Attribute-Based Group Homomorphic Encryption and Additively Homomorphic IBE

Michael Clear* and Ciarán McGoldrick†

* Georgetown Unversity
† Trinity College Dublin
{clearm, Ciaran.McGoldrick}@scss.tcd.ie

Abstract. Group Homomorphic Encryption (GHE), formally defined by Armknecht, Katzenbeisser and Peter, is a public-key encryption primitive where the decryption algorithm is a group homomorphism. Hence it suports homomorphic evaluation of a single algebraic operation such as modular addition or modular multiplication. Most classical homomorphic encryption schemes such as as Goldwasser-Micali and Paillier are instances of GHE. In this work, we extend GHE to the attribute-based setting. We introduce and formally define the notion of Attribute-Based GHE (ABGHE) and explore its properties. Our main result is the construction of an Identity-Based Encryption (IBE) scheme supporting homomorphic addition modulo a poly-sized prime e, which is an instance of ABGHE. Our construction builds upon the IBE scheme of Boneh, LaVigne and Sabin (BLS). BLS relies on a hash function that maps identities to e^{th} residues. However there is no known way to securely instantiate such a function. Our construction extends BLS so that it can use a hash function that can be securely instantiated. We prove our scheme IND-ID-CPA secure under the (slightly modified) e^{th} residuosity assumption in the random oracle model and show that it supports a (modular) additive homomorphism. By using multiple instances of the scheme with distinct primes and leveraging the Chinese Remainder Theorem, we can support homomorphic addition modulo a "large" (i.e. superpolynomial) integer, the first such IBE scheme. We also show that our scheme for $e > 2$ is anonymous assuming the hardness of deciding solvability of a special system of multivariate polynomial equations. Finally, we define a primitive for attribute-based group homomorphisms in the multi-key setting, introduce an important security property and present a generic construction of the primitive meeting this security property.

1 Introduction

The primary subclasses of homomorphic encryption are group homomorphic encryption (GHE) and fully homomorphic encryption (FHE). In a nutshell, a public key encryption scheme is said to be *group homomorphic* if its decryption algorithm is a group homomorphism [1]. Although GHE only permits evaluation of a single algebraic operation, it is a very powerful primitive for the following reasons:

1. It is used as a building block in protocols for Private Information Retrieval [2], Electronic Voting [3–7], Oblivious Polynomial Evaluation [8], Private Outsourced Computation [9] and the Millionaire's Problem [10].
2. FHE is currently impractical for many applications, and even if it were to become more practical, it may add unnecessary overhead, especially in applications that only require a single algebraic operation.

GHE is the "classical" flavor of homomorphic encryption. It allows unbounded applications of the group operation. Goldwasser and Micali [11] constructed the first GHE scheme. The Goldwasser-Micali (GM) cryptosystem supports addition modulo 2 i.e. the XOR operation. Other additively-homomorphic GHE schemes from the literature include Benaloh [3], Naccache-Stern [12], Okamoto-Uchiyama [13], Paillier [14] and Damgård-Jurik [7]. Other instances of GHE include [15–17].

In this paper we consider GHE in the attribute-based setting. Let us first review what Attribute Based Encryption (ABE) is. Goyal et al. [18] formulated two complimentary flavors of ABE: Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). In KP-ABE, a user Alice encrypts her message with a descriptive tag or *attribute** while a Trusted Authority (TA) issues secret keys for *access policies* that permit users to decrypt ciphertexts with certain attributes. In CP-ABE, on the other hand, an encryptor specifies an access

*Some authors refer to what we call an *attribute* as a "set of attributes". The latter notion is modelled by viewing an *attribute* as a set (of "subattributes").

policy when encrypting her message, and the TA issues secret keys to parties that correspond to attributes. So the situation is the reverse of KP-ABE.

Let us consider KP-ABE in slightly more detail. When encrypting a message m, Alice chooses a descriptive attribute a from some set $\mathbb{A}$. The TA issues secret keys for *access policies* to users depending on their credentials. To be more precise, a policy $f\colon \mathbb{A} \to \{0,1\}$ can be viewed as a predicate whose domain is $\mathbb{A}$. Hence, if a user Bob is given a secret key for a policy f, he can decrypt messages with attributes that satisfy f. More precisely, let c_a be a ciphertext that encrypts the message m with some attribute $a \in \mathbb{A}$. Then Bob can recover the message m if and only if $f(a) = 1$.

Note that both forms of ABE are a generalization of Identity-Based Encryption (IBE) [19] in which the attributes are user identities (such as an email address) and there is a one-to-one correspondence between policies and attributes; that is, for each attribute a there is a unique policy f_a with $f_a(x) = 1$ if and only if $x = a$.

Why consider attribute-based GHE? One of the motivations for studying attribute-based GHE stems from the fact that it can be employed in several applications. Furthermore several applications of public-key GHE can be adapted to provide more flexible cryptographic access control by employing attribute-based GHE. We now take a look at some possible applications:

Private Information Retrieval Private Information Retrieval (PIR) [20] addresses the following problem. Suppose there is a database D with n items $x_1, \ldots, x_n$. Suppose a user wishes to query D to obtain item x_i in such a way that $i \in [n]$ remains private from D. A trivial solution is for D to send back the whole database, but this requires linear communication (in n). Hence, PIR is the problem of privately querying an item from a database with *sublinear* communication. PIR has been realized from GHE [2].

Now consider the case where the sender and receiver are different parties. Furthermore, the intended receiver may not be a known independent party with a public key, but rather one or more parties in an attribute-based infrastructure whose policies fulfill an attribute chosen by the sender that describes the data. These requirements can be satisfied by using the PIR protocol from [2] (which uses GHE) with an attribute-based GHE scheme instead of a public-key GHE scheme.

Data Aggregation in Wireless Sensor Networks There have been numerous approaches in recent years to apply IBE to Wireless Sensor Networks (WSNs). Notable contributions in this regard include [21–24]. One prevalent paradigm of a WSN involves a source node that collects sensor measurements in some environment, and forwards these measurements along an established route to a base station. Security becomes an issue in a hostile environment where malicious nodes may intercept the transmitted data. Since the autonomous sensor nodes are heavily resource-constrained, it is imperative to conserve energy where possible to prolong the lifetime and effectiveness of the network.

IBE is a natural choice for this application because nodes deployed in the field neither have to store sensitive secret keys (for symmetric encryption) nor expensively fetch, store and validate public keys for particular base stations (traditional PKI). Instead, since all nodes are identified with a unique network address, it is possible to establish well-defined identity strings. In addition, all nodes can be pre-loaded with the public parameters of the IBE scheme prior to deployment. Accordingly, in order for a node to transmit to a particular base station B with address a_B, it can derive the public key for B from a_B and the public parameters.

The most costly activity for nodes in a WSN is radio usage. Thus, it is essential to minimize the number of transmissions necessary to accomplish the network's goals. As such, a widely-used optimization strategy is aggregation of data along the path from the source to the sink (the base station). There may be a multitude of sources transmitting independent data along a particular path towards a sink. An intermediate node on the path acting as a relay, or router, may coalesce a collection of data it receives from multiple sources by performing some applicable aggregation function. An example would be to take the sum of the incoming measurements, and forward this sum to the base station. But how can this be accomplished if the data emerging from the sources is encrypted with the identity (i.e. network address in this case) of the ultimate destination, namely that of a base station? A solution to this problem is identity-based GHE with an additive homomorphism.

While identity-based GHE is advantageous to WSNs, even greater flexibility is afforded in terms of more fine-grained access control if attribute-based GHE is employed. Consider the following scenario. A WSN is deployed in an area in which sensors measure moisture and temperature. The area is divided up into N regions,

labeled $R_1, \ldots, R_N$. Each of these regions contains one or more base stations. Suppose it is sufficient for the base stations to determine the aggregate moisture and/or aggregate temperature measured in their region. Furthermore, we assume sensor nodes have the capability (such as via GPS) to determine which region they are in. To cut down on communication, aggregator nodes are employed to aggregate reported measurements that are sent by the sensor nodes as they are transmitted en-route to a base station. To minimize data exposure in the presence of adversarial nodes, an attribute-based GHE scheme is deployed within the WSN. The attribute-based GHE scheme supports an additive homomorphism to satisfy the needs of aggregation as described. Every node, prior to its deployment, is pre-loaded with the public parameters of the scheme. The WSN administrator operates the TA offline, unconnected to the WSN.

A *plaintext* in the system is an integer from the set $\mathcal{M} \triangleq \{0, \ldots, M\}$; sensor readings are assumed to take on values in the range $0, \ldots, M$ for some M. An *attribute* in the system is of the form $(\mathsf{type}, \mathsf{region})$ where $\mathsf{type} \in \{\mathsf{MOISTURE}, \mathsf{TEMPERATURE}\}$ and $\mathsf{region} \in \{R_1, \ldots, R_N\}$. Let $\mathbb{A}$ be the set of attributes. Let $\mathbb{F}$ be a class of access policies modeled as predicates (i.e. Boolean-valued functions), where every policy $f : \mathbb{A} \to \{0, 1\} \in \mathbb{F}$ maps an attribute to $\{0, 1\}$ (denoting false and true respectively).

Adhering to the principle of least privilege, a base station B in region R_1, whose purpose is to monitor moisture content in that region, is issued a secret key for the following policy, denoted f:

$$f(a := (\mathsf{type}, \mathsf{region})) \triangleq (\mathsf{type} = \mathsf{MOISTURE}) \wedge (\mathsf{region} = R_1).$$

Another base station B' whose purpose is to monitor both moisture and temperature in the regions R_1 and R_2 is issued a secret key for the following policy, denoted f':

$$f'(a := (\mathsf{type}, \mathsf{region})) \triangleq (\mathsf{type} = \mathsf{MOISTURE} \ \vee \ \mathsf{type} = \mathsf{TEMPERATURE}) \\ \wedge \ (\mathsf{region} = R_1 \ \vee \ \mathsf{region} = R_2).$$

Suppose an aggregator node near B' receives encrypted readings from two different sensor nodes. The first reading originated in R_1 and has the attribute $a_1 := (\mathsf{type} := \mathsf{MOISTURE}, \mathsf{region} := R_1)$ while the second reading originated in R_2 and has the attribute $a_2 := (\mathsf{type} := \mathsf{MOISTURE}, \mathsf{region} := R_2)$. With an attribute-based GHE scheme the aggregator can add the two encrypted readings homomorphically irrespective of the fact that they have *different* attributes. Suppose it subsequently forwards the encrypted result to B'. Intuitively, B' should be able to recover the plaintext because its policy f' authorizes both attributes; that is, we have $f'(a_1) = f'(a_2) = 1$. In contrast, if the base station B gets hold of the ciphertext, it should not be able to recover the plaintext because its policy f is satisfied by only one of the attributes, namely a_1.

Participatory Sensing In participatory sensing, users with personal mobile devices, such as phones that are equipped with sensors, share data acquired from these sensors with a network. We refer to these entities as mobile nodes. Other entities, called queriers, subscribe to receive certain types of data. De Cristofaro and Soriente [25–27] presented a model called PEPSI for participatory sensing with privacy-enhanced capabilities using provably-secure cryptographic primitives. Günther et al. [28] improved the security of PEPSI by making it resistant to collusion between mobile nodes and queriers. An interesting feature that Günther et al. incorporate in their model, called PEPSIco, is support for data aggregation, which they argue is useful to reduce the amount of information to be sent to queriers, cutting down on communication cost. Günther et al. give a realization of PEPSIco with data aggregation based on additively homomorphic IBE. This is an application where identity-based GHE would be a perfect fit. A possible avenue for future work would be to consider what other functionality could be achieved if attribute-based GHE were employed.

1.1 Contributions

Our first contribution is a formal definition of Attribute-Based Group Homormophic Encryption (ABGHE) along with an analysis of its properties. We then examine existing schemes from the literature and show that they meet our definition of ABGHE for either an additive or multiplicative homomorphism. Most such schemes are in fact Identity-Based Group Homomorphic Encryption (IBGHE) schemes i.e. instances of ABGHE whose class of access policies are point functions. We then present a possibility result for IBGHE from indistinguishability obfuscation for any group $(S, \cdot)$ for which a (public-key) GHE scheme exists.

