Fully Adaptive Threshold IBE and Signatures in the Standard Model

Jiayun Yan[1], Yu Li[2], Jie Chen[3], Haifeng Qian[1], Xiaofeng Chen[4], and Debiao He[3]

[1] East China Normal University, Shanghai, China
yanjiayun01@163.com, hfqian@cs.ecnu.edu.cn
[2] University of Wollongong, Wollongong NSW, Australia
im@liyu.im
[3] Wuhan University, Wuhan, China
s080001@e.ntu.edu.sg, hedebiao@whu.edu.cn
[4] Xidian University, Xi'an, China
xfchen@xidian.edu.cn

Abstract. We present fully adaptive secure threshold IBE and threshold signatures, which rely on the k-Linear assumption in the standard model over asymmetric pairing groups. In particular, our threshold signature scheme achieves a non-interactive signing process and an adaptively secure guarantee as strong as Das-Ren (CRYPTO'24), while their proof relies on the random oracle model. We achieve our results by following steps: First, we design two threshold IBE schemes against adaptive corruptions in the composite-order and prime-order groups by adopting the dual system groups encoding. Second, we provide a generic transform from threshold IBE to threshold signatures, following Naor's paradigm, which reduces the fully adaptive corruption security of threshold signatures to threshold IBE. Third, we present two threshold signatures instantiations in composite-order and prime-order groups.

Keywords: Threshold IBE · Threshold signatures · Adaptive corruptions · Dual system groups

1 Introduction

Threshold cryptography [26,34,15] avoids single points of failure by splitting cryptographic keys into $n > 1$ shares, which are held by servers in such a way that at least t out of n servers should contribute to private key operations. In (t, n)-threshold cryptosystems, the adversary breaking into up to $t-1$ servers should be unable to threaten the system's security. Threshold signatures [25,26,15] protect the signing key by sharing it among a group of signers so that an adversary must corrupt a threshold number of signers to be able to forge signatures. The increasing demand for decentralized applications has resulted in the large-scale adoption of threshold signature schemes.

Static vs. Adaptive Security. Most threshold systems [44,10,28] have been analyzed in a static corruption model, where the adversary chooses which servers

it wants to corrupt before the scheme is set up. Adaptive adversaries can choose whom to corrupt at any time, as a function of their entire view of the protocol execution, and are known to be strictly stronger. Clearly, an adaptive adversary is a far stronger notion than static security and matches reality more closely. Unfortunately, proving adaptive security for threshold signatures is highly challenging and previous works rely on the non-standard model [4,24,5].

The Challenge: Adaptive Corruptions in the Standard Model. Threshold IBE was proposed by Boneh et al. [10], which is based on static security. While designing adaptively secure threshold IBE and signatures is non-trivial, preserving compatibility with non-threshold schemes poses an additional challenge. In the standard model, the work of Libert and Yung [40] achieved adaptive security, but its security level was weak, as it prevented adversaries from querying the signature share of the challenge message. Although recent research [22,24,5,3,2,36] has achieved stronger adaptive security (allowing querying the challenge message), these advances rely on random oracle models. Very recently, Gong et al. [32] constructed a threshold batched IBE in the standard model. However, their construction is limited to *static security* (for the threshold setting) and relies on q-type assumptions. Therefore, how to achieve this stronger adaptive security in the standard model under standard static assumptions remains an open and challenging question.

Table 1. Comparison of existing threshold signature schemes.

Reference	Non-interactive	Adaptive corruption	Assumption	Security model
[LY13] [40]	✓	✓(weak)	Subgroup Decision	Standard Model
[CKM23] [22]	✗	✓	AOMDL	ROM + AGM
[CGRS23] [19]	✗	✗	AOMDL	ROM
[BLT+24][5]	✗	✓	DDH	ROM
[DR24] [24]	✓	✓	DDH + co-CDH	ROM
[BDLR24] [2]	✗	✓	DDH	ROM
[BDLR25] [3]	✗	✓	DDH	ROM
Ours	✓	✓	k-Linear	Standard Model

– AOMDL: Algebraic One-More Discrete Logarithm assumption; ROM: Random Oracle Model; AGM: Algebraic Group Model.
– [40] achieves weak adaptive corruption, meaning that the challenge message $m^\star$ was never submitted to the signing oracle; [22,5,24,3,2] and our schemes achieve adaptive corruption, meaning that the adversary is allowed to query the signature share of the challenge message $m^\star$.

