On Black-Box Constructions of Predicate Encryption from Trapdoor Permutations

Jonathan Katz* and Arkady Yerukhimovich

Department of Computer Science, University of Maryland
{jkatz,arkady}@cs.umd.edu

Abstract. *Predicate encryption* is a recent generalization of identity-based encryption (IBE), broadcast encryption, attribute-based encryption, and more. A natural question is whether there exist *black-box* constructions of predicate encryption based on generic building blocks, e.g., trapdoor permutations. Boneh et al. (FOCS 2008) recently gave a negative answer for the specific case of IBE.

We show both negative and positive results. First, we identify a combinatorial property on the sets of predicates/attributes and show that, for any sets having this property, no black-box construction of predicate encryption from trapdoor permutations (or even CCA-secure encryption) is possible. Our framework implies the result of Boneh et al. as a special case, and also rules out, e.g., black-box constructions of forward-secure encryption and broadcast encryption (with many excluded users). On the positive side, we identify conditions under which predicate encryption schemes *can* be constructed based on any CPA-secure (standard) encryption scheme.

1 Introduction

In a *predicate encryption* scheme [6, 13] an authority generates a master public key and a master secret key, and uses the master secret key to derive personal secret keys for individual users. A personal secret key corresponds to a predicate in some class $\mathcal{F}$, and ciphertexts are associated (by the sender) with an attribute in some set $\mathbb{A}$; a ciphertext associated with the attribute $I \in \mathbb{A}$ can be decrypted by a secret key SK_f corresponding to the predicate $f \in \mathcal{F}$ if and only if $f(I) = 1$. The basic security guarantee provided by such schemes is that a ciphertext associated with an attribute I hides all information about the underlying message unless one has a personal secret key giving the explicit ability to decrypt; in other words, if an adversary $\mathcal{A}$ holds keys $SK_{f_1}, \ldots, SK_{f_\ell}$ for which $f_1(I) = \cdots = f_\ell(I) = 0$, then $\mathcal{A}$ should learn nothing about the message. (A formal definition is given later.)

By choosing $\mathcal{F}$ and $\mathbb{A}$ appropriately, predicate encryption yields as special cases many notions that are interesting in their own right. For example, by taking

* Work done while visiting IBM. Research supported by DARPA, and by the US Army Research Laboratory and the UK Ministry of Defence under agreement number W911NF-06-3-0001.

$\mathbb{A} = \{0,1\}^n$ and letting $\mathcal{F} = \{f_{ID}\}_{ID \in \{0,1\}^n}$ be the class of point functions (so that $f_{ID}(ID') = 1$ iff $ID = ID'$) we recover the notion of identity-based encryption (IBE) [19, 4]. Similarly, it can be observed that predicate encryption encompasses fuzzy IBE [18], forward-secure (public-key) encryption [7], (public-key) broadcast encryption [9], attribute-based encryption [11, 2, 15], and more as special cases.

Most (though not all) existing constructions of predicate encryption schemes rely on bilinear maps. A natural question is: *what are the minimal assumptions on which predicate encryption can be based?* Of course, the answer will depend on the specific predicate class $\mathcal{F}$ and attribute set $\mathbb{A}$ of interest; in particular, Boneh and Waters [6] show that if $\mathcal{F}$ is polynomial size then (for any $\mathbb{A}$) one can construct a predicate encryption scheme for $(\mathcal{F}, \mathbb{A})$ from any (standard) public-key encryption scheme. On the other hand, Boneh et al. [5] have recently shown that there is no *black-box* construction of IBE from trapdoor permutations.

1.1 Our Results

The specific question we consider is: *for which* $(\mathcal{F}, \mathbb{A})$ *can we construct a predicate encryption scheme over* $(\mathcal{F}, \mathbb{A})$ *based on CPA-secure encryption?* We show both negative and positive results. Before describing these results in more detail, we provide some background intuition.

A natural combinatorial construction of a predicate encryption scheme over some $(\mathcal{F}, \mathbb{A})$ from a CPA-secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is as follows: The authority includes several public keys $pk_1, \ldots, pk_q$ in the master public key, and each personal secret key is some subset of the corresponding secret keys $sk_1, \ldots, sk_q$. Encryption of a message m with respect to an attribute I requires "sharing" m in some way to yield $m_1, \ldots, m_q$, and the resulting ciphertext is $\mathsf{Enc}_{pk_1}(m_1), \ldots, \mathsf{Enc}_{pk_q}(m_q)$. Intuitively, this works if:

Correctness: Let $SK_f = \{sk_{i_1}, \ldots, sk_{i_t}\}$ be a personal secret key for which $f(I) = 1$. Then the "shares" $m_{i_1}, \ldots, m_{i_t}$ should enable recovery of m.

Security: Let $\{sk_{i_1}, \ldots, sk_{i_k}\} = \bigcup_{f \in \mathcal{F}: f(I)=0} SK_f$. Then the set of "shares" $m_{i_1}, \ldots, m_{i_k}$ should leak no information about m.[1]

Roughly, our negative result can be interpreted as showing that this is essentially the *only* way to construct predicate encryption (in a black-box way) from CPA-secure encryption; our positive result shows how to implement the above for a specific class of predicate encryption schemes. We now provide further details.

Impossibility results. Our negative results are in the same model used by Boneh et al. [5], which builds on the model used in the seminal work of Impagliazzo and Rudich [12]. Specifically, as in [5] our negative results hold relative to a *random* oracle (with trapdoor) and so rule out black-box constructions from trapdoor permutations as well as from any (standard) CCA-secure public-key encryption scheme.

[1] This is stronger than what is required, but makes sense in a black-box setting where computational hardness comes only from the underlying CPA-secure scheme.

A slightly informal statement of our result follows. Fix $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$, a sequence of predicate classes and attribute sets indexed by the security parameter n. We say that $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ can be *q-covered* if for every set system $\{S_f\}_{f\in\mathcal{F}_n}$ with $S_f \subseteq [q(n)]$ ($[q] \stackrel{\text{def}}{=} \{1, \ldots, q\}$), there are polynomially-many predicates $f^*, f_1, \ldots, f_p \in \mathcal{F}_n$ such that, with high probability:

1. $S_{f^*} \subseteq \bigcup_{i=1}^{p} S_{f_i}$.
2. There exists an $I \in \mathbb{A}_n$ with $f_1(I) = \cdots = f_p(I) = 0$ but $f^*(I) = 1$.

$\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is *easily covered* if it is q-covered for *every* polynomial q. We show:

Theorem *If $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is easily covered, there is no black-box construction of a predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ based on trapdoor permutations (or CCA-secure encryption).*

Intuitively, if $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is easily covered then the combinatorial approach discussed earlier cannot work: letting $q(n)$ be the (necessarily) polynomial number of keys for the underlying (standard) encryption scheme, no matter how the secret keys $\{sk_i\}_{i=1}^{q}$ are apportioned to the personal secret keys $\{SK_f\}_{f\in\mathcal{F}_n}$, an adversary can carry out the following attack (cf. Definition 2, below):

1. Request the keys $SK_{f_1}, \ldots, SK_{f_p}$, where each $SK_{f_i} = \{sk_1, \ldots, \} \subseteq \{sk_i\}_{i=1}^{q}$.
2. Request the challenge ciphertext C to be encrypted using an attribute I for which $f_1(I) = \cdots = f_p(I) = 0$ but $f^*(I) = 1$.
3. Compute the key $SK_{f^*} \subseteq \bigcup_i SK_{f_i}$ and use this key to decrypt C.

