Policy-based Non-interactive Outsourcing of Computation using Multikey FHE and CP-ABE

Michael Clear* and Ciarán McGoldrick
School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

Keywords: Non-interactive Computing Delegation, Multikey FHE, CP-ABE, Homomorphic Encryption, Access Policy Composition

Abstract: We consider the problem of outsourced computation that operates on encrypted inputs supplied by multiple independent parties. To facilitate fine-grained access control, it would be desirable if each party could encrypt her input under an appropriate access policy. Moreover, a party should only be authorized to decrypt the result of a computation performed on a set of encrypted inputs if his credentials satisfy the composition of all input policies. There has been limited success so far achieving homomorphic encryption in the functional setting; that is, for primitives such as Ciphertext-Policy Attribute Based Encryption (CP-ABE) and Identity Based Encryption (IBE). We introduce a new primitive that captures homomorphic encryption with support for access policies and policy composition. We then present a generic construction using CP-ABE and multikey Fully-Homomorphic encryption (FHE). Furthermore, we show that a CP-ABE scheme that is homomorphic for circuits of polylogarithmic depth in some parameter m implies a CP-ABE scheme that is homomorphic for circuits of arity m and unbounded depth.

1 INTRODUCTION

With the advent of cloud computing, there is a rapidly expanding interest in using remote data centers to perform large computational tasks. Many organizations do not have the computational resources to perform such tasks and the low cost,scalable and highly available model offered by remote providers present an attractive option to organizations. A significant downside of delegating computing jobs to the cloud is the risk of exposure of the delegator's sensitive data. Indeed, sending such data in an unencrypted form may be strictly prohibited by government and industry policies. A number of cryptographic primitives have been proposed to preserve privacy in computing tasks carried out by untrusted or semi-trusted parties. A well-known example is fully-homomorphic encryption (FHE), which was first realized in 2009 by Gentry (Gentry, 2009). FHE allows us to outsource a computation to a cloud provider in such a way that the cloud provider can carry out the computation without being able to see the inputs and outputs. Gentry's construction is public-key and thus allows public delegatability insofar as the sender(s) of inputs to the cloud need not have access to the secret key needed to decrypt the result. Therefore, multiple encryptors may independently contribute data that is to be (potentially) incorporated into a large remote computation.

1.1 The Problem Domain

In standard public-key FHE, there is only a single target recipient. This may be ill-suited to the needs of a large organization. Consider a scenario where staff have restricted access to data based on their department and position. The organization has opted to avail of the computational resources of a cloud provider for the purpose of delegating sizeable computational tasks. Each sender of data acts independently since they are potentially unaware of other's participation.

To comply with the organization's privacy regulations, each sender must encrypt her data under an appropriate access policy that specifies the credentials a staff member must have in order to access the data (or any derivative thereof). We assume such an access policy is feasibly determined from the data source and context.

The computation to be performed, and the inputs to be used, may be decided at a later stage by a sub-

*The author's work is funded by the Irish Research Council EMBARK Initiative.

Clear M. and McGoldrick C..
Policy-based Non-interactive Outsourcing of Computation using Multikey FHE and CP-ABE.
DOI: 10.5220/0004534304440452
In *Proceedings of the 10th International Conference on Security and Cryptography* (SECRYPT-2013), pages 444-452
ISBN: 978-989-8565-73-0

set of the senders, or another delegated authority. The results of the computation are then subsequently returned to the organization, and they should *only* be accessible to a given staff member if her credentials satisfy the cumulative policies associated with *all* the inputs used.

One solution is to use public-key FHE together with a trusted access control system (ACS), which holds the private key for the FHE scheme. The role of the ACS is to grant users (i.e. staff members in the above scenario) access to a plaintext after verifying that their credentials satisfy the policy set associated with the corresponding ciphertext. Access control of this form facilitates expressive policies. However, it must be used in conjunction with a cryptographic primitive such as a non-interactive zero-knowledge proof system as, otherwise, unauthorized users may collude in order to report an incorrect policy.

This approach suffers from a number of drawbacks:

- All parties interested in a result are required to contact the ACS, which must remain online and exhibit high availability in order to guarantee satisfactory responsiveness. The ACS may therefore act as a bottleneck, especially under high load scenarios.
- Adhering to the principle of least privilege, the organization may wish to limit the capabilities of the ACS. In particular, it may have reservations about the ACS being compromised, and potentially providing an attacker access to all results returned from the cloud.
- Remote users, with appropriate valid credentials, cannot directly query the cloud for data and decrypt non-interactively. All requests must be routed via the organization's ACS.

Many of these shortcomings are flexibly addressed through a functional encryption (FE) approach. In the FE setting, a trusted authority (TA) authenticates and authorizes users by issuing them secret keys for certain *capabilities*. For our purposes, we deal with a special case of FE known as ciphertext-policy attribute-based encryption (CP-ABE) where the *capabilities* correspond to credentials or *attributes*. Note that we use the term *attribute* here to refer to (collectively) what some authors describe as a particular set of attributes. A user with a secret key for an attribute a can decrypt any ciphertext encrypted under a policy satisfied by a. A principal advantage of CP-ABE over an ACS-based solution is that once the user is issued a secret key for a, no further interaction with the TA is required (for a certain period of time i.e. a may be time-limited) throughout which the user can decrypt an arbitrary number of ciphertexts non-interactively. The advantages of ABE in distributed environments have been investigated in other work, such as (Pirretti et al., 2010). Although CP-ABE has some deficiencies, such as inherent escrow (which the ACS approach suffers from also) and a lack of support for revocation, it is well-suited to achieving fine-grained access control with minimal interaction.

