Tracing Insider Attacks in the Context of Predicate Encryption Schemes

Jonathan Katz and Dominique Schröder
University of Maryland
Email: {jkatz,schroder}@cs.umd.edu

***Abstract*—In a predicate encryption scheme an authority generates master public and secret keys, and uses the master secret key to derive personal secret keys for authorized users. Each user's personal secret key SK_f corresponds to a *predicate* f defining the access rights of that user, and each ciphertext is associated (by the sender) with an *attribute*. The security provided is that a ciphertext associated with attribute I can be decrypted only using a personal secret key SK_f for which $f(I) = 1$, i.e., for which the given access rights f allow decryption of ciphertexts having attribute I. Predicate encryption generalizes identity-based encryption, broadcast encryption, attribute-based encryption, and more, and has been suggested as a mechanism for implementing secure information flow and distributed access control in scenarios involving multiple security domains.**

In this work, we introduce and study the notion of *traceability* for predicate encryption schemes, thus generalizing the analogous notion that has been defined in the specific context of broadcast encryption. Traceability allows a group manager to apprehend malicious insiders who leak their personal secret keys to an adversary, or to determine which authorized users' keys have been compromised. In addition to defining the notion, we show how to add traceability to the most expressive predicate encryption scheme currently known.

I. Introduction

Traditional public-key encryption is coarse-grained: a sender encrypts a message M with respect to some public key PK, and only the (unique) user who holds the secret key SK associated with PK can decrypt and recover the message. In many natural settings, however, it would be preferable to allow the sender to define a *policy* determining who is allowed to recover the encrypted data. For example, classified data might be associated with certain keywords; the data, once encrypted, should be automatically accessible to users who are allowed to read *all* classified information, or to users allowed to read information associated with the particular keywords in question, but to no one else.

Over the past few years, the notion of *predicate encryption* [9], [16] (and a further generalization termed *functional encryption* [7]) has been suggested to provide exactly this sort of fine-grained access to encrypted data. At a high level (formal definitions are given in Section II), in a predicate encryption scheme there is a central authority who generates master public and secret keys, and who publishes the master public key. Authorized users of the system can obtain personal secret keys from the authority. These personal secret keys correspond to predicates (i.e., boolean functions) in some class $\mathcal{F}$, and the secret key SK_f given to an authorized user depends on the predicate $f \in \mathcal{F}$ that determines what type of data this user should have access to. Senders, who need only know the master public key, associate ciphertexts with an attribute in some set $\mathbb{A}$; a ciphertext associated with the attribute $I \in \mathbb{A}$ can be decrypted using a secret key SK_f if and only if $f(I) = 1$.

As a simple example just to illustrate the point, we can imagine that a user is given a secret key SK_f for the predicate

$$f(\mathsf{att}) = 1 \text{ iff } \mathsf{att} \in \{\mathsf{unclassified}, \mathsf{secret}\}.$$

At the time of encryption, a sender associates a ciphertext with an attribute in the set $\mathbb{A} = \{\mathsf{unclassified}, \mathsf{secret}, \mathsf{topsecret}\}$. The user with secret key SK_f as above will only be able to decrypt if the classification level of the ciphertext is unclassified or secret, but will not be able to decrypt if the ciphertext is classified as topsecret.

The "basic" level of security achieved here is that a ciphertext associated with the attribute I does not reveal any information about the underlying plaintext unless one is in possession of a secret key giving the explicit ability to decrypt. That is, if we consider an adversary $\mathcal{A}$ who holds keys $SK_{f_1}, \ldots, SK_{f_\ell}$, then $\mathcal{A}$ learns nothing about the underlying plaintext from a ciphertext associated with attribute I if $f_1(I) = \cdots = f_\ell(I) = 0$. This notion of security notion is called *payload hiding* [16]. A stronger notion of security called *attribute hiding* [16] requires (informally) that ciphertexts should also hide the attribute with which they are associated (in addition to hiding the message). That is, an adversary holding secret keys as above learns only $f_1(I), \ldots, f_\ell(I)$ (and the message, in case one of these evaluates to 1), but nothing else about I. We refer to Section II for formal definitions.

Predicate encryption is a powerful abstraction that generalizes and unifies several pre-existing primitives. Identity-based encryption (IBE) [21], [5] is equivalent to predicate encryption for the class $\mathcal{F}$ of equality predicates; the standard notion of security for IBE corresponds to payload hiding, while *anonymous* IBE [3], [1] corresponds to attribute hiding. Attribute-based encryption schemes [20], [15], [2], [19], and schemes supporting range queries [9], [22] can also be cast in the framework of predicate encryption, as can (public-key) broadcast encryption [13] and forward-secure encryption [10].

