Function-Private Subspace-Membership Encryption and Its Applications

Dan Boneh* Ananth Raghunathan† Gil Segev‡

Abstract

Boneh, Raghunathan, and Segev (CRYPTO '13) have recently put forward the notion of *function privacy* and applied it to identity-based encryption, motivated by the need for providing predicate privacy in public-key searchable encryption. Intuitively, their notion asks that decryption keys reveal essentially no information on their corresponding identities, beyond the absolute minimum necessary. While Boneh et al. showed how to construct function-private identity-based encryption (which implies predicate-private encrypted keyword search), searchable encryption typically requires a richer set of predicates.

In this paper we significantly extend the function privacy framework. First, we introduce the notion of *subspace-membership* encryption, a generalization of inner-product encryption, and formalize a meaningful and realistic notion for capturing its function privacy. Then, we present a generic construction of a *function-private* subspace-membership encryption scheme based on *any* inner-product encryption scheme. Finally, we show that function-private subspace-membership encryption can be used to construct function-private identity-based encryption. These are the first generic constructions of function-private encryption schemes based on non-function-private ones, resolving one of the main open problems posed by Boneh, Raghunathan, and Segev.

Keywords: Function privacy, functional encryption.

*Stanford University, Stanford, CA 94305, USA. Email: `dabo@cs.stanford.edu`.

†Stanford University, Stanford, CA 94305, USA. Email: `ananthr@stanford.edu`.

‡School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem 91904, Israel. Email: `segev@cs.huji.ac.il`. Supported by the European Union's Seventh Framework Programme (FP7) via a Marie Curie Career Integration Grant, by the Israel Science Foundation (Grant No. 483/13), and by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11). Most of the work was done while the author was visiting Stanford University.

Contents

1 Introduction

Predicate encryption systems [BW07, KSW08] are public-key schemes where a single public encryption key has *many* corresponding secret keys: every secret key corresponds to a predicate $p : \Sigma \to \{0, 1\}$ where Σ is some pre-defined set of indices (or attributes). Plaintext messages are pairs (x, m) where $x \in \Sigma$ and m is in some message space. A secret key sk_p for a predicate p has the following semantics: if c is an encryption of the pair (x, m) then sk_p can be used to decrypt c only if the "index" x satisfies the predicate p. More precisely, attempting to decrypt c using sk_p will output m if $p(x) = 1$ and output $\perp$ otherwise. A predicate encryption system is secure if it provides semantic security for the pair (x, m) even if the adversary has a few benign secret keys (see Section 2.3).

The simplest example of predicate encryption is a system supporting the set of equality predicates, that is, predicates $p_{\mathsf{id}} : \Sigma \to \{0, 1\}$ defined as $p_{\mathsf{id}}(x) = 1$ iff $x = \mathsf{id}$. In such a system there is a secret key $\mathsf{sk}_{\mathsf{id}}$ for every $\mathsf{id} \in \Sigma$ and given the encryption c of a pair (x, m) the key $\mathsf{sk}_{\mathsf{id}}$ can decrypt c and recover m only when $x = \mathsf{id}$. It is easy to see that predicate encryption for the set of equality predicates is the same thing as (anonymous) identity-based encryption [BCOP04, ABC$^+$08].

Currently the most expressive collusion-resistant predicate encryption systems [KSW08, AFV11] support the family of inner product predicates: for a vector space $\Sigma = \mathbb{F}_q^\ell$ this is the set of predicates $p_v : \Sigma \to \{0, 1\}$ where $v \in \Sigma$ and $p_v(x) = 1$ iff $x \perp v$. This family of predicates includes the set of equality predicates and others.

Searching on encrypted data. Predicate encryption systems provide a general framework for searching on encrypted data. Consider a mail gateway whose function is to route incoming user email based on characteristics of the email. For example, emails from "boss" that are marked "urgent" are routed to the user's cell phone as are all emails from "spouse." All other emails are routed to the user's desktop. When the emails are transmitted in the clear the gateway's job is straight forward. However, when the emails are encrypted with the user's public key the gateway cannot see data needed for the routing decision. The simplest solution is to give the gateway the user's secret key, but this enables the gateway to decrypt all emails and exposes more information than the gateway needs.

A better solution is to encrypt emails using predicate encryption. The email header functions as the index x and the the routing instructions are used as m. The gateway is given a secret key sk_p corresponding to the "route to cell phone" predicate. This secret key enables the gateway to learn the routing instructions for messages satisfying the predicate p, but learn nothing else about emails.

Function privacy. A limitation of many existing predicate encryption systems is that the secret key sk_p reveals information about the predicate p. As a result, the gateway, and anyone else who has access to sk_p, learns the predicate p. Since in many practical settings it is important to keep the predicate p secret, our goal is to provide *function privacy*: sk_p should reveal as little information about p as possible.

At first glance it seems that hiding p is impossible: given sk_p the gateway can itself encrypt messages (x, m) and then apply sk_p to the resulting ciphertext. In doing so the gateway learns if $p(x) = 1$ which reveals some information about p. Nevertheless, despite this inherent limitation, function privacy can still be achieved.

Towards a solution. In recent work Boneh, Raghunathan, and Segev [BRS13] put forward a new notion of function privacy and applied it to identity-based encryption systems (i.e. to predicate encryption supporting equality predicates). They observe that if the identity id is chosen from a distribution with super-logarithmic min-entropy then the inherent limitation above is not a problem since the attacker cannot learn id from $\mathsf{sk}_{\mathsf{id}}$ by a brute force search since there are too many potential identities to test. They define function privacy for IBE systems by requiring that when id has sufficient min-entropy then $\mathsf{sk}_{\mathsf{id}}$ is indistinguishable from a secret key derived for an independently and uniformly distributed identity. This enables function private keyword searching on encrypted data. They then construct several IBE systems supporting function-private keyword searching.

While Boneh et al. [BRS13] showed how to achieve function privacy for equality predicates, encrypted search typically requires a richer set of searching predicates, including conjunctions, disjunctions, and many others. The authors left open the important question of achieving function privacy for a larger family of predicates.

Our contributions. In this paper we extend the framework and techniques of Boneh et al. [BRS13] for constructing function-private encryption schemes. We put forward a generalization of inner-product predicate encryption [KSW08, Fre10, AFV11], which we denote subspace-membership encryption, and present a definitional framework for capturing its function privacy. Our framework identifies the minimal restrictions under which a strong and meaningful notion of function privacy can be obtained for subspace-membership encryption schemes.

Then, we present a generic construction of a *function-private* subspace-membership encryption scheme based on any underlying inner-product encryption scheme (even when the underlying scheme is *not* function private). Our construction is efficient, and in addition to providing function privacy, it preserves the security properties of the underlying scheme. Finally, we present a generic construction of a function-privacy identity-based encryption scheme based on any underlying function-private subspace-membership encryption scheme.

These are the first generic constructions of function-private encryption schemes based on non-function-private ones. Recall that even for the case of identity-based encryption, Boneh et al. [BRS13] were not able to provide a generic construction, and had to individually modify various existing schemes.

1.1 Overview of Our Contributions

A subspace-membership encryption scheme is a predicate encryption scheme supporting subspace-membership predicates. That is, an encryption of a message is associated with an attribute $\mathbf{x} \in \mathbb{S}^{\ell}$, and secret keys are derived for subspaces defined by all vectors in $\mathbb{S}^{\ell}$ orthogonal to a matrix $\mathbf{W} \in \mathbb{S}^{m\times\ell}$ (for integers $m, \ell \in \mathbb{N}$ and an additive group $\mathbb{S}$).[1] Decryption recovers the message iff $\mathbf{W}\cdot\mathbf{x} = \mathbf{0}$. We refer the reader to Section 2.3 for the standard definitions of the functionality and data security of predicate encryption (following [KSW08, AFV11]).

Function privacy for subspace-membership encryption. Our goal is to design subspace-membership encryption schemes in which a secret key, $\mathsf{sk}_{\mathbf{W}}$, does not reveal any information, beyond the absolute minimum necessary, on the matrix $\mathbf{W}$. Formalizing a realistic notion of function privacy, however, is not straightforward due to the actual functionality of subspace-membership encryption encryption. Specifically, assuming that an adversary who is given a secret key $\mathsf{sk}_{\mathbf{W}}$ has some a-priori information that the matrix $\mathbf{W}$ belongs to a small set of matrices (e.g., $\{\mathbf{W}_0, \mathbf{W}_1\}$), then the adversary may be able to fully recover $\mathbf{W}$: The adversary simply needs to encrypt a (possibly random) message m for some attribute $\mathbf{x}$ that is orthogonal to $\mathbf{W}_0$ but not to $\mathbf{W}_1$, and then run the decryption algorithm on the given secret key $\mathsf{sk}_{\mathbf{W}}$ and the resulting ciphertext to identify the one that decrypts correctly. In fact, as in [BRS13], as long as the adversary has some a-priori

[1]Note that by setting $m = 1$ one obtains the notion of an inner-product encryption scheme [KSW08, Fre10, AFV11].

information according to which the matrix $\mathbf{W}$ is sampled from a distribution whose min-entropy is at most logarithmic in the security parameter, there is a non-negligible probability for a full recovery.

In the setting of subspace-membership encryption (unlike that of identity-based encryption [BRS13]), however, the requirement that $\mathbf{W}$ is sampled from a source of high min-entropy does not suffice for obtaining a meaningful notion of function privacy. In Section 3 we show that even if $\mathbf{W}$ has nearly full min-entropy, but two of its columns may be correlated, then a meaningful notion of function privacy is not within reach.

In this light, our notion of function privacy for subspace-encryption schemes focuses on secret key $\mathsf{sk}_{\mathbf{W}}$ for which the columns of $\mathbf{W}$ form a block source. That is, each column of $\mathbf{W}$ should have a reasonable amount of min-entropy even given all previous columns. Our notion of function privacy requires that such a secret key $\mathsf{sk}_{\mathbf{W}}$ (where $\mathbf{W}$ is sampled from an *adversarially*-chosen distribution) be indistinguishable from a secret key for a subspace chosen uniformly at random.

A function-private construction from inner-product encryption. Given any underlying inner-product encryption scheme we construction a function-private subspace-membership encryption scheme quite naturally. We modify the key-generation algorithm as follows: for generating a secret key for a subspace described by $\mathbf{W}$, we first sample a uniform $\mathbf{s} \leftarrow \mathbb{S}^m$ and use the key-generation algorithm of the underlying scheme for generating a secret key for the vector $\mathbf{v} = \mathbf{W}^{\mathsf{T}}\mathbf{s}$. Observe that as long as the columns of $\mathbf{W}$ form a block source, then the leftover hash lemma for block sources guarantees that $\mathbf{v}$ is statistically close to uniform. In particular, essentially no information on $\mathbf{W}$ is revealed.

We also observe that extracting from the columns of $\mathbf{W}$ using the same seed for the extractor $\langle \mathbf{s}, \cdot \rangle$ interacts nicely with the subspace-membership functionality. Indeed, if $\mathbf{W} \cdot \mathbf{x} = \mathbf{0}$, it holds that $\mathbf{v}^{\mathsf{T}}\mathbf{x} = 0$ and vice-versa with high probability. We note that the case where the attribute set is small requires some additional refinement that we omit from this overview, and we refer the reader to Section 4 for more details.

