Fully Secure Attribute-Based Systems with Short Ciphertexts/Signatures and Threshold Access Structures

Cheng Chen[1], Jie Chen[2], Hoon Wei Lim[2], Zhenfeng Zhang[1], Dengguo Feng[1], San Ling[2], and Huaxiong Wang[2]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
{chencheng,zfzhang,feng}@is.iscas.ac.cn
[2] Division of Mathematical Sciences,
School of Physical & Mathematical Sciences,
Nanyang Technological University, Singapore
s080001@e.ntu.edu.sg, {hoonwei,lingsan,hxwang}@ntu.edu.sg

Abstract. It has been an appealing but challenging goal in research on *attribute-based encryption* (ABE) and *attribute-based signatures* (ABS) to design a *secure* scheme with *short* ciphertexts and signatures, respectively. While recent results show that some promising progress has been made in this direction, they do not always offer a satisfactory level of security, i.e. achieving *selective* rather than *full* security.

In this paper, we aim to achieve *both* full security and short ciphertexts/signatures for *threshold* access structures in the ABE/ABS setting. Towards achieving this goal, we propose generic property-preserving conversions from inner-product systems to attribute-based systems. We first give concrete constructions of fully secure IPE/IPS with constant-size ciphertexts/signatures in the composite order groups. By making use of our IPE/IPS schemes as building blocks, we then present concrete constructions of fully secure key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) with constant-size ciphertexts, and a fully secure ABS with constant-size signatures with perfect privacy for threshold access structures. These results give rise to the first constructions satisfying the aforementioned requirements. Our schemes reduce the number of pairing evaluations to a constant, a very attractive property for practical attribute-based systems. Furthermore, we show that our schemes can be extended to support large attribute universes and more expressive access structures.

1 Introduction

Attribute-Based Encryption. The notion of attribute-based encryption (ABE) [14] was initially developed from the fuzzy identity-based encryption (FIBE) primitive [31], which allows some sort of error-tolerance. That is, identities are viewed as sets of attributes, and a user can decrypt if she possesses keys for enough of (but not necessarily all) attributes a ciphertext is encrypted under. At the same time, colluding users cannot combine their keys to decrypt a

E. Dawson (Ed.): RSA 2013, LNCS 7779, pp. 50–67, 2013.

ciphertext which none of them were able to decrypt independently. Since then, ABE finds many useful applications in cryptographic access control systems, and is further categorized into ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE). In the former, a secret key is associated with an attribute set; a user can decrypt a ciphertext if and only if the attribute set satisfies the access structure associated with the ciphertext. In the latter, on the contrary, a secret key is associated with an access structure; a user can decrypt a ciphertext associated with an attribute set if and only if the attribute set satisfies the access structure associated with the user's secret key. Recently, the signature analogue of ABE, i.e. attribute-based signatures (ABS), has been introduced [24] (see also [12,23,29]). ABS offers an interesting property in that a signature does not reveal the identity of the signer (hence preserving the privacy of the signer), since it is generated and can be verified based on only the signer's attributes.

While it is desirable that an attribute-based system to be as expressive as possible (in terms of enforcing an access control policy), two major factors to consider when designing an ABE/ABS scheme are *efficiency* and *security*. Majority of existing ABE and ABS schemes have linear-size ciphertexts and signatures, respectively, in the maximal number of attributes. Indeed, recent proposals, such as [1,10,11,15,16], have focused on reducing the sizes of ciphertexts and signatures in the attribute-based setting. Of these, Herranz et al. [16] presented a CP-ABE scheme supporting threshold access policies with constant-size ciphertexts; while Attrapadung and Libert [1] proposed a KP-ABE scheme supporting general access structures with constant-size ciphertexts; and Herranz et al. [15] gave two constructions of ABS with constant-size signatures for threshold predicates. While these works have taken a significant step forward towards improving the efficiency of ABE/ABS, they have so far not achieved a satisfactory level of security. In other words, the aforementioned schemes achieve better efficiency at the expense of weaker security. They are proven to be only *selectively secure*, i.e. an adversary is required to announce the target he intends to attack before seeing the public (system) parameters. The goal of this paper is to offer solutions that achieve both *full security* and *constant-size* ABE ciphertexts or ABS signatures.

Predicate Encryption. Functional encryption (FE) [2,8,22,28] is recently seen as a new vision of public key encryption. In an FE system, a decryption key allows a user to learn a function of the encrypted data. Given a functionality $F(\cdot,\cdot)$, an authority holding a master secret key can generate a decryption key SK_k that is able to compute the function $F(k,x)$ from the encryption of x. The security of the FE system guarantees that one cannot learn anything more about x. ABE and predicate encryption are both example primitives that satisfy the notion of FE.

The concept of predicate encryption (PE) was proposed by Katz, Sahai and Waters [19]. Particularly, they devised a PE scheme for inner products: a ciphertext encrypted for the attribute vector $\boldsymbol{y}$ can only be opened by a key $\boldsymbol{x}$ that gives an inner-product $\boldsymbol{x} \cdot \boldsymbol{y} = 0$. They showed that the inner-product

encryption (IPE) suffices to give functional encryption associated with the evaluation of polynomials or formulae in conjunctive/disjunctive normal form (CNF/DNF). Attrapadung and Libert [2] proposed a fully secure IPE scheme with constant-size ciphertexts based on Waters' tag-based IBE scheme [34]; while Okamoto and Takashima [30] also proposed an IPE scheme with analogous properties on dual pairing vector spaces. We note that it seems possible to construct fully secure ABE with constant-size ciphertexts directly from these two IPE schemes. However, the resulting ABE schemes have two notable shortcomings: (i) the ABE schemes are rather complex[1] and it is not always clear how full security can be proven; and (ii) the access structures are restricted to a single AND/OR-gate.

Our Approach. In this paper, we consider how PE can be used to construct fully secure ABE and ABS with constant-size ciphertexts and signatures, respectively. Moreover, we would like our constructions to support *threshold access structures*. (Henceforth, we use a prefix 't' to indicate that an attribute-based system supports threshold access structures, for example tKP-ABE and tCP-ABE.)

Our general idea is to construct attribute-based systems from inner-product systems by extending the technique from [19]: we treat a vector space as an attribute universe, where each coordinate corresponds to an attribute; for an attribute subset S, a coordinate is equal to 1 if its corresponding attribute is an element of S, otherwise, the coordinate equals to 0. If two subsets have t common attributes, the corresponding vectors overlap in exactly t coordinates, and the inner-product of them equals to t. In addition, we require some coordinates to express threshold values and to allow an inner-product between the vector associated with the attribute subset S and the vector associated with an access structure (if S satisfies the access structure).