This constitutes a valid attack since SK_{f^*} suffices to decrypt C yet the adversary only requested $SK_{f_1}, \ldots, SK_{f_p}$, none of which suffices on its own to decrypt C.

Turning this intuition into a formal proof must, in particular, implicitly show that the combinatorial approach sketched earlier is essentially the *only* black-box approach to building predicate encryption schemes from trapdoor permutations. Moreover, we actually prove a stronger *quantitative* version of the above theorem showing, roughly, that if $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is q-covered then any predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ must use at least $q + 1$ underlying encryption keys.

One might wonder whether the "easily covered" condition is useful for determining whether there exist black-box constructions of predicate encryption schemes over $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ of interest. We show that it is, in that the following corollary can be proven fairly easily given the above:

Corollary *There are no black-box constructions of (1) identity-based encryption, (2) forward-secure encryption (for a super-polynomial number of time periods), or (3) broadcast encryption (where a super-polynomial number of users can be excluded) from trapdoor permutations.*

The first result was shown in [5]; the point is that our impossibility result strictly generalizes theirs. Moreover, as indicated earlier, we prove a *quantitative* version of their result (as well as all other results stated in the above corollary).

Positive result. On the positive side, we show that the combinatorial approach suggested at the outset *can* be implemented for $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ having the following

property: for each $I \in \mathbb{A}_n$ there are at most polynomially-many $f \in \mathcal{F}_n$ for which $f(I) = 0$; i.e., for each I there are at most polynomially-many predicates that are "excluded". (The positive result from [6], where there are only polynomially-many predicates, is thus obtained as a corollary.) This is proved by analogy to broadcast encryption, using the combinatorial techniques from [14].

1.2 Comparison to the Results of Boneh et al.

Our proof relies heavily on the impossibility result from [5]. Our contribution lies in finding the right combinatorial generalization (specifically, the "easily covered" property described earlier) of the *specific* property used by Boneh et al. for the particular case of IBE, adapting their proof to our setting, and applying their ideas to the more general case of predicate encryption. Our generalization, in turn, allows us to show impossibility for several cryptosystems of interest besides IBE (cf. the corollary stated earlier), as well as to give quantitative versions of their earlier result. Our positive results have no analogue in [5].

2 Definitions

2.1 Predicate Encryption

We provide a functional definition of predicate encryption, followed by a weak definition of security that we use when proving impossibility and the standard definition of security [13] that we use when proving our positive result.

Definition 1. *Fix $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$, where $\mathcal{F}_n$ is a set of (efficiently computable) predicates over the set of attributes $\mathbb{A}_n$. A* predicate encryption scheme *over $\{\mathcal{F}_n, \mathbb{A}_n\}_{n\in\mathbb{N}}$ consists of four* PPT *algorithms* (Setup, KeyGen, Enc, Dec) *such that:*

- Setup *is a deterministic algorithm that takes as input a master secret key $MSK \in \{0,1\}^n$ and outputs a master public key MPK.*
- KeyGen *is a deterministic algorithm that takes as input the master secret key MSK and a predicate $f \in \mathcal{F}_n$ and outputs a secret key $SK_f = \mathsf{KeyGen}_{MSK}(f)$. (The assumption that* KeyGen *is deterministic is without loss of generality, since MSK may include a key for a pseudorandom function.)*
- Enc *takes as input the public key MPK, an attribute $I \in \mathbb{A}_n$, and a bit b. It outputs a ciphertext $C \leftarrow \mathsf{Enc}_{MPK}(I, b)$.*
- Dec *takes as input a secret key SK_f and ciphertext C. It outputs either a bit b or the distinguished symbol $\perp$.*

It is required that for all n, all $MSK \in \{0,1\}^n$ and $MPK = \mathsf{Setup}(MSK)$, all $f \in \mathcal{F}_n$ and $SK_f = \mathsf{KeyGen}_{MSK}(f)$, all $I \in \mathbb{A}_n$, and all $b \in \{0,1\}$, that if $f(I) = 1$ then $\mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, b)) = b$.

Definition 2. *A predicate encryption scheme over $(\mathcal{F}, \mathbb{A})$ is* weakly payload hiding *if the advantage of any* PPT *adversary $\mathcal{A}$ in the following game is negligible:*

1. *$\mathcal{A}(1^n)$ outputs $I^* \in \mathbb{A}_n$ and $(f_1, \ldots, f_p) \in \mathcal{F}_n$ such that $f_i(I^*) = 0$ for all i.*
2. *Choose $MSK \leftarrow \{0,1\}^n$; let $MPK := \mathsf{Setup}(MSK)$ and set $SK_{f_i} := \mathsf{KeyGen}(MSK, f_i)$ for all i. Choose $b \leftarrow \{0,1\}$, and compute the ciphertext $C^* \leftarrow \mathsf{Enc}_{MPK}(I^*, b)$. Then $\mathcal{A}$ is given $(MPK, SK_{f_1}, \ldots, SK_{f_p}, C^*)$.*
3. *$\mathcal{A}$ outputs b' and succeeds if $b' = b$.*

The advantage of $\mathcal{A}$ is defined as $\left|\Pr[\mathcal{A}\ succeeds] - \frac{1}{2}\right|$.

Definition 3. *A predicate encryption scheme over $(\mathcal{F}, \mathbb{A})$ is* payload hiding *if the advantage of any* PPT *adversary $\mathcal{A}$ in the following game is negligible:*

1. *A random $MSK \in \{0,1\}^n$ is chosen, and $\mathcal{A}$ is given $MPK := \mathsf{Setup}(MSK)$.*
2. *$\mathcal{A}$ adaptively requests keys $SK_{f_1}, \ldots$ corresponding to predicates $f_1, \ldots \in \mathcal{F}_n$.*
3. *At some point, $\mathcal{A}$ outputs $I^* \in \mathbb{A}_n$. A random $b \in \{0,1\}$ is chosen and $\mathcal{A}$ is given the ciphertext $C^* \leftarrow \mathsf{Enc}_{MPK}(I^*, b)$. $\mathcal{A}$ may continue to request keys for predicates of its choice.*
4. *$\mathcal{A}$ outputs b' and succeeds if (1) $\mathcal{A}$ never requested a key for a predicate f with $f(I^*) = 1$, and (2) $b' = b$.*

The advantage of $\mathcal{A}$ is defined as $\left|\Pr[\mathcal{A}\ succeeds] - \frac{1}{2}\right|$.

Our construction of Section 5 can be modified to achieve the even stronger notion of *attribute hiding*; we refer to [13] for a definition.