1.2 Related Work

As discussed above, the notion of function privacy was recently put forward by Boneh, Raghunathan, and Segev [BRS13]. One of the main motivations of Boneh et al. was that of designing public-key searchable encryption schemes [BCOP04, GSW04, ABC+08, BW07, SBC+07, KSW08, BSNS08, CKRS09, ABN10, AFV11] that are keyword private. That is, public-key searchable encryption schemes in which search tokens hide, as much as possible, their corresponding predicates. They presented a framework for modeling function privacy, and constructed various function-private anonymous identity-based encryption schemes (which, in particular, imply public-key keyword-private searchable encryption schemes).

More generally, the work of Boneh et al. initiated the study of function privacy in functional encryption [BSW11, O'N10, BO12, GVW12, AGVW13, GKP+13], where a functional secret key sk_f corresponding to a function f enables to compute $f(m)$ given an encryption $c = \mathsf{Enc}_{\mathsf{pk}}(m)$. Intuitively, in this setting function privacy guarantees that a functional secret key sk_f does not reveal information about f beyond what is already known and what can be obtained by running the decryption algorithm on test ciphertexts. In [BRS13], the authors also discuss connections of function privacy to program obfuscation.

Our notion of subspace-membership encryption generalizes that of inner-product encryption introduced by Katz, Sahai, and Waters [KSW08]. They defined and constructed predicate encryption schemes for predicates corresponding to inner products over $\mathbb{Z}_N$ (for some large N). Informally, this class of predicates corresponds to functions $f_{\mathbf{v}}$ where $f_{\mathbf{v}}(\mathbf{x}) = 1$ if and only if $\langle \mathbf{v}, \mathbf{x} \rangle = 0$. Subsequently, Freeman [Fre10] modified their construction to inner products over groups of prime order

p, and Agrawal, Freeman, and Vaikuntanathan [AFV11] constructed an inner-product encryption scheme over $\mathbb{Z}_p$ for a small prime p. Other results on inner product encryption study adaptive security [OT12], delegation in the context of hierarchies [OT09], and generalized IBE [BH08].

Finally, we note that function privacy in the symmetric-key setting, where the encryptor and decryptor have a shared secret key, was studied by Shen, Shi, and Waters [SSW09]. They designed a function-private inner-product encryption scheme. As noted by Boneh et al. [BRS13], achieving function privacy in the public-key setting is a more subtle task due to the inherent conflict between privacy and functionality.

1.3 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we introduce standard notation, definitions, and tools. In Section 3 we introduce the notions of subspace-membership encryption and function privacy for subspace-membership encryption. In Section 4 we present generic constructions of function-private subspace-membership encryption schemes based on any inner-product encryption scheme. In Section 5 we show that function-private subspace-membership encryption implies, in particular, function-private identity-based encryption. Finally, in Section 6 we discuss several open problems that arise from this work.

2 Preliminaries

2.1 Notation

For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$, and by U_n the uniform distribution over the set $\{0,1\}^n$. For a random variable X we denote by $x \leftarrow X$ the process of sampling a value x according to the distribution of X. Similarly, for a finite set S we denote by $x \leftarrow S$ the process of sampling a value x according to the uniform distribution over S. We denote by $\mathbf{x}$ (and sometimes $\boldsymbol{x}$) a vector $(x_1, \ldots, x_{|\mathbf{x}|})$. We denote by $\mathbf{X} = (X_1, \ldots, X_T)$ a joint distribution of T random variables. A non-negative function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if it vanishes faster than any inverse polynomial. A non-negative function $f : \mathbb{N} \to \mathbb{R}$ is *super-polynomial* if it grows faster than any polynomial.

The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. A *$k$-source* is a random variable X with $\mathbf{H}_\infty(X) \geq k$. A *$(T, k)$-block source* is a random variable $\mathbf{X} = (X_1, \ldots, X_T)$ where for every $i \in [T]$ and $x_1, \ldots, x_{i-1}$ it holds that $\mathbf{H}_\infty(X_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) \geq k$. The *statistical distance* between two random variables X and Y over a finite domain Ω is $\mathbf{SD}(X, Y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. Two random variables X and Y are *δ-close* if $\mathbf{SD}(X, Y) \leq \delta$. Two distribution ensembles $\{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $\{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are *statistically indistinguishable* if it holds that $\mathbf{SD}(X_\lambda, Y_\lambda)$ is negligible in λ. They are *computationally indistinguishable* if for every probabilistic polynomial-time algorithm $\mathcal{A}$ it holds that $\left|\Pr\left[\mathcal{A}(1^\lambda, x) = 1\right] - \Pr\left[\mathcal{A}(1^\lambda, y) = 1\right]\right|$ is negligible in λ, where $x \leftarrow X_\lambda$ and $y \leftarrow Y_\lambda$.

2.2 The Leftover Hash Lemma

Definition 2.1. A collection $\mathcal{H}$ of functions $H : U \to V$ is *universal* if for any $x_1, x_2 \in U$ such that $x_1 \neq x_2$ it holds that $\Pr_{H \leftarrow \mathcal{H}}[H(x_1) = H(x_2)] = 1/|V|$.

Lemma 2.2 (Leftover hash lemma for block sources [CG88, HILL99, Zuc96, CV08])**.** *Let $\mathcal{H}$ be a universal collection of functions $H : U \to V$, and let $\mathbf{X} = (X_1, \ldots, X_\ell)$ be an (ℓ, k)-block-source where $k \geq \log|V| + 2\log(1/\epsilon) + \Theta(1)$. Then, the distribution $(H, H(X_1), \ldots, H(X_\ell))$, where $H \leftarrow \mathcal{H}$, is $\epsilon\ell$-close to the uniform distribution over $\mathcal{H} \times V^\ell$.*

2.3 Predicate Encryption

We use the definition of Katz, Sahai, and Waters [KSW08], which is based on the definition of *searchable encryption* proposed in [BCOP04, BW07].

Definition 2.3 ([KSW08, Def. 2.1])**.** A (key-policy) predicate encryption scheme for the class of predicates $\mathcal{F}$ over the set of attributes Σ consists of four randomized PPT algorithms Setup, KeyGen, Enc, and Dec such that:

1. **Setup:** Setup takes as input the security parameter 1^λ and outputs public parameters pp and a master secret key msk.
2. **Key generation:** KeyGen takes as input the master secret key msk and a predicate $f \in \mathcal{F}$ and outputs a key sk_f.
3. **Encryption:** Enc takes as input the public key pp, an attribute $I \in \Sigma$, and a message M in some associated message space $\mathcal{M}$. It returns a ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pp}, I, \mathsf{M})$.
4. **Decryption:** Dec takes as input a secret key sk_f and ciphertext c. It outputs either M or $\perp$.

Correctness requires that for all $\lambda \in \mathbb{N}$, for all $(\mathsf{pp}, \mathsf{msk})$ generated by $\mathsf{Setup}(1^\lambda)$, for all $f \in \mathcal{F}$, for all keys $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$, for all $I \in \Sigma$:

- If $f(I) = 1$, then $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{Enc}(\mathsf{pp}, I, \mathsf{M})) = \mathsf{M}$.
- If $f(I) = 0$, then $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{Enc}(\mathsf{pp}, I, \mathsf{M})) = \perp$ with all but negligible probability in λ.

There are several notions of security for predicate encryption schemes. The most basic is *payload hiding*, which guarantees that no efficient adversary can obtain any information about the encrypted message, but allows information about the attributes to be revealed. A stronger notion is *attribute hiding*, which guarantees in addition that no efficient adversary can obtain information about the attribute associated with a ciphertext. We consider two definitions, attribute hiding and weak atribute hiding following the work of Katz, Sahai, and Waters [KSW08] and Agrawal, Freeman, and Vaikuntanathan [AFV11].

Definition 2.4 ([KSW08, AFV11])**.** A predicate encryption scheme Π for the class of predicates $\mathcal{F}$ over the set of attributes Σ is *attribute hiding* if for all probabilistic polynomial-time adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in distinguishing the experiments $\mathsf{Expt}^{(0)}_{\mathsf{AH},\Pi,\mathcal{A}}(\lambda)$ and $\mathsf{Expt}^{(1)}_{\mathsf{AH},\Pi,\mathcal{A}}(\lambda)$ is negligible in the security parameter λ, where for each $b \in \{0,1\}$ the experiment $\mathsf{Expt}^{(b)}_{\mathsf{AH},\Pi,\mathcal{A}}$ is defined as follows:

1. $\mathcal{A}(1^\lambda)$ outputs a pair $(I_0, I_1) \in \Sigma$.
2. $\mathsf{Setup}(1^\lambda)$ is run to generate $(\mathsf{pp}, \mathsf{msk})$ and the adversary is given pp.
3. $\mathcal{A}$ (adaptively) requests keys for predicates $f_1, \ldots, f_Q \in \mathcal{F}$ subject to the restriction $f_i(I_0) = f_i(I_1)$ for every $i \in [Q]$. In response to each query, $\mathcal{A}$ receives $\mathsf{sk}_{f_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f_i)$.
4. $\mathcal{A}$ outputs two equal-length messages $\mathsf{M}_0, \mathsf{M}_1 \in \mathcal{M}$. If there exists $i \in [Q]$ such that $f_i(I_0) = f_i(I_1) = 1$ then it must hold that $\mathsf{M}_0 = \mathsf{M}_1$. The adversary $\mathcal{A}$ receives ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pp}, I_b, m_b)$.
5. $\mathcal{A}$ (adaptively) requests additional keys subject to the same restrictions as before.
6. $\mathcal{A}$ outputs a guess b'. The experiment outputs this bit b'.

The advantage of adversary $\mathcal{A}$ is defined as follows:

$$\mathbf{Adv}^{\mathsf{AH}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{AH},\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{AH},\Pi,\mathcal{A}} = 1\right] \right|.$$

A predicate encryption scheme Π is said to be *weakly attribute hiding* if the adversary $\mathcal{A}$, in step (3) is restricted to query secret keys for predicates f_i with $f_i(I_0) = f_i(I_1) = 0$. The experiments $\mathsf{Expt}^{(b)}_{\mathsf{wAH},\Pi,\mathcal{A}}(\lambda)$ for $b \in \{0,1\}$ and advantage $\mathbf{Adv}^{\mathsf{wAH}}_{\Pi,\mathcal{A}}(\lambda)$ are defined in an analogous manner.

2.4 Identity-Based Encryption

An identity-based encryption (IBE) scheme [Sha84, BF03] is a quadruple $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of probabilistic polynomial-time algorithms. The setup algorithm, Setup, takes as input the security parameter 1^λ and outputs the public parameters pp of the scheme together with a corresponding master secret key msk. The encryption algorithm, Enc, takes as input the public parameters pp, an identity id, and a message m, and outputs a ciphertext $c = \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m)$. The key-generation algorithm, KeyGen, takes as input the master secret key msk and an identity id, and outputs a secret key $\mathsf{sk_{id}}$ corresponding to id. The decryption algorithm, Dec, takes as input the public parameters pp, a ciphertext c, and a secret key $\mathsf{sk_{id}}$, and outputs either a message m or the symbol $\perp$. For such a scheme we denote by $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ its identity space and message space, respectively.

