ABE with Tag Made Easy

Concise Framework and New Instantiations in Prime-order Groups

Jie Chen[1,2⋆] and Junqing Gong(✉)[3⋆⋆]

[1] East China Normal University, Shanghai, China
S080001@e.ntu.edu.sg
[2] Jinan University, Guangzhou, China
[3] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France
junqing.gong@ens-lyon.fr

Abstract. Among all existing identity-based encryption (IBE) schemes in the bilinear group, Wat-IBE proposed by Waters [CRYPTO, 2009] and JR-IBE proposed by Jutla and Roy [AsiaCrypt, 2013] are quite special. A secret key and/or ciphertext in these two schemes consist of several group elements and an integer which is usually called *tag*. A series of prior work was devoted to extending them towards more advanced attribute-based encryption (ABE) including inner-product encryption (IPE), hierarchical IBE (HIBE). Recently, Kim *et al.* [SCN, 2016] introduced the notion of tag-based encoding and presented a generic framework for extending Wat-IBE. We may call these ABE schemes *ABE with tag* or *tag-based ABE*. Typically, a tag-based ABE construction is more efficient than its counterpart without tag. However the research on tag-based ABE severely lags—We do not know how to extend JR-IBE in a systematic way and there is no tag-based ABE for boolean span program even with Kim *et al.*'s generic framework.

In this work, we proposed a generic framework for tag-based ABE which is based on JR-IBE and compatible with Chen *et al.*'s (attribute-hiding) predicate encoding [EuroCrypt, 2015]. The adaptive security in the standard model relies on the k-linear assumption in the asymmetric prime-order bilinear group. This is the first framework showing how to extend JR-IBE systematically. In fact our framework and its simple extension are able to cover most concrete tag-based ABE constructions in previous literature. Furthermore, since Chen *et al.*'s predicate encoding supports a large number of predicates including boolean span program, we can now give the first (both key-policy and ciphertext-policy) tag-based ABE for boolean span program in the standard model. Technically our framework is based on a simplified version of JR-IBE. Both the description and its proof are quite similar to the prime-order IBE derived

⋆ School of Computer Science and Software Engineering. Supported by the National Natural Science Foundation of China (Nos. 61472142, 61632012) and the Science and Technology Commission of Shanghai Municipality (No. 14YF1404200). Homepage: http://www.jchen.top

⋆⋆ Partially supported by the French ANR ALAMBIC project (ANR-16-CE39-0006).

from Chen *et al.*'s framework. This not only allows us to work with Chen *et al.*'s predicate encoding but also provides us with a clear explanation of JR-IBE and its proof technique.

Keywords. Attribute-based encryption, Predicate encoding, Prime-order bilinear group, Attribute-hiding, Delegation

1 Introduction

An *attribute-based encryption* [BSW11] (ABE) is an advanced cryptographic primitive supporting fine-grained access control.[4] Such a system is established by an authority (a.k.a. key generation center, KGC for short). On a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ (and other system parameter), the authority publishes master public key mpk. Each user will receive a secret key sk_y associated with a policy $y \in \mathcal{Y}$ when he/she joins in the system. A sender can create a ciphertext ct_x associated with an attribute $x \in \mathcal{X}$. A user holding sk_y can decrypt the ciphertext ct_x if $\mathsf{P}(x,y) = 1$ holds, otherwise he/she will infer nothing about the plaintext. This notion covers many concrete public-key encryptions such as identity-based encryption [Sha84] (IBE), fuzzy IBE [SW05], ABE for boolean span program [GPSW06] and inner-product encryption [KSW08] (IPE).

The basic security requirement is *collusion-resistance*. Intuitively, it is required that two (or more) users who are not authorized to decrypt a ciphertext individually can not do that by collusion either. The notion is formalized via the so-called *adaptive security model* [BF01] where the adversary holding mpk can get secret keys $\mathsf{sk}_{y_1}, \ldots, \mathsf{sk}_{y_q}$ for $y_1, \ldots, y_q \in \mathcal{Y}$ and a challenge ciphertext ct^* for target $x^* \in \mathcal{X}$ via *key extraction queries* and *challenge query* respectively. We emphasize that the adversary can make oracle queries in an adaptive way. Although several weaker security models [CHK03,CW14] were introduced and widely investigated, this paper will focus on the adaptive security model.

Dual System Methodology and Predicate Encoding. A recent breakthrough in this field is the *dual system methodology* invented by Waters [Wat09] in 2009. He obtained the first adaptively secure IBE construction with compact parameters under standard complexity assumptions in the standard model. Inspired by his novel proof technique, the community developed many ABE constructions in the next several years. More importantly, the dual system methodology finally led us to a clean and systematic understanding of ABE. In 2014, Wee [Wee14] and Attrapadung [Att14] introduced the notion of *predicate encod-*

[4] The terminology introduced by Boneh *et al.* [BSW11] is *functional encryption*. However the notion we will consider here is restricted. Therefore we adopt a moderate terminology, i.e., *attribute-based encryption*. We note that the attribute-based encryption in our paper is more general than that introduced by Goyal *et al.* [GPSW06] which is named ABE for boolean formula/span program.

ing[5] and proposed their respective frameworks in the composite-order bilinear group. For certain predicate, if we can construct a predicate encoding for it, their frameworks will immediately give us a full-fledged ABE scheme for the predicate. This significantly simplifies the process of designing an ABE scheme. In fact the powerful frameworks allow them to give many new concrete constructions.

As Wee pointed out in his landmark work [Wee14], the framework reflects the common structures and properties shared among a large group of dual-system ABE schemes while predicate encodings give predicate-dependent features. More concretely, investigating the IBE instance derived from a framework will show the basic construction and proof technique captured the framework, then developing and analysing various predicate encodings will tell us how to extend the IBE instance to more complex cases.

Based on pioneering work by Wee [Wee14] and Attrapadung [Att14], a series of progresses have been made to support more efficient prime-order bilinear groups, employ more advanced proof techniques and more complex predicate encodings [CGW15,AC16,Att16,AC17,AY15].

ABE with Tag. The proof technique behind all these frameworks can be traced back to the work from Lewko and Waters [LW10,LW12] and their variants. However there are two dual-system IBE schemes located beyond these frameworks: one is the first dual-system IBE by Waters [Wat09] and the other one is the IBE scheme based on quasi-adaptive non-interactive zero-knowledge (QA-NIZK) proof by Jutla and Roy [JR13]. In this paper, we call them Wat-IBE and JR-IBE, respectively, for convenience.

Apart from several group elements as usual, a ciphertext and/or a key of Wat-IBE and JR-IBE also include an integer which is called *tag*. This distinguishes them from other dual-system IBE schemes. We must emphasize that the tag plays an important role in the security proof and results in a different proof technique. Therefore we believe they deserve the terminology *IBE with tag* or *tag-based IBE*. Accordingly, an ABE with a tag in the ciphertext and/or key will be called *ABE with tag* or *tag-based ABE*.

There have been several concrete tag-based ABE schemes derived from Wat-IBE or JR-IBE such as [RCS12,RS14,Ram16,RS16]. Recently, Kim *et al.* [KSGA16] introduced the notion of *tag-based encoding* and developed a new framework based on Wat-IBE [Wat09], which is the first systematic study for tag-based ABE. All these work show that a tag-based ABE typically has shorter master public key and ciphertexts/keys, especially for complex predicates. This is a desirable advantage in most application settings of ABE.

[5] Attrapadung [Att14] actually introduced another terminology *pair encoding*. Although predicate encoding and pair encoding serve the same methodology, Attrapadung's framework is based on more advanced proof technique originated from [LW12].

1.1 Motivation

We review previous tag-based ABE in more detail. Ramanna *et al.* [RCS12] simplified Wat-IBE (with stronger assumptions) and extended it to build hierarchical IBE (HIBE) and broadcast encryption. Ramanna and Sarkar [RS14] described two HIBE constructions derived from JR-IBE. Then two IPE schemes were proposed by Ramanna [Ram16]. Although both of them were constructed from JR-IBE, the first one borrows some techniques from Wat-IBE. The recent work [RS16] by Ramanna and Sarkar provided us with two identity-based broadcast encryption schemes, both of which comes from JR-IBE. Kim *et al.*'s tag-based encoding and generic framework [KSGA16] allow them to present several new IPE, (doubly) spatial encryption with various features.

One may notice that there is no framework based on JR-IBE and all previous extensions were obtained in a somewhat ad-hoc manner. This immediately arises our first question.

Question 1: *Is it possible to propose a framework based on* JR-IBE*?*

We note that although both Wat-IBE and JR-IBE take the tag as an important component, their proof techniques are different. That is they use the tag in their own ways in the security proof. As a matter of fact, Wat-IBE requires distinct tags on ciphertext and secret key, respectively, while a secret key in JR-IBE has no tag. To our best knowledge, there is no explicit evidence demonstrating that a framework based on Wat-IBE (like that in [KSGA16]) implies a framework based on JR-IBE.

Furthermore, we surprisingly found that there is no tag-based ABE for boolean span program even with a generic framework! In fact, Kim *et al.* reported that their tag-based encoding is seemingly incompatible with linear secret sharing scheme which is a crucial ingredient of ABE for boolean span program. Hence it's natural to ask the following question.

Question 2: *Is there any limitation on the predicate for tag-based ABE?*

To some extent, we are asking *whether the tag-based proof techniques (used for* Wat-IBE *and* JR-IBE*) makes a trade-off between efficiency and expressiveness?* A very recent work [KSG+17] proposed an ABE with tag supporting boolean span program based on Wat-IBE. However the security analysis was given in the semi-adaptive model [CW14], which is much weaker than the standard adaptive security model (see [GKW16] for more discussions).

1.2 Our Contribution

In this paper, we propose a new framework for tag-based ABE. The framework is based on JR-IBE and can work with the predicate encoding defined in [CGW15]. The adaptive security in the standard model (without random oracles) relies on the k-linear assumption (k-Lin) in the prime-order bilinear

group. Our framework is also compatible with *attribute-hiding predicate encoding* from [CGW15] and implies a family of tag-based ABE with *weak attribute-hiding* feature. Here *weak attribute-hiding* means that ciphertext ct_x reveals no information about x against an adversary who are not authorized to decrypt ct_x.

With this technical result, we are ready to answer the two questions:

Answer to Question 1: Our framework itself readily gives an affirmative answer to the question. Luckily, by defining new concrete predicate encodings motivated by [Ram16,RS16], our framework is able to cover these previous tag-based ABE. In order to capture the HIBE schemes proposed in [RS14], we need to extend both the framework and the predicate encoding to support *predicate with delegation*. (See Section 6.) However we note that Ramanna's first tag-based IPE [Ram16] and Ramanna and Sarkar's second identity-based broadcast encryption with tag [RS16] still fall out of our framework because they involve further developments of JR-IBE's proof technique which has not been captured by our framework.

Answer to Question 2: We highlight that both Chen *et al.*'s framework without tag [CGW15] and our tag-based framework are compatible with the predicate encoding described in [CGW15] (and its attribute-hiding variant). This answers **Question 2** with negation, that is there should be no restriction on predicates for tag-based ABE. Concretely, we can construct a series of new tag-based ABE schemes including ABE for boolean span program (for both key-policy and ciphertext-policy cases; see Section 5) thanks to concrete encodings listed in [CGW15].

We compare our framework with the CGW framework by Chen *et al.* [CGW15] and the KSGA framework by Kim *et al.* [KSGA16] in Table 1. Here we only focus on the space efficiency, i.e., the size of mpk, sk and ct. The comparison regarding the decryption time is analogous to that for the size of ct.

In general, our framework has shorter master public key and shorter ciphertexts than the CGW framework at the cost of slightly larger secret keys. We highlight that the cost we pay here is constant for specific assumption (i.e., consider k as a constant) while the improvement we gain will be proportional to n and $|\mathsf{sE}|$, respectively. In fact our work can be viewed as an improvement of CGW framework using the tag-based technique underlying JR-IBE and improves a family of concrete ABE constructions in a systematic way.

Superficially, the KSGA framework is much more efficient than ours (and CGW framework as well). However, as we have mentioned, this framework is less expressive. In particular, the KSGA framework works with *tag-based predicate encoding* [KSGA16] which fails to support many important predicates such as boolean/arithmetic span program. That is this framework does not imply more efficient ABE for these predicates. It's also worth noting that our framework has shorter ciphertext for concrete predicate encodings with $|\mathsf{sE}| < 5$, including predicate encoding for inner-product encryption with short ciphertext [CGW15]. Namely our framework also implies a more efficient IPE scheme. Actually the CGW framework also has a similar advantage but for predicate encodings with $|\mathsf{sE}| < 3$.

Table 1. Comparison among CGW [CGW15], KSGA [KSGA16], and our framework in terms of space efficiency. All of them work with asymmetric bilinear group (p, G_1, G_2, G_T, e). In the table, $|G_1|$, $|G_2|$ and $|G_T|$ represent the element sizes of three groups, respectively. Parameter n is the number of common parameter of the predicate encoding, $|\mathsf{sE}|$ and $|\mathsf{rE}|$ are the respective size of sender and receiver encodings.