One major advantage of such a conversion technique is that the resulting attribute-based construction preserves the sizes of ciphertexts/signatures and the security of the corresponding inner-product scheme. This implies that we can obtain fully (or adaptively) secure tABE with constant-size ciphertexts and fully secure tABS (in terms of unforgeability and perfect privacy) with constant-size signatures, so long as the IPE and the signature analogue (IPS) used in the conversion comply to these properties. We also note that there currently seems to be no suitable IPS candidate for our purpose. For the sake of simplicity, we construct IPE and IPS schemes with the required properties in the composite order group setting as an intermediate step towards achieving fully secure and efficient tABE and tABS. Although it is possible to construct the schemes under the prime order groups (as we will discuss in Section 5), our IPE/IPS schemes are more compact in the composite order groups setting since they do not employ additional tags as with the schemes in [2,30].

[1] Current constructions [2,21,28,29,30] under the prime order groups and proven secure using the dual encryption system proof methodology typically have a multitude of parameters and intricate compositions, in comparisons to those under the composite order groups [20,22].

Moreover, since the secret key components (of IPE/IPS) used in our conversion are independent from each other, it becomes more natural to derive the required security proof using the dual system proof technique as compared to those in [15], for example. We can now make an (ABE/ABS) secret key semi-functional by turning the secret key components sequentially in a hybrid security manner.

Our Contributions. We first give appropriate formal definitions and security models for predicate signatures. We then specify three generic property-preserving conversions: (i) IPE to tKP-ABE, (ii) IPE to tCP-ABE, and (iii) IPS to tABS. Further, we give concrete constructions of IPE and IPS in the composite order group setting. Our IPE scheme is fully secure with constant-size ciphertexts and our IPS scheme is fully unforgeable and perfectly private, and has constant-size signatures. They are proven secure under the complexity assumptions used by Lewko and Waters [22].

Using our IPE scheme as a building block, we present concrete constructions of fully secure tKP-ABE and tCP-ABE with constant-size ciphertexts. The ciphertexts of both the tKP-ABE and the tCP-ABE schemes consist of 3 group elements. The security of our tKP-ABE and tCP-ABE inherits the security of the underlying IPE scheme. We also give a fully secure tABS construction that relies on our IPS scheme with constant-size signatures. Our tABS produces signatures that each also consists of 3 group elements. The full unforgeability and prefect privacy properties are preserved from the underlying IPS scheme. To the best of our knowledge, there are no previous schemes that satisfy these properties. In addition, our schemes reduce the number of pairing evaluations to a constant; this appears to be a very attractive property for attribute-based systems. Table 1 shows that in comparisons with previous work, our attribute-based schemes have better efficiency and higher security. Here, PP denotes public parameters, SK denotes secret keys, CT denotes ciphertexts, Sig denotes signatures, all in the attribute-based setting. Pai denotes the number of paring computations required in the scheme.

We remark that all our schemes in Section 4 are for small universes. Thus as a further contribution, we show that the schemes can be extended to support large universes[2] by borrowing the tricks from [31] in the standard model. Moreover, we show that our constructions can deal with more general access structures, as discussed in Section 5.

2 Predicate Encryption and Signatures

We give the definitions and security models for predicate encryption and predicate signature. We also show how these definitions capture the notions of ABE/ABS and IPE/IPS and provide example instantiations of these primitives.

[2] The attribute universe is a set containing all the attributes defined for an attribute-based system. In the small universe case, the size of the attribute universe is defined at system setup. In the large universe case, the number of attributes is unlimited.

Table 1. Comparisons between existing and our ABE/ABS systems

	scheme	security	size of PP	size of SK	size of CT or Sig	expressiveness	Pai
CP-ABE	EM+09 [11]	selective	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	(n,n)-threshold	2
	CZF11 [10]	selective	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	and-gate	2
	HLR10 [16]	selective	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	threshold	3
	GZC12 [18]	selective	$\mathcal{O}(n)$	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	3
	OT10 [28]	full	$\mathcal{O}(n)^2$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	general	$\mathcal{O}(n)$
	Our CP-ABE	full	$\mathcal{O}(n)$	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	2
KP-ABE	ABP11 [1]	selective	$\mathcal{O}(n)$	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	general	3
	OT10 [28]	full	$\mathcal{O}(n)^2$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	general	$\mathcal{O}(n)$
	Our KP-ABE	full	$\mathcal{O}(n)$	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	2
ABS	HLLR12a [15]	selective	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	threshold	12
	HLLR12b [15]	selective	$\mathcal{O}(n)$	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	3
	OT11 [29]	full	$\mathcal{O}(n)^2$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	general	$\mathcal{O}(n)$
	Our ABS	full	$\mathcal{O}(n)$	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	3

2.1 Predicate Encryption

Predicate encryption (PE) is a variant of functional encryption, which was formally defined in [8]. We now define the syntax of predicate encryption and its security model. (Our definitions follow the general framework of those given in [2,19].[3])

Let $\mathfrak{R} : \mathcal{K} \times \mathcal{X} \rightarrow \{0,1\}$ be a predicate, where $\mathcal{K}$ and $\mathcal{X}$ denote "role" and "policy" spaces. A predicate encryption scheme $\Pi_{\mathsf{PE}} = (\mathsf{PE.Setup}, \mathsf{PE.KeyGen}, \mathsf{PE.Enc}, \mathsf{PE.Dec})$ for $\mathfrak{R}$ consists of four probabilistic polynomial-time (PPT) algorithms that are described as follows:

- $\mathsf{PE.Setup}(\kappa, \mathit{des})$: The algorithm takes a security parameter κ and a scheme description des as input. It outputs some public parameters PP and a master secret key MSK.
- $\mathsf{PE.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y})$: The algorithm takes as input the public parameters PP, a master key MSK and a role $\boldsymbol{y} \in \mathcal{K}$. It returns a secret key $\mathsf{SK}_{\boldsymbol{y}}$ associated with $\boldsymbol{y}$.
- $\mathsf{PE.Enc}(\mathsf{PP}, \boldsymbol{x}, M)$: The algorithm takes as input a message M, an encrypting policy $\boldsymbol{x} \in \mathcal{X}$ and public parameters PP. It outputs a ciphertext CT.
- $\mathsf{PE.Dec}(\mathsf{PP}, \boldsymbol{x}, \mathsf{SK}_{\boldsymbol{y}}, \mathsf{CT})$: The algorithm takes as input a secret key $\mathsf{SK}_{\boldsymbol{y}}$, a ciphertext CT with a policy $\boldsymbol{x}$ and public parameters PP. It outputs a message M or $\perp$.

For correctness, we require that, for all $\boldsymbol{y} \in \mathcal{K}$ and $\boldsymbol{x} \in \mathcal{X}$, if $\mathfrak{R}(\boldsymbol{x}, \boldsymbol{y}) = 1$, then

$$\mathsf{PE.Dec}(\mathsf{PP}, \boldsymbol{x}, \mathsf{PE.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y}), \mathsf{PE.Enc}(\mathsf{PP}, \boldsymbol{x}, M)) = M,$$

[3] Our definition of predicate encryption here and throughout the paper refers to the class of PE with public index (as with [8]), in which the decryption algorithm should input the index component, as well as the bit length, of the plaintext. This type of PE has also been informally referred to as "payload hiding" in the literature [19].

where PP and MSK have been obtained by properly executing the PE.Setup algorithm.