Functionality. In terms of functionality, we require that the decryption algorithm is correct with all but a negligible probability. Specifically, for any security parameter $\lambda \in \mathbb{N}$, for any identity $\mathsf{id} \in \mathcal{ID}_\lambda$, and for any message $m \in \mathcal{M}_\lambda$ it holds that

$$\mathsf{Dec}(\mathsf{pp}, \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}), \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m)) = m$$

with probably at least $1 - \nu(\lambda)$ for a negligible function $\nu(\cdot)$, where the probability it taken over the internal randomness of the algorithm Setup, KeyGen, Enc, and Dec.

Data privacy. We consider the standard selective notion of anonymity and message indistinguishability under a chosen-identity adaptive-chosen-plaintext attack known as $\mathsf{anon\text{-}IND\text{-}sID\text{-}CPA}$ and abbreviated to sDP in the rest of the paper.

Definition 2.5 (Selective data privacy – $\mathsf{anon\text{-}IND\text{-}sID\text{-}CPA}$)**.** An identity-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ over a identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ is *selective data private* if for any probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

$$\mathbf{Adv}^{\mathsf{sDP}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{sDP},\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{sDP},\Pi,\mathcal{A}}(\lambda) = 1\right]\right| \leq \nu(\lambda),$$

where for each $b \in \{0,1\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{(b)}_{\mathsf{sDP},\Pi,\mathcal{A}}(\lambda)$ is defined as follows:

1. $(\mathsf{id}_0^*, \mathsf{id}_1^*, \mathsf{state}_1) \leftarrow \mathcal{A}(1^\lambda)$, where $\mathsf{id}_0^*, \mathsf{id}_1^* \in \mathcal{ID}_\lambda$.
2. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
3. $(m_0^*, m_1^*, \mathsf{state}_2) \leftarrow \mathcal{A}(\mathsf{state}_1)$, where $m_0^*, m_1^* \in \mathcal{M}_\lambda$.
4. $c^* \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_b^*, m_b^*)$.
5. $b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(c^*, \mathsf{state}_2)$, where $b' \in \{0,1\}$.
6. Denote by $\mathcal{S}$ the set of identities with which $\mathcal{A}$ queried $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$.
7. If $\mathcal{S} \cap \{\mathsf{id}_0^*, \mathsf{id}_1^*\} = \emptyset$ then output b', and otherwise output $\perp$.

Function Privacy. We consider the notion of function privacy introduced by Boneh, Raghunathan, and Segev [BRS13]. A function-private IBE scheme informally requires that no adversary learn anything about id from the secret key $\mathsf{sk_{id}}$ beyond the absolute minimum necessary.

Definition 2.6 (Real-or-random function-privacy oracle)**.** The real-or-random function-privacy oracle $\mathsf{RoR^{FP}}$ takes as input triplets of the form $(\mathsf{mode}, \mathsf{msk}, \mathbf{ID})$, where $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, msk is a master secret key, and $\mathbf{ID} = (ID_1, \dots, ID_T) \in \mathcal{ID}^T$ is a circuit representing a joint distribution over $\mathcal{ID}^T$. If $\mathsf{mode} = \mathsf{real}$ then the oracle samples $(\mathsf{id}_1, \dots, \mathsf{id}_T) \leftarrow \mathbf{ID}$ and if $\mathsf{mode} = \mathsf{rand}$ then the oracle samples $(\mathsf{id}_1, \dots, \mathsf{id}_T) \leftarrow \mathcal{ID}^T$ uniformly. It then invokes the algorithm $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ on each of $\mathsf{id}_1, \dots, \mathsf{id}_T$ and outputs a vector of secret keys $(\mathsf{sk}_{\mathsf{id}_1}, \dots, \mathsf{sk}_{\mathsf{id}_T})$.

Definition 2.7 (Function-privacy adversary)**.** A (T, k)-block-source function-privacy adversary $\mathcal{A}$ is an algorithm that is given as input a pair $(1^\lambda, \mathsf{pp})$ and oracle access to $\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode}, \mathsf{msk}, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, and to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$, and each of its queries to $\mathsf{RoR}^{\mathsf{FP}}$ is a (T, k)-block-source.

Definition 2.8 (IBE Function privacy)**.** An identity-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is (T, k)-*source function private* if for any probabilistic polynomial-time (T, k)-source function-privacy adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

$$\mathbf{Adv}^{\mathsf{FP\text{-}IBE}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP\text{-}IBE},\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP\text{-}IBE},\Pi,\mathcal{A}}(\lambda) = 1\right] \right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi,\mathcal{A}}(\lambda)$ is defined as follows:

1. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $b \leftarrow \mathcal{A}^{\mathsf{RoR}^{\mathsf{FP\text{-}IBE}}(\mathsf{mode},\mathsf{msk},\cdot),\mathsf{KeyGen}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{pp})$.
3. Output b.

In addition, such a scheme is *statistically* (T, k)-source function private if the above holds for any *computationally-unbounded* (T, k)-source enhanced function-privacy adversary making a polynomial number of queries to the $\mathsf{RoR}^{\mathsf{FP\text{-}IBE}}$ oracle.

3 Subspace-Membership Encryption and Its Function Privacy

In this section we formalize the notion of subspace-membership encryption and its function privacy within the framework of Boneh, Raghunathan and Segev [BRS13]. A subspace-membership encryption scheme is a predicate encryption scheme [BW07, KSW08] supporting the class of predicates $\mathcal{F}$, over an attribute space $\Sigma = \mathbb{S}^\ell$, defined as

$$\mathcal{F} = \left\{ f_{\mathbf{W}} : \mathbf{W} \in \mathbb{S}^{m \times \ell} \right\} \qquad \text{with} \qquad f_{\mathbf{W}}(\mathbf{x}) = \begin{cases} 1 & \mathbf{W} \cdot \mathbf{x} = \mathbf{0} \in \mathbb{S}^m \\ 0 & \text{otherwise} \end{cases}$$

for integers $m, \ell \in \mathbb{N}$, and an additive group $\mathbb{S}$. Informally, in a subspace-membership encryption, an encryption of a message is associated with an attribute $\mathbf{x} \in \mathbb{S}^\ell$, and secret keys are derived for subspaces defined by all vectors in $\mathbb{S}^\ell$ orthogonal to a matrix $\mathbf{W} \in \mathbb{S}^{m \times \ell}$. Decryption recovers the message if and only if $\mathbf{W} \cdot \mathbf{x} = \mathbf{0}$. We refer the reader to Section 2.3 for the standard definitions of the functionality and data security of predicate encryption (following [KSW08, AFV11]). Subspace-membership encryption with delegation was also studied in [OT09, OT12]. Here we do not need the delegation property.

Based on the framework introduced by Boneh, Raghunathan, and Segev [BRS13], our notion of function privacy for subspace-membership encryption considers adversaries that are given the public parameters of the scheme and can interact with a "real-or-random" function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ defined as follows, and with a key-generation oracle.

Definition 3.1 (Real-or-random function-privacy oracle)**.** The real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ takes as input triplets of the form $(\mathsf{mode}, \mathsf{msk}, V)$, where $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, msk is a master secret key, and $V = (V_1, \dots, V_\ell) \in \mathbb{S}^{m \times \ell}$ is a circuit representing a joint distribution over $\mathbb{S}^{m \times \ell}$ (i.e., each V_i is a distribution over $\mathbb{S}^m$). If $\mathsf{mode} = \mathsf{real}$ then the oracle samples $\mathbf{W} \leftarrow V$ and if $\mathsf{mode} = \mathsf{rand}$ then the oracle samples $\mathbf{W} \leftarrow \mathbb{S}^{m \times \ell}$ uniformly. It then invokes the algorithm $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ on $\mathbf{W}$ for outputting a secret key $\mathsf{sk}_{\mathbf{W}}$.

Definition 3.2 (Function-privacy adversary)**.** An (ℓ, k)-block-source function-privacy adversary $\mathcal{A}$ is an algorithm that is given as input a pair $(1^\lambda, \mathsf{pp})$ and oracle access to $\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode}, \mathsf{msk}, \cdot)$ for some $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, and to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$. It is required that each of $\mathcal{A}$'s queries to $\mathsf{RoR}^{\mathsf{FP}}$ be an (ℓ, k)-block-source.

Definition 3.3 (Function-private subspace-membership encryption)**.** A subspace-membership encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is (ℓ, k)*-block-source function private* if for any probabilistic polynomial-time (ℓ, k)-block-source function-privacy adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that

$$\mathbf{Adv}^{\mathsf{FP}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\Pi,\mathcal{A}}(\lambda) = 1\right] \right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{mode}}_{\mathsf{FP},\Pi,\mathcal{A}}(\lambda)$ is defined as follows:

1. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $b \leftarrow \mathcal{A}^{\mathsf{RoR}^{\mathsf{FP}}(\mathsf{mode},\mathsf{msk},\cdot),\mathsf{KeyGen}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{pp})$.
3. Output b.

In addition, such a scheme is *statistically* (ℓ, k)-block-source function private if the above holds for any *computationally-unbounded* (ℓ, k)-block-source function-privacy adversary making a polynomial number of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle.

Multi-shot vs. single-shot adversaries. Note that Definition 3.3 considers adversaries that query the function-privacy oracle for any polynomial number of times. In fact, as adversaries are also given access to the key-generation oracle, this "multi-shot" definition is polynomially equivalent to its "single-shot" variant in which adversaries query the real-or-random function-privacy oracle $\mathsf{RoR}^{\mathsf{FP}}$ at most once. This is proved via a straightforward hybrid argument, where the hybrids are constructed such that only one query is forwarded to the function-privacy oracle, and all other queries are answered using the key-generation oracle.

The block-source requirement on the columns of W. Our definition of function privacy for subspace-membership encryption requires that a secret key $\mathsf{sk}_{\mathbf{W}}$ reveals no unnecessary information about $\mathbf{W}$ as long as the columns of $\mathbf{W}$ form a block source (i.e., each column is unpredictable even given the previous columns). One might consider a stronger definition, in which the columns of $\mathbf{W}$ may be arbitrarily correlated, as long as each column of $\mathbf{W}$ is sufficiently unpredictable. Such a definition, however, is impossible to satisfy.

Specifically, consider the special case of inner-product encryption (i.e., $m = 1$), and an adversary that queries the real-or-random oracle with a distribution over vectors $\mathbf{w} \in \mathbb{S}^\ell$ defined as follows: sample $\ell - 1$ independent and uniform values $u_1, \ldots, u_{\ell-1} \leftarrow \mathbb{S}$ and output $\mathbf{w} = (u_1, 2u_1, u_2, \ldots, u_{\ell-1})$. Such a distribution clearly has high min-entropy (specifically, $(\ell - 1) \log |\mathbb{S}|$ bits), and each coordinate of $\mathbf{w}$ has min-entropy $\log |\mathbb{S}|$ bits. However, secret keys for vectors drawn from this distribution can be easily distinguished from secret keys for vectors drawn from the uniform distribution over $\mathbb{S}^\ell$: encrypt a message M to the attribute $\mathbf{x} = (-2, 1, 0, \ldots, 0) \in \mathbb{S}^\ell$ and check to see if decryption succeeds in recovering M. For a random vector $\mathbf{w} \in \mathbb{S}^\ell$ the decryption succeeds only with probability $1/|\mathbb{S}|$ giving the adversary an overwhelming advantage.