Our central contribution is the construction of an IBGHE for addition modulo a poly-sized prime e. Our construction builds on the IBE scheme of Boneh, LaVigne and Sabin (BLS) [29], which uses a hash function

that maps identities to e^{th} residues; there is no known way to securely instantiate such a function and therefore BLS is useful only as a public-key scheme - in fact, the proof of security in [29] treats it as such. We extend BLS so that it uses a hash function that can be securely instantiated. We prove our scheme IND-ID-CPA secure under a (slightly modified) e^{th} residuosity assumption in the random oracle model. We then show that the scheme supports homomorphic addition modulo a poly-sized prime e and prove that it satisfies the properties of an ABGHE. By using multiple instances of the scheme with distinct primes and leveraging the Chinese Remainder Theorem, we can support homomorphic addition modulo a "large" (i.e. superpolynomial) integer, the first such IBE scheme, solving an open problem mentioned in [30]. We also show that our scheme for $e > 2$ is anonymous assuming the hardness of deciding solvability of a special system of multivariate polynomial equations.

Finally we explore group homomorphisms in the multi-key setting; that is, when secret keys for multiple policies can be passed to the decryption algorithm. We give a formal definition of a flavor of ABGHE with support for multiple decryption keys which we call Multi-Key Attribute-Based Homomorphic Encryption for a group (ABHEg). We present a generic construction of this primitive from any ABGHE scheme and show that it meets an important security property that prevents a decryptor from learning information about the inputs to a homomorphic evaluation beyond what is revealed by the result.

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 $x \xleftarrow{\$} D$ that x is sampled according to D. If S is a set, $y \xleftarrow{\$} S$ denotes that y is sampled from x according to the uniform distribution on S.

The support of a predicate $f : A \to \{0,1\}$ for some domain A is denoted by $\mathsf{supp}(f)$, and is defined by the set $\{a \in A : f(a) = 1\}$.

The set of contiguous integers $\{1, \ldots, k\}$ for some $k > 1$ is denoted by $[k]$.

2.2 Attribute Based Encryption

Definition 1. *A (Key-Policy) Attribute-Based Encryption (ABE) scheme is a tuple of PPT algorithms (G, K, E, D) defined with respect to a message space $\mathcal{M}$, an attribute space $\mathbb{A}$, class of access policies $\mathbb{F}$ and a ciphertext space $\hat{\mathcal{C}}$ as follows:*

- $\mathsf{G}(1^\lambda)$*:*
  On input (in unary) a security parameter λ, generate public parameters PP and a master secret key MSK. Output $(\mathsf{PP}, \mathsf{MSK})$.
- $\mathsf{K}(\mathsf{MSK}, f)$*:*
  On input master secret key MSK and an access policy (predicate) $f : \mathbb{A} \to \{0,1\} \in \mathbb{F}$: derive and output a secret key sk_f for predicate f.
- $\mathsf{E}(\mathsf{PP}, a, m)$*:*
  On input public parameters PP, an attribute $a \in \mathbb{A}$, and a message $m \in \mathcal{M}$, output a ciphertext $c \in \mathcal{C} \subseteq \hat{\mathcal{C}}$ that encrypts m under identity a.
- $\mathsf{D}(\mathsf{sk}_f, c)$*:*
  On input a secret key sk_f for predicate $f \in \mathbb{F}$ and a ciphertext $c \in \hat{\mathcal{C}}$, output m' if c is a valid encryption under some attribute a and $f(a) = 1$; output a failure symbol $\perp$ otherwise.

Identity-Based Encryption (IBE) is a special case of ABE where the attributes correspond to identities (such as an email address) and there is a one-to-one correspondence between attributes and policies i.e. for each attribute $a \in \mathbb{A}$, there is a unique policy $f \in \mathbb{F}$ with $f(x) = 1$ iff $x = a$.

2.3 Public-Key GHE

An important subclass of partial homomorphic encryption is the class of public-key encryption schemes that admit a group homomorphism between their ciphertext space and plaintext space. This class corresponds to what is considered "classical" HE [1], where a single group operation is supported, most usually addition. Gjøsteen [15] examined the abstract structure of these cryptosystems in terms of groups, and characterized their security as relying on the hardness of a subgroup membership problem. Armknecht, Katzenbeisser and Peter [1] rigorously formalized the notion, and called it *group homomorphic encryption* (GHE). We recap with the formal definition of GHE by Armknecht, Katzenbeisser and Peter [1].

Definition 2 (GHE, Definition 1 in [1]). *A public-key encryption scheme $\mathcal{E} = (G, E, D)$ is called* group homomorphic*, if for every $(\mathsf{pk}, \mathsf{sk}) \leftarrow G(1^\lambda)$, the plaintext space $\mathcal{M}$ and the ciphertext space $\hat{\mathcal{C}}$ (written in multiplicative notation) are non-trivial groups such that*

- *the set of all encryptions $\mathcal{C} := \{c \in \hat{\mathcal{C}} \mid c \leftarrow E_{\mathsf{pk}}(m), m \in \mathcal{M}\}$ is a non-trivial subgroup of $\hat{\mathcal{C}}$*
- *the restricted decryption $D^*_{\mathsf{sk}} := D_{\mathsf{sk}|\mathcal{C}}$ is a group epimorphism (surjective homomorphism) i.e.*

$$D^*_{\mathsf{sk}} \text{ is surjective and } \forall c, c' \in \mathcal{C} \ : \ D_{\mathsf{sk}}(c \cdot c') = D_{\mathsf{sk}}(c) \cdot D_{\mathsf{sk}}(c')$$

- sk *contains an efficient* decision function $\delta : \hat{\mathcal{C}} \to \{0, 1\}$ *such that*

$$\delta(c) = 1 \iff c \in \mathcal{C}$$

- *the decryption on $\hat{\mathcal{C}} \setminus \mathcal{C}$ returns the symbol $\perp$.*

We are interested in schemes whose plaintext space forms a group and which allow that operation to be homomorphically applied an unbounded number of times. There exist schemes however that do not satisfy all the requirements of GHE, namely their ciphertext space does not form a group but instead forms a quasigroup (a group without associativity). We can define what we call Quasigroup Homomorphic Encryption (QHE) analogously to Definition 2 by replacing the term 'group' with 'quasigroup' in the definition. An example of such a scheme is the public-key[†] variant of Cocks' IBE [31], which was shown to be inherently XOR-homomorphic by Joye [32].

2.4 e^{th} Residuosity

An integer x is said to be a quadratic residue modulo an integer m if x is congruent to a square modulo m. We denote the set of quadratic residues modulo p as $\mathbb{QR}(m)$. The Legendre symbol of an integer x modulo a prime p is defined as

$$\left(\frac{x}{p}\right) = \begin{cases} 0 & \text{if } p|x \\ 1 & \text{if } x \in \mathbb{QR}(p) \\ -1 & \text{otherwise} \end{cases}$$

The Jacobi symbol generalizes the Legendre symbol to composite modulii. For a composite modulus $m = p_1^{a_1} \cdots p_n^{a_n}$, it is defined as

$$\left(\frac{x}{m}\right) = \left(\frac{x}{p_1}\right)^{a_1} \cdots \left(\frac{x}{p_n}\right)^{a_n}$$

We now generalize quadratic residues to e^{th} power residues. We define the e^{th} power residue symbol as follows:

Definition 3 (Based on Definition 4.1 in [33]). *Let $e \geq 2$ be an integer, and let $\zeta_e \in \bar{\mathbb{Q}}$ be a primitive e^{th} root of unity (note that $\bar{\mathbb{Q}}$ is the algebraic closure of $\mathbb{Q}$). Let K be the number field $\mathbb{Q}(\zeta_e)$, and let $\mathcal{O}_K = \mathbb{Z}[\zeta_e]$ be the ring of integers in K. Let $\mathfrak{p}$ be a prime ideal of $\mathcal{O}_K$ that does not contain e. For $x \in \mathcal{O}_K$, the e^{th} power residue symbol of $x \mod \mathfrak{p}$, denoted $\left(\frac{x}{\mathfrak{p}}\right)_e$ is defined as*

$$\left(\frac{x}{\mathfrak{p}}\right)_e = \begin{cases} 0 & \text{if } x \in \mathfrak{p} \\ \zeta_e^i & \text{if } x \notin \mathfrak{p} \end{cases}$$

where i is the unique integer modulo e such that $\zeta_e^i \equiv x^{(\mathcal{N}(\mathfrak{p})-1)/e} \pmod{\mathfrak{p}}$ and $\mathcal{N}(\mathfrak{p})$ is the norm of $\mathfrak{p}$.

[†]Every IBE can be viewed as a public-key scheme

Let $e \geq 2$ be an integer. Let N be a positive integer. An integer $x \in \mathbb{Z}_N^*$ is said to be an e^{th} residue modulo N if there is an integer $y \in \mathbb{Z}_N^*$ such that $y^e \equiv x \mod N$. We denote the set of e^{th} residues in $\mathbb{Z}_N^*$ by $\mathbb{ER}(N)$. A superset of $\mathbb{ER}(N)$ is the set of integers in $\mathbb{Z}_N^*$ with a power residue symbol of 1, which we denote as $\mathbb{PR}(N)$.

Definition 4 (e^{th} Residuosity (ER) Assumption). *For a PPT algorithm* $\mathsf{RSAgen}(\lambda)$ *that generates two equally sized primes p and q, the e^{th} residuosity assumption is that the following two distributions are computationally indistinguishable*[‡]

$$\{(N, v) : (p, q) \leftarrow \mathsf{RSAgen}(\lambda), N \leftarrow pq, v \xleftarrow{\$} \mathbb{ER}(N)\}$$

$$\approx_C$$

$$\{(N, v) : (p, q) \leftarrow \mathsf{RSAgen}(\lambda), N \leftarrow pq, v \xleftarrow{\$} \mathbb{PR}(N) \setminus \mathbb{ER}(N)\}.$$

Let $N = pq$ be a product of two primes p and q with $p \equiv q \equiv 1 \mod e$. An e^{th} root of unity in $\mathbb{Z}_N$ is an integer μ such that $\mu^e \equiv 1 \mod N$. The trivial root of unity is 1. A root of unity μ is said to be *degenerate* if either $\mu \equiv 1 \mod p$ or $\mu \equiv 1 \mod q$ since given such a μ one can trivially learn the factorization of N. For one of the schemes in this work, it is necessary to publish a nontrivial, non-degenrate root of unity as part of the public parameters. This is in order to compute the e^{th} power residue symbol which is needed for the scheme. It is believed that revealing such a root of unity does not make factorization of N easier, but nevertheless it serves as additional information for the adversary, and therefore must be made explicit in the assumption we use for security. Hence, we follow [29] and modify the ER assumption to incorporate this information.

Definition 5 (Modified e^{th} Residuosity (MER) Assumption, [29]). *Let $\mathcal{Z}$ be the set of nontrivial, non-degenerate roots of unity in $\mathbb{Z}_N$. For a PPT algorithm* $\mathsf{RSAgen}(\lambda)$ *that generates two equally sized primes p and q, the modified e^{th} residuosity assumption is that the following two distributions are computationally indistinguishable*

$$\{(N, v, \mu) : (p, q) \leftarrow \mathsf{RSAgen}(\lambda), N \leftarrow pq, v \xleftarrow{\$} \mathbb{ER}(N), \mu \xleftarrow{\$} \mathcal{Z}\}$$

$$\approx_C$$

$$\{(N, v, \mu) : (p, q) \leftarrow \mathsf{RSAgen}(\lambda), N \leftarrow pq, v \xleftarrow{\$} \mathbb{PR}(N) \setminus \mathbb{ER}(N), \mu \xleftarrow{\$} \mathcal{Z}\}.$$

3 Attribute-Based GHE

3.1 Formal Definition

In this section, we present a formal definition of attribute-based GHE (ABGHE), extending Definition 2.