Security Model. In this paper, we consider only the payload-hiding security, which requires that ciphertexts hide the encrypted messages from an adversary but they do not hide their underlying encrypting policies. Let κ be a security parameter. We describe the security model against chosen plaintext attacks (CPA) for a PE scheme Π_{PE} by considering the following security game between an adversary $\mathcal{A}$ and its challenger.

- **Setup**. The challenger runs the $\mathsf{PE.Setup}(\kappa, des)$ algorithm and gives the public parameters PP to the adversary.
- **Phase 1**. The adversary adaptively submits a role $\boldsymbol{y} \in \mathcal{K}$ and the challenger answers with a secret key $\mathsf{SK}_{\boldsymbol{y}}$ to the adversary.
- **Challenge**. The adversary submits two messages M_0 and M_1 of equal length and a challenge policy $\boldsymbol{x} \in \mathcal{X}$. The challenger chooses $\mu \in \{0, 1\}$ at random and encrypts M_μ under $\boldsymbol{x}$. The resulting ciphertext CT is given to the adversary.
- **Phase 2**. The adversary is allowed to continue to make queries as Phase 1.
- **Guess**. Finally, the adversary outputs a guess μ' of μ. We say that $\mathcal{A}$ is successful if none of the role $\boldsymbol{y}$ in Phases 1 & 2 that satisfies $\Re(\boldsymbol{x}, \boldsymbol{y}) = 1$ has been queried and $\mu' = \mu$. The success probability is defined as $Succ^{CPA}_{\mathcal{A},\Pi_{\mathsf{PE}}}(\kappa)$.

Definition 1. *For a PE scheme Π_{PE}, the advantage of an adversary $\mathcal{A}$ in the game is defined as $Adv^{CPA}_{\mathcal{A},\Pi_{\mathsf{PE}}}(\kappa) = |Succ^{CPA}_{\mathcal{A},\Pi_{\mathsf{PE}}}(\kappa) - \frac{1}{2}|$. A PE scheme Π_{PE} is secure if $Adv^{CPA}_{\mathcal{A},\Pi_{\mathsf{PE}}}(\kappa)$ is negligible with respect to the security parameter κ, for any PPT adversary $\mathcal{A}$.*

Note that a weaker model that considers selective security can be defined as with the above security game with the exception that the adversary $\mathcal{A}$ is allowed to choose the challenge encrypting policy $\boldsymbol{x}$ before the setup phase.

Variants. There exist many public key primitives that can be viewed as special cases of PE, for example, identity-based encryption (IBE) [4,9], hierarchical IBE (HIBE) [13], broadcast encryption [6], ABE [31,14], IPE [2,30], and spatial encryption (SE) [5]. We provide the definitions of ABE and IPE using the syntax of PE in the full version of this paper.

2.2 Predicate Signatures

We now define predicate signatures using the syntax of PE. In predicate signatures, the signing and verification algorithms are parameterized by a role $\boldsymbol{y}$ and a policy predicate $\boldsymbol{x}$, respectively. A predicate signature generated by a signer with role $\boldsymbol{y}$ is said to be correctly verified by the public parameters and a policy predicate $\boldsymbol{x}$ if $\Re(\boldsymbol{x}, \boldsymbol{y}) = 1$ holds. No other information is revealed by the signature. A predicate signature (PS) scheme $\Pi_{\mathsf{PS}} = (\mathsf{PS.Setup}, \mathsf{PS.KeyGen}, \mathsf{PS.Sign}, \mathsf{PS.Verify})$ for $\Re$ then consists of four probabilistic PPT algorithms that are described as follows:

- $\mathsf{PS.Setup}(\kappa, des)$: The algorithm takes a security parameter κ and a scheme description des as input. It outputs some public parameters PP and a master secret key MSK.
- $\mathsf{PS.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y})$: The algorithm takes as input the public parameters PP, a master key MSK and a role $\boldsymbol{y} \in \mathcal{K}$. It returns a secret key $\mathsf{SK}_{\boldsymbol{y}}$ associated with $\boldsymbol{y}$.
- $\mathsf{PS.Sign}(\mathsf{PP}, \mathsf{SK}_{\boldsymbol{y}}, \boldsymbol{x}, M)$: The algorithm takes as input a message M, a secret key $\mathsf{SK}_{\boldsymbol{y}}$, a signing policy $\boldsymbol{x} \in \mathcal{X}$ and public parameters PP. It outputs a signature σ.
- $\mathsf{PS.Verify}(\mathsf{PP}, \boldsymbol{x}, \sigma, M)$: The algorithm takes as input a message M, a signature σ with a policy $\boldsymbol{x}$ and public parameters PP. It outputs 1 if the signature is deemed valid and 0 otherwise.

For correctness, for all $\boldsymbol{y} \in \mathcal{K}$ and $\boldsymbol{x} \in \mathcal{X}$, if $\mathfrak{R}(\boldsymbol{x}, \boldsymbol{y}) = 1$, it is required that

$$\mathsf{PS.Verify}(\mathsf{PP}, \boldsymbol{x}, \mathsf{PS.Sign}(\mathsf{PP}, \mathsf{PS.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y}), \boldsymbol{x}, M), M) = 1$$

and the values $\mathsf{PP}, \mathsf{MSK}$ have been obtained by properly executing the algorithms $\mathsf{PS.Setup}$.

Security Model. We consider two essential security properties for a PS scheme: unforgeability and signer privacy.

Unforgeability: A PS scheme must provide the typical unforgeability property, even against colluding users. Let κ be a security parameter. We then define unforgeability under chosen message attacks (UF-CMA) for a PS scheme Π_{PS} by considering the following security game between an adversary $\mathcal{A}$ and its challenger:

- **Setup**. The challenger runs $\mathsf{PS.Setup}(\kappa, des)$, and sends the public parameters PP to $\mathcal{A}$.
- **Query**. $\mathcal{A}$ can make secret key and signature queries.
  - **Secret key queries**. $\mathcal{A}$ adaptively chooses a role $\boldsymbol{y} \in \mathcal{K}$ and receives the secret key $\mathsf{SK}_{\boldsymbol{y}} = \mathsf{PS.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y})$ from the challenger.
  - **Signature queries**. $\mathcal{A}$ adaptively chooses a pair $(\boldsymbol{x}, M)$ consisting of a policy $\boldsymbol{x}$ and a message M. The challenger chooses a role $\boldsymbol{y}$ that $\mathfrak{R}(\boldsymbol{x}, \boldsymbol{y}) = 1$, runs $\mathsf{SK}_{\boldsymbol{y}} = \mathsf{PS.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y})$ and computes a signature $\sigma = \mathsf{PS.Sign}(\mathsf{PP}, \mathsf{SK}_{\boldsymbol{y}}, \boldsymbol{x}, M)$ which is returned to $\mathcal{A}$.
- **Forgery**. At the end of the game, $\mathcal{A}$ outputs a tuple $(\boldsymbol{x}^*, M^*, \sigma^*)$. $\mathcal{A}$ is successful if:
  - $\mathcal{A}$ has not made any signature query for the pair $(\boldsymbol{x}^*, M^*)$;
  - None of the role $\boldsymbol{y}$ in secret key queries phase satisfies $\mathfrak{R}(\boldsymbol{x}^*, \boldsymbol{y}) = 1$;
  - $\mathsf{PS.Verify}(\mathsf{PP}, \boldsymbol{x}^*, \sigma^*, M^*) = 1$.

