Subset Predicate Encryption and its Applications

Jonathan Katz[1], Matteo Maffei[2], Giulio Malavolta[3], and Dominique Schröder[3]

[1] University of Maryland
[2] TU Vienna
[3] Friedrich-Alexander-University Erlangen-Nürnberg

Abstract. In this work we introduce the notion of Subset Predicate Encryption, a form of attribute-based encryption (ABE) in which a message is encrypted with respect to a set s' and the resulting ciphertext can be decrypted by a key that is associated with a set s if and only if $s \subseteq s'$. We formally define our primitive and identify several applications. We also propose two new constructions based on standard assumptions in bilinear groups; the constructions have very efficient decryption algorithms (consisting of one and two pairing computations, respectively) and small keys: in both our schemes, private keys contain only two group elements. We prove selective security of our constructions without random oracles. We demonstrate the usefulness of Subset Predicate Encryption by describing several black-box transformations to more complex primitives, such as identity-based encryption with wildcards and ciphertext-policy ABE for DNF formulas over a small universe of attributes. All of the resulting schemes are as efficient as the base Subset Predicate Encryption scheme in terms of decryption and key generation.

1 Introduction

Attribute-Based Encryption (ABE), and more generically functional encryption, introduces a new communication paradigm where the sender is allowed to specify a certain policy that the receiver must satisfy in order to read the data. Since its introduction in [15], ABE has had a tremendous impact in the research community and a plethora of different construction have been proposed, from different assumption and with different security notions and functionalities. However, more effort is needed towards the adoption of ABE schemes on a large scale as we only know a bunch of schemes that are efficient enough to be deployed in practice. In this work we contribute to the understanding of efficiency tradeoffs in ABE (and weaker instances of functional encryption) by proposing a new prospective for the construction of efficient schemes. With this aim in mind, we introduce the notion of Subset Predicate Encryption (SPE). In a SPE scheme, sets are defined over some finite universe of elements. A user with a secret key for the set s can decrypt a ciphertext encrypted with the public key s' if and only if s defines a *subset* of s'. A SPE scheme must enforce that an adversary knowing the key for some set s cannot derive a valid key for any set different from s (e.g., by stripping off part of s from its original key). In particular, users must not be allowed to combine different keys in a meaningful manner (e.g., to decrypt any ciphertext that no user could have decrypted individually). A perhaps more natural way to look at SPE is as a generalization of broadcast encryption (BE): In this perspective BE can be seen as a special case of SPE where secret keys are associated with singleton subsets, i.e., $|s| = 1$. SPE opens the possibility to efficiently enforce expressive access control policies in several interesting scenarios, as described below.

Concise Access Control. An important aspect of SPE is that it enables access control over data in a very concise fashion. For instance, let us consider a corporate setting, where all users of the system encrypt all messages under the sets corresponding to the attributes of the fields "{sender, receiver, department, current-date}". Deriving keys in a hierarchical fashion is straightforward, however our system allows us to assign keys for more complex policies in a concise way. As an example, we can generate a key for decrypting all messages exchanged on a certain day across multiple departments by simply deriving a key for current-date. Furthermore, we can generate a key to read all the messages sent *from and to* Alice with a single key for the set corresponding to the element Alice.

Pattern Matching. Imagine a scenario where each email is encrypted under the set corresponding to the words of the subject (assuming a subject of a fixed length). We could disclose the content of all emails containing a certain word (buy, as an example) in the subject by simply creating a key for the set corresponding to the element buy. It is important to note that the position of the word must not be necessarily known in advance, since the decryption is successful if the set encoded in the key matches any subset of the set encoded in the ciphertext.

A blackbox instantiation of SPE from Identity Based Encryption seems not be easily achievable: One could express the same functionality by encrypting the same message for the powerset of a given identity, but it is easy to see that the size of the ciphertext would grow exponentially in the length of the identity. While we conjecture that SPE is strictly more expressive than IBE, it is not hard to show that SPE is implied by any generic ABE system. However, due to the simplicity of our primitive, there is hope to create a SPE scheme in a more efficient manner, without resorting to generic ABE solutions. In particular, we are interested in maximizing the efficiency of the system for the end-users, both in terms of computation and in terms of storage. An efficient decryption algorithm is an important feature of any encryption scheme as it allows computationally-constrained devices to be integrated in the system: Since decryption is arguably the most recurrent operation (a user typically encrypts once for multiple receivers), its running time is fundamental for the scalability of the system. Additionally, small private keys are convenient as they are often stored in tamper-resistant memory, which in general is very costly. This can be especially critical in small devices, such as sensors, for which low cost solutions are often required.

In this work, we focus on the improvement over these two aspects and we present two cryptographic constructions for SPE with a very efficient decryption algorithm and constant-size private keys. Perhaps surprisingly, our abstraction turns out to subsume more complex primitives, such as ABE for DNF formulas over a small universe of attributes, and we show how to generically instantiate them from a SPE scheme. All of the resulting schemes inherit the efficiency of our constructions.

1.1 Our Contributions

We formalize the notion of Subset Predicate Encryption and its security guarantees using standard game-based definitions. We provide two instantiations for a SPE scheme from bilinear maps. Both of the schemes are proven secure in the selective security model without random oracles. Our first construction offers an extremely efficient decryption operation consisting of only a *single* pairing. Moreover, the secret keys are very compact as each key is composed of a group element and an integer value. The security of this scheme relies on the hardness of the Decisional q-Bilinear Diffie-Hellman Inversion assumption over bilinear groups. Our second scheme has a slightly less efficient decryption procedure (where two pairings are computed) but is based on the Decisional Bilinear Diffie-Hellman assumption. In this scheme, each private key is as large as two group elements.

We describe several generic black-box transformations that turn SPE into more expressive primitives. Our first transformation turns any SPE into an Identity-Based Encryption scheme with wildcards (WIBE), whereas our second transformation yields an ABE scheme for formulas in their DNF over for a small universe of attributes. A nice feature of these transformations is that the resulting schemes maintain the same decryption algorithm and key sizes of the base construction. Beyond being an interesting primitive on its own right, we believe that the conceptual simplicity of SPE might help in the future design of efficient WIBE and ABE schemes.

We summarize a comparison of our instantiations against the most efficient known WIBE schemes in Table 1: Our transformation yields the first scheme with constant-size keys and the decryption of our first construction is roughly 50% faster than the best instantiation of [1]. The performance of the ABE schemes derived generically from our instantiations of SPE are shown in Table 2. With respect to the best known instance of ABE in terms of key-size [11], both of our instantiations cut the size of the keys down to 50%. Furthermore the decryption algorithm of our first construction computes only one pairing and one modular exponentiation (while the second computes two pairings). This is unprecedented in the context of ABE, where in the fastest known scheme [16] the amount of modular exponentiations is linear in the size of the universe of attributes. This means that our schemes have an arbitrarily more efficient decryption, depending on the size of the universe of attributes. For a fair comparison we shall mention that the aforementioned schemes are more expressive the ours and satisfy stronger security notions.

Table 1: Comparison amongst the most efficient wildcard IBE schemes in the literature in terms of size of the public parameters ($|\mathsf{pk}|$), size of the decryption keys ($|\mathsf{sk}|$), size of the ciphertexts ($|\mathsf{c}|$), number of operations required for decrypting ($\mathsf{Decrypt}$), and complexity assumptions. Here ω denotes the depth of the hierarchy, P denotes the number of pairing operations and E the number of modular exponentiations.