Therefore, restricting function privacy adversaries to query the $\mathsf{RoR}^{\mathsf{FP}}$ oracle only with sources whose columns form block sources is essential for achieving a meaningful notion of function privacy.

On correlated $\mathsf{RoR}^{\mathsf{FP}}$ queries. In Definition 3.2 we consider adversaries that receives only a single secret key $\mathsf{sk}_{\mathbf{W}}$ for each query to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. Our definition easily generalizes to include adversaries that are allowed to query the $\mathsf{RoR}^{\mathsf{FP}}$ oracle with *correlated* queries. More specifically, an adversary can receive secret keys $\mathsf{sk}_{\mathbf{W}_1}, \ldots, \mathsf{sk}_{\mathbf{W}_T}$ for any parameter T that is polynomial in the security parameter. The $\mathsf{RoR}^{\mathsf{FP}}$ oracle samples subspaces $\mathbf{W}_1, \ldots, \mathbf{W}_T$ from an adversarially chosen joint distribution over $\left(\mathbb{S}^{m \times \ell}\right)^T$ with the restriction that for every $1 \leq i \leq T$, the columns of $\mathbf{W}_i$ come from a (ℓ, k)-block-source even conditioned on any fixed values for $\mathbf{W}_1, \ldots, \mathbf{W}_{i-1}$.[2]

Function privacy of existing inner-product encryption schemes. The inner-product predicate encryption scheme from lattices [AFV11] is trivially not function private as the secret key includes the corresponding function $f_{\mathbf{v}}$ as part of it (this is necessary for the decryption algorithm to work correctly). The scheme constructed from bilinear groups with composite order [KSW08] however presents no such obvious attack, but we were not able to prove its function privacy based on any standard cryptographic assumption.

4 A Generic Construction Based on Inner-Product Encryption

In this section we present a generic construction of a function-private subspace-membership encryption scheme starting from any inner-product encryption scheme. In Section 4.1 we consider the case of a large attribute space $\mathbb{S}$ (of size super-polynomial in the security parameter), and in Section 4.2 we extend our construction to the case of a smaller attribute space $\mathbb{S}$.

4.1 Large Attribute Space

Our construction. Let $\mathcal{IP} = (\mathsf{IP.Setup}, \mathsf{IP.KeyGen}, \mathsf{IP.Enc}, \mathsf{IP.Dec})$ be an inner-product encryption scheme with attribute set $\Sigma = \mathbb{S}^{\ell}$. We construct a subspace-membership encryption scheme $\mathcal{SM} = (\mathsf{SM.Setup}, \mathsf{SM.KeyGen}, \mathsf{SM.Enc}, \mathsf{SM.Dec})$ as follows.

- **Setup:** $\mathsf{SM.Setup}$ is identical to $\mathsf{IP.Setup}$. On input the security parameter it outputs public parameters pp and the master secret key msk by running $\mathsf{IP.Setup}(1^{\lambda})$.

- **Key generation:** $\mathsf{SM.KeyGen}$ takes as input the master secret key msk and a function $f_{\mathbf{W}}$ where $\mathbf{W} \in \mathbb{S}^{m \times \ell}$ and proceeds as follows. It samples uniform $\mathbf{s} \leftarrow \mathbb{S}^m$ and computes $\mathbf{v} = \mathbf{W}^{\mathsf{T}}\mathbf{s} \in \mathbb{S}^{\ell}$. Next, it computes $\mathsf{sk}_{\mathbf{v}} \leftarrow \mathsf{IP.KeyGen}(\mathsf{msk}, \mathbf{v})$ and outputs $\mathsf{sk}_{\mathbf{W}} \stackrel{\mathsf{def}}{=} \mathsf{sk}_{\mathbf{v}}$.

- **Encryption:** $\mathsf{SM.Enc}$ is identical to $\mathsf{IP.Enc}$. On input the public parameters, an attribute $\mathbf{x} \in \mathbb{S}^{\ell}$, and a message M, it outputs a ciphertext $c \leftarrow \mathsf{IP.Enc}(\mathsf{pp}, \mathbf{x}, \mathsf{M})$.

- **Decryption:** $\mathsf{SM.Dec}$ is identical to $\mathsf{IP.Dec}$. On input the public parameters pp, a secret key $\mathsf{sk}_{\mathbf{W}}$, and a ciphertext c, the algorithm outputs $\mathsf{M} \leftarrow \mathsf{IP.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{W}}, c)$.

Correctness. Correctness of the construction follows from the correctness of the underlying inner-product encryption scheme. For every $\mathbf{W} \in \mathbb{S}^{m \times \ell}$ and every $\mathbf{x} \in \mathbb{S}^{\ell}$, it suffices to show the following:

- If $f(I) = 1$, then it holds that $\mathbf{W} \cdot \mathbf{x} = \mathbf{0}$. This implies $\mathbf{x}^{\mathsf{T}}\mathbf{v} = \mathbf{x}^{\mathsf{T}}\left(\mathbf{W}^{\mathsf{T}}\mathbf{s}\right) = 0$ and therefore $\mathsf{SM.Dec}$ correctly outputs M as required.

- If $f(I) = 0$, then it holds that $\mathbf{e} \stackrel{\mathsf{def}}{=} \mathbf{W} \cdot \mathbf{x} \neq \mathbf{0} \in \mathbb{S}^m$. As $\mathbf{x}^{\mathsf{T}}\mathbf{v} = \mathbf{x}^{\mathsf{T}}\left(\mathbf{W}^{\mathsf{T}}\mathbf{s}\right) = \mathbf{e}^{\mathsf{T}}\mathbf{s}$, for any $\mathbf{e} \neq \mathbf{0}$ the quantity $\mathbf{x}^{\mathsf{T}}\mathbf{v}$ is zero with probability $1/|\mathbb{S}|$ over choices of $\mathbf{s}$. As $1/|\mathbb{S}|$ is negligible in λ whenever $|\mathbb{S}|$ is super-polynomial in λ, the proof of correctness follows.

[2] Or equivalently, the columns of $[\ \mathbf{W}_1 \mid \mathbf{W}_2 \mid \cdots \mid \mathbf{W}_T\ ]$ are distributed according to a $(T\ell, k)$-block-source.

Security. We state the following theorem about the security of our construction.

Theorem 4.1. *If $\mathcal{IP}$ is an attribute hiding (resp. weakly attribute hiding) inner-product encryption scheme for an attribute set $\mathbb{S}$ of size super-polynomial in the security parameter, then it holds that:*

1. *The scheme $\mathcal{SM}$ is an attribute hiding (resp. weakly attribute hiding) subspace-membership encryption scheme under the same assumption as the security of the underlying inner-product encryption scheme.*

2. *The scheme $\mathcal{SM}$ when $m \geq 2$ is statistically function private for (ℓ, k)-block-sources for any $\ell = \mathrm{poly}(\lambda)$ and $k \geq \log |\mathbb{S}| + \omega(\log \lambda)$.*

Proof. We first prove the attribute-hiding property of the scheme, and then prove its function privacy.

Attribute hiding. Attribute-hiding property of $\mathcal{SM}$ follows from the attribute-hiding property of $\mathcal{IP}$ in a rather straightforward manner. Given a challenger for the attribute-hiding property of $\mathcal{IP}$, an $\mathcal{SM}$ adversary $\mathcal{A}$ can be simulated by algorithm $\mathcal{B}$ as follows: $\mathcal{A}$'s challenge attributes are forwarded to the $\mathcal{IP}$-challenger and the resulting public parameterers are published. Secret key queries can be simulated by first sampling uniform $\mathbf{s} \leftarrow \mathbb{S}^m$, then computing $\mathbf{v} = \mathbf{W}^\intercal \mathbf{s}$ and forwarding $\mathbf{v}$ to the $\mathcal{IP}$ key generation oracle. Similarly, the challenge messages from the adversary are answered by forwarding them to the challenger. The details are as follows.

Let $\mathsf{X} \in \{\mathsf{AH}, \mathsf{wAH}\}$. Given an adversary $\mathcal{A}$ that makes Q secret key queries in total and has a non-negligible advantage $\mathbf{Adv}^{\mathsf{X}}_{\mathcal{SM},\mathcal{A}}(\lambda)$ (see Definition 2.3) we construct an adversary $\mathcal{B}$ that interacts with an inner-product encryption attribute hiding challenger with advantage $\mathbf{Adv}^{\mathsf{X}}_{\mathcal{IP},\mathcal{B}}(\lambda) \approx \mathbf{Adv}^{\mathsf{X}}_{\mathcal{SM},\mathcal{A}}(\lambda)$ as follows.

Adversary $\mathcal{A}$ outputs a pair of attributes $\mathbf{x}_0$ and $\mathbf{x}_1$ and $\mathcal{B}$ forwards them to the $\mathcal{IP}$-challenger. $\mathcal{B}$ receives pp (but not msk) and forwards pp to the adversary. For $i \in [Q]$, on the i^{th} KeyGen query $\mathbf{W}_i$ from $\mathcal{A}$, algorithm $\mathcal{B}$ samples a random $\mathbf{s}_i \leftarrow \mathbb{S}^m$ and computes $\mathbf{v}_i = \mathbf{W}_i{}^\intercal \mathbf{s}_i$. It forwards $\mathbf{v}_i$ to the KeyGen oracle provided by the $\mathcal{IP}$-challenger and receives $\mathsf{sk}_{\mathbf{v}_i} = \mathsf{sk}_{\mathbf{W}_i}$. The algorithm $\mathcal{B}$ answers $\mathcal{A}$'s KeyGen query with $\mathsf{sk}_{\mathbf{W}_i}$.

$\mathcal{A}$ outputs two messages M_0 and M_1. If there exists an $i \in [Q]$ such that $\mathbf{v}_i{}^\intercal \mathbf{x}_0 = 0$ or $\mathbf{v}_i{}^\intercal \mathbf{x}_1 = 0$, the algorithm $\mathcal{B}$ aborts and outputs a uniform bit. Otherwise, it forwards M_0 and M_1 to the $\mathcal{IP}$-challenger and receives a challenge ciphertext c which it forwards to $\mathcal{A}$. Finally, $\mathcal{B}$ receives a guess b from $\mathcal{A}$ and outputs the bit b.