Definition 6 (Attribute Based Group Homomorphic Encryption (ABGHE)). *Let $\mathcal{E} = (G, K, E, D)$ be an ABE scheme with message space $\mathcal{M}$, attribute space $\mathbb{A}$, ciphertext space $\widehat{\mathcal{C}}$ and class of predicates $\mathbb{F}$. The scheme $\mathcal{E}$ is group homomorphic if for every $(\mathsf{PP}, \mathsf{MSK}) \leftarrow G(1^\lambda)$, every $f \in \mathbb{F} : \mathsf{supp}(f) \neq \emptyset$, and every $\mathsf{sk}_f \leftarrow K(\mathsf{MSK}, f)$, the message space $(\mathcal{M}, \cdot)$ is a non-trivial group, and there is a binary operation $* : \widehat{\mathcal{C}}^2 \to \widehat{\mathcal{C}}$ such that the following properties are satisfied for the restricted ciphertext space $\widehat{\mathcal{C}_f} = \{c \in \widehat{\mathcal{C}} : D_{\mathsf{sk}_f}(c) \neq \perp\}$:*

GH.1: *The set of all encryptions $\mathcal{C}_f = \{c \mid c \leftarrow E(\mathsf{PP}, a, m), a \in \mathsf{supp}(f), m \in \mathcal{M}\} \subseteq \widehat{\mathcal{C}_f}$ is a non-trivial group with respect to the operation $*$.*
GH.2: *The restricted decryption $D^*_{\mathsf{sk}_f} := D_{\mathsf{sk}_f | \mathcal{C}_f}$ is surjective and $\forall c, c' \in \mathcal{C}_f \quad D_{\mathsf{sk}_f}(c * c') = D_{\mathsf{sk}_f}(c) \cdot D_{\mathsf{sk}_f}(c')$.*

Let us consider Definition 6 in more detail. Let $f \in \mathbb{F}$ be any policy that is satisfied by at least one attribute i.e. $\mathsf{supp}(f) \neq \emptyset$. Furthermore, D_{sk_f} is the decryption function indexed by some secret key sk_f for f. We restrict ourselves to the set of ciphertexts $\widehat{\mathcal{C}_f} \in \widehat{\mathcal{C}}$ that decrypt to a plaintext under D_{sk_f}. In other words, this is the set of ciphertexts that do not yield the failure symbol $\perp$ on decryption with D_{sk_f}. Now the set of

[‡]Any PPT distinguisher has only a negligible advantage (in λ) of distinguishing the distributions.

honest encryptions with any attribute satisfying f (let this be $\mathcal{C}_f$) should be a subset of $\widehat{\mathcal{C}_f}$. This is captured by GH.1 in Definition 6. However, GH.1 makes an even stronger demand. It requires that $\mathcal{C}_f$ be a non-trivial group with respect to the operation $*$. The homomorphism is described by GH.2. In our case, it means that for any honestly generated ciphertexts $c, c' \in \mathcal{C}_f$, we have $D_{\mathsf{sk}_f}(c * c') = D_{\mathsf{sk}_f}(c) \cdot D_{\mathsf{sk}_f}(c')$.

Is $\widehat{\mathcal{C}_f} = \mathcal{C}_f$? This is not always the case. This is exemplified by the identity-based XOR-homomorphic construction from [30] where elements of $\widehat{\mathcal{C}_f} \setminus \mathcal{C}_f$ are computationally indistinguishable from $\mathcal{C}_f$ without the master secret key. This illustrates that an efficient decision function cannot decide between elements of $\widehat{\mathcal{C}_f} \setminus \mathcal{C}_f$ and $\mathcal{C}_f$ in all cases. Let sk_f be any secret key for a policy f. Suppose there is a decision function $\delta_f : \widehat{\mathcal{C}} \to \{0,1\}$ embedded in sk_f that can determine whether an element of $\widehat{\mathcal{C}}$ is an honest encryption that is decryptable by f i.e. $\delta_f(c) = 1 \iff c \in \mathcal{C}_f$. In this case, the decryption function D_{sk_f} simply outputs $\perp$ on input c if and only if $\delta_f(c) = 0$; it outputs the recovered plaintext otherwise. As a result, we then indeed have that $\widehat{\mathcal{C}_f} = \mathcal{C}_f$. Armknecht et al. introduced the decision function in their definition of GHE for the public-key setting in order to assist their characterization of IND-CCA1 security. However, an efficient decision function does not always exist in the ABE setting. The reason for this is that the decryptor is only given *partial* secret key information sufficient for her policy f, but other information may remain computationally hidden from her without the master secret key. Therefore, a decryptor may not be able to efficiently tell whether a ciphertext c is in $\mathcal{C}_f$.

It is always the case that a scheme can be adapted so that $(\hat{\mathcal{C}_f}, *)$ forms a group (or is computationally indistinguishable from one without the master secret key) provided $(\mathcal{C}_f, *)$ is a group. This can be seen by considering the following two cases. In the first case there is an efficient decision function embedded in a description of sk_f that can distinguish elements not in $\mathcal{C}_f$ and thus output $\perp$ on decryption of these elements. Therefore we have $\hat{\mathcal{C}_f} = \mathcal{C}_f$. In the second case, no such decision function exists and the sets $\hat{\mathcal{C}_f} \setminus \mathcal{C}_f$ and $\mathcal{C}_f$ are computationally indistinguishable, which means that $(\hat{\mathcal{C}_f}, *)$ is computationally indistinguishable from a group without the master secret key (as otherwise an efficient decision function would exist).

3.2 Properties

We will now establish some properties about ABGHE schemes. To help us in this task, we first define a particular ABGHE scheme which we make reference to throughout the section. Let $\mathcal{E} = (G, K, E, D)$ be a ABGHE scheme satisfying Definition 6 with message space $(\mathcal{M}, \cdot)$, attribute space $\mathbb{A}$, access policies $\mathbb{F}$, ciphertext space $\widehat{\mathcal{C}}$ and binary operation $* : \widehat{\mathcal{C}} \times \widehat{\mathcal{C}} \to \widehat{\mathcal{C}}$. Fix any $(\mathsf{PP}, \mathsf{MSK}) \leftarrow G(1^\lambda)$. Note that the identity element of $(\mathcal{M}, \cdot)$ is written as $1 \in \mathcal{M}$. We assume that $\mathbb{F}$ is free of any *degenerate* policies; that is, policies f with $f(a) = 0 \; \forall a \in \mathbb{A}$.

Partition of Access Policies A fundamental property of an ABGHE scheme is that its class of access policies $\mathbb{F}$ can be partitioned into equivalence classes via a natural relation $\sim$. The relation is defined for any $f, g \in \mathbb{F}$ as

$$f \sim g \quad \text{iff } \mathsf{supp}(f) \cap \mathsf{supp}(g) \neq \emptyset.$$

Now $\sim$ is clearly reflexive and symmetric, but it is not necessarily transitive in the case of an arbitrary ABE scheme. However if the scheme is group homomorphic, i.e. it satisfies Definition 6, then $\sim$ is also transitive, and hence an equivalence relation. We now show this formally.

Lemma 1 (transitivity of $\sim$). *Let $f_1, f_2, g \in \mathbb{F}$ such that $\mathsf{supp}(f_1) \cap \mathsf{supp}(g) \neq \emptyset$ and $\mathsf{supp}(f_2) \cap \mathsf{supp}(g) \neq \emptyset$. Then $\mathsf{supp}(f_1) \cap \mathsf{supp}(f_2) \neq \emptyset$.*

Proof. By GH.1 in Definition 6 we have that $\mathcal{C}_{f_1} \subset \widehat{\mathcal{C}}$, $\mathcal{C}_{f_2} \subset \widehat{\mathcal{C}}$ and $\mathcal{C}_g \subset \widehat{\mathcal{C}}$ are non-trivial groups under the operation $*$. Let e be the identity element of $\mathcal{C}_g$. For any $x \in \mathcal{C}_{f_1} \cap \mathcal{C}_g$ we have $x * e = x$. Therefore $e \in \mathcal{C}_{f_1}$. Analogously, we have $e \in \mathcal{C}_{f_2}$. It follows from GH.2 in Definition 6 that $D_{\mathsf{sk}_{f_1}}(e) = D_{\mathsf{sk}_{f_2}}(e) = 1 \in \mathcal{M}$ for any $\mathsf{sk}_{f_1} \leftarrow K(\mathsf{MSK}, f_1)$ and $\mathsf{sk}_{f_2} \leftarrow K(\mathsf{MSK}, f_2)$. It follows that e is associated with an attribute that satisfies both f_1 and f_2. □

Each equivalence class in $\mathbb{F}/\sim$ consists of policies linked together because their support sets share a common attribute. The equivalence classes in $\mathbb{F}/\sim$ correspond to disjoint sets of attributes. For example, in the case of IBE, we have $|\mathbb{F}/\sim| = |\mathbb{A}|$. In contrast, for a more complex class of access policies, we may have $|\mathbb{F}/\sim| = 1$. This is particularly true when there is an access policy that is satisfied by all attributes. The following corollary follows immediately from Lemma 1.

Corollary 1. *If the tautology predicate* $\top$ *(i.e.* $\top(a) = 1\ \forall a \in \mathbb{A}$*) is in* $\mathbb{F}$*, then there exists an attribute* $\mathfrak{a} \in \mathbb{A}$ *such that* $f(\mathfrak{a}) = 1\ \forall f \in \mathbb{F}$*.*

The corollary tells us that if there is a policy that is satisfied by every attribute, then there is at least one attribute $\mathfrak{a}$ that satisfies every policy.

Multiplying a ciphertext c by a ciphertext created with attribute $\mathfrak{a}$ preserves the access restrictions of the ciphertext c. In other words, suppose d is an encryption under attribute $\mathfrak{a}$ and one obtains $e = c * d$, then any policy f that can decrypt c can also decrypt e. This follows immediately from GH.2. Thus encryptions under attribute $\mathfrak{a}$ can be seen as "neutral". In schemes that are attribute-hiding (i.e. where the attribute associated with a ciphertext is hidden) this is advantageous as it possible to encrypt plaintexts under the netural attribute $\mathfrak{a}$ in order to perform evaluation with some ciphertext without affecting the access permissions of the ciphertext.

Each equivalence class in $\mathbb{F}/\sim$ has its own identity element. For all policies $f_1, f_2 \in \mathbb{F}$ with $\mathsf{supp}(f_1) \subset \mathsf{supp}(f_2)$, then $\mathcal{C}_{f_1}$ is a subgroup of $\mathcal{C}_{f_2}$.

Subgroup Membership Problem Armknecht et al. characterize the semantic security of (public-key) GHE as a subgroup membership problem, which can be generalized easily to the attribute-based setting. To describe this, we first establish some notation. For any attribute $a \in \mathbb{A}$ and any plaintext $\mu \in \mathcal{M}$, we define the set $\mathcal{C}_\mu^{(a)}$ as the image of $E_{\mathsf{PP}}(a, \mu)$ i.e. the set of legally generated encryptions of μ under attribute a. In addition, we define $\mathcal{C}^{(a)} = \bigcup_{\mu \in \mathcal{M}} \mathcal{C}_\mu^{(a)}$. Recall that we are using multiplicative notation for groups and that we denote the identity element in $(\mathcal{M}, \cdot)$ by $1 \in \mathcal{M}$.

Suppose the adversary's target attribute is $a^* \in \mathbb{A}$. In the subgroup membership problem (SMP), he is given an element $c^* \in \mathcal{C}^{(a^*)}$ which is sampled in one of two ways: (1). the element c^* is uniformly sampled from $\mathcal{C}^{(a^*)}$; or (2). the element c^* is uniformly sampled from $\mathcal{C}_1^{(a^*)}$. The goal is to distinguish both of these distributions given oracle access to K_{MSK} conditioned on the fact that the adversary cannot query an $f \in \mathbb{F}$ with $f(a^*) = 1$. More precisely, we assume the hardness of a family of subgroup membership problems $\{\mathsf{SMP}_{a^*}\}_{a^* \in \mathbb{A}}$. It can be shown that solving a problem in this family is equivalent to attacking the semantic security of the scheme. For more details, we refer the reader to [1] wherein Armknecht et al. characterize the security of public-key GHE as a subgroup membership problem; the characterization holds analogously for ABGHE.

4 Possibility Result from Indistinguishability Obfuscation

It is interesting to consider whether we can give a possibility result for ABGHE by relying on indinstuishability obfuscation [34]. It was shown in [35] that attribute-based FHE can be realized from indistinguishability obfuscation. The authors use the technique of punctured programming [36], which involves using indistinguishability obfuscation together with a puncturable pesudorandom function (PRF) [37–39]. In essence, the public parameters contain an obfuscation of a program that maps an attribute to a public key of an FHE scheme. Then the FHE scheme is used for encryption and evaluation. If we replace the FHE scheme with a (public-key) GHE scheme, we obtain an identity-based GHE scheme (i.e. an instance of ABGHE). We state this formally in the following theorem:

Theorem 1. *Assuming indistinguishability obfuscation and one-way functions, if there exists an IND-CPA secure public-key GHE scheme for the group* $(S, \odot)$ *where* S *is a finite set, then there exists an IND-sID-CPA secure identity-based GHE for the group* $(S, \odot)$*.*

Proof. The theorem follows immediately from Theorem 1 in [35] by replacing the FHE scheme with a GHE scheme. □

Unfortunately we cannot obtain an ABGHE scheme in this manner for a more complex class of access policies than IBE. The reason for this is that the above construction is inherently "single-attribute" i.e. it only supports evaluation on ciphertexts with the same attribute (i.e. identity). Therefore, for a more complex class of access policies, the construction does not meet the criteria of ABGHE. This is because each attribute is mapped on to a unique public-key in the GHE scheme but we cannot perform evaluation on ciphertexts that are encrypted with different public keys (not while keeping the ciphertext the same size).