The expressivity of a given predicate encryption scheme depends on the attributes $\mathbb{A}$ and class of predicates $\mathcal{F}$ that are supported. Ideally we would like a predicate encryption scheme where $\mathbb{A} = \{0,1\}^*$ and $\mathcal{F}$ is the class of all (polynomial-time computable) predicates, but such a scheme is not yet known (and there are technical reasons to believe that such full expressivity may be impossible to achieve [7]). The most expressive scheme currently available, due to Katz et al. [16], is an attribute-hiding scheme supporting *inner products*; specifically, for some modulus N and parameter ℓ the set of attributes is $\mathbb{A} = \mathbb{Z}_N^\ell$ and the class of predicates is $\mathcal{F} = \{f_{\vec{v}} \mid \vec{v} \in \mathbb{Z}_N^\ell\}$ where $f_{\vec{v}}(\vec{x}) \overset{\text{def}}{=} 1$ iff $\langle \vec{v}, \vec{x} \rangle = 0 \bmod N$. (Here, $\langle \cdot, \cdot \rangle$ denotes the standard inner product.) It is shown in [16] that this scheme implies anonymous IBE, as well as predicate classes including polynomials, CNF/DNF formulae, threshold functions, and more.

A. Tracing Insider Attacks

Inspired by work adding traceability to broadcast encryption schemes [11], [4], [12], [8], we consider the analogous problem in the more general context of predicate encryption. (Recall that predicate encryption can be viewed as encompassing broadcast encryption as a special case.) The basic idea is as follows. Say an adversary $\mathcal{A}$ obtains the secret keys of some set S of authorized users. For our purposes, it does not matter whether the adversary obtains these keys by compromising a user's device, or through outright collusion with the user(s) in question (or any combination of both). At some later point in time, the central authority may observe that some unauthorized user (i.e., the adversary) is able to decrypt some ciphertext(s) that it should not be able to. We provide the authority with the ability to *trace* at least one user u in the set S. We leave the decision of what to do after this tracing step out of scope of the present work; at a minimum, however, the authority would then want to *revoke* u's privileges [14], [18], [17]. (We leave the exact mechanism for doing so to future work.)

We introduce a new definition of security corresponding to the above informal desideratum. To make this definition meaningful we first modify the standard definition of predicate encryption so as to incorporate both a predicate f and a user-id id into each personal secret key; this makes sense in any system where there may be multiple authorized users $id_1, id_2, \ldots$ who all have access rights defined by the same predicate f. We then consider two versions of traceability, a "weak" version that we argue is insufficient in practice, and a "strong" version that strengthens the informal description in the previous paragraph. Finally, we show how to integrate traceability (of either the "weak" or "strong" type) into the most expressive predicate encryption scheme to date [16]. Actually, our constructions have the advantage of being completely *generic*, in that they can be applied to *any* inner-product predicate encryption scheme. Our work implies traceable versions of any of the predicate encryption schemes implied by inner-product schemes [16], e.g., equality predicates, threshold functions, and more.

II. Definitions

We begin by reviewing definitions for predicate encryption that are essentially from prior work (with the exception that our definition of predicate encryption now explicitly incorporate identities). Then we introduce our new definitions of traceability for predicate encryption schemes.

A. Predicate Encryption Schemes

We begin with a functional definition of predicate encryption. The definition follows [16], except that we incorporate identities into the (personal) secret keys. These identities are not utilized in normal usage of the scheme (and, in particular, senders do not need to know the identities of any users in order to encrypt), but are used during tracing as described further below. (In particular, we recover the definition in [16] if the identity is always set to 0^n when running GenKey.) The addition of identities is natural in any setting where there may be multiple authorized users $id_1, id_2, \ldots$ all of whom have identical access rights defined by the same predicate f.

Definition 1 *A* predicate encryption scheme *for the class of predicates $\mathcal{F}$ over the set of attributes $\mathbb{A}$ consists of four* PPT *algorithms* $\mathsf{Setup}, \mathsf{GenKey}, \mathsf{Enc}, \mathsf{Dec}$ *such that:*

- Setup *takes as input the security parameter 1^n and outputs a (master) public key PK and a (master) secret key SK.*
- GenKey *takes as input the master secret key SK, an identity $id \in \{0,1\}^n$, and a (description of a) predicate $f \in \mathcal{F}$. It outputs a key $SK_{id,f}$. We denote this as $SK_{id,f} \leftarrow \mathsf{GenKey}_{SK}(id, f)$.*
- Enc *takes as input the public key PK, an attribute $I \in \mathbb{A}$, and a message M in some associated message space. It returns a ciphertext C. We write this as $C \leftarrow \mathsf{Enc}_{PK}(I, M)$.*
- Dec *takes as input a secret key $SK_{id,f}$ and a ciphertext C. It outputs either a message M or the distinguished symbol $\perp$.*

For correctness, we require that for all n, all (PK, SK) generated by $\mathsf{Setup}(1^n)$, all $id \in \{0,1\}^n$, all $f \in \mathcal{F}$, any key $SK_{id,f} \leftarrow \mathsf{GenKey}_{SK}(id, f)$, and all $I \in \mathbb{A}$:

- *If $f(I) = 1$ then $\mathsf{Dec}_{SK_{id,f}}(\mathsf{Enc}_{PK}(I, M)) = M$.*
- *If $f(I) = 0$ then $\mathsf{Dec}_{SK_{id,f}}(\mathsf{Enc}_{PK}(I, M)) = \perp$ with all but negligible probability.*

To recover prior definitions of predicate encryption, we say that a predicate encryption scheme *does not support identities* if id is always set to 0^n when running GenKey. In that case we can simply omit the first input to GenKey.