WIBE Scheme	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{sk}\rvert$	$\lvert\mathsf{c}\rvert$	$\mathsf{Decrypt}$	Assumption
BBG-WIBE [1]	$(\omega+4)\mathbb{G}$	$(\omega+2)\mathbb{G}$	$(\omega+2)\mathbb{G}+\mathbb{G}_T$	2P	ω-BDHI
Waters-WIBE [1]	$(n+1)(\omega+3)\mathbb{G}$	$(\omega+1)\mathbb{G}$	$(n+1)\omega\mathbb{G}+\mathbb{G}_T$	$(\omega+1)$P	DBDH
Construction 1	$(2\omega+2)\mathbb{G}_1+\mathbb{G}_T$	$\mathbb{G}_2+\mathbb{Z}_p$	$(2\omega+1)\mathbb{G}_1+\mathbb{G}_T$	1P+1E	q-BDHI
Construction 2	$(2\omega+1)\mathbb{G}_1+2\mathbb{G}_2$	$\mathbb{G}_1+\mathbb{G}_2$	$2\omega\mathbb{G}_1+\mathbb{G}_2+\mathbb{G}_T$	2P	DBDH

Table 2: Comparison amongst the most efficient ABE schemes in the literature. Here we additionally compare the schemes by the family of predicates supported by the scheme (f), which can either be arbitrary Boolean formulas (Bool), zero inner-product predicates (InnerProd), or formulas in their DNF. We denote the number of disjunctive clauses in a DNF formula by γ.

ABE Scheme	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{sk}\rvert$	$\lvert\mathsf{c}\rvert$	$\mathsf{Decrypt}$	Assumption	f
CP-ABE [11]	$(2\mathbb{U}+3)\mathbb{G}_1+\mathbb{G}_T$	$(2\mathbb{U}+4)\mathbb{G}_2$	$(2\mathbb{U}+2)\mathbb{G}_2+\mathbb{G}_T$	4P+4$\mathbb{U}$E	SXDH	Bool
ZIPE [11]	$(2\mathbb{U}+4)\mathbb{G}_1+\mathbb{G}_T$	$4\mathbb{G}_2$	$(2\mathbb{U}+2)\mathbb{G}_2+\mathbb{G}_T$	4P+2$\mathbb{U}$E	SXDH	InnerProd
KP-ABE [16]	$(\mathbb{U}+1)\mathbb{G}+\mathbb{G}_T$	$2\mathbb{U}\mathbb{G}+\mathbb{U}^2\mathbb{G}$	$(\mathbb{U}+1)\mathbb{G}+\mathbb{G}_T$	2P+2$\mathbb{U}$E	$\mathbb{U}$-BDHE	Bool
Construction 1	$(\mathbb{U}+2)\mathbb{G}_1+\mathbb{G}_T$	$\mathbb{G}_2+\mathbb{Z}_p$	$\gamma((\mathbb{U}+1)\mathbb{G}_1+\mathbb{G}_T)$	1P+1E	q-BDHI	DNF
Construction 2	$(\mathbb{U}+1)\mathbb{G}_1+2\mathbb{G}_2$	$\mathbb{G}_1+\mathbb{G}_2$	$\gamma(2\mathbb{U}\mathbb{G}_1+\mathbb{G}_2+\mathbb{G}_T)$	2P	DBDH	DNF

2 Related Work

Identity-Based Encryption was first proposed by Shamir [22], and the first efficient realization was presented in the seminal work of Boneh and Franklin [8], where they suggested the usage of bilinear maps for cryptographic purposes. Canetti *et. al* [10] introduced the first construction that was provably secure without random oracles: the authors defined a slightly weaker security model (selective security) where the attacker is required to commit to the challenge identity prior to the beginning of the experiment. In the same settings, Boneh and Boyen [5] showed two efficient and practical schemes in the standard model. The first scheme with full security was presented by Boneh and Boyen [6] and later Waters [24] constructed a more efficient variant with an elegant security proof. Several other schemes have followed, such as [9]. It is worth mentioning that the notion of IBE has also been extended to support a hierarchical key-derivation structure [13,7].

Attribute-Based Encryption was envisioned by Sahai and Waters [21] as a generalization of IBE, where keys and ciphertext are generated under sets of attributes and it is possible to encode arbitrary access formulas. The concept of ABE was refined by Goyal *et al.* in [15], where the authors proposed two complementary notions: (i) Key-Policy ABE (KP-ABE) allows one to encode sets of attributes in ciphertexts and embed access formulas in users' secret keys, whereas in (ii) Ciphertext Policy ABE (CP-ABE) formulas are attached to the ciphertexts. Goyal *et al.* [15] described a selectively-secure construction of KP-ABE that allows polices to be expressed by any monotonic formula. The first efficient CP-ABE system was proposed by Bethencourt *et al.* [4] with a security proof in the generic group model, while the first CP-ABE scheme in the standard model is due to Waters [23]. In [14] Goyal *et al.* showed how to generically transform a KP-ABE into a CP-ABE. Until recently, all of the known attribute-based systems were proven secure only in the selective sense: a fully secure ABE was first proposed by Lewko *et al.* [18], leveraging the dual system encryption technique. In light of this, several efficient and adaptively-secure ABE schemes were recently proposed by Chen *et al.* [11] in the prime-order settings, constructed on a novel framework based on clever predicate encodings. ABE was further generalized as Predicate Encryption (PE) [17], where the ciphertext is required to hide the set of attributes associated to it, in addition to the message.

An ABE scheme with an efficient decryption algorithm was introduced by Attrapadung *et al.* [3], where the authors presented an ABE system with constant-size ciphertexts. As a result, the decryption algorithm requires a constant number of pairings. In this perspective, Hohenberger and Waters [16] improved this result with a scheme that computes only two pairings in the decryption algorithm. However, this comes at the cost of an increase in the size of the secret keys. A revocation system with small keys was proposed by Lewko *et al.* [19], along with an ABE system where the size of the secret keys grows linearly in the number of attributes. Finally, it is worth mentioning that Okamoto and Takashima provided an inner-product encryption scheme with constant-size keys [20] (later improved in [11]). However, the generic transformation from inner-products to arbitrary Boolean formulas introduces an overhead in the encoding of attributes exponential in the number of variables (see [17]), making this primitive less appealing for practical purposes.

On a different line of research, Abdalla *et al.* [1] proposed the notion of IBE with wildcards (WIBE): in this primitive, one is allowed to specify certain positions of the identity associated to a ciphertext that are not required to match with the secret key. A related notion was formalized and instantiated by Abdalla *et al.* in [2], where one can include wildcards in the key generation phase. Both of these works build on top of various Hierarchical IBE schemes and therefore inherit the long size of the keys, typically linear in the depth of the hierarchy.

Hence, this work improves the state-of-the-art by presenting the most efficient constructions in terms of key size *and* decryption operations supporting complex functionalities (beyond the simple IBE). We stress that in this work we consider only the notion of selective security.

