GHz AMD dual-core Athlon64. To encrypt a 128 bit key, one would expect the total time to be on the order of 16 seconds on the same machine since 128 + 1 primes must be generated. However, the authors give a variant that

*Reported as emerging from personal communication in [11].

instead requires primes of length $\log_2 \sqrt{N}$ bits at the expense of an increase in ciphertext length. On the same benchmark machine, a time of 11 ms is reported for a 512-bit prime generation, which brings the total time down to ≈ 1.4 seconds. However, this variant is not anonymous.

While we have not implemented the constructions in [11], we believe they are significantly slower than the scheme in [12] and the one presented in this work. Encryption time is quartic in the security parameter as opposed to cubic for standard number-theoretic schemes. A variant of the non-anonymous BGH construction appeared in [15]. The authors of that work claim their variant achieves higher performance for both encryption and decryption as a trade-off for increased ciphertext size, which is $2 \cdot \log_2 \lceil \sqrt{\ell} \rceil + 2\ell$ bits for an ℓ-bit plaintext. We describe in Appendix C why their proof of security only goes through if a sender encrypts $\log \lambda^{\omega(1)}$ bits where λ is the security parameter. While this fact hinders the space efficiency of the scheme, our experiments show that its performance is on par with Cocks for a similar level of security. Therefore, it is arguably the most practical IBE based on quadratic residuosity, lowering ciphertext size without hindering performance.

2 Preliminaries

2.1 Notation

A quantity is said to be negligible with respect to some parameter λ, written $\mathsf{negl}(\lambda)$, if it is asymptotically bounded from above by the reciprocal of all polynomials in λ.

For a probability distribution D, we denote by $x \overset{\$}{\leftarrow} D$ that x is sampled according to D. If S is a set, $y \overset{\$}{\leftarrow} S$ denotes that y is sampled from x according to the uniform distribution on S.

The set of contiguous integers $\{1, \ldots, k\}$ for some $k > 1$ is denoted by $[k]$. Let D_1 and D_2 be distributions. We write $D_1 \approx D_2$ to denote the fact that D_1 and D_2 are statistically indistinguishable. In addition, we write $D_1 \underset{C}{\approx} D_2$ to denote the fact that both distributions are computationally indistinguishable.

2.2 Security Definition for Anonymous IBE (ANON-IND-ID-CPA)

An IBE scheme is said to be anonymous if any PPT adversary has only a negligible advantage in the following game. This is referred to as ANON-IND-ID-CPA security. At the beginning of the game, the adversary $\mathcal{A}$ is handed the public parameters. It then proceeds to make queries for secret keys corresponding to identities $\mathsf{id}_1, \ldots, \mathsf{id}_{q_1}$ for some integer q_1 that is polynomial in the security parameter. Then it sends to the challenger two identities id_0^* and id_1^* such that $\mathsf{id}_0^* \neq \mathsf{id}_1^* \neq \mathsf{id}_i$ for $1 \leq i \leq q_1$. It also sends two messages m_0 and m_1. The challenger samples a bit b uniformly, and sends the encryption of m_b under id_b^* to $\mathcal{A}$. In the final phase, $\mathcal{A}$ is allowed to query secret keys for further identities $\mathsf{id}_{q_1+1}, \ldots, \mathsf{id}_{q_1+q_2}$ where q_2 is polynomial in the security parameter, and $\mathsf{id}_0^* \neq \mathsf{id}_1^* \neq \mathsf{id}_{q_1+i}$ for $1 \leq i \leq q_2$. Finally, $\mathcal{A}$ outputs a guess b' and is said to win if $b' = b$.

2.3 Quadratic Residues and Jacobi Symbols

Let m be an integer. A quadratic residue in the residue ring $\mathbb{Z}_m$ is an integer x such that $x \equiv y^2 \mod m$ for some $y \in \mathbb{Z}_m$. The set of quadratic residues in $\mathbb{Z}_m$ is denoted $\mathbb{QR}(m)$. If m is prime, it is easy to determine whether any $x \in \mathbb{Z}_m$ is a quadratic residue. If m is an odd prime number, we can define the Legendre symbol as a function of any integer $x \in \mathbb{Z}$ with respect to m as

$$\left(\frac{x}{m}\right) = \begin{cases} 1 & \text{if } x \in \mathbb{QR}(m) \\ -1 & \text{if } x \not\equiv 0 \mod m \text{ and } x \notin \mathbb{QR}(m) \\ 0 & \text{if } x \equiv 0 \mod m \end{cases} .$$

The above function can be generalized to positive odd moduli $M = m_1^{\alpha_1} \ldots m_k^{\alpha_k}$ where $m_1, \ldots, m_k$ are prime, and $\alpha_1, \ldots, \alpha_k$ are positive integers. The generalization is called a Jacobi symbol and is defined as

$$\left(\frac{x}{M}\right) = \left(\frac{x}{m_1}\right)^{\alpha_1} \cdots \left(\frac{x}{m_k}\right)^{\alpha_k} .$$

where $\left(\dfrac{x}{m_i}\right)$ denotes the Legendre symbol of x with respect to m_i for $1 \leq i \leq k$. The subset of $\mathbb{Z}_M$ with Jacobi symbol $+1$ is denoted by $\mathbb{J}(M)$; that is, $\mathbb{J}(M) = \{x \in \mathbb{Z} : \left(\dfrac{x}{M}\right) = 1\}$. Naturally, $\mathbb{QR}(M) \subseteq \mathbb{J}(M)$.

2.4 Quadratic Residuosity Problem

Let N be a product of two odd primes p and q. The quadratic residuosity problem is to determine, given input $(N, x) \in \mathbb{Z}_N^2$ where $x \in \mathbb{J}(N)$, whether $x \in \mathbb{QR}(N)$, and it is believed to be intractable.

2.5 Blum Integers

Finally, the schemes in this paper make use of Blum integers. A Blum integer is a product of two primes that are both congruent to 3 modulo 4. As a result, we define $\mathsf{BlumGen}(1^\lambda)$ as a PPT algorithm which takes as input a security parameter λ and outputs two equally-sized primes p and q, whose lengths depend on λ, such that

$$p \equiv q \equiv 3 \pmod 4.$$

2.6 Cocks Scheme

Let $H : \{0,1\}^* \to \mathbb{J}(N)$ be a full-domain hash that sends an identity string $\mathsf{id} \in \{0,1\}^*$ to an integer in $\mathbb{Z}_N$ whose Jacobi symbol is $+1$. A secret key in Cocks' system is a Rabin signature for id. Therefore, to guarantee existential unforgeability of such signatures, the random oracle model is needed.

- $\mathsf{Cocks}.\mathbf{Setup}(1^\lambda)$:
  1. Repeat: $(p, q) \leftarrow \mathsf{BlumGen}(1^\lambda)$.

     Note that by definition of $\mathsf{BlumGen}$, we have $p \equiv q \equiv 3 \pmod 4$.
  2. $N \leftarrow pq$
  3. Output $(\mathsf{PP} := N, \mathsf{MSK} := (N, p, q))$
- $\mathsf{Cocks}.\mathbf{KeyGen}(\mathsf{MSK}, \mathsf{id})$:
  1. Parse MSK as (N, p, q).
  2. $a \leftarrow H(\mathsf{id})$.
  3. $r \leftarrow a^{\frac{N+5-p-q}{8}} \pmod N$.