2.2 A Random Trapdoor Permutation Oracle

We assume the reader is familiar with the usual model in which black-box impossibility results are proved; see [12, 17, 5] for further details. We show an oracle $\mathcal{O}$ relative to which trapdoor permutations and CCA-secure encryption exist, yet any construction of a predicate encryption scheme (for certain $(\mathcal{F}, \mathbb{A})$) relative to $\mathcal{O}$ is insecure against a polynomial-time adversary given access to $\mathcal{O}$ and a PSPACE oracle. Our oracle $\mathcal{O} = (g, e, d)$ is defined as follows, for each $n \in \mathbb{N}$:

- g is chosen uniformly from the space of permutations on $\{0,1\}^n$. We view g as taking a secret key sk as input, and returning a public key pk.
- $e : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ maps a public key pk and a "message" $m \in \{0,1\}^n$ to a "ciphertext" $c \in \{0,1\}^n$. It is chosen uniformly subject to the constraint that $e(pk, \cdot)$ is a permutation on $\{0,1\}^n$ for every pk.
- $d : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ maps a secret key sk and a ciphertext c to a message m. We require that $d(sk, c)$ outputs the unique m for which $e(g(sk), m) = c$.

With overwhelming probability $\mathcal{O}$ is a trapdoor permutation [10, 5]. Moreover, since the components of $\mathcal{O}$ are chosen at *random* subject to the above constraints (and not with some "defect" as in, e.g., [10]), $\mathcal{O}$ implies CCA-secure encryption [1].

We denote a query α to $\mathcal{O}$ as, e.g., $\alpha \stackrel{\text{def}}{=} [g(sk) = pk]$ and similarly for e and d queries. In describing our attack in the next section, we often use a partial oracle $\mathcal{O}'$ that is defined only on some subset of the possible inputs. We always enforce that such oracles be *consistent*:

Definition 4. *A partial oracle $\mathcal{O}' = (g', e', d')$ is* consistent *if:*

1. *For every $pk \in \{0,1\}^n$, the (partial) function $e'(pk, \cdot)$ is one-to-one.*
2. *For every $sk \in \{0,1\}^n$, the (partial) function $d'(sk, \cdot)$ is one-to-one.*
3. *For all $x \in \{0,1\}^n$, and all sk such that $g'(sk) = pk$ is defined, the value $e'(pk, x) = c$ is defined if and only if $d'(sk, c) = x$ is defined.*

3 An Impossibility Result for Predicate Encryption

We define a combinatorial property on $(\mathcal{F}_n, \mathbb{A}_n)$ and formally state our impossibility result. We describe in Section 3.1 an adversary $\mathcal{A}$ attacking any black-box construction of a predicate encryption scheme satisfying the conditions of our theorem; an analysis of $\mathcal{A}$ is given in Appendix A and the full version.

Fix a set $\mathcal{F}$ and a positive integer q, and let $[q] \stackrel{\text{def}}{=} \{1, \ldots, q\}$. An *$\mathcal{F}$-set system over $[q]$* is a collection of sets $\{S_f\}_{f \in \mathcal{F}}$ where each $f \in \mathcal{F}$ is associated with a set $S_f \subseteq [q]$.

Definition 5. *Let $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ be a sequence of predicates and attributes. We say $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ can be q-*covered *if there exist* PPT *algorithms (A_1, A_2, A_3), where $A_2(1^n, f)$ is deterministic and outputs $I \in \mathbb{A}_n$ with $f(I) = 1$, such that for n sufficiently large:*

For any $\mathcal{F}_n$-set system $\{S_f\}_{f \in \mathcal{F}_n}$ over $[q(n)]$, if we compute

$$f^* \leftarrow A_1(1^n); \;\; I^* := A_2(1^n, f^*); \;\; f_1, \ldots, f_p \leftarrow A_3(1^n, f^*),$$

then with probability at least $4/5$,

1. *$S_{f^*} \subseteq \bigcup S_{f_i}$;*
2. *$f_i(I^*) = 0$ for all i.*

$\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ is easily covered *if it can be q-covered for every polynomial q.*

Although the above definition may seem rather complex and hard to use, we show in Section 4 that it can be applied quite easily to several interesting classes of predicate encryption schemes. Moreover, the definition is natural given the attack we will describe in the following section.

A black-box construction of predicate encryption is *q-bounded* if each of its algorithms makes at most q queries to $\mathcal{O}$. We now state our main result:

Theorem 1. *If $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ can be q-covered, then there is no q-bounded black-box construction of a weakly payload-hiding predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ from trapdoor permutations (or CCA-secure encryption).*

Since each algorithm defining the predicate encryption scheme can make at most polynomially-many queries to its oracle, we have

Corollary 1. *If $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ is easily covered, there is* **no** *black-box construction of a weakly payload-hiding predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ from trapdoor permutations (or CCA-secure encryption).*

3.1 The Attack

Fix an $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ that can be q-covered, and let $\mathsf{PE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ each of whose algorithms makes at most $q = \mathsf{poly}(n)$ queries to $\mathcal{O} = (g, e, d)$. We assume, without loss of generality, that before any algorithm of PE makes a query of the form $[d(sk, \star)]$, it first makes the query $[g(sk)]$.

We begin the proof of Theorem 1 by describing an adversary $\mathcal{A}$ attacking PE. Adversary $\mathcal{A}$ is given access to $\mathcal{O}$ and makes a polynomial number of calls to this oracle; as described, $\mathcal{A}$ is not efficient but it runs in polynomial time given access to a PSPACE-complete oracle (or if $\mathcal{P} = \mathcal{NP}$) and this suffices to prove black-box impossibility as in previous work [12, 17, 5]. Our description of the attack is directly motivated by the attacker described in [5].

Let A_1, A_2, and A_3 be as guaranteed by Definition 5, and let $p = \mathsf{poly}(n)$ bound the number of predicates output by A_3. Throughout $\mathcal{A}$'s execution, when it makes a query to $\mathcal{O}$ it stores the query and the response in a list L. We also require that before $\mathcal{A}$ makes any query of the form $[d(sk, \star)]$, it first makes the query $[g(sk)]$. Furthermore, once the query $[g(sk) = pk]$ has been made then $[e(pk, x) = y]$ is added to L if and only if $[d(sk, y) = x]$ is added to L.

Setup and challenge. $\mathcal{A}(1^n)$ computes $f^* \leftarrow A_1(1^n)$, $I^* := A_2(1^n, f^*)$, and $(f_1, \ldots, f_p) \leftarrow A_3(1^n, f^*)$. Then:

1. If $f_i(I^*) = 0$ for all i, then $\mathcal{A}$ outputs $(I^*, f_1, \ldots, f_p)$ and receives the values $(MPK, SK_{f_1}, \ldots, SK_{f_p}, C^*)$ from the challenger (cf. Definition 2).
2. Otherwise, $\mathcal{A}$ aborts and outputs a random bit $b' \leftarrow \{0,1\}$.

Step 1: Discovering important public keys. For $i = 1$ to p, adversary $\mathcal{A}$ does the following:

1. Compute $I_{f_i} = A_2(1^n, f_i)$, and choose random $b \leftarrow \{0,1\}$ and $r \leftarrow \{0,1\}^n$.
2. Compute $\mathsf{Dec}^{\mathcal{O}}_{SK_{f_i}}\left(\mathsf{Enc}^{\mathcal{O}}_{MPK}(I_{f_i}, b; r)\right)$, storing all $\mathcal{O}$-queries in the list L.