3 Preliminaries

We denote by $\lambda \in \mathbb{N}$ the security parameter and by $\mathsf{poly}(\lambda)$ any function that is bounded by a polynomial in λ. We address any function that is *negligible* in the security parameter with $\mathsf{negl}(\lambda)$. We say that an algorithm is PPT if it is modelled as a probabilistic Turing machine whose running time is bounded by some function $\mathsf{poly}(\lambda)$. Given a set S, we denote by $x \leftarrow S$ the sampling of and element uniformly at random in S. For an arbitrary pair of binary strings (a, b) of the same length ℓ, we write $a \subseteq b$ if for all $i \in \{1, \ldots, \ell\}$ such that $a_i = 1$ then $b_i = 1$. Given a binary string a, we say that an index $i \in a$ if $a_i = 1$.

3.1 Bilinear Maps

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of large prime order p. Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be respective generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2$ be a function that maps pairs of elements $\in (\mathbb{G}_1, \mathbb{G}_2)$ to elements of some cyclic group $\mathbb{G}_T$ of order p. Throughout the following sections we write all of the group operations mutiplicatively, with identity elements denoted by 1. We further require that:

- The map e and all the group operations in $\mathbb{G}_2$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are efficiently computable.
- The map e is non degenerate, i.e., $e(g_1, g_2) \neq 1$.
- The map e is bilinear, i.e., $\forall u \in \mathbb{G}_1, \forall v \in \mathbb{G}_2, \forall (a, b) \in \mathbb{Z}_p^2, e(u^a, v^b) = e(u, v)^{ab}$.

3.2 Complexity Assumptions

In the following we formally define the Decisional q-Bilinear Diffie-Hellman Inversion assumption and the Decisional Bilinear Diffie-Hellman assumption. Both of the conjectures are widely used in pairing-based cryptographic constructions, among the others we mention the work of Boneh and Boyen [5]. We must point out that a sub-exponential attack is known for the former assumption [12], and therefore the security parameter of any scheme based on such a conjecture must be increased correspondingly. This, however, does not have a severe impact on the efficiency of the constructions, as discussed in [5].

Definition 1 (q-Decision-BDHI Assumption). *The q-Decision-BDHI assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if, for all* PPT *algorithms $\mathcal{A}$, there exists a negligible function* negl *such that*

$$\left| \begin{array}{r} \mathsf{Pr}\left[1 \leftarrow \mathcal{A}\left(g_1, g_1^x, g_2, g_2^x, \ldots, g_2^{x^q}, e(g_1, g_2)^{1/x}\right)\right] - \\ \mathsf{Pr}\left[1 \leftarrow \mathcal{A}\left(g_1, g_1^x, g_2, g_2^x, \ldots, g_2^{x^q}, T\right)\right] \end{array} \right| \leq \mathsf{negl}(\lambda)$$

where the probability is taken over the random choice of the generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, the random choice of $x \in \mathbb{Z}_p^*$, the random choice of $T \in \mathbb{G}_T$, and the random coins of $\mathcal{A}$.

Definition 2 (DBDH Assumption). *The DBDH assumption holds in* $(\mathbb{G}_1, \mathbb{G}_2)$ *if, for all* PPT *algorithms* $\mathcal{A}$*, there exists a negligible function* negl *such that*

$$\left| \begin{array}{r} \mathsf{Pr}\left[1 \leftarrow \mathcal{A}\left(g_1, g_1^a, g_1^b, g_1^c, g_2, g_2^a, g_2^b, g_2^c, e(g_1, g_2)^{abc}\right)\right] - \\ \mathsf{Pr}\left[1 \leftarrow \mathcal{A}\left(g_1, g_1^a, g_1^b, g_1^c, g_2, g_2^a, g_2^b, g_2^c, e(g_1, g_2)^{z}\right)\right] \end{array} \right| \leq \mathsf{negl}(\lambda)$$

where the probability is taken over the random choice of the generators $g_1 \in \mathbb{G}_1$ *and* $g_2 \in \mathbb{G}_2$*, the random choice of* $(a, b, c, z) \in (\mathbb{Z}_p^*)^4$*, and the random coins of* $\mathcal{A}$*.*

4 Subset Predicate Encryption

In this section, we formally introduce the concept of Subset Predicate Encryption. Our definition is very close to the standard Identity-Based Encryption, except that we do not necessarily require the string associated with the secret key to match the string embedded in the ciphertext. In fact, we allow anybody who owns a key for a string that matches any *subset* of the string of the ciphertext, to decrypt the latter.

Definition 3 (Subset Predicate Encryption). *A Subset Predicate Encryption (SPE) scheme consists of four* PPT *algorithms* Setup*,* KeyGen*,* Encrypt*, and* Decrypt *such that:*

$(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$. *The setup algorithm takes as input the security parameter* 1^λ *and a length parameter* n*. It outputs public parameters* pk *and the master secret key* msk*.*

$\mathsf{sk}_s \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{pk}, s)$. *The key-derivation algorithm takes as input the master secret key* msk*, public parameters* pk*, and a string* $s \in \{0, 1\}^n$*. It outputs a private key* sk_s*. We assume that* s *can be recovered from* sk_s*.*

$\mathsf{c} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}, s)$. *The encryption algorithm takes as input public parameters* pk*, a message* m*, and a string* $s \in \{0, 1\}^n$*. It outputs a ciphertext* c*. We assume that* s *can be recovered from* c*.*

$\mathsf{m} \leftarrow \mathsf{Decrypt}(\mathsf{sk}_s, \mathsf{pk}, \mathsf{c})$. *The decryption algorithm takes as input the private key* sk_s*, the public parameters* pk*, and a ciphertext* c*. It outputs a message* m *or a designated failure symbol* $\perp$*.*

Our notion of correctness for SPE is defined as follows:

Definition 4 (Correctness). *Correctness requires that for all security parameters* λ*, all* n*, all* $(\mathsf{pk}, \mathsf{msk})$ *output by* $\mathsf{Setup}(1^\lambda, 1^n)$*, all* $s' \in \{0, 1\}^n$*, all* s *such that* $s \subseteq s'$*, all* sk_s *output by* $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{pk}, s)$*, all* m*, and all* c *output by* $\mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}, s')$*, we have* $\mathsf{Decrypt}(\mathsf{sk}_s, \mathsf{pk}, \mathsf{c}) = \mathsf{m}$*.*

Security. In the following, we define the security model for SPE schemes. Informally, the adversary should be unable to learn anything about the content of a ciphertext associated with some set s^* even if it has obtained secret keys corresponding to arbitrary sets $s_1, \dots, s_q$, so long as none of those satisfies $s_i \subseteq s^*$. Our definition corresponds to "selective" security, whereby the attacker is required to commit to the s^* that he wants to be challenged on before seeing the public parameters of the scheme. Alternatively one could consider the stronger "adaptive" notion, where the challenge set is revealed by the adversary only in the challenge phase.