The advantage of the adversary $\mathcal{A}$ in successfully breaking the UF-CMA security of a PS scheme Π_{PS} is defined as $Succ^{UF\text{-}CMA}_{\mathcal{A},\Pi_{\mathsf{PS}}}(\kappa) = Pr[\mathcal{A}\ wins]$.

Definition 2. *A PS scheme Π_{PS} is UF-CMA if $Succ^{UF\text{-}CMA}_{\mathcal{A},\Pi_{\mathsf{PS}}}(\kappa)$ is negligible with respect to the security parameter κ, for any PPT adversary $\mathcal{A}$.*

Similarly, if the adversary $\mathcal{A}$ is allowed to choose the challenge signing policy $\boldsymbol{x}$ before the setup phase, we then have a weaker model called selective unforgeability.

Perfect Privacy: This property is required to achieve anonymous ABS in the sense that PS signatures reveal no information except that the role information that has been used to generate the signatures. Perfect privacy must hold even against an unbounded adversary which has knowledge of the master secret key.

Definition 3. *A PS scheme Π_{PS} is perfectly private, if for any message M, any two roles $\boldsymbol{y}_1, \boldsymbol{y}_2$, any secret keys $\mathsf{SK}_1 = \mathsf{PS.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y}_1), \mathsf{SK}_2 = \mathsf{PS.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \boldsymbol{y}_2)$, and any policy $\boldsymbol{x}$ such that $\mathfrak{R}(\boldsymbol{x}, \boldsymbol{y}_1) = 1$ and $\mathfrak{R}(\boldsymbol{x}, \boldsymbol{y}_2) = 1$, the distribution of $\mathsf{PS.Sign}(\mathsf{PP}, \mathsf{SK}_1, \boldsymbol{x}, M)$ is identical to that of $\mathsf{PS.Sign}(\mathsf{PP}, \mathsf{SK}_2, \boldsymbol{x}, M)$.*

Identity-based signatures (IBS) [17], identity-based ring signatures (IBRS) [37], and ABS [12,15,23,24,29] are example of special cases of PS. Moreover, as we define below, the notion of inner-product signatures (IPS) is also a variant of PS.

Inner-Product Signatures. The notion of inner-product signatures (IPS) can be defined as with PS, except with the following modification:

- The setup algorithm defines a positive integer N and a dimension n;
- The role space $\mathcal{K} := \{\boldsymbol{v} := (v_1, \ldots, v_n) \in \mathbb{Z}_N^n\}$;
- The policy space $\mathcal{X} := \{\boldsymbol{x} := (x_1, \ldots, x_n) \in \mathbb{Z}_N^n\}$;
- The predicate $\mathfrak{R} : \mathcal{K} \times \mathcal{X} \to \{0, 1\}$ is defined as

$$\mathfrak{R}(\boldsymbol{v}, \boldsymbol{x}) := \begin{cases} 1 & \text{if } \langle \boldsymbol{v}, \boldsymbol{x} \rangle = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The detailed description of ABS can refer to [24,29]. In this paper, we are mainly concerned with the notions of ABS and IPS.

3 Generic Constructions

We describe transformation from inner-product systems to attribute-based systems supporting threshold access structures. We first recall the definition of an access structure.

Definition 4. *Let $\mathbb{U} = \{att_1, att_2, \ldots, att_n\}$ be a set of attributes. An access structure is a set collection $\mathbb{A} \subseteq 2^{\{att_1, att_2, \ldots, att_n\}} \backslash \emptyset$. An access structure is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

We use (Ω, t) to denote a threshold access structure Γ in $\mathbb{U}$ if there exist a threshold t and a subset $\Omega \subseteq \mathbb{U}$ such that $S \in \Gamma \Leftrightarrow |S \cap \Omega| \geq t$. When the access structures are restricted to the threshold setting, we call it threshold CP-ABE (tCP-ABE), threshold KP-ABE (tKP-ABE) and threshold ABS (tABS).

3.1 Generic Construction of tKP-ABE from IPE

To construct a tKP-ABE scheme over an attribute universe $\mathbb{U} := \{att_1, \ldots, att_n\}$, we require an $(n+1)$-dimensional IPE scheme. Given an IPE scheme Π_{IPE} with four algorithms: $(\mathsf{IPE.Setup}, \mathsf{IPE.KeyGen}, \mathsf{IPE.Enc}, \mathsf{IPE.Dec})$, we construct a tKP-ABE scheme Π_{tKP} with the corresponding four algorithms: $(\mathsf{tKP.Setup}, \mathsf{tKP.KeyGen}, \mathsf{tKP.Enc}, \mathsf{tKP.Dec})$ as follows:

- $\mathsf{tKP.Setup}(\kappa, \mathbb{U})$: It runs $\mathsf{IPE.Setup}(\kappa, n+1)$ and outputs public parameters PP and a master key MSK.
- $\mathsf{tKP.Enc}(\mathsf{PP}, S, M)$: For a subset $S \subseteq \mathbb{U}$, it first computes a vector $\boldsymbol{x} := (x_1, \ldots, x_{n+1})$ as follows:

$$x_1 := -1, \quad \text{for } 1 \leq i \leq n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}$$

  Then runs $\mathsf{IPE.Enc}(\mathsf{PP}, \boldsymbol{x}, M)$ and outputs a ciphertext CT.
- $\mathsf{tKP.KeyGen}(\mathsf{PP}, \Gamma := (\Omega, t), \mathsf{MSK})$: For a threshold access structure (Ω, t) (where we denote $m := |\Omega|$), it computes a vector $\boldsymbol{v} := (v_1, \ldots, v_n)$ as follows:

$$\text{for } 1 \leq i \leq n : v_i := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