5 Additively Homomorphic ABGHE

5.1 XOR-Homomorphic IBE

Clear, Hughes and Tewari [30] present an XOR-homomorphic variant of the Cocks IBE scheme [31] which has a security reduction from the quadratic residuosity problem. This construction is shown in [30] to satisfy the properties of an ABGHE for the XOR operation. Joye [32] shows that the Cocks cryptosystem itself is inherently XOR-homomorphic but the operation on the ciphertext space is not associative and hence is an instance of Attribute-Based Quasigroup Homomoprhic Encryption (ABQHE). Ciphertexts in the scheme from [30] require 4 elements of $\mathbb{Z}_N$ where N is an RSA modulus whereas ciphertexts in Cocks' cryptosystem require only 2 elements of $\mathbb{Z}_N$. The ciphertext space complexity of CHT was improved recently in [40] to 2 elements of $\mathbb{Z}_N$ (like Cocks). The scheme however is not an ABGHE but an ABQHE.

5.2 Homomorphic IBE for Addition Modulo a Prime

Boneh, LaVigne and Sabin [29] presented an IBE scheme whose security relies on the MER assumption. However, their scheme uses a hash function that maps identity strings to e^{th} residues in $\mathbb{Z}_N$. It is not known how such a function can be instantiated without compromising security. We extend their construction so that it uses a hash function that can be instantiated. We then prove our construction secure under the MER assumption in the random oracle model. We show that the construction is group homomorphic for the additive group $(\mathbb{Z}_e, +)$ for prime e i.e. we show it meets the criteria for ABGHE. This is the only additively group-homomorphic IBE we are aware of with a message space larger than 2 elements. First, we need to introduce some functions that are used by the scheme along with an overview on how e^{th} power residue symbols are computed for integers in $\mathbb{Z}_N$.

e^{th} Power Residue Symbols in $\mathbb{Z}_N$ Let $e \geq 2$ be an integer. Let $N = pq$ be a product of two primes p and q with $p \equiv q \equiv 1 \mod e$. The symbol $\left(\frac{x}{N}\right)_e$ for integers x is always 1 for odd e and ± 1 for even e, so for $e > 2$, we need to find a way to extract more information about x so we can map it to one of e symbols. We follow the approach taken in [33].

Let ζ_e and K be as defined in Definition 3. Note that we can take K to be $\mathbb{Q}[x]/\Phi_e(x)$ where $\Phi_e(x)$ is the e^{th} cyclotomic polynomial; accordingly, we have $\zeta_e = x$. Given p and q, we can compute an element $\mu \in \mathbb{Z}_N^*$ that is a primitive root of unity modulo p and modulo q. In schemes described later, we require that μ be published as part of the public parameters. For a fixed μ, we define the ideal $\mathfrak{N} = N\mathcal{O}_K + (\zeta_e - \mu)\mathcal{O}_K$. Let $\mu_p = \mu \mod p$ and $\mu_q = \mu \mod q$. We also define the ideals $\mathfrak{p} = p\mathcal{O}_K + (\zeta_e - \mu_p)\mathcal{O}_K$ and $\mathfrak{q} = q\mathcal{O}_K + (\zeta_e - \mu_q)\mathcal{O}_K$. It holds that $\mathfrak{N} = \mathfrak{pq}$. Furthermore, we define a function $J_N : \mathbb{Z}_N \to \{0, \ldots, e-1\}$ as follows

$$J(x) = \begin{cases} 0 \text{ if } \mathsf{gcd}(x, N) \neq 1 \\ i \text{ if } \mathsf{gcd}(x, N) = 1 \text{ and } \left(\frac{x}{\mathfrak{N}}\right)_e = \zeta_e^i \end{cases}$$

Additionally, we define J_p analogous to J_N except with ideal $\mathfrak{p}$ and modulus p, and similarily, we define J_q using ideal $\mathfrak{q}$ and modulus q. When an integer x is an e^{th} power residue modulo N, we have $J_N(x) = 0$. We establish some important properties:

- $$J_N(x) \equiv J_p(x) + J_q(x) \mod e \quad \forall x \in \mathbb{Z}_N \tag{5.1}$$
- **Homomorphic property**

$$J_N(xy) \equiv J_N(x) + J_N(y) \mod e \quad \forall x, y \in \mathbb{Z}_N^* \tag{5.2}$$

The homomorphic property is also satisfied by J_p and J_q.

Boneh, LaVigne and Sabin (BLS) Scheme We now describe the BLS scheme. While the scheme is described as an IBE in [29], as aforementioned, there is no efficient means to realize the hash function it depends on. Therefore, it is effectively only useful as a public-key scheme, and in fact the security proof in [29] treats it as such. For this reason, we present it here as a public-key scheme.

The scheme is parameterized by a prime e. Note the scheme employs the function J_N which implicitly uses the root of unity μ published in the public key.

- $\mathsf{Gen}(1^\lambda)$: Generate two RSA primes p and q with $e|p-1$ and $e|q-1$ and let $N=pq$. Uniformly choose a nontrivial, nondegenerate root of unity $\mu \in \mathbb{Z}_N$. Uniformly sample an integer $r \xleftarrow{\$} \mathbb{Z}_N^*$ and set $v \leftarrow r^e \mod N$. Output $(\mathsf{pk} := (N, \mu, v), \mathsf{sk} := r)$.
- $\mathsf{Encrypt}(\mathsf{pk}, m)$: Given public key $\mathsf{pk} := (N, \mu, v)$ and message $m \in \{0, \ldots, e-1\}$, perform the following steps. Generate a uniformly random polynomial $f(x) \xleftarrow{\$} \mathbb{Z}_N^*[x]$ of degree $e-1$ and compute $g(x) \leftarrow f(x)^e \mod x^e - v$. Choose a uniformly random $t \xleftarrow{\$} \mathbb{Z}_N^*$ and compute the polynomial $c(x) \leftarrow \frac{g(x)}{t}$. Output $\mathsf{CT} := (c(x), d := m + J_N(t) \mod e)$.
- $\mathsf{Decrypt}(\mathsf{sk}, \mathsf{CT})$: Given secret key $\mathsf{sk} := r$ and ciphertext $\mathsf{CT} := (c(x), d)$, output $d + J_N(c(r)) \mod e$.

BLS is proven semantically secure under the MER assumption in the standard model.

Our Construction Our approach to circumventing the uninstantiability of the hash function employed in the IBE-version of BLS is akin to the original Cocks scheme. As part of the public parameters, we publish $e-1$ e^{th} non-residues (with $J_N(x) = 0$ for all non-residues x). Then for any integer a satisfying $J(a) = 0$, either a is an e^{th} residue or its product with one of the $e-1$ non-residues is an e^{th} residue. We also make some simplifications to BLS such as removing an element of $\mathbb{Z}_e$ from the ciphertext. We assume a hash function $H : \{0,1\}^* \to \{x \in \mathbb{Z}_N : J_N(x) = 0\}$ that maps identity strings to elements of $x \in \mathbb{Z}_N$ with $J_N(x) = 0$ (i.e. the power residue symbol of the element is 1).

The scheme is parameterized with a prime e. We make use of the functions J_N and J_p defined earlier which implicitly use a root of unity μ published in the public parameters.

Remark 1. We sometimes omit "mod N" for ease of presentation. This is particularly the case for products involving the elements α_i (as described below) to avoid clutter.

- $\mathsf{Setup}(1^\lambda)$: Generate two RSA primes p and q with $e|p-1$ and $e|q-1$ and let $N=pq$. Sample uniformly an element $\gamma \xleftarrow{\$} \mathbb{Z}_N^*$ with $J_N(\gamma) = 0$ and $J_p(\gamma) \neq 0$. For every $i \in [e]$, set $\alpha_i \leftarrow \gamma^{i-1} \mod N$. Uniformly choose a nontrivial, nondegenerate root of unity $\mu \in \mathbb{Z}_N$. Output $\mathsf{PP} := (N, \mu, \alpha_1, \ldots, \alpha_e)$ and $\mathsf{MSK} := (p, q, \alpha_1, \ldots \alpha_e)$.
- $\mathsf{KeyGen}(\mathsf{MSK}, \mathsf{id})$: Given master secret key $\mathsf{MSK} := (p, q, \alpha_1, \ldots, \alpha_e)$ and an identity string $\mathsf{id} \in \{0,1\}^*$, compute $a \leftarrow H(\mathsf{id})$. Check which of $\alpha_1 \cdot a, \ldots, \alpha_e \cdot a$ is an e^{th} residue and let the index in the list be i. Then compute the e^{th} root of $\alpha_i \cdot a$ using p and q; denote this root by r. Output $\mathsf{sk}_{\mathsf{id}} = (i, r)$.
- $\mathsf{Encrypt}(\mathsf{PP}, \mathsf{id}, m)$: Given public parameters $\mathsf{PP} := (N, \mu, \alpha_1, \ldots, \alpha_e)$, an identity string $\mathsf{id} \in \{0,1\}^*$ and a message $m \in \{0, \ldots, e-1\}$, first compute $a \leftarrow H(\mathsf{id})$. We define the subalgorithm $\mathcal{E}$ that takes an integer v and message m' as input and outputs a polynomial in $\mathbb{Z}_N[x]$.

  $\mathcal{E}(v, m')$:
  - Generate a uniformly random polynomial $f(x) \xleftarrow{\$} \mathbb{Z}_N^*[x]$ of degree $e-1$.
  - Compute $g(x) \leftarrow f(x)^e \mod x^e - v$.
  - Choose a uniformly random $t \xleftarrow{\$} \mathbb{Z}_N^*$ such that $J(t) = m'$
  - Output the polynomial $c(x) = t \cdot g(x)$.

  The encryption algorithm outputs $\mathsf{CT} = (a, \mathcal{E}(\alpha_1 \cdot a, m), \ldots, \mathcal{E}(\alpha_e \cdot a, m))$.
- $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}}, \mathsf{CT})$: On input a secret key $\mathsf{sk}_{\mathsf{id}} := (i, r)$ and a ciphertext $\mathsf{CT} := (a, c_1(x), \ldots, c_e(x))$, output $m \leftarrow J_N(c_i(r))$.

Correctness The correctness of decryption follows in the same way as BLS; since, $f(x)^3 = g(x)^3 + (x^3 - \alpha_i \cdot a)$, we have $f(r)^3 = g(r)^3$ when $r^3 \equiv \alpha_i \cdot a$ and $J_N(tg(r)^3) = J_N(t)$. It is necessary that the product of one of the α_i's with a gives an e^{th} residue. An element of $v \in \mathbb{Z}_N^*$ is an an e^{th} residue iff $J_N(v) = J_p(v) = 0$. Let $k = J_p(a)$. Then multiplying a with an element α satisfying $J_N(\alpha) = 0$ and $J_p(\alpha) = e - k$ guarantees that the

resulting element is an e^{th} residue (recall that $J_p(xy) = Jp(x) + Jp(y) \mod e$). So we need to show that for each $z \in \mathbb{Z}_e$, there is an α_i with $J_p(\alpha_i) = z$. In the setup, we sample a γ with $J_N(\gamma) = 0$ and $J_p(\gamma) \neq 0$. Let $g = J_p(\gamma)$. Then $J_p(\gamma^j) = jg \mod e$ for $j \in \{0, \ldots, e-1\}$ and since e is prime, this generates all elements in the additive group $\mathbb{Z}_e$.

Security Now we will reduce the security of our construction to that of BLS. When we refer to BLS hereafter, we will assume that its encryption algorithm is the same as $\mathcal{E}$ above i.e. it outputs a polynomial $\mathsf{CT} := c(x) = t \cdot g(x)$. This does not affect its security. However, there is an obstacle that we must contend with in the security reduction. Given a BLS public key, we cannot generate a $\gamma \in \mathbb{PR}(N) \setminus \mathbb{ER}(N)$ (note that this is precisely the set $\{x : J_N(x) = 0 \wedge J_p(x) \neq 0\}$) with probability 1 which is needed to correctly simulate the public parameters of our scheme. To address this, we consider a modified BLS scheme, denoted BLS$'$, that generates such a γ and outputs it as part of the public key. We first show that BLS$'$ is semantically secure under the MER assumption. Then we will base our security reduction on BLS$'$.

Lemma 2. *BLS$'$ is IND-CPA secure under the MER assumption.*

Proof. We will prove the lemma via a hybrid argument.
Game 0: This is the real IND-CPA game.