In [16], two definitions of security for predicate encryption are given. The first, *payload hiding,* ensures secrecy of the encrypted message; the second, *attribute hiding,* ensures secrecy of the associated attribute as well. We require the stronger notion of attribute hiding for our work, and our construction achieves this stronger notion, so we only present that definition here. Our definition uses the "selective" notion of security introduced by [10]; this is not essential for our results, but is the notion achieved by the construction in [16].

Definition 2 *A predicate encryption scheme with respect to $\mathcal{F}$ and $\mathbb{A}$ is* attribute hiding *if for all* PPT *adversaries $\mathcal{A}$, the* *advantage of $\mathcal{A}$ in the following experiment is negligible in the security parameter n:*

1) *$\mathcal{A}(1^n)$ outputs $I_0, I_1 \in \mathbb{A}$.*
2) *$\mathsf{Setup}(1^n)$ is run to generate PK and SK, and the adversary is given PK.*
3) *$\mathcal{A}$ may adaptively request keys for any identity/predicate pairs $(id_1, f_1), \ldots, (id_\ell, f_\ell) \in \{0,1\}^n \times \mathcal{F}$, subject to the restriction that $f_i(I_0) = f_i(I_1)$ for all i. In response to each such query, $\mathcal{A}$ is given the corresponding key $SK_{id_i, f_i} \leftarrow \mathsf{GenKey}_{SK}(id_i, f_i)$.*
4) *$\mathcal{A}$ outputs two equal-length messages M_0, M_1. If there is an i for which $f_i(I_0) = f_i(I_1) = 1$, then it is required that $M_0 = M_1$. A random bit b is chosen, and $\mathcal{A}$ is given the ciphertext $C \leftarrow \mathsf{Enc}_{PK}(I_b, M_b)$.*
5) *The adversary may continue to request keys for additional predicates, subject to the same restrictions as before.*
6) *$\mathcal{A}$ outputs a bit b', and succeeds if $b' = b$.*

The advantage of $\mathcal{A}$ is the absolute value of the difference between its success probability and $1/2$.

Note that the definition captures the informal notion of attribute hiding described in the Introduction: given some ciphertext $C \leftarrow \mathsf{Enc}_{PK}(I, M)$, the adversary learns nothing about I other than $f_1(I), \ldots, f_\ell(I)$; moreover, the adversary learns nothing about M unless the adversary has in its possession a secret key SK_{id_i, f_i} for which $f_i(I) = 1$.

B. Traceability

Here we define new notions of traceability for predicate encryption schemes. Our definitions are loosely based on the definitions of [6], which were given for the specific case of broadcast encryption. Below, we define an algorithm Trace that, intuitively, takes as input the master secret key SK, an attribute I, and a "decryption box" $\mathcal{D}$ that decrypts ciphertexts associated with the attribute I with high probability; Trace is supposed to output the identity of some user whose secret key was used to construct $\mathcal{D}$. We say that an algorithm Trace is efficient if the following holds. Fix some security parameter n and master public key PK, and define

$$\mathsf{Succ}(I, \mathcal{D}) \stackrel{\text{def}}{=} \Pr[\mathcal{D}(\mathsf{Enc}_{PK}(I, M)) = M],$$

where the probability is taken over random choice of the message M (and random coins of $\mathcal{D}$ in case it is randomized). Then we require that there is a polynomial p such that for all $I, \mathcal{D}$ with $\epsilon \stackrel{\text{def}}{=} \mathsf{Succ}(I, \mathcal{D}) > 0$, the expected running time of $\mathsf{Trace}_{SK}(I, \mathcal{D})$ is bounded by $\epsilon^{-1} \cdot p(n)$. This ensures that expected running time of the following experiment is bounded by a fixed polynomial $p(n)$: (1) first run $\mathcal{D}(\mathsf{Enc}_{PK}(I, M))$ (for random M) and check whether decryption succeeds; if it does, then (2) run $\mathsf{Trace}_{SK}(I, \mathcal{D})$.