It is not trivial to reconcile the features of FHE and CP-ABE. There are currently no known fully-homomorphic CP-ABE schemes. Indeed, there are also no fully-homomorphic identity-based encryption (IBE) schemes, which is a weaker primitive than CP-ABE.

1.2 Contributions

In this work, we propose a syntax for a more general primitive which seeks to capture the requirements of the problem space described above, while incorporating properties from FHE and CP-ABE. We call this primitive *policy-based homomorphic encryption* (PBHE). The formulation of PBHE extends the recent definition of multikey FHE by Lopez-Alt, Tromer and Vaikuntanathan (López-Alt et al., 2012). Central to PBHE is the notion of access policy composition, and we define the syntax and the correctness properties of PBHE in terms of an algebraic structure defined on access policies. PBHE can be instantiated by any homomorphic CP-ABE scheme or any standard homomorphic public-key cryptosystem.

Another contribution of this work is the construction of a new PBHE scheme that supports fully-homomorphic evaluation of circuits whose input ciphertexts are encrypted under a bounded number of independently-chosen policies. This scheme fulfills the requirements of the scenario outlined above for a bounded number of senders. This bound is polynomial in the security parameter.

Finally, and leveraging the work of (López-Alt et al., 2012), we prove that if a CP-ABE scheme $\mathcal{E}$ is homomorphic for a class of circuits of polylogarithmic depth in a parameter m (which is polynomial in the security parameter), then there exists a scheme $\mathcal{E}'$ that is homomorphic for all circuits with arity m and with arbitrary depth. This is a significant result as obtaining homomorphic CP-ABE for circuits of unbounded depth has been impeded by the fact that there does not seem to be a way to employ bootstrapping in the functional setting (non-interactively) since bootstrapping requires encryptions of the secret key bits to be available as part of the public key.

We note that our work in this paper is limited to

the semi-honest model. In particular, we assume that the cloud is semi-honest. We leave to future work the challenge of securing against malicious adversaries, especially in verifying that a function was evaluated correctly.

1.3 Related Work

Homomorphic encryption in a multi-user setting is considered in (Xiao et al., 2012). The authors propose a new homomorphic symmetric encryption scheme that is shown to be secure under a factoring assumption, provided an attacker is limited to obtaining a bounded number (polynomial in the security parameter) of plaintext-ciphertext pairs. The authors also propose a system model with mutually untrusted components that enables a master key for their scheme to be derived from any user's key by splitting it into shares that are distributed to each component by a dealer at system initialization time. However, their solution requires interaction with a server known as a *key agent* for every request/response to/from the cloud. Furthermore, their solution does not support any level of expressive access control.

The notion of multikey FHE was recently presented in (López-Alt et al., 2012) along with a concrete construction based on NTRU (Hoffstein et al., 1998). In that work, multikey FHE is used to construct an "on-the-fly"multi-party computation (MPC) protocol that is secure in the malicious setting. In such an MPC protocol, a number of parties independently send encrypted inputs to an evaluator without interaction. The evaluator then computes a function F over the inputs and sends the encrypted result to each party. It is then possible for the senders to run an interactive MPC protocol to jointly decrypt the result, verify each other's participation, and verify F was honestly computed by the evaluator. While we make use of multikey FHE for our generic construction in Section 5 and as a basis for the syntax of PBHE in Section 4, we address a different problem than (López-Alt et al., 2012) i.e. we do not target MPC wherein each party wishes to keep his input secret. In our protocol, interactive decryption is avoided at the loss of verifiability. Achieving the latter in a meaningful way is a topic for future work.

Additional related work arises in the ABE setting, such as the construction of CP-ABE (Bethencourt et al., 2007), and in the area of access control facilitating access policy composition (Bonatti et al., 2002; Bruns et al., 2007; Ni et al., 2009). More recently (Rao et al., 2011)'s work on policy composition has targeted real-world access control languages like XACML (Moses et al., 2005). In our context the objects to protect are *data*, and the policies are not enforced by a server but rather by an encryption scheme, so it is important to note that the scope for policy composition is far more restrictive as it is necessary to preserve semantic security.

Homomorphic encryption in the functional setting was investigated recently in (Clear et al., 2013) and a group-homomorphic IBE scheme was presented therein. Thus far, the IBE variant of the scheme from (Gentry et al., 2010) is the only IBE scheme to the best of our knowledge that can compactly evaluate 2-DNF formulae.

2 PRELIMINARIES

2.1 Notation

A quantity t is said to be negligible in a parameter κ if it holds that $t < 1/f(\kappa)$ for all polynomials f.

If D is a random variable, the notation $x \xleftarrow{\$} D$ denotes the fact that x is sampled according to the distribution on D. If instead that D is a set, then the notation is understood to mean that x is uniformly sampled from D.

2.2 Access Policies

An access policy is a predicate that grants or denies permission to access a particular object in some specific manner. Some contexts require rich policies that present multiple outcomes for an access. For example, Bruns, Dantas and Huth (Bruns et al., 2007) represent a policy as a four-valued predicate whose range is $\{$*grant*, *deny*, *unspecified* or *conflict*$\}$. Access control systems with these requirements typically accommodate many modes of access to an object. In our case, the objects correspond to data, and it is meaningful in this context to either grant or deny (mutually exclusive) access to a datum. Therefore, we naturally represent an access policy as a two-valued predicate.