  Then for $1 \leq j \leq m - t + 1$ it runs $\mathsf{IPE.SK}_i := \mathsf{IPE.KeyGen}(\mathsf{PP}, \boldsymbol{v}_j, \mathsf{MSK})$, where $\boldsymbol{v}_j := (t + j - 1, v_1, \ldots, v_n)$. Outputs the secret key $\mathsf{KP.SK}_{(\Omega,t)} := \{\mathsf{IPE.SK}_j\}_{1 \leq j \leq m-t+1}$.
- $\mathsf{tKP.Dec}(\mathsf{PP}, \mathsf{CT}, S, \mathsf{KP.SK}_{(\Omega,t)})$: For a ciphertext CT with the subset S and a secret key parsed as $\mathsf{KP.SK}_{(\Omega,t)} := \{\mathsf{IPE.SK}_1, \ldots, \mathsf{IPE.SK}_{m-t+1}\}$, if $k := |S \cap \Omega| \geq t$, it runs $\mathsf{IPE.Dec}(\mathsf{PP}, \boldsymbol{x}, \mathsf{CT}, \mathsf{IPE.SK}_{k-t+1})$ and outputs the message M, where $\boldsymbol{x} := (x_1, \ldots, x_{n+1})$:

$$x_1 := -1, \quad \text{for } 1 \leq i \leq n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}$$

Correctness. For the vector $\boldsymbol{x} := (x_1, \ldots, x_{n+1})$ corresponding to the subset S in the ciphertext and the vector $\boldsymbol{v}_{k-t+1} := (v_1, \ldots, v_{n+1})$ corresponding to the secret key component $\mathsf{IPE.SK}_{k-t+1}$ in the tKP-ABE, we have

$$x_1 \cdot v_1 = -k, \quad \sum_{i=2}^{n+1} x_i \cdot v_i = k$$

So we have $\langle \boldsymbol{x}, \boldsymbol{v}_{k-t+1} \rangle = 0$. This implies that the resulting tKP-ABE scheme inherits the decryptability from the underlying IPE scheme, i.e.,

$$\mathsf{IPE.Dec}(\mathsf{PP}, \boldsymbol{x}, \mathsf{IPE.Enc}(\mathsf{PP}, \boldsymbol{x}, M), \mathsf{IPE.KeyGen}(\mathsf{PP}, \boldsymbol{v}, \mathsf{MSK})) = M$$

iff $\langle \boldsymbol{x}, \boldsymbol{v} \rangle = 0$.

Theorem 1. *The resulting tKP-ABE scheme is (selectively) secure if the underlying IPE is (selectively) secure.*

3.2 Generic Construction of tCP-ABE from IPE

To construct a tCP-ABE scheme over an attribute universe $\mathbb{U} := \{att_1, \ldots, att_n\}$, we require an $(n + 2)$-dimensional IPE scheme. Given an IPE scheme Π_{IPE} with four algorithms: $(\mathsf{IPE.Setup}, \mathsf{IPE.KeyGen}, \mathsf{IPE.Enc}, \mathsf{IPE.Dec})$, we construct a tCP-ABE scheme Π_{tCP} with the corresponding four algorithms: $(\mathsf{tCP.Setup}, \mathsf{tCP.KeyGen}, \mathsf{tCP.Enc}, \mathsf{tCP.Dec})$ as follows:

- $\mathsf{tCP.Setup}(\kappa, \mathbb{U})$: It runs $\mathsf{IPE.Setup}(\kappa, n + 2)$ and outputs public parameters PP and a master key MSK.
- $\mathsf{tCP.Enc}(\mathsf{PP}, \Gamma := (\Omega, t), M)$: For a threshold access structure (Ω, t), it computes a vector $\boldsymbol{x} := (x_1, \ldots, x_{n+2})$ as follows:

$$x_1 := -t, \quad \text{for } 1 \le i \le n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}, \quad x_{n+2} := 1$$

  Then runs $\mathsf{IPE.Enc}(\mathsf{PP}, \boldsymbol{x}, M)$ and outputs a ciphertext CT.
- $\mathsf{tCP.KeyGen}(\mathsf{PP}, S, \mathsf{MSK})$: For a subset $S \subseteq \mathbb{U}$, it first computes a vector $\boldsymbol{v} := (v_1, \ldots, v_n)$ as follows:

$$\text{for } 1 \le i \le n : v_i := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}$$

  Then for $1 \le i \le |S| - 1$ it runs $\mathsf{IPE.SK}_i := \mathsf{IPE.KeyGen}(\mathsf{PP}, \boldsymbol{v}_i, \mathsf{MSK})$, where $\boldsymbol{v}_i := (1, v_1, \ldots, v_n, 1 - i)$. Outputs the secret key $\mathsf{CP.SK}_S := \{\mathsf{IPE.SK}_i\}_{1 \le i \le |S|-1}$.
- $\mathsf{tCP.Dec}(\mathsf{PP}, \mathsf{CT}, \Gamma := (\Omega, t), \mathsf{CP.SK}_S)$: For a ciphertext CT with the threshold (Ω, t) and a secret key parsed as $\mathsf{KP.SK}_S := \{\mathsf{IPE.SK}_i\}_{1 \le i \le |S|-1}$, if $k := |S \cap \Omega| \ge t$, it runs $\mathsf{IPE.Dec}(\mathsf{PP}, \boldsymbol{x}, \mathsf{CT}, \mathsf{IPE.SK}_{k-t+1})$ and outputs the message M, where $\boldsymbol{x} := (x_1, \ldots, x_{n+2})$:

$$x_1 := -t, \quad \text{for } 1 \le i \le n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}, \quad x_{n+2} := 1$$

Correctness. The security reduction can follow closely with that of the generic construction for the above tKP-ABE and we will not discuss any further here.

Theorem 2. *The resulting tCP-ABE scheme is (selectively) secure if the underlying IPE is (selectively) secure.*

3.3 Generic Construction of tABS from IPS

To construct a tABS scheme over an attribute universe $\mathbb{U} := \{att_1, \ldots, att_n\}$, we require an $(n+2)$-dimensional IPS scheme. Given an IPS scheme Π_{IPS} with four algorithms: $(\mathsf{IPS.Setup}, \mathsf{IPS.KeyGen}, \mathsf{IPS.Sign}, \mathsf{IPS.Verify})$, we construct a tABS scheme Π_{tABS} with the corresponding four algorithms: $(\mathsf{tABS.Setup}, \mathsf{tABS.KeyGen}, \mathsf{tABS.Sign}, \mathsf{tABS.Verify})$ as follows:

- $\mathsf{tABS.Setup}(\kappa, \mathbb{U})$: It runs $\mathsf{IPS.Setup}(\kappa, n+2)$ and outputs public parameters PP and a master key MSK.
- $\mathsf{tABS.KeyGen}(\mathsf{PP}, S, \mathsf{MSK})$: For a subset $S \subseteq \mathbb{U}$, it first computes a vector $\boldsymbol{v} := (v_1, \ldots, v_n)$ as follows:

$$\text{for } 1 \leq i \leq n : v_i := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}$$