Step 2: Discovering frequent queries for I^*. $\mathcal{A}$ repeats the following $q \cdot p^3$ times: Choose random $b \leftarrow \{0,1\}$ and $r \leftarrow \{0,1\}^n$; compute $\mathsf{Enc}^{\mathcal{O}}_{MPK}(I^*, b; r)$, storing all $\mathcal{O}$-queries in L.

Step 3: Discovering secret queries and decrypting the challenge. $\mathcal{A}$ chooses $k \leftarrow [q \cdot p^3]$ and runs the following k times.

1. $\mathcal{A}$ uniformly generates a secret key MSK' and a consistent partial oracle $\mathcal{O}'$ for which (1) $\mathsf{Setup}^{\mathcal{O}'}(MSK') = MPK$; (2) for all i it holds that $\mathsf{KeyGen}^{\mathcal{O}'}_{MSK'}(f_i) = SK_{f_i}$; (3) the oracle $\mathcal{O}'$ is consistent with L; and (4) the key $SK'_{f^*} \stackrel{\text{def}}{=} \mathsf{KeyGen}^{\mathcal{O}'}_{MSK'}(f^*)$ is well-defined.

   We denote by L' the set of queries in $\mathcal{O}'$ that are not in L (the "invented queries"). Note that $|L'| \leq q \cdot (p+2)$, since at most q queries are made by Setup and $\mathsf{KeyGen}(f)$ makes at most q queries for each of $SK_{f^*}, SK_{f_1}, \ldots, SK_{f_p}$.

2. $\mathcal{A}$ chooses $b \leftarrow \{0,1\}$ and $r \leftarrow \{0,1\}^n$, and computes $C := \mathsf{Enc}^{\mathcal{O}}_{MPK}(I^*, b; r)$ (storing all $\mathcal{O}$-queries in L). For an oracle $\mathcal{O}''$ defined below, $\mathcal{A}$ then does:
   (a) In iteration $k' < k$, adversary $\mathcal{A}$ computes $\mathsf{Dec}^{\mathcal{O}''}_{SK'_{f^*}}(C)$.
   (b) In iteration k, adversary $\mathcal{A}$ computes $b' = \mathsf{Dec}^{\mathcal{O}''}_{SK'_{f^*}}(C^*)$.

Output: $\mathcal{A}$ Outputs the bit b' computed in the k^{th} iteration of step 3.

Before defining the oracle $\mathcal{O}''$ used above, we introduce some notation. Let L, $\mathcal{O}'$, and MSK' be as above, and note that we can view L and $\mathcal{O}'$ as a tuple of (partial) functions (g, e, d) and (g', e', d') where g', e', and d' extend g, e, and d, respectively. Define the following:

- $\mathcal{Q}'_S$ is the set of pk for which $[g'(sk) = pk]$ is queried during computation of $\mathsf{Setup}^{\mathcal{O}'}(MSK')$.
- $\mathcal{Q}'_K$ is the set of pk for which $[g'(sk) = pk]$ is queried during computation of $\mathsf{KeyGen}^{\mathcal{O}'}_{MSK'}(f)$ for some $f \in \{f^*, f_1, \ldots, f_p\}$.
- $\mathcal{Q}'_{K-S} = \mathcal{Q}'_K \setminus \mathcal{Q}'_S$.
- L_g is the set of pk for which the query $[g(sk) = pk]$ is in L.

Note that $\mathcal{A}$ can compute each of these sets from its view. Note further that $\mathcal{Q}'_S, \mathcal{Q}'_K, \mathcal{Q}'_{K-S}, \mathcal{O}'$ are fixed throughout an iteration of step 3, but L_g may change as queries are answered.

Oracle $\mathcal{O}''$ is defined as follows. For any query whose answer is defined by $\mathcal{O}'$, return that answer. Otherwise:

1. For an encryption query $e(pk, x)$ with $pk \in \mathcal{Q}'_{K-S} \setminus L_g$, return a random y consistent with the rest of $\mathcal{O}''$. Act analogously for a decryption query $d(sk, y)$ with $pk \in \mathcal{Q}'_{K-S} \setminus L_g$ (where $pk = g(sk)$).
2. For a decryption query $d(sk, y)$, if there exists a pk with $[g(sk) = pk] \in \mathcal{O}'$ but[2] there exists an $sk' \neq sk$ with $[g(sk') = pk] \in L$, then use $\mathcal{O}''$ to answer the query $d(sk', y)$.
3. In any other case, query the real oracle $\mathcal{O}$ and return the result. Store the query/answer in L (note that this might affect L_g as well).

An analysis of $\mathcal{A}$, proving Theorem 1, appears in Appendix A and the full version of our paper. The analysis is very similar to the one given in [5], with the main difference being Proposition 1.

4 Impossibility for Specific Cases

We use Theorem 1 to rule out black-box constructions of predicate encryption schemes in several specific cases of interest. Specifically, we consider the cases of identity-based encryption, forward-secure encryption, and broadcast encryption. We begin with a useful lemma.

[2] Although $\mathcal{O}'$ is chosen to be consistent, a conflict can occur since L is updated as $\mathcal{A}$ makes additional queries to the real oracle $\mathcal{O}$.

Lemma 1. *Fix $q(\cdot)$, and assume $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$ has the following property: For sufficiently large n, there exist $f_1, \ldots, f_{5q} \in \mathcal{F}_n$ and $I_1, \ldots, I_{5q} \in \mathbb{A}_n$ such that:*

$$\text{For all } i \in \{1, \ldots, 5q\} \text{ it holds that } f_i(I_i) = 1 \text{ but } f_j(I_i) = 0 \text{ for } j > i.$$

Then $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$ can be q-covered. If the above holds for every polynomial q, then $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$ is easily covered.

Proof. We show that, under the stated assumption, $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$ satisfies Definition 5. Fix q and n large enough so that the condition of the lemma holds, and let $f_1, \ldots, f_{5q}$ and $I_1, \ldots, I_{5q}$ be as stated. Define algorithms A_1, A_2, A_3 as follows:

1. $A_1(1^n)$ chooses $i \leftarrow \{0, \ldots, 5q\}$ and outputs $f^* = f_i$.
2. $A_2(1^n, f^*)$ finds i for which $f^* = f_i$ and outputs $I^* = I_i$.
3. $A_3(1^n, f^*)$ finds i for which $f^* = f_i$ and outputs $f_{i+1}, \ldots, f_{5q}$. (If $i = 5q$ then output nothing.)

Note that $A_2(1^n, f^*)$ always outputs I^* with $f^*(I^*) = 1$. We show that for *any* $\mathcal{F}_n$-set system $\{S_f\}_{f\in\mathcal{F}_n}$ over $[q]$, the conditions of Definition 5 hold. We begin with the following claim:

Claim. For any $\mathcal{F}_n$-set system $\{S_f\}_{f\in\mathcal{F}_n}$ over $[q]$, there are at most q values $i \in \{1, \ldots, 5q\}$ for which $S_{f_i} \not\subseteq \bigcup_{i<j\leq 5q} S_{f_j}$. (By convention, the union is the empty set if $j = 5q$.)