2.3 CP-ABE Syntax

A CP-ABE scheme for a class of access policies $\mathbb{F}$ defined over a domain of attributes $\mathbb{A}$ with message space $\mathbb{M}$ is a tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$. As mentioned in the introduction, we refer to the entity that an access policy is applied to as an *attribute* instead of a set of attributes as in (Bethencourt et al., 2007). An attribute a in a domain $\mathbb{A}$ may be viewed as a set of "sub-attributes". Accordingly, we express access policies as predicates i.e. $\mathbb{F} \subseteq \mathbb{A} \rightarrow$

$\{0,1\}$. The Trusted Authority (TA) runs Setup to generate the public parameters PP and a master secret key MSK. It runs $\mathsf{sk}_a \leftarrow \mathsf{Extract}(\mathsf{MSK}, a)$ to derive a secret key for an attribute $a \in \mathbb{A}$.

There are two main definitions of semantic security are distinguished by whether the adversary is allowed to make adaptive requests for secret keys. In the non-adaptive (IND-NA-CPA) game, a challenger hands PP to the adversary $\mathcal{A}$ who can make queries to an extraction oracle $\mathcal{X} := \mathsf{Extract}(\mathsf{MSK}, \cdot)$ to obtain secret keys for certain attributes. At the end of this phase, $\mathcal{A}$ chooses a target policy $f^* \in \mathbb{F}$ and two messages $m_0, m_1 \in \mathbb{M}$. The challenger uniformly samples a bit $b \xleftarrow{\$} \{0,1\}$ and gives an encryption of m_b under f^* to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs a guess bit b' and is said to win if $b = b'$.

The advantage of $\mathcal{A}$ is defined as $\Pr[b' = b] - 1/2$. A CP-ABE scheme is said to be IND-NA-CPA secure if every PPT adversary $\mathcal{A}$ has only a negligible advantage in the above game. In the adaptive variant (IND-AD-CPA), $\mathcal{A}$ is also allowed to make queries to $\mathcal{X}$ after the challenge ciphertext is generated.

2.4 Multikey FHE

Multikey FHE allow multiple independently-generated keys to be used together in a homomorphic evaluation. The syntax of multikey FHE imposes a limit N on the number of such keys that can be supported. Furthermore, the size of the evaluated ciphertext does not depend on the size of the circuit (or number of inputs), but instead on the number of independent keys used. In order to decrypt, the parties who have the corresponding secret keys must collaborate in an MPC protocol.

Definition 2.1 (Based on Definition 2.1 in (López-Alt et al., 2012))**.** *A multikey $\mathbb{C}$-homomorphic scheme family for a class of circuits $\mathbb{C}$ and message space $\mathbb{M}$ is a family of PPT algorithms $\{\mathcal{E}^{(N)} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})\}_{N>0}$ where $\mathcal{E}^{(N)}$ is defined as follows:*

- $\mathsf{MKFHE.Gen}$ *takes as input the security parameter 1^κ and outputs a tuple $(\mathsf{pk}, \mathsf{sk}, \mathsf{vk})$ where pk is a public key, sk is a secret key and vk is an evaluation key.*
- $\mathsf{MKFHE.Enc}$ *takes as input a public key pk and a message $m \in \mathbb{M}$, and outputs an encryption of m under pk.*
- $\mathsf{MKFHE.Dec}$ *takes as input N secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$ and a ciphertext c, and outputs a message $m' \in \mathbb{M}$.*
- $\mathsf{MKFHE.Eval}$ *takes as input a circuit $C \in \mathbb{C}$, and ℓ pairs $(c_1, \mathsf{vk}_1), \ldots, (c_\ell, \mathsf{vk}_\ell)$ and outputs a ciphertext $c^\star$.*

Informally, evaluation is only required to be *correct* if at most N keys are used in $\mathsf{MKFHE.Eval}$; that is, $|\{\mathsf{vk}_1, \ldots, \mathsf{vk}_\ell\}| \leq N$. Furthermore, the size of an evaluated ciphertext $c^\star$ must only depend polynomially on the security parameter κ and the number of keys N, and not on the size of the circuit.

The IND-CPA security game for multikey homomorphic encryption is the same as that for standard public-key encryption; note that the adversary is given the evaluation key vk.

3 HOMOMORPHIC CP-ABE WITH BOUNDED COMPOSITION

Let $\mathbb{F}$ be a set of valid access policies which accept or reject members of a set of attributes $\mathbb{A}$. An access policy $f \in \mathbb{F}$ is represented as a predicate $\mathbb{A} \to \{0,1\}$. Recall that our goal is to facilitate joint computation on encrypted inputs contributed by multiple independent parties, who may be unaware of each other. Moreover, each party has the liberty to encrypt her inputs under an independently-chosen policy. Accordingly, it is necessary to support composition of these policies. Intuitively, one would expect that the result of the joint computation be decryptable by users with an attribute that satisfies the composite policy. Therefore, we introduce a binary composition operation $\odot$ defined on $\mathbb{F}$.

We begin by giving a precise definition of homomorphic CP-ABE. A CP-ABE scheme is homomorphic for a class of circuits $\mathbb{C}$ if there is an additional algorithm Eval and a composition operation $\odot : \mathbb{F}^2 \to \mathbb{F}$ such that over all choices of $f_1, \ldots, f_\ell \in \mathbb{F}$, $m_1, \ldots, m_\ell \in \mathbb{M}$, $c_1 \leftarrow \mathsf{Enc}(\mathsf{PP}, f_1, m_1), \ldots, c_\ell \leftarrow \mathsf{Enc}(\mathsf{PP}, f_\ell, m_\ell)$ and $C \in \mathbb{C}$, the ciphertext $c' \leftarrow \mathsf{Eval}(\mathsf{PP}, C, c_1, \ldots, c_\ell)$ satisfies

- **Correctness**

$$\mathsf{Dec}(\mathsf{sk}_a, c') = C(m_1, \ldots, m_\ell) \text{ iff } f'(a) = 1 \quad (1)$$

  for any $a \in \mathbb{A}$ and $\mathsf{sk}_a \leftarrow \mathsf{Extract}(\mathsf{MSK}, a)$.