     Therefore, either $r^2 \equiv a \pmod N$ or $r^2 \equiv -a \pmod N$.
  4. Output $\mathsf{sk_{id}} := (N, \mathsf{id}, r)$

  Remark 1. It is important that this algorithm always output the same square root, since otherwise N can be factored. To achieve this, one may store the root or calculate it deterministically as done so above.
- $\mathsf{Cocks}.\mathbf{Encrypt}(\mathsf{PP}, \mathsf{id}, m)$:
  1. Parse PP as N.
  2. $a \leftarrow H(\mathsf{id})$
  3. Generate $t_1, t_2 \xleftarrow{\$} \mathbb{Z}_N^*$ such that $\left(\dfrac{t_1}{N}\right) = \left(\dfrac{t_2}{N}\right) = \nu(m)$ (Recall that $\nu(m)$ maps $m \in \{0,1\}$ into $\{-1, 1\}$).
  4. Output $\psi := (t_1 + a t_1^{-1}, t_2 - a t_2^{-1})$
- $\mathsf{Cocks}.\mathbf{Decrypt}(\mathsf{sk_{id}}, \psi)$:
  1. Parse ψ as (ψ_1, ψ_2)
  2. Parse $\mathsf{sk_{id}}$ as (N, id, r)
  3. $a \leftarrow H(\mathsf{id})$

4. If $r^2 \equiv a \pmod{N}$, set $d \leftarrow \psi_1$. Else if $r^2 \equiv -a \pmod{N}$, set $d \leftarrow \psi_2$.
   Else output $\perp$ and abort.
5. Output $\nu^{-1}(\left(\frac{d+2r}{N}\right))$

3 Time-Efficient Universally Anonymous IBE

3.1 Overview of our construction

In order to explain our construction, it is necessary to first describe the XOR-homomorphic variant of Cocks' scheme from [14]. Let $R = \mathbb{Z}_N[x]$ be a polynomial ring over $\mathbb{Z}_N$. Let a be an integer in $\mathbb{J}(N)$. Then let R_a be the quotient ring $R/(x^2 - a)$. Recall the generalization of Galbraith's test to the ring R as follows.

Definition 1 (Galbraith's Test over R). *Define Galbraith's Test for the ring R as the function* $\mathsf{GT} : \mathbb{Z}_N \times R \to \{-1, 0, +1\}$ *given by*

$$\mathsf{GT}(a, c(x), N) = \left(\frac{c_0^2 - c_1^2 a}{N}\right).$$

Define the subset $G_a \subset R_a$ as follows:

$$G_a = \{c(x) \in R_a : \mathsf{GT}(a, c(x), N) = 1\}.$$

Therefore, this is the subset of R_a that passes Galbraith's test. Define the subset $\bar{G_a} \subset R_a$ as follows:

$$\bar{G_a} = \{c(x) \in R_a : \mathsf{GT}(a, c(x), N) = -1\}.$$

Correspondingly, this is the subset of R_a that fails Galbraith's test. Now define the subset $S_a \subset G_a$:

$$S_a = \{2hx + (t + ah^2t^{-1}) \in G_a \mid h \in \mathbb{Z}_N, t, (t + ah^2t^{-1}) \in \mathbb{Z}_N^*\}.$$

The subset S_a is precisely the image of the following algorithm $\mathcal{E}$ which takes as input an integer $a \in \mathbb{J}(N)$ (i.e. $\left(\frac{a}{N}\right) = 1$) along with a message bit $m \in \{0, 1\}$ and produces an element of S_a that encrypts m. This is central to the XOR-homomorphic variant of the Cocks scheme presented in [14], which is referred to as xhIBE in that paper. Like Cocks' original scheme, xhIBE requires a ciphertext to have two components. As such, $\mathcal{E}$ can be viewed as the encryption algorithm for a single component. Accordingly, to encrypt a message m in xhIBE, the sender runs $\mathcal{E}(a, m)$ and $\mathcal{E}(-a, m)$ to produce the first and second component of a ciphertext respectively. A formal description of xhIBE is given in Appendix A.

Algorithm $\mathcal{E}(a, m)$:

1. Choose an integer $t \xleftarrow{\$} \mathbb{Z}_N^*$ uniformly such that
$$\left(\frac{t}{N}\right) = \nu(m).$$
2. Choose an integer $h \xleftarrow{\$} \mathbb{Z}_N$ uniformly.
3. Compute $c(x) \leftarrow 2hx + (t + ah^2t^{-1}) \in R$
4. Repeat steps 1-4 until $(t + ah^2t^{-1}) \in \mathbb{Z}_N^*$.
5. Output $c(x)$.

With overwhelming probability, $(t + ah^2t^{-1})$ will be invertible in $\mathbb{Z}_N$.

In addition, we define a decryption algorithm $\mathcal{D}$ which takes an integer $r \in \mathbb{Z}_N$ and a polynomial in R as input, and outputs a bit $m \in \{0, 1\}$. This is defined as follows:

Algorithm $\mathcal{D}(r, c(x))$:

1. Compute $j = \left(\frac{c(r)}{N}\right) \in \{-1, 0, +1\}$.
2. If $j = 0$, output $\perp$.
3. Else output $\nu^{-1}(j) \in \{0, 1\}$.

Note that for the sake of notational convenience, it is assumed that N is an implicit input in $\mathcal{E}$ and $\mathcal{D}$. Suppose $a \in \mathbb{QR}(N)$. Then let $r \in \mathbb{Z}_N$ such that $r^2 \equiv a \mod N$. It can be shown that $\mathcal{D}(r, \cdot)$ whose domain is restricted to $S_a = \mathsf{image}(\mathcal{E}(a, \cdot))$ is a group homomorphism $(S_a, *) \to (\mathbb{Z}_2, +)$. Therefore for $m_1, m_2 \in \{0, 1\}$:

$$\mathcal{D}(r, \mathcal{E}(a, m_1) * \mathcal{E}(a, m_2)) = m_1 \oplus m_2.$$

In fact, for any $c(x), d(x) \in R$ with $\mathcal{D}(r, c(x)), \mathcal{D}(r, d(x)) \in \{0, 1\}$, it holds that

$$\mathcal{D}(r, c(x)d(x)) = \mathcal{D}(r, c(x)) \oplus \mathcal{D}(r, d(x)).$$

Naturally this means that an XOR homomorphism exists even between elements of G_a and $\bar{G_a}$.

Let $g(x) \in \bar{G_a}$. Below are some basic facts which we prove in Section 3.3.

1. $g(x)G_a = \bar{G_a}$.
2. $\{h(x) \xleftarrow{\$} \bar{G_a}\} \approx \{g(x)h'(x) \mid h'(x) \xleftarrow{\$} G_a\}$.
3. $\{h(x) \xleftarrow{\$} \bar{G_a}\} \underset{C}{\approx} \{g(x)h'(x) \mid h'(x) \xleftarrow{\$} S_a\}$.

Property 3 states that the uniform distribution defined over $\bar{G_a}$ and the distribution of multiplying $g(x)$ by uniformly random elements from S_a are computationally indistinguishable (without access to p and q).

We need two hash functions. Like Cocks' scheme, a full-domain hash $H : \{0, 1\}^* \to \mathbb{J}(N)$ is employed that maps identity strings to elements of $\mathbb{Z}_N$ whose Jacobi symbol is $+1$. Another hash function $H' : \{0, 1\}^* \to R$ is needed that maps an identity string id to an element $g(x) \in R$ such that $\mathsf{GT}(H(\mathsf{id}), g(x), N) = \mathsf{GT}(-H(\mathsf{id}), g(x), N) = -1$ i.e. the $g(x)$ is taken to pass Galbraith's test for both $a = H(\mathsf{id})$ and $-a$. Roughly speaking, an example of constructing such as hash function using H is via a form of rejection sampling i.e. to sample $g'(x)_i \xleftarrow{\$} H(\mathsf{id} \parallel i)$ for consecutive integers $i > 0$ until $\mathsf{GT}(a, g'(x)_i, N) = \mathsf{GT}(-a, g'(x)_i, N) = -1$. In the security proofs, H is modelled as a random oracle on $\mathbb{J}(N)$ and H' is modelled as a random oracle whose response when queried on id is distributed according to the uniform distribution on $\bar{G}_{H(\mathsf{id})} \cap \bar{G}_{-H(\mathsf{id})}$. To anonymize a ciphertext *component* (recall that this discussion is simplified to deal with a single component of a ciphertext corresponding to $a = H(\mathsf{id})$, the steps are repeated for the case of $-a$) $c(x)$ associated with an identity id, the following steps are performed:

1. $a \leftarrow H(\mathsf{id})$
2. $c'(x) \leftarrow \mathcal{E}(a, 0)$.
3. Uniformly sample a bit $b \xleftarrow{\$} \{0, 1\}$.
4. If $b = 0$, output $c'(x)c(x)$.
5. Else compute $g(x) \leftarrow H'(\mathsf{id})$, and output $g(x)c(x)c'(x)$.