Consider the following experiment parameterized by λ:

1. The attacker specifies a universe of elements $\{0,1\}^n$ (i.e., a bound n on the size of the universe) and a challenge set $s^* \in \{0,1\}^n$.
2. $\mathsf{Setup}(1^\lambda, 1^n)$ is run to obtain $(\mathsf{pk}, \mathsf{msk})$, and the adversary is given pk.
3. The adversary is allowed to query for private keys for arbitrary sets $s_1, \dots, s_q$ such that for all $i \in \{1, \dots q\}$ it holds that $s_i \not\subseteq s^*$.
4. The adversary outputs a message pair $(\mathsf{m}_0, \mathsf{m}_1)$ with $|\mathsf{m}_0| = |\mathsf{m}_1|$. A uniform bit $\mathsf{b} \in \{0,1\}$ is chosen, and the ciphertext $\mathsf{c} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}_b, s^*)$ is computed and given to the adversary.
5. The adversary may continue to request private keys for arbitrary sets, subject to the same restriction as before.
6. Finally, the adversary outputs a guess b' for b.

The advantage of the adversary in this experiment is defined as $|\mathsf{Pr}[\mathsf{b}' = \mathsf{b}] - \frac{1}{2}|$.

Definition 5 (Selective Security). *A SPE scheme is* selectively secure *if the advantage of any* PPT *adversary in the above experiment is negligible.*

4.1 Generic Instantiations

Before presenting our schemes we first describe some potential approaches to instantiate Subset Predicate Encryption and we show their drawbacks.

SPE from PE. One can instantiate SPE from any predicate encryption for inner products as follows: Given a universe of n attributes, keys for a set s are associated with the binary vector $(s_1, \dots, s_n)$. A ciphertext for a set s' is encrypted under the vector $(s'_1 \oplus 1, \dots, s'_n \oplus 1)$. The inner product of the two vectors is 0 if and only if $s \subseteq s'$, therefore correctness and security follow. This instantiation however, inherently generates ciphertexts and secret keys that grow linearly with the size of the universe of elements.

SPE from WIBE. We observe that if we encode a set s in a ciphertext as a string where the 1s are substituted with the wildcard symbol, then an IBE with wildcards supports the same functionality as a SPE. Given the current state-of-the-art for WIBE schemes, this approach suffers from the same drawbacks as described above.

SPE from Fuzzy IBE. It is an easy exercise to instantiate SPE from the Fuzzy Identity-Based Encryption of Sahai and Waters [21]: Setting the degree d of the polynomial associated with the secret key to be equal to the number of components of the key itself, one can ensure that the decryptor needs to use all of the components of the secret key in order to decrypt a ciphertext. It follows that a key associated with a string s can decrypt any ciphertext encrypted under any s' such that $s \subseteq s'$. However the decryption algorithm requires to interpolate the polynomial in the exponent, which incurs in one pairing operation per element associated with the secret key. Additionally, the size of the key grows linearly with the number of elements associated with it, which is, in the average case, linear in the security parameter. Our observation is that our primitive does not require the flexibility of a Fuzzy IBE scheme, and therefore we can hope to achieve better performance at the cost of sacrificing the malleability in the manipulation of the secret keys.

5 Our Constructions

In this section we present our two instantiations from bilinear maps.

5.1 First Scheme

In the following we describe our first construction, inspired by second scheme presented in [5]. The key difference is that our ciphertexts is composed by disjoint components, each corresponding to an element of the public parameters. This additional flexibility allows one to choose an arbitrary subset of elements in the decryption phase.

Construction 1. Our first construction consists of the following algorithms.

$\mathsf{Setup}(1^\lambda, 1^n)$: To generate the SPE system given a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$, with respective generators (g_1, g_2), the setup algorithm selects a random generator $h \in \mathbb{G}_2$ and it computes $v = e(g_1, h)$. Then it samples a random $x_0 \in \mathbb{Z}_p^*$ and a random vector $(x_1, \dots, x_n) \in (\mathbb{Z}_p^*)^n$ and sets $X_0 = g_1^{x_0}$ and for all $i \in \{1, \dots, n\} : X_i = g_1^{x_i}$. The public parameters pk and the master secret key are given by

$$\mathsf{pk} = (g_1, X_0, X_1, \dots, X_n, v) \in \mathbb{G}_1^{n+2} \times \mathbb{G}_T$$

$$\mathsf{msk} = (x_0, x_1, \dots, x_n, h) \in (\mathbb{Z}_p^*)^{n+1} \times \mathbb{G}_2$$

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{pk}, s)$: To generate the private key associated with the set s, the key generation algorithm picks a random $\kappa \in \mathbb{Z}_p$ such that $\sum_{i \in s} x_i + \kappa x_0 \neq 0 \mod p$ and computes $K = h^{\frac{1}{\sum_{i \in s} x_i + \kappa x_0}}$. The private key is defined as

$$\mathsf{sk}_s = (\kappa, K) \in \mathbb{Z}_p \times \mathbb{G}_2$$

$\mathsf{c} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}, s)$: The encryption of a message $m \in \mathbb{G}_T$ for a given set s is done by picking a random $\rho \in \mathbb{Z}_p^*$ and returning the following ciphertext

$$\mathsf{c} = (m \cdot v^\rho, X_0^\rho, \forall_{i \in s} : X_i^\rho) \in \mathbb{G}_T \times \mathbb{G}_1^{|s|+1}$$

$\mathsf{m} \leftarrow \mathsf{Decrypt}(\mathsf{sk}_s, \mathsf{pk}, \mathsf{c})$. To decrypt a ciphertext $\mathsf{c} = (A, B, C_1, \ldots, C_\ell)$, for some positive integer $\ell \leq n$, using the private key $\mathsf{sk}_s = (\kappa, K)$, return

$$\frac{A}{e\left(B^{\kappa} \prod_{i \in s} C_i, K\right)}$$

To check that the system is consistent it is enough to observe that, for a valid private key sk_s and a valid ciphertext encoded under a string s' such that $s \subseteq s'$, there always exists an element C_i for all $i \in s$, thus we have

$$\begin{aligned}
\frac{A}{e\left(B^{\kappa} \prod_{i \in s} C_i, K\right)} &= \frac{A}{e\left((X_0^{\rho})^{\kappa} g_1^{\rho \cdot \sum_{i \in s} x_i}, h^{\frac{1}{\sum_{i \in s} x_i + \kappa x_0}}\right)} \\
&= \frac{m \cdot v^{\rho}}{e\left(g_1^{\rho \cdot (\kappa x_0 + \sum_{i \in s} x_i)}, h^{\frac{1}{\sum_{i \in s} x_i + \kappa x_0}}\right)} \\
&= \frac{m \cdot e(g_1, h)^{\rho}}{e(g_1, h)^{\rho}} \\
&= m
\end{aligned}$$

Here we elaborate the formal guarantees of our construction. The security proof is non-trivial as our reduction is required to include in the challenge ciphertext each group element separately (as opposed to their product), this arises subtle issues in the generation of the secret key that we address in the following.

Theorem 1. *Assume that the q-Decision-BDHI assumption holds in groups $(\mathbb{G}_1, \mathbb{G}_2)$ of size p. Then Construction 1 is a selectively-secure SPE scheme.*

Proof. Assume towards contradiction that there exists an adversary $\mathcal{A}$ that has advantage $\epsilon(\lambda)$ in attacking the SPE system, for some non negligible function $\epsilon(\lambda)$. Then we can construct the following reduction $\mathcal{R}$ against the q-Decision-BDHI assumption in $(\mathbb{G}_1, \mathbb{G}_2)$.