- **Compactness**

$$|c'| = \mathsf{poly}(\kappa, |f'|) \quad (2)$$

where $f' = f_1 \odot \ldots \odot f_\ell$.

The main idea in this paper is to exploit multikey FHE and CP-ABE to construct a new CP-ABE scheme that is homomorphic for a class of circuits $\mathbb{C}$ of bounded arity. However, we can only achieve

this for certain policy algebras $(\mathbb{F}, \odot)$. Let $\mathcal{E}_{\mathsf{ABE}}$ be a CP-ABE scheme and let $\mathcal{E}_{\mathsf{MKFHE}}$ be a multikey FHE scheme. Roughly speaking, to encrypt a message m under policy f in our scheme, (1) a key triple $(\mathsf{pk}, \mathsf{vk}, \mathsf{sk})$ is generated for $\mathcal{E}_{\mathsf{MKFHE}}$; (2) m is encrypted with $\mathcal{E}_{\mathsf{MKFHE}}$ under pk; (3) sk is encrypted with $\mathcal{E}_{\mathsf{ABE}}$ under policy f; (4) the two previous ciphertexts along with vk constitute the ciphertext that is produced. Therefore, $\mathcal{E}_{\mathsf{MKFHE}}$ is used for hiding the message and for homomorphic computation whereas $\mathcal{E}_{\mathsf{ABE}}$ enforces the access policies by appropriately hiding the secret keys for $\mathcal{E}_{\mathsf{MKFHE}}$. Technically, it is the number of compositions in our scheme that must be bounded and not the arity of the circuits. However, the former implies the latter due to the syntactic restrictions of homomorphic CP-ABE (See Section 4).

It might seem necessary that $\odot$ be both commutative and associative. However, we only require that these properties hold with respect to semantics. We say that two policies $f, g \in \mathbb{F}$ are *semantically equivalent*, written $f \sim g$, if for all attributes $a \in \mathbb{A}$, we have that $f(a) = g(a)$. Formally, it is required that $\sim$ be a congruence relation with respect to $\odot$ and that $(\mathbb{F}/\sim, \odot)$ be a commutative semigroup. In sum, the properties that $\odot$ must satisfy for any $f, g, h \in \mathbb{F}$ are as follows:

1. $$f \odot g \sim g \odot f \tag{3}$$
2. $$(f \odot g) \odot h \sim f \odot (g \odot h) \tag{4}$$
3. $$(f \odot g)(a) \Rightarrow f(a) \wedge g(a) \tag{5}$$
   for any $a \in \mathbb{A}$.

Note that the last property is necessary for semantic security.

We denote the *size* of a policy $f \in \mathbb{F}$ by its length, written $|f| \in \mathbb{N}$. For some algebras, the size of policies do not always grow with composition. Consider the following semilattices (commutative idempotent semigroups).

- the *Kronecker semilattice* where $\odot$ is defined as:
$$f \odot g = \begin{cases} f & \text{if } f = g \\ z & \text{otherwise} \end{cases}$$
and z is a distinguished policy in $\mathbb{F}$ with the property that $z(a) = 0 \quad \forall a \in \mathbb{A}$.
- the *meet semilattice* where $\odot$ is defined as $\wedge$.

Using our approach as described above, we cannot construct homomorphic CP-ABE for idempotent algebras $(\mathbb{F}, \odot)$ because $|f \odot f| = |f|$, which implies that the compactness condition given by 2 cannot be satisfied since the ciphertexts in our scheme grow with composition. However, we have obtained the following result. Suppose that $\mathcal{E}_{\mathsf{ABE}}$ is a somewhat-homomorphic CP-ABE scheme with a policy algebra $(\mathbb{F}, \odot)$. More precisely, suppose that $\mathcal{E}_{\mathsf{ABE}}$ is homomorphic for a class of circuits $\mathbb{C}$ of depth that is polylogarithmic in the security parameter. Then there exists a CP-ABE scheme for $(\mathbb{F}, \odot)$ that is homomorphic for a class of circuits of arbitrary depth whose arity is bounded by a fixed polynomial in the security parameter. Informally, the theorem gives us a way to trade "breadth" (arity) for depth.

Theorem 3.1. *Let $\mathcal{E}_{\mathsf{ABE}}$ be a CP-ABE scheme with attribute space $\mathbb{A}$, message space $\mathbb{M}_{\mathsf{ABE}}$ and whose policy algebra $(\mathbb{F}, \odot)$ is an idempotent semigroup. Let κ be the security parameter. Let $m = \mathsf{poly}(\kappa)$. If $\mathcal{E}_{\mathsf{ABE}}$ is homomorphic for all circuits whose depth is bounded from above by $O(\log^2 m)$, then there exists a secure CP-ABE scheme that is homomorphic for all circuits of arbitrary depth with at most m inputs.*

The proof is deferred to the extended version (Clear and McGoldrick, 2013) of this work.