  Then for $1 \leq i \leq |S| - 1$ it runs $\mathsf{IPS.SK}_i := \mathsf{IPS.KeyGen}(\mathsf{PP}, \boldsymbol{v}_i, \mathsf{MSK})$, where $\boldsymbol{v}_i := (1, v_1, \ldots, v_n, 1 - i)$. Outputs the secret key $\mathsf{ABS.SK}_S := \{\mathsf{IPS.SK}_i\}_{1 \leq i \leq |S|-1}$.
- $\mathsf{tABS.Sign}(\mathsf{PP}, \mathsf{ABS.SK}_S, \Gamma := (\Omega, t), M)$: If $k := |S \cap \Omega| \geq t$, it runs $\sigma \leftarrow \mathsf{IPS.Sign}(\mathsf{PP}, \boldsymbol{x}, \mathsf{IPS.SK}_{k-t+1}, \Gamma || M)$, where $\boldsymbol{x} := (x_1, \ldots, x_{n+2})$ as follows:

$$x_1 := -t, \quad \text{for } 1 \leq i \leq n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}, \quad x_{n+2} := 1$$

  And it outputs σ as the signature.
- $\mathsf{tABS.Verify}(\mathsf{PP}, \sigma, \Gamma := (\Omega, t), M)$: It runs $\mathsf{IPS.Verify}(\mathsf{PP}, \boldsymbol{x}, \sigma, \Gamma || M)$ where $\boldsymbol{x} := (x_1, \ldots, x_{n+2})$ as follows:

$$x_1 := -t, \quad \text{for } 1 \leq i \leq n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}, \quad x_{n+2} := 1$$

  And outputs the result.

Correctness. The deduction can follow closely with that of the generic construction of tKP-ABE above. And we omit it here.

Theorem 3. *The resulting tABS scheme is (selectively) unforgeable and perfectly private if the underlying IPS is (selectively) unforgeable and perfectly private.*

The security proofs of Theorems 1, 2 & 3 can be easily obtained from the definitions of ABE/ABS and IPE/IPS. Due to space constraints, we omit them here.

4 Concrete Constructions of tABE and tABS

We are now ready to describe how to construct a threshold attribute-based system from an inner-product system. For the space considersion, we give concrete constructions of IPE/IPS which are tailored to our needs in the full version of this paper. Making use of our IPE/IPS schemes as building blocks, we propose constructions for fully secure tKP-ABE and tCP-ABE with constant-size ciphertexts, as well as fully secure and perfectly private tABS with constant-size signatures. Due to the space limitation, we only give the instances of tKP-ABE and ABS. The tCP-ABE scheme can be easily obtained from the tKP-ABE scheme and we omit it here. The correctness and security of our schemes follows from the generic conversions. Our schemes are for small universes of attributes $\mathbb{U} := \{att_1, \ldots, att_n\}$ and based on composite order groups. (The definition of composite order bilinear groups can be found in the full version of this paper or [22].)

Composite Order Bilinear Groups. We define composite order bilinear groups as follows: let $\mathcal{G}_c$ be a group generator which outputs $\mathcal{I} := (N = p_1p_2p_3, G, G_T, e)$ where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1p_2p_3$, and e is a bilinear map, $e : G \times G \to G_T$ such that $e(g, g) \neq 1$ for g and for any $u, v \in \mathbb{Z}_N$, it holds that $e(g^u, g^v) = e(g, g)^{uv}$. We say that G is a bilinear group if the group operation in G and the bilinear map $e : G \times G \to G_T$ are both efficiently computable. Notice that the map e is symmetric since $e(g^u, g^v) = e(g, g)^{uv} = e(g^v, g^u)$. We let $G_{p_1}, G_{p_2}, G_{p_3}$ denote the subgroups of order p_1, p_2, p_3 in G, respectively. Furthermore, for $a, b, c \in \{1, p_1, p_2, p_3\}$ we denote by G_{abc} the subgroup of order abc. From the fact that the group is cyclic it is simple to verify that if g and h are group elements of different order (and thus belonging to different subgroups), then $e(g, h) = 1$. This is called the orthogonality property and is a crucial tool in our constructions.

4.1 Fully Secure tKP-ABE with Constant-Size Ciphertexts

- $\mathsf{tKP.Setup}(\kappa, \mathbb{U} := \{att_1, \ldots, att_n\})$: The setup algorithm chooses a random description $\mathcal{I} := (N = p_1p_2p_3, G, G_T, e)$ with $G = G_{p_1} \times G_{p_2} \times G_{p_3}$. It then randomly picks $\alpha, a_0, \ldots, a_{n+1} \in \mathbb{Z}_N$ and $X_3 \in G_{p_3}$. It then sets $\boldsymbol{h} := (h_0, \ldots, h_{n+1}) = (g^{a_0}, g^{a_1}, \ldots, g^{a_{n+1}})$. It outputs the public parameters and master key as $\mathsf{PP} := (\mathcal{I}, g, \boldsymbol{h}, e(g, g)^{\alpha})$, $\mathsf{MSK} := (\alpha, X_3)$, respectively.
- $\mathsf{tKP.KeyGen}(\mathsf{PP}, \Gamma := (\Omega, t), \mathsf{MSK})$: For a threshold access structure (Ω, t) (where we let $m := |\Omega|$), the algorithm computes a vector $\boldsymbol{v} := (v_1, \ldots, v_n)$ as follows:
$$\text{for } 1 \leq i \leq n : v_i := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}.$$
Then for $1 \leq i \leq m - t + 1$, the algorithm randomly picks $r_i \in \mathbb{Z}_N$ and $(R_{0,i}, \ldots, R_{n+1,i}) \in G_{p_3}^{n+2}$, and outputs the secret key element $\mathsf{SK}_i := (K_{0,i}, K_{1,i}, \ldots, K_{n+1,i})$ by setting

$$K_{0,i} := g^{r_i} R_{0,i}, \quad K_{1,i} := g^{\alpha} h_0^{r_i} R_{1,i}, \quad \left\{ K_{j,i} := \left(h_1^{-\frac{v_{j,i}}{v_{1,i}}} h_j \right)^{r_i} R_{j,i} \right\}_{j=2,\ldots,n+1},$$

where $\boldsymbol{v}_i := (v_{1,i}, \ldots, v_{n+1,i}) = (t+i-1, v_1, \ldots, v_n)$. It also outputs the secret key $\mathsf{KP.SK}_{(\Omega,t)} := \{\mathsf{SK}_i\}_{1 \le i \le m-t+1}$.

– $\mathsf{tKP.Enc}(\mathsf{PP}, S, M)$: For a subset $S \subseteq \mathbb{U}$ and a message $M \in G_T$ to encrypt, the algorithm first computes a vector $\boldsymbol{x} := (x_1, \ldots, x_{n+1})$ as follows:

$$x_1 := -1, \quad \text{for } 1 \le i \le n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}.$$

It then randomly picks $s \in \mathbb{Z}_N$ and computes the ciphertext $\mathsf{CT} := (C, C_0, C_1)$ as $C := M \cdot e(g,g)^{\alpha s}, \quad C_0 := g^s, \quad C_1 := \left(h_0 \prod_{j=1}^{n+1} h_j^{x_j} \right)^s$.