Note that the construction is universally anonymous in that anyone can anonymize a ciphertext without having the secret key for the target identity and without access to the random coins used by the encryptor.

The decryption function $\mathcal{D}'$ for our construction is defined in terms of $\mathcal{D}$.

$$\mathcal{D}'(r, c(x)) = \begin{cases} \mathcal{D}(r, c(x)) \oplus \mathcal{D}(r, g(x)) & \text{if } c(x) \in \bar{G_a} \\ \mathcal{D}(r, c(x)) & \text{if } c(x) \in G_a \\ \perp & \text{otherwise} \end{cases}$$

3.2 Formal Description

Our scheme is referred to as UAIBE for the remainder of the paper; a formal description is as follows.
Setup(1^λ) : On input a security parameter 1^λ in unary, generate $(p, q) \leftarrow \mathsf{BlumGen}(1^\lambda)$. Compute $N = pq$. Output public parameters $\mathsf{PP} = (N, H, H')$ and master secret key $\mathsf{MSK} = (N, p, q)$, where H

is a hash function $H : \{0,1\}^* \to \mathbb{J}(N)$, and H' is a hash function $H' : \{0,1\}^* \to R$ with the property that for any identity $\mathsf{id} \in \{0,1\}^*$, $a \leftarrow H(\mathsf{id})$ and $g(x) \leftarrow H'(\mathsf{id})$, it holds that

$$\mathsf{GT}(a, g(x), N) = \mathsf{GT}(-a, g(x), N) = -1.$$

KeyGen(MSK, id) : On input master secret key $\mathsf{MSK} = (N, p, q)$ and identity $\mathsf{id} \in \{0,1\}^*$, perform the following steps:

1. Compute $a \leftarrow H(\mathsf{id}) \in \mathbb{J}(N)$.
2. If $r \in \mathbb{QR}(N)$, compute the square root $r = a^{1/2}$;
3. Else compute $r = (-a)^{1/2}$.
4. Output (N, id, r) as the secret key for identity id.

See the description of Cocks' scheme in Section 2.6 for a convenient way to compute a square root in $\mathbb{Z}_N$ deterministically.

Encrypt(PP, id, m): On input public parameters $\mathsf{PP} = (N, H, H')$, an identity $\mathsf{id} \in \{0,1\}^*$, and message $m \in \{0,1\}$ run:

1. Compute $a \leftarrow H(\mathsf{id}) \in \mathbb{J}(N)$.
2. Compute $g(x) \leftarrow H'(\mathsf{id}) \in R$.
3. Compute $c(x) \leftarrow \mathcal{E}(a, m)$.
4. Compute $d(x) \leftarrow \mathcal{E}(-a, m)$.
5. Uniformly sample two bits $v_1, v_2 \xleftarrow{\$} \{0,1\}$.
6. If $v_1 = 1$, then set $c(x) \leftarrow c(x) * g(x)$.
7. If $v_2 = 1$, then set $d(x) \leftarrow d(x) * g(x)$.
8. Output $\mathbf{c} := (c(x), d(x))$.

Decrypt($\mathsf{sk_{id}}$, c): On input a secret key $\mathsf{sk_{id}} = (N, \mathsf{id}, r)$ and a ciphertext $\mathbf{c} = (c(x), d(x))$, do:

1. Compute $a \leftarrow H(\mathsf{id}) \in \mathbb{J}(N)$.
2. Compute $g(x) \leftarrow H'(\mathsf{id}) \in R$.
3. If $r^2 \equiv a \mod N$, set $e(x) \leftarrow c(x)$. Else if $r^2 \equiv -a \mod N$, set $e(x) \leftarrow d(x)$. Else output $\perp$ and abort.
4. If $\mathsf{GT}(r^2 \mod N, e(x)) = -1$, set $e(x) \leftarrow e(x) * g(x)$.
5. Output $\mathcal{D}(r, e(x))$.

3.3 Security

Lemma 1. *Let $f(x), g(x) \in R_a$. Then $\mathsf{GT}(a, f(x)g(x), N) = \mathsf{GT}(a, f(x), N) \cdot \mathsf{GT}(a, g(x), N)$.*

Proof. Consider the product $v(x) = f(x)g(x) \in R_a$. We have that $v_0 = f_0g_0 + f_1g_1a$ and $v_1 = f_0g_1 + f_1g_0$. It is easy to verify that

$$\left(\frac{(f_0g_0 + f_1g_1a)^2 - (f_0g_1 + f_1g_0)^2a}{N}\right) = \left(\frac{(f_0^2 - af_1^2)(g_0^2 - ag_1^2)}{N}\right) = \mathsf{GT}(a, f(x), N) \cdot \mathsf{GT}(a, g(x), N).$$

. □

Lemma 2. *Let $g(x) \in \bar{G_a}$. Then $g(x) \cdot G_a = \bar{G_a}$.*

Proof. By Lemma 1, $g(x)h(x) \in \bar{G_a}$ for any $h(x) \in G_a$.

By Lemma 1 in [14], G_a is a multiplicative group in R_a. Hence, $|g(x) \cdot G_a| = |G_a|$. We claim that every $t(x) \in \bar{G_a}$ can be expressed as $g(x)t'(x)$ for some $t'(x) \in G_a$. Assume the contrary for the purpose of contradiction i.e. there exists a $t(x) \notin g(x) \cdot G_a$. It follows that $t(x) \cdot G_a \cap g(x) \cdot G_a = \emptyset$. But by Lemma 1, $t(x)^2 \in G_a$ and $g(x)t(x) \in G_a$. From the commutativity of R_a, we have $g(x) \cdot t(x)^2 = t(x) \cdot (t(x)g(x))$, which implies that $t(x) \cdot G_a \cap g(x) \cdot G_a \neq \emptyset$, a contradiction. The lemma follows. □

We include the following result from [14] that is used in the proofs below.

Corollary 1 (Corollary 2, [14]). *The distributions* $\{(N, a, t + ah^2t^{-1}, 2h) : N \leftarrow \mathsf{Setup}(1^\lambda), a \xleftarrow{\$} \mathbb{J}, t, h \xleftarrow{\$} \mathbb{Z}_N^*)\}$ *and* $\{(N, a, z_0, z_1) : N \leftarrow \mathsf{Setup}(1^\lambda), a \xleftarrow{\$} \mathbb{J}, z_0 + z_1x \xleftarrow{\$} G_a \setminus S_a\}$ *are indistinguishable assuming the hardness of the quadratic residuosity problem.*

Corollary 2. *Let* $g(x) \in \bar{G_a}$. *Then*

1. $\{h(x) \xleftarrow{\$} \bar{G_a}\} \approx \{g(x)h'(x) \mid h'(x) \xleftarrow{\$} G_a\}$.
2. $\{h(x) \xleftarrow{\$} \bar{G_a}\} \underset{C}{\approx} \{g(x)h'(x) \mid h'(x) \xleftarrow{\$} S_a\}$.