Definition 3 *A predicate encryption scheme with respect to $\mathcal{F}$ and $\mathbb{A}$ satisfies* weak traceability *if there exists an efficient algorithm* Trace *such that for all* PPT *adversaries $\mathcal{A}$, the probability that $\mathcal{A}$ succeeds in the following experiment is negligible in the security parameter n:*

1) *$\mathsf{Setup}(1^n)$ is run to generate PK and SK, and the adversary is given PK.*
2) *$\mathcal{A}$ may adaptively request keys for any identity/predicate pairs $(id_1, f_1), \ldots, (id_\ell, f_\ell) \in \{0,1\}^n \times \mathcal{F}$. In response, the adversary $\mathcal{A}$ is given the corresponding keys $SK_{id_i, f_i} \leftarrow \mathsf{GenKey}_{SK}(id_i, f_i)$.*
3) *The adversary outputs some attribute $I \in \mathbb{A}$ along with a "decryption box" $\mathcal{D}$ (specified as a boolean circuit).*
4) *Choose random message M. If $\mathcal{D}(\mathsf{Enc}_{PK}(I, M)) \neq M$ then set $\mathsf{succ} = 0$. Otherwise, set $\mathsf{succ} = 1$ and run $id \leftarrow \mathsf{Trace}_{SK}(I, \mathcal{D})$.*

$\mathcal{A}$ succeeds *if $\mathsf{succ} = 1$ and $id \notin \{id_1, \ldots, id_\ell\}$.*

We argue that the above, although it is the "natural" extension of the definition of traceability from the setting of broadcast encryption, is not suitable in practice. Consider an adversary $\mathcal{A}$ who corrupts user id_1 associated with predicate f_1, and user id_2 associated with predicate f_2, and assume that f_1 is a "low security" user (say, f_1 only allows decryption of unclassified documents) whereas f_2 is a "high security" user (e.g., f_2 allows decryption of unclassified, secret, or top secret documents). The above definition of weak traceability would consider a scheme to be secure even if $\mathcal{A}$ could output a decryption box that decrypts top secret documents with probability 1, but Trace only outputs the identity id_1 of the "low security" user. This is unsatisfying, as the authority knows that $\mathcal{A}$ must have obtained the secret key of *some* user who was authorized to decrypt top secret documents (this follows from the security guaranteed by Definition 2), yet the authority was only able to trace a "low security" user.

With the above in mind, we also define the following stronger notion of traceability:

Definition 4 *A predicate encryption scheme with respect to $\mathcal{F}$ and $\mathbb{A}$ satisfies* strong traceability *if there exists an efficient algorithm* Trace *such that for all* PPT *adversaries $\mathcal{A}$, the probability that $\mathcal{A}$ succeeds in the following experiment is negligible in the security parameter n:*

1) *$\mathsf{Setup}(1^n)$ is run to generate PK and SK, and the adversary is given PK.*
2) *$\mathcal{A}$ may adaptively request keys for any identity/predicate pairs $(id_1, f_1), \ldots, (id_\ell, f_\ell) \in \{0,1\}^n \times \mathcal{F}$. In response, the adversary $\mathcal{A}$ is given the corresponding keys $SK_{id_i, f_i} \leftarrow \mathsf{GenKey}_{SK}(id_i, f_i)$.*
3) *The adversary outputs some attribute $I \in \mathbb{A}$ along with a "decryption box" $\mathcal{D}$ (specified as a boolean circuit).*
4) *Choose random message M. If $\mathcal{D}(\mathsf{Enc}_{PK}(I, M)) \neq M$ then set $\mathsf{succ} = 0$. Otherwise, set $\mathsf{succ} = 1$ and run $id \leftarrow \mathsf{Trace}_{SK}(I, \mathcal{D})$.*

Set $S_I = \{id_i \mid f_i(I) = 1\}$, i..e, this is the set of identities of the users corrupted by $\mathcal{A}$ whose keys enable decryption of ciphertexts associated with the attribute I. Then $\mathcal{A}$ succeeds *if $\mathsf{succ} = 1$ and $id \notin S_I$.*

A scheme satisfying strong traceability ensures that the authority traces the identity of a user whose secret key enables decryption of ciphertexts associated with the attribute I for which the given decryption box works. It thus matches our intuitive notion of traceability more closely.

III. Weak Traceability for Predicate Encryption Schemes

Here we show that weak traceability can be easily integrated into any predicate encryption scheme by combining it with any broadcast encryption scheme. We view the simplicity of this construction, and the fact that it is obtained by independently using predicate encryption and broadcast encryption without tightly integrating the two, as a further argument against the weak notion of traceability.

Fix some set U of authorized users. Broadcast encryption is simply a predicate encryption scheme where the set of attributes $\mathbb{A}_{\mathsf{bc}}$ consists of all subsets of U, and the set of predicates is given by $\mathcal{F}_{\mathsf{bc}} = \{f_{id}\}_{id \in U}$ where

$$f_{id}(S) = 1 \text{ iff } id \in S.$$

That is, the sender chooses a set S of users authorized to read the given content, and only users in the set S can decrypt the resulting ciphertext. In broadcast encryption, a user's identity uniquely defines their predicate and thus we do not include a second id in a user's personal secret key. (Alternately, we could simply restrict the system to only ever generate keys of the form $SK_{id,f_{id}}$.) That is, standard broadcast encryption corresponds to a predicate encryption scheme for $\mathcal{F}_{\mathsf{bc}}, \mathbb{A}_{\mathsf{bc}}$ that does not support identities. Given this, our notions of weak and strong traceability collapse, and both become equivalent to the standard notion of traceability considered in the context of broadcast encryption [6].