– $\mathsf{tKP.Dec}(\mathsf{PP}, \mathsf{CT}, S, \mathsf{KP.SK}_S)$: For a ciphertext CT parsed as (C, C_0, C_1) with the subset S and a secret key $\mathsf{KP.SK}_{(\Omega,t)}$ parsed as $\{\mathsf{SK}_1, \ldots, \mathsf{SK}_{m-t+1}\}$, if $k := |S \cap \Omega| \ge t$, the algorithm first computes $\boldsymbol{x} := (x_1, \ldots, x_{n+1})$:

$$x_1 := -1, \quad \text{for } 1 \le i \le n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}.$$

It then uses $\mathsf{SK}_{k-t+1} := (K_0, K_1, \ldots, K_{n+1})$ to decrypt: $e(g,g)^{\alpha s} = e(C_0, K_1 \prod_{j=2}^{n+1} K_j^{x_j}) / e(C_1, K_0)$ and recovers the message as $M := C / e(g,g)^{\alpha s}$.

4.2 Fully Secure tABS with Constant-Size Signatures

– $\mathsf{tABS.Setup}(\kappa, \mathbb{U} := \{att_1, \ldots, att_n\})$: The setup algorithm chooses a random description $\mathcal{I} := (N = p_1 p_2 p_3, G, G_T, e)$ with $G = G_{p_1} \times G_{p_2} \times G_{p_3}$. It then randomly picks $\alpha, a_0, \ldots, a_{n+2}, b_0, b_1, b_2 \in \mathbb{Z}_N$, $X_3 \in G_{p_3}$ and a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_N$. The algorithm sets $\boldsymbol{h} := (h_0, \ldots, h_{n+2}) = (g^{a_0}, g^{a_1}, \ldots, g^{a_{n+2}})$, and outputs the public parameters and master key as $\mathsf{PP} := (\mathcal{I}, g, \boldsymbol{h}, g^{b_0}, g^{b_1}, g^{b_2}, X_3, e(g,g)^{\alpha}), \quad \mathsf{MSK} := (\alpha)$.

– $\mathsf{tABS.KeyGen}(\mathsf{PP}, S, \mathsf{MSK})$: For a subset $S \subseteq \mathbb{U}$, the algorithm first computes a vector $\boldsymbol{v} := (v_1, \ldots, v_n)$ as follows:

$$\text{for } 1 \le i \le n : v_i := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}.$$

Then for $1 \le i \le |S| - 1$, the algorithm randomly picks $r_i \in \mathbb{Z}_N$ and $(R_{0,i}, \ldots, R_{n+2,i}) \in G_{p_3}^{n+3}$, and outputs the secret key element $\mathsf{SK}_i := (K_{0,i}, K_{1,i},$
$\ldots, K_{n+2,i})$ by setting

$$K_{0,i} := g^{r_i} R_{0,i}, \quad K_{1,i} := g^{\alpha} h_0^{r_i} R_{1,i}, \quad \left\{ K_{j,i} := \left(h_1^{-\frac{v_{j,i}}{v_{1,i}}} h_j \right)^{r_i} R_{j,i} \right\}_{j=2,\ldots,n+2},$$

where $\boldsymbol{v}_i := (v_{1,i}, \ldots, v_{n+2,i}) = (1, v_1, \ldots, v_n, 1-i)$. The algorithm also outputs the secret key $\mathsf{ABS.SK}_S := \{\mathsf{SK}_i\}_{1 \le i \le |S|-1}$.

– $\mathsf{tABS.Sign}(\mathsf{PP}, \mathsf{ABS.SK}_S, \Gamma := (\Omega, t), M)$: To sign a message M with a threshold access structure (Ω, t) with a secret key $\mathsf{ABS.SK}_S$ parsed as $\{\mathsf{SK}_i\}_{1 \le i \le |S|-1}$, if $k := |S \cap \Omega| \ge t$, the algorithm uses $\mathsf{SK}_{k-t+1} := (K_0, K_1, \ldots, K_{n+2})$. It first computes $\boldsymbol{v} := (v_1, \ldots, v_{n+2})$ and $\boldsymbol{x} := (x_1, \ldots, x_{n+2})$ as follows:

$$v_1 := 1, \quad \text{for } 1 \le i \le n : v_{i+1} := \begin{cases} 1 & \text{if } att_i \in S \\ 0 & \text{otherwise} \end{cases}, \quad v_{n+2} := k - t;$$

$$x_1 := -t, \quad \text{for } 1 \le i \le n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}, \quad x_{n+2} := 1.$$

The algorithm then randomly picks $r_1, r_2, \alpha' \in \mathbb{Z}_N$ and $R_0, \ldots, R_{n+2}$, $R'_1, R'_2, R'_3 \in G_{p_3}$, and computes

$$K'_0 := K_0 \cdot g^{r_1} R_0, \quad K'_1 := K_1 \cdot g^{\alpha'} h_0^{r_1} R_1, \quad \left\{ K'_i := K_i \cdot (h_1^{-\frac{v_i}{v_1}} h_i)^{r_1} R_i \right\}_{i=2,\ldots,n+2},$$
$$K'_{n+1} := g^{r_2} R'_1, \quad K'_{n+2} := g^{-\alpha'} g^{r_2 b_0} R'_2, \quad K'_{n+3} := ((g^{b_1})^{H(M||\Gamma,\boldsymbol{x})} g^{b_2})^{r_2} R'_3.$$

It outputs the signature $\sigma := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ by setting

$$\sigma_1 := K'_1 \cdot \prod_{i=2}^{n} (K'_i)^{x_i}, \quad \sigma_2 := K'_0, \quad \sigma_3 := K'_{n+2} \cdot K'_{n+3}, \quad \sigma_4 := K'_{n+1}.$$

– $\mathsf{tABS.Verify}(\mathsf{PP}, \sigma, \Gamma := (\Omega, t), M)$: On input a signature σ parsed as $(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ and a threshold (Ω, t), the algorithm computes $\boldsymbol{x} := (x_1, \ldots, x_{n+2})$, where

$$x_1 := -t, \quad \text{for } 1 \le i \le n : x_{i+1} := \begin{cases} 1 & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}, \quad x_{n+2} := 1.$$

The algorithm outputs 1 if

$$e(g,g)^{\alpha} = \frac{e(g, \sigma_1 \cdot \sigma_3)}{e\big(h_0 \prod_{i=1}^{n+2} h_i^{x_i}, \sigma_2\big) \cdot e\big(g^{b_0} (g^{b_1})^{H(M||\Gamma,\boldsymbol{x})} g^{b_2}, \sigma_4\big)}.$$

Otherwise, it outputs 0.