The reduction $\mathcal{R}$ takes as input a tuple $\left(g_1, g_1^{\alpha}, g_2, g_2^{\alpha}, \ldots, g_2^{\alpha^q}, T\right)$, where T is either $e(g_1, g_2)^{1/\alpha}$ or a random element of $\mathbb{G}_T$. The algorithm $\mathcal{R}$ interacts with $\mathcal{A}$ in the selective-security game as follows:

Preparation: The reduction $\mathcal{R}$ samples a vector $(w_0, \ldots, w_{q-1}) \in (\mathbb{Z}_p^*)^q$, let

$$f(\alpha) = w_0 \prod_{j=1}^{q-1} (\alpha + w_j) = \sum_{j=0}^{q-1} c_j \alpha^j$$

for some coefficients c_j where $c_0 \neq 0$. The algorithm sets $h = \prod_{j=1}^{q-1} \left(g_2^{\alpha^j}\right)^{c_j} = g_2^{f(\alpha)}$. The variable w_0 ensures that h is a uniformly distributed generator of $\mathbb{G}_2$. Note that we can assume that $h \neq 1$ otherwise it must have been the case that there exists a $j \in \{1 \ldots q-1\}$ such that $w_j = -\alpha$ and thus the algorithm can efficiently output a solution to the decisional problem. We observe that for all $j \in \{1 \dots q-1\}$ it is easy for $\mathcal{R}$ to compute the tuple $\left(w_j, h^{\frac{1}{\alpha+w_j}}\right)$ by considering

$$\frac{f(\alpha)}{(\alpha + w_j)} = \sum_{j=0}^{q-2} d_j \alpha^j$$

and setting $h^{\frac{1}{\alpha+w_j}} = g_2^{\frac{f(\alpha)}{(\alpha+w_j)}} = \prod_{j=0}^{q-2} \left(g_2^{\alpha^j}\right)^{d_j}$. Additionally, the reduction $\mathcal{R}$ computes

$$T_h = T^{c_0} \cdot \prod_{j=1}^{q-1} e\left(g_1, g_2^{c_j \alpha^{j-1}}\right)$$

It is easy to see that whenever T is uniformly distributed in $\mathbb{G}_T$ then so is T_h, whereas whenever $T = e(g_1, g_2)^{1/\alpha}$ then $T_h = e(g_1, h)^{1/\alpha}$.

Initialization: The experiment begins with $\mathcal{A}$ outputting bound n on the universe of elements and a challenge set $s^* \in \{0,1\}^n$.

Setup: To generate the public parameters, the algorithm $\mathcal{R}$ proceeds by uniformly sampling for all $i \in s^*$ an element $a_i \in \mathbb{Z}_p^*$ and setting $X_i = g_1^{x_i} = g_1^{-a_i \cdot \alpha}$. For all $i \in \overline{s^*}$ the reduction picks a pair $(a_i, b_i) \in (\mathbb{Z}_p^*)^2$ and sets $X_i = g_1^{x_i} = g_1^{-a_i \cdot (\alpha + b_i)}$. The public parameters provided to the adversary are

$$(g_1, X_0 = g_1^{\alpha}, X_1, \dots, X_n, v = e(g, h))$$

where h is defined as specified above. We remark that h is a uniformly distributed element in $\mathbb{G}_T$. Since all of the other elements of the public parameters are uniformly distributed over $\mathbb{G}_1$ to the view of the adversary, we can conclude that the public parameters are correctly distributed according to our construction.

Phase 1: The adversary can issue up to $q-1$ private key queries for some sets s^j under the constraint that for all $j \in \{1, \dots, q-1\}$ it holds that $s^j \not\subseteq s^*$. The algorithm $\mathcal{R}$ responds to each query j as follows: let $\left(w_j, h^{\frac{1}{\alpha+w_j}}\right)$ the j-th pair constructed in the preparation phase, the reduction computes an $r \in \mathbb{Z}_p$ that satisfies

$$\left(r - \sum_{i \in s^j} a_i\right)(\alpha + w_j) = -\alpha \sum_{i \in s^j} a_i - \sum_{i \in s^j \cap \overline{s^*}} a_i b_i + \alpha r$$

Expanding the equation we obtain

$$r = \sum_{i \in s^j} a_i - \frac{\sum_{i \in s^j \cap \overline{s^*}} a_i b_i}{w_j}$$

Note that the unknown α cancels out of the equation and the algorithm can evaluate the expression. The secret key for the set s^j is set to be

$$\mathsf{sk}_{s^j} = \left(r, h^{\frac{1}{(\alpha+w_j)(r - \sum_{i \in s^j} a_i)}}\right).$$

We note that the key is functional, as

$$h^{\frac{1}{(\alpha+w_j)(r-\sum_{i\in s^j} a_i)}} = h^{\frac{1}{-\alpha\sum_{i\in s^j} a_i - \sum_{i\in s^j\cap\overline{s^*}} a_i b_i + \alpha r}} = h^{\frac{1}{\sum_{i\in s^j} x_i + \alpha r}}$$

it allows the adversary to decrypt the ciphertexts that he is intended to. To argue about the correct distribution of the key it is enough to observe that the value w_j is sampled uniformly at random from $\mathbb{Z}_p^*$, therefore whenever $\sum_{i\in s^j\cap\overline{s^*}} a_i b_i \neq 0$ then r is a uniformly distributed element of $\mathbb{Z}_p$. First we point out that the set $s^j \cap \overline{s^*}$ is never empty due to the non-triviality of the game, i.e., $s^j \not\subseteq s^*$, secondly we observe that the expression $\sum_{i\in s^j\cap\overline{s^*}} a_i b_i$ can return at most 2^n different results, due to the total number of elements' combinations. Therefore by choosing a large enough size of p, e.g. $2^{2\cdot n}$, we ensure that the probability of $\sum_{i\in s^j\cap\overline{s^*}} a_i b_i$ returning 0 is negligible in the security parameter (recall that for all $i \in \{1,\dots,n\}$ it holds that a_i and b_i are elements uniformly distributed in $\mathbb{Z}_p^*$). For completeness, we note that this procedure will fail to produce a private key for an $s \subseteq s^*$ since in that case we obtain $r = \sum_{i\in s^j} a_i$ and therefore $h^{\frac{1}{(\alpha+w_j)(r-\sum_{i\in s^j} a_i)}} = h^{\frac{1}{(\alpha+w_j)\cdot 0}}$.

Challenge: The adversary outputs two messages $(m_0, m_1) \in \mathbb{G}_T^2$. The reduction $\mathcal{R}$ samples a random $\mathsf{b} \in \{0,1\}$ and a random $z \in \mathbb{Z}_p^*$ and hands over to the attacker the challenge ciphertext

$$\mathsf{c}^* = \left(m_{\mathsf{b}} \cdot T_h^z, g_1^z, \forall_{i\in s^*} : g_1^{-a_i z}\right)$$

Consider $\rho = z/\alpha$. We shall note that whenever $T_h = e(g_1, h)^{1/\alpha}$ then c^* is a valid ciphertext as

$$\begin{aligned}
&m_{\mathsf{b}} \cdot T_h^z = m_{\mathsf{b}} \cdot e(g_1, h)^{z/\alpha} = m_{\mathsf{b}} \cdot v^\rho \\
&g_1^z = \left(X_0^{1/\alpha}\right)^z = X_0^\rho \\
&\forall_{i\in s^*} : g_1^{-a_i z} = \left(X_i^{1/\alpha}\right)^z = X_i^\rho
\end{aligned}$$

On the other hand, whenever T_h is uniform in $\mathbb{G}_T$, then the message m_{b} is hidden from the view of the adversary in an information theoretic sense.