Let $\mathcal{P} = (\mathsf{Setup}, \mathsf{GenKey}, \mathsf{Enc}, \mathsf{Dec})$ be a predicate encryption scheme for $\mathcal{F}, \mathbb{A}$ that does not support identities, and let $\mathcal{BC} = (\mathsf{Setup}^{\mathsf{bc}}, \mathsf{GenKey}^{\mathsf{bc}}, \mathsf{Enc}^{\mathsf{bc}}, \mathsf{Dec}^{\mathsf{bc}})$ be a broadcast encryption scheme. Let U denote the universe of possible user identities. Construct predicate encryption scheme $\mathcal{P}' = (\mathsf{Setup}', \mathsf{GenKey}', \mathsf{Enc}', \mathsf{Dec}')$ for $\mathcal{F}, \mathbb{A}$ as follows:

1) $\mathsf{Setup}'(1^n)$ runs $(SK, PK) \leftarrow \mathsf{Setup}(1^n)$ and $(SK^{\mathsf{bc}}, PK^{\mathsf{bc}}) \leftarrow \mathsf{Setup}^{\mathsf{bc}}(1^n)$. The master secret key is $SK' = (SK, SK^{\mathsf{bc}})$, and the master public key is $PK' = (PK, PK^{\mathsf{bc}})$.
2) $\mathsf{GenKey}'_{SK'}(id, f)$, for $f \in \mathcal{F}$, first computes the key $SK_f \leftarrow \mathsf{GenKey}_{SK}(f)$ and then computes the key $SK^{\mathsf{bc}}_{id} \leftarrow \mathsf{GenKey}^{\mathsf{bc}}_{SK^{\mathsf{bc}}}(id)$. It outputs the personal secret key $SK'_{id,f} = (SK_f, SK^{\mathsf{bc}}_{id})$.
3) $\mathsf{Enc}'_{PK'}(I, M)$ chooses random r and then computes $C \leftarrow \mathsf{Enc}_{PK}(I, r)$ and $C^{\mathsf{bc}} \leftarrow \mathsf{Enc}^{\mathsf{bc}}_{PK^{\mathsf{bc}}}(U, r \oplus M)$. It outputs the ciphertext $C' = (C, C^{\mathsf{bc}})$.
4) $\mathsf{Dec}'_{SK'_{id,f}}(C')$ parses $SK'_{id,f}$ as $(SK_f, SK^{\mathsf{bc}}_{id})$, and parses ciphertext C' as (C, C^{bc}). It then computes $r \leftarrow \mathsf{Dec}_{SK_f}(C)$ and $r' \leftarrow \mathsf{Dec}^{\mathsf{bc}}_{SK_{id}}(C^{\mathsf{bc}})$. It outputs the message $r \oplus r'$.

The above construction just runs the original predicate encryption scheme in parallel with the broadcast encryption scheme. To encrypt a message M with attribute I, the sender first "secret shares" the message M as $(r, r \oplus M)$; encrypts the first share r using the original predicate encryption scheme and attribute I; and encrypts the second share $r \oplus M$ using the broadcast encryption scheme and the entire identity space U (so that every user in the system is allowed to decrypt this second share). One can easily verify that correctness holds.

The tracing algorithm is the obvious one. Given a decryption box $\mathcal{D}$ and some attribute I (such that the decryption box succeeds in decrypting ciphertexts associated with the attribute I with high probability), we simply run the tracing algorithm of the underlying broadcast encryption scheme on the second component C^{bc} of a well-formed ciphertext. We omit further details, which are straightforward.

Theorem 1: If $\mathcal{P}$ is a payload-hiding (resp., attribute-hiding) predicate encryption scheme for $\mathcal{F}, \mathbb{A}$, and $\mathcal{BC}$ is a secure broadcast encryption scheme, then $\mathcal{P}'$ is a payload-hiding (resp., attribute-hiding) predicate encryption scheme for $\mathcal{F}, \mathbb{A}$ that satisfies weak traceability.

Proof (Sketch): We consider the case of attribute-hiding, and thus need to show that $\mathcal{P}'$ satisfies both Definitions 2 and 3. That $\mathcal{P}'$ satisfies Definition 2 follows from the fact that $\mathcal{P}$ is attribute hiding. In a bit more detail, consider a ciphertext $C' = (C, C^{\mathsf{bc}})$ of the derived scheme $\mathcal{P}'$. The second component C^{bc} contains no information about either the message M or the attribute I that was used during encryption. (It does have information on $r \oplus M$, but since r was chosen at random this alone does not reveal anything about M.) By the assumption that $\mathcal{P}$ is attribute hiding, we have that the first component C reveals nothing to an adversary (in a computational sense) about I or r (assuming $\mathcal{A}$ did not explicitly request a secret key revealing such information). It follows that $\mathcal{P}'$ is attribute hiding.