Game 1: We make one change from Game 0, namely we set $\gamma \leftarrow u^e \mod N$ for a uniformly chosen $u \xleftarrow{\$} \mathbb{Z}_N^*$.

Game 0 and Game 1 are computationally indistinguishable due to MER. In Game 0, γ is sampled uniformly from $\mathbb{PR}(N) \setminus \mathbb{ER}(N)$ and in Game 1, γ is sampled uniformly from $\mathbb{ER}(N)$.

Game 2: The change we make in this game is to encrypt a fixed element $w \in \mathbb{Z}_e$ instead of m_b, where m_0 and m_1 are the challenge messages and b is a random bit. The adversary has a zero advantage in this game.

Game 1 and Game 2 are computationally indistinguishable by the semantic security of BLS. Given a BLS public key (N, μ, v), we use these values in the public key and generate γ as in Game 2. When the adversary provides the challenge plaintexts (m_0, m_1), we choose a random b and forward the challenge plaintexts (m_b, w) to the BLS challenger, and return the challenge ciphertext CT^* provided by the BLS challenger. If CT^* encrypts m_b then Game 1 is perfectly simulated whereas if it encrypts w, Game 2 is perfectly simulated. Therefore, a non-negligible advantage distinguishing the hybrids implies a non-negligble advantage breaking the semantic security of BLS. □

Theorem 2. *Our scheme is IND-ID-CPA secure under the MER assumption in the random oracle model.*

Proof. Let $\mathcal{A}$ be the adversary in the IND-ID-CPA game against our scheme. We show that a non-negligble advantage by $\mathcal{A}$ implies a non-negligble advantage against the IND-CPA security of BLS$'$. We construct a simulator $\mathcal{S}$ that interacts in th IND-CPA game and simulates the view of $\mathcal{A}$. The hash function H in our IBE scheme is modeled as a random oracle. We now describe how $\mathcal{S}$ works.

Given a public key (N, μ, v, γ) of BLS$'$ by the IND-CPA challenger, $\mathcal{S}$ uses this information to construct public parameters $(N, \mu, \alpha_1, \ldots, \alpha_e)$, which it gives to $\mathcal{A}$. Let Q be the number of non-adaptive calls to the random oracle H. We assume that $\mathcal{A}$ makes a call to H for identity id prior to making a secret key query for id. The simulator picks a random $k \in [Q]$. The simulator answers calls to H as follows. On the j-th call to H with identity string id_j, perform the following steps:

- If $j = k$:
  - Choose a random $i \xleftarrow{\$} [e]$.
  - Add tuple $(\mathsf{id}_k, \perp, i)$ to table T.
  - Output $v \cdot \alpha_i^{-1} \mod N$
- Else:
  - Choose a random $i \xleftarrow{\$} [e]$.
  - Choose a random $r \xleftarrow{\$} \mathbb{Z}_N^*$.
  - Add tuple (id_j, r, i) to T.
  - Output $r^e \cdot \alpha_i^{-1}$.

The simulator handles secret key queries as follows. On querying the secret key for identity id, perform the following steps.

- If $\mathsf{id} = \mathsf{id}_k$, output a random bit and abort the simulation.

- Fetch tuple (id_j, r, i) from T with $\mathsf{id}_j = \mathsf{id}$.
- Output r.

When $\mathcal{A}$ sends its target identity id^* and pair of challenge plaintexts (m_0, m_1), the simulator checks if $\mathsf{id}^* = \mathsf{id}_k$. If this is not the case, $\mathcal{S}$ outputs a random bit and aborts. Otherwise, it forwards (m_0, m_1) to the IND-CPA challenger. Subsequently, the IND-CPA challenger gives $\mathcal{S}$ its challenge ciphertext $\mathsf{CT}^* := c^*(x)$. The simulator performs the following steps:

- Fetch $(\mathsf{id}_k, \perp, i)$ from T.
- Set $c_i(x) \leftarrow c^*(x)$.
- Set $a \leftarrow v \cdot \alpha_i^{-1} \mod N$
- Compute $c_j(x) \leftarrow \mathcal{E}(\alpha_j \cdot a, u_j)$ with $u_j \xleftarrow{\$} \mathbb{Z}_e$ for all $j \in [e] \setminus \{i\}$.
- Set $\mathsf{CT} \leftarrow (a, c_1(x), \ldots, c_e(x))$.

The simulator then gives CT to $\mathcal{A}$ as its challenge ciphertext. We claim that CT is identically distributed to a ciphertext in the real game. Firstly, since $a \cdot \alpha_i \equiv v \mod N$, we have that $c_i(x)$ is perfectly simulated. For all other $j \in [e]$ with $j \neq i$, the element $a \cdot \alpha_j$ is an e^{th} non-residue. It is shown in [29] that ciphertext polynomials computed with an e^{th} non-residue give no information about the plaintext. Therefore, in the view of $\mathcal{A}$, the challenge ciphertext CT is perfectly simulated. Finally, $\mathcal{S}$ outputs $\mathcal{A}$'s guess bit. The probability that the simulation does not abort is $1/Q$. It follows that if $\mathcal{A}$ has advantage ϵ attacking the IND-ID-CPA security of our scheme then $\mathcal{S}$ has advantage ϵ/Q attacking the IND-CPA security of BLS′. Since a non-negligble ϵ would contradict Lemma 2 assuming MER holds, the result follows. □

Homomorphism We now show that our construction is additively homomorphic for the group $(\mathbb{Z}_e, +)$. Given two ciphertexts $\mathsf{CT}_1 := (a, c_1(x), \ldots, c_e(x))$ and $\mathsf{CT}_2 := (a, d_1(x), \ldots, d_e(x))$ encrypted with the same identity id with $a = H(\mathsf{id})$, we compute the i-th component of the resulting ciphertext as $e_i(x) = c_i(x) \cdot d_i(x) \pmod{x^e - \alpha_i \cdot a}$ for $i \in [e]$. Consider the i-th component of the ciphertexts such that $\alpha_i \cdot a \in \mathbb{Z}_N$ is an e^{th} residue. Suppose we have that $c_i(x) = t_1 \cdot f_1(x)^e \pmod{x^e - \alpha_i \cdot a}$ and $d_i(x) = t_2 \cdot f_2(x)^e \pmod{x^e - \alpha_i \cdot a}$. Let r be the e^{th} root of $\alpha_i \cdot a$. To see that multiplication modulo $(x^e - \alpha_i \cdot a)$ is homomorphic, observe that

$$
\begin{aligned}
J_N(c_i(x)d_i(x) \pmod{x^e - \alpha_i \cdot a}(r)) &= J_N((t_1 \cdot f_1(x)^e) \cdot (t_2 \cdot f_2(x)^e) \pmod{x^e - \alpha_i \cdot a}(r)) && (5.3)\\
&= J_N((t_1 \cdot t_2)(f_1(x) \cdot f_2(x))^e \pmod{x^e - \alpha_i \cdot a}(r)) && (5.4)\\
&= J_N((t_1 \cdot t_2) \cdot (f_1(r) \cdot f_2(r))^e) && (5.5)\\
&= J_N(t_1 \cdot t_2) && (5.6)\\
&= J_N(t_1) + J_N(t_2) \pmod{e} && (5.7)
\end{aligned}
$$

Recall the homomorphic property of J_N i.e. $J_N(xy) = J_N(x) + J_N(y) \mod e$.

Keeping with the notation we have established so far, let us first fix some identity $\mathsf{id} \in \{0,1\}^*$ and let $a = H(\mathsf{id})$, then let f be the predicate that outputs 1 for identity id and 0 otherwise. Let (i, r) be a secret key for id. The ciphertext space $\hat{\mathcal{C}}_f$ is defined as follows:

$$
\begin{aligned}
\hat{\mathcal{C}}_f \triangleq \{(a, (c_1(x), \ldots, c_e(x)) \in \mathbb{Z}_N^e : \mathsf{deg}(c_1) = \cdots = \mathsf{deg}(c_e) = e - 1,\\
\left(\frac{c_i(r)}{\mathfrak{N}}\right)_e \neq 0,\\
c_j(x) \text{ is invertible in } \mathbb{Z}_N[x]/(x^e - \alpha_j \cdot a)\ \forall j \in [e]\}.
\end{aligned}
$$

The binary operation $*$ can be defined on $\hat{\mathcal{C}}$ as follows: given two ciphertexts $\mathsf{CT}_1 := (a_1, c_1(x), \ldots, c_e(x))$ and $\mathsf{CT}_2 := (a_2, d_1(x), \ldots, d_e(x))$, their product under $*$ is defined as $\mathsf{CT}' := (a_1, c_1(x) \cdot d_1(x) \pmod{x^e - \alpha_1 \cdot a_1}, \ldots, c_e(x) \cdot d_e(x) \pmod{x^e - \alpha_e \cdot a_1}$ if $a_1 = a_2$, and $\mathsf{CT}' := Z$ otherwise, where $Z \in \hat{\mathcal{C}}$ is the null ciphertext.

Lemma 3. *$(\hat{\mathcal{C}}_f, *)$ is a group.*

Proof. It is sufficient to consider a single component of the ciphertext because the same analysis applies for each component. Let $v = \alpha_i \cdot a$ for some j. We can view the j-th component as an element in the ring $R_a = \mathbb{Z}_N[x]/(x^e - v)$. Let $c(x)$ be the j-th polynomial component of a ciphertext in $\hat{\mathcal{C}}_f$. By definition, $c(x)$ is invertible. Consider the case where $j = i$. By definition, we have $\left(\frac{c(r)}{\mathfrak{N}}\right)_e \neq 0$. Applying $*$ to $c(x)$ and any other element of $\hat{\mathcal{C}}_f$ preserves this condition. Therefore $\hat{\mathcal{C}}_f$ is closed under $*$. It follows $(\hat{\mathcal{C}}_f, *)$ is a group. □

We denote the set of legal encryptions under identity id by $\mathcal{C}_f$ where f is defined as before. We have the following straightforward lemma:

Lemma 4. *$(\mathcal{C}_f, *)$ is a subgroup of $\hat{\mathcal{C}}_f$.*

Proof. We focus on a single component, say the j-th, of a ciphertext. Let $c(x)$ be such a component. Then $c(x)$ is of the form $t \cdot f(x)^e$ for some $f(x)$ that is a unit[§] in $\mathbb{Z}_N[x]/(x^e - \alpha_j \cdot a)$ and $t \in \mathbb{Z}_N^*$. Naturally we have that $c(x) \in \hat{\mathcal{C}}_f$. Multiplying $c(x)$ by another element $d(x)$ with the same form yields an element of the same form. □

Theorem 3. *Our scheme is an ABGHE scheme i.e. it satisfies Definition 6.*

Proof. By Lemma 4 the scheme satisfies GH.1. By the derivation given in equations 5.3 - 5.7 the scheme satisfies GH.2. Therefore the scheme is an ABGHE. □

Homomorphic Addition Modulo a "Large" Modulus Our scheme supports homomorphic addition modulo a "small" (i.e. poly-sized) prime. However if we use multiple instances of the scheme with distinct primes, we can leverage the Chinese Remainder Theorem to support addition modulo a square-free integer M provided M factors into a polynomial number of poly-sized primes. Hence we can support modular addition with an exponentially-large modulus. This is the first IBE scheme admitting a modular additive homomorphism with a superpolynomial modulus, solving an open problem mentioned in [30].

Concretely, suppose our desired square-free modulus is $M = p_1 \cdots p_n$. We employ n instances of our scheme $\{\mathcal{E}_i\}_{i\in[n]}$ with the e parameter for $\mathcal{E}_i$ set to p_i for all $i \in [n]$.

- $\mathsf{Setup}(1^\lambda)$: Output $(\mathsf{PP} := (\mathsf{PP}_1, \ldots, \mathsf{PP}_n), \mathsf{MSK} := (\mathsf{MSK}_1, \ldots, \mathsf{MSK}_n)$ where $(\mathsf{PP}_i, \mathsf{MSK}_i) \leftarrow \mathcal{E}_\mathsf{i}.\mathsf{Setup}(1^\lambda)$ for $i \in [n]$.
- $\mathsf{KeyGen}(\mathsf{MSK} := (\mathsf{MSK}_1, \ldots, \mathsf{MSK}_n), \mathsf{id})$: Output $\mathsf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ where $\mathsf{sk}_i \leftarrow \mathcal{E}_\mathsf{i}.\mathsf{KeyGen}(\mathsf{MSK}_i, \mathsf{id})$ for $i \in [n]$.
- $\mathsf{Encrypt}(\mathsf{PP} := (\mathsf{PP}_1, \ldots, \mathsf{PP}_n), \mathsf{id}, m)$: Output $c := (c_1, \ldots, c_n)$ where $c_i \leftarrow \mathcal{E}_\mathsf{i}.\mathsf{Encrypt}(\mathsf{PP}_i, m \mod p_i)$ for $i \in [n]$.
- $\mathsf{Decrypt}(\mathsf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_n), c := (c_1, \ldots, c_n))$: Output $\mathsf{CRT}((m_1, \ldots, m_n), (p_1, \ldots, p_n))$ where $m_i \leftarrow \mathcal{E}_\mathsf{i}.\mathsf{Decrypt}(\mathsf{sk}_i, c_i)$ for $i \in [n]$.
- **Additive Homomorphism**: Let $*^i$ denote the binary operation on the ciphertext space of $\mathcal{E}_i$. We define $*$, the binary operation on the ciphertext space of this construction, as follows:
- $c * c' = (c_1, \ldots, c_n) * (c'_1, \ldots, c'_n) \triangleq (c_1 *^1 c'_1, \ldots, c_n *^n c'_n)$

The ciphertext space complexity of this scheme is $\sum p_i^2$.