Proof. (1). From Lemma 2, each element in $\bar{G_a}$ can be represented as $g(x)h'(x)$ for a unique $h'(x) \in G_a$. Therefore, if $h'(x)$ is sampled uniformly from G_a, then $h'(x)g(x)$ is uniformly distributed in $\bar{G_a}$.

(2). By Corollary 1, $G_a \underset{C}{\approx} S_a$ without knowledge of the prime factors of N, and thus this property follows from (1). □

Theorem 1. *UAIBE is ANON-IND-ID-CPA-secure in the random oracle model assuming the hardness of the quadratic residuosity problem.*

Proof. We prove the theorem by showing that a poly-bounded adversary has a negligible advantage distinguishing between the following series of games.

Game 0 This is the ANON-IND-ID-CPA game between the challenger and an adversary $\mathcal{A}$ with the scheme UAIBE as described in Section 3.2.

Game 1 The only change in this game from Game 0 is as follows. Let b denote the bit chosen by the challenger to choose either between the tuples (id_0, m_0) or (id_1, m_1) supplied by the adversary. Let $a = H(\mathsf{id}_b)$. Instead of encrypting m_b, we instead encrypt a random bit $b' \in \{0, 1\}$ i.e. we have $c(x) \leftarrow \mathcal{E}(a, b')$ and $d(x) \leftarrow \mathcal{E}(-a, b')$.

We argue that if there is an efficient distinguisher $\mathcal{A}$ that can distinguish between Game 0 and Game 1, then there is efficient adversary $\mathcal{B}$ that can use $\mathcal{A}$ to attack the IND-ID-CPA security of xhIBE. Secret key queries from $\mathcal{A}$ are relayed to $\mathcal{B}$'s oracle. When $\mathcal{A}$ chooses its challenge tuples (id_0, m_0) and (id_1, m_1), perform the following:

1. If $b' = m_b$, output a random bit and abort.
2. Else choose challenge identity $\mathsf{id}^* = \mathsf{id}_b$.
3. When $\mathcal{B}$'s IND-ID-CPA challenger responds with a challenge ciphertext $(c(x)^*, d(x)^*)$, choose two random bits $u_0, u_1 \xleftarrow{\$} \{0, 1\}$: if $u_0 = 1$, set $c(x)^* \leftarrow c(x)^*g(x)$; if $u_1 = 1$, set $d(x)^* \leftarrow d(x)^*g(x)$ where $g(x) \leftarrow H'(\mathsf{id}^*)$ (this oracle can be provided by $\mathcal{B}$).
4. Give $(c(x)^*, d(x)^*)$ to $\mathcal{A}$, and output $\mathcal{A}$'s guess.

If $\mathcal{A}$ has advantage ϵ distinguishing games Game 0 and Game 1, then $\mathcal{B}$ has an advantage of $\frac{1}{2}\epsilon$.

Game 2 To recap, note that the challenge ciphertexts in Game 1 have the distribution $\{(c(x), d(x)) \xleftarrow{\$} S_a \times S_{-a} : a = H(\mathsf{id}_b), b \xleftarrow{\$} \{0, 1\}\}$. This is because by definition for any $a \in \mathbb{J}(N)$, we have $S_a = \mathsf{image}(\mathcal{E}(a, \cdot))$ and $S_{-a} = \mathsf{image}(\mathcal{E}(-a, \cdot))$. The next step is to replace S_a with G_a. Instead of setting $c(x) \leftarrow \mathcal{E}(a, b')$ where $a = H(\mathsf{id}_b)$, we choose $c(x) \xleftarrow{\$} G_a$.

Corollary 2 1 shows that $S_a \underset{C}{\approx} G_a$ for any $a \in \mathbb{J}(N)$ without access to the factorization of N. We follow a similar argument to the above to "embed" the challenge element from either S_a or G_a. We handle secret key queries without the factors of N by programming the oracle responses from H. Suppose the adversary queries the secret key for an identity id'. Assume without loss of generality that it first queries the random oracle H on id'. On the first such query, we uniformly sample a secret key $r' \xleftarrow{\$} \mathbb{Z}_N^*$, set $a' \leftarrow {r'}^2 \mod N \in \mathbb{J}(N)$, store the tuple (id', r', a') and return a'. This has the correct distribution and secret keys can easily be extracted. A non-negligible advantage distinguishing Game 1 and Game 2 translates to a non-negligible advantage distinguishing the distributions S_a and G_a, which contradicts Corollary 2 in [14].

Game 3 The change from Game 2 to Game 3 is similar to that from Game 1 to Game 2, namely the second ciphertext component $d(x)$ is sampled from G_{-a} instead of S_{-a} where $a = H(\mathsf{id}_b)$. The argument for indistuinguishability is analogous to that of the last game.

Game 4 This game is identical to Game 3 except that instead of setting $a \leftarrow H(\mathsf{id}_b)$, we instead set $a \xleftarrow{\$} \mathbb{J}(N)$. Furthermore, step 2 of $\mathsf{Encrypt}$ is replaced with $g(x) \leftarrow \bar{G_a} \cup \bar{G}_{-a} \in R$.

Clearly, the adversary has a zero advantage in this game since a ciphertext reveals nothing about the challenger's bit b. We now show that a ciphertext in Game 4 is indistinguishable from a ciphertext in Game 3. Observe that *each component* of the latter is computationally indistinguishable from a uniformly random element of the set of units in R. The units in R are precisely those elements $u(x)$ satisfying

$$\mathsf{GT}(a', u(x), N) \in \{-1, 1\}$$

with respect to any $a' \in \mathbb{J}(N)$; that is, the set of units is $G_{a'} \cup \bar{G}_{a'}$.

In Game 3, half of the time the ciphertext component $c(x)$ (resp. $d(x)$) is uniformly distributed in $\bar{G}_a$ (resp. $\bar{G}_{-a}$) according to Corollary 2, and the other half it is uniformly distributed in G_a (resp. G_{-a}), by definition of Game 3. Thus, each component is a uniformly random element of the set of units in R. But similarly, we have that each component of a ciphertext in Game 4 is also uniformly distributed in the set of units in R. Therefore, both games are indistinguishable to a poly-bounded adversary.

We can conclude that an adversary's advantage is negligible distinguishing between Game 0 and Game 4, which implies that its advantage attacking the ANON-IND-ID-CPA security of UAIBE is also negligible. □

3.4 Comparison with Ateniese and Gasti's Construction

Our proposed construction has several advantages. Firstly, it is arguably conceptually simpler than existing anonymous variants of Cocks' scheme. Furthermore, like the construction put forward in [12], it is universally anonymous, which may be useful in settings where messages pass through multiple systems, some of which need to know the recipient's identity whereas others should not be privy to this information. Hence, a trusted proxy can be tasked with anonymizing ciphertexts without access to the secret key. The scheme is also group-homomorphic for the XOR operation; this is useful in some settings as discussed in [14], although anonymity must be sacrificed for homomorphic operations to be performed. Another advantage of our scheme is that it faster run-time performance than other anonymous IBEs based on quadratic residuosity. We elaborate more on its performance in this section by comparing it to its nearest rival (in terms of run-tie performance), namely the Ateniese and Gasti (AG) scheme from [12]. However, the most significant downside of the scheme is its poor space efficiency; ciphertext expansion is double that of Cocks, and almost double that of AG.