Observe that the algorithm $\mathcal{B}$ simulates the adversary queries honestly. For $b \in \{0, 1\}$, let $\mathsf{E}^{(b)}_{\mathcal{IP}}$ denote the event $\left[\mathsf{Expt}^{(b)}_{\mathsf{X},\mathcal{IP},\mathcal{B}}(\lambda) = 1\right]$ and let $\mathsf{E}^{(b)}_{\mathcal{SM}}$ denote the event $\left[\mathsf{Expt}^{(b)}_{\mathsf{X},\mathcal{SM},\mathcal{A}}(\lambda) = 1\right]$. Let Abort denote the event that $\mathcal{B}$ aborts (for either $b \in \{0, 1\}$, as the abort condition is independent of the bit b) and outputs a uniform bit. Therefore,

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{X}}_{\mathcal{IP},\mathcal{B}}(\lambda) &\stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{X},\mathcal{IP},\mathcal{B}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{X},\mathcal{IP},\mathcal{B}} = 1\right]\right| \\
&= \left|\Pr\left[\mathsf{E}^{(0)}_{\mathcal{IP}}\right] - \Pr\left[\mathsf{E}^{(1)}_{\mathcal{IP}}\right]\right| \\
&\geq \left|\Pr\left[\mathsf{E}^{(0)}_{\mathcal{IP}} \,\middle|\, \overline{\mathsf{Abort}}\right] - \Pr\left[\mathsf{E}^{(1)}_{\mathcal{IP}} \,\middle|\, \overline{\mathsf{Abort}}\right]\right| - \Pr[\mathsf{Abort}] && (4.1) \\
&\geq \left|\Pr\left[\mathsf{E}^{(0)}_{\mathcal{SM}}\right] - \Pr\left[\mathsf{E}^{(1)}_{\mathcal{SM}}\right]\right| - \Pr[\mathsf{Abort}] && (4.2) \\
&\geq \mathbf{Adv}^{\mathsf{X}}_{\mathcal{SM},\mathcal{A}}(\lambda) - \frac{2Q}{|\mathbb{S}|}. && (4.3)
\end{aligned}$$

Here, Equation (4.1) follows from a standard probability argument. Equation (4.2) follows from the fact that if $\mathcal{B}$ does not abort, the events $\mathsf{E}_{\mathcal{IP}}^{(b)}$ and $\mathsf{E}_{\mathcal{SM}}^{(b)}$ are identical. Equation (4.3) follows by bounding the probability that $\mathcal{B}$ aborts. $\Pr[\mathsf{Abort}]$ can be derived using the same argument used to show correctness: for every $i \in [Q]$, if $\mathbf{W}_i \cdot \mathbf{x}_0 \neq \mathbf{0}$, then $\mathbf{x}_0{}^\intercal \mathbf{v}_i = 0$ with probability at most $1/|\mathbb{S}|$ (and similarly with $\mathbf{x}_1$). The abort probability therefore follows from a straightforward union bound. As Q is polynomial in λ, $\mathbf{Adv}^{\mathsf{X}}_{\mathcal{IP},\mathcal{B}}(\lambda)$ remains non-negligible if $\mathbf{Adv}^{\mathsf{X}}_{\mathcal{SM},\mathcal{A}}(\lambda)$ is non-negligible, completing the proof.

Function privacy. Let $\mathcal{A}$ be a computationally unbounded (ℓ, k)-block-source function-privacy adversary that makes a polynomial number $Q = Q(\lambda)$ of queries to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. We prove that the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{real}}_{\mathsf{FP},\mathcal{SM},\mathcal{A}}$ is statistically close to the distribution of $\mathcal{A}$'s view in the experiment $\mathsf{Expt}^{\mathsf{rand}}_{\mathsf{FP},\mathcal{SM},\mathcal{A}}$ (we refer the reader to Definition 3.3 for the descriptions of these experiments). We denote these two distributions by $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$, respectively.

As the adversary $\mathcal{A}$ is computationally unbounded, we assume without loss of generality that $\mathcal{A}$ does not query the $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ oracle—such queries can be internally simulated by $\mathcal{A}$. Moreover, as discussed in Section 3, it suffices to focus on adversaries $\mathcal{A}$ that query the $\mathsf{RoR}^{\mathsf{FP}}$ oracle exactly once. From this point on we fix the public parameters pp chosen by the setup algorithm, and show that the two distributions $\mathsf{View}_{\mathsf{real}}$ and $\mathsf{View}_{\mathsf{rand}}$ are statistically close for any such pp.

Denote by $V = (V_1, \ldots, V_\ell)$ the random variable corresponding to the (ℓ, k)-source with which $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle. For each $i \in [\ell]$, let $(w_{i,1}, \ldots, w_{i,m})$ denote a sample from V_i. Also, let $\mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{S}^m$. As $\mathcal{A}$ is computationally unbounded, and having fixed the public parameters, we can in fact assume that

$$\mathsf{View}_{\mathsf{mode}} = \left(\left(\sum_{i=1}^{m} s_i \cdot w_{i,1} \right), \ldots, \left(\sum_{i=1}^{m} s_i \cdot w_{i,\ell} \right) \right) \tag{4.4}$$

for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, where $\mathbf{W} = \{w_{i,j}\}_{i \in [m], j \in [\ell]}$ is drawn from V for $\mathsf{mode} = \mathsf{real}$, $\mathbf{W}$ is uniformly distributed over $\mathbb{S}^{m \times \ell}$ for $\mathsf{mode} = \mathsf{rand}$, and $s_i \leftarrow \mathbb{S}$ for every $i \in [\ell]$. For $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ we prove that the distribution $\mathsf{View}_{\mathsf{mode}}$ is statistically close to a uniform distribution over $\mathbb{S}^m$.

Note that the collection of functions $\{g_{s_1,\ldots,s_m} : \mathbb{S}^m \to \mathbb{S}\}_{s_1,\ldots,s_m \in \mathbb{S}}$ defined by $g_{s_1,\ldots,s_m}(w_1, \ldots, w_m) = \sum_{j=1}^{m} s_j \cdot w_j$ is universal. This enables us to directly apply the Leftover Hash Lemma for block-sources (Lemma 2.2) implying that for our choice of parameters m, ℓ and k the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ.[3] The same clearly holds also for $\mathsf{View}_{\mathsf{rand}}$, as the uniform distribution over $\mathbb{S}^{m \times \ell}$ is, in particular, a (ℓ, k)-block-source. This completes the proof of function privacy.

∎

Theorem 4.1 for correlated $\mathsf{RoR}^{\mathsf{FP}}$ queries. Recollect that the definition of function privacy for subspace membership (Definition 3.3) extends to adversaries that query the $\mathsf{RoR}^{\mathsf{FP}}$ oracle with secret keys for T correlated subspaces $\mathbf{W}_1, \ldots, \mathbf{W}_T$ for any $T = \mathrm{poly}(\lambda)$. If the columns of the jointly sampled subspaces $[\mathbf{W}_1 | \mathbf{W}_2 | \cdots | \mathbf{W}_T]$ form a block source, we can extend the proof of function privacy to consider such correlated queries. The adversaries view comprises T terms as in Equation (4.4) with randomly sampled vectrs $\mathbf{s}_1, \ldots, \mathbf{s}_T$ in place of $\mathbf{s}$. The collection of functions g remains universal and a simple variant of Lemma 2.2 implies that for our choice of parameters, the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ (and similarly for $\mathsf{View}_{\mathsf{rand}}$).

[3] We note here that a weaker version of Lemma 2.2 will suffice as the adversary's view does not include $(s_1, \ldots, s_m)$.

4.2 Small Attribute Space

We also consider constructing subspace-membership encryption schemes where we do not place any restrictions on the size of the underlying attribute space $\mathbb{S}$. In our generic construction, observe that correctness requires that $1/|\mathbb{S}|$ be negligible in λ. If $|\mathbb{S}|$ is not super-polynomial in the security parameter, then correctness fails with a non-negligible probability. Additionally, this breaks the proof of attribute-hiding security in Theorem 4.1: In Equation (4.3), if the quantity $2Q/|\mathbb{S}|$ is non-negligible, then a non-negligible advantage of an adversary $\mathcal{A}$ *does not* translate to a non-negligible advantage for the reduction algorithm $\mathcal{B}$ against the inner-product encryption scheme.

To overcome this difficulty, we refine the construction as follows using a parameter $\tau = \tau(\lambda) \in \mathbb{N}$. We split the message into τ secret shares and apply parallel repetition of τ copies of the underlying inner-product encryption scheme, where each copy uses independent public parameters and master secret keys. For the proof of security, it suffices to have τ such that the quantity $\tau/|\mathbb{S}|^\tau$ is negligible in λ. The details of the construction are as follows.

Our construction. Let $\mathcal{IP} = (\mathsf{IP.Setup}, \mathsf{IP.KeyGen}, \mathsf{IP.Enc}, \mathsf{IP.Dec})$ be an inner-product encryption scheme with attribute set $\Sigma = \mathbb{S}^\ell$. We construct a subspace-membership encryption scheme $\mathcal{SM}_\tau = (\mathsf{SM.Setup}, \mathsf{SM.KeyGen}, \mathsf{SM.Enc}, \mathsf{SM.Dec})$ parameterized by a parameter $\tau = \tau(\lambda)$ as follows.

- **Setup:** On input the security parameter 1^λ, $\mathsf{SM.Setup}$ runs algorithm $\mathsf{IP.Setup}(1^\lambda)$ τ times independently. It outputs public parameters $\mathsf{pp} = (\mathsf{pp}_1, \ldots, \mathsf{pp}_\tau)$ and the master secret key $\mathsf{msk} = (\mathsf{msk}_1, \ldots, \mathsf{msk}_\tau)$.

- **Key generation:** $\mathsf{SM.KeyGen}$ takes as input the master secret key msk and a function $f_{\mathbf{W}}$ where $\mathbf{W} \in \mathbb{S}^{m\times\ell}$ and proceeds as follows. It samples uniform and independent $\mathbf{s}_1, \ldots, \mathbf{s}_\tau \leftarrow \mathbb{S}^m$ and computes $\mathbf{v}_i = \mathbf{W}^\mathsf{T}\mathbf{s}_i \in \mathbb{S}^\ell$ for $i \in [\tau]$. Next, it computes $\mathsf{sk}_i \leftarrow \mathsf{IP.KeyGen}(\mathsf{msk}_i, \mathbf{v}_i)$ and outputs $\mathsf{sk}_{\mathbf{W}} \overset{\mathsf{def}}{=} (\mathsf{sk}_1, \ldots, \mathsf{sk}_\tau)$.

- **Encryption:** On input the public parameters, an attribute $\mathbf{x} \in \mathbb{S}^\ell$, and a message M, the algorithm $\mathsf{SM.Enc}$ samples $\mathsf{M}_1, \ldots, \mathsf{M}_\tau \leftarrow \mathcal{M}$ uniformly at random subject to $\mathsf{M} = \mathsf{M}_1 \oplus \cdots \oplus \mathsf{M}_\tau$. Next, it computes ciphertexts $c_i = \mathsf{IP.Enc}(\mathsf{pp}_i, \mathbf{x}, \mathsf{M}_i)$. It outputs ciphertext $(c_1, \ldots, c_\tau)$.

- **Decryption:** On input the public parameters $\mathsf{pp} = (\mathsf{pp}_1, \ldots, \mathsf{pp}_\tau)$, a secret key $\mathsf{sk}_{\mathbf{W}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_\tau)$, and a ciphertext $c = (c_1, \ldots, c_\tau)$, the algorithm first $\mathsf{SM.Dec}$ computes $\mathsf{M}_i \leftarrow \mathsf{IP.Dec}(\mathsf{pp}_i, \mathsf{sk}_i, c_i)$. If $\mathsf{M}_i = \perp$ for any $i \in [\tau]$, the decryption algorithm outputs $\perp$. Else, it outputs $\mathsf{M}_1 \oplus \cdots \oplus \mathsf{M}_\tau$.