4 POLICY-BASED HOMOMORPHIC ENCRYPTION

Our approach is applicable to policy algebras $(\mathbb{F}, \odot)$ where the policy size always grows with composition. An example of such an algebra is the free semigroup $\mathfrak{F}^*$ on a set $\mathfrak{F}$. Moreover, our approach can handle at most N compositions where N is the maximum number of independent users supported by the multikey FHE scheme $\mathcal{E}_{\mathsf{MKFHE}}$. Observe that the inputs encrypted by the same user under the same policy need *not* be composed together with $\odot$. Therefore, the scheme can handle more than N inputs, but at most N independent policies. However, the syntax of homomorphic CP-ABE is too limited to capture this exemption. This fact serves to motivate the formulation of a more general primitive which we refer to as policy-based homomorphic encryption (PBHE). Our formulation of PBHE is influenced considerably by the definition of multikey FHE, and inherits many of its properties.

Definition 4.1. *A Policy-Based Homomorphic Encryption (PBHE) scheme for a class of circuits $\mathbb{C}$, a commutative semigroup of access policies $(\mathbb{F}/\sim, \odot)$ and a set of attributes $\mathbb{A}$ is a family of algorithms $\{\mathcal{E}^{(N)} := (\mathsf{Setup}, \mathsf{Extract}, \mathsf{GenKey}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})\}_{N \geq 1}$ where $\mathcal{E}^{(N)}$ is defined as follows:*

- $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\kappa)$*: Given a security parameter* κ*, output public parameters* PP *and a master secret key* MSK.
- $\mathsf{sk}_a \leftarrow \mathsf{Extract}(\mathsf{MSK}, a)$*: Given a master secret key* MSK *and an attribute* $a \in \mathbb{A}$*, output a secret key* sk_a *for* a.
- $\mathsf{vk}_f \leftarrow \mathsf{GenKey}(\mathsf{PP}, f)$*: Given public parameters* PP *and an access policy* $f \in \mathbb{F}$*, output a pair of encryption and evaluation keys* $(\mathsf{ek}_f, \mathsf{vk}_f)$ *for* f.
- $c \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ek}_f, m)$*: Given public parameters* PP*, an encryption key* ek_f *for policy* f*, and a plaintext* $m \in \mathbb{M}$*, output a ciphertext* c *that encrypts* m *under policy* f.
- $m \leftarrow \mathsf{Dec}(\mathsf{sk}_a, c)$*: Given a secret key* sk_a *for attribute* $a \in \mathbb{A}$ *and a ciphertext* c *that encrypts a message* m *under access policy* f*, output* m *iff* $f(a) = 1$ *and* $\perp$ *otherwise.*
- $c' \leftarrow \mathsf{Eval}(C, (c_1, \mathsf{vk}_1), \ldots, (c_\ell, \mathsf{vk}_\ell))$*: Given a circuit* $C \in \mathbb{C}$ *and a sequence of* ℓ *pairs of ciphertext and evaluation keys, output a ciphertext* c'.

For every $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\kappa)$, every collection of $t \leq N$ access policies $f_1, \ldots, f_t \in \mathbb{F}$, and every collection of key-pairs $\mathsf{K} := \{(\mathsf{ek}_i, \mathsf{vk}_i) \leftarrow \mathsf{GenKey}(\mathsf{PP}, f_i)\}_{i \in [t]}$, every sequence of ℓ tuples $\{(c_i, \mathsf{vk}_{v_i}) : v_i \in [t], c_i \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{ek}_{v_i}, m_i)\}_{i \in [\ell]}$ and all attributes $a \in \mathbb{A}$ and secret keys $\mathsf{sk}_a \leftarrow \mathsf{Extract}(\mathsf{MSK}, a)$, and all circuits $C \in \mathbb{C}$, the following properties are satisfied for every $c' = \mathsf{Eval}(C, (c_1, \mathsf{vk}_{v_1}), \ldots, (c_\ell, \mathsf{vk}_{v_\ell}))$ where $f' = \bigodot_{j \in \{v_i, \ldots, v_\ell\}} f_j$:

- **Correctness:**
  1. $\mathsf{vk}_i = \mathsf{vk}_j \Rightarrow f_i = f_j \quad$ for $i, j \in [t]$.
  2. $\mathsf{Dec}(\mathsf{sk}_a, c') = C(m_1, \ldots, m_\ell)$ iff $f'(a) = 1$ and $\perp$ otherwise.
- **Compactness:** $|c'| = \mathsf{poly}(\kappa, |f'|)$

Informally, the first correctness condition requires that evaluation keys be uniquely associated with an access policy. Besides including information necessary for evaluation, which could instead be embedded in the ciphertext, the main role of an evaluation key is to allow ciphertexts produced by the same encryptor to be grouped together into classes. The composition operation is not applied *among* the members of such classes according to the second correctness condition. In other words, composition is performed on equivalence classes where the equivalence relation is defined by equality of evaluation keys. The motivation for this is to compensate for the non-idempotency of an operation $\odot$. For example, it may be the case that the ciphertexts produced by the same encryptor share information that can be exploited to assist homomorphic computation among them. This is exemplified by multikey FHE.

Security. The semantic security definition for PBHE is similar to that of CP-ABE. We also refer to this as IND-AD-CPA security. In fact, the security game is the same as that for CP-ABE except that the adversary is also given $(\mathsf{ek}, \mathsf{vk}) \leftarrow \mathsf{PBHE.GenKey}(\mathsf{PP}, f^\star)$ after it chooses a target policy $f^\star$. There is a subtlety with respect to the ciphertexts outputted by the evaluation algorithm, namely ensuring that a user can decrypt such ciphertexts if and only if they have a secret key for an attribute that satisfies the *composite* policy. This is explored further in the extended version.

5 CONSTRUCTION OF PBHE

In this section, we construct a new generic PBHE scheme that can be instantiated by an IND-AD-CPA secure CP-ABE scheme together with any IND-CPA secure multikey FHE scheme.