	$\lvert\mathsf{mpk}\rvert$		$\lvert\mathsf{sk}\rvert$		$\lvert\mathsf{ct}\rvert$		Sec.
	$\lvert G_1\rvert$	$\lvert G_T\rvert$	$\lvert G_2\rvert$	$\lvert\mathbb{Z}_p\rvert$	$\lvert G_1\rvert$	$\lvert\mathbb{Z}_p\rvert$	
CGW	$n(k^2+k)+k$	k	$(\lvert\mathsf{rE}\rvert+1)(k+1)$	0	$(\lvert\mathsf{sE}\rvert+1)(k+1)$	0	k-Lin
	$6n+2$	2	$3\cdot\lvert\mathsf{rE}\rvert+3$	0	$3\cdot\lvert\mathsf{sE}\rvert+3$	0	DLIN
	$2n+1$	1	$2\cdot\lvert\mathsf{rE}\rvert+2$	0	$2\cdot\lvert\mathsf{sE}\rvert+2$	0	SXDH
KSGA	$n+11$	1	$\lvert\mathsf{rE}\rvert+7$	$\lvert\mathsf{rE}\rvert$	$\lvert\mathsf{sE}\rvert+8$	$\lvert\mathsf{sE}\rvert$	DLIN
Ours	$(n+1)k^2+k$	k	$(\lvert\mathsf{rE}\rvert+1)(k+1)+k$	0	$\lvert\mathsf{sE}\rvert\cdot k+k+1$	$\lvert\mathsf{sE}\rvert$	k-Lin
	$4n+6$	2	$3\cdot\lvert\mathsf{rE}\rvert+5$	0	$2\cdot\lvert\mathsf{sE}\rvert+3$	$\lvert\mathsf{sE}\rvert$	DLIN
	$n+2$	1	$2\cdot\lvert\mathsf{rE}\rvert+3$	0	$\lvert\mathsf{sE}\rvert+2$	$\lvert\mathsf{sE}\rvert$	SXDH

In Table 2, we compare concrete instantiations derived from CGW, KSGA, and our framework. Here we only take inner-product encryption (IPE) and key-policy ABE for boolean span program (KP-ABE-BSP) as examples. It is clear that our KP-ABE-BSP has the shortest master public key and ciphertexts (also fastest decryption algorithm). Under the DLIN assumption, our IPE has the shortest ciphertext but its master public key is larger than the IPE derived from KSGA framework. However if we are allowed to use stronger SXDH assumption, our IPE will have the shortest master public key.

1.3 Overview of Method: A Simplified JR-IBE

Our framework is based on JR-IBE. In Jutla and Roy's original paper [JR13], JR-IBE was derived from the QA-NIZK proof for a specific subspace language. Although it's important to describe/explain an IBE from the angle of NIZK proof, it's still a big challenge to work on it directly. Therefore the foundation of our framework is a simplified (and slightly generalized) version of JR-IBE. The simplified JR-IBE is similar to the prime-order IBE instantiated from the CGW framework [CGW15] and its proof analysis is cleaner and much easier to follow. With the benefits of these features, we are able to develop our new framework for tag-based ABE which is based on JR-IBE's proof technique [JR13] and is compatible with the predicate encoding from [CGW15]. The adaptive security relies on the k-Lin assumption, which is a generalized form of SXDH used in [JR13].

Simplified JR-IBE. We assume there is an asymmetric prime-order bilinear group (p, G_1, G_2, G_T, e). Let $g_1 \in G_1$, $g_2 \in G_2$ and $e(g_1, g_2) \in G_T$ be the respec-

Table 2. Comparison among CGW [CGW15], KSGA [KSGA16], and ours framework in terms of concrete instantiations. In the table, $|G_1|$, $|G_2|$ and $|G_T|$ have the same meanings as Table 1. The parameter ℓ is the dimension of vector space for IPE while it stands for the size of universe for ABE.

	IPE						KP-ABE-BSP						Sec.
	$\|\mathsf{mpk}\|$		$\|\mathsf{sk}\|$		$\|\mathsf{ct}\|$		$\|\mathsf{mpk}\|$		$\|\mathsf{sk}\|$		$\|\mathsf{ct}\|$		
	$\|G_1\|$	$\|G_T\|$	$\|G_2\|$	$\|\mathbb{Z}_p\|$	$\|G_1\|$	$\|\mathbb{Z}_p\|$	$\|G_1\|$	$\|G_T\|$	$\|G_2\|$	$\|\mathbb{Z}_p\|$	$\|G_1\|$	$\|\mathbb{Z}_p\|$	
CGW	$6\ell+2$	2	$3\ell+6$	0	6	0	$6\ell+2$	2	$3\ell+3$	0	$3\ell+3$	0	DLIN
	$2\ell+1$	1	$2\ell+4$	0	4	0	$2\ell+1$	1	$2\ell+2$	0	$2\ell+2$	0	SXDH
KSGA	$\ell+11$	1	$\ell+6$	$\ell-1$	9	1	—						DLIN
Ours	$4\ell+6$	2	$3\ell+8$	0	5	1	$4\ell+6$	2	$3\ell+5$	0	$2\ell+3$	ℓ	DLIN
	$\ell+2$	1	$2\ell+5$	0	3	1	$\ell+2$	1	$2\ell+3$	0	$\ell+2$	ℓ	SXDH

tive generators, we will use the following notation: $[a]_s = g_s^a$ for all $a \in \mathbb{Z}_p$ and $s \in \{1, 2, T\}$. The notation can also be naturally applied to a matrix over $\mathbb{Z}_p$.

Let $\mathbb{Z}_p$ be the identity space. The JR-IBE can be re-written as

$$\begin{aligned}
\mathsf{mpk} &: [\mathbf{A}]_1,\ [\mathbf{W}_0^\top \mathbf{A}]_1,\ [\mathbf{W}_1^\top \mathbf{A}]_1,\ \boxed{[\mathbf{W}^\top \mathbf{A}]_1},\ [\mathbf{k}^\top \mathbf{A}]_T \\
\mathsf{ct}_{\mathsf{id}} &: [\mathbf{A}\mathbf{s}]_1,\ [(\mathbf{W}_0 + \mathsf{id} \cdot \mathbf{W}_1 + \boxed{\tau \cdot \mathbf{W}})^\top \mathbf{A}\mathbf{s}]_1,\ [\mathbf{k}^\top \mathbf{A}\mathbf{s}]_T \cdot m,\ \tau \\
\mathsf{sk}_{\mathsf{id}} &: [\mathbf{r}]_2,\ [\mathbf{k} + (\mathbf{W}_0 + \mathsf{id} \cdot \mathbf{W}_1)\mathbf{r}]_2,\ \boxed{[\mathbf{W}\mathbf{r}]_2}
\end{aligned}$$

Here $\mathbf{A} \leftarrow \mathbb{Z}_p^{(k+1)\times k}$ acts as the basis, matrices $\mathbf{W}_0, \mathbf{W}_1, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times k}$ are common parameters, the vector $\mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$ is the master secret value, vectors $\mathbf{s}, \mathbf{r} \leftarrow \mathbb{Z}_p^k$ are random coins for the ciphertext and the secret key, respectively, and the tag τ is a random element in $\mathbb{Z}_p$.

The boxed parts involving matrix $\mathbf{W}$ are relevant to the tag while the remaining structure is quite similar to the IBE implied by the CGW framework. The main difference is that we do not need another basis $\mathbf{B}$ in $\mathsf{sk}_{\mathsf{id}}$, which reduces the size of $\mathbf{W}, \mathbf{W}_0, \mathbf{W}_1$ and thus shortens mpk and $\mathsf{ct}_{\mathsf{id}}$. In fact this structure has been used in a recent tightly secure IBE [BKP14,GDCC16], and we indeed borrow some proof technique from the tight reduction method there.

Proof Blueprint. Our proof mainly follows the tag-based proof strategy given by Jutla and Roy [JR13] which basically employs the dual system methodology [Wat09]. The first step is to transform the normal challenge ciphertext (see the real system above) into the semi-functional (SF) one:

$$\mathsf{ct}_{\mathsf{id}^*}^{\mathrm{SF}} : [\mathbf{A}\mathbf{s} + \boxed{\mathbf{b}\hat{s}}]_1,\ [(\mathbf{W}_0 + \mathsf{id}^* \cdot \mathbf{W}_1 + \tau \cdot \mathbf{W})^\top (\mathbf{A}\mathbf{s} + \boxed{\mathbf{b}\hat{s}})]_1,\ [\mathbf{k}^\top (\mathbf{A}\mathbf{s} + \boxed{\mathbf{b}\hat{s}})]_T \cdot m,\ \tau$$

where $\mathbf{b} \leftarrow \mathbb{Z}_p^{k+1}$ and $\hat{s} \leftarrow \mathbb{Z}_p$. One may prove such a transformation is not detectable by the adversary from the k-Lin assumption following [CGW15]. The

next step is to convert secret keys revealed to adversary from normal form into semi-functional (SF) form defined as follows:

$$\mathsf{sk}_{\mathsf{id}}^{\mathrm{SF}} \ : \ [\mathbf{r}]_2,\ [\mathbf{k} + \boxed{\alpha \mathbf{a}^{\perp}} + (\mathbf{W}_0 + \mathsf{id} \cdot \mathbf{W}_1)\mathbf{r}]_2,\ [\mathbf{W}\mathbf{r}]_2$$

where $\mathbf{a}^{\perp} \leftarrow \mathbb{Z}_p^{k+1}$ with $\mathbf{A}^{\top}\mathbf{a}^{\perp} = \mathbf{0}$ and $\alpha \leftarrow \mathbb{Z}_p$. As usual, the conversion will be done in a one-by-one manner. That is we deal with single secret key each time which arises a security loss proportional to the number of secret keys sent to the adversary. When all secret keys and the challenge ciphertext become semi-functional, it would be quite direct to decouple the message m from the challenge ciphertext which implies the adaptive security.

Tag-based Technique, Revisited. In fact what we have described still follows [CGW15] (and most dual-system proofs [Wat09]). However the proof for the indistinguishability between normal and SF secret key heavily relies on the tag as [JR13] and deviate from [CGW15].

Recall that a SF key is a normal key with additional entropy $\alpha\mathbf{a}^{\perp}$. To replace a normal key with a SF key, we use the following lemma in [BKP14,GDCC16] which states that

$$(\ [\mathbf{r}]_2,\ [\mathbf{v}^{\top}\mathbf{r}]_2\ ) \approx_c (\ [\mathbf{r}]_2,\ [\mathbf{v}^{\top}\mathbf{r} + \hat{r}]_2\ ) \quad \text{given} \quad [\mathbf{Z}]_2, [\mathbf{v}^{\top}\mathbf{Z}]_2 \tag{1}$$

where $\mathbf{v}, \mathbf{r} \leftarrow \mathbb{Z}_p^k$, $\hat{r} \leftarrow \mathbb{Z}_p$ and $\mathbf{Z} \leftarrow \mathbb{Z}_p^{k \times k}$. Following [BKP14,GDCC16], we pick $\gamma_0, \gamma_1 \leftarrow \mathbb{Z}_p$ and embed the secret vector $\mathbf{v}$ into common parameter $\mathbf{W}_0$ and $\mathbf{W}_1$ as follows

$$\mathbf{W}_0 = \widetilde{\mathbf{W}}_0 + \gamma_0 \cdot \mathbf{a}^{\perp}\mathbf{v}^{\top} \quad \text{and} \quad \mathbf{W}_1 = \widetilde{\mathbf{W}}_1 + \gamma_1 \cdot \mathbf{a}^{\perp}\mathbf{v}^{\top}$$

where $\widetilde{\mathbf{W}}_0, \widetilde{\mathbf{W}}_1 \leftarrow \mathbb{Z}_p^{(k+1)\times k}$. The lemma shown in Equation (1) will allow us to move from a normal key to the following transitional form

$$[\mathbf{r}]_2,\ [\mathbf{k} + (\mathbf{W}_0 + \mathsf{id} \cdot \mathbf{W}_1)\mathbf{r} + \boxed{(\gamma_0 + \mathsf{id} \cdot \gamma_1) \cdot \mathbf{a}^{\perp}\hat{r}}]_2,\ [\mathbf{W}\mathbf{r}]_2. \tag{2}$$

However we must ensure that $\mathbf{v}$ will not appear in mpk and $\mathsf{ct}_{\mathsf{id}}$, both of which consist of elements in G_1; otherwise, we can not apply the lemma at all since there is not element from G_1 with information on $\mathbf{v}$.

It is direct to see that $[\mathbf{W}_0^{\top}\mathbf{A}]_1$ and $[\mathbf{W}_1^{\top}\mathbf{A}]_1$ in mpk reveal nothing about $\mathbf{v}$ from the fact that $\mathbf{A}^{\top}\mathbf{a}^{\perp} = \mathbf{0}$, but the challenge ciphertext may include $\mathbf{v}$ since we have

$$\mathbf{W}_0 + \mathsf{id}^* \cdot \mathbf{W}_1 + \tau \cdot \mathbf{W} = \widetilde{\mathbf{W}}_0 + \mathsf{id}^* \cdot \widetilde{\mathbf{W}}_1 + \tau \cdot \mathbf{W} + \boxed{(\gamma_0 + \mathsf{id}^* \cdot \gamma_1)\mathbf{a}^{\perp}\mathbf{v}^{\top}}$$

and $\mathbf{A}\mathbf{s} + \mathbf{b}\hat{s} \in \mathbb{Z}_p^{k+1}$ with high probability. Fortunately, the tag can help us to circumvent the issue: set

$$\mathbf{W} = \widetilde{\mathbf{W}} - \mathbf{a}^{\perp}\mathbf{v}^{\top} \quad \text{and} \quad \tau = \gamma_0 + \mathsf{id}^* \cdot \gamma_1,$$

we can see that

$$\tau \cdot \mathbf{W} + (\gamma_0 + \mathsf{id}^* \cdot \gamma_1)\mathbf{a}^\perp \mathbf{v}^\top = \tau \cdot \widetilde{\mathbf{W}}$$

where there is no $\mathbf{v}$ anymore and the proof strategy now works well. We note that both $\mathbf{W}$ and τ are distributed correctly.

Then we can move from the transitional key (see Equation 2) to the SF key using the statistical argument asserting that

$$\{\ \gamma_0 + \mathsf{id}^* \cdot \gamma_1,\ \ \gamma_0 + \mathsf{id} \cdot \gamma_1\ \}$$

are uniformly distributed over $\mathbb{Z}_p^2$. Chen *et al.*'s proof [CGW15] also involves this statistical argument. However, in Chen *et al.*'s proof [CGW15], both values appear "on the exponent" while $\gamma_0 + \mathsf{id}^* \cdot \gamma_1$ is given out "directly" as the tag in our case.