Weak traceability follows from traceability of $\mathcal{BC}$. Intuitively, in order to decrypt ciphertexts of the form (C, C^{bc}), the decryption box must in particular be able to decrypt the second component C^{bc} of such ciphertexts with "high" probability. But then the tracing property of $\mathcal{BC}$ implies that it is possible to trace the identity of at least one user whose key has been compromised by the adversary. This is all that is required in order to satisfy the definition of weak traceability. ■

IV. Strong Traceability for Inner-Product Encryption Schemes

In this section we show how to obtain strong traceability for any inner-product encryption scheme. The result here is incomparable to what is achieved in the previous section. On the one hand, we are obtaining the stronger notion of traceability here. On the other hand, our result in this section is specific to inner-product encryption and does not extend to arbitrary predicate encryption schemes. We remark further that here we require the initial inner-product scheme to satisfy attribute-hiding; payload-hiding is not sufficient.

As described in the Introduction, an inner-product encryption scheme is a predicate encryption scheme where the class of attributes is $\mathbb{A} = \mathbb{Z}^\ell$ for arbitrary ℓ, and the class of predicates is $\mathcal{F} = \{f_{\vec{v}} \mid \vec{v} \in \mathbb{Z}^\ell\}$ where $f_{\vec{v}}(\vec{x}) \overset{\text{def}}{=} 1$ iff

$\langle \vec{v}\,, \vec{x}\, \rangle = 0$.[1] (Here, $\langle \cdot, \cdot \rangle$ denotes the standard inner product.)

Let $\mathcal{P} = (\mathsf{Setup}, \mathsf{GenKey}, \mathsf{Enc}, \mathsf{Dec})$ be an inner-product encryption scheme that does not support identities. We construct an inner-product encryption scheme $\mathcal{P}' = (\mathsf{Setup}', \mathsf{GenKey}', \mathsf{Enc}', \mathsf{Dec}')$ for vectors of length ℓ, where there are n users in the system. We label users in unary, so user $i \in \{1, \ldots, n\}$ is associated with identity $id_i \overset{\text{def}}{=} 0^{i-1}10^{n-i}$. In the following, set $\ell' = \ell + n$:

- $\mathsf{Setup}'(1^n)$ runs $\mathsf{Setup}(1^n)$ to obtain master secret and public keys (SK, PK).
- Let $id_i \in \{0,1\}^n$ be an identity, and let $\vec{v} \in \mathbb{Z}^\ell$ be a vector. Then $\mathsf{GenKey}'_{SK}(id_i, \vec{v}\,)$ does the following. Let $\vec{v}\,' \in \mathbb{Z}^{\ell'}$ be the vector obtained by appending id_i to $\vec{v}$ (where we view the identity id_i as a vector in $\mathbb{Z}^n$). Output $SK'_{id_i,\vec{v}} \leftarrow \mathsf{GenKey}_{SK}(\vec{v}\,')$.
- $\mathsf{Enc}'_{PK}(\vec{x}\,, M)$ does as follows. Let $\vec{x}\,' \in \mathbb{Z}^{\ell'}$ be the vector obtained by appending 0^n to $\vec{x}$. Output $\mathsf{Enc}_{PK}(\vec{x}\,', M)$.
- $\mathsf{Dec}'_{SK'_{id_i,\vec{v}}}(C)$ simply outputs $\mathsf{Dec}_{SK'_{id_i,\vec{v}}}(C)$.

That is, we "embed" an inner-product computation over $\mathbb{Z}^\ell$ into an inner-product computation over $\mathbb{Z}^{\ell'}$. To see that correctness holds, note that the secret key $SK'_{id_i,\vec{v}}$ (in $\mathcal{P}'$) corresponds to the secret key $SK_{\vec{v}\,'}$ (in $\mathcal{P}$), where

$$\vec{v}\,' = (v_1, \ldots, v_\ell, \underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0),$$

and $\vec{v} = (v_1, \ldots, v_\ell)$. Encryption with respect to the vector $\vec{x}$ (in $\mathcal{P}'$) corresponds to encryption with respect to $\vec{x}\,'$ (in $\mathcal{P}$), where

$$\vec{x}\,' = (x_1, \ldots, x_\ell, 0, \ldots, 0).$$

Thus, $\langle \vec{v}\,, \vec{x}\, \rangle = 0$ iff $\langle \vec{v}\,', \vec{x}\,' \rangle = 0$, regardless of the specific identity id_i of the user.

To trace, we have the authority use "ill-formed" ciphertexts that are encrypted with respect to vectors $\vec{x}\,'$ whose last n bits are not identically 0. We describe the procedure informally, and leave the formal details for the full version of this work. Assume for simplicity that an adversary $\mathcal{A}$ outputs a decryption box $\mathcal{D}$ that *always* succeeds in decrypting ciphertexts that are encrypted (in $\mathcal{P}'$) with respect to some vector $\vec{x} = (x_1, \ldots, x_\ell)$. Let $S_{\vec{x}}$ denote the set of identities of those users who were corrupted by $\mathcal{A}$ and are associated with a vector $\vec{v}$ such that $\langle \vec{v}\,, \vec{x}\, \rangle = 0$. We show how one of those users can be traced.

