A Note on Attribute-Based Group Homomorphic Encryption

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 supports 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. We then examine the algebraic structure on attributes induced by the group operation in an ABGHE. This algebraic stricture is a bounded semilattice. We consider some possible semilattices and how they can be realized by an ABGHE supporting inner product predicates. We then examine existing schemes from the literature and show that they meet our definition of ABGHE for either an additive or multiplicative homomorphism. Some of these 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.

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 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

[*]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").

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 the algebraic structure on attributes induced by the group operation in an ABGHE. This algebraic stricture is a bounded semilattice. We consider some possible semilattices and how they can be realized by an ABGHE supporting inner product predicates. We then examine existing schemes from the literature and show that they meet our definition of ABGHE for either an additive or multiplicative homomorphism. Some of these 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.

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 [29], which was shown to be inherently XOR-homomorphic by Joye [30].

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 3 (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 3 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 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 3. 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 $D_{\mathsf{sk}_f}(c * c') = D_{\mathsf{sk}_f}(c) \cdot D_{\mathsf{sk}_f}(c')$.

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

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 [31] 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 3 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 3, 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 3 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 3 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 Attribute Semilattices

The group operation on the ciphertext space induces a binary operation on attributes. We have a binary operation $\odot : \mathbb{A} \times \mathbb{A} \to \mathbb{A}$ that is associative, commutative, idempotent and has an associated identity element in $\mathbb{A}$, Therefore $(\mathbb{A}, \odot)$ is a bounded semilattice (commutative idempotent monoid). Suppose we have ciphertext c associated with attribute $a_1 \in \mathbb{A}$ and ciphertext c_2 associated with attribute $a_2 \in \mathbb{A}$. Then evaluating $c_1 * c_2$ gives a ciphertext that is associated with the attribute $a_1 \odot a_2$.

In an ABGHE scheme, to ensure correctness and semantic security, an attribute semilattice $(\mathbb{A}, \odot)$ with respect to a class of access policies $\mathbb{F}$ must satisfy the following property.

- For every $f \in \mathbb{F}$, f is a semilattice homomorphism between $(\mathbb{A}, \odot)$ and $(\{0,1\}, \wedge)$; that is, for all $a, b \in \mathbb{A}$

$$f(a \odot b) = f(a) \wedge f(b) \tag{4.1}$$

We will now consider some examples of attribute semilattices for particular useful applications. In these cases, the attribute semilattice is finite and each element has a polynomial-size representation.

4.1 Sets

First we consider an application where the attributes are sets and policies decide subset inclusion. Let U be a finite set that we will call the *universe*. We define the attribute space $\mathbb{A}$ as $\mathbb{A} \triangleq \{A : A \subseteq U, |A| = \mathsf{poly}(1^\lambda)\}$ and we define the class of access policies $\mathbb{F}$ thus

$$\mathbb{F} \triangleq \{A \mapsto \begin{cases} 1 & \text{if } S \subseteq A \\ 0 & \text{otherwise} \end{cases} : S \in \mathbb{A}\}.$$

We define the $\odot$ operation for any two attributes $A, B \in \mathbb{A}$ as

$$A \odot B = A \cap B.$$

It is easy to see that $(\mathbb{A}, \odot)$ is a bounded semilattice with U as the identity element and furthermore that every function in $\mathbb{F}$ is a semilattice homomorphism from $(\mathbb{A}, \odot)$ to $(\{0,1\}, \wedge)$.

4.2 Vector Matching

In the next application, the attributes are binary vectors and the policies specify a pattern that is to be matched. The pattern may include wildcards ("don't cares"), denoted by the "*" symbol. We define the attribute space as $\mathbb{A} \triangleq \{0,1,*\}^n$, a set of vectors of length n. Note that we need to include the wildcard element in the definition as we will need it below, but an encryptor generating a fresh ciphertext would most likely opt not to use it, choosing instead a binary vector.