Correctness. Correctness of the scheme follows from correctness of the underlying inner-product encryption scheme. For every $\mathbf{W} \in \mathbb{S}^{m\times\ell}$ and $\mathbf{x} \in \mathbb{S}^\ell$, it sfufices to show the following:

- If $f(I) = 1$, then it holds that $\mathbf{W} \cdot \mathbf{x} = \mathbf{0}$. This implies that for every $i \in [\tau]$, $\mathbf{x}^\mathsf{T}\mathbf{v}_i = \mathbf{x}^\mathsf{T}(\mathbf{W}^\mathsf{T}\mathbf{s}_i) = 0$. The correctness of the underlying inner-product encryption implies that M_i is successfully recovered from c_i. Thus it follows that if $f(I) = 1$, then $\mathsf{SM.Dec}$ outputs M as required.

- If $f(I) = 0$, then it holds that $\mathbf{e} \overset{\mathsf{def}}{=} \mathbf{W} \cdot \mathbf{x} \neq \mathbf{0}$. As $\mathbf{x}^\mathsf{T}\mathbf{v}_i = \mathbf{x}^\mathsf{T}(\mathbf{W}^\mathsf{T}\mathbf{s}_i) = \mathbf{e}^\mathsf{T}\mathbf{s}_i$, for any $\mathbf{e} \neq \mathbf{0}$ the quantity $\mathbf{x}^\mathsf{T}\mathbf{v}_i$ is zero with probability $1/|\mathbb{S}|$ over choices of $\mathbf{s}$. The decryption algorithm fails to output $\perp$ only if $\mathbf{x}^\mathsf{T}\mathbf{v}_i = 0$ for *every* $i \in [\tau]$. As vectors $\mathbf{s}_i$ are sampled independently of $\mathbf{e}$ the error probability is at most $(1/|\mathbb{S}|)^\tau$ which is negligible for our choice of parameters.

We state the following theorem about the security of our construction.

Theorem 4.2. *If $\mathcal{IP}$ is an attribute hiding (resp. weakly attribute hiding) inner-product encryption scheme, then it holds that:*

1. *For any τ such that $\tau/|\mathbb{S}|^\tau = 2^{-\omega(\log\lambda)}$, the scheme $\mathcal{SM}_\tau$ is an attribute hiding (resp. weakly attribute hiding) subspace-membership encryption scheme under the same assumption as the security of the underlying inner-product encryption scheme.*

2. *For any τ such that $\tau/|\mathbb{S}|^\tau = 2^{-\omega(\log\lambda)}$ and $m > \tau$, the scheme $\mathcal{SM}_\tau$ is statistically function private for (ℓ, k)-block-sources for any $\ell = \mathrm{poly}(\lambda)$ and $k \geq \tau \cdot \log|\mathbb{S}| + \omega(\log\lambda)$.*

Proof. The proof of Theorem 4.2 follows the proof outline of Theorem 4.1 with the following important differences. In the proof that $\mathcal{SM}_\tau$ is attribute hiding, the proof follows from considering τ hybrid experiments. Starting with $\mathsf{Expt}^{(0)}_{\mathsf{X},\mathcal{SM}_\tau,\mathcal{B}}(\lambda)$ (for $\mathsf{X} \in \{\mathsf{AH}, \mathsf{wAH}\}$), each successive experiment replaces one more component of the ciphertext $(c_1, \ldots, c_\tau)$ with an encryption of M_1 under $\mathbf{x}_1$. The final experiment is therefore $\mathsf{Expt}^{(1)}_{\mathsf{X},\mathcal{SM}_\tau,\mathcal{B}}(\lambda)$. The proof that any two successive experiments are indistinguishable follows directly from the proof of the attribute hiding property of $\mathcal{SM}$ in Theorem 4.1.

In the proof of function privacy, observe that the difference between the two schemes is that the adversary has τ repetitions of the same view with independent vectors $\mathbf{s}_i$. We can still apply the Leftover Hash Lemma for block-sources (Lemma 2.2) to show that for sources with slightly larger min-entropy (at least $\tau \cdot |\mathbb{S}| + \omega(\log\lambda)$) there is still enough entropy "leftover" to allow for τ parallel repetitions.

Attribute hiding. Let $\mathsf{X} \in \{\mathsf{AH}, \mathsf{wAH}\}$. Consider the following experiments interacting with an adversary $\mathcal{A}$. Experiment Expt_0 is identical to $\mathsf{Expt}^{(0)}_{\mathsf{X},\mathcal{SM}_\tau,\mathcal{A}}$. Let $\mathsf{M}^{(0)}$ and $\mathsf{M}^{(1)}$ be the two challenge messages constructed by $\mathcal{A}$ and let $\mathsf{M}^{(b)}_1, \ldots, \mathsf{M}^{(b)}_\tau$ denote the additive shares of $\mathsf{M}^{(0)}$ constructed during encryption.

For $i \in [\tau]$, experiment Expt_i is derived from Expt_{i-1} by replacing c_i in $\mathcal{A}$'s challenge ciphertext with $\mathsf{Enc}\left(\mathsf{pp}_i, \mathbf{x}_1, \mathsf{M}^{(1)}_i\right)$ instead of $\mathsf{Enc}\left(\mathsf{pp}_i, \mathbf{x}_0, \mathsf{M}^{(0)}_i\right)$. It follows that experiment Expt_τ is identical to $\mathsf{Expt}^{(1)}_{\mathsf{X},\mathcal{SM}_\tau,\mathcal{A}}$.

For every $i \in [\tau]$, given an adversary $\mathcal{A}$ such that $\left|\Pr\left[\mathsf{Expt}_{i-1}(\lambda) = 1\right] - \Pr[\mathsf{Expt}_i(\lambda) = 1]\right|$ is non-negligible, we construct an adversary $\mathcal{B}$ that interacts with an inner-product encryption attribute hiding challenger with non-negligible advantage

$$\mathbf{Adv}^{\mathsf{X}}_{\mathcal{IP},\mathcal{B}}(\lambda) \approx \left|\Pr\left[\mathsf{Expt}_{i-1}(\lambda) = 1\right] - \Pr[\mathsf{Expt}_i(\lambda) = 1]\right|$$

as follows.

Fix an $i \in [\tau]$. As in the proof of Theorem 4.1, $\mathcal{A}$ outputs a pair of attributes $\mathbf{x}_0$ and $\mathbf{x}_1$ and $\mathcal{B}$ forwards them to the $\mathcal{IP}$-challenger. Upon receiving pp, it samples independent $(\mathsf{pp}_j, \mathsf{msk}_j) \leftarrow \mathsf{Setup}(1^\lambda)$ for $j \in [\tau]\backslash\{i\}$ and sets $\mathsf{pp}_i = \mathsf{pp}$. Algorithm $\mathcal{A}$ receives public parameters $(\mathsf{pp}_1, \ldots, \mathsf{pp}_\tau)$. To answer secret key queries, secret keys $\{\mathsf{sk}_j\}_{j\in[\tau]\backslash\{i\}}$ are computed honestly using their respective master secret keys, and sk_i is simulated using the $\mathcal{IP}$-challenger as in the proof of Theorem 4.1. Algorithm $\mathcal{B}$ answers the adversary's secret key query with the tuple $(\mathsf{sk}_1, \ldots, \mathsf{sk}_\tau)$. For secret key query $\gamma \in [Q]$, we let $(\mathbf{v}_{\gamma,1}, \mathbf{v}_{\gamma,2}, \ldots, \mathbf{v}_{\gamma,\tau})$ denote the components constructed by $\mathcal{B}$ as in the key generation algorithm.

$\mathcal{A}$ outputs two messages $\mathsf{M}^{(0)}$ and $\mathsf{M}^{(1)}$. If there exists an $\gamma \in [Q]$ such that for at least one $b \in \{0,1\}$, for *every* $j \in [\tau]$, $\mathbf{v}_{\gamma,j}{}^{\mathsf{T}}\mathbf{x}_b = 0$, then $\mathcal{B}$ aborts and outputs a uniform bit. Otherwise, it samples $\mathsf{M}_2, \ldots, \mathsf{M}_\tau \leftarrow \mathcal{M}$ and then computes $\mathsf{M}^*_0 = \mathsf{M}_0 \oplus (\mathsf{M}_2 \oplus \cdots \oplus \mathsf{M}_\tau)$ and $\mathsf{M}^*_1 = \mathsf{M}_1 \oplus$

$(\mathsf{M}_2 \oplus \cdots \oplus \mathsf{M}_\tau)$. Intuitively, for $j \neq i$, message-shares M_j are *independent* of M_b and provide no information to $\mathcal{A}$. Thus, they play the role of *both shares* $\mathsf{M}_j^{(0)}$ and $\mathsf{M}_j^{(1)}$. Algorithm $\mathcal{B}$ forwards M_0^* and M_1^* to the $\mathcal{IP}$-challenger to receive a challenge ciphertext c. It honestly computes the remaining ciphertext components: for $1 \leq j \leq i-1$, $c_j \leftarrow \mathsf{IP.Enc}(\mathsf{pp}_j, \mathbf{x}_0, \mathsf{M}_j)$ and for $i+1 \leq j \leq \tau$, $c_j \leftarrow \mathsf{IP.Enc}(\mathsf{pp}_j, \mathbf{x}_1, \mathsf{M}_j)$. Finally, it sets $c_i = c$ that it received from the $\mathcal{IP}$-challenger. It returns a challenge ciphertext $(c_1, \ldots, c_\tau)$. After answering further key-generation queries, $\mathcal{B}$ receives a guess b from $\mathcal{A}$ and outputs the bit b.

Observe that the algorithm $\mathcal{B}$ simulates the adversary honestly. As in the derivation of Equation (4.3), for every $i \in [\tau]$, it holds that:

$$\left| \Pr\left[\mathsf{Expt}_{i-1}(\lambda) = 1\right] - \Pr[\mathsf{Expt}_i(\lambda) = 1] \right| \leq \mathbf{Adv}^{\mathsf{X}}_{\mathcal{IP},\mathcal{B}}(\lambda) + \frac{2Q}{|\mathbb{S}|^\tau}, \tag{4.5}$$

where the term $2Q/|\mathbb{S}|^\tau$ is the probability $\mathcal{B}$ aborts and is derived exactly as in the proof of correctness of the scheme $\mathcal{SM}_\tau$.