Phase 2: The adversary can issue additional private key queries for a total of at most $q-1$. The reduction answer as specified in Phase 1.

Guess: The adversary outputs a guess b' and the reduction returns $\mathsf{b} = \mathsf{b}'$ to the challenger.

As argued above, when the input tuple contains a $T = (g_1, g_2)^{1/\alpha}$, then the view of the adversary perfectly resembles the inputs that he is expecting in the standard experiment for SPE security. It follows that the advantage of the adversary is, as assumed, greater than some non negligible $\epsilon(\lambda)$. On the other hand, when the input tuple contains a T uniformly distributed in $\mathbb{G}_T$, then the view of the adversary contains no information about the secret bit b. Thus in this case $\mathcal{A}$ cannot do better than guessing. It follows that

$$\left| \begin{array}{r} \mathsf{Pr}\left[1 \leftarrow \mathcal{R}\left(g_1, g_1^x, g_2, g_2^x, \ldots, g_2^{x^q}, e(g_1, g_2)^{1/x}\right)\right] - \\ \mathsf{Pr}\left[1 \leftarrow \mathcal{R}\left(g_1, g_1^x, g_2, g_2^x, \ldots, g_2^{x^q}, T\right)\right] \end{array} \right| \geq \\ |1/2 + \epsilon(\lambda) - 1/2| = \epsilon(\lambda)$$

This represents a contradiction to the q-Decision-BDHI assumption and it concludes our proof. □

5.2 Second Scheme

Our second scheme can be seen as a descendant of the celebrated IBE of Waters [24]. On a very high-level, our main observation is that the scheme satisfies our notion of security if the group elements of the ciphertext are not multiplied together. This change, together with our different notion of security, forces us to develop a different proof strategy.

Construction 2. Our second construction consists of the following algorithms.

$\mathsf{Setup}(1^\lambda, 1^n)$: To generate the SPE system given a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ with respective generators (g_1, g_2), the setup algorithm selects a random $\alpha \in \mathbb{Z}_p^*$ and sets $h = g_2^\alpha$. Then it samples a random vector $(x_1, \ldots, x_n) \in (\mathbb{Z}_p^*)^n$ and sets for all $i \in \{1, \ldots, n\} : X_i = g_1^{x_i}$. The public parameters pk and the master secret key msk are given by

$$\mathsf{pk} = (g_1, X_1, \ldots, X_n, g_2, h) \in \mathbb{G}_1^{n+1} \times \mathbb{G}_2^2$$

$$\mathsf{msk} = g_1^\alpha \in \mathbb{G}_1$$

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{pk}, s)$: To generate the private key associated with the set s, the key generation algorithm picks a random $r \in \mathbb{Z}_p$ and defines the private key as

$$\mathsf{sk}_s = \left(g_1^\alpha \left(\prod_{i \in s} X_i \right)^r, g_2^r \right) \in \mathbb{G}_1 \times \mathbb{G}_2$$

$\mathsf{c} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{m}, s)$: The encryption of a message $m \in \mathbb{G}_T$ for a given set s is done by picking a random $\rho \in \mathbb{Z}_p^*$ and returning the following ciphertext

$$\mathsf{c} = (m \cdot e(g_1, h)^\rho, g_2^\rho, \forall_{i \in s} : X_i^\rho) \in \mathbb{G}_T \times \mathbb{G}_2 \times \mathbb{G}_1^{|s|}$$

$\mathsf{m} \leftarrow \mathsf{Decrypt}(\mathsf{sk}_s, \mathsf{pk}, \mathsf{c})$. To decrypt a ciphertext $\mathsf{c} = (A, B, C_1, \ldots, C_\ell)$, for some positive integer $\ell \leq n$, using the private key $\mathsf{sk}_s = (K, R)$, return

$$A \cdot \frac{e\left(\prod_{i \in s} C_i, R\right)}{e(K, B)}$$

To check that the system is correct we observe, as before, that

$$\begin{aligned}
A \cdot \frac{e\left(\prod_{i\in s} C_i, R\right)}{e(K,B)} &= m \cdot e(g_1,h)^{\rho} \cdot \frac{e\left(\prod_{i\in s} X_i^{\rho}, g_2^r\right)}{e\left(g_1^{\alpha}\left(\prod_{i\in s} g_1^{x_i}\right)^r, g_2^{\rho}\right)} \\
&= m \cdot e(g_1,g_2)^{\alpha\rho} \cdot \frac{e\left(\prod_{i\in s} X_i^{\rho}, g_2^r\right)}{e(g_1^{\alpha}, g_2^{\rho}) e\left(\left(\prod_{i\in s} g_1^{x_i}\right)^r, g_2^{\rho}\right)} \\
&= m \cdot e(g_1,g_2)^{\alpha\rho} \cdot \frac{e\left(\prod_{i\in s} X_i, g_2\right)^{r\rho}}{e(g_1,g_2)^{\alpha\rho} e\left(\prod_{i\in s} X_i, g_2\right)^{r\rho}} \\
&= m
\end{aligned}$$

The construction above is a secure SPE scheme if the DBDH assumption holds.

Theorem 2. *Assume that the DBDH assumption holds in groups* $(\mathbb{G}_1, \mathbb{G}_2)$ *of size* p*. Then Construction 2 is a selectively-secure SPE scheme.*

Proof. Assume towards contradiction that there exists an adversary $\mathcal{A}$ that has advantage $\epsilon(\lambda)$ in attacking the SPE system, for some non negligible function $\epsilon(\lambda)$. Then we can construct the following reduction $\mathcal{R}$ against the DBDH assumption in $(\mathbb{G}_1, \mathbb{G}_2)$.
The reduction $\mathcal{R}$ takes as input a tuple $(g_1, A_1, B_1, C_1, g_2, A_2, B_2, C_2, Z)$, where Z is either $e(g_1,g_2)^{abc}$ or a random element of $\mathbb{G}_T$. The algorithm $\mathcal{R}$ interacts with $\mathcal{A}$ in the selective-security game as follows:

<u>Initialization</u>: The experiment begins with $\mathcal{A}$ outputting bound n on the universe of elements and a challenge set $s^* \in \{0,1\}^n$.

<u>Setup</u>: To generate the public parameters, the algorithm $\mathcal{R}$ proceeds by uniformly sampling for all $i \in s^*$ an pair of elements $y_i \in \mathbb{Z}_p^*$ and setting $X_i = g_1^{y_i}$. For all $i \in \overline{s^*}$ the reduction picks a pair $(y_i, w_i) \in (\mathbb{Z}_p^*)^2$ and sets $X_i = A_1^{w_i} g_1^{y_i} = g_1^{a\cdot w_i + y_i}$. The public parameters provided to the adversary are

$$(A_1, X_1, \ldots, X_n, g_2, B_2)$$

Since all of the elements of the public parameters are uniformly distributed over the corresponding group to the view of the adversary, we can conclude that the public parameters are correctly distributed according to our construction.