Our Contributions. In this work, we propose fully adaptive threshold IBE and non-interactive threshold signatures based on standard assumptions in the standard model. Specifically, we present instantiations in both composite-order groups and prime-order groups. We summarize existing threshold signatures in Table 1.

– **Fully Adaptive Security:** Unlike Libert and Yung [40] (weak adaptive) and Chu et al. [19], our scheme handles fully adaptive corruptions. The adversary can corrupt up to $t-1$ authorities at any phase.
– **Standard Model & Assumptions:** Compared to these fully adaptive threshold signatures [22,5,24,3,2], which are in the random oracle models, our threshold schemes rely on the standard model under the k-Linear assumption in pairing groups. Note that the size of the combined signature in [24] is quite efficient, only one group element, while ours requires two group elements.
– **Non-Interactive Signing:** Our resulting threshold signatures require no interaction between signers during the signing phase, ensuring low latency and high scalability.

1.1 Technical Overview

High-level Overview. Intuitively, we observe that Naor's generic transform converts an IND-ID-CCA secure identity-based encryption (IBE) scheme into an unforgeability under chosen message attack (UF-CMA) secure signature scheme. In this paradigm, the identity id in the IBE scheme corresponds to the message m to be signed, and the user secret key $\mathsf{sk}_{\mathsf{id}}$ corresponds to the signature σ. This naturally inspires us to explore whether an IBE with threshold authority key generation can be transformed into threshold signatures. This can be achieved by applying Shamir's secret sharing to divide the master secret key of the fully secure IBE scheme among n authorities. In particular, we aim to construct an *adaptively secure* threshold signatures by building on a fully secure threshold IBE against adaptive corruption. Wee's dual system groups can achieve fully adaptive security in the IBE scheme. As a result, we reduce the challenge of constructing an adaptively secure threshold signature scheme to the problem of building a fully secure threshold IBE scheme against adaptive corruption.

Technical Challenge. A major technical challenge arises when the adversary queries challenge identity $\mathsf{id}^\star$, which cannot be handled directly via Waters' dual system proof due to their restriction $\mathsf{id} \neq \mathsf{id}^\star$. However, in both adaptively secure threshold IBE (signatures), the adversary is allowed to query secret key shares associated with the challenge identity $\mathsf{id}^\star$ (the challenge message $m^\star$ for signatures). It is not trivial to extend Naor's paradigm [11] to threshold settings, which introduces new challenges. The adversary can not only retrieve the key, but also directly corrupt some of the signers. This hybrid attack requires that the scheme remain secure even when partial key information is leaked. Therefore, we must address two distinct dimensions of adaptivity: *adaptive corruption of authorities* and *adaptive chosen identity/message*. We tackle these challenges using a hybrid strategy.

Handling Adaptive Corruptions via Dual System Groups. Wee's dual system group encoding framework [47] was adopted to construct our threshold

IBE against adaptive corruption, which builds upon Waters' dual system encryption methodology [46]. Our resulting threshold IBE handles secret key share of identity where $\mathsf{id} \neq \mathsf{id}^\star$ in a straightforward manner by responding using standard dual system techniques. For corruption queries, the normal master secret key shares of authority i are responded. Due to the threshold constraint t and the indistinguishability guarantees of the dual system method, the adversary gains no advantage in the final game.

Handling Adaptive Messages via Guessing Strategy. For the challenge identity (or message in signatures), we observe that in the threshold security game, the adversary outputs a forgery on a message $m^\star$ after completing queries. Crucially, unlike standard signatures, the adversary is permitted to query signature shares for $m^\star$, provided that the number of queried shares plus corrupted authorities is strictly less than t.