Anonymity The XOR-homomorphic scheme CHT mentioned earlier is not anonymous as a result of a test due to Galbraith[¶]. Consider an identity id and let $a = H(\mathsf{id})$. Ciphertexts in CHT are a pair of polynomials $(c(x), d(x)) \in (\mathbb{Z}_N[x])^2$. We will consider only a single ciphertext component here, say the first $(c(x))$, which is encrypted with respect to a. The observations also hold with respect to the second component by replacing a with $-a$. We define Galbraith's Test for ciphertext polynomials as the function $\mathsf{GT} : \mathbb{Z}_N \times \mathbb{Z}_N[x] \to \{-1, 0, +1\}$ given by

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

For encryptions $c(x)$ (recall we are just considering one component) encrypted under identity id, we have $\mathsf{GT}(a, c(x)) = 1$. For encryptions $c'(x)$ under a different identity, it is the case that $\mathsf{GT}(a, c'(x)) = 1$ with probability negligbly close to $1/2$.

For convenience, let us denote our scheme that extends BLS, as described above, for the case of $e = 2$ (i.e. admitting an XOR homomorphism) by $\mathcal{E}_2$. Although $\mathcal{E}_2$ is algorithmically different to CHT, it shares many of the same properties. In particular it is easy to see that Galbraith's test is applicable in the same way. An

[§]We omitted an explicit check for this in the encryption algorithm since a non-unit occurs with negligible probability

[¶]Reported as emerging from personal communication in [41]

anonymous variant of CHT was proposed in [42] and the techniques are also applicable to $\mathcal{E}_2$. However the approach to achieve anonymity in [42] loses the homomorphic property i.e. one cannot homomorphically operate on anonymized ciphertexts. It remains an open problem to construct an anonymous additively homomorphic IBE.

We now turn our attention to investigating whether our scheme for the case of $e > 2$ is anonymous. Let us start with the simplest case: $e = 3$. We will denote our scheme for this case by $\mathcal{E}_3$. As usual, for identity id, we let $a = H(\mathsf{id})$. Consider a ciphertext polynomial $c(x)$. Let $\hat{C}_a$ be the set of polynomials for a single component of the ciphertext space $\hat{C}_f$ (where f is the predicate as defined earlier for identity id). Let $C_a \subset \hat{C}_a$ be the set of polynomials for a single component in the image of the encryption algorithm with respect to a. We are interested in determining whether there is an efficient algorithm to distinguish between an element of C_a and $\hat{C}_a \setminus C_a$. Membership of C_a implies the existence of a solution $(z_0, z_1, z_2) \in \mathbb{Z}_N^3$ to a system of three trivariate polynomial equations of degree 3. Deciding solvability of a system of multivariate polynomial equations in general is NP-complete. However for the special system of equations of interest here, with certain structure, we must make an explicit assumption about the hardness of deciding its solvability.

Let $A = \{x \in \mathbb{Z}_N^* : J_N(x) = 0\}$. We now define an assumption under which we prove anonymity of $\mathcal{E}_3$.

Definition 7 (Special Trivariate Equations Solvability (STES) Assumption). *Given $(a, c_0, c_1, c_2) \in A \times \hat{C}_a$ where $a \xleftarrow{\$} A$, consider an algorithm $\mathcal{A}$ to decide the solvability of the following system of trivariate polynomial equations in variables z_0, z_1 and z_2*

$$\begin{aligned} z_0^3 + az_1^3 + a^2 z_2^3 + 6az_0z_1z_2 &= c_0 \\ 3z_0^2 z_1 + 3z_0 z_2^2 + 3az_1^2 z_2 &= c_1 \\ 3z_0^2 z_2 + 3z_0 z_1^2 + 3az_1 z_2 &= c_2 \end{aligned}$$

Let S be the set of instances in $A \times \hat{C}_a$ that are solvable and let $\bar{S}$ be the unsolvable instances. The advantage of $\mathcal{A}$ deciding correctly $\mathsf{Adv}_{\mathcal{A}}$ is defined as

$$\mathsf{Adv}_{\mathcal{A}} \triangleq \mathsf{Pr}[s \xleftarrow{\$} S : \mathcal{A}(s) \to 1] - \mathsf{Pr}[\bar{s} \xleftarrow{\$} \bar{S} : \mathcal{A}(\bar{s}) \to 1].$$

The STES assumption is that for every PPT algorithm $\mathcal{A}$ it holds that $\mathsf{Adv}_{\mathcal{A}} < \mathsf{negl}(\lambda)$.

Lemma 5. *The sets C_a and $\hat{C}_a \setminus C_a$ are computionally indistinguishable for $a \xleftarrow{\$} A$ assuming the hardness of STES.*

Proof. We have that a is uniformly chosen from A. The elements of C_a are the honestly generated ciphertext polynomials. In the encryption algorithm a polynomial $f(x)$ is cubed. By simple algebra, cubing this polynomial and reducing according to the equivalence relation $x^3 \equiv a$ induced by the quotient of the ring $\mathbb{Z}_N/(x^3 - a)$ yields the trivariate equations given in Definition 7 in terms of the coefficients of $f(x)$. Solvability of this system of equations is equivalent to membership of C_a. An algorithm that efficiently distinguishes between C_a and $\hat{C}_a \setminus C_a$ can therefore be used to solve STES. □

Theorem 4. *$\mathcal{E}_3$ is anonymous under the STES assumption.*

Proof. In the anonymity security game, the adversary chooses two target identities id and id'.
Game 0: This is the real game.
Game 1: In this game, we change how the challenge ciphertext is generated if the challenger's bit $\beta = 0$ (i.e. using identity id). If $\beta = 0$, we sample the challenge ciphertext uniformly from $\hat{C}_a$ instead of C_a where a is what is returned by $H(\mathsf{id})$.

To invoke Lemma 5 to argue indistinguishability of $\hat{C}_a$ and C_a, we need to program the output of the random oracle H on identity id to be a, given as part of the STES instance, which is distributed correctly. In a similar manner to the proof of Theorem 2, we must guess one of the identities the adversary chooses from its queries to H and abort with a random bit if we guessed incorrectly. This step loses a factor of roughly $1/Q$ where Q is the number of queries to H prior choosing the target identities.
Game 2: In this game, we change how the challenge ciphertext is generated if the challengers bit $\beta = 1$ (i.e. using identity id'). If $\beta = 1$, we sample the challenge ciphertext uniformly from $\hat{C}_b$ instead of C_b where b is what is returned by $H(\mathsf{id}')$.

Indistinguishability follows in the same manner as the transition between Game 0 to Game 1.

Game 3: In this game, we sample the challenge ciphertext always from $\hat{C}_a$. The adversary has zero advantage in this game as it learns no infromation about β.

We must show that $\hat{C}_a$ and $\hat{C}_b$ are computationally indistinguishable. An element of $\hat{C}_a$ is also an element of $\hat{C}_b$ with all but negligible probability since the event that it is noninvertible in $\mathbb{Z}_N[x]/(x^3 - b)$ and the event that the power residue symbol of its evaluation at a certain point is zero both occur with negligible probability. The result follows. □

For each instance of the scheme with $e > 3$, we obtain a system of multivarate polynomial equations of degree e in e variables and we must make an explicit assumption about deciding its solvability.

5.3 Multiplicatively Homomorphic ABGHE Schemes

It is a well-known that a scheme with a multiplicative homomorphism can be transformed into one with an additive homomorphism, where the addition takes place in the exponent, and a discrete logarithm problem must be solved to recover the result. This gives rise to the following theorem, which holds true in the public-key setting as well (a fortiori because public-key HE is a special case of ABHE):

Theorem 5. *Let $\mathcal{E} = (G, K, E, D)$ be a multiplicatively homomorphic ABGHE where $(\mathcal{M}, \cdot)$ is cyclic. For any positive integer $M = \mathsf{poly}(\lambda)$ with $M \mid |\mathcal{M}|$, there is an additively homomorphic ABGHE scheme with plaintext group $(\mathbb{Z}_M, +)$.*

Proof. We define a new scheme $\mathcal{E}'$ whose setup and key generation algorithms are the same as $\mathcal{E}$. Let $g \in \mathcal{M}$ be a generator for $(\mathcal{M}, \cdot)$. The element $h := g^{|\mathcal{M}|/M}$ is a generator for a subgroup of $\mathcal{M}$ of order M. One can define the encryption algorithm E' as follows: on input a message $\mu \in \{0, \ldots, M-1\}$ and attribute a, compute $c \leftarrow E_{\mathsf{PP}}(a, h^\mu)$ and output c. The image of $E'_{\mathsf{PP}}(a, \cdot)$ with domain $\mathbb{Z}_M$ is a subgroup of $E_{\mathsf{PP}}(a, \cdot)$ with domain $\mathcal{M}$ with respect to operation $*$. This satisfies GH.1. The decryption algorithm is defined as $D'_{\mathsf{sk}_f}(c) = \log_h(D_{\mathsf{sk}_f}(c))$. Let c be an encryption of $x \in \mathbb{Z}_M$ and c' be an encryption of $y \in \mathbb{Z}_M$. These elements can respectively be viewed as encryptions in the scheme $\mathcal{E}$ of $h^x \in \mathcal{M}$ and $h^y \in \mathcal{M}$ respectively. Because D satisfies GH.2, we have

$$D'_{\mathsf{sk}_f}(c * c') = \log_h D_{\mathsf{sk}_f}(c * c') = \log_h \left(D'_{\mathsf{sk}_f}(c) \cdot D'_{\mathsf{sk}_f}(c')\right) = \log_h \left(h^x \cdot h^y\right) = \log_h \left(h^{x+y}\right) = x + y.$$

Therefore, the scheme also satisfies GH.2. □

A related fact, and one that shows up more frequently, is when M does not divide the group order $|\mathcal{M}|$ and is instead some polynomially sized bound. In this case, we get a bounded (aka "quasi") additively homomorphic scheme, but it is not group homomorphic in the sense of Definition 6 since one cannot perform an unbounded number of homomorphic operations.

Günther et al. [28] modified the Boneh-Franklin IBE [43] so that it is additively homomorphic in a bounded sense (i.e. it is additively homomorphic for $\mathbb{Z}_M$ for some M that does not divide the order of the group $(\mathcal{M}, \cdot)$). In fact, we could interpret the construction of Günther et al. as first transforming Boneh-Franklin into an ABGHE with a multiplicative homomorphism and then applying the above transformation to yield a *bounded* additive homomorphism. The same transformation can be applied to other pairings-based IBE schemes including [44, 45].

We now take a look at existing ABGHE schemes that are multiplicatively homomorphic. We recommend that the reader keep in mind that a *bounded* additive homomorphism can be obtained from these schemes via the above transformation.

As we have seen, many pairings-based ABE schemes are multiplicatively homomorphic. To illustrate the properties of a concrete ABGHE, we now examine such a construction due to Katz, Sahai and Waters (KSW) [46] (Appendix C); we call this scheme KSW. The security of KSW relies on non-standard assumptions on bilinear groups, assumptions that are justified by the authors in the generic group model.

Let m be a product of three "large" primes and let n be a positive integer that is polynomial in the security parameter. In KSW, an attribute is an n-dimensional vector over $\mathbb{Z}_m$ and a predicate (i.e. access policy) also corresponds to an n-dimensional vector over $\mathbb{Z}_m$. For $\boldsymbol{v} \in \mathbb{Z}_m^n$, a predicate $f_{\boldsymbol{v}} : \mathbb{Z}_m^n \to \{0, 1\}$ is defined by

$$f_{\boldsymbol{v}}(\boldsymbol{w}) = \begin{cases} 1 & \text{iff } \langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0 \\ 0 & \text{otherwise} \end{cases}$$

These predicates are called inner-product predicates.