1.4 Related Work and Discussion

Since the work by Wee [Wee14] and Attrapadung [Att14], predicate/pair encodings and corresponding generic frameworks have been extended and improved via various methods [CGW15,AC16,Att16,AC17,ABS17]. Before that, there were many early-age ABE from pairing such as IBE [BF01,BB04b,BB04a,Wat05,Gen06], fuzzy IBE [SW05], inner-product encryption [KSW08,OT12], and ABE for boolean formula [GPSW06,OSW07,BSW07,OT10,LOS+10,Wat11,LW12]. Most of them has been covered by generic frameworks. Attrapadung *et al.* [AHY15] even gave a generic framework for *tightly secure* IBE based on broadcast encodings and reached a series of interesting constructions. However IBE schemes with exponent-inverse structure [Gen06,Wee16,CGW17] are out of the scope of current predicate encodings, and we still have no framework supporting *fully* attribute-hiding feature [OT12].

JR-IBE has been used to construct more advanced primitives [WS16,WES17]. We note that our framework is seemingly not powerful enough to cover them. However we believe our simplified JR-IBE (and our framework as well) can shed the light on more extensions in the future.

Organization. Our paper is organized as follows. The next section will give several basic notions. Our generic framework for ABE with tag will be given in Section 3. We also prove its adaptive security in the same section. We then show the compatibility of our framework with attribute-hiding encodings in Section 4. The next section, Section 5, illustrates the new ABE constructions derived from our framework. The last section, Section 6, shows how to extend our framework to support delegation.

2 Preliminaries

Notation. We use $s \leftarrow S$ to indicate that s is selected uniformly from finite set S. For a probability distribution $\mathcal{D}$, notation $x \leftarrow \mathcal{D}$ means that x is sampled

according to $\mathcal{D}$. We consider λ as security parameter and a function $f(\lambda)$ is negligible in λ if, for each $c \in \mathbb{N}$, there exists λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$. "p.p.t." stands for "probabilistic polynomial time". For a matrix $\mathbf{A} \in \mathbb{Z}_p^{k \times k'}$ with $k > k'$, we let $\overline{\mathbf{A}}$ be the matrix consist of the first k' rows and $\underline{\mathbf{A}}$ be the matrix with all remaining rows.

2.1 Attribute-Based Encryptions

Syntax. An attribute-based encryption (ABE) scheme for predicate $\mathsf{P}(\cdot,\cdot)$ consists of the following p.p.t. algorithms.

- $\mathsf{Setup}(1^\lambda, \mathsf{P}) \to (\mathsf{mpk}, \mathsf{msk})$. The *setup* algorithm takes as input the security parameter λ and a description of predicate P and returns master public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$. We assume that mpk contains descriptions of domains $\mathcal{X}$ and $\mathcal{Y}$ of P as well as message space $\mathcal{M}$.
- $\mathsf{Enc}(\mathsf{mpk}, x, m) \to \mathsf{ct}_x$. The *encryption* algorithm takes as input the master public key mpk, an index (attribute) $x \in \mathcal{X}$ and a message $m \in \mathcal{M}$ and outputs a ciphertext ct_x.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y) \to \mathsf{sk}_y$. The *key generation* algorithm takes as input the master public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$ and an index (policy) $y \in \mathcal{Y}$ and generates a secret key sk_y.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x) \to m$. The *decryption* algorithm takes as input the master public key mpk, a secret key sk_y and a ciphertext ct_x with $\mathsf{P}(x, y) = 1$ and outputs message m.

Correctness. For all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{P})$, all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ satisfying $\mathsf{P}(x, y) = 1$, and all $m \in \mathcal{M}$, it is required that

$$\Pr\left[\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x) = m \;\middle|\; \begin{array}{c} \mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y) \\ \mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, m) \end{array}\right] = 1.$$

Security. For all adversary $\mathcal{A}$, define advantage function $\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A}}(\lambda)$ as follows.

$$\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\beta = \beta' \;\middle|\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{P}),\ \beta \leftarrow \{0,1\} \\ (x^*, m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{mpk}) \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^*, m_\beta^*) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{mpk}, \mathsf{ct}^*) \end{array}\right] - \frac{1}{2}\right|.$$

An ABE scheme is said to be *adaptively secure* if $\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A}}(\lambda)$ is negligible in λ and $\mathsf{P}(x^*, y) = 0$ holds for each query y sent to oracle $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \cdot)$ for all p.p.t. adversary $\mathcal{A}$. We may call x^* *the target attribute* and each y a *key extraction query.*

2.2 Prime-order Bilinear Groups and Cryptographic Assumption

We assume a group generator GrpGen which takes as input security parameter 1^λ and outputs group description $\mathcal{G} = (p, G_1, G_2, G_T, e)$. Here G_1, G_2, G_T are cyclic groups of prime order p of $\Theta(\lambda)$ bits and $e : G_1 \times G_2 \to G_T$ is a non-degenerated bilinear map. We assume that descriptions of G_1 and G_2 contain respective generators g_1 and g_2.

Let $s \in \{1, 2, T\}$. For $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_p^{k \times k'}$, we define the *implicit representation* [EHK+13] as

$$[\mathbf{A}]_s = \begin{pmatrix} g_s^{a_{11}} & \cdots & g_s^{a_{1k'}} \\ \vdots & & \vdots \\ g_s^{a_{k1}} & \cdots & g_s^{a_{kk'}} \end{pmatrix} \in G_s^{k \times k'}.$$

Given $[\mathbf{A}]_1 \in G_1^{k \times n}$ and $[\mathbf{B}]_2 \in G_2^{k \times n'}$, define $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{A}^\top \mathbf{B}]_T \in G_T^{n \times n'}$.

We also use the following notations: given $[\mathbf{a}]_s, [\mathbf{b}]_s \in G_s^k$ and $c \in \mathbb{Z}_p$, define

$$[\mathbf{a}]_s \cdot [\mathbf{b}]_s = [\mathbf{a} + \mathbf{b}]_s \in G_s^k \quad \text{and} \quad [\mathbf{a}]_s^c = [c\mathbf{a}]_s \in G_s^k; \tag{3}$$

for $([\mathbf{a}_1]_s, \ldots, [\mathbf{a}_n]_s), ([\mathbf{b}_1]_s, \ldots, [\mathbf{b}_n]_s) \in (G_s^k)^n$, we define

$$([\mathbf{a}_1]_s, \ldots, [\mathbf{a}_n]_s) \cdot ([\mathbf{b}_1]_s, \ldots, [\mathbf{b}_n]_s) = ([\mathbf{a}_1]_s \cdot [\mathbf{b}_1]_s, \ldots, [\mathbf{a}_n]_s \cdot [\mathbf{b}_n]_s) \in (G_s^k)^n;$$

for $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_p^n$ and $[\mathbf{b}]_s \in G_s^k$, we define

$$[\mathbf{b}]_s^{\mathbf{a}} = ([\mathbf{b}]_s^{a_1}, \ldots, [\mathbf{b}]_s^{a_n}) \in (G_s^k)^n. \tag{4}$$

Let $\mathcal{D}_k$ be a matrix distribution sampling matrix $\mathbf{A} \in \mathbb{Z}_p^{(k+1) \times k}$ along with a non-zero vector $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}$ satisfying $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$. We need the following lemma with respect to $\mathcal{D}_k$.

Lemma 1 (Basic Lemma, [GDCC16,GHKW16]). *With probability* $1 - 1/p$ *over* $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$ *and* $\mathbf{b} \leftarrow \mathbb{Z}_p^{k+1}$*, it holds that*

$$\mathbf{b} \notin \mathsf{span}(\mathbf{A}) \quad \textit{and} \quad \mathbf{b}^\top \mathbf{a}^\perp \neq 0.$$

We review the matrix decisional Diffie-Hellman (MDDH) assumption in the prime-order bilinear groups as follows.

Assumption 1 ($\mathcal{D}_k$-MDDH, [EHK+13]) *Let* $s \in \{1, 2\}$*. For any adversary* $\mathcal{A}$*, define the advantage function* $\mathsf{Adv}_{\mathcal{A}}^{\mathcal{D}_k}(\lambda)$ *as follows*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathcal{D}_k}(\lambda) = |\Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_s, [\mathbf{A}\mathbf{s}]_s) = 1] - \Pr[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_s, [\mathbf{u}]_s) = 1]|$$

where $\mathcal{G} \leftarrow \mathsf{GrpGen}(1^\lambda)$*,* $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$*,* $\mathbf{s} \leftarrow \mathbb{Z}_p^k$ *and* $\mathbf{u} \leftarrow \mathbb{Z}_p^{k+1}$*. The assumption says that* $\mathsf{Adv}_{\mathcal{A}}^{\mathcal{D}_k}(\lambda)$ *is negligible in* λ *for all p.p.t. adversary* $\mathcal{A}$*.*

2.3 Predicate Encodings

This subsection reviews the notion of predicate encoding [Wee14,CGW15] and shows some useful notations and facts.

Definition. A $\mathbb{Z}_p$-linear predicate encoding for $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ consists of five deterministic algorithms

$$\begin{array}{lll} \mathsf{sE} : \mathcal{X} \times \mathbb{Z}_p^n \to \mathbb{Z}_p^{n_s} & & \mathsf{sD} : \mathcal{X} \times \mathcal{Y} \times \mathbb{Z}_p^{n_s} \to \mathbb{Z}_p \\ \mathsf{rE} : \mathcal{Y} \times \mathbb{Z}_p^n \to \mathbb{Z}_p^{n_r} & \mathsf{kE} : \mathcal{Y} \times \mathbb{Z}_p \to \mathbb{Z}_p^{n_r} & \mathsf{rD} : \mathcal{X} \times \mathcal{Y} \times \mathbb{Z}_p^{n_r} \to \mathbb{Z}_p \end{array}$$

for some $n, n_s, n_r \in \mathbb{N}$ with the following features:

(linearity) For all $(x,y) \in \mathcal{X} \times \mathcal{Y}$, $\mathsf{sE}(x,\cdot)$, $\mathsf{rE}(y,\cdot)$, $\mathsf{kE}(y,\cdot)$, $\mathsf{sD}(x,y,\cdot)$, $\mathsf{rD}(x,y,\cdot)$ are $\mathbb{Z}_p$-linear. A $\mathbb{Z}_p$-linear function $L : \mathbb{Z}_p^n \to \mathbb{Z}_p^{n'}$ can be encoded as a matrix $\mathbf{L} = (l_{i,j}) \in \mathbb{Z}_p^{n \times n'}$ such that

$$L : (w_1, \ldots, w_n) \mapsto (\textstyle\sum_{i=1}^n l_{i1} w_i, \ldots, \sum_{i=1}^n l_{in'} w_i). \tag{5}$$

(restricted α-reconstruction) For all $(x,y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x,y) = 1$, all $\mathbf{w} \in \mathbb{Z}_p^n$ and all $\alpha \in \mathbb{Z}_p$, we have

$$\mathsf{sD}(x, y, \mathsf{sE}(x, \mathbf{w})) = \mathsf{rD}(x, y, \mathsf{rE}(y, \mathbf{w})) \quad \text{and} \quad \mathsf{rD}(x, y, \mathsf{kE}(y, \alpha)) = \alpha.$$

(α-privacy) For all $(x,y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x,y) = 0$ and all $\alpha \in \mathbb{Z}_p$, the following distributions are identical.

$$\begin{aligned} &\{\ x, y, \alpha, \mathsf{sE}(x, \mathbf{w}), \mathsf{kE}(y, \alpha) + \mathsf{rE}(y, \mathbf{w}) : \mathbf{w} \leftarrow \mathbb{Z}_p^n\ \} \quad \text{and} \\ &\{\ x, y, \alpha, \mathsf{sE}(x, \mathbf{w}), \mathsf{rE}(y, \mathbf{w}) : \mathbf{w} \leftarrow \mathbb{Z}_p^n\ \}. \end{aligned}$$

We call n the *parameter size* and use $|\mathsf{sE}|$ and $|\mathsf{rE}|$ to denote n_s and n_r, respectively, which indicate the *sizes of sender encodings and receiver encoding*. We note that $|\mathsf{sE}|$ and $|\mathsf{rE}|$ may depend on x and y respectively. For all $x \in \mathcal{X}$, we define the distribution

$$\mathsf{sE}(x) := \{\ \mathsf{sE}(x, \mathbf{w}) : \mathbf{w} \leftarrow \mathbb{Z}_p^n\ \}.$$

More Notations and Useful Facts. Assume $s \in \{1, 2, T\}$. We can naturally define a series of $\mathbb{Z}_p$-linear functions from $\mathbf{L}$ as follows:

$$\begin{array}{rccc} L : & (G_s^k)^n & \to & (G_s^k)^{n'} \\ & ([\mathbf{w}_1]_s, \ldots, [\mathbf{w}_n]_s) & \mapsto & (\prod_{i=1}^n [\mathbf{w}_i]_s^{l_{i1}}, \ldots, \prod_{i=1}^n [\mathbf{w}_i]_s^{l_{in'}}) \end{array} \tag{6}$$

Because they essentially share the same structure (i.e., $\mathbf{L}$), we employ the same notation L. It should be clear from the context. Then we highlight three properties regarding functions L with respective to the same $\mathbf{L}$ as follows:

($L(\cdot)$ and pairing e are commutative) For any $\mathbf{a}, \mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}_p^k$, we have

$$e([\mathbf{a}]_1, L([\mathbf{b}_1]_2, \ldots, [\mathbf{b}_n]_2)) = L(e([\mathbf{a}]_1, [\mathbf{b}_1]_2), \ldots, e([\mathbf{a}]_1, [\mathbf{b}_n]_2)) \tag{7}$$

$$e(L([\mathbf{b}_1]_1, \ldots, [\mathbf{b}_n]_1), [\mathbf{a}]_2) = L(e([\mathbf{b}_1]_1, [\mathbf{a}]_2), \ldots, e([\mathbf{b}_n]_1, [\mathbf{a}]_2)) \tag{8}$$

($L(\cdot)$ and $[\cdot]_s$ are commutative) For any $w_1, \ldots, w_n \in \mathbb{Z}_p$, we have

$$L([w_1]_s, \ldots, [w_n]_s) = [L(w_1, \ldots, w_n)]_s. \tag{9}$$

($L(\cdot)$ and "exponentiation" are commutative) For any $\mathbf{w} \in \mathbb{Z}_p^n$ and $[\mathbf{a}]_s \in G_s^k$, we have

$$[\mathbf{a}]_s^{L(\mathbf{w})} = L([\mathbf{a}]_s^{\mathbf{w}}). \tag{10}$$

Concrete Instantiations. As an example, we show the predicate encoding for equality predicate as below. This is the simplest encoding and extracted from classical Lewko-Waters IBE [LW10].