Inspired by this constraint, we simulate the threshold IBE security game by assuming that the adversary queries at most q distinct identities and corrupts at most $t-1$ authorities. We unify both phases of the adaptive query process and rearrange all ids in the sequence they are queried. Then, we employ a guessing strategy where we randomly select a position $k \in [1, q+1]$ as the predicted index for the challenge identity $\mathsf{id}^\star$ (assuming the $(q+1)$-th position if $\mathsf{id}^\star$ is never queried). Based on this guess strategy, all the secret key shares of id_i are transformed into semi-functional (SF) secret key shares using the dual system groups, independent of the query phase. Since the dual system proof does not distinguish between query phases or positions, this guessing strategy allows us to adapt Waters' and Wee's techniques to prove fully adaptive security. It is worth noting that this reduction, which guesses the index of the challenge identity $\mathsf{id}^\star$ among the q queries, incurs a security loss of $O(1/q)$.

Proof Strategy. Throughout the proof, we employ a hybrid simulation strategy. For secret key queries on an identity id, we generate shares using the dual system transformation. In contrast, for corruption queries on an authority i, we respond with the normal master secret key share msk_i. This approach naturally supports adaptive corruption, as the simulator can truthfully reveal the keys of any $t-1$ authorities chosen dynamically by the adversary. The security of this strategy relies on the threshold guarantee of Shamir's secret sharing. Even though the adversary holds up to $t-1$ normal master key shares, these insufficient shares are information-theoretically strictly hidden from the full master secret. Consequently, the adversary cannot leverage these normal shares to distinguish the SF components embedded in the secret key shares.

Specifically, in $\mathsf{Game}_{2,\kappa,2}$, we transition the secret key share from pseudo-normal to pseudo-SF by replacing the secret parameter α in the SF space with a fresh random value. This step is justified by the α-privacy property established in the dual system group framework [47], which guarantees that the secret in the SF component is statistically indistinguishable from uniform randomness. Crucially, in our threshold setting, the adversary holds fewer than t shares, leaving the normal component of the secret key share information-theoretically

underdetermined. This unknown normal component acts as a one-time pad that masks the transition occurring in the SF space. Consequently, by combining the α-privacy of the encoding randomness with the threshold constraint, the joint distribution of the shares remains indistinguishable to the adversary, even under adaptive corruptions. In all subsequent games, any SF secret key share returned to the adversary also follows this pattern. Thus, our proof technique works in both static and adaptive corruption settings.

From Threshold IBE to Threshold Signatures. We apply a threshold-adapted Naor transform. Since our underlying TIBE supports adaptive corruptions and adaptive identities, the resulting threshold signature scheme inherits these properties, yielding fully adaptive security under adaptive corruptions.

Threshold IBE in Composite Order Groups. We construct our threshold IBE scheme by adopting the dual system groups framework introduced by Wee [47]. Specifically, we first instantiate a new IBE scheme in asymmetric composite-order bilinear groups (where the group order is a product of two primes) by applying the transformation technique of Chen and Wee [17] to the Lewko-Waters IBE [38]. In this construction, the master secret key msk contains components from both the normal and semi-functional subgroups.

Let $\mathbb{G}$ and $\mathbb{H}$ be composite-order groups of order $N = p_1p_2$, with subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2} \subset \mathbb{G}$ and $\mathbb{H}_{p_1}, \mathbb{H}_{p_2} \subset \mathbb{H}$. Let g, g_1, g_2 be generators of $\mathbb{G}, \mathbb{G}_{p_1}, \mathbb{G}_{p_2}$, and h, h_1, h_2 be generators of $\mathbb{H}, \mathbb{H}_{p_1}, \mathbb{H}_{p_2}$, respectively. Let $\mathsf{H} : \mathbb{G}_T \to \mathbb{G}_{p_2}$ be a hash function. The ciphertext and secret key structure are as follows:

$$\mathsf{ct} = (\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_0) = \left(\mathsf{c}_1 = g_1^{s(a\mathsf{id}+b)}, \mathsf{c}_2 = g_1^s, \mathsf{c}_0 = \mathsf{H}(e(g_1, h_1)^{\alpha \cdot s}) \cdot m\right),$$

$$\mathsf{sk} = (\mathsf{k}_1, \mathsf{k}_2) = \left(\mathsf{k}_1 = h^\alpha \cdot h_1^{r(a\mathsf{id}+b)}, \mathsf{k}_2 = h_1^r\right),$$

where $a, b, s, r \leftarrow_R \mathbb{Z}_N$, $\alpha \in \mathbb{Z}_N$. Building on this IBE scheme, we present a threshold setting against adaptive corruptions. We employ Shamir's secret sharing to distribute the master secret α among authorities and use Lagrange interpolation to reconstruct the randomness r during the combination phase. The security proof follows the sequence of games outlined in Table 2.

Table 2. Sequence of games in the semi-functional space in the asymmetric bilinear groups composed of two subgroups.

Game	ct	msk	sk	Justification	Remark
0	g_1	h^α	h_1		Real scheme
1	$g_1 \mapsto g$	h^α	h_1	DS1	Normal to SF cipheretxt
$2.i.1$	g	h^α	$h_1 \mapsto h$	DS2	Normal to pesudo-normal sk
$2.i.2$	g	$h^{\alpha_i} h_2^{\alpha'_i}$	h	α-privacy & $\boldsymbol{w}$-hiding	Pesudo-normal to pesudo-SF sk
$2.i.3$	g	$h^{\alpha_i} h_2^{\alpha'_i}$	$h \mapsto h_1$	DS2	pesudo-SF to SF sk
3	g	$h^{\alpha_i} h_2^{\alpha'_i}$	h_1	Statistical	Final scheme

Porting to Prime Order Groups. We transform the above composite-order construction into prime-order construction by using the encoding in Chen et al. [16] as follows:

$$\mathsf{ct} : g_1 \mapsto [\mathbf{A}]_1 , \quad g_1^s \mapsto [\mathbf{As}]_1 , \quad g_2 \mapsto \left[\mathbf{b}^{\perp}\right]_1 .$$

$$\mathsf{sk} : h_1 \mapsto [\mathbf{B}]_2 , \quad h_1^r \mapsto [\mathbf{Br}]_2 , \quad h_2 \mapsto \left[\mathbf{a}^{\perp}\right]_2 .$$

where $\mathbf{A} \leftarrow_R \mathbb{Z}_p^{(k+1)\times k}, \mathbf{B} \leftarrow_R \mathbb{Z}_p^{(k+1)\times k}$, $\mathbf{a}^{\perp}, \mathbf{b}^{\perp} \leftarrow_R \mathbb{Z}_p^{k+1}$, and $\mathbf{s}, \mathbf{r} \leftarrow_R \mathbb{Z}_p^k$.

1.2 Related Works

Threshold signatures were first introduced by Desmedt [25]. Since then, numerous threshold signature schemes with various properties have been proposed [25,15,8]. Most of the threshold signature schemes [30,31,8,14,21,19,7,33] are proven secure only against a static adversary. However, the adaptive adversary is a safer and more realistic assumption for decentralized settings. We next review threshold signatures with adaptive security.

Adaptive Security. General techniques for achieving adaptive security have been studied [15,41], However, these techniques rely on heavy cryptographic machinery. Existing adaptively secure threshold signature scheme have to make major sacrifices, such as relying on parties to erase their internal states [15,40], inefficient cryptographic primitives like non-committing encryptions [35,41], or strong and non-standard assumptions such as one more discrete logarithm (OMDL) in the Algebraic Group Model [4,21].