First, the authority uses $\mathcal{D}$ to try to decrypt ciphertexts of the form $\mathsf{Enc}_{PK}(\vec{x}\,'_1, M)$, where

$$\vec{x}\,'_1 \overset{\text{def}}{=} (x_1, \ldots, x_\ell, r_1, \underbrace{0, \ldots, 0}_{n-1})$$

for a random r_1. If decryption fails, then this implies that $\mathcal{A}$ has obtained a key $SK'_{id,\vec{v}}$ where $\langle \vec{x}\,, \vec{v}\, \rangle = 0$ and $id[1]$ (that is, the first bit of id) is equal to 1. (This, in turn, means $id_1 \in S_{\vec{x}}$ and so tracing is complete.) To see this, note that if $\mathcal{A}$ had *not* obtained such a key then (with overwhelming probability over choice of r_1) $\mathcal{A}$ would not be able to distinguish encryptions (in $\mathcal{P}$) with respect to $\vec{x}\,'$ and encryptions with respect to $\vec{x}\,'_1$ (and so decryption would then have to succeed for encryptions with respect to the latter). Note that we rely here on the fact that $\mathcal{P}$ is attribute hiding.

If, on the other hand, decryption succeeds (above), then this implies that $\mathcal{A}$ has obtained a key $SK'_{id,\vec{v}}$ where $\langle \vec{x}\,, \vec{v}\, \rangle = 0$ and $id[1] = 0$; if not, then with overwhelming probability the adversary has no keys of the form $SK_{\vec{v}\,'}$ with $\langle \vec{x}\,'_1, \vec{v}\,' \rangle = 0$ (and so, by payload-hiding of $\mathcal{P}$, would be unable to decrypt successfully). We note that it is, of course, entirely possible that the adversary has obtained two keys $SK'_{id_1,\vec{v}_1}$ and $SK'_{id_2,\vec{v}_2}$ for which $\langle \vec{x}\,, \vec{v}_1 \rangle = \langle \vec{x}\,, \vec{v}_2 \rangle = 0$ but $id_1[1] = 0$ and $id_2[1] = 1$. In any event, the authority at this point has insufficient information to definitively trace a corrupted user, and the authority proceeds as described below.

The authority continues in a series of at most n stages, where in the ith stage the authority uses $\mathcal{D}$ to try to decrypt ciphertexts of the form $\mathsf{Enc}_{PK}(\vec{x}\,'_i, M)$, where

$$\vec{x}\,'_i \overset{\text{def}}{=} (x_1, \ldots, x_\ell, r_1, \ldots, r_i, \underbrace{0, \ldots, 0}_{n-i})$$

for random $r_1, \ldots, r_i$. The authority does this until the first stage i for which $\mathcal{D}$ fails to decrypt, and at that point the authority outputs identity id_i and terminates. That this succeeds in tracing a corrupted user $id_i \in S_{\vec{x}}$ follows from the next two claims.

Claim 1: With overwhelming probability there exists some i, with $1 \le i \le n$, for which decryption in stage i fails.

Proof: Consider decryption in stage n. The authority uses $\mathcal{D}$ to try to decrypt ciphertexts of the form $\mathsf{Enc}_{PK}(\vec{x}\,'_n, M)$, where

$$\vec{x}\,'_i \overset{\text{def}}{=} (x_1, \ldots, x_\ell, r_1, \ldots, r_n)$$

for random $r_1, \ldots, r_n$. But with overwhelming probability (over choice of $r_1, \ldots, r_n$), the adversary $\mathcal{A}$ holds *no* keys $SK_{\vec{v}\,'}$ for which $\langle \vec{x}\,'_n, \vec{v}\,' \rangle = 0$. Thus decryption will fail by stage n at the latest. ■

Claim 2: Say decryption fails in stage i but succeeded in stage $i-1$. (If $i = 1$ then the latter just means that decryption of ciphertexts of the form $\mathsf{Enc}_{PK}(\vec{x}\,', M)$ succeeded, which is true by assumption.) Then with overwhelming probability it holds that $id_i \in S_{\vec{x}}$.

Proof: Since the behavior of $\mathcal{D}$ changes in going from phase $i-1$ to phase i, this implies that $\mathcal{A}$ must have obtained some key $SK_{\vec{v}\,'}$ for which exactly one of $\langle \vec{v}\,', \vec{x}\,'_{i-1} \rangle$ or $\langle \vec{v}\,', \vec{x}\,'_i \rangle$ is 0. With overwhelming probability over choice of r_i, it could not be the case that $\langle \vec{v}\,', \vec{x}\,'_{i-1} \rangle \neq 0$ but $\langle \vec{v}\,', \vec{x}\,'_i \rangle = 0$; thus, it must be that $\langle \vec{v}\,', \vec{x}\,'_{i-1} \rangle = 0$ and $\langle \vec{v}\,', \vec{x}\,'_i \rangle \neq 0$. For the type of secret keys generated by $\mathcal{P}'$, however, this can only occur if $\vec{v}\,'$ has its $(\ell + i)$th component equal to 1, which implies that $\vec{v}\,'$ corresponds to identity id_i. Moreover, since $\langle \vec{v}\,', \vec{x}\,'_{i-1} \rangle = 0$ (and components $\ell + 1$