3.5 Analysis of Ateniese and Gasti's Construction (AG)

Encryption in the AG scheme requires a number of Galbraith test computations per bit of plaintext. Recall that evaluating a Galbraith test entails a costly Jacobi symbol computation. The main intuition behind AG is to "embed" a Cocks ciphertext within a sequence of integers T_i. Its position, k, in such a sequence is distributed according to a geometric distribution with parameter $p = 1/2$. Furthermore, the terms $T_1, \ldots, T_{k-1}$ are chosen such that $\mathsf{GT}(a, T_i, N) = -1$ for $i \in [k-1]$. The intuition behind this approach is grounded in the fact that Galbraith's test can be shown (see Section 2.3 in [12]) to be the "best test" possible in attacking the anonymity of Cocks' scheme. Since the probability of a random element in $\mathbb{Z}_N^*$ passing Galbraith's test is 1/2, the position of the first element in a random sequence to pass Galbraith's test is distributed according to a geometric distribution with parameter $p = 1/2$. A hash function is used to generate the sequence of integers based on short binary strings incorporated in an AG ciphertext. We defer the details to Appendix B, but it sufficient here to note that ℓ is a global parameter in AG that determines the number of such binary strings (this is closely related to the number of Galbraith tests that must be performed on average during encryption).

Let Y be a random variable representing the number of Galbraith tests evaluated in AG per bit of plaintext. A lower bound for the expected value $E[Y]$ of Y can be derived as

$$E[Y] \geq 4(1 + (\log \kappa - 1) \cdot 2^{-\ell})$$

where κ is the security parameter. A rough lower bound on the variance $\mathsf{Var}(Y)$ is

$$\mathsf{Var}(Y) \geq 2^{2-2\ell}(-8 + 7 \cdot 2^{2\ell} + 2^{1+\ell} - 3 \cdot 2^{2+\ell}\ell).$$

See Appendix B for the derivations of these inequalities. Ateniese and Gasti found $\ell = 6$ to be a good compromise between ciphertext size and performance. See Appendix B for supporting analysis. Setting $\ell = 6$ results in a mean number of Galbraith tests per bit of plaintext of ≈ 4.22 with a standard deviation of ≈ 6.92. Our scheme on the other hand does not require any Galbraith test to be performed during encryption.

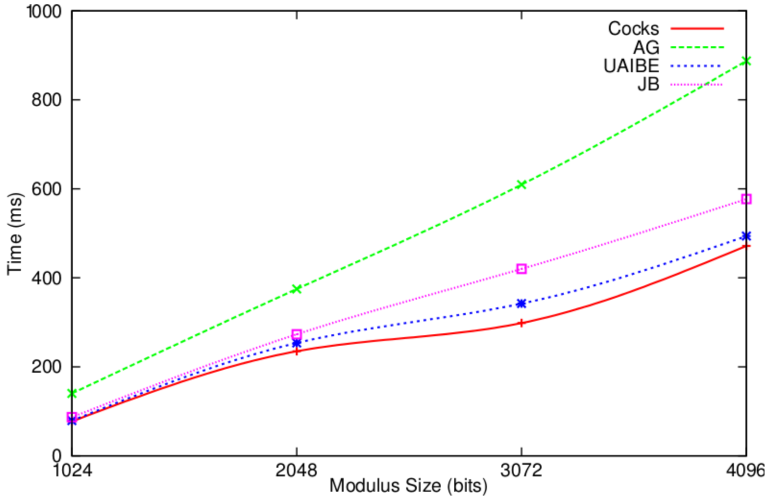

Fig. 1. Average times to encrypt a 128-bit message for Cocks, AG and UAIBE.

4 Experimental Results

To perform an empirical comparison between our scheme and AG, both schemes were implemented in C using the OpenSSL library. Our implementation was based on code provided by the authors of [12]. Our code is available at [16]. The following experiment was run for each of the four schemes: Cocks, AG, UAIBE and JB. The latter is a shorthand for our modification to the construction of Jhanwar and Barua described in Appendix C. Note that JB is not anonymous and its inclusion here is to demonstrate the fact that it achieves comparable efficiency to Cocks. Hence, AG and UAIBE are the two anonymous schemes being compared.

1. For each t in the set $\{1024, 2048, 3072, 4096\}$:
   (a) A modulus N of t bits is generated along with primes p and q that constitute the master secret key.
   (b) The public key a and secret key r are derived for some predefined identity string id. A random 128-bit message m is generated.
   (c) The following is repeated 50 times:
      i. Encrypt m under identity id to produce ciphertext c.
      ii. Decrypt c with secret key r and verify the decrypted message matches m.
      iii. The time elapsed performing step 3.(a) and 3.(b) is calculated.
   (d) An average over the times calculated in step 3.(c) is obtained.

The code was compiled with optimization flag ’-02’ using GCC version 4.4.5-8 with OpenSSL version 0.9.8o. The benchmarks were executed on a machine with 4 GB of RAM and an Intel Core i5-3340M CPU clocked at 2.70 GHz. The benchmark machine was running GNU/Linux 3.2.41 (x86-64). Our implementation however was unoptimized and did not exploit parallelization. For the interested reader, the implementation of encryption in Cocks, AG and UAIBE involved precomputation of random integers with Jacobi symbol -1 and $+1$. This is not needed for JB.

The results of the experiment (average encryption times) are shown in Figure 1. Note that UAIBE and Cocks exhibit similar performance whereas JB is only marginally less efficient than Cocks. On the other hand, AG performs notably worse than UAIBE on average. To illustrate the comparison, encryption and decryption times for all four schemes for the case of a 1024-bit modulus are presented in Table 1.

Table 1. Encryption and decryption times in milliseconds for a 128-bit message with a key size of 1024 bits, averaged over 50 runs.

Scheme	Encryption -Mean (Std Dev)	Decryption - Mean (Std Dev)
Cocks	77.39 (3.05)	13.32 (0.14)
AG	140.35 (19.22)	40.79 (1.68)
UAIBE	79.02 (3.14)	27.52 (0.41)
JB	86.78 (0.93)	21.97 (0.42)

5 Conclusions and Future Work

We have presented a new universally anonymous IBE scheme and shown it be ANON-IND-ID-CPA-secure in the random oracle model assuming the hardness of the quadratic residuosity problem. We have shown that the complexity of encryption and decryption is less than that of the universally anonymous scheme proposed in [12], albeit at the cost of increased ciphertext expansion. We hope to reduce the size of ciphertexts in future work. Furthermore, due to time constraints we have been unable to give a performance comparison between the schemes considered here and the original scheme due to Boneh, Gentry and Hamburg; we hope to explore this also as part of future work.

In addition, this paper identifies an improvement to the work of Jhanwar and Barua [15] that provides their scheme with IND-ID-CPA security. Our experimental results have shown that this scheme has comparable performance to the original Cocks scheme, and has reduced ciphertext size. Extending it so that it also supports anonymity is another goal of future work.

Acknowledgments. The authors would like to thank the anonymous reviewers for their many helpful comments.

References

1. Clear, M., Tewari, H., McGoldrick, C.: Anonymous ibe from quadratic residuosity with improved performance. In Pointcheval, D., Vergnaud, D., eds.: AFRICACRYPT. Volume 8469 of Lecture Notes in Computer Science., Springer (2014) 377–397
2. Shamir, A.: Identity-based cryptosystems and signature schemes. Lecture Notes in Computer Science **196** (1985) 47–53
3. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Proceedings of the 8th IMA International Conference on Cryptography and Coding, London, UK, Springer-Verlag (2001) 360–363
4. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag (2001) 213–229
5. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing, New York, NY, USA, ACM (2008) 197–206
6. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Proc. of Eurocrypt'10. Volume 6110 of LNCS. (2010) 553–572
7. Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In Cachin, C., Camenisch, J., eds.: EUROCRYPT. Volume 3027 of Lecture Notes in Computer Science., Springer (2004) 506–522
8. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption, Springer-Verlag (2001) 566–582
9. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In Dwork, C., ed.: CRYPTO. Volume 4117 of Lecture Notes in Computer Science., Springer (2006) 290–307
10. Crescenzo, G.D., Saraswat, V.: Public key encryption with searchable keywords based on jacobi symbols. In Srinathan, K., Rangan, C.P., Yung, M., eds.: INDOCRYPT. Volume 4859 of Lecture Notes in Computer Science., Springer (2007) 282–296

11. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, IEEE Computer Society (2007) 647–657
12. Ateniese, G., Gasti, P.: Universally anonymous IBE based on the quadratic residuosity assumption. In: Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology. CT-RSA '09, Berlin, Heidelberg, Springer-Verlag (2009) 32–47
13. Hayashi, R., Tanaka, K.: Universally anonymizable public-key encryption. In Roy, B.K., ed.: ASIACRYPT. Volume 3788 of Lecture Notes in Computer Science., Springer (2005) 293–312
14. Clear, M., Hughes, A., Tewari, H.: Homomorphic encryption with access policies: Characterization and new constructions. In Youssef, A., Nitaj, A., Hassanien, A., eds.: Progress in Cryptology AFRICACRYPT 2013. Volume 7918 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 61–87
15. Jhanwar, M., Barua, R.: A variant of boneh-gentry-hamburgs pairing-free identity based encryption scheme. In Yung, M., Liu, P., Lin, D., eds.: Information Security and Cryptology. Volume 5487 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 314–331
16. Code available on Github: `https://github.com/ciphron/anonibe`.
17. Barua, R., Jhanwar, M.: On the number of solutions of the equation rx 2 + sy 2 = 1 (mod n). Sankhya A **72** (2010) 226–236

A XOR-Homomorphic Variant of Cocks' Scheme

The XOR-Homomorphic variant of Cocks' scheme from [14] is described below with respect to the algorithms $\mathcal{E}$ and $\mathcal{D}$ defined in Section 3.1. The Setup and KeyGen algorithms are identical to those of the Cocks system, which is presented in Section 2.6. The other algorithms are defined as follows.

$\mathsf{xhIBE.Encrypt}(\mathsf{PP}, \mathsf{id}, m)$:

1. Parse PP as N.
2. $a \leftarrow H(\mathsf{id})$.
3. Compute $c(x) \leftarrow \mathcal{E}(a, m)$.
4. Compute $d(x) \leftarrow \mathcal{E}(-a, m)$.
5. Output $\psi := (c(x), d(x), a)$.

The third component a is necessary to perform homomorphic operations. See $\mathsf{xhIBE.Add}$ below.

$\mathsf{xhIBE.Decrypt}(\mathsf{sk}_{\mathsf{id}}, \psi)$:

1. Parse $\mathsf{sk}_{\mathsf{id}}$ as (N, id, r).
2. Parse ψ as $(c(x), d(x), a)$.
3. If $r^2 \equiv a \mod N$ and $\mathsf{GT}(a, c(x), N) = 1$, output $\mathcal{D}(r, c(x))$.
4. Else if $r^2 \equiv -a \mod N$ and $\mathsf{GT}(-a, d(x), N) = 1$, output $\mathcal{D}(r, d(x))$.
5. Else output $\perp$.

$\mathsf{xhIBE.Add}(\mathsf{PP}, \psi_{\mathbf{1}}, \psi_{\mathbf{2}})$:

1. Parse $\psi_{\mathbf{1}}$ as $(c_1(x), d_1(x), a)$
2. Parse $\psi_{\mathbf{2}}$ as $(c_2(x), d_2(x), a)$
3. Output $(c_1(x) * c_2(x) \pmod{x^2 - a}, d_1(x) *_{R_{-a}} d_2(x) \pmod{x^2 + a})$.

We briefly describe why the scheme is XOR-homomorphic. We will restrict our attention to the first component of a ciphertext for simplicity, since the situation is analogous for the second component with respect to $-a$ instead of a. Therefore, we assume that the secret key for identity id is $r \in \mathbb{Z}_N^*$ such that $r^2 = a \pmod N$ where $a = H(\mathsf{id})$. A plaintext bit encoded as an element of $\{-1, 1\}$ is recovered from a ciphertext polynomial $c(x)$ by computing $\left(\frac{c(r)}{N}\right)$. It is easy to see that $\left(\frac{c'(r)}{N}\right) = \left(\frac{c_1(r)}{N}\right) \cdot \left(\frac{c_2(r)}{N}\right) \in \{-1, 1\}$ where $c'(x) = c_1(x)c_2(x) \pmod{x^2 - a}$ (which is what is computed in $\mathsf{xhIBE.Add}$). Note that $(\{-1, 1\}, *)$ and $(\{0, 1\}, \oplus)$ are isomorphic.

B Expected Number of Galbraith Tests in the Ateniese and Gasti Scheme

Ateniese and Gasti proposed the following approach to anonymize a Cocks ciphertext $(c, d) \in \mathbb{Z}_N^*$ which has been computed with public key $a = H(\mathsf{id})$, . Two integers k_1 and k_2 are independently sampled according to a geometric distribution with parameter 1/2. Two sequences of integers $T_1, \ldots, T_m \in \mathbb{Z}_N^*$ and $V_1, \ldots, V_m \in \mathbb{Z}_N^*$ are randomly generated subject to the condition that for $1 \leq i < k_1$ and $1 \leq j < k_2$

$$\mathsf{GT}(a, Z_1 - T_i, N) = -1 \text{ and } \mathsf{GT}(-a, Z_2 - V_j, N) = -1 \tag{B.1}$$

where $Z_1 = c + T_{k_1}$ and $Z_2 = d + T_{k_2}$. Note that since $\mathsf{GT}(a, c, N) = 1$ and $\mathsf{GT}(-a, d, N) = 1$ by virtue of (c, d) being a Cocks ciphertext, it obviously holds that $\mathsf{GT}(a, Z_1 - T_{k_1}, N) = \mathsf{GT}(-1, Z_2 - T_{k_2}, N) = 1$. The anonymized ciphertext is outputted as $(Z_1, T_1, \ldots, T_m) \in (\mathbb{Z}_N^*)^{m+1}$ and $(Z_2, V_1, \ldots, V_m) \in (\mathbb{Z}_N^*)^{m+1}$. If m is large enough, i.e. polynomial in the security parameter, it can be shown that this construction is ANON-IND-ID-CPA-secure.

A significant disadvantage of this construction is the fact that $2(m + 1)$ elements of $\mathbb{Z}_N^*$ are needed per bit of plaintext in comparison to the 2 elements required by Cocks. To address this, Ateniese and Gasti present a more space-efficient variant.

The main difference in the space-efficient variant is in how the T_i and V_i are generated. A new global parameter $\ell \in \mathbb{N}$ is fixed. Also, the existence of a hash function $G : \{0, 1\}^* \to \mathbb{Z}_N$ is assumed. Let X be a multi-bit message. Alice chooses a random identifier MID_X when encrypting X. Now to encrypt the j-th bit of X, she computes a ciphertext

$$(Z_1, \alpha_1, \ldots, \alpha_\ell) \text{ and } (Z_2, \beta_1, \ldots, \beta_\ell)$$

where $\alpha_i, \beta_i \in \{0, 1\}^e$ for $i < \ell$, and $\alpha_\ell, \beta_\ell \in \{0, 1\}^{e'}$. Note that e and $e' > e$ are fixed global parameters. The sequences T_i and V_i are generated as follows:

$$T_i = G(\mathsf{MID} \parallel 0 \parallel \alpha_i \parallel j) \text{ and } V_i = G(\mathsf{MID} \parallel 1 \parallel \beta_i \parallel j) \tag{B.2}$$

for $1 \leq i < \ell$ and

$$T_i = G(\mathsf{MID} \parallel 0 \parallel \alpha_\ell \parallel j) \text{ and } V_i = G(\mathsf{MID} \parallel 1 \parallel \beta_\ell \parallel j) \tag{B.3}$$

for $i \geq \ell$. Alice must choose appropriate α_i and β_i in order to satisfy B.1. When $k_1 \leq \ell$ and $k_2 \leq \ell$, this is not too costly because each selection affects only one member of the respective sequence. Moreover, this will be the case with high probability for sufficiently large ℓ, However, as pointed out in [12], in the case when either $k_1 \geq \ell$ or $k_2 \geq \ell$, the cost is exponential in $k_1 - \ell$ or $k_2 - \ell$ respectively.