(encoding for equality [LW10]) Let $\mathcal{X} = \mathcal{Y} = \mathbb{Z}_p$ and $\mathsf{P}(x, y) = 1$ iff $x = y$. Let $n = 2$ and $w_1, w_2 \leftarrow \mathbb{Z}_p$. Define

$$\begin{array}{lll} \mathsf{sE}(x, (w_1, w_2)) := w_1 + xw_2 & & \mathsf{sD}(x, y, c) := c \\ \mathsf{rE}(y, (w_1, w_2)) := w_1 + yw_2 & \mathsf{kE}(y, \alpha) := \alpha & \mathsf{rD}(x, y, k) := k \end{array}$$

Here we have $|\mathsf{sE}| = |\mathsf{rE}| = 1$.

3 ABE with Tags from Predicate Encodings

3.1 Construction

Our generic tag-based ABE from predicate encodings is described below.

- $\mathsf{Setup}(1^\lambda, \mathsf{P})$: Let n be parameter size of predicate encoding $(\mathsf{sE}, \mathsf{rE}, \mathsf{kE}, \mathsf{sD}, \mathsf{rD})$ for P. We sample

$$\mathbf{A} \leftarrow \mathcal{D}_k, \quad \mathbf{W}_1, \cdots, \mathbf{W}_n, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \quad \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$$

and output the master public and secret key pair

$$\begin{aligned} \mathsf{mpk} &:= \{\ [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \cdots, [\mathbf{W}_n^\top \mathbf{A}]_1, [\mathbf{W}^\top \mathbf{A}]_1, [\mathbf{k}^\top \mathbf{A}]_T\ \} \\ \mathsf{msk} &:= \{\ \mathbf{W}_1, \cdots, \mathbf{W}_n, \mathbf{W};\ \mathbf{k}\ \}. \end{aligned}$$

- $\mathsf{Enc}(\mathsf{mpk}, x, m)$: On input $x \in \mathcal{X}$ and $m \in G_T$, pick $\mathbf{s} \leftarrow \mathbb{Z}_p^k$ and $\boldsymbol{\tau} \leftarrow \mathsf{sE}(x)$. Output

$$\mathsf{ct}_x := \left\{ \begin{array}{l} C_0 := [\mathbf{As}]_1, \\ \mathbf{C}_1 := \mathsf{sE}(x, [\mathbf{W}_1^\top \mathbf{As}]_1, \ldots, [\mathbf{W}_n^\top \mathbf{As}]_1) \cdot [\mathbf{W}^\top \mathbf{As}]_1^{\boldsymbol{\tau}}, \\ C := [\mathbf{k}^\top \mathbf{As}]_T \cdot m,\ \boldsymbol{\tau} \end{array} \right\}$$

- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$: On input $y \in \mathcal{Y}$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{sk}_y := \left\{ \begin{array}{l} K_0 := [\mathbf{r}]_2, \\ \mathbf{K}_1 := \mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r}]_2, \ldots, [\mathbf{W}_n \mathbf{r}]_2), \\ K_2 := [\mathbf{Wr}]_2 \end{array} \right\}$$

- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x)$: Compute

$$K \leftarrow e(C_0, \mathsf{rD}(x, y, \mathbf{K}_1)) \cdot e(C_0, K_2)^{\mathsf{sD}(x,y,\boldsymbol{\tau})} / e(\mathsf{sD}(x, y, \mathbf{C}_1), K_0)$$

and recover the message as $m \leftarrow C/K \in G_T$.

Correctness. For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $\mathsf{P}(x, y) = 1$, we may use the following abbreviation

$$\mathsf{sE}(x, \cdot) = \mathsf{sE}(\cdot),\ \mathsf{rE}(y, \cdot) = \mathsf{rE}(\cdot),\ \mathsf{kE}(y, \cdot) = \mathsf{kE}(\cdot);$$
$$\mathsf{sD}(x, y, \cdot) = \mathsf{sD}(\cdot),\ \mathsf{rD}(x, y, \cdot) = \mathsf{rD}(\cdot)$$

and have

$$\begin{aligned}
& e(C_0, \mathsf{rD}(\mathbf{K}_1)) \cdot e(C_0, K_2)^{\mathsf{sD}(\boldsymbol{\tau})} \\
\overset{(a)}{=}\ & e([\mathbf{As}]_1, \mathsf{rD}(\mathsf{kE}([\mathbf{k}]_2))) \cdot e([\mathbf{As}]_1, \mathsf{rD}(\mathsf{rE}([\mathbf{W}_1\mathbf{r}]_2, \ldots, [\mathbf{W}_n\mathbf{r}]_2))) \cdot [\mathbf{s}^\top\mathbf{A}^\top\mathbf{Wr}]_T^{\mathsf{sD}(\boldsymbol{\tau})} \\
\overset{(b)}{=}\ & \mathsf{rD}(\mathsf{kE}([\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}]_T)) \cdot \mathsf{rD}(\mathsf{rE}([\mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_1\mathbf{r}]_T, \ldots, [\mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_n\mathbf{r}]_T)) \cdot [\mathbf{s}^\top\mathbf{A}^\top\mathbf{Wr}]_T^{\mathsf{sD}(\boldsymbol{\tau})} \\
\overset{(c)}{=}\ & [\mathsf{rD}(\mathsf{kE}(\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}))]_T \cdot [\mathsf{rD}(\mathsf{rE}(\mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_1\mathbf{r}, \ldots, \mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_n\mathbf{r}))]_T \cdot [\mathbf{s}^\top\mathbf{A}^\top\mathbf{Wr}]_T^{\mathsf{sD}(\boldsymbol{\tau})} \\
\overset{(d)}{=}\ & [\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}]_T \cdot [\mathsf{sD}(\mathsf{sE}(\mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_1\mathbf{r}, \ldots, \mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_n\mathbf{r}))]_T \cdot [\mathbf{s}^\top\mathbf{A}^\top\mathbf{Wr}]_T^{\mathsf{sD}(\boldsymbol{\tau})} \\
\overset{(e)}{=}\ & [\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}]_T \cdot \mathsf{sD}(\mathsf{sE}([\mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_1\mathbf{r}]_T, \ldots, [\mathbf{s}^\top\mathbf{A}^\top\mathbf{W}_n\mathbf{r}]_T)) \cdot [\mathbf{s}^\top\mathbf{A}^\top\mathbf{Wr}]_T^{\mathsf{sD}(\boldsymbol{\tau})} \\
\overset{(f)}{=}\ & [\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}]_T \cdot e(\mathsf{sD}(\mathsf{sE}([\mathbf{W}_1^\top\mathbf{As}]_1, \ldots, [\mathbf{W}_n^\top\mathbf{As}]_1)), [\mathbf{r}]_2) \cdot e([\mathbf{W}^\top\mathbf{As}]_1^{\mathsf{sD}(\boldsymbol{\tau})}, [\mathbf{r}]_2) \\
\overset{(g)}{=}\ & [\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}]_T \cdot e(\mathsf{sD}(\mathsf{sE}([\mathbf{W}_1^\top\mathbf{As}]_1, \ldots, [\mathbf{W}_n^\top\mathbf{As}]_1)), [\mathbf{r}]_2) \cdot e(\mathsf{sD}([\mathbf{W}^\top\mathbf{As}]_1^{\boldsymbol{\tau}}), [\mathbf{r}]_2) \\
\overset{(h)}{=}\ & [\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}]_T \cdot e(\mathsf{sD}(\mathsf{sE}([\mathbf{W}_1^\top\mathbf{As}]_1, \ldots, [\mathbf{W}_n^\top\mathbf{As}]_1) \cdot [\mathbf{W}^\top\mathbf{As}]_1^{\boldsymbol{\tau}}), [\mathbf{r}]_2) \\
=\ & [\mathbf{s}^\top\mathbf{A}^\top\mathbf{k}]_T \cdot e(\mathsf{sD}(\mathbf{C}_1), K_0)
\end{aligned}$$

which is sufficient for the correctness. We list the properties and the facts justifying each (labelled) equality in Table 3.

Security Result. We give the following main theorem stating that the above generic tag-based ABE scheme is adaptively secure under standard assumption in the standard model. The remaining of this section will be devoted to the proof of the main theorem.

Theorem 1 (Main Theorem). *For any p.p.t. adversary $\mathcal{A}$ making at most q key extraction queries, there exists algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{ABE}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathcal{D}_k}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{B}_2}^{\mathcal{D}_k}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{B}_3}^{\mathcal{D}_k}(\lambda) + 2^{-\Omega(\lambda)}$$

and $\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot k^2 \cdot \mathrm{poly}(\lambda, n)$.

3.2 Proving the Main Theorem: High-level roadmap

From a high level, the proof basically follows the common dual system methodology. We first define semi-functional distributions under the master public key

$$\mathsf{mpk} = \{\ [\mathbf{A}]_1, [\mathbf{W}_1^\top\mathbf{A}]_1, \ldots, [\mathbf{W}_n^\top\mathbf{A}]_1, [\mathbf{W}^\top\mathbf{A}]_1, [\mathbf{k}^\top\mathbf{A}]_T\ \}.$$

where $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k, \mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$ as follows.

Table 3. Properties and Facts for Correctness.

Eq.	Properties	Facts
(a)	Composition of two linear functions are linear.	Both $e([\mathbf{As}]_1, \cdot)$ and $\mathsf{rD}(x, y, \cdot)$ are linear functions.
(b)	Linear functions and pairings are commutative. (cf. Eq.(7))	Both $\mathsf{rD}(x, y, \mathsf{kE}(y, \cdot))$ and $\mathsf{rD}(x, y, \mathsf{rE}(y, \cdot))$ are $\mathbb{Z}_p$-valued linear functions.
(c)	Linear functions and $[\cdot]_s$ are commutative. (cf. Eq.(9))	Both $\mathsf{rD}(x, y, \mathsf{kE}(y, \cdot))$ and $\mathsf{rD}(x, y, \mathsf{rE}(y, \cdot))$ are $\mathbb{Z}_p$-valued linear functions. $\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}, \mathbf{s}^\top \mathbf{A}^\top \mathbf{W}_1 \mathbf{r}, \ldots, \mathbf{s}^\top \mathbf{A}^\top \mathbf{W}_n \mathbf{r} \in \mathbb{Z}_p$.
(d)	Restricted α-reconstruction.	$\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}, \mathbf{s}^\top \mathbf{A}^\top \mathbf{W}_1 \mathbf{r}, \ldots, \mathbf{s}^\top \mathbf{A}^\top \mathbf{W}_n \mathbf{r} \in \mathbb{Z}_p$.
(e)	Linear functions and $[\cdot]_s$ are commutative. (cf. Eq.(9))	$\mathsf{sD}(x, y, \mathsf{sE}(x, \cdot))$ is a $\mathbb{Z}_p$-valued linear function. $\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}, \mathbf{s}^\top \mathbf{A}^\top \mathbf{W}_1 \mathbf{r}, \ldots, \mathbf{s}^\top \mathbf{A}^\top \mathbf{W}_n \mathbf{r} \in \mathbb{Z}_p$.
(f)	Linear functions and pairings are commutative. (cf. Eq.(8))	$\mathsf{sD}(x, y, \mathsf{sE}(x, \cdot))$ is a $\mathbb{Z}_p$-valued linear function.
(g)	Linear functions and "exponentiations" are commutative. (cf. Eq.(10))	$\mathsf{sD}(x, y, \cdot)$ is a $\mathbb{Z}_p$-valued linear function.
(h)	Composition of two linear functions are linear.	Both $e(\cdot, [\mathbf{r}]_2)$ and $\mathsf{sD}(x, y, \cdot)$ are linear functions.

(semi-functional ciphertext) A semi-functional ciphertext for target attribute $x^* \in \mathcal{X}$ and challenge message pair $(m_0^*, m_1^*) \in \mathcal{M} \times \mathcal{M}$ is defined as follows:

$$\left\{ [\,\boxed{\mathbf{c}}\,]_1,\ \mathsf{sE}(x^*, [\,\mathbf{W}_1^\top \boxed{\mathbf{c}}\,]_1, \ldots, [\,\mathbf{W}_n^\top \boxed{\mathbf{c}}\,]_1) \cdot [\,\mathbf{W}^\top \boxed{\mathbf{c}}\,]_1^{\boldsymbol{\tau}^*}, [\,\mathbf{k}^\top \boxed{\mathbf{c}}\,]_T \cdot m_\beta^*,\ \boldsymbol{\tau}^* \right\}$$

where $\mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$ and $\boldsymbol{\tau}^* \leftarrow \mathsf{sE}(x^*)$.

(semi-functional secret key) A semi-functional secret key for policy $y \in \mathcal{Y}$ is defined as follows:

$$\left\{ [\mathbf{r}]_2, \mathsf{kE}(y, [\,\mathbf{k} + \boxed{\alpha \mathbf{a}^\perp}\,]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r}]_2, \ldots, [\mathbf{W}_n \mathbf{r}]_2), [\mathbf{W}\mathbf{r}]_2 \right\}$$

where $\alpha \leftarrow \mathbb{Z}_p$. Note that all semi-functional secret keys in the system will share the same α.

Game Sequence. Our proof employs the following game sequence.