Proof. Define $\mathbf{S}_i \stackrel{\text{def}}{=} \bigcup_{i<j\leq 5q} S_{f_j}$, with $\mathbf{S}_{5q} = \emptyset$. Note that $\mathbf{S}_{i-1} = \mathbf{S}_i \cup S_{f_i}$, and so $S_{f_i} \not\subseteq \bigcup_{i<j\leq 5q} S_{f_j} = \mathbf{S}_i$ iff $\mathbf{S}_i \subsetneq \mathbf{S}_{i-1}$. Since

$$\mathbf{S}_{5q} \subseteq \mathbf{S}_{5q-1} \subseteq \cdots \subseteq \mathbf{S}_1 \subseteq [q],$$

there can be at most q indices i where this occurs. □

Fixing an arbitrary $\mathcal{F}_n$-set system $\{S_f\}_{f\in\mathcal{F}_n}$ over $[q]$, let $\mathbb{I} \subset \{1, \ldots, 5q\}$ be the set of indices for which $S_{f_i} \subseteq \bigcup_{i<j\leq q} S_{f_j}$; the claim above shows that $|\mathbb{I}| \geq 4q$. If A_1 chooses $i \in \mathbb{I}$ then:

1. $S_{f^*} = S_{f_i} \subseteq \bigcup_{i<j\leq q} S_{f_j}$.
2. $f_j(I^*) = f_j(I_i) = 0$ for all the predicates $f_{i+1}, \ldots, f_q$ output by A_3.

Since A_1 chooses $i \in \mathbb{I}$ with probability $4/5$, this proves the lemma. □

We now apply Lemma 1 to several specific cases.

Identity-based encryption. It is easy to see that IBE for identities $\{\mathcal{I}_n\}$ can be viewed as an instance of predicate encryption by setting $\mathbb{A}_n = \mathcal{I}_n$ and $\mathcal{F}_n = \{f_{ID}\}_{ID\in\mathcal{I}_n}$ where

$$f_{ID}(ID') \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } ID' = ID \\ 0 & \text{otherwise} \end{cases}.$$

Let $N = |\mathcal{I}_n|$ denote the size of the identity space. Boneh et al. [5] already rule out black-box constructions of IBE from trapdoor permutations for $N = \omega(\mathsf{poly}(n))$; the next theorem shows that our Theorem 1 generalizes their result:

Theorem 2. *There is no black-box construction (from trapdoor permutations or CCA-secure encryption) of an IBE scheme for $5N$ identities where each algorithm makes fewer than N queries to its oracle.*

As a corollary, there is no black-box construction of an IBE scheme (from trapdoor permutations or CCA-secure encryption) for a super-polynomial number of identities.

Proof. Let $\mathcal{I}_n = \{ID_1, \ldots, ID_{5N}\}$. It is not hard to see that $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$ can be N-covered: take $f_{ID_1}, \ldots, f_{ID_{5N}}$ and set $I_i = ID_i$ for all i. Then apply Theorem 1. □

Forward-secure public-key encryption. In a forward-secure public-key encryption scheme [7] secret keys are associated with time periods; the secret key at time period i enables decryption for ciphertexts encrypted at any time $j \geq i$. (We refer the reader to [7] for further discussion.) A forward-secure encryption scheme supporting $N = N(n)$ time periods can be cast as a predicate encryption scheme by letting $\mathbb{A}_n = \{1, \ldots, N\}$ and $\mathcal{F}_n = \{f_i\}_{1\leq i\leq N}$ where

$$f_i(j) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } j \geq i \\ 0 & \text{otherwise} \end{cases}.$$

(A forward-secure encryption scheme imposes the additional requirement that $SK_{f_{i+1}}$ can be derived from SK_{f_i}; since we do not impose this requirement our impossibility result is even stronger.) A black-box construction of a forward-secure encryption scheme from any CPA-secure encryption scheme exists for any $N = \mathsf{poly}(n)$: the master public key contains public keys $\{pk_1, \ldots, pk_N\}$, and the secret key at period i is $SK_{f_i} = \{sk_i, \ldots, sk_N\}$; encryption at period j uses pk_j. While such a scheme is trivial as far as forward-secure encryption goes (since the public/secret key lengths are linear in N), it satisfies the definition. The next theorem indicates that, in some sense, this trivial construction is almost optimal as far as black-box constructions are concerned; moreover, there is no black-box construction supporting a super-polynomial number of time periods. (In contrast, there exist schemes based on specific assumptions [7, 3] that support an unbounded number of time periods.)

Theorem 3. *There is no black-box construction (from trapdoor permutations or CCA-secure encryption) of a forward-secure encryption scheme for $5N$ periods where each algorithm in the scheme makes fewer than N queries to its oracle.*

As a corollary, there is no black-box construction of a forward-secure encryption scheme (from trapdoor permutations or CCA-secure encryption) supporting a super-polynomial number of time periods.

Proof. $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$ can be N-covered, as taking $f_1, \ldots, f_{5N}$ and setting $I_i = i$ for all i satisfies the conditions of Lemma 1. Then apply Theorem 1. □

Broadcast encryption. Finally, we look at the case of (public-key) broadcast encryption [9]. Here, there is a fixed public key and a set of users $\mathcal{U} = \{1, \ldots, U\}$

each with their own personal secret key; it should be possible for a sender to encrypt a message in such a way that only some subset $\mathcal{U}' \subset \mathcal{U}$ of users can decrypt. Consider the case where at most $k = k(n) < U$ users are excluded; we refer to this as *k-exclusion broadcast encryption*. This can also be modeled by predicate encryption, if we let $\mathbb{A}_n = \{\mathcal{U}' \subseteq \mathcal{U} \mid |\mathcal{U}'| \geq U - k\}$ and define $\mathcal{F}_n = \{f_i\}_{i \in \mathcal{U}}$ where

$$f_i(\mathcal{U}') \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } i \in \mathcal{U}' \\ 0 & \text{otherwise} \end{cases}.$$

Theorem 4. *There is no black-box construction (from trapdoor permutations or CCA-secure encryption) of a $(5k)$-exclusion broadcast encryption scheme where each algorithm in the scheme makes k or fewer queries to its oracle.*

As a corollary, there is no black-box construction of a k-exclusion broadcast encryption scheme (from trapdoor permutations or CCA-secure encryption) for super-polynomial k.

Proof. We show that $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ can be k-covered. Take $f_1, \ldots, f_{5k}$ and define

$$I_i \stackrel{\text{def}}{=} \mathcal{U} \setminus \{i, \ldots, 5k\}$$

for $i \in \{1, \ldots, 5k\}$. (So $I_{5k} = \mathcal{U}$.) Note that $|I_i| \geq U - 5k$ always, and these satisfy the conditions of Lemma 1. Applying Theorem 1 concludes the proof. □

5 A Possibility Result for Predicate Encryption

Here we show that for the class of predicates and attributes $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ where (roughly) for each $I \in \mathbb{A}_n$ there are at most polynomially-many $f \in \mathcal{F}_n$ with $f(I) = 0$, there is a black-box construction of a predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ based on any CPA-secure encryption scheme. We remark that while we only prove payload hiding, our construction can in fact be shown to be attribute hiding [13] as well.