We associate with an access policy a vector $\boldsymbol{w} \in \{0,1,*\}^n$, and define a predicate $f_{\boldsymbol{w}}$ indexed by $\boldsymbol{w}$ as follows

$$f_{\boldsymbol{w}}(\boldsymbol{v}) = \bigwedge_{i \in [n]} w_i = v_i \vee w_i = *.$$

The class of access policies is $\mathbb{F} \triangleq \{f_{\boldsymbol{w}} : \boldsymbol{w} \in \{0,1,*\}^n\}$.

Now we are ready to specify the semilattice operation $\odot$. Intuitively it works by retaining matching components and encoding non-matching components with the wildcard symbol. Formally, we first define a function $g : \{0,1,*\} \times \{0,1,*\} \to \{0,1,*\}$ as follows

$$g(u,v) = \begin{cases} u \text{ if } u = v \\ * \text{ otherwise} \end{cases}.$$

Then we define $\odot$ as

$$\boldsymbol{u} \odot \boldsymbol{v} = (g(u_1, v_1), \ldots, g(u_n, v_n)).$$

Now $(\mathbb{A}, \odot)$ is a semilattice and every policy in $\mathbb{F}$ is a semilattice homomorphism to $(\{0,1\}, \wedge)$ as required. However this semilattice does not have an identity element. For completeness, we can add a logical identity element $\boldsymbol{e}$ to $\mathbb{A}$ and further define $\odot$ to treat it as such.

This semilattice is a special case of the first example by choosing appropriate sets.

4.3 Integer Vector Comparison

Let $B = \mathsf{poly}(\lambda)$ be a positive integer. In this example, the attributes are vectors in $\{0, \ldots, B\}^n$, of length n, and the policies decide whether all the components of an associated vector are less than the corresponding components of an attribute vector. We set $\mathbb{A} \triangleq \{0, \ldots, B\}^n$.

We associate with an access policy a vector $\boldsymbol{w} \in \{0, \ldots, B\}^n$, and define a predicate $f_{\boldsymbol{w}}$ indexed by $\boldsymbol{w}$ as follows

$$f_{\boldsymbol{w}}(\boldsymbol{v}) = \bigwedge_{i \in [n]} w_i < v_i.$$

The class of access policies is $\mathbb{F} \triangleq \{f_{\boldsymbol{w}} : \boldsymbol{w} \in \{0, \ldots, B\}^n\}$.

The $\odot$ operation is the component-wise minimum of two attribute vectors; that is,

$$\boldsymbol{u} \odot \boldsymbol{v} = (\min(u_1, v_1), \ldots, \min(u_n, v_n)).$$

Now $(\mathbb{A}, \odot)$ is a bounded semilattice and every policy in $\mathbb{F}$ is a semilattice homomorphism to $(\{0,1\}, \wedge)$ as required. The identity element is $(B, \ldots, B)$. This semilattice is also a special case of the first example by choosing appropriate sets.

Remark 1. Note that the above semilattices are meet-semilattices. Join-semilattices can be obtained from duality i.e. replacing intersection with union, U with $\emptyset$, subset with superset, min with max and so on.

5 Inner Product Predicates

We will now discuss a rich and expressive class of access policies known as inner product predicates. Let m be some modulus and let n be a positive integer that is polynomial in the security parameter. 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 $\boldsymbol{w} \in \mathbb{Z}_m^n$, a predicate $f_{\boldsymbol{w}} : \mathbb{Z}_m^n \to \{0,1\}$ is defined by