- G_0 is the real security game defined as Section 2.1.
- G_1 is identical to G_0 except that the challenge ciphertext is semi-functional.
- $\mathsf{G}_{2.i}$ (for $i \in [0, q]$) is identical to G_1 except that the first i key extraction queries are replied with semi-functional secret keys.
- G_3 is identical to $\mathsf{G}_{2.q}$ except that the challenge ciphertext is a semi-functional ciphertext for random message $m^* \in \mathcal{M}$.

Roughly speaking, we are going to prove that

$$\mathsf{G}_0 \overset{\text{lem 4}}{\approx} \mathsf{G}_1 = \mathsf{G}_{2.0} \overset{\text{sec 3.3}}{\approx} \mathsf{G}_{2.1} \overset{\text{sec 3.3}}{\approx} \cdots \overset{\text{sec 3.3}}{\approx} \mathsf{G}_{2.q} \overset{\text{lem 5}}{=} \mathsf{G}_3.$$

Here "$\approx$" indicates that two games are *computationally* indistinguishable while "$=$" means that they are *statistically* indistinguishable. Let $\mathsf{Adv}^{i.j}_{\mathcal{A}}(\lambda)$ be the advantage function of any p.p.t. adversary $\mathcal{A}$ making at most q key extraction queries in $\mathsf{G}_{i.j}$ with security parameter λ.

We begin with two simple lemmas. First, it is not hard to see that G_1 and $\mathsf{G}_{2.0}$ are actually the same and we have the following lemma.

Lemma 2 ($\mathsf{G}_1 = \mathsf{G}_{2.0}$)**.** *For any adversary $\mathcal{A}$, we have* $\mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{2.0}_{\mathcal{A}}(\lambda)$.

Next, observe that the challenge ciphertext in the last game G_3 is created without secret bit $\beta \in \{0,1\}$. In other words, the challenge ciphertext reveals nothing about β. Therefore adversary has no advantage in guessing β and we have the lemma below.

Lemma 3. *For any adversary $\mathcal{A}$, we have* $\mathsf{Adv}^{3}_{\mathcal{A}}(\lambda) = 0$.

Following Chen *et al.*'s proof [CGW15], we can prove Lemma 4 showing $\mathsf{G}_0 \approx \mathsf{G}_1$ and Lemma 5 showing $\mathsf{G}_{2.q} = \mathsf{G}_3$. Due to the lack of space, we omit the proofs.

Lemma 4 ($\mathsf{G}_0 \approx \mathsf{G}_1$)**.** *For any p.p.t. adversary $\mathcal{A}$ making at most q key extraction queries, there exists an algorithm $\mathcal{B}$ such that*

$$\mid \mathsf{Adv}^{0}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) \mid \;\leq\; \mathsf{Adv}^{\mathcal{D}_k}_{\mathcal{B}}(\lambda) + 1/p$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q \cdot k^2 \cdot \mathrm{poly}(\lambda, n)$.

Lemma 5 ($\mathsf{G}_{2.q} = \mathsf{G}_3$)**.** *For any adversary $\mathcal{A}$, we have*

$$|\mathsf{Adv}^{2.q}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{3}_{\mathcal{A}}(\lambda)| = 1/p.$$

In order to complete the proof, we prove that $\mathsf{G}_{2.i}$ is indistinguishable with $\mathsf{G}_{2.i+1}$ for all $i \in [0, q-1]$. The details are deferred to the next subsection.

3.3 Proving the Theorem: Filling the gap between $\mathsf{G}_{2.i}$ and $\mathsf{G}_{2.i+1}$

This subsection proves the indistinguishability of $\mathsf{G}_{2.i}$ and $\mathsf{G}_{2.i+1}$. We introduce an auxiliary game sequence which is based on the proof idea from Jutla and Roy [JR13]. In particular, we need the following two auxiliary distributions.

(pesudo-normal secret key) A pseudo-normal secret key for policy $y \in \mathcal{Y}$ is defined as follows:

$$\left\{ \begin{array}{c} [\mathbf{r}]_2,\ \mathsf{kE}(y,[\mathbf{k}]_2) \cdot \mathsf{rE}(y,[\mathbf{W}_1\mathbf{r}]_2,\ldots,[\mathbf{W}_n\mathbf{r}]_2) \cdot \boxed{[\mathbf{a}^\perp \hat{r}]_2^{\mathsf{rE}(y,\gamma_1,\ldots,\gamma_n)}}, \\ [\mathbf{W}\mathbf{r}]_2 \cdot \boxed{[\mathbf{a}^\perp \hat{r}]_2^{-1}} \end{array} \right\}$$

where $\hat{r} \leftarrow \mathbb{Z}_p$ and $\gamma_1, \ldots, \gamma_n \leftarrow \mathbb{Z}_p$ are the random coins for tag $\boldsymbol{\tau}^*$ in the challenge ciphertext. Recall that we compute $\boldsymbol{\tau}^* = \mathsf{sE}(x^*, (\gamma_1, \ldots, \gamma_n))$ for target attribute x^*.

(pesudo-semi-functional secret key) A pseudo-semi-functional secret key for policy $y \in \mathcal{Y}$ is defined as follows:

$$\left\{ \begin{array}{c} [\mathbf{r}]_2,\ \mathsf{kE}(y, [\,\mathbf{k} + \boxed{\alpha \mathbf{a}^\perp}\,]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1\mathbf{r}]_2, \ldots, [\mathbf{W}_n\mathbf{r}]_2) \cdot [\mathbf{a}^\perp \hat{r}]_2^{\mathsf{rE}(y,\gamma_1,\ldots,\gamma_n)}, \\ [\mathbf{W}\mathbf{r}]_2 \cdot [\mathbf{a}^\perp \hat{r}]_2^{-1} \end{array} \right\}$$

where $\hat{r} \in \mathbb{Z}_p$ and $\gamma_1, \ldots, \gamma_n \in \mathbb{Z}_p$ are defined as before and $\alpha \in \mathbb{Z}_p$ is the one used in the semi-functional secret key.

We note that the random coins $\gamma_1, \ldots, \gamma_n$ for $\boldsymbol{\tau}^*$ are independent of x^* and thus we can pick them at the very beginning and use them to create secret keys of these two forms at *any* point.

Game Sub-sequence. For each $i \in [0, q-1]$, we define

- $\mathsf{G}_{2.i.1}$ is identical to $\mathsf{G}_{2.i}$ except that the $i+1$st key extraction query y is answered with a pseudo-normal secret key.
- $\mathsf{G}_{2.i.2}$ is identical to $\mathsf{G}_{2.i.1}$ except that the $i+1$st key extraction query y is answered with a pseudo-semi-functional secret key.

With this sub-sequence, we will prove that

$$\mathsf{G}_{2.i} \overset{\text{lem } 6}{\approx} \mathsf{G}_{2.i.1} \overset{\text{lem } 8}{=} \mathsf{G}_{2.i.2} \overset{\text{lem } 7}{\approx} \mathsf{G}_{2.i+1}.$$

We first prove Lemma 6 showing that $\mathsf{G}_{2.i} \approx \mathsf{G}_{2.i.1}$ and note that, following almost the same strategy, we can also prove that $\mathsf{G}_{2.i.2} \approx \mathsf{G}_{2.i+1}$. Hence we will show the corresponding result in Lemma 7 but omit the proof.

Lemma 6 ($\mathsf{G}_{2.i} \approx \mathsf{G}_{2.i.1}$). *For any p.p.t. adversary $\mathcal{A}$ making at most q key extraction queries, there exists an algorithm $\mathcal{B}$ such that*

$$|\ \mathsf{Adv}^{2.i}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{2.i.1}_{\mathcal{A}}(\lambda)\ | \ \leq\ \mathsf{Adv}^{\mathcal{D}_k}_{\mathcal{B}}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q \cdot k^2 \cdot \mathrm{poly}(\lambda, n)$.

Proof. Given $(\,\mathcal{G}, [\mathbf{M}]_2, [\mathbf{t}]_2 = [\mathbf{M}\mathbf{u} + \mathbf{e}v]_2\,)$ where $\mathbf{u} \leftarrow \mathbb{Z}_p^k$, $\mathbf{e} = (0, \ldots, 0, 1)^\top \in \mathbb{Z}_p^{k+1}$ and either $v \leftarrow \mathbb{Z}_p$ or $v = 0$, algorithm $\mathcal{B}$ works as follows:

Initialize. Sample $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, $\quad \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$ and $\beta \leftarrow \{0,1\}$. Pick

$$\widetilde{\mathbf{W}}_1, \cdots, \widetilde{\mathbf{W}}_n, \widetilde{\mathbf{W}} \leftarrow \mathbb{Z}_p^{(k+1)\times k} \quad \text{and} \quad \gamma_1, \ldots, \gamma_n \leftarrow \mathbb{Z}_p$$

and program "hidden parameter" $\gamma_1, \ldots, \gamma_n$ into $\mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W}$ as follows

$$\mathbf{W}_1 = \widetilde{\mathbf{W}}_1 + \gamma_1 \mathbf{V},\ \ldots,\ \mathbf{W}_n = \widetilde{\mathbf{W}}_n + \gamma_n \mathbf{V}, \quad \mathbf{W} = \widetilde{\mathbf{W}} - \mathbf{V}$$

where $\mathbf{V} = \mathbf{a}^\perp \cdot (\underline{\mathbf{M}}\overline{\mathbf{M}}^{-1}) \in \mathbb{Z}_p^{(k+1)\times k}$. One may check that all $\mathbf{W}_i$ and $\mathbf{W}$ are uniformly distributed over $\mathbb{Z}_p^{(k+1)\times k}$ as required. Due to the fact that $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$, we can return the master public key as follows

$$\mathsf{mpk} = \{\ [\mathbf{A}]_1, [\widetilde{\mathbf{W}}_1^\top \mathbf{A}]_1, \ldots, [\widetilde{\mathbf{W}}_n^\top \mathbf{A}]_1, [\widetilde{\mathbf{W}}^\top \mathbf{A}]_1, [\mathbf{k}^\top \mathbf{A}]_T\ \}.$$

We note that $\mathcal{B}$ can not compute $\mathbf{V}$ and thus does not know $\mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W}$.

Challenge ciphertext. For target attribute x^* and challenge message pair (m_0^*, m_1^*), we sample $\mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$, compute tag $\boldsymbol{\tau}^* = \mathsf{sE}(x^*, (\gamma_1, \ldots, \gamma_n))$ using the "hidden parameter" and create the challenge ciphertext as follows

$$\{\, [\mathbf{c}]_1, \mathsf{sE}(x^*, [\widetilde{\mathbf{W}}_1^\top \mathbf{c}]_1, \ldots, [\widetilde{\mathbf{W}}_n^\top \mathbf{c}]_1) \cdot [\widetilde{\mathbf{W}}^\top \mathbf{c}]_1^{\boldsymbol{\tau}^*}, [\mathbf{k}^\top \mathbf{c}]_T \cdot m_\beta^*,\ \boldsymbol{\tau}^* \,\}$$

Let $\mathbf{v} = \underline{\mathbf{M}}\overline{\mathbf{M}}^{-1}$, we show that

$$\begin{aligned} & \mathsf{sE}(x^*, [\mathbf{W}_1^\top \mathbf{c}]_1, \ldots, [\mathbf{W}_n^\top \mathbf{c}]_1) \cdot [\mathbf{W}^\top \mathbf{c}]_1^{\boldsymbol{\tau}^*} \\ = {} & \mathsf{sE}(x^*, [\widetilde{\mathbf{W}}_1^\top \mathbf{c}]_1, \ldots, [\widetilde{\mathbf{W}}_n^\top \mathbf{c}]_1) \cdot [\widetilde{\mathbf{W}}^\top \mathbf{c}]_1^{\boldsymbol{\tau}^*} \cdot \\ & \boxed{\mathsf{sE}(x^*, [\gamma_1 \mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1, \ldots, [\gamma_n \mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1) \cdot [-\mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1^{\boldsymbol{\tau}^*}} \end{aligned}$$

from the linearity of $\mathsf{sE}(x^*, \cdot)$ and

$$\begin{aligned} & \mathsf{sE}(x^*, [\gamma_1 \mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1, \ldots, [\gamma_n \mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1) \cdot [-\mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1^{\boldsymbol{\tau}^*} \quad \text{(the boxed part)} \\ = {} & \mathsf{sE}(x^*, [\mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1^{(\gamma_1, \ldots, \gamma_n)}) \cdot [-\mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1^{\boldsymbol{\tau}^*} \\ = {} & [\mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1^{\mathsf{sE}(x^*, (\gamma_1, \ldots, \gamma_n))} \cdot [\mathbf{v}^\top \mathbf{a}^{\perp\top} \mathbf{c}]_1^{-\boldsymbol{\tau}^*} = \mathbf{0}, \end{aligned}$$

where the first equality is mainly implied by Eq.(3) and Eq.(4), and the second equality comes from the fact shown in Eq.(10). This is sufficient to see that our simulation is perfect.