Our construction relies on the notion of an (N, k)-cover free family [8]:

Definition 6. *An (N, k)-*cover free family over $[U]$* *is a family $\mathcal{S} = \{S_1, \ldots, S_N\}$, with $S_i \subseteq [U]$, such that for any distinct sets $S, S_1, \ldots, S_k \in \mathcal{S}$ it holds that $S \setminus \bigcup_{i=1}^{k} S_i \neq \emptyset$.*

For any $k = \mathsf{poly}(n)$ and $N = 2^{\mathsf{poly}(n)}$ there exist [14, 16] explicit, polynomial-time constructions of an (N, k)-cover free family over $[U]$ with $|U| = \mathsf{poly}(n)$. (The specific results of [14, 16] can be used to improve the efficiency of the construction that follows, but our only goal here is to show a construction that can be implemented in polynomial time.)

Theorem 5. *Fix $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ and set $\mathsf{Neg}_I \stackrel{\text{def}}{=} \{f \in \mathcal{F}_n : f(I) = 0\}$ for $I \in \mathbb{A}_n$. If there is a poly-time algorithm $\mathsf{ListNeg}$ for which $\mathsf{ListNeg}(1^n, I) = \mathsf{Neg}_I$, then there is a black-box construction of a predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ from any CPA-secure encryption scheme.*

Proof. Since $\mathsf{ListNeg}$ runs in polynomial time, there is a polynomial k for which $|\mathsf{Neg}_I| \leq k(n)$ for all $I \in \mathbb{A}_n$. Say predicates in $\mathcal{F}_n$ can be represented using $\ell(n) = \mathsf{poly}(n)$ bits. Let $\{U_n\}$ be such that $U_n = \mathsf{poly}(n)$ and such that, for each n, there is an explicit $(2^{\ell(n)}, k(n))$-cover free family $\mathcal{S} = \{S_1, \ldots, S_{2^{\ell(n)}}\}$ over $[U_n]$. Identifying $\mathcal{F}_n$ with a subset of $[2^{\ell(n)}]$, we can view the cover-free family as $\mathcal{S} = \{S_f\}_{f \in \mathcal{F}_n}$.

Let $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ be a CPA-secure encryption scheme. Our construction of a predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ is as follows:

- Setup, on input 1^n and a sufficiently long random string MSK, runs $\mathsf{Gen}'(1^n)$ a total of $U = U_n$ times to generate keys $(pk_1, sk_1), \ldots, (pk_U, sk_U)$. The master public key is $\{pk_1, \ldots, pk_U\}$.
- KeyGen, given the secret keys $\{sk_i\}_{i=1}^U$ and a predicate $f \in \mathcal{F}_n$, outputs the subset $\{sk_i\}_{i \in S_f}$.
- Enc, given the public key, an attribute $I \in \mathbb{A}_n$, and a message m, computes $\mathsf{Neg}_I = \mathsf{ListNeg}(I)$ and sets $\bar{U} = [U] \setminus \left(\bigcup_{f \in \mathsf{Neg}_I} S_f\right)$. The ciphertext is $(I, \{C_i\}_{i \in \bar{U}})$ where $C_i \leftarrow \mathsf{Enc}'_{pk_i}(m)$.
- Dec, given the secret key $\{sk_i\}_{i \in S_f}$ for a predicate f and a ciphertext $(I, \{C_i\}_{i \in \bar{U}})$ for which $f(I) = 1$, first finds an index i for which $i \in S_f \cap \bar{U}$. (Such an index must exist, since

  $$S_f \setminus \bar{U} = S_f \setminus \bigcup_{f' : f'(I) = 0} S_{f'},$$

  and there are at most k predicates f' that the union is taken over.) The output is $\mathsf{Dec}'_{sk_i}(C_i)$.

It is easy to see that the above construction satisfies correctness. We now prove security (in the sense of Definition 3). Let $\mathcal{A}$ be an adversary attacking the scheme. We may assume without loss of generality that $\mathcal{A}$ never requests a secret key for a predicate f for which $f(I^*) = 1$ (where I^* is the attribute used to encrypt the challenge ciphertext), since $\mathcal{A}$ cannot succeed if that occurs.

For simplicity we prove security in a non-uniform model, but the proof can be modified easily to hold in the uniform model in the standard way. We consider $U+1$ hybrid experiments $H_0, \ldots, H_{U+1}$, where H_0 corresponds to the experiment of Definition 3 when $b = 0$ is encrypted, and H_{U+1} corresponds to the experiment of Definition 3 when $b = 1$ is encrypted. Let δ_i denote the probability that $\mathcal{A}$ outputs '0' in H_i. We show that $|\delta_i - \delta_{i+1}|$ is negligible for all i; since $U = U_n$ is polynomial in n, this proves that $|\delta_0 - \delta_{U+1}|$ is negligible and thus completes the proof.

Experiment H_i is defined as follows: Steps 1 and 2 are exactly as in Definition 3. In step 3, however, when encrypting the challenge ciphertext for the attribute I^*, let $\bar{U}^* = [U] \setminus \mathsf{Neg}_{I^*}$ and set the ciphertext equal to $\left(I, \{C_j\}_{j \in \bar{U}^*}\right)$, where

$$C_j \leftarrow \begin{cases} \mathsf{Enc}'_{pk_j}(1) & j < i \\ \mathsf{Enc}'_{pk_j}(0) & j \geq i \end{cases}.$$

$\mathcal{A}$ may continue to request secret keys as in Definition 3.

We now prove that $|\delta_j - \delta_{j+1}|$ is negligible for any j. Fix j and consider the following adversary $\mathcal{A}'$ attacking the underlying encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$. Given public key pk and ciphertext C (which is either an encryption of 0 or 1), the adversary $\mathcal{A}'$ proceeds as follows:

1. Set $pk_j = pk$. For $i \neq j$, compute $(pk_i, sk_i) \leftarrow \mathsf{Gen}'(1^n)$. Give the master public key $\{pk_1, \ldots, pk_U\}$ to $\mathcal{A}$.
2. When $\mathcal{A}$ requests a secret key for a predicate f, then if $j \notin S_f$ give to $\mathcal{A}$ the secret keys $\{sk_i\}_{i \in S_f}$. Otherwise, abort and output a random bit.
3. When $\mathcal{A}$ outputs I^*, compute $\mathsf{Neg}_{I^*} = \mathsf{ListNeg}(I^*)$ and then set

$$\bar{U}^* = [U] \setminus \left(\bigcup_{f \in \mathsf{Neg}_{I^*}} S_f \right).$$

   If $j \notin \bar{U}^*$ then abort and output a random bit. Otherwise, give $\mathcal{A}$ the ciphertext $(I, \{C_i\}_{i \in \bar{U}^*})$ where

$$C_i \leftarrow \begin{cases} \mathsf{Enc}'_{pk_i}(1) & i < j \\ C & i = j \\ \mathsf{Enc}'_{pk_i}(0) & i > j \end{cases}.$$

4. Subsequent secret key queries made by $\mathcal{A}$ are answered as before. Finally, $\mathcal{A}'$ outputs whatever bit is output by $\mathcal{A}$.