Using a straightforward triangle inequality and τ applications of Equation (4.5), it holds that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{X}}_{\mathcal{SM}_\tau,\mathcal{A}}(\lambda) &= \left| \Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{X},\mathcal{SM}_\tau,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{X},\mathcal{SM}_\tau,\mathcal{A}}(\lambda) = 1\right] \right| \\
&= \left|\Pr[\mathsf{Expt}_0(\lambda) = 1] - \Pr[\mathsf{Expt}_\tau(\lambda) = 1]\right| \\
&\leq \sum_{i=1}^{\tau} \left| \Pr\left[\mathsf{Expt}_{i-1}(\lambda) = 1\right] - \Pr[\mathsf{Expt}_i(\lambda) = 1] \right| \\
&\leq \tau \cdot \mathbf{Adv}^{\mathsf{X}}_{\mathcal{IP},\mathcal{B}}(\lambda) + \frac{2\tau Q}{|\mathbb{S}|^\tau},
\end{aligned}$$

which is negligible from our choice of parameters and the fact that $\mathcal{IP}$ is attribute hiding.

Function privacy. The proof of function privacy of $\mathcal{SM}_\tau$ is almost identical to the proof of function privacy of $\mathcal{SM}$ (see proof of Theorem 4.1). If we let $V = (V_1, \ldots, V_\ell)$ the random variable corresponding to the (ℓ, k)-source with which $\mathcal{A}$ queries the $\mathsf{RoR}^{\mathsf{FP}}$ oracle, for each $i \in [\ell]$, let $(w_{i,1}, \ldots, w_{i,m})$ denote a sample from V_i, for each $j \in [\tau]$, let $\mathbf{s}_j = (s_{j,1}, \ldots, s_{j,m}) \in \mathbb{S}^m$, as $\mathcal{A}$ is computationally unbounded, and having fixed the public parameters, we can in fact assume that (as in the previous proof),

$$\begin{aligned}
\mathsf{View}_{\mathsf{mode}} = \Bigg(&\left(\sum_{i=1}^{m} s_{1,i} \cdot w_{i,1}\right), \ldots, \left(\sum_{i=1}^{m} s_{1,i} \cdot w_{i,\ell}\right), \\
&\left(\sum_{i=1}^{m} s_{2,i} \cdot w_{i,1}\right), \ldots, \left(\sum_{i=1}^{m} s_{2,i} \cdot w_{i,\ell}\right), \\
&\vdots \\
&\left(\sum_{i=1}^{m} s_{\tau,i} \cdot w_{i,1}\right), \ldots, \left(\sum_{i=1}^{m} s_{\tau,i} \cdot w_{i,\ell}\right) \Bigg)
\end{aligned}$$

for $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$, where $\mathbf{W} = \{w_{i,j}\}_{i \in [m], j \in [\ell]}$ is drawn from V for $\mathsf{mode} = \mathsf{real}$, $\mathbf{W}$ is uniformly distributed over $\mathbb{S}^{m \times \ell}$ for $\mathsf{mode} = \mathsf{rand}$, and $s_{j,i} \leftarrow \mathbb{S}$ for every $i \in [\ell]$ and $j \in [\tau]$. For $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ we prove that the distribution $\mathsf{View}_{\mathsf{mode}}$ is statistically-close to uniform.

Note that the collection of functions $\mathcal{G} \stackrel{\mathsf{def}}{=} \{g_{s_1,\ldots,s_m} : \mathbb{S}^m \to \mathbb{S}\}_{s_1,\ldots,s_m \in \mathbb{S}}$ defined by $g_{s_1,\ldots,s_m}(w_1, \ldots, w_m) = \sum_{j=1}^{m} s_j \cdot w_j$ is universal. Additionally, if $(g_1, \ldots, g_\tau) \leftarrow \mathcal{G}^\tau$ be τ independent and uniform samples from the family $\mathcal{G}$, for any distinct $(w_1, \ldots, w_m)$ and $(w'_1, \ldots, w'_m)$ in $\mathbb{S}^m$, it holds that

$$\Pr_{(g_1,\ldots,g_\tau)\leftarrow\mathcal{G}^\tau}\left[\bigwedge_{i=1}^{\tau} g_i(w_1, \ldots, w_m) = g_i(w'_1, \ldots, w'_m)\right] = \prod_{i=1}^{\tau} \Pr_{g_i\leftarrow\mathcal{G}}\left[g_i(w_1, \ldots, w_m) = g_i(w'_1, \ldots, w'_m)\right]$$
$$= \left(\frac{1}{|\mathbb{S}|}\right)^{\tau}.$$

Here, the first equality follows from the independence of the functions $g_1, \ldots, g_\tau$. Therefore, the collection $\mathcal{G}^\tau \stackrel{\mathsf{def}}{=} \{g_1, \ldots, g_\tau : \mathbb{S}^m \to \mathbb{S}^\tau\}_{g_1,\ldots,g_\tau\leftarrow\mathcal{G}}$ is also a universal collection.

Thus allows us to apply Lemma 2.2 implying that for our choice of parameters m, ℓ, and k the statistical distance between $\mathsf{View}_{\mathsf{real}}$ and the uniform distribution is negligible in λ.[4] The same clearly holds also for $\mathsf{View}_{\mathsf{rand}}$, as the uniform distribution over $\mathbb{S}^{m\times\ell}$ is, in particular, an (ℓ, k)-block-source. This completes the proof of function privacy. ∎

5 Application: Function-Private Identity-Based Encryption

In this section we present a generic construction of a function-private identity-based encryption scheme from any function-private subspace-membership encryption scheme with a relatively large attribute space (super-polynomial in the security parameter). Combining this with our construction from Section 4.1 yields, in particular, a construction of a function-private identity-based encryption scheme from any inner-product encryption scheme with a relatively large attribute space. Note that this does not require the underlying inner-product encryption scheme to provide any form of function privacy.

The scheme. Let $m = m(\lambda) > 1$, $\ell = \ell(\lambda) = m + 1$, and let $\mathcal{SM} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a *large-attribute* subspace-membership encryption scheme with parameters m and $\ell = m + 1$ over an attribute set $\mathbb{S} = \mathbb{S}(\lambda)$ (as defined in Section 3) such that $|\mathbb{S}|$ is super-polynomial in the security parameter. We construct an identity-based encryption scheme $\mathcal{IBE} = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ for the identity space $\mathcal{ID} = \mathbb{S}^m$ as follows.

- **Setup:** The algorithm Setup' is identical to the algorithm Setup of the underlying scheme $\mathcal{SM}$.

- **Key generation:** To generate a secret key for an identity $\mathsf{id} = (\mathsf{id}_0, \ldots, \mathsf{id}_{m-1}) \in \mathcal{ID}$, the key-generation algorithm KeyGen' first constructs a matrix $\mathbf{W}_{\mathsf{id}} \in \mathbb{S}^{m\times(m+1)}$ by sampling a uniformly distributed matrix $\mathbf{R} \in \mathbb{S}^{m\times m}$ and setting

$$\mathbf{W}_{\mathsf{id}} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{R} \cdot \begin{pmatrix} \mathsf{id}_0 \\ \vdots \\ \mathsf{id}_{m-1} \end{pmatrix} \end{array}\right]. \tag{5.1}$$

  The algorithm then runs $\mathsf{KeyGen}(\mathsf{msk}, \mathbf{W}_{\mathsf{id}})$ and outputs the resulting secret key $\mathsf{sk}_{\mathbf{W}_{\mathsf{id}}}$.

- **Encryption:** To encrypt a message M for an identity $(\mathsf{id}_0, \ldots, \mathsf{id}_{m-1}) \in \mathbb{S}^m$, the encryption algorithm Enc' sets $\mathbf{x} = (\mathsf{id}_0, \mathsf{id}_1, \ldots, \mathsf{id}_{m-1}, -1)^\mathsf{T}$ and outputs $\mathsf{Enc}(\mathsf{pp}, \mathbf{x}, \mathsf{M})$.

[4] We note here that a weaker version of Lemma 2.2 will suffice as the adversary's view does not include $s_{j,i}$ for $i \in [\ell]$ and $j \in [\tau]$.

- **Decryption:** The algorithm Dec' is identical to the algorithm Dec of the underlying scheme $\mathcal{SM}$.

Correctness. Let $\vec{\mathsf{id}} \in \mathbb{S}^m$ denote the column vector $(\mathsf{id}_0, \ldots, \mathsf{id}_{m-1})^\intercal$. Consider an attribute $\mathbf{x} = (\mathsf{id}_0, \mathsf{id}_1, \ldots, \mathsf{id}_{m-1}, -1)^\intercal$ corresponding to an identity id. The matrix $\mathbf{W}_{\mathsf{id}}$ is constructed such that $\mathbf{W}_{\mathsf{id}} \cdot \mathbf{x} = \mathbf{R} \cdot \vec{\mathsf{id}} - \mathbf{R} \cdot \vec{\mathsf{id}} = \mathbf{0}$. Thus, correctness of the IBE scheme follows directly from the correctness of the underlying SME scheme.

Data privacy. The data privacy of the IBE scheme follows from the attribute hiding property of the underlying subspace-membership encryption scheme.

Theorem 5.1. *If $\mathcal{SM}$ is a (large-attribute) subspace-membership encryption scheme that is attribute hiding, then $\mathcal{IBE}$ constructed above is* selectively *data private.*

Proof. Recollect the definition of anon-IND-sID-CPA (sDP) data privacy for an IBE scheme from Definition 2.5. Any adversary $\mathcal{A}$ that has a non-negligible advantage in the sDP game can be used to construct an algorithm $\mathcal{B}$ that breaks the attribute hiding property (see Definition 2.3) of the underlying subspace-membership encryption scheme as follows. (We actually show a slightly stronger result below where in the sDP data privacy game, the adversary chooses the challenge messages M_0^* and M_1^* *after* receiving secret keys of his choice.)

Upon receiving the challenge identities id_0^* and id_1^* from $\mathcal{A}$, the algorithm $\mathcal{B}$ constructs the attributes $\mathbf{x}_0^*$ and $\mathbf{x}_1^*$ as constructed in the encryption algorithm and sends this as the challenge attributes to the attribute-hiding challenger. Queries to the $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ oracle from $\mathcal{A}$ are answered by $\mathcal{B}$ by constructing the appropriate subspace $\mathbf{W}$ and requesting secret keys $\mathsf{sk}_{\mathbf{W}}$ from the attribute-hiding challenger.

Finally, the challenge messages M_0^* and M_1^* chosen by $\mathcal{A}$ are forwarded by $\mathcal{B}$ to the attribute-hiding challenger. It receives a challenge ciphertext c^* that it forwards to $\mathcal{A}$. The algorithm $\mathcal{B}$ continues to simulate $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ queries as above and finally returns the bit b' that algorithm $\mathcal{A}$ outputs.

Observe that the algorithm $\mathcal{A}$ cannot issue a key-generation query for any identity that is either id_0^* or id_1^*. As the set $\mathbb{S}$ is superpolynomially large, with overwhelming probability, the matrix $\mathbf{R}$ in the construction of $\mathbf{W}_{\mathsf{id}}$ is full-rank in $\mathbb{S}$ and this implies that for all identities $\mathsf{id} \notin \{\mathsf{id}_0^*, \mathsf{id}_1^*\}$, $\mathbf{W}_{\mathsf{id}} \cdot \mathbf{x}_0^* \neq \mathbf{0}$ and $\mathbf{W}_{\mathsf{id}} \cdot \mathbf{x}_1^* \neq \mathbf{0}$. Thus, the queries issued by algorithm $\mathcal{B}$ are that of an allowed attribute-hiding adversary.