Key extraction. We consider three cases: (1) For the first i queries y, we sample $\mathbf{r}' \leftarrow \mathbb{Z}_p^k$ and implicitly set

$$\mathbf{r} = \overline{\mathbf{M}}\mathbf{r}' \in \mathbb{Z}_p^k.$$

The vector $\mathbf{r}$ here is uniformly distributed as required and we can compute $[\mathbf{r}]_2$ and simulate

$$[\mathbf{V}\mathbf{r}]_2 = [\mathbf{a}^\perp \cdot (\underline{\mathbf{M}}\mathbf{r}')]_2.$$

These suffice for creating the secret key as

$$\left\{[\mathbf{r}]_2, \mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\widetilde{\mathbf{W}}_1 \mathbf{r}]_2 \cdot [\gamma_1 \mathbf{V}\mathbf{r}]_2, \ldots, [\widetilde{\mathbf{W}}_n \mathbf{r}]_2 \cdot [\gamma_n \mathbf{V}\mathbf{r}]_2), [\widetilde{\mathbf{W}}\mathbf{r}]_2 \cdot [-\mathbf{V}\mathbf{r}]_2\right\}$$

because $\mathbf{k}$, $\widetilde{\mathbf{W}}_1, \ldots, \widetilde{\mathbf{W}}_n, \widetilde{\mathbf{W}}$ and $\gamma_1, \ldots, \gamma_n$ are all known to $\mathcal{B}$. (2) For the $i+1$st query y, we implicitly set

$$\mathbf{r} = \overline{\mathbf{M}}\mathbf{u} = \bar{\mathbf{t}} \in \mathbb{Z}_p^k.$$

The vector $\mathbf{r}$ is distributed properly and $[\mathbf{r}]_2 = [\,\bar{\mathbf{t}}\,]_2$ can be simulated. Then we may produce the secret key as follows

$$\left\{ \begin{array}{c} [\,\bar{\mathbf{t}}\,]_2,\ \mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\widetilde{\mathbf{W}}_1 \bar{\mathbf{t}}\,]_2 \cdot [\,\gamma_1 \mathbf{a}^\perp \underline{\mathbf{t}}\,]_2, \ldots, [\,\widetilde{\mathbf{W}}_n \bar{\mathbf{t}}\,]_2 \cdot [\,\gamma_n \mathbf{a}^\perp \underline{\mathbf{t}}\,]_2), \\ [\,\widetilde{\mathbf{W}}\bar{\mathbf{t}}\,]_2 \cdot [\,-\mathbf{a}^\perp \underline{\mathbf{t}}\,]_2 \end{array} \right\}.$$

(3) For the remaining $q - i - 1$ queries, we may work just as in the first case except that we employ $\mathbf{k} + \alpha \mathbf{a}^\perp$ in the place of $\mathbf{k}$.

Finalize. Output 1 when $\beta = \beta'$ and 0 otherwise.

Observe that, in the reply to the $i+1$st key extraction query, we have

$$\mathbf{a}^\perp \underline{\mathbf{t}} = \mathbf{a}^\perp(\underline{\mathbf{M}}\mathbf{u} + v) = \mathbf{a}^\perp(\underline{\mathbf{M}}\overline{\mathbf{M}}^{-1})(\overline{\mathbf{M}}\mathbf{u}) + \mathbf{a}^\perp v = \mathbf{V}\bar{\mathbf{t}} + \mathbf{a}^\perp v.$$

Therefore we have

$$\begin{aligned}
&\mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\widetilde{\mathbf{W}}_1 \bar{\mathbf{t}}]_2 \cdot [\gamma_1 \mathbf{a}^\perp \underline{\mathbf{t}}]_2, \ldots, [\widetilde{\mathbf{W}}_n \bar{\mathbf{t}}]_2 \cdot [\gamma_n \mathbf{a}^\perp \underline{\mathbf{t}}]_2) \\
&= \mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r}]_2 \cdot [\gamma_1 \mathbf{a}^\perp v]_2, \ldots, [\mathbf{W}_n \mathbf{r}]_2 \cdot [\gamma_n \mathbf{a}^\perp v]_2) \\
&= \mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r}]_2, \ldots, [\mathbf{W}_n \mathbf{r}]_2) \cdot \mathsf{rE}(y, [\gamma_1 \mathbf{a}^\perp v]_2, \ldots, [\gamma_n \mathbf{a}^\perp v]_2) \\
&= \mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r}]_2, \ldots, [\mathbf{W}_n \mathbf{r}]_2) \cdot \mathsf{rE}(y, [\mathbf{a}^\perp v]_2^{(\gamma_1, \ldots, \gamma_n)}) \\
&= \mathsf{kE}(y, [\mathbf{k}]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r}]_2, \ldots, [\mathbf{W}_n \mathbf{r}]_2) \cdot [\mathbf{a}^\perp v]_2^{\mathsf{sE}(y, \gamma_1, \ldots, \gamma_n)} \\
&\quad [\widetilde{\mathbf{W}} \bar{\mathbf{t}}]_2 \cdot [-\mathbf{a}^\perp \underline{\mathbf{t}}]_2 \\
&= [\mathbf{W}\mathbf{r}]_2 \cdot [-\mathbf{a}^\perp v]_2 \\
&= [\mathbf{W}\mathbf{r}]_2 \cdot [\mathbf{a}^\perp v]_2^{-1}
\end{aligned}$$

It is now clear that the simulation is identical to $\mathsf{G}_{2.i}$ when $v = 0$; and if v is a random element in $\mathbb{Z}_p$, the simulation is identical to $\mathsf{G}_{2.i.1}$ where $\hat{r} = v$. □

Lemma 7 ($\mathsf{G}_{2.i.2} \approx \mathsf{G}_{2.i+1}$). *For any p.p.t. adversary $\mathcal{A}$ making at most q key extraction queries, there exists an algorithm $\mathcal{B}$ such that*

$$\mid \mathsf{Adv}_{\mathcal{A}}^{2.i.2}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{2.i+1}(\lambda) \mid \;\leq\; \mathsf{Adv}_{\mathcal{B}}^{\mathcal{D}_k}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q \cdot k^2 \cdot \mathrm{poly}(\lambda, n)$.

Proof. The proof is similar to that for Lemma 6. □

We complete the proof by proving Lemma 8 which states that $\mathsf{G}_{2.i.1}$ and $\mathsf{G}_{2.i.2}$ are statistically indistinguishable. This is derived from the α-privacy of predicate encodings.

Lemma 8 ($\mathsf{G}_{2.i.1} = \mathsf{G}_{2.i.2}$). *For any adversary $\mathcal{A}$, we have*

$$\mathsf{Adv}_{\mathcal{A}}^{2.i.1}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{2.i.2}(\lambda).$$

Proof. We prove the lemma for any fixed

- $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$, $\mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times k}$, $\mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$, $\beta \leftarrow \{0,1\}$;
- random coin $\mathbf{c} \leftarrow \mathbb{Z}_p^{k+1}$ for semi-functional challenge ciphertext ct^*;
- $\alpha \leftarrow \mathbb{Z}_p$, random coin $\mathbf{r} \leftarrow \mathbb{Z}_p^k$ for each key extraction query and the extra random coin $\hat{r} \leftarrow \mathbb{Z}_p$ for the $i+1$st one.

Let $\mathsf{sk}_y^{(b)}$ be the reply to the $i+1$st key extraction query y in $\mathsf{G}_{2.i.b}$ $(b = 1, 2)$ and ct^* is the challenge ciphertext for target attribute x^*. It is sufficient to show

$$\{\ (\mathsf{sk}_y^{(1)}, \mathsf{ct}^*)\ \} = \{\ (\mathsf{sk}_y^{(2)}, \mathsf{ct}^*)\ \}$$

where the probability space is defined by $\gamma_1, \ldots, \gamma_n \leftarrow \mathbb{Z}_p$. In fact, we can further reduce to the claim that

$$\{\ \mathsf{kE}(y, [\alpha \mathbf{a}^\perp]_2) \cdot [\mathbf{a}^\perp \hat{r}]_2^{\mathsf{rE}(y,\gamma_1,\ldots,\gamma_n)},\quad \mathsf{sE}(x^*, \gamma_1, \ldots, \gamma_n)\ \} \text{ and}$$
$$\{\ [\mathbf{a}^\perp \hat{r}]_2^{\mathsf{rE}(y,\gamma_1,\ldots,\gamma_n)},\quad \mathsf{sE}(x^*, \gamma_1, \ldots, \gamma_n)\ \}$$

are statistically close over the same probability space as before. One can rewrite the first distribution as

$$\{\ [\mathbf{a}^\perp \hat{r}]_2^{\mathsf{kE}(y,\alpha/\hat{r})+\mathsf{rE}(y,\gamma_1,\ldots,\gamma_n)},\quad \mathsf{sE}(x^*, \gamma_1, \ldots, \gamma_n)\ \}.$$

which is statistically close to the second one by the α-privacy of predicate encoding. This readily proves the lemma. □

4 Tag-based ABE with Weak Attribute-hiding Property

This section shows that our framework (presented in Section 3) is compatible with the *attribute-hiding* predicate encoding [CGW15]. This means that our framework can derive a series of attribute-hiding ABE with tag.

4.1 Preliminaries

Definition. We may call an ABE with (weak) attribute-hiding the *predicate encryption* (PE for short). A PE scheme is also defined by four p.p.t algorithms $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$ as in Section 2, but the security is defined in a slightly different way. For all adversary $\mathcal{A}$, define the advantage function $\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda)$ as

$$\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[\beta = \beta' \;\middle|\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{P}),\ \beta \leftarrow \{0,1\} \\ (x_0^*, x_1^*, m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{mpk}) \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_\beta^*, m_\beta^*) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{mpk}, \mathsf{ct}^*) \end{array} \right] - \frac{1}{2} \right|.$$

A PE scheme is said to be *adaptively secure* and *weakly attribute-hiding* if $\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda)$ is negligible in λ and $\mathsf{P}(x_0^*, y) = \mathsf{P}(x_1^*, y) = 0$ holds for each query y sent to oracle $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \cdot)$ for all p.p.t. adversary $\mathcal{A}$.

Attribute-hiding Predicate Encoding. A $\mathbb{Z}_p$-linear predicate encoding $(\mathsf{sE}, \mathsf{rE}, \mathsf{kE}, \mathsf{sD}, \mathsf{rD})$ for $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is *attribute-hiding* [CGW15] if it has the following two additional properties:

(x-oblivious α-reconstruction) $\mathsf{sD}(x, y, \cdot)$ and $\mathsf{rD}(x, y, \cdot)$ are independent of x.

(attribute-hiding) For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 0$, the following distributions are identical.

$$\{\ x, y, \mathsf{sE}(x, \mathbf{w}), \mathsf{rE}(y, \mathbf{w}) : \mathbf{w} \leftarrow \mathbb{Z}_p^n\ \} \quad \text{and} \quad \{\ x, y, \mathbf{r} : \mathbf{r} \leftarrow \mathbb{Z}_p^{|\mathsf{sE}|+|\mathsf{rE}|}\ \}.$$

4.2 Construction and Security Analysis

Assuming an attribute-hiding predicate encoding, we can construct a predicate encryption with tag as in Section 3.1. Technically we prove the following theorem stating that the generic tag-based PE scheme is adaptively secure and weakly attribute-hiding under standard assumption in the standard model.

Theorem 2 (Weak AH). *For any p.p.t. adversary $\mathcal{A}$ making at most q key extraction queries, there exists algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that*

$$\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathcal{D}_k}_{\mathcal{B}_1}(\lambda) + q \cdot \mathsf{Adv}^{\mathcal{D}_k}_{\mathcal{B}_2}(\lambda) + q \cdot \mathsf{Adv}^{\mathcal{D}_k}_{\mathcal{B}_3}(\lambda) + 2^{-\Omega(\lambda)}$$

and $\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot k^2 \cdot \mathrm{poly}(\lambda, n)$.

Proof Overview. We prove the theorem using almost the same game sequence as described in Section 3.2 and Section 3.3. We just describe the differences between them. Firstly, the challenge ciphertext will be generated for identity x^*_β. Secondly, we need to re-define *pseudo-semi-functional secret keys* and *semi-functional secret keys* as follows.

(pesudo-semi-functional secret key) A pseudo-semi-functional secret key for policy $y \in \mathcal{Y}$ is defined as follows:

$$\left\{ \begin{array}{l} [\mathbf{r}]_2, \\ \mathsf{kE}(y, [\mathbf{k} + \boxed{\alpha \mathbf{a}^\perp}]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r} + \boxed{\mathbf{a}^\perp \hat{u}_1}]_2, \ldots, [\mathbf{W}_n \mathbf{r} + \boxed{\mathbf{a}^\perp \hat{u}_n}]_2) \\ \qquad\qquad \cdot [\mathbf{a}^\perp \hat{r}]_2^{\mathsf{rE}(y, \gamma_1, \ldots, \gamma_n)}, \\ [\mathbf{W}\mathbf{r}]_2 \cdot [\mathbf{a}^\perp \hat{r}]_2^{-1} \end{array} \right\}$$

where $\hat{r} \in \mathbb{Z}_p$, $\gamma_1, \ldots, \gamma_n \in \mathbb{Z}_p$, $\alpha \in \mathbb{Z}_p$ are defined as before and $\hat{u}_1, \ldots, \hat{u}_n \leftarrow \mathbb{Z}_p$ are fresh for each pseudo-semi-functional secret key.

(semi-functional secret key) A semi-functional secret key for policy $y \in \mathcal{Y}$ is defined as follows:

$$\left\{ [\mathbf{r}]_2, \mathsf{kE}(y, [\mathbf{k} + \boxed{\alpha \mathbf{a}^\perp}]_2) \cdot \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{r} + \boxed{\mathbf{a}^\perp \hat{u}_1}]_2, \ldots, [\mathbf{W}_n \mathbf{r} + \boxed{\mathbf{a}^\perp \hat{u}_n}]_2), [\mathbf{W}\mathbf{r}]_2 \right\}$$

where $\alpha \in \mathbb{Z}_p$ are defined as before and $\hat{u}_1, \ldots, \hat{u}_n \leftarrow \mathbb{Z}_p$ are fresh for each semi-functional secret key.

Finally, we add an additional game G_4 shown below. Its preceding game G_3 (the final game in the previous game sequence) is restated so as to emphasize the difference between them.

- G_3 is identical to $\mathsf{G}_{2.q}$ except that the challenge ciphertext is a semi-functional ciphertext for attribute x^*_β and $\boxed{\text{random message } m^* \in \mathcal{M}}$.
- G_4 is identical to G_3 except that the challenge ciphertext is a semi-functional ciphertext for $\boxed{\text{random attribute } x^* \in \mathcal{X}}$ and random message $m^* \in \mathcal{M}$.