Let $\Pr_j[\cdot]$ denote the probability of an event in experiment H_j. We have

$$\begin{aligned} &\left|\Pr[\mathcal{A}' \text{ outputs } 0 \mid C \leftarrow \mathsf{Enc}'_{pk}(0)] - \Pr[\mathcal{A}' \text{ outputs } 0 \mid C \leftarrow \mathsf{Enc}'_{pk}(1)]\right| \\ &\qquad = \left|\Pr\left[j \in \bar{U}^*\right] \cdot \Pr_j\left[\mathcal{A} \text{ outputs } 0 \mid j \in \bar{U}^*\right]\right. \\ &\qquad\quad \left. - \Pr\left[j \in \bar{U}^*\right] \cdot \Pr_{j+1}\left[\mathcal{A} \text{ outputs } 0 \mid j \in \bar{U}^*\right]\right|, \end{aligned}$$

using the facts that (1) $\Pr[j \in \bar{U}^*]$ is independent of whether C is an encryption of 0 or 1 and (2) when C is an encryption of 0 (resp., 1) then the view of $\mathcal{A}$ (assuming $j \in \bar{U}^*$) is identical to its view in H_j (resp., H_{j+1}). Note further that

$$\Pr_j[\mathcal{A} \text{ outputs } 0 \mid j \notin \bar{U}^*] = \Pr_{j+1}[\mathcal{A} \text{ outputs } 0 \mid j \notin \bar{U}^*]$$

since the challenge ciphertext is distributed identically in each case. It follows that

$$\begin{aligned} &\left|\Pr[\mathcal{A}' \text{ outputs } 0 \mid C \leftarrow \mathsf{Enc}'_{pk}(0)] - \Pr[\mathcal{A}' \text{ outputs } 0 \mid C \leftarrow \mathsf{Enc}'_{pk}(1)]\right| \\ &\qquad = \left|\Pr\left[j \in \bar{U}^*\right] \cdot \Pr_j\left[\mathcal{A} \text{ outputs } 0 \mid j \in \bar{U}^*\right]\right. \\ &\qquad\quad \left. - \Pr\left[j \in \bar{U}^*\right] \cdot \Pr_{j+1}\left[\mathcal{A} \text{ outputs } 0 \mid j \in \bar{U}^*\right]\right| \\ &\qquad = \left|\delta_j - \delta_{j+1}\right|, \end{aligned}$$

concluding the proof. □

Acknowledgments

We thank the authors of [5] for providing us with the full version of their paper.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official positions or policies, either expressed or implied, of the US Government, the US Army Research Laboratory, DARPA, the UK Government, or the UK Ministry of Defence. No official endorsement of any kind should be inferred. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References

1. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
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, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, 2005.
4. D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
5. D. Boneh, P. A. Papakonstantinou, C. Rackoff, Y. Vahlis, and B. Waters. On the impossibility of basing identity-based encryption on trapdoor permutations. In *49th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 283–292. IEEE, 2008.
6. 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.
7. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, 2007.
8. P. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israeli Journal of Mathematics*, 51:79–89, 1985.
9. A. Fiat and M. Naor. Broadcast encryption. In *Advances in Cryptology — Crypto '93*, volume 773 of *LNCS*, pages 480–491. Springer, 1994.
10. R. Gennaro, Y. Gertner, J. Katz, and L. Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Computing*, 35(1):217–246, 2005.
11. 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 Conference on Computer and Communications Security*, pages 89–98. ACM Press, 2006.
12. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 44–61. ACM Press, 1989.
13. 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.

14. R. Kumar, S. Rajagopalan, and A. Sahai. Coding constructions for blacklisting problems without computational assumptions. In *Advances in Cryptology — Crypto '99*, volume 1666 of *LNCS*, pages 609–623. Springer, 1999.
15. R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM CCS '07: 14th ACM Conference on Computer and Communications Security*, pages 195–203. ACM Press, 2007.
16. E. Porat and A. Rothschild. Explicit non-adaptive combinatorial group testing schemes. In *Intl. Colloquium on Automata, Languages, and Programming (ICALP), Part I*, volume 5125 of *LNCS*, pages 748–759. Springer, 2008.
17. O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In *1st Theory of Cryptography Conference — TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, 2004.
18. 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.
19. A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology — Crypto '84*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.

A Proof Details

We analyze the success probability of the adversary $\mathcal{A}$ from Section 3.1. Due to space limitations, the proof cannot be reproduced here in its entirety; we have instead aimed to describe those parts of our proof that differ most prominently from the proof of Boneh et al. [5]. The most significant new element in our proof is Proposition 1.

Toward analyzing the success probability of $\mathcal{A}$, we describe a series of experiments, the first of which corresponds to adversary $\mathcal{A}$ interacting in the experiment from Definition 2. We show that, as long as no "bad" events (to be defined later) occur, the statistical distance between the transcripts generated in each of these experiments is not too large. This allows us to bound $\mathcal{A}$'s success probability by comparing it to an appropriate event in the final experiment.

Expt_0**:** This corresponds to $\mathcal{A}$ interacting in the experiment from Definition 2.

Expt_1**:** This is the same as Expt_0 except that $\mathcal{O}''$ (as defined after the k^{th} repetition of step 3) is used instead of $\mathcal{O}$ to compute the challenge ciphertext C^*.

Expt_2**:** This is the same as Expt_1 except that $\mathcal{O}''$ never queries $\mathcal{O}$ (cf. step 3 in the definition of $\mathcal{O}''$); instead, any such queries are answered randomly (subject to ensuring that $\mathcal{O}''$ remains consistent).

Expt_3**:** This is the following experiment with no adversary and using the real oracle $\mathcal{O}$:

Setup and challenge.

1. Compute $f^* \leftarrow A_1(1^n)$, $I^* = A_2(1^n, f^*)$, and $\{f_1, \ldots, f_p\} \leftarrow A_3(1^n, f^*)$.
2. Choose at random $MSK \leftarrow \{0,1\}^n$ and compute $MPK := \mathsf{Setup}^{\mathcal{O}}(MSK)$. If $f_i(I^*) = 1$ for some i, abort and output a random bit.

3. For every predicate $f \in \{f^*, f_1, \ldots, f_p\}$ compute $SK_f := \mathsf{KeyGen}^{\mathcal{O}}_{MSK}(f)$.

Step 1: Discovering important public keys. For $i = 1$ to p do:

1. Compute $I_{f_i} \leftarrow A_2(1^n, f_i)$, and choose random $b_i \leftarrow \{0,1\}$ and $r_i \leftarrow \{0,1\}^n$.
2. Compute $\mathsf{Dec}^{\mathcal{O}}_{SK_{f_i}}(\mathsf{Enc}^{\mathcal{O}}_{MPK}(I_{f_i}, b_i; r_i))$.