[1] Actually, for the inner-product scheme constructed in [16] all vectors lie in $\mathbb{Z}_N^\ell$ and the inner product is taken modulo N, where N is a modulus defined by the master public key. For simplicity we treat all vectors as lying in $\mathbb{Z}^\ell$, but our results extend easily to the other case.

through $\ell + n$ of $\vec{v}\,'$, with the exception of component $\ell + i$, must be equal to 0), thus means that $\vec{v}\,'$ corresponds to some $\vec{v} \in \mathbb{Z}^\ell$ with $\langle \vec{v}, \vec{x} \rangle = 0$; hence $id_i \in S_{\vec{x}}$ as desired. ■

In summary, we have:

Theorem 2: If $\mathcal{P}$ is an attribute-hiding inner-product encryption scheme, then $\mathcal{P}'$ is an attribute-hiding inner-product encryption scheme that satisfies strong traceability.

A full proof of the above will appear in the full version of this work.

V. Conclusions and Future Work

In this work we have introduced the notion of traceability in the context of predicate encryption schemes. This strictly generalizes prior definitions of traceability that were introduced in the specific context of broadcast encryption. We also showed two constructions of predicate encryption schemes with traceability: a generic construction for any predicate encryption scheme that achieves a weak notion of traceability, and a construction specific to the case of inner-product encryption that achieves a strong notion of traceability.

Going forward, there are a few interesting avenues of exploration:

1) First, we would like to find a (generic) way of adding strong traceability to an arbitrary predicate encryption scheme. Even though an inner-product scheme implies several interesting notions of predicate encryption (see [16] for details) it would still be useful to have direct constructions that are more efficient.
2) Second, we can hope for a more efficient construction even for the specific application to inner-product encryption. The construction given here adds overhead linear in n (the number of users) to the original scheme; however, we expect that we can use techniques of [8] to reduce this to $\sqrt{n}$. Alternately, we can explore the case where traceability is only required so long as $\mathcal{A}$ obtains fewer than t personal secret keys (for some fixed parameter t), a weaker notion than the full collusion resistance studied in this work.
3) Perhaps most interesting is the challenge of additionally dealing with *revocation*. Namely, once a compromised secret key is identified, can the authority easily and efficiently revoke the permissions associated with that secret key?

Acknowledgments

The first author would like to thank David Pointcheval for suggesting this problem, and Michael Gerbush and Brent Waters for initial discussions on the topic.

This research was sponsored in part by the US Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defence or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *Journal of Cryptology*, 21(3):350–391, 2008.

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security & Privacy*, pages 321–334. IEEE, 2007.

[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology — Eurocrypt 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, 2004.

[4] D. Boneh and M. K. Franklin. An efficient public key traitor tracing scheme. In *Advances in Cryptology — Crypto '99*, volume 1666 of *LNCS*, pages 338–353. Springer, 1999.

[5] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[6] D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *Advances in Cryptology — Eurocrypt 2006*, volume 4004 of *LNCS*, pages 573–592. Springer, 2006.

[7] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *8th Theory of Cryptography Conference — TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, 2011.

[8] D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *ACM CCS '06: 13th ACM Conf. on Computer and Communications Security*, pages 211–220. ACM Press, 2006.

[9] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *4th Theory of Cryptography Conference — TCC 2007*, volume 4392 of *LNCS*, pages 535–554. Springer, 2007.

[10] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, 2007.

[11] B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Advances in Cryptology — Crypto '94*, volume 839 of *LNCS*, pages 257–270. Springer, 1994.

[12] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Information Theory*, 46(3):893–910, 2000.

[13] A. Fiat and M. Naor. Broadcast encryption. In *Advances in Cryptology — Crypto '93*, volume 773 of *LNCS*, pages 480–491. Springer, 1994.

[14] E. Gafni, J. Staddon, and Y. L. Yin. Efficient methods for integrating traceability and broadcast encryption. In *Advances in Cryptology — Crypto '99*, volume 1666 of *LNCS*, pages 372–387. Springer, 1999.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS '06: 13th ACM Conf. on Computer and Communications Security*, pages 89–98. ACM Press, 2006.

[16] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Advances in Cryptology — Eurocrypt 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, 2008.

[17] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Advances in Cryptology — Crypto 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, 2001.

[18] M. Naor and B. Pinkas. Efficient trace and revoke schemes. In *Financial Cryptography and Data Security 2000*, volume 1962 of *LNCS*, pages 1–20. Springer, 2000.

[19] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *14th ACM Conf. on Computer and Communications Security (CCS)*, pages 195–203. ACM Press, 2007.

[20] A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, 2005.

[21] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology — Crypto '84*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.

[22] E. Shi, J. Bethencourt, H. T.-H. Chan, D. X. Song, and A. Perrig. Multi-dimensional range queries over encrypted data. In *IEEE Symposium on Security & Privacy*, pages 350–364. IEEE, 2007.