<u>Phase 1</u>: The adversary can issue up to $q-1$, for some polynomial bound q, private key queries for some sets s^j under the constraint that for all $j \in \{1,\ldots,q-1\}$ it holds that $s^j \not\subseteq s^*$. The algorithm $\mathcal{R}$ responds to each query j as follows: the reduction samples an $r \in \mathbb{Z}_p$ and sets

$$sk_{s^j} = \left(B_1^{\frac{-\sum_{i\in s^j} y_i}{\sum_{i\in s^j} w_i}} \cdot \left(\prod_{i\in s^j} X_i' \right)^r, B_2^{\frac{-1}{\sum_{i\in s^j} w_i}} \cdot g_2^r \right)$$

We observe that

$$\begin{aligned}
sk_{s^j} &= \left(g_1^{b\cdot\frac{-\sum_{i\in s^j} y_i}{\sum_{i\in s^j} w_i}+r\left(\sum_{i\in s^j} aw_i+y_i\right)}, g_2^{r-\frac{b}{\sum_{i\in s^j} w_i}} \right)\\
&= \left(g_1^{ab+b\cdot\left(\frac{-\sum_{i\in s^j} y_i}{\sum_{i\in s^j} w_i}-a\right)+r\left(\sum_{i\in s^j} aw_i+y_i\right)}, g_2^{r-\frac{b}{\sum_{i\in s^j} w_i}} \right)\\
&= \left(g_1^{ab-\frac{b}{\sum_{i\in s^j} w_i}\cdot\left(\sum_{i\in s^j} y_i+a\cdot\sum_{i\in s^j} w_i\right)+r\left(\sum_{i\in s^j} aw_i+y_i\right)}, g_2^{r-\frac{b}{\sum_{i\in s^j} w_i}} \right)\\
&= \left(g_1^{ab+\left(\sum_{i\in s^j} aw_i+y_i\right)\left(r-\frac{b}{\sum_{i\in s^j} w_i}\right)}, g_2^{r-\frac{b}{\sum_{i\in s^j} w_i}} \right)\\
&= \left(g_1^{ab}\cdot\left(\prod_{i\in s^j} X_i\right)^{\left(r-\frac{b}{\sum_{i\in s^j} w_i}\right)}, g_2^{r-\frac{b}{\sum_{i\in s^j} w_i}} \right)
\end{aligned}$$

which gives us a functional and correctly distributed key, as r is uniformly distributed over $\mathbb{Z}_p^*$. For completeness we note that the procedure fails whenever $\sum_{i\in s^j} w_i = 0$, which happens only in case the queried set s^j is a subset of the challenge set s^*, except with negligible probability (for a large enough p).

Challenge: The adversary outputs two messages $(m_0, m_1) \in \mathbb{G}_T^2$. The reduction $\mathcal{R}$ samples a random $\mathsf{b} \in \{0,1\}$ and hands over to the attacker the challenge ciphertext

$$\mathsf{c}^* = (m_{\mathsf{b}} \cdot Z, C_2, \forall_{i\in s^*} : C_1^{y_i})$$

We shall note that whenever $Z = e(g_1, h)^{abc}$ then c^* is a valid ciphertext as

$$\begin{aligned}
&m_{\mathsf{b}} \cdot Z = m_{\mathsf{b}} \cdot e(g_1, g_2)^{abc} = m_{\mathsf{b}} \cdot e(A_1, B_2)^c\\
&C_2 = g_2^c\\
&\forall_{i\in s^*} : C_1^{y_i} = (g_1^{y_i})^c = X_i^c
\end{aligned}$$

On the other hand, whenever Z is uniform in $\mathbb{G}_T$, then the message m_{b} is hidden from the view of the adversary in an information theoretic sense.

Phase 2: The adversary can issue additional private key queries for a total of at most $q-1$. The reduction answer as specified in Phase 1.

Guess: The adversary outputs a guess b' and the reduction returns $\mathsf{b} = \mathsf{b}'$ to the challenger.

As argued above, when the input tuple contains a $Z = (g_1, g_2)^{abc}$, then the view of the adversary perfectly resembles the inputs that he is expecting in the standard experiment for SPE security. It follows that the advantage of the adversary is, as assumed, greater than some non negligible $\epsilon(\lambda)$. On the other hand, when the input tuple contains a Z uniformly distributed in $\mathbb{G}_T$, then the view of the adversary contains no information about the secret bit b. Thus in this case $\mathcal{A}$ cannot do better than guessing. It follows that

$$\begin{aligned}\Big| \mathsf{Pr}\left[1 \leftarrow \mathcal{A}\left(g_1, g_1^a, g_1^b, g_1^c, g_2, g_2^a, g_2^b, g_2^c, e(g_1, g_2)^{abc}\right)\right] - \\ \mathsf{Pr}\left[1 \leftarrow \mathcal{A}\left(g_1, g_1^a, g_1^b, g_1^c, g_2, g_2^a, g_2^b, g_2^c, e(g_1, g_2)^{z}\right)\right] \Big| \geq \\ |1/2 + \epsilon(\lambda) - 1/2| = \epsilon(\lambda)\end{aligned}$$

This represents a contradiction to the DBDH assumption and it concludes our proof. □

Large Universe Construction. We note that we can extend our second construction to support elements that were not considered in the setup phase, assuming the existence of a random oracle. Assume that all parties have access to the function $H : \{0,1\}^* \to \mathbb{G}_1$, we can remove the group elements from the public parameters and substitute them with the description of H. This extension yields a scheme with *constant-size* public parameters for an *exponentially-large* universe of elements.

6 Generic Transformations

In the following we describe some black-box transformations from SPE to well known cryptographic primitives.

Identity Based Encryption. As an easy warm up we show how to deploy SPE in order to achieve standard Identity Based Encryption (IBE). Although not surprising, this will guide us through the subsequent transformations. We first initialize the system by running the Setup algorithm with a length parameter of $2 \cdot n$ and the corresponding security parameter λ. The KeyGen algorithm, on input $\mathsf{ID} \in \{0,1\}^n$, generates $s \in \{0,1\}^{2 \cdot n}$ by setting, for all $i \in \{1, \ldots, 2 \cdot n\}$:

$$s_i = \begin{cases} 1 - \mathsf{ID}_{i/2} & \text{if } i = 0 \\ \mathsf{ID}_{(i+1)/2} & \text{if } i = 1 \end{cases} \pmod 2$$

Then the standard KeyGen algorithm is executed on s and the corresponding output is returned. The same modification is applied to the $\mathsf{Encrypt}$ algorithm.

To better visualize this transformation one can imagine the Setup algorithm to return two arrays of elements $(x_1^0, \ldots, x_n^0)$ and $(x_1^1, \ldots, x_n^1)$. The identities ID in the set $\{0,1\}^n$ index the binary choice of each element $x_i^{\mathsf{ID}_i}$ between the two arrays. It is important to note that all of the valid sets contain the same amount of elements, i.e. n many, and that any two sets differ in at least one position. This implies that no valid identity is a subset of any other and the security of the IBE scheme follows from the security of the underlying SPE.