Step 2: Decrypting the challenge.

1. Choose $r \leftarrow \{0,1\}^n$, $b \leftarrow \{0,1\}$ and compute $C^* := \mathsf{Enc}^{\mathcal{O}}_{MPK}(I^*, b; r)$.
2. Compute $b' := \mathsf{Dec}^{\mathcal{O}}_{SK_{f^*}}(C^*)$ and output b'. Note that $b' = b$ always.

This completes the description of Expt_3.

For $i \in \{0,1,2\}$ we will be interested in the following transcripts defined in the course of Expt_i. These transcripts contain, in particular, all oracle queries/answers.

- trans^i_{setup}: The transcript of the setup phase. This includes the computation of MPK and $SK_{f_1}, \ldots, SK_{f_p}$, as well as the computation of SK_{f^*} for the f^* chosen by the adversary. (Even though SK_{f^*} is not computed in the experiment, SK_{f^*} is well defined given f^*, MSK, and $\mathcal{O}$.)
- trans^i_{pks}: The transcript of step 1 ("discovering important public keys").
- trans^i_{freq}: The transcript of step 2 ("discovering frequent queries for I^*").
- $\mathsf{trans}^i_{sim\text{-}setup}$: This is the transcript defined by the adversary's choice of MSK' and $\mathcal{O}'$ in the k^{th} repetition of step 3, and can be viewed as the adversary's "guess" for trans^i_{setup}.
- trans^i_*: The transcript of the encryption of C/decryption of C^* in the k^{th} repetition of step 3.
- $\mathsf{trans}^i = (\mathsf{trans}^i_{setup}, \mathsf{trans}^i_{pks}, \mathsf{trans}^i_{sim\text{-}setup}, \mathsf{trans}^i_*)$.

For Expt_3 we define

- $\mathsf{trans}^3_{sim\text{-}setup}$: The transcript of the "setup and challenge" step.
- trans^3_{pks}: The transcript of step 1 ("discovering important public keys").
- trans^3_*: The transcript of step 2 ("decrypting the challenge").
- $\mathsf{trans}^3 = (\mathsf{trans}^3_{pks}, \mathsf{trans}^3_{sim\text{-}setup}, \mathsf{trans}^3_*)$.

For a given transcript, we partition the set of public keys used (i.e., the set of pk's for which $[g(\cdot) = pk] \in \mathsf{trans}$) into the following sets:

- We let $\mathcal{Q}_S(\mathsf{trans})$ denote the public keys queried during execution of Setup:

$$\mathcal{Q}_S(\mathsf{trans}) \stackrel{\text{def}}{=} \{pk \mid \text{the query } [g(\cdot) = pk] \in \mathsf{trans} \text{ is asked by } \mathsf{Setup}\}.$$

  Intuitively, these are the pk's whose corresponding sk's are "useful" for decrypting ciphertexts.
- We let $\mathcal{Q}_K(\mathsf{trans})$ denote the public keys queried by the KeyGen algorithm when some personal secret key is derived:

$$\mathcal{Q}_K(\mathsf{trans}) \stackrel{\text{def}}{=} \{pk \mid [g(\cdot) = pk] \in \mathsf{trans} \text{ is asked by } \mathsf{KeyGen}_{MSK}(\cdot)\}$$
$$\mathcal{Q}_{K-S}(\mathsf{trans}) \stackrel{\text{def}}{=} \mathcal{Q}_K(\mathsf{trans}) \setminus \mathcal{Q}_S(\mathsf{trans}).$$

– Finally, we will also look at the public keys "discovered" during encryption and decryption (cf. step 3 of the experiments):

$$\mathcal{Q}_{ENC+DEC}(\mathsf{trans}, I, f) \stackrel{\text{def}}{=} \{pk \mid [g(\cdot) = pk] \text{ asked by } \mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, \cdot; \cdot))\}$$

A.1 Bounding Probabilities of Bad Events

Fixing the master secret key MSK and the oracle $\mathcal{O}$ (this fixes MPK as well as $\{SK_f\}_{f \in \mathcal{F}}$), we define four "bad" events and bound the probabilities of each of them. Here, we will only describe and bound one of these events; we refer to the full version of our paper for the remainder of the proof.

Let E^i_{NC} be the event that either of the following is true (in Expt_i):

1. $\exists f_i \in \{f_1, \ldots, f_p\}$ such that $f_i(I^*) = 1$.
2. The following condition holds:

$$\mathcal{Q}_{ENC+DEC}(\mathsf{trans}^i_*, I^*, f^*) \bigcap \mathcal{Q}_S(\mathsf{trans}^i_{sim\text{-}setup})$$
$$\not\subseteq \left(\bigcup_{f \in \{f_1, \ldots, f_p\}} \mathcal{Q}_{ENC+DEC}(\mathsf{trans}^i_{pks}, I_f, f) \right) \bigcap \mathcal{Q}_S(\mathsf{trans}^i_{sim\text{-}setup}),$$

where $I_f := A_2(1^n, f)$.

Intuitively, the second condition above is the event that the public keys that are "useful" for $f_1, \ldots, f_p$ does not contain the public keys that are "useful" for f^*.

We bound the probability of E^3_{NC} using the assumed easily-covered property of $\{(\mathcal{F}_n, \mathbb{A}_n)\}$; this is the crux of our proof, and is what motivates Definition 5.

Proposition 1. $\Pr[E^3_{NC}] \leq 1/5$.

Proof. Fix $\mathcal{O}$ and $MSK \in \{0,1\}^n$, thus fixing $\mathsf{trans}^3_{sim\text{-}setup}$. If for each $f \in \mathcal{F}_n$ we fix a random tape r_f that is sufficiently long to run $\mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, b; r))$ (where $I \stackrel{\text{def}}{=} A_2(f)$), then this defines, for each f, the set

$$S_f \stackrel{\text{def}}{=} \left\{ pk \mid [g(\cdot) = pk] \text{ asked by } \mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, b; r)) \right\} \cap \mathcal{Q}_S(\mathsf{trans}^3_{sim\text{-}setup}).$$

Numbering the (at most q) public keys in $\mathcal{Q}_S(\mathsf{trans}^3_{sim\text{-}setup})$ in lexicographic order, we can view these $\{S_f\}_{f \in \mathcal{F}_n}$ as an $\mathcal{F}_n$-set system over $[q]$. The fact that $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ can be q-covered implies that there exists a polynomial p such that

$$\Pr\left[\begin{array}{c} \forall f \in \mathcal{F}_n : r_f \leftarrow \{0,1\}^* \\ f^* \leftarrow A_1,\ I^* := A_2(1^n, f^*) \\ \{f_1, \ldots, f_p\} \leftarrow A_3(f^*) \end{array} : \left(S_{f^*} \subseteq \bigcup_{i=1}^{p} S_{f_i} \right) \bigwedge \left(\forall i : f_i(I^*) = 0 \right) \right] \geq \frac{4}{5}.$$

The above is a lower bound on the probability that E^3_{NC} does not occur. □