With the extended game sequence, we will prove that

$$\mathsf{G}_0 \approx \mathsf{G}_1 = \mathsf{G}_{2.0} \approx \mathsf{G}_{2.1} \approx \cdots \approx \mathsf{G}_{2.q} = \boxed{\mathsf{G}_3 = \mathsf{G}_4}.$$

where "$\mathsf{G}_{2.i} \approx \mathsf{G}_{2.i+1}$" for all $i \in [0, q]$ will be proved using the game sub-sequence

$$\mathsf{G}_{2.i} \approx \boxed{\mathsf{G}_{2.i.1} = \mathsf{G}_{2.i.2}} \approx \mathsf{G}_{2.i+1}.$$

Because of the similarity of game sequences, most lemmas we have presented in Section 3.2 and Section 3.3 still hold and can be proved in the same way. Due to the lack of space, we omit them. The proofs of "$\mathsf{G}_3 = \mathsf{G}_4$" and "$\mathsf{G}_{2.i.1} = \mathsf{G}_{2.i.2}$" (the boxed parts) mainly follow [CGW15] and our proof in previous section. Finally, we point out the fact: The challenge ciphertext in G_3 still leaks information of $\beta \in \{0, 1\}$ via x^*_β, but such a dependence is removed in G_4. Therefore $\mathsf{Adv}^4_{\mathcal{A}}(\lambda) = 0$ for any $\mathcal{A}$.

5 New Tag-based ABE

In this section we will exhibit two tag-based ABE for boolean span program derived from our generic tag-based ABE and two concrete encodings in [CGW15].

Boolean Span Program. Assume $n \in \mathbb{N}$. Let $[n]$ be the *attribute universe*. A span program over $[n]$ is defined by $(\mathbf{M}, \rho)$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell'}$ and $\rho : [\ell] \to [n]$. We use $\mathbf{M}_i$ to denote the ith row of $\mathbf{M}$. For an input $\mathbf{x} = (x_1, \ldots, x_n) \in \{0, 1\}^n$, we say

$$\mathbf{x} \text{ satisfies } (\mathbf{M}, \rho) \iff \mathbf{1} \in \mathsf{span}(\mathbf{M}_{\mathbf{x}})$$

where $\mathbf{1} = (1, 0, \ldots, 0) \in \mathbb{Z}_p^{1 \times \ell'}$ and $\mathbf{M}_{\mathbf{x}} = \{\, \mathbf{M}_j : x_{\rho(j)} = 1 \,\}$. In this case one can efficiently find coefficients $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that

$$\textstyle\sum_{j \,:\, x_{\rho(j)}=1} \omega_j \mathbf{M}_j = \mathbf{1}.$$

Here we will assume $n = \ell$ and ρ is an identity map following [CGW15].

5.1 Key-policy Construction

The corresponding predicate is defined as

$$\mathsf{P}(\mathbf{x}, \mathbf{M}) = 1 \iff \mathbf{x} \text{ satisfies } \mathbf{M}$$

where $\mathbf{x} \in \mathcal{X} = \{0, 1\}^\ell$ and $\mathbf{M} \in \mathcal{Y} = \mathbb{Z}_p^{\ell \times \ell'}$. Our concrete KP-ABE scheme is described below:

– $\mathsf{Setup}(1^\lambda, 1^\ell)$: Sample

$$\mathbf{A} \leftarrow \mathcal{D}_k, \quad \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{U}_2, \ldots, \mathbf{U}_{\ell'}, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \quad \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$$

and output

$$\begin{aligned} \mathsf{mpk} &:= \{\, [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \cdots, [\mathbf{W}_\ell^\top \mathbf{A}]_1, [\mathbf{W}^\top \mathbf{A}]_1, [\mathbf{k}^\top \mathbf{A}]_T \,\} \\ \mathsf{msk} &:= \{\, \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{U}_2, \ldots, \mathbf{U}_{\ell'}, \mathbf{W};\ \mathbf{k} \,\} \end{aligned}$$

– $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, m)$: On input $\mathbf{x} \in \{0,1\}^\ell$ and $m \in G_T$, pick $\mathbf{s} \leftarrow \mathbb{Z}_p^k$ and $w_1, \ldots, w_\ell \leftarrow \mathbb{Z}_p$. Output

$$\mathsf{ct}_{\mathbf{x}} := \left\{ \begin{array}{l} C_0 := [\mathbf{A}\mathbf{s}]_1, \\ C_1 := [x_1(\mathbf{W}_1 + w_1\mathbf{W})^\top \mathbf{A}\mathbf{s}]_1, \\ \quad \vdots \\ C_\ell := [x_\ell(\mathbf{W}_\ell + w_\ell\mathbf{W})^\top \mathbf{A}\mathbf{s}]_1, \\ C := [\mathbf{k}^\top \mathbf{A}\mathbf{s}]_T \cdot m, \\ \boldsymbol{\tau} := (\tau_1 := x_1 w_1, \ldots, \tau_\ell := x_\ell w_\ell) \end{array} \right\} \in G_1^{k+1} \times (G_1^k)^\ell \times G_T \times \mathbb{Z}_p^\ell$$

– $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{M})$: On input $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell'}$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{sk}_{\mathbf{M}} := \left\{ \begin{array}{l} K_0 := [\mathbf{r}]_2, \\ K_1 := [(\mathbf{k} || \mathbf{U}_2\mathbf{r} || \cdots || \mathbf{U}_{\ell'}\mathbf{r})\mathbf{M}_1^\top + \mathbf{W}_1\mathbf{r}]_2, \\ \quad \vdots \\ K_\ell := [(\mathbf{k} || \mathbf{U}_2\mathbf{r} || \cdots || \mathbf{U}_{\ell'}\mathbf{r})\mathbf{M}_\ell^\top + \mathbf{W}_\ell\mathbf{r}]_2, \\ K_t := [\mathbf{W}\mathbf{r}]_2 \end{array} \right\} \in G_2^k \times (G_2^{k+1})^\ell \times G_2^{k+1}$$

– $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{M}}, \mathsf{ct}_{\mathbf{x}})$: Find out $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that $\sum_{j:x_j=1} \omega_j \mathbf{M}_j = \mathbf{1}$ and compute

$$K \leftarrow e(C_0, \prod_{j:x_j=1} (K_j \cdot K_t^{\tau_j})^{\omega_j}) \cdot e(\prod_{j:x_j=1} C_j^{-\omega_j}, K_0)$$

Recover the message as $m \leftarrow C/K \in G_T$.

5.2 Ciphertext-policy Construction

The corresponding predicate is defined as

$$\mathsf{P}(\mathbf{M}, \mathbf{x}) = 1 \quad \Longleftrightarrow \quad \mathbf{x} \text{ satisfies } \mathbf{M}$$

where $\mathbf{M} \in \mathcal{X} = \mathbb{Z}_p^{\ell \times \ell'}$ and $\mathbf{x} \in \mathcal{Y} = \{0,1\}^\ell$. Our concrete CP-ABE scheme is described below:

– $\mathsf{Setup}(1^\lambda, 1^\ell)$: Sample

$$\mathbf{A} \leftarrow \mathcal{D}_k, \quad \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{V}, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \quad \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$$

and output

$$\begin{aligned} \mathsf{mpk} &:= \{\ [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_\ell^\top \mathbf{A}]_1, [\mathbf{V}^\top \mathbf{A}]_1, [\mathbf{W}^\top \mathbf{A}]_1, [\mathbf{k}^\top \mathbf{A}]_T\ \} \\ \mathsf{msk} &:= \{\ \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{V}, \mathbf{W};\ \mathbf{k}\ \} \end{aligned}$$

– $\mathsf{Enc}(\mathsf{mpk}, \mathbf{M}, m)$: On input $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell'}$ and $m \in G_T$, pick $\mathbf{s} \leftarrow \mathbb{Z}_p^k$ and

$$w_1, \ldots, w_\ell \leftarrow \mathbb{Z}_p,\ v \leftarrow \mathbb{Z}_p,\ \mathbf{u} \leftarrow \mathbb{Z}_p^{\ell'-1},\ \mathbf{U}_2, \ldots, \mathbf{U}_{\ell'} \leftarrow \mathbb{Z}_p^{(k+1)\times k}.$$

Output

$$\mathsf{ct}_{\mathbf{M}} := \left\{ \begin{array}{l} C_0 := [\mathbf{As}]_1, \\ C_1 := [(\mathbf{W}_1 + w_1\mathbf{W})^\top \mathbf{As} + ((\mathbf{V} + v\mathbf{W})^\top \mathbf{As} || \mathbf{U}_2^\top \mathbf{As} || \cdots || \mathbf{U}_{\ell'}^\top \mathbf{As}) \mathbf{M}_1^\top]_1, \\ \quad \vdots \\ C_\ell := [(\mathbf{W}_\ell + w_\ell\mathbf{W})^\top \mathbf{As} + ((\mathbf{V} + v\mathbf{W})^\top \mathbf{As} || \mathbf{U}_2^\top \mathbf{As} || \cdots || \mathbf{U}_{\ell'}^\top \mathbf{As}) \mathbf{M}_\ell^\top]_1, \\ C := [\mathbf{k}^\top \mathbf{As}]_T \cdot m, \\ \boldsymbol{\tau} := (\ \tau_1 := w_1 + \mathbf{M}_1 \binom{v}{\mathbf{u}},\ \ldots,\ \tau_\ell := w_\ell + \mathbf{M}_\ell \binom{v}{\mathbf{u}}\ ) \end{array} \right\} \in G_1^{k+1} \times (G_1^k)^\ell \times G_T \times \mathbb{Z}_p^\ell$$

– $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x})$: On input $\mathbf{x} \in \{0,1\}^\ell$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{sk}_{\mathbf{x}} := \left\{ \begin{array}{r} K_0 := [\mathbf{r}]_2, \\ K_1 := [x_1\mathbf{W}_1\mathbf{r}]_2 \\ \vdots \\ K_\ell := [x_\ell\mathbf{W}_\ell\mathbf{r}]_2 \\ K_{\ell+1} := [\mathbf{k} + \mathbf{Vr}]_2 \\ K_t := [\mathbf{Wr}]_2 \end{array} \right\} \in G_2^k \times (G_2^{k+1})^{\ell+1} \times G_2^{k+1}$$

– $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x}}, \mathsf{ct}_{\mathbf{M}})$: Find out $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that $\sum_{j:x_j=1} \omega_j \mathbf{M}_j = \mathbf{1}$ and compute

$$K \leftarrow e(C_0, K_{\ell+1} \cdot \prod_{j:x_j=1} K_j^{\omega_j} \cdot K_t^{\sum_{j:x_j=1} \omega_j \tau_j}) \cdot e(\prod_{j:x_j=1} C_j^{-\omega_j}, K_0)$$

Recover the message as $m \leftarrow C/K \in G_T$.

Using concrete encodings in [CGW15], we can derive more tag-based ABE instantiations. With our framework, we can also reproduce several previous concrete tag-based ABE schemes (such as those in [Ram16,RS16]) with simple proofs under the k-Lin assumption. In fact we just need to extract respective concrete encodings from them and apply our framework.

6 How to Support Predicate Encoding with Delegation

Our framework shown in Section 3 and extended in Section 4 cannot cover the HIBE schemes proposed by Ramanna and Sarkar [RS14]. This section further develops the framework and predicate encoding to accommodate the delegation mechanism in HIBE system.

We first recall a notion from [CGW15]. A predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is *delegatable* if there exists a partial ordering $\leq$ on $\mathcal{Y}$ such that

$$(y \leq y') \wedge \mathsf{P}(x,y) = 1 \Longrightarrow \mathsf{P}(x,y') = 1 \quad \text{for all } x \in \mathcal{X}.$$

One of classical delegatable predicates is the predicate for HIBE: Let $\ell \in \mathbb{N}$ and $\mathcal{X} = \mathcal{Y} = \mathbb{Z}_p^{\leq \ell}$. The predicate is

$$\mathsf{P}(\mathbf{x}, \mathbf{y}) = 1 \quad \iff \quad \mathbf{y} \text{ is a prefix of } \mathbf{x}.$$

The partial ordering is the prefix relation, that is $\mathbf{y} \leq \mathbf{y}'$ iff $\mathbf{y}'$ is a prefix of $\mathbf{y}$.

6.1 Syntax and Definition

An ABE scheme for a delegatable predicate P consists of algorithms Setup, KeyGen, Enc, Dec as defined in Section 2.1 and a delegation algorithm Del.

– $\mathsf{Del}(\mathsf{mpk}, \mathsf{sk}_{y'}, y) \to \mathsf{sk}_y$. The *delegation* algorithm takes as input the master public key mpk, a secret key $\mathsf{sk}_{y'}$ for $y' \in \mathcal{Y}$ and index (policy) $y \in \mathcal{Y}$ with $y \leq y'$, and generates a secret key sk_y for y.

We require that the delegation algorithm is *path-oblivious* which means secret keys generated by KeyGen and Del have the same distribution, that is

$$\begin{aligned}&\{(\mathsf{sk}_{y'}, \mathsf{sk}_y) : \mathsf{sk}_{y'} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y'), \mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)\} \\ =&\{(\mathsf{sk}_{y'}, \mathsf{sk}_y) : \mathsf{sk}_{y'} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y'), \mathsf{sk}_y \leftarrow \mathsf{Del}(\mathsf{mpk}, \mathsf{sk}_{y'}, y)\}\end{aligned}$$

for all $y, y' \in \mathcal{Y}$ satisfying $y \leq y'$. The assumption is natural and allows us to continue working with the security model described in Section 2.1; otherwise one should turn to the model described in [SW08] where an adversary can decide how to create secret keys—using KeyGen or Del.