Remark. Concrete constructions of CP-ABE and multi-key FHE already exist which fulfill the properties we need. Examples of the former include (Bethencourt et al., 2007; Waters, 2011) and an example of the latter is the NTRU-based construction from (López-Alt et al., 2012).

Let $\mathcal{E}_{\mathsf{ABE}} = (\mathsf{ABE.Setup}, \mathsf{ABE.Extract}, \mathsf{ABE.Enc}, \mathsf{ABE.Dec})$ be a CP-ABE scheme for a class of policies $\mathbb{F}_{\mathsf{ABE}}$, a set of attributes $\mathbb{A}_{\mathsf{ABE}}$ and a message space $\mathbb{M}_{\mathsf{ABE}}$. Let $\{\mathcal{E}^{(N)}_{\mathsf{MKFHE}} = (\mathsf{MKFHE.Gen}, \mathsf{MKFHE.Enc}, \mathsf{MKFHE.Dec}, \mathsf{MKFHE.Eval})\}_{N>0}$ be a family of multikey fully-homomorphic encryption schemes. In our generic PBHE scheme, the set of attributes $\mathbb{A}$ is defined as $\mathbb{A} \triangleq \mathbb{A}_{\mathsf{ABE}}$. Now we need to define an algebraic structure of access policies $(\mathbb{F}, \odot)$ that obeys the three properties given by 3, 4 and 5.

5.1 Supported Access Policies and Composition

Define a subset $\mathfrak{F} \subseteq \mathbb{F}_{\mathsf{ABE}}$ that is closed under $\wedge$. We define $(\mathbb{F}, \odot)$ as the free semigroup $\mathfrak{F}^*$ on $\mathfrak{F}$ i.e. the set of finite strings composed of elements of $\mathfrak{F}$. For brevity, we will use juxtaposition instead of explicitly writing $\odot$ when representing policies in $\mathbb{F}$. The semantic interpretation of a policy $\mathfrak{f}_1 \ldots \mathfrak{f}_\ell \in \mathbb{F}$ is such that the following holds

$$\mathfrak{f}_1 \ldots \mathfrak{f}_\ell \sim \mathfrak{f}_1 \wedge \ldots \wedge \mathfrak{f}_\ell.$$

Moreover, the size $|f|$ of a policy $f = \mathfrak{f}_1 \ldots \mathfrak{f}_\ell$ is the sum $\sum_{i=1}^{\ell} |\mathfrak{f}_i|$. This is to be distinguished from the *length* of a policy in $f \in \mathbb{F}$, written $\lambda(f)$, which is the

length of the corresponding string of elements from $\mathfrak{F}$.

Note that any policy $f \in \mathbb{F}$ can be transformed into a semantically equivalent policy f' with $\lambda(f') = 1$. Thus, we assume without loss of generality that this is what the GenKey algorithm takes as input.

5.2 Key Trees

Now we show how our PBHE scheme enforces access policies in $\mathbb{F}$ and how it handles composition. By abuse of notation, we write $\mathsf{ABE.Enc}(\mathsf{PP}, f, M)$ for $M \notin \mathbb{M}_{\mathsf{ABE}}$ to signify the encryption of multiple elements of $\mathbb{M}_{\mathsf{ABE}}$ in order to "cover" M. The analogous notion is also assumed for decryption.

Let f be a policy in $\mathbb{F}$. Our approach involves mapping f to a binary tree τ_f (which we call a *key tree*) whose nodes are associated with ciphertexts in the CP-ABE scheme. Each leaf node corresponds to an element of $\mathfrak{F}$ while an interior node corresponds to the conjunction of its left and right branches. More precisely, the leaf nodes are encryptions of the secret keys for $\mathcal{E}_{\mathsf{MKFHE}}$ XORed with a random *blinding* string. An interior node encrypts the concatenation of the blinding strings of its child nodes XORed with a new blinding string. Thus, in order to access the secret keys at the leaves, it is necessary to decrypt from the root down. Thus, if a user's attribute satisfies the root policy, she can decrypt every layer and eventually recover the secret keys hidden by the ciphertexts at the leaves. Indeed, satisfying the root policy is a sufficient and necessary condition to recover *any* secret key.

A key tree τ_f for a policy $f := \mathfrak{f}_1 \dots \mathfrak{f}_\ell$ consists of a list of CP-ABE ciphertexts $[\psi_i]_{i \le 2\ell-1}$. We associate with each leaf node in τ_f a unique key tuple $(\mathsf{pk}, \mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{MKFHE.Gen}(1^\kappa)$ in the multikey FHE scheme. Roughly speaking, we set the leaf node of τ_f to an encryption of $\mathsf{sk} \oplus r$ under $\mathfrak{f}$ in the CP-ABE scheme where r is a random string of length $|\mathsf{sk}|$. Every tree is associated with such a random value, and thus for convenience, we will sometimes refer to a pair $(r, [\psi_1]_{i \le 2^{h+1}-1})$ as a "tree".