We now compute the average number of Galbraith tests per bit of plaintext. In fact, it suffices to restrict our attention to a single ciphertext component because we can double the result to obtain the total number of Galbraith tests.

Now the expected number of Galbraith tests is computed as follows. Let X be random variable following a geometric distribution with parameter 1/2 over the space $\{0, 1, 2, \ldots\}$. Denote by Y' the random variable that determines the number of Galbraith tests performed. There are always at least k Galbraith tests performed, where $k \overset{\$}{\leftarrow} X$. Thus,

$$E[Y'] \geq E[X] = 1.$$

Consider a random variable Z giving the number of tests performed when selecting $\alpha_1, \ldots, \alpha_{\ell-1}$. It holds that $E[Z] = 2 \cdot E[\mathsf{min}(X, \ell - 1)]$, since there are 2 expected trials per α_i for $i \leq k$ subject to the constraint that $k \leq \ell - 1$. We calculate $E[\mathsf{min}(X, \ell - 1)]$ as follows:

$$\sum_{k=0}^{\ell-1} \frac{k}{2^{k+1}} + (\ell - 1) \sum_{k=0}^{\ell-1} \frac{1}{2^{k+1}} = 1 - 2^{1-\ell}.$$

It is necessary to subtract $E[\mathsf{min}(X, \ell - 1)]$ from $E[Z]$ because these particular tests are already incorporated into $E[X]$. Therefore, we now have

$$E[Y'] \geq E[X] + E[Z] - E[\mathsf{min}(X, \ell - 1)] = 2(1 - 2^{-\ell}).$$

There is a $1/2^\ell$ chance that $k \geq \ell$. In this case, a single binary string, namely $\alpha_\ell \in \{0, 1\}^{e'}$ must be selected that satisfies $k - \ell$ Galbraith tests. Conditioned on $k \geq \ell$, the expected value of k is $\ell + 1$, and the expected number of trials per selection of α_ℓ is therefore $2((\ell + 1) - \ell) = 2$. Now it remains

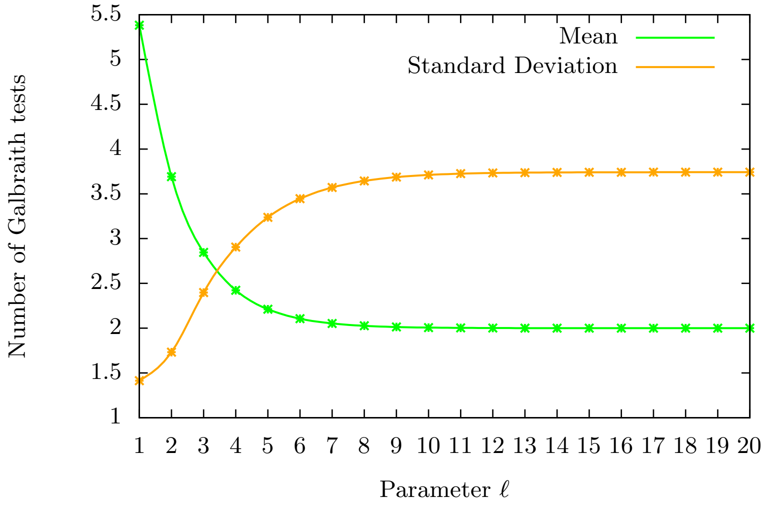

Fig. 2. Dependence on number of Galbraith tests on parameter ℓ

to compute the expected number of selections of α_ℓ. It turns out that this is equivalent to the St. Petersburg lottery. Thus, the expected value is infinite if no bound is set on k and equal to the bound otherwise, all conditioned on $k \geq \ell$. To preserve security, this bound cannot be polylogarithmic in the security parameter κ. However, setting it as such allows us to derive a (loose) lower bound on the number of selections. As a consequence, we formulate a lower bound on the number of selections as

$$\frac{\log \kappa}{2^\ell}.$$

A lower bound on the expected number of tests induced by $k \geq \ell$ is

$$\frac{\log \kappa}{2^{\ell-1}}.$$

Putting all components together yields

$$E[Y'] \geq 2(1 + (\log \kappa - 1) \cdot 2^{-\ell}).$$

A rough lower bound on the variance $\mathsf{Var}(Y')$ can be calculated in a similar manner as

$$\mathsf{Var}(Y') \geq \mathsf{Var}(Z) = 2^{1-2\ell}(-8 + 7 \cdot 2^{2\ell} + 2^{1+\ell} - 3 \cdot 2^{2+\ell}\ell).$$

Figure 2 shows approximations for the mean and standard deviation based on these lower bounds by taking the security parameter κ to be 80 (the value used in [12]). The figure supports the empirical findings of [12] from which $\ell = 6$ was found to be a good compromise between ciphertext size and performance.

C Variant of the Boneh, Gentry and Hamburg (BGH) Construction with Improved Performance

We first give an overview of the $\mathsf{BasicIBE}$ system proposed in Section 3.1 of [11], which we refer to here as BGH.

C.1 Boneh, Gentry and Hamburg BasicIBE System (BGH)

Definition 2 (Definition 3.1, [11]). *A deterministic algorithm Q that takes a tuple (N, R, S) as input, where $N \in \mathbb{Z}^+$ and $R, S \in \mathbb{Z}_N$, and outputs two polynomials $f, g \in \mathbb{Z}_N[x]$ is said to be* IBE compatible *if it satisfies the following conditions:*

1. *If $R, S \in \mathbb{QR}(N)$, then $f(r)g(s) \in \mathbb{QR}(N)$ for all square roots r and s of R and S respectively.*
2. *If $R \in \mathbb{QR}(N)$, then $f(r)f(-r)S \in \mathbb{QR}(N)$ for all square roots r of R.*

BGH can be described abstractly with respect to an algorithm Q that meets the conditions of IBE compatibility in Definition 2. The scheme can handle plaintexts of ℓ bits where ℓ is a global parameter. A user's identity id is mapped to a set of ℓ integers $R_1, \ldots, R_\ell$ via a hash function $H : \{0,1\}^* \times [\ell] \to \mathbb{J}(N)$; that is, $R_i \leftarrow H(\mathsf{id}, i)$ for $1 \in [\ell]$. A secret key for id consists of ℓ integers $r_1, \ldots, r_\ell$ where r_i a square root of R_i if $R_i \in \mathbb{QR}(N)$ and a square root of uR_i otherwise, where $u \in \mathbb{J}(N) \setminus \mathbb{QR}(N)$ is part of the public parameters.

To encrypt a message $m = m_1, \ldots, m_\ell \in \{-1, +1\}^\ell$ under identity id, the sender first generates a random $s \in \mathbb{Z}_N$ and sets $S \leftarrow s^2$. Then for $j \in [\ell]$, the sender obtains $(f_j, g_j) \leftarrow Q(N, R_j, S)$ and $(\bar{f}_j, \bar{g}_j) \leftarrow Q(N, uR_j, S)$ where $R_j \leftarrow H(\mathsf{id}, j)$, and sets $c_j \leftarrow m_j \cdot \left(\frac{g_j(s)}{N}\right)$ and $\bar{c}_j \leftarrow m_j \cdot \left(\frac{\bar{g}(s)}{N}\right)$. The ciphertext is outputted as $(S, c_1, \ldots, c_\ell, \bar{c}_1, \ldots, \bar{c}_\ell)$.