6.2 Predicate Encoding Supporting Delegation

A predicate encoding for delegatable predicate P is composed of five algorithms sE, sD, rE, kE, rD satisfying all requirements described in Section 2.3 and an extra algorithm

$$\mathsf{dE} : \mathcal{Y} \times \mathcal{Y} \times \mathbb{Z}_p^{n_r} \to \mathbb{Z}_p^{n'_r}$$

with the following features: (1) for all $\mathbf{w} \leftarrow \mathbb{Z}_p^n$, $\alpha \leftarrow \mathbb{Z}_p$ and $y, y' \in \mathcal{Y}$ with $y \leq y'$, it holds that

$$\mathsf{dE}(y, y', \mathsf{kE}(y', \alpha) + \mathsf{rE}(y', \mathbf{w})) = \mathsf{kE}(y, \alpha) + \mathsf{rE}(y, \mathbf{w})$$

and (2) $\mathsf{dE}(y, y', \cdot)$ is $\mathbb{Z}_p$-linear. A predicate encoding for HIBE from Boneh-Boyen-Goh's HIBE with constant-size ciphertext [BBG05] is as follows:

(encoding for HIBE [BBG05]) Let $\mathbf{w} \leftarrow \mathbb{Z}_p^{1\times(\ell+1)}$, $\mathbf{x} = (x_1, \ldots, x_{\ell_x})$ and $\mathbf{y} = (y_1, \ldots, y_{\ell_y})$ for $\ell_x, \ell_y \leq \ell$. Define

$$\begin{aligned}&\mathsf{sE}(\mathbf{x}, \mathbf{w}) := \mathbf{w}\,(1, \mathbf{x}, \mathbf{0})^\top && \mathsf{sD}(\mathbf{x}, \mathbf{y}, c) := c \\ &\mathsf{rE}(\mathbf{y}, \mathbf{w}) := \mathbf{w} \begin{pmatrix} 1 & \mathbf{y} \\ & \mathbf{I} \end{pmatrix}^\top && \mathsf{rD}(\mathbf{x}, \mathbf{y}, \mathbf{k}) := \mathbf{k}\,(1, x_{\ell_y+1}, \ldots, x_{\ell_x}, \mathbf{0})^\top \\ &\mathsf{kE}(\mathbf{y}, \alpha) := (\alpha, \mathbf{0})\end{aligned}$$

As shown in [KSGA16], the encoding is linear and satisfies α-restriction and α-privacy. Besides that, for $\mathbf{y}' = (y_1, \ldots, y_{\ell'_y})$ and $\mathbf{y} = (y_1, \ldots, y_{\ell_y})$ with $\ell'_y \leq \ell_y$, we also define

$$\mathsf{dE}(\mathbf{y}, \mathbf{y}', \mathbf{k}') = \mathbf{k}' \begin{pmatrix} 1 & y_{\ell'_y+1} & \cdots & y_{\ell_y} \\ & & & \\ & & \mathbf{I} & \end{pmatrix}^\top .$$

It's straightforward to show that dE meets two requirements.

6.3 Generic Construction and Security Analysis

A direct way to support delegation in our framework in Section 3.1 is to apply dE to $\mathbf{K}_1$. However this delegation algorithm is not path-oblivious. Following [RS14], we publish $\mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W}$ in the master public key mpk in a proper form which makes it possible to publicly re-randomize any secret key.

Construction. Our tag-based ABE supporting delegation is as follows. We highlight all terms we add for delegation in the dashboxes.

– $\mathsf{Setup}(1^\lambda, \mathsf{P})$: Let n be the parameter size of predicate encoding supporting delegation $(\mathsf{sE}, \mathsf{rE}, \mathsf{kE}, \mathsf{rE}, \mathsf{rD}, \boxed{\mathsf{dE}})$ for P. Sample

$$\mathbf{A} \leftarrow \mathcal{D}_k, \quad \boxed{\mathbf{Z} \leftarrow \mathbb{Z}_p^{k \times k}}, \quad \mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \quad \mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$$

and output the master public and secret key pair

$$\mathsf{mpk} := \left\{ \begin{array}{l} [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_n^\top \mathbf{A}]_1, [\mathbf{W}^\top \mathbf{A}]_1 \\ \boxed{[\mathbf{Z}]_2, [\mathbf{W}_1 \mathbf{Z}]_2, \ldots, [\mathbf{W}_n \mathbf{Z}]_2, [\mathbf{W}\mathbf{Z}]_2} \end{array}, [\mathbf{k}^\top \mathbf{A}]_T \right\}$$
$$\mathsf{msk} := \{ \mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W}; \ \mathbf{k} \}.$$

– $\mathsf{Del}(\mathsf{mpk}, \mathsf{sk}_{y'}, y)$: Let $\mathsf{sk}_{y'} = \{ K'_0, \mathbf{K}'_1, K'_2 \}$. Sample $\widetilde{\mathbf{r}} \leftarrow \mathbb{Z}_p^k$ and compute a re-randomizer

$$\{ \widetilde{K}_0 := [\mathbf{Z}\widetilde{\mathbf{r}}], \ \widetilde{\mathbf{K}}_1 := \mathsf{rE}(y, [\mathbf{W}_1 \mathbf{Z}\widetilde{\mathbf{r}}]_2, \ldots, [\mathbf{W}_n \mathbf{Z}\widetilde{\mathbf{r}}]_2), \ \widetilde{K}_2 := [\mathbf{W}\mathbf{Z}\widetilde{\mathbf{r}}]_2 \}$$

and output

$$\mathsf{sk}_y := \{ K_0 := K'_0 \cdot \widetilde{K}_0, \ \mathbf{K}_1 := \mathsf{dE}(y, y', \mathbf{K}'_1) \cdot \widetilde{\mathbf{K}}_1, \ K_2 := K'_2 \cdot \widetilde{K}_2 \}$$

The remaining algorithms KeyGen, Enc and Dec are defined as in Section 3.1. The algorithm Del is path-oblivious: if we let $\mathbf{r}'$ be the random coin for $\mathsf{sk}_{y'}$, the random coin in sk_y will be $\mathbf{r} = \mathbf{r}' + \mathbf{Z}\widetilde{\mathbf{r}}$ which is independent of $\mathbf{r}'$ thanks to $\widetilde{\mathbf{r}}$.

Security. Observe that the only difference in the security game here is that the mpk sent to the adversary also includes

$$[\mathbf{Z}]_2, [\mathbf{W}_1 \mathbf{Z}]_2, \ldots, [\mathbf{W}_n \mathbf{Z}]_2, [\mathbf{W}\mathbf{Z}]_2,$$

since Del will not be involved. Therefore we can prove the adaptive security of our ABE scheme as in Section 3.2 and Section 3.3. In fact what we need to show here is how to simulate these extra entries in mpk in our previous proofs.

To prove Lemma 4, Lemma 5, Lemma 8, the simulator knows $\mathbf{W}_1, \ldots, \mathbf{W}_n$ and $\mathbf{W}$. It can sample matrix $\mathbf{Z} \leftarrow \mathbb{Z}_p^{k\times k}$ and simulate the extra entries directly. For Lemma 6 and Lemma 7, we recall that the simulator received $(\mathcal{G}, [\mathbf{M}]_2, [\mathbf{t}]_2)$ and implicitly define

$$\mathbf{W}_1 = \widetilde{\mathbf{W}}_1 + \gamma_1 \mathbf{V},\ \ldots,\ \mathbf{W}_n = \widetilde{\mathbf{W}}_n + \gamma_n \mathbf{V},\quad \mathbf{W} = \widetilde{\mathbf{W}} - \mathbf{V}$$

where $\widetilde{\mathbf{W}}_1, \ldots, \widetilde{\mathbf{W}}_n, \widetilde{\mathbf{W}} \leftarrow \mathbb{Z}_p^{(k+1)\times k}$, $\gamma_1, \ldots, \gamma_n \leftarrow \mathbb{Z}_p$ and $\mathbf{V} = \mathbf{a}^\perp \cdot (\underline{\mathbf{M}}\overline{\mathbf{M}}^{-1})$. As we have mentioned, the simulator can not calculate $\mathbf{V}$ and does not know $\mathbf{W}_1, \ldots, \mathbf{W}_n, \mathbf{W}$. However it can still simulate the extra entries as follows: Sample $\widetilde{\mathbf{Z}} \leftarrow \mathbb{Z}_p^{k\times k}$ and define

$$\mathbf{Z} = \overline{\mathbf{M}}\widetilde{\mathbf{Z}}.$$

Since $\overline{\mathbf{M}}$ is full-rank with high probability, matrix $\mathbf{Z}$ is distributed correctly and we have

$$\mathbf{W}_i\mathbf{Z} = \widetilde{\mathbf{W}}_i\overline{\mathbf{M}}\widetilde{\mathbf{Z}} + \gamma_i \mathbf{a}^\perp\underline{\mathbf{M}}\widetilde{\mathbf{Z}} \ \text{ for all } i \in [n] \quad \text{and} \quad \mathbf{W}\mathbf{Z} = \widetilde{\mathbf{W}}\overline{\mathbf{M}}\widetilde{\mathbf{Z}} - \mathbf{a}^\perp\underline{\mathbf{M}}\widetilde{\mathbf{Z}}.$$

That means we can simulate all extra entries from $[\mathbf{M}]_2$. This is sufficient to finish our proof.

Acknowledgement. We want to thank Benoît Libert, Somindu C. Ramanna and Hoeteck Wee for their helpful discussion, and thank Jiangtao Li for his proof-reading at early stage. We also thank all anonymous reviewers of ASIACRYPT 2017 for their valuable comments.

References

ABS17. Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. Generic transformations of predicate encodings: Constructions and applications. In *CRYPTO 2017*, volume 10401 of *LNCS*, pages 36–66. Springer, Cham, July 2017.

AC16. Shashank Agrawal and Melissa Chase. A study of pair encodings: Predicate encryption in prime order groups. In *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 259–288. Springer, Heidelberg, January 2016.

AC17. Shashank Agrawal and Melissa Chase. Simplifying design and analysis of complex predicate encryption schemes. In *Advances in Cryptology - EUROCRYPT 2017, Part I*, pages 627–656, 2017.

AHY15. Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. A framework for identity-based encryption with almost tight security. In *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 521–549. Springer, Heidelberg, November / December 2015.

AL10. Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *PKC 2010*, volume 6056 of *LNCS*, pages 384–402. Springer, Heidelberg, May 2010.

Att14. Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.

Att16. Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Heidelberg, December 2016.

AY15. Nuttapong Attrapadung and Shota Yamada. Duality in ABE: Converting attribute based encryption for dual predicate and dual policy via computational encodings. In *CT-RSA 2015*, volume 9048 of *LNCS*, pages 87–105. Springer, Heidelberg, April 2015.

BB04a. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, Heidelberg, May 2004.

BB04b. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Heidelberg, August 2004.

BBG05. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, Heidelberg, May 2005.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

BKP14. Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) identity-based encryption from affine message authentication. In *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, Heidelberg, August 2014.

BSW07. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011.

CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, April 2015.

CGW17. Jie Chen, Junqing Gong, and Jian Weng. Tightly secure IBE under constant-size master public key. In *Public-Key Cryptography - PKC 2017, Part I*, pages 207–231, 2017.

CHK03. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, Heidelberg, May 2003.

CW14. Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In *SCN 14*, volume 8642 of *LNCS*, pages 277–297. Springer, Heidelberg, September 2014.

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

GDCC16. Junqing Gong, Xiaolei Dong, Jie Chen, and Zhenfu Cao. Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 624–654. Springer, Heidelberg, December 2016.

Gen06. Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, Heidelberg, May / June 2006.

GHKW16. Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Heidelberg, May 2016.

GKW16. Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. In *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 361–388. Springer, Heidelberg, October / November 2016.

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS 06*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

JR13. Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2013.

KSG+17. Jongkil Kim, Willy Susilo, Fuchun Guo, Man Ho Au, and Surya Nepal. An efficient KP-ABE with short ciphertexts in prime ordergroups under standard assumption. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017*, pages 823–834, 2017.

KSGA16. Jongkil Kim, Willy Susilo, Fuchun Guo, and Man Ho Au. A tag based encoding: An efficient encoding for predicate encryption in prime order groups. In *SCN 16*, volume 9841 of *LNCS*, pages 3–22. Springer, Heidelberg, August / September 2016.

KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, April 2008.

LOS+10. 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 *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 62–91. Springer, Heidelberg, May 2010.

LW10. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Heidelberg, February 2010.

LW12. Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 180–198. Springer, Heidelberg, August 2012.

OSW07. Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *ACM CCS 07*, pages 195–203. ACM Press, October 2007.

OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Heidelberg, August 2010.

OT12. Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 591–608. Springer, Heidelberg, April 2012.

Ram16. Somindu C. Ramanna. More efficient constructions for inner-product encryption. In *ACNS 16*, volume 9696 of *LNCS*, pages 231–248. Springer, Heidelberg, June 2016.

RCS12. Somindu C. Ramanna, Sanjit Chatterjee, and Palash Sarkar. Variants of waters' dual system primitives using asymmetric pairings - (extended abstract). In *PKC 2012*, volume 7293 of *LNCS*, pages 298–315. Springer, Heidelberg, May 2012.

RS14. Somindu C. Ramanna and Palash Sarkar. Efficient (anonymous) compact HIBE from standard assumptions. In *ProvSec 2014*, volume 8782 of *LNCS*, pages 243–258. Springer, Heidelberg, October 2014.

RS16. Somindu C. Ramanna and Palash Sarkar. Efficient adaptively secure IBBE from the SXDH assumption. *IEEE Trans. Information Theory*, 62(10):5709–5726, 2016.

Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.

SW08. Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 560–578. Springer, Heidelberg, July 2008.

Wat05. Brent R. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Heidelberg, May 2005.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, August 2009.

Wat11. Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer, Heidelberg, March 2011.

Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, February 2014.

Wee16. Hoeteck Wee. Déjà Q: Encore! Un petit IBE. In *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 237–258. Springer, Heidelberg, January 2016.

WES17. Yohei Watanabe, Keita Emura, and Jae Hong Seo. New revocable IBE in prime-order groups: Adaptively secure, decryption key exposure resistant, and with short public parameters. In *Topics in Cryptology - CT-RSA 2017*, pages 432–449, 2017.

WS16. Yohei Watanabe and Junji Shikata. Identity-based hierarchical key-insulated encryption without random oracles. In *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 255–279. Springer, Heidelberg, March 2016.