Now to construct a key tree for a policy f, consider an algorithm $\mathsf{MkTree}^\star$ which proceeds as follows:

1. On input $f \in \mathbb{F}$, decompose f into $\mathfrak{f}_1 \dots \mathfrak{f}_k$.
2. For $1 \le i \le k$:
   (a) Set $(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{vk}_i) \leftarrow \mathsf{MKFHE.Gen}(1^\kappa)$.
   (b) Uniformly sample $r_i \xleftarrow{\$} \{0,1\}^{|\mathsf{sk}_i|}$.
   (c) Compute $\psi_i \leftarrow \mathsf{ABE.Enc}(\mathsf{PP}, \mathfrak{f}_i, r_i \oplus \mathsf{sk}_i)$.
   (d) Set $\tau_i \leftarrow (r_i, [\psi_i])$.
3. For $1 \le i \le \lceil \lg k \rceil$:
   (a) For $1 \le j \le \lceil k/2^i \rceil$:
      i. If $2j > \lceil k/2^{i-1} \rceil$, set $\tau_j \leftarrow \tau_{2j-1}$.
      ii. Else set $\tau_j \leftarrow \mathsf{Combine}(\tau_{2j-1}, \tau_{2j})$
4. Output $(\tau_1, (\mathsf{pk}_1, \mathsf{vk}_1), \dots, (\mathsf{pk}_k, \mathsf{vk}_k))$.

where Combine is defined below.

Note that we denote by MkTree the variant of $\mathsf{MkTree}^\star$ that outputs only the first component of the tuple outputted by $\mathsf{MkTree}^\star$, namely the tree τ_1.

To combine two trees $\tau_f := (r, [\chi_i]_{i \le 2\ell_1-1})$ and $\tau_g := (s, [\psi_i]_{i \le 2\ell_2-1})$ for policies $f := \mathfrak{f}_1 \dots \mathfrak{f}_{\ell_1}$ ($g := \mathfrak{g}_1 \dots \mathfrak{g}_{\ell_2}$ resp.), the following algorithm is used (we refer to this algorithm as Combine):

1. Uniformly sample $t \xleftarrow{\$} \{0,1\}^{|r\|s|}$.
2. Compute $\omega \leftarrow \mathsf{ABE.Enc}(\mathsf{PP}, \mathfrak{f}' \wedge \mathfrak{g}', \oplus(r \parallel s))$ where $\mathfrak{f}' = \mathfrak{f}_1 \wedge \dots \wedge \mathfrak{f}_{\ell_1}$ and $\mathfrak{g}' = \mathfrak{g}_1 \wedge \dots \wedge \mathfrak{g}_{\ell_2}$.
3. Construct the tree $(t, [\omega, \chi_1, \dots, \chi_{2\ell_1-1}, \psi_1, \dots, \psi_{2\ell_2-1}])$.

Decrypting a tree with a secret key sk_a for an attribute $a \in \mathbb{A}$ is defined recursively:

- $$\mathsf{DecTree}(\mathsf{sk}_a, (t, [\omega])) = \begin{cases} [\pi \oplus t] & \text{if } \pi \neq \perp \\ [] & \text{otherwise} \end{cases}$$
  where $\pi = \mathsf{ABE.Dec}(\mathsf{sk}_a, \omega)$.
- $$\mathsf{DecTree}(\mathsf{sk}_a, (t_1 \parallel t_2, [\omega] \parallel [\chi_i]_{i \le 2\ell_1-1} \parallel [\psi_i]_{i \le 2\ell_2-1})) =$$
  $$\begin{cases} \mathsf{DecTree}(\mathsf{sk}_a, (r \oplus t_1, [\chi_i]_{i \le 2\ell_1-1})) \\ \parallel \mathsf{DecTree}(\mathsf{sk}_a, (s \oplus t_2, [\psi_i]_{i \le 2\ell_2-1})) & \text{if } \pi = (r\|s) \\ [] & \text{if } \pi = \perp \end{cases}$$
  where $\pi = \mathsf{ABE.Dec}(\mathsf{sk}_a, \omega)$.

Therefore, DecTree produces a list of secret keys for $\mathcal{E}_{\mathsf{MKFHE}}$, which may be empty if the attribute does not satisfy the associated policy.

5.3 Construction

For brevity, we will assume that all policies f passed as input to GenKey satisfy $\lambda(f) = 1$. Our PBHE scheme $\mathcal{E}_{\mathsf{PBHE}}$ is defined as follows:

- $\mathsf{Setup}(1^\kappa)$: Given a security parameter κ, generate $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{ABE.Setup}(1^\kappa)$. Output $(\mathsf{PP}, \mathsf{MSK})$.
- $\mathsf{Extract}(\mathsf{MSK}, a)$: Given a master secret key MSK and an attribute $a \in \mathbb{A}$, output $\mathsf{sk}_a \leftarrow \mathsf{ABE.Extract}(\mathsf{MSK}, a)$.