Now it can be deduced from the first condition in Definition 2 that

$$\left(\frac{g(s)}{N}\right) = \left(\frac{f(r)}{N}\right)$$

where $(f, g) \leftarrow Q(N, R, S)$ and $r^2 = R$ and $s^2 = S$. Therefore given a ciphertext $(S, c_1, \ldots, c_\ell, \bar{c}_1, \ldots, \bar{c}_\ell)$ and a secret key $r_1, \ldots, r_\ell$, a decryptor recovers the j-th bit m_j as follows, assuming without loss of generality that $r_j^2 = R_j$ (replace R_j with uR_j and c_j with $\bar{c}_j$ otherwise): compute $(f_j, g_j) \leftarrow Q(N, R_j, S)$, and set $m_j \leftarrow c_j \cdot \left(\frac{f(r)}{N}\right)$.

Correctness and IND-ID-CPA-security follow from Condition 1 and Condition 2 in definition 2 (respectively), as shown in [11].

Boneh, Gentry and Hamburg instantiate Q in their work and therefore obtain a concrete scheme that is both space-efficient and secure in the random oracle model assuming the hardness of the quadratic residuosity problem. However, their instantiation of Q is computationally expensive. Running Q is the primary bottleneck of their system. The essence of their approach entails solving equations of the form $Rx^2 + Sy^2 = 1$ modulo N to yield $(x, y) \in \mathbb{Z}_N^2$ and outputting polynomials $f(r) \leftarrow xr + 1$ and $g(s) \leftarrow 2ys + 2$. The method they propose to solve such equations involves the generation of primes, which is the main expense.

C.2 Jhanwar and Burua (JB) Variant

An alternative approach was explored in [15] based on finding a random point $(x, y) \in \mathbb{Z}_N^2$ on the curve $Rx^2 + Sy^2 = 1$ by making use of the following lemma.

Lemma 3 (Lemma 2.1, [17]). *Let N be prime. Let $R, S \in \mathbb{Z}_N$ where $S \in \mathbb{QR}(N)$. Let s be a square root of S modulo N. Then any solution $(x_0, y_0) \in \mathbb{Z}_N^2$ to the equation $Rx^2 + Sy^2 = 1$ is of the form*

$$\left(\frac{-2st}{R + St^2}, \frac{R - St^2}{s(R + St^2)}\right) \in \mathbb{Z}_N^2$$

for some $t \in \mathbb{Z}_N^*$ such that $R + St^2 \in \mathbb{Z}_N^*$.

In [15], the authors exploit Lemma 3 to generate a random solution to $Rx^2 + Sy^2 = 1$ by choosing a $t \in \mathbb{Z}_N^*$ uniformly at random. However, only the sender, who has access to s, can generate such a solution. Therefore, it is necessary to incorporate the x-coordinate in the ciphertext per bit of plaintext so that the receiver can form the polynomial $f(r) \leftarrow xr + 1$. This leads to considerable ciphertext expansion compared to BGH since 2ℓ elements $x_1, \ldots, x_\ell, \bar{x}_1, \ldots, \bar{x}_\ell$ must be incorporated in the ciphertext. To counteract this considerable blowup in ciphertext size, an optimization is employed in [15] based on a product formula due to Boneh, Gentry and Hamburg. We refer to the variant of BGH proposed in [15] by Jhanwar and Barua as JB. The main modifications to BGH employed in JB are as follows.

1. A global parameter κ is derived from ℓ (see below).
2. A user's identity is mapped to a single integer $R \in \mathbb{Z}_N$ instead of ℓ integers $R_1, \ldots, R_\ell$ in BGH. Naturally, a secret key also consists of a single integer (a square root of either R or uR).
3. During encryption, the first κ bits are encrypted by (1). choosing integers $s_1, \ldots, s_\kappa$, whose squares are denoted by $S_1, \ldots, S_\kappa$; (2) obtaining solutions (x_j, y_j) and $(\bar{x}_j, \bar{y}_j)$ to the equations $Rx^2 + S_j y^2 = 1$ and $uRx^2 + S_j y^2 = 1$ respectively via Lemma 3 for $i \in [\kappa]$; and (3). encrypting the j-th bit $m_j \in \{-1, 1\}$ by setting $c_j \leftarrow m_j \cdot \left(\frac{g_j(s_j)}{N}\right)$ and $\bar{c}_j \leftarrow m_j \cdot \left(\frac{\bar{g}_j(s_j)}{N}\right)$ where $g_j(s_j) = 2y_j s_j + 2$ and $\bar{g}_j(s_j) = 2\bar{y}_j s_j + 2$.
4. For $j > \kappa$, the j-th bit is encrypted as follows: (1). compute the unique integers $j_1, j_2 \in \{0, \ldots, \kappa - 1\}$ such that $j = j_1 \cdot \kappa + j_2$; (2). derive the solutions (x_j, y_j) and $(\bar{x}_j, \bar{y}_j)$ to the equations $Rx^2 + S_{j_1} S_{j_2} y^2 = 1$ and $uRx^2 + S_{j_1} S_{j_2} y^2 = 1$ respectively from the solutions (x_{j_1}, y_{j_1}) and (x_{j_2}, y_{j_2}) using the product formula (see Lemma 2 in [15]); (3). set $s_j \leftarrow s_{j_1} s_{j_2}$; and (4) compute c_j and $\bar{c}_j$ in the same manner as the case for $j \leq \kappa$.

Since the product formula allows a decryptor to deduce x_j from x_{j_1} and x_{j_2} for $j > \kappa$, it follows that only the integers $x_1, \ldots, x_\kappa$ and $\bar{x}_1, \ldots, \bar{x}_\kappa$ need be stored in a ciphertext.

C.3 Security Analysis of JB

The modified BGH system described above is claimed to be IND-ID-CPA secure by Theorem 2 in [15]. We make an important observation here concerning this theorem. Jhanwar and Barua propose setting $\kappa = \lceil \sqrt{\ell} \rceil$ to ensure the ciphertext size is kept "small". However, their argument that Game 5 and Game 6 in the proof of Theorem 2 are indistinguishable in the view of an adversary bounds the probability of an attacker guessing correctly by $\frac{1}{2^\kappa}$. Hence if ℓ is polylogarithmic in the security parameter, it follows that an adversary has a non-negligible advantage distinguishing both games, which invalidates the proof of security. As a result, to guarantee μ bits of security, it becomes necessary to ensure that plaintexts consist of at least μ^2 bits. Concretely, a plaintext of 800 bytes would have to be encrypted to guarantee 80 bits of security if the parameter setting proposed in [15] is employed. A more sensible setting is

$$\kappa = \mathsf{min}(\mathsf{max}(\mu, \sqrt{\ell}), \ell) \tag{C.1}$$

where μ is the desired security level and ℓ is the length of a plaintext in bits. Even with this change, the scheme still provides excellent performance. In concrete terms, we see that to encrypt a 128-bit symmetric key using a 1024-bit modulus, the ciphertext size is 20,512 bytes (note that $\kappa = 80$) in comparison to 32,768 bytes for Cocks. Furthermore, the scheme outperforms Cocks. The modified scheme with κ chosen according to Equation C.1 achieves comparable efficiency to Cocks, but with lower ciphertext expansion. Our experimental results in Section 4 provide a performance comparison.

Anonymity Given the performance benefits of this scheme, a natural question is whether an anonymous variant can be constructed. Unfortunately, attempts to exploit the same techniques to construct an anonymous IBE have not been successful. It may be tempting to start from the anonymous IBE presented in [11] and incorporate the solutions to the relevant equations in the ciphertext. However, it then becomes easy for an attacker to tell whose identity was used to create a ciphertext.