Roughly speaking, in a ciphertext, all components of its attribute vector $\boldsymbol{w} \in \mathbb{Z}_m^n$ (which represent the sub-attributes) are blinded by the same uniformly random "blinding" element $b \in \mathbb{Z}_m$. The decryption algorithm multiplies each component by the corresponding component in the predicate vector, and the blinding element b is eliminated when the inner product evaluates to zero with all but negligible probability, which allows decryption to proceed.

Let $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ be ciphertexts with attribute vectors $\boldsymbol{a_1} \in \mathbb{Z}_m^n$ and $\boldsymbol{a_2} \in \mathbb{Z}_m^n$ respectively. It can be easily shown that the pairwise product $\boldsymbol{c'} = \boldsymbol{c_1} * \boldsymbol{c_2}$ of $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ produces a ciphertext that is associated with both $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ in a somewhat "isolated" way. The effect this has is conjunctive. So a predicate vector $\boldsymbol{v}$ has to satisfy $\langle \boldsymbol{v}, \boldsymbol{a_1} \rangle = 0$ and $\langle \boldsymbol{v}, \boldsymbol{a_2} \rangle = 0$ for decryption of $\boldsymbol{c'}$ to succeed (except with negligible probability). Furthermore, the effect on the underlying plaintexts is multiplicative (in a group of order m). Therefore, KSW is an ABGHE scheme with a multiplicative homomorphism. Another property that KSW satisfies is attribute privacy - the attribute vector is hidden by the ciphertext.

KSW also helps us illustrate the aforementioned properties of ABGHE. Consider Corollary 1, which tells us that if a "tautology" predicate $\top$ (i.e. a predicate that holds true for every attribute) is in the class of supported policies, then there is an attribute $\mathfrak{a} \in \mathbb{A}$ that satisfies all policies. In the case of KSW, such a predicate $\top$ is given by the zero vector. Accordingly, the attribute $\mathfrak{a}$ is also given by the zero vector.

On a technical note the ciphertexts in KSW are elements of the product group $\hat{\mathcal{C}} := \mathbb{G}_T \times \mathbb{G}^{2n+1}$ where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order m. The operation $*$ on $\hat{\mathcal{C}}$ corresponds to the operation of this product group. The plaintext group is $(\mathcal{M} := \mathbb{G}_T, \cdot)$. The identity element of the ciphertext space $\hat{\mathcal{C}}$ is $1_{\hat{\mathcal{C}}} := (1_{\mathbb{G}_T}, 1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}}) \in \hat{\mathcal{C}}$ where $1_{\mathbb{G}_T}$ is the identity element of $\mathbb{G}_T$ and $1_{\mathbb{G}}$ is the identity element of $\mathbb{G}$. Note that the identity element $1_{\hat{\mathcal{C}}}$ of $\hat{\mathcal{C}}$ is an encryption of $1 \in \mathcal{M}$ under $\mathfrak{a}$, which is the zero attribute vector in KSW.

6 Multi-Key Setting

We now define a variant of ABGHE that permits multiple keys to be passed to the decryption algorithm. Furthermore in this primitive the size of an evaluated ciphertext is permitted to grow with the number of distinct attributes used in the evaluation. Suppose we generate ciphertexts $c_1, \ldots, c_\ell$ where $c_i \leftarrow E(a_i, m_i)$ with $a_i \in \mathbb{A}$ and $m_i \in \mathcal{M}$ for $i \in [\ell]$. Suppose we compute the product $c' = c_1 * \cdots * c_\ell$. A person who does not have a secret key for a predicate f that holds for all attributes $a_1, \ldots, a_\ell$ cannot decrypt the ciphertext. Consider a person who has keys for multiple predicates $f_1, \ldots, f_k$ such that some attributes satisfy f_1, others f_2 and so on. It would be desirable for such a person to be able to decrypt the ciphertext by using all their keys, provided every attribute in the set $\{a_1, \ldots, a_\ell\}$ is satisfied by at least one of $f_1, \ldots, f_k$. The primitive we define captures this functionality. Our definition permits the size of a ciphertext to grow with the number of distinct attributes. Let $\mathscr{d} = |\{a_1, \ldots, a_\ell\}|$ be the number of distinct attributes in the above example. Then we require that the size of c' be polynomial in the security parameter λ and the number of distinct attributes $\mathscr{d}$.

Since the size of an evaluated ciphertext can grow with the number of distinct attributes, the ciphertext space does not form a group. Instead we generalize the $*$ operation on the ciphertext space to be a PPT algorithm H that takes two ciphertexts as input and outputs another ciphertext. This algorithm may be randomized. However our definition captures the fundamental property of ABGHE in that the group operation can be homomorphically applied an unbounded number of times. We refer to this primitive as *Multi-Key Attribute-Based Homomorphic Encryption for a group* which we abbreviate to multi-key ABHEg.

Definition 8 (Multi-Key Attribute-Based Homomorphic Encryption for a group (ABHEg)). *A multi-key ABHEg scheme is a tuple of PPT algorithms $\mathcal{E} = (G, K, E, D, H)$ with message space $\mathcal{M}$, attribute space $\mathbb{A}$, ciphertext space $\widehat{\mathcal{C}}$ and class of predicates $\mathbb{F}$ such that for every $(\mathsf{PP}, \mathsf{MSK}) \leftarrow G(1^\lambda)$, every $k > 0$, every $f_1, \ldots, f_k \in \mathbb{F}$:supp$(f_1) \cup \ldots \cup \mathsf{supp}(f_k) \neq \emptyset$, and every $\mathsf{sk}_{f_1} \leftarrow K(\mathsf{MSK}, f_1), \ldots, \mathsf{sk}_{f_k} \leftarrow K(\mathsf{MSK}, f_k)$, the message space $(\mathcal{M}, \cdot)$ is a non-trivial group, and the following properties are satisfied for the restricted ciphertext space $\widehat{\mathcal{C}_{f_1,\ldots,f_k}} = \{c \in \widehat{\mathcal{C}} : D_{\mathsf{sk}_{f_1},\ldots,\mathsf{sk}_{f_k}}(c) \neq \perp\}$:*

MK.1: $\forall c_1, c_2 \in \widehat{\mathcal{C}_{f_1,\ldots,f_l}} \quad D_{\mathsf{sk}_{f_1},\ldots,\mathsf{sk}_{f_k}}(H(c_1, c_2)) = D_{\mathsf{sk}_{f_1},\ldots,\mathsf{sk}_{f_k}}(c_1) \cdot D_{\mathsf{sk}_{f_1},\ldots,\mathsf{sk}_{f_k}}(c_2)$.

MK.2: *There is an attribute function* attr *that returns the attribute(s) associated with a ciphertext such that*

- *for any $c \leftarrow E(\mathsf{PP}, a, m)$ with $a \in \mathbb{A}$ and $m \in \mathcal{M}$, it holds that $\mathsf{attr}(c) = \{a\}$*
- *for any $c_1, c_2 \in \widehat{\mathcal{C}_{f_1,\ldots,f_k}}$, it holds that $\mathsf{attr}(H(c_1, c_2)) = \mathsf{attr}(c_1) \cup \mathsf{attr}(c_2)$.*

MK.3: *For any* $c \in \widehat{\mathcal{C}_{f_1,\ldots,f_k}}$*, it holds that* $|c| = \mathsf{poly}(\lambda, |\mathsf{attr}(c)|)$.

Note that this definition captures the essential property of ABGHE that the group operation can be homomorphically applied an unbounded number of times but it does not require the ciphertext space to form a group (or indeed any algebraic structure). In ABHEg, the evaluation algorithm H may be randomized.

Security In addition to semantic security we also require that a ciphertext emerging from evaluation does not reveal anything to a decryptor about the input plaintexts used to compute the result. More precisely, suppose we obtain an encryption of $m' \in \mathcal{M}$ via the evaluation algorithm such that m' is obtained as a product $m' = m_1 \cdots m_\ell \in \mathcal{M}$, then the decryptor should not learn anything about the m_i. We capture this formally in the following security definition.

Let $\mathcal{E} = (G, K, E, D, H)$ be a multi-key ABHEg scheme. The adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is a pair of PPT algorithms. The algorithm $\mathcal{A}_1$ outputs ℓ attribute-message pairs. We now consider two experiments termed the *real world* and *ideal world* respectively. In the real world, each of the plaintexts is encrypted with E and the evaluation algorithm H is applied to the ciphertexts to yield an evaluated ciphertext c'. In the ideal world, a simulator algorithm $\mathcal{S}$ generates c' given only the distinct attributes in the set of attributes outputted by $\mathcal{A}_1$ in addition to the plaintext result $m' = m_1 \cdots m_\ell \in \mathcal{M}$.

Definition 9 (Input Hiding). *We define the following two experiments:*

- $\mathbf{Exp}^{\mathsf{REAL}}_{\mathcal{E},\mathcal{A}}(\lambda)$ ***(Real World):***
  1. $(\mathsf{PP}, \mathsf{MSK}) \leftarrow G(1^\lambda)$.
  2. $((a_1, m_1), \ldots, (a_\ell, m_\ell)) \leftarrow \mathcal{A}_1^{K(\mathsf{MSK},\cdot)}(\mathsf{PP})$.
  3. $c_i \leftarrow E_{\mathsf{PP}}(a_i, m_i)$ *for* $i \in [\ell]$.
  4. $c' \leftarrow H(c_1, H(c_2, H(\cdots, c_\ell)))$.
  5. $b \leftarrow \mathcal{A}_2^{K(\mathsf{MSK},\cdot)}(c')$
  6. *Output* b.
- $\mathbf{Exp}^{\mathsf{IDEAL}}_{\mathcal{E},\mathcal{A},\mathcal{S}}(\lambda)$ ***(Ideal World):***
  1. $(\mathsf{PP}, \mathsf{MSK}) \leftarrow G(1^\lambda)$.
  2. $((a_1, m_1), \ldots, (a_\ell, m_\ell)) \leftarrow \mathcal{A}_1^{K(\mathsf{MSK},\cdot)}(\mathsf{PP})$.
  3. *Let* $\mathfrak{a}_1, \ldots, \mathfrak{a}_d$ *be the distinct attributes in* $\{a_1, \ldots, a_\ell\}$.
  4. $m' \leftarrow m_1 \cdots m_\ell$.
  5. $c' \leftarrow \mathcal{S}(\mathsf{PP}, \mathfrak{a}_1, \ldots, \mathfrak{a}_d, m')$.
  6. $b \leftarrow \mathcal{A}_2^{K(\mathsf{MSK},\cdot)}(c')$
  7. *Output* b.

Then $\mathcal{E}$ *is said to be* ***input hiding*** *if there exists a PPT simulator* $\mathcal{S}$ *such that for every pair of PPT algorithms* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$*, it holds that*

$$|\mathsf{Pr}[\mathbf{Exp}^{\mathsf{REAL}}_{\mathcal{E},\mathcal{A}} \to 1] - \mathsf{Pr}[\mathbf{Exp}^{\mathsf{IDEAL}}_{\mathcal{E},\mathcal{A},\mathcal{S}} \to 1]| < \mathsf{negl}(\lambda).$$

6.1 Generic Construction of Multi-Key ABHEg From ABGHE

We now show how to construct multi-key ABHEg from any ABGHE scheme.

Let $\mathcal{E} = (G, K, E, D)$ be an ABGHE scheme with message group $(\mathcal{M}, \cdot)$, attribute space $\mathbb{A}$, class of predicates $\mathbb{F}$ and ciphertext space $\hat{\mathcal{C}}$ equipped with binary operation $*$. We assume without loss of generality that there is a strict total order $\prec$ defined on $\mathbb{A}$. We also assume that $\mathcal{E}$ is not attribute-hiding; more precisely, the attribute associated with a ciphertext is readily obtained from the ciphertext (if this is not naturally the case, it can be done by appending the attribute to the ciphertext). Therefore, we define the function attr that gives the attribute associated with a ciphertext. We use the ABGHE scheme $\mathcal{E}$ to construct a multi-key ABHEg scheme $\mathcal{E}' = (G', K', E', D', H')$.