- $\mathsf{GenKey}(\mathsf{PP}, f)$: Given public parameters PP and an access policy $f \in \mathbb{F}$, run:
  1. On assumption (above) f can be parsed as $\mathfrak{f}$ where $\mathfrak{f} \in \mathfrak{F}$.
  2. Compute $(\tau, (\mathsf{pk}, \mathsf{vk})) \leftarrow \mathsf{MkTree}^{\star}(f)$.
  3. Set $\mathsf{ek} \leftarrow ((\mathsf{pk}, \mathsf{vk}), \tau)$.
  4. Output $(\mathsf{ek}, \mathsf{vk})$.
- $\mathsf{Enc}(\mathsf{PP}, \mathsf{ek}_f, m)$: Given public parameters PP, an encryption key ek_f for policy f, and a plaintext $m \in \mathbb{M}$, run:
  1. Parse ek_f as $((\mathsf{pk}, \mathsf{vk}), \tau)$.
  2. Compute $c^* \leftarrow \mathsf{MKFHE.Enc}(\mathsf{pk}, m)$.
  3. Output $\mathbf{c} := (c^*, \tau)$.
- $\mathsf{Dec}(\mathsf{sk}_a, \mathbf{c})$: Given a secret key sk_a for attribute $a \in \mathbb{A}$ and a ciphertext $\mathbf{c}$ that encrypts a message m under access policy f, run:
  1. Parse $\mathbf{c}$ as (c^*, τ).
  2. If $\mathsf{DecTree}(\mathsf{sk}_a, \tau) = []$ (empty list), then output $\perp$ and abort.
  3. Set $[\mathsf{sk}_1, \ldots, \mathsf{sk}_k] \leftarrow \mathsf{DecTree}(\mathsf{sk}_a, \tau)$
  4. Compute $m \leftarrow \mathsf{MKFHE.Dec}(\mathsf{sk}_1, \ldots, \mathsf{sk}_k, c^*)$.
  5. Output m.
- $\mathsf{Eval}(C, (\mathbf{c_1}, \mathsf{vk}_1), \ldots, (\mathbf{c}_\ell, \mathsf{vk}_\ell))$: Given a circuit $C \in \mathbb{C}$ and a sequence of ℓ pairs of ciphertext and evaluation keys, perform the following steps:
  1. Parse each $\mathbf{c_i}$ as (c_i^*, τ_i).
  2. Set $V := \{\mathsf{vk}_1, \ldots, \mathsf{vk}_\ell\}$.
  3. For each $\mathsf{vk} \in V$, set $\mathfrak{C}_{\mathsf{vk}} := \{c_i^* : \mathsf{vk}_i = \mathsf{vk}\}$.
  4. Set $\mathfrak{T} := \{\tau_i\}_{i \in [\ell]}$ (recall that ciphertexts encrypted under the same vk have the same τ component).
  5. Run $\mathsf{Combine}$ to recursively build a tree τ from all elements in $\mathfrak{T}$.
  6. Set $c^* \leftarrow \mathsf{MKFHE.Eval}(C, (c_1, \mathsf{vk}_1), \ldots, (c_\ell, \mathsf{vk}_\ell))$.
  7. Output (c^*, τ).

Theorem 5.1. *If $\mathcal{E}_{\mathsf{ABE}}$ is an IND-AD-CPA-secure CP-ABE scheme and $\mathcal{E}^{(N)}_{\mathsf{MKFHE}}$ is an IND-CPA-secure multikey FHE scheme, then $\mathcal{E}^{(N)}_{\mathsf{PBHE}}$ is IND-AD-CPA-secure.*

The proof is presented in the extended version (Clear and McGoldrick, 2013).

6 CONCLUSIONS AND FUTURE WORK

We have initiated the study of homomorphic encryption with support for fine-grained access control and composition. Furthermore, we have proposed a syntax for a primitive that captures the problem of homomorphic encryption in this setting. An instantiation of this primitive was presented that makes use of both CP-ABE and multikey FHE, and shown to be semantically secure. Given that there are currently no known fully-homomorphic (or even somewhat-homomorphic) CP-ABE schemes, it seems that our construction is the only way to achieve the same goal, albeit for a bounded number of independent users. In future work, we hope to move beyond the semi-honest model and tackle the problem of verifiability.

REFERENCES

Bethencourt, J., Sahai, A., and Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 321–334, Washington, DC, USA. IEEE Computer Society.

Bonatti, P., De Capitani di Vimercati, S., and Samarati, P. (2002). An algebra for composing access control policies. *ACM Trans. Inf. Syst. Secur.*, 5(1):1–35.

Bruns, G., Dantas, D. S., and Huth, M. (2007). A simple and expressive semantic framework for policy composition in access control. In *Proceedings of the 2007 ACM workshop on Formal methods in security engineering*, FMSE '07, pages 12–21, New York, NY, USA. ACM.

Clear, M., Hughes, A., and Tewari, H. (2013). Homomorphic encryption with access policies: Characterization and new constructions. In Youssef, A., Nitaj, A., and Hassanien, A., editors, *Progress in Cryptology AFRICACRYPT 2013*, volume 7918 of *Lecture Notes in Computer Science*, pages 61–87. Springer Berlin Heidelberg.

Clear, M. and McGoldrick, C. (2013). Policy-Based Non-interactive Outsourcing of Computation using multikey FHE and CP-ABE. Cryptology ePrint Archive. http://eprint.iacr.org/.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing STOC 09*, (September):169.

Gentry, C., Halevi, S., and Vaikuntanathan, V. (2010). A Simple BGN-Type Cryptosystem from LWE. In Gilbert, H., editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 506–522. Springer.

Hoffstein, J., Pipher, J., and Silverman, J. H. (1998). NTRU: a ring-based public key cryptosystem. *Lecture Notes in Computer Science*, 1423:267–288.

López-Alt, A., Tromer, E., and Vaikuntanathan, V. (2012). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 1219–1234, New York, NY, USA. ACM.

Moses, T. et al. (2005). Extensible access control markup language (xacml) version 2.0. *Oasis Standard*, 200502.

Ni, Q., Bertino, E., and Lobo, J. (2009). D-algebra for composing access control policy decisions. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 298–309, New York, NY, USA. ACM.

Pirretti, M., Traynor, P., McDaniel, P., and Waters, B. (2010). Secure attribute-based systems. *Journal of Computer Security*, 18(5):799–837.

Rao, P., Lin, D., Bertino, E., Li, N., and Lobo, J. (2011). Fine-grained integration of access control policies. *Computers & Security*, 30(23):91 – 107. ¡ce:title¿Special Issue on Access Control Methods and Technologies¡/ce:title¿.

Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Catalano, D., Fazio, N., Gennaro, R., and Nicolosi, A., editors, *Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer.

Xiao, L., Bastani, O., and Yen, I.-L. (2012). An efficient homomorphic encryption protocol for multi-user systems. *IACR Cryptology ePrint Archive*, 2012:193.