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

Inner product predicates can realize the access policies discussed in the three examples given in the previous section. Note that in the case of the first example, the universe U is required to by of polynomial size. We will consider an encoding for the first example, namely subset inclusion, and remind the reader that the other examples can be viewed as special cases of this, although there is a more direct encoding for vector matching (see [32]). The encoding we consider is based on the idea that a subset S of U can be represented as a binary vector of dimension $n = |U|$, the characteristic vector of S. An attribute $A \subseteq U$ is encoded as the "inverted" characteristic vector of A in which a zero indicates membership of the set and a one indicates non-membership. As such, U is encoded as the zero vector. On the other hand, a predicate corresponding to a subset $S \subseteq U$ is encoded as the characteristic vector of S i.e. a one indicates membership of the set and a zero indicates non-membership. As such, a predicate corresponding to the empty set is encoded as the zero vector. It is easy to see that if $S \subseteq A$, then the inner product of their two encodings is zero, and if $S \not\subseteq A$, then the inner product of their two encodings is non-zero provided $n < m$. For correctness and security, we require that $m > n$.

5.1 ABGHE for Inner Product Predicates

Katz, Sahai and Waters (KSW) [32] (Appendix C) present a scheme that satisfies the properties of an ABGHE and supports inner product predicates. The security of KSW relies on non-standard assumptions on bilinear groups, assumptions that are justified by the authors of [32] in the generic group model.

In KSW, we take m to be the product of three "large" primes. Roughly speaking, in a ciphertext, all components of its attribute vector $\boldsymbol{v} \in \mathbb{Z}_m^n$ 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 $\boldsymbol{w}$ has to satisfy $\langle \boldsymbol{w}, \boldsymbol{a_1} \rangle = 0$ and $\langle \boldsymbol{w}, \boldsymbol{a_2} \rangle = 0$ for decryption of $\boldsymbol{c'}$ to succeed (except with negligible probability). This "simulates" the semilattice operation in the previous section (where the elements of the semilattice are encoded as above), ensuring the property given in 4.1 is satisfied. Therefore KSW can concretely realize the semilattices in the previous section.

Furthermore, the effect of the pairwise product on two ciphertexts 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 Additive and Multiplicative Homomorphisms

The only additively homomorphic ABGHE schemes we are aware of are IBGHE schemes. Clear, Hughes and Tewari [31] present an XOR-homomorphic variant

of the Cocks IBE scheme [29] which has a security reduction from the quadratic residuosity problem. This construction is shown in [31] to satisfy the properties of an ABGHE for the XOR operation. Joye [30] 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 [31] 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 [33] to 2 elements of $\mathbb{Z}_N$ (like Cocks). The scheme however is not an ABGHE but an ABQHE.

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 1. *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 3 since one cannot perform an unbounded number of homomorphic operations.

Günther et al. [28] modified the Boneh-Franklin IBE [34] 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 [35, 36].

As note earlier, KSW is multiplicatively homomorphic and a *bounded* additive homomorphism can be obtained via the above transformation. It does however support a richer class of access policies than IBE.

It is an open problem to construct additively homomorphic ABGHE for a rich class of access policies such as inner product predicates; that is, to find a scheme that is group homomorphic for the plaintext group $(\mathbb{Z}_m, +)$ for some modulus m with simultaneous support for inner product predicates.

7 Possibility Result for IBGHE from Indistinguishability Obfuscation

It is interesting to consider whether we can give a possibility result for ABGHE by relying on indinstuishability obfuscation [37]. It was shown in [38] that attribute-based FHE can be realized from indistinguishability obfuscation. The authors use the technique of punctured programming [39], which involves using indistinguishability obfuscation together with a puncturable pesudorandom function (PRF) [40–42]. 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 2. *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 [38] 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).

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. 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
30. Joye, M.: On identity-based cryptosystems from quadratic residuosity. `http://joye.site88.net/papers/gcocks.pdf` (2015) `http://joye.site88.net/papers/gcocks.pdf`.
31. 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
32. 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

33. LaVigne, R.: Simple homomorphisms of cocks ibe and applications. IACR Cryptology ePrint Archive **2016** (2016) 1150
34. 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
35. 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
36. 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
37. 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
38. 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
39. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: Deniable encryption, and more. IACR Cryptology ePrint Archive **2013** (2013) 454
40. 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
41. 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
42. 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