Identity Based Encryption with Wildcards. Here show how to modify our primitive in a black-box fashion to handle *wildcards* in both the ciphertexts and the keys: this allows us to specify certain positions of the identity encoded in the ciphertext (in the key, respectively) that are not required to match the key (the ciphertext, respectively) for the decryption to be successful. IBE schemes that allow for wildcards in the ciphertexts are known in the literature as WIBE [1], whereas schemes that support wildcards in the keys are called Wicked IBE [2]. We stress that, as opposed to the original proposals, our generic transformation does not support a hierarchical structure of identities, since it is not clear how to delegate keys in the general settings. In the following we describe how to modify the $\mathsf{Encrypt}$ and the KeyGen algorithms to handle wildcards. We denote the wildcard with the distinguished symbol $*$.

We first initialize the system by running the Setup algorithm with a parameter of $2 \cdot n$ and the corresponding security parameter λ. The $\mathsf{Encrypt}$ algorithm is modified to take as input $\mathsf{ID} \in \{0,1,*\}^n$ and generate $s \in \{0,1\}^{2 \cdot n}$ as follows:

$$s_i = \begin{cases} 1 - \mathsf{ID}_{i/2} & \text{if } \mathsf{ID}_{i/2} \in \{0,1\} \land i = 0 \\ \mathsf{ID}_{(i+1)/2} & \text{if } \mathsf{ID}_{(i+1)/2} \in \{0,1\} \land i = 1 \\ 1 & \text{if } \mathsf{ID}_{i/2} = * \land i = 0 \\ 1 & \text{if } \mathsf{ID}_{(i+1)/2} = * \land i = 1 \end{cases} \pmod 2$$

for all $i \in \{1, \ldots, 2 \cdot n\}$. As before, if we consider Setup to output two vectors $(x_1^0, \ldots, x_n^0)$ and $(x_1^1, \ldots, x_n^1)$, then the identities $\mathsf{ID} \in \{0,1,*\}^n$ represent the binary choice over the elements of the two vectors except when $s_i = *$, in which case both x_i^0 and x_i^1 are included in the set. We observe that the decryption is successful whenever one owns a key that encodes a subset of s, which matches the policy enforced by the WIBE scheme. Therefore the security of the SPE carries over.

We can encode wildcards in the decryption keys applying a similar modification to the KeyGen algorithm, that differs in assigning $s_i = 0$, as opposed to 1, whenever the corresponding bit of ID is $*$. We note that the two modifications are not mutually exclusive and can coexist for a hybrid of the two approaches.

Ciphertext Policy Attribute Based Encryption. Perhaps the most interesting feature of our primitive is that it can be used to obtain a Ciphertext-Policy Attribute Based Encryption (CP-ABE) scheme for a small universe of attributes. The transformation is as follows. Fix a universe of attributes $\mathbb{U}$ of size n, we uniquely assign to each attribute $a \in \mathbb{U}$ an index $i \in \{1, \ldots, n\}$. The private key associated with a set of attributes A will be the key associated with the set $\mathbb{U} \backslash \mathsf{A}$. Specifically, we construct the private key for A by executing KeyGen on input s^{A}, where

$$s_i^{\mathsf{A}} = \begin{cases} 0 & \text{if } a_i \in \mathsf{A} \\ 1 & \text{if } a_i \notin \mathsf{A} \end{cases}$$

To encrypt a message m using a DNF formula $C_1 \lor \cdots \lor C_t$, where each C_j represents a conjunction over some subset of the attributes, the sender processes each of the t clauses independently. For the ith clause C_j, the sender encrypts the message running $\mathsf{Encrypt}$ on input s^{C_j}, where

$$s_i^{C_j} = \begin{cases} 0 & \text{if } a_i \in C_j \\ 1 & \text{if } a_i \notin C_j \end{cases}$$

The algorithm returns the concatenation of the ciphertexts corresponding to each clause. To decrypt, the receiver finds some clause C_j that is satisfied by his attributes A. Note that this means $C_j \subseteq \mathsf{A}$, or equivalently $\mathbb{U} \setminus \mathsf{A} \subseteq \mathbb{U} \setminus C_j$. Thus, the receiver will be able to decrypt the ciphertext corresponding to that clause if and only if its key is associated with a set s such that $s \subseteq s^{C_j}$.

Acknowledgements This research is based upon work supported by the German research foundation (DFG) through the collaborative research center 1223, by the German Federal Ministry of Education and Research (BMBF) through the project PROMISE (16KIS0763), and by the state of Bavaria at the Nuremberg Campus of Technology (NCT). NCT is a research cooperation between the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Technische Hochschule Nürnberg Georg Simon Ohm (THN). We thank the anonymous reviewers for their valuable comments that helped to improve our paper. We thank Vincenzo Iovino for the insightful discussions on this work.

References

1. Michel Abdalla, Dario Catalano, Alex Dent, John Malone-Lee, Gregory Neven, and Nigel Smart. Identity-based encryption gone wild. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 300–311, Venice, Italy, July 10–14, 2006. Springer, Heidelberg, Germany.
2. Michel Abdalla, Eike Kiltz, and Gregory Neven. Generalized key delegation for hierarchical identity-based encryption. In Joachim Biskup and Javier López, editors, *ESORICS 2007: 12th European Symposium on Research in Computer Security*, volume 4734 of *Lecture Notes in Computer Science*, pages 139–154, Dresden, Germany, September 24–26, 2007. Springer, Heidelberg, Germany.
3. Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 90–108, Taormina, Italy, March 6–9, 2011. Springer, Heidelberg, Germany.
4. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334, Oakland, CA, USA, May 20–23, 2007. IEEE Computer Society Press.
5. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

6. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany.
7. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.
8. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.
9. Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th Annual Symposium on Foundations of Computer Science*, pages 647–657, Providence, RI, USA, October 20–23, 2007. IEEE Computer Society Press.
10. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.
11. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 595–624, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.
12. Jung Hee Cheon. Security analysis of the strong Diffie-Hellman problem. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.
13. Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566, Queenstown, New Zealand, December 1–5, 2002. Springer, Heidelberg, Germany.
14. Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. Bounded ciphertext policy attribute based encryption. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008: 35th International Colloquium on Automata, Languages and Programming, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 579–591, Reykjavik, Iceland, July 7–11, 2008. Springer, Heidelberg, Germany.
15. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309.
16. Susan Hohenberger and Brent Waters. Attribute-based encryption with fast decryption. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 162–179, Nara, Japan, February 26 – March 1, 2013. Springer, Heidelberg, Germany.

17. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany.
18. Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.
19. Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *2010 IEEE Symposium on Security and Privacy*, pages 273–285, Berkeley/Oakland, CA, USA, May 16–19, 2010. IEEE Computer Society Press.
20. Tatsuaki Okamoto and Katsuyuki Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In Dongdai Lin, Gene Tsudik, and Xiaoyun Wang, editors, *CANS 11: 10th International Conference on Cryptology and Network Security*, volume 7092 of *Lecture Notes in Computer Science*, pages 138–159, Sanya, China, December 10–12, 2011. Springer, Heidelberg, Germany.
21. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.
22. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1984. Springer, Heidelberg, Germany.
23. Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70, Taormina, Italy, March 6–9, 2011. Springer, Heidelberg, Germany.
24. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.