A ciphertext in $\mathcal{E}'$ is a sequence of ciphertexts $c_1, \ldots, c_t \in \hat{\mathcal{C}}^*$ such that $\mathsf{attr}(c_1) \prec \cdots \prec \mathsf{attr}(c_t)$. In other words, a ciphertext in $\mathcal{E}'$ is a sequence of ciphertexts in $\mathcal{E}$ ordered by attribute. As a warm up, consider the following simple construction, which does not meet our secuirty requirement of *input hiding* but serves to illustrate our approach. First we need to formally define the ciphertext space $\hat{\mathcal{C}}'$ of $\mathcal{E}'$:

$$\hat{\mathcal{C}}' \triangleq \{(c_1, \ldots, c_t) \in \hat{\mathcal{C}}^* \mid c_1, \ldots, c_t \in \hat{\mathcal{C}}, \mathsf{attr}(c_1) \prec \cdots \prec \mathsf{attr}(c_t), |t| \leq |\mathbb{A}|\}.$$

The setup algorithm G', key generation algorithm K' and encryption algorithm E' are equivalent to G, K and E respectively. It remains to define the decryption algorithm D'. Recall that since we are constructing a multi-key ABHEg scheme, the decryption algorithm D' takes a sequence of secret keys as input. Given secret keys for policies $f_1, \ldots, f_k$ and a ciphertext $\mathsf{CT} := c_1, \ldots, c_t$, the algorithm D' runs as follows. Firstly, each component c_i is decrypted using D with the secret key for any policy $f \in \{f_1, \ldots, f_k\}$ with $f(\mathsf{attr}(c_i)) = 1$ (if no such f is found, then $\perp$ is returned). Let $m_i \leftarrow D(\mathsf{sk}_f, c_i)$. Then output $m_1 \cdots m_t \in \mathcal{M}$.

Next we define the evaluation algorithm H'. Given two ciphertexts $\mathsf{CT}_1 := (c_1^{(1)}, \ldots, c_{t_1}^{(1)}) \in \hat{\mathcal{C}}'$ and $\mathsf{CT}_2 := (c_1^{(2)}, \ldots, c_{t_2}^{(2)}) \in \hat{\mathcal{C}}'$, one approach to realize H' is to apply $*$ to the ciphertexts with matching attributes, append the remaining ciphertexts and sort the resulting sequence. Our construction $\mathcal{E}'$ thus far satisfies the conditions for multi-key ABHEg. However it is clearly not *input hiding*. This is because we only want a decryptor to learn a single value $m_1 \cdots m_t \in \mathcal{M}$ and not the *components* $m_1, \ldots, m_t$ of this value. Our goal is to prevent the decryptor from seeing each m_i. The main idea to achiave this is to generate uniformly random values $r_1, \ldots, r_t \in \mathcal{M}$ subject to the condition that $r_1 \cdots r_t = 1 \in \mathcal{M}$, and blind each component's value m_i with r_i by multiplying c_i by an encryption of r_i. We now formally describe the H' algorithm:

- $\mathrm{H}'(\mathsf{CT}_1, \mathsf{CT}_2)$:
  - Parse CT_1 as $c_1^{(1)}, \ldots, c_{t_1}^{(1)}$.
  - Parse CT_2 as $c_1^{(2)}, \ldots, c_{t_2}^{(2)}$.
  - Apply merge sort to $c_1^{(1)}, \ldots, c_{t_1}^{(1)}$ and $c_1^{(2)}, \ldots, c_{t_2}^{(2)}$ where two ciphertexts c and d are compared as $\mathsf{attr}(c) \prec \mathsf{attr}(d)$. This gives a list of sorted ciphertexts $c_1, \ldots, c_{t_1+t_2}$.
  - For every pair of adjacent elements c_i, c_{i+1} with matching attributes, compute $d_i \leftarrow c_i * c_{i+1}$ and replace the occurrence of c_i, c_{i+1} in the list with d_i. This gives a sorted list $d_1, \ldots, d_t$ with $t = t_1 + t_2 - t'$ where t' is the number of matching attributes.
  - Set $a_i \leftarrow \mathsf{attr}(d_i)$ for $i \in [t]$.
  - Uniformly sample $r_i \xleftarrow{\$} \mathcal{M}$ for $i \in [t-1]$.
  - Set $r_t \leftarrow (r_1 \cdots r_{t-1})^{-1}$.
  - Output $d_1 * E_{\mathsf{PP}}(a_1, r_1)$, $\ldots$,$\mathrm{d}_t * E_{\mathsf{PP}}(a_t, r_t)$.

Theorem 6. *If $\mathcal{E}$ is an ABGHE scheme, then $\mathcal{E}'$ is input hiding.*

Proof. The simulator $\mathcal{S}$ can produce a ciphertext c' that is identically distributed to the corresponding ciphertext produced in the real world experiment. On input public parameters PP, attributes $\mathfrak{a}_1, \ldots, \mathfrak{a}_{\mathfrak{d}}$ and plaintext result m', the simulator generates random elements $r_1, \ldots, r_{\mathfrak{d}-1} \xleftarrow{\$} \mathcal{M}$ and sets $r_{\mathfrak{d}} \leftarrow m' \cdot (r_1 \cdots r_{\mathfrak{d}-1})^{-1}$. It sorts the attributes $\mathfrak{a}_1, \ldots, \mathfrak{a}_{\mathfrak{d}}$ according to $\prec$ to yield $a_1, \ldots, a_{\mathfrak{d}}$. It computes $c' \leftarrow E_{\mathsf{PP}}(a_1, r_1), \ldots, E_{\mathsf{PP}}(a_{\mathfrak{d}}, r_{\mathfrak{d}})$. In the view of the adversary, each component is a random ciphertext in the image of E (which is a group) that encrypts a uniformly random element subject to the condition that the product of $r_1 \cdots r_{\mathfrak{d}} = m'$. This is identical to a ciphertext in the real world where each component encrypts a random element subject to the condition that the product of the random elements is m'. □

References

1. Armknecht, F., Katzenbeisser, S., Peter, A.: Group homomorphic encryption: characterizations, impossibility results, and applications. Designs, Codes and Cryptography (2012) 1–24
2. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science. FOCS '97, Washington, DC, USA, IEEE Computer Society (1997) 364–
3. Benaloh, J.D.C.: Verifiable Secret-ballot Elections. PhD thesis, Yale University, New Haven, CT, USA (1987) AAI8809191.
4. Cohen, J.D., Fischer, M.J.: A robust and verifiable cryptographically secure election scheme. In: Proceedings of the 26th Annual Symposium on Foundations of Computer Science, Washington, DC, USA, IEEE Computer Society (1985) 372–382
5. Cramer, R., Franklin, M.K., Schoenmakers, B., Yung, M.: Multi-authority secret-ballot elections with linear work. In Maurer, U.M., ed.: EUROCRYPT. Volume 1070 of Lecture Notes in Computer Science., Springer (1996) 72–83

6. Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. In Fumy, W., ed.: Advances in cryptology — EUROCRYPT '97: International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11–15, 1997: proceedings. Volume 1233 of Lecture Notes in Computer Science., Berlin, Germany / Heidelberg, Germany / London, UK / etc., Springer-Verlag (1997) 103–118 Sponsored by the International Association for Cryptologic Research (IACR).
7. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography. PKC '01, London, UK, UK, Springer-Verlag (2001) 119–136
8. Naor, M., Pinkas, B.: Oblivious polynomial evaluation. SIAM J. Comput. **35** (2006) 1254–1281
9. Sander, T., Young, A.L., Yung, M.: Non-interactive cryptocomputing for nc^1. In: FOCS, IEEE Computer Society (1999) 554–567
10. Fischlin, M.: A cost-effective pay-per-multiplication comparison method for millionaires. In Naccache, D., ed.: CT-RSA. Volume 2020 of Lecture Notes in Computer Science., Springer (2001) 457–472
11. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28** (1984) 270–299 See also preliminary version in 14th STOC, 1982.
12. Naccache, D., Stern, J.: A new public key cryptosystem based on higher residues. In Gong, L., Reiter, M.K., eds.: ACM Conference on Computer and Communications Security, ACM (1998) 59–66
13. Okamoto, T., Uchiyama, S.: A new public-key cryptosystem as secure as factoring. Lecture Notes in Computer Science **1403** (1998) 308–318
14. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In Stern, J., ed.: EUROCRYPT. Volume 1592 of Lecture Notes in Computer Science., Springer (1999) 223–238
15. Gjøsteen, K.: Homomorphic cryptosystems based on subgroup membership problems. In: Proceedings of the 1st international conference on Progress in Cryptology in Malaysia. Mycrypt'05, Berlin, Heidelberg, Springer-Verlag (2005) 314–327
16. Gjsteen, K.: Symmetric subgroup membership problems. In Vaudenay, S., ed.: Public Key Cryptography. Volume 3386 of Lecture Notes in Computer Science., Springer (2005) 104–119
17. Damgrd, I.: Towards practical public key systems secure against chosen ciphertext attacks. In Feigenbaum, J., ed.: CRYPTO. Volume 576 of Lecture Notes in Computer Science., Springer (1991) 445–456
18. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications security. CCS '06, New York, NY, USA, ACM (2006) 89–98
19. Shamir, A.: Identity-based cryptosystems and signature schemes. Lecture Notes in Computer Science **196** (1985) 47–53
20. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. Journal of the Association for Computing Machinery **45** (1998) 965–981
21. Oliveira, L., Scott, M., Lopez, J., Dahab, R.: Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. In: Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on. (2008) 173–180
22. Liu, A., Ning, P.: Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In: IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks, Washington, DC, USA, IEEE Computer Society (2008) 245–256
23. Oliveira, L.B., Aranha, D.F., Morais, E., Daguano, F., L?pez, J., Dahab, R.: Tinytate: Computing the tate pairing in resource-constrained sensor nodes. Network Computing and Applications, IEEE International Symposium on **0** (2007) 318–323
24. Szczechowiak, P., Kargl, A., Scott, M., Collier, M.: On the application of pairing based cryptography to wireless sensor networks. In: WiSec '09: Proceedings of the second ACM conference on Wireless network security, New York, NY, USA, ACM (2009) 1–12
25. De Cristofaro, E., Soriente, C.: Short paper: Pepsi—privacy-enhanced participatory sensing infrastructure. In: Proceedings of the Fourth ACM Conference on Wireless Network Security. WiSec '11, New York, NY, USA, ACM (2011) 23–28
26. Cristofaro, E.D., Soriente, C.: Extended capabilities for a privacy-enhanced participatory sensing infrastructure (PEPSI). IEEE Transactions on Information Forensics and Security **8** (2013) 2021–2033
27. Cristofaro, E.D., Soriente, C.: Participatory privacy: Enabling privacy in participatory sensing. IEEE Network **27** (2013) 32–36
28. Günther, F., Manulis, M., Peter, A.: Privacy-enhanced participatory sensing with collusion resistance and data aggregation. In: Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings. (2014) 321–336
29. Boneh, D., LaVigne, R., Sabin, M.: Identity-based encryption with eth residuosity and its incompressibility. Project report, Stanford 2013. `http://www.truststc.org/education/reu/13/Papers/SabinM_Paper.pdf` (2012) `http://www.truststc.org/education/reu/13/Papers/SabinM_Paper.pdf`.

30. 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
31. 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
32. Joye, M.: On identity-based cryptosystems from quadratic residuosity. `http://joye.site88.net/papers/gcocks.pdf` (2015) `http://joye.site88.net/papers/gcocks.pdf`.
33. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. J. Cryptology **26** (2013) 39–74
34. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS, IEEE Computer Society (2013) 40–49
35. Clear, M., McGoldrick, C.: Bootstrappable identity-based fully homomorphic encryption. In: Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings. (2014) 1–19
36. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: Deniable encryption, and more. IACR Cryptology ePrint Archive **2013** (2013) 454
37. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In Sako, K., Sarkar, P., eds.: ASIACRYPT (2). Volume 8270 of Lecture Notes in Computer Science., Springer (2013) 280–300
38. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In Krawczyk, H., ed.: Public Key Cryptography. Volume 8383 of Lecture Notes in Computer Science., Springer (2014) 501–519
39. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In Sadeghi, A.R., Gligor, V.D., Yung, M., eds.: ACM Conference on Computer and Communications Security, ACM (2013) 669–684
40. LaVigne, R.: Simple homomorphisms of cocks ibe and applications. IACR Cryptology ePrint Archive **2016** (2016) 1150
41. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS, IEEE Computer Society (2007) 647–657
42. 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
43. 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
44. Gentry, C.: Practical identity-based encryption without random oracles. In Vaudenay, S., ed.: EUROCRYPT. Volume 4004 of Lecture Notes in Computer Science., Springer (2006) 445–464
45. Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In Halevi, S., ed.: CRYPTO. Volume 5677 of Lecture Notes in Computer Science., Springer (2009) 619–636
46. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology. EUROCRYPT'08, Berlin, Heidelberg, Springer-Verlag (2008) 146–162