5 Extensions

5.1 Constructions in Prime Order Groups

Using groups of prime order can potentially lead to more efficient systems (via faster group operations) and security under different assumptions. A natural problem is how to construct the prime order group variants of our systems. This depends on the constructions of the underlying IPE/IPS. For IPE, [2,30]

gave two fully secure IPE schemes with constant-size ciphertexts in prime order groups. For IPS, we present a fully secure and perfectly private IPS scheme with constant-size signatures in the prime order groups based on the HIPE (Hierarchical Inner-product Encryption) scheme of [30]. The detail of the construction is given in the full version of this paper. With that, we can make use of the above constructions to obtain the desired attribute-based schemes in the prime order groups.

5.2 Large Universe Constructions

Our constructions in Section 4 are limited to the small-universe case where the set of attributes $\mathbb{U}$ is defined at system setup and the size of the public parameters grows with $|\mathbb{U}|$. We now show how to extend them to the large universe setting where the number of attributes is unlimited and the public parameter size is constant. In the random oracle model, it is easy to overcome the dimension-limitation and achieve a "large-dimension" in the inner-product systems.

We now turn to realizing a large universe constructions in the standard model. From the concrete constructions presented in Section 4, we can apply the tricks used for the large universe constructions in [31,35]. As in the random oracle model, we "program" each coordinate parameter element by using a hash function that has enough degrees of randomness to plug in the same information. The tradeoff is that we need to define the maximum number of attributes max that any one key may have in the setup phase. Moreover, the public parameters grow linearly with max. We stress that this does not limit the number of attributes that may be used in the system. We realize construction in the standard model by adapting the construction of tKP-ABE in the full version of this paper. We remark that similar techniques can be used to realize large universe variant of our ABE/ABS constructions based on composite groups and ABE constructions based on the prime order groups converted from [2] in the standard model, although we do not provide the details in this paper.

5.3 More General Access Structures

Our generic conversions can be extended to admit weighted threshold access structures which are more general than threshold. We use $\Gamma := (\Omega, \omega, t)$ to denote a weighted threshold access structure [3] over $\mathbb{U}$ if there exist a threshold t and an assignment of weights $\omega : \mathbb{U} \to \mathbb{Z}_N$ such that $S \in \Gamma \Leftrightarrow \Sigma_{att \in S} \omega(att) \geq t$. We can make our generic conversion support weighted threshold access structure by a slight modification. For a weighted threshold access structure $\Gamma := (\Omega, \omega, t)$, we set the vector $\boldsymbol{x} := (x_1, \ldots, x_n)$ as

$$\text{for } 1 \leq i \leq n : x_i := \begin{cases} \omega(att_i) & \text{if } att_i \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

and the vector $\boldsymbol{v} := (v_1, \ldots, v_n)$ expressing subset $S \subseteq \mathbb{U}$ is unchange.

We can compute the sum of weights of the attributes in S by computing the inner-product of the two vectors. This way, we can realize the weighted threshold access structures.

Acknowledgment. The work is supported by the National Basic Research Program of China (No. 2013CB338003), and the National Natural Science Foundation of China (No.61170278, 91118006).

References

1. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011)
2. Attrapadung, N., Libert, B.: Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 384–402. Springer, Heidelberg (2010)
3. Beimel, A., Tassa, T., Weinreb, E.: Characterizing Ideal Weighted Threshold Secret Sharing. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 600–619. Springer, Heidelberg (2005)
4. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
5. Boneh, D., Hamburg, M.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
6. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
7. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
8. Boneh, D., Sahai, A., Waters, B.: Functional Encryption: Definitions and Challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011)
9. Canetti, R., Halevi, S., Katz, J.: A Forward-Secure Public-Key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003, vol. 2656, pp. 254–271. Springer, Heidelberg (2003)
10. Chen, C., Zhang, Z., Feng, D.: Efficient Ciphertext Policy Attribute-Based Encryption with Constant-Size Ciphertext and Constant Computation-Cost. In: Boyen, X., Chen, X. (eds.) ProvSec 2011. LNCS, vol. 6980, pp. 84–101. Springer, Heidelberg (2011)
11. Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M.: A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. LNCS, vol. 5451, pp. 13–23. Springer, Heidelberg (2009)

12. Escala, A., Herranz, J., Morillo, P.: Revocable Attribute-Based Signatures with Adaptive Security in the Standard Model. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 224–241. Springer, Heidelberg (2011)
13. Gentry, C., Silverberg, A.: Hierarchical ID-Based Cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
14. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: CCS 2006, pp. 89–98. ACM Press (2006)
15. Herranz, J., Laguillaumie, F., Libert, B., Ràfols, C.: Short Attribute-Based Signatures for Threshold Predicates. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 51–67. Springer, Heidelberg (2012)
16. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant Size Ciphertexts in Threshold Attribute-Based Encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 19–34. Springer, Heidelberg (2010)
17. Hess, F.: Efficient Identity Based Signature Schemes Based on Pairings. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003)
18. Ge, A., Zhang, R., Chen, C., Ma, C., Zhang, Z.: Threshold Ciphertext Policy Attribute-Based Encryption with Constant Size Ciphertexts. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 336–349. Springer, Heidelberg (2012)
19. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
20. Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
21. Lewko, A.: Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (2012)
22. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
23. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: Asiaccs 2010, pp. 60–69. ACM Press, New York (2010)
24. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-Based Signatures. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 376–392. Springer, Heidelberg (2011)
25. Mordini, E., Massari, S.: Body, biometrics and identity. Bioethics 22(9), 488–498 (2008)
26. Nagar, A., Rane, S., Vetro, A.: Alignment and bit extraction for secure fingerprint biometrics. In: SPIE Conference on Electronic Imaging (2010)
27. Nandakumar, K., Jain, A.: Multibiometric Template Security Using Fuzzy Vault. In: International Conference on Biometrics: Theory, Applications and Systems, pp. 1–6 (2008)
28. Okamoto, T., Takashima, K.: Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)

29. Okamoto, T., Takashima, K.: Efficient Attribute-Based Signatures for Non-monotone Predicates in the Standard Model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 35–52. Springer, Heidelberg (2011)
30. Okamoto, T., Takashima, K.: Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption. In: Lin, D., Tsudik, G., Wang, X. (eds.) CANS 2011. LNCS, vol. 7092, pp. 138–159. Springer, Heidelberg (2011)
31. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
32. Shamir, A.: How to share a secret. Communications. ACM 22(11), 612–613 (1979)
33. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
34. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
35. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
36. Yang, P., Cao, Z., Dong, X.: Fuzzy Identity Based Signature. Cryptology ePrint Archive, Report 2008/002 (2008), http://eprint.iacr.org/
37. Zhang, F., Kim, K.: ID-Based Blind Signature and Ring Signature from Pairings. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 533–547. Springer, Heidelberg (2002)