Finally, it follows in a straightforward manner that for $b \in \{0, 1\}$, if $\mathcal{B}$ is interacting with $\mathsf{Expt}^{(b)}_{\mathsf{AH},\mathcal{SM},\mathcal{B}}$, then the challenger ciphertext is an encryption of M_b^* to the identity id_b^*. Thus, algorithm $\mathcal{B}$ simulates $\mathsf{Expt}_{\mathsf{sDP},\mathcal{IBE},\mathcal{A}}$ correctly. From this, we conclude that $\mathbf{Adv}^{\mathsf{AH}}_{\mathcal{SM},\mathcal{B}}(\lambda) = \mathbf{Adv}^{\mathsf{sDP}}_{\mathcal{IBE},\mathcal{A}}(\lambda) - \mathrm{negl}(\lambda)$ which completes the proof of the selective data privacy of the IBE scheme. ∎

Function privacy. We show that with overwhelming probability over the choice of the matrix $\mathbf{R} \in \mathbb{S}^{m \times m}$, if the identity id is sampled from a k-source, then the columns of $\mathbf{W}_{\mathsf{id}}$ are distributed according to a $(m+1, k)$-block-source. And if the identity is sampled uniformly from $\mathcal{ID}$, then $\mathbf{W}_{\mathsf{id}}$ is distributed uniformly over $\mathbb{S}^{m \times (m+1)}$. This allows us to simulate, in a straightforward manner, a $\mathsf{RoR}^{\mathsf{IBE}}$ oracle given access to a RoR oracle for the subspace membership predicate. Thus, we can state the following theorem.

Theorem 5.2. *If $\mathcal{SM}$ is a subspace membership encryption scheme with parameters $m = \mathrm{poly}(\lambda)$ and $\ell = m + 1$ that satisfies function privacy against $(m+1, k)$-block-source adversaries, then the IBE scheme constructed above is* statistically *function private against k-source adversaries.*

Before we prove Theorem 5.2, we note that from Theorem 4.1, we can construct a subspace-membership scheme for any $m \geq 2$ that is *statistically* function private against $(m+1, k)$-block-source adversaries for any $k \geq \log|\mathbb{S}| + \omega(\log \lambda)$ from an underlying (large attribute-space) $\mathcal{IP}$ scheme in a *black-box* manner. This gives us the first black-box IBE schemes that are function-private against k-sources for $k \geq \log|\mathbb{S}| + \omega(\log \lambda)$.

Proof of Theorem 5.2. Recollect the definition of the vector $\vec{\mathsf{id}}$ corresponding to the identity id. We first consider the case when id (and hence, $\vec{\mathsf{id}}$) is sampled from a k-source. Observe that the first m columns of $\mathbf{W}_{\mathsf{id}}$ correspond to columns of $\mathbf{R}$ that is sampled uniformly at random from all $\mathbb{S}^{m\times m}$ matrices. It follows that each of the first m columns have entropy $m \cdot \log|\mathbb{S}| \geq k$ even conditioned on previous columns.

The last column is of the form $\mathbf{R} \cdot \vec{\mathsf{id}}$. With overwhelming probability over the choice of $\mathbf{R}$, it holds that $\mathbf{R}$ is full-rank over $\mathbb{S}$. As $\mathbf{R}$ is invertible, it injectively maps identities to the final column. Thus, the final column has the same entropy k as the distribution of $\vec{\mathsf{id}}$ and this holds even given $\mathbf{R}$. Thus, the columns of $\mathbf{W}_{\mathsf{id}}$ come from a $(m+1, k)$-block-source, as required.

A similar argument allows us to conclude that if $\vec{\mathsf{id}}$ is distributed uniformly in $\mathbb{S}^d$, the columns of $\mathbf{W}_{\mathsf{id}}$ have entropy $m \cdot |\mathbb{S}|$ conditioned on the previous columns. Thus, it is sampled uniformly from the set of all matrices in $\mathbb{S}^{m\times(m+1)}$.

With these two observations, the proof of the theorem is straightforward. Given an IBE function-privacy adversary, we can construct an adversary that breaks the function privacy of the subspace-membership scheme as follows. The setup algorithm simply outputs parameters pp output by the underlying subspace-membership setup algorithm. To simulate responses to the $\mathsf{RoR}^{\mathsf{IBE}}$ oracle, the reduction algorithm samples an identity id from the adversary's distribution, constructs $\mathbf{W}_{\mathsf{id}}$ and forwards the query to the $\mathsf{RoR}^{\mathsf{FP}}$ oracle of the subspace-membership encryption scheme. It returns the secret key sk received from the oracle to the IBE function-privacy adversary.

As proved above, when interacting with a k-source IBE function-privacy adversary the reduction simulates a valid $(m+1, k)$-block-source adversary against $\mathcal{SM}$. Moreover, if $\mathsf{mode} = \mathsf{real}$, then the replies to the $\mathsf{RoR}^{\mathsf{IBE}}$ query correspond to secret keys for identities drawn from the real distribution, and if $\mathsf{mode} = \mathsf{rand}$, then the replies correspond to secret keys for identities drawn from the uniform distribution. Thus, any adversary that breaks the function-privacy of the IBE scheme can be used to break the function-privacy of the underlying subspace-membership encryption scheme with identical advantage. This concludes the proof of the theorem. ∎

Fully-secure function-private IBE. Current constructions of inner-product encryption schemes [KSW08, AFV11] satisfy a selective notion of security where the challenge attributes are chosen by the adversary before seeing the public parameters. Our transformation of inner-product encryption schemes to function-private IBE schemes (via subspace membership) is not limited to selective security. Starting from an inner-product encryption scheme satisfying an adaptive version of attribute hiding, our construction yields fully-secure function-private IBE schemes. We also note that the standard complexity leveraging approach (see [BB11, Section 7.1]) gives a generic transformation from selectively-secure IBE to fully-secure IBE. This approach does not modify the key generation algorithm and therefore preserves function privacy.

6 Conclusions and Open Problems

Our work proposes subspace-membership encryption and constructs the first such function-private schemes from any inner-product encryption scheme. We also show its application to constructing function-private IBE schemes. In this section, we discuss a few extensions and open problems that arise from this work.

Function privacy from computational assumptions. In this work we construct subspace-membership schemes that are *statistically* function private. Although the construction of inner-product encryption schemes from lattices [AFV11] presents an immediate function-privacy attack, we were unable to find such attacks for the construction from composite-order groups [KSW08] (or its prime order variant [Fre10]). We conjecture that suitable "min-entropy" variants of the decisional Diffie-Hellman assumption [Can97] have a potential for yielding a proof of computational function privacy for these schemes.

Other predicates. A pre-cursor to the work on predicate encryption supporting inner-products was work on predicate encryption supporting comparison and range queries by Boneh and Waters [BW07]. They achieve this by constructing predicate encryption supporting an interesting primitive, denoted Hidden-Vector Encryption (HVE). Briefly, in HVE, attributes correspond to vectors over an alphabet Σ and secret keys correspond to vectors over the *augmented* alphabet $\Sigma \cup \{\star\}$. Decryption works if the attributes and secret key match for every coordinate that is not a $\star$.

HVE can be implemented using inner-product encryption schemes [KSW08] but it breaks function privacy in a rather trivial manner. Formalizing function privacy for HVE does not immediately follow from the notion of function privacy for inner-products because of the role played by $\star$. The questions of formalizing function privacy (which in turn will imply realistic notions also for encryption supporting range and comparison queries) and designing function-private HVE schemes are left as open problems. It is also open to formalize security and design function-private encryption schemes that support polynomial evaluation predicates [KSW08].

Enhanced function privacy. A stronger notion of function privacy, denoted enhanced function privacy [BRS13], asks that an adversary learn nothing more than the minimum necessary from a secret key even given corresponding ciphertexts with attributes that allow successful decryption. Constructing enhanced function-private schemes for subspace membership and inner products is an interesting line of research that may require new ideas and techniques.

Acknowledgements

We thank Venkata Koppula and Kim Ramchen for their comments on an earlier draft of this work. This work was supported by NSF, the DARPA PROCEED program, an AFO SR MURI award, a grant from ONR, an IARPA project provided via DoI/NBC, and by Samsung. Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or IARPA. Distrib. Statement "A:" Approved for Public Release, Distribution Unlimited.

References

[ABC+08] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *Journal of Cryptology*, 21(3):350–391, 2008.

[ABN10] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In *Proceedings of the 7th Theory of Cryptography Conference*, pages 480–497, 2010.

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *Advances in Cryptology – ASIACRYPT '11*, pages 21–40, 2011.

[AGVW13] Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In *Advances in Cryptology – CRYPTO '13*, pages 500–518, 2013.

[BB11] Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, 2011.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology – EUROCRYPT '04*, pages 506–522, 2004.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. Preliminary version in *Advances in Cryptology – CRYPTO '01*, pages 213–229, 2001.

[BH08] Dan Boneh and Michael Hamburg. Generalized identity based and broadcast encryption schemes. In *Advances in Cryptology – ASIACRYPT '08*, pages 455–470, 2008.

[BO12] Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. Cryptology ePrint Archive, Report 2012/515, 2012.

[BRS13] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In *Advances in Cryptology – CRYPTO '13*, pages 461–478, 2013.

[BSNS08] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Public key encryption with keyword search revisited. In *Proceedings on the International Conference Computational Science and Its Applications*, pages 1249–1259, 2008.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Proceedings of the 8th Theory of Cryptography Conference*, pages 253–273, 2011.

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *Proceedings of the 4th Theory of Cryptography Conference*, pages 535–554, 2007.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology – CRYPTO '97*, pages 455–469, 1997.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CKRS09] Jan Camenisch, Markulf Kohlweiss, Alfredo Rial, and Caroline Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography*, pages 196–214, 2009.

[CV08] Kai-Min Chung and Salil P. Vadhan. Tight bounds for hashing block sources. In *APPROX-RANDOM*, pages 357–370, 2008.

[Fre10] David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *Advances in Cryptology – EUROCRYPT '10*, pages 44–61, 2010.

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the 45th Annual ACM Symposium on the Theory of Computing*, pages 555–564, 2013.

[GSW04] Philippe Golle, Jessica Staddon, and Brent R. Waters. Secure conjunctive keyword search over encrypted data. In *Proceedings of the 2nd International Conference on Applied Cryptography and Network Security*, pages 31–45, 2004.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology – CRYPTO '12*, pages 162–179, 2012.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Advances in Cryptology – EUROCRYPT '08*, pages 146–162, 2008.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. IACR Cryptology ePrint Archive, Report 2010/556, 2010.

[OT09] Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In *Advances in Cryptology – ASIACRYPT '09*, pages 214–231, 2009.

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *Advances in Cryptology – EUROCRYPT '12*, pages 591–608, 2012.

[SBC+07] Elaine Shi, John Bethencourt, Hubert T.-H. Chan, Dawn Song, and Adrian Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy*, pages 350–364, 2007.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – CRYPTO '84*, pages 47–53, 1984.

[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In *Proceedings of the 6th Theory of Cryptography Conference*, pages 457–473, 2009.

[Zuc96] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, 1996.