Other Algebraic Structures. In the pairing setting, a natural construction is the (non-interactive) threshold version of the BLS signature scheme [12], which has been modified to achieve adaptive security in [39]. Recently, Bacho and Loss [4] have proven the adaptive security of threshold BLS in the AGM. Das et al. have constructed weighted threshold signatures in the pairing setting [23], and Crites et al. have constructed structure-preserving threshold signatures in the pairing setting [20]. Threshold signatures have been constructed based on RSA [42,45]. Several works have proposed threshold signing protocols for ECDSA signatures [14,29,1].

Non-interactive Threshold Signatures. A non-interactive threshold signature scheme requires each signer to send a single message to sign. Practical, robust, non-interactive threshold signatures were described by Shoup [44] under the RSA assumption and by Katz and Yung [37] assuming the hardness of factoring. Boldyreva [8] presented a non-interactive threshold BLS signature scheme. Until recently, these schemes were proven secure against static adversaries only. Recently, Das et al. proved adaptive security for non-interactive threshold signature [24] in the standard assumption and ROM. This implies that a new protocol is needed with adaptive security under the standard model and standard assumptions.

Paper Organization. We discuss the related knowledge in Sec. 2. In Sec. 3, we present threshold IBE construction in composite-order groups. In Sec. 4, we then

describe our threshold IBE scheme in prime-order groups. In Sec. 5, we describe our generic construction of threshold signatures from threshold IBE. In Sec. 6, we present two instantiations of the threshold signatures in composite-order groups and prime-order groups.

2 Preliminaries

Notations. For any integer n, we use $[n]$ to denote the ordered set $\{1, 2, \cdots, n\}$. If S is a finite set, then $r \leftarrow_R S$ denotes sampling r uniformly at random from S. If f is an algorithm or function, then $y \leftarrow f(x)$ denotes the output of this algorithm with x as input. $y := x$ denotes that y is defined or substituted by x. Unless otherwise specified, algorithms in this work are randomized, and PPT stands for probabilistic polynomial time. We use lowercase letters (e.g., a, r, s) to denote elements in vectors or matrices, bold lowercase letters (e.g., $\mathbf{a}, \mathbf{s}, \mathbf{z}$) to denote vectors and bold uppercase letters (e.g., $\mathbf{A}, \mathbf{B}$) to denote matrices. We say a function $\varepsilon(\lambda)$ is *negligible* in λ, if $\varepsilon(\lambda) = o(1/\lambda^c)$ for every $c \in \mathbb{Z}$, and we write $\mathsf{negl}(\lambda)$ to denote a negligible function in λ.

2.1 Shamir secret sharing and Bilinear pairing groups

Shamir secret sharing. The Shamir secret sharing [43] embeds the secret α in the constant term of a polynomial $\alpha(x) = \alpha + a_1 x + a_2 x^2 + \cdots a_{t-1} x^{t-1}$, where other coefficients $a_1, \cdots, a_{t-1}$ are chosen uniformly randomly from a field $\mathbb{Z}_p$. The i-th share of the secret is $\alpha(x)$, i.e., the polynomial evaluated at i. Given t distinct shares, one can efficiently reconstruct the polynomial and the secret α using Lagrange interpolation. Also, α is information-theoretically hidden from an adversary that knows $t-1$ or fewer shares.

Definition 1 (Asymmetric bilinear pairing groups). *Asymmetric bilinear pairing groups $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, e)$ are a tuple of a prime p, cyclic (multiplicative) groups $\mathbb{G}, \mathbb{H}$ and $\mathbb{G}_T$ of order p, $g \neq 1 \in \mathbb{G}$, $h \neq 1 \in \mathbb{H}$, and a polynomial-time computable nondegenerate bilinear pairing $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ i.e., $e(g^s, h^t) = e(g, h)^{st}$ and $e(g, h) \neq 1$. We refer to $\mathbb{G}$ and $\mathbb{H}$ as the source groups and refer to $\mathbb{G}_T$ as the target group.*

Remark 1. We follow the notation and algebraic framework for Diffie-Hellman-like assumptions in [27]. Given $a \in \mathbb{Z}_p$, we use $[a]_1$ to denote $g^a \in \mathbb{G}$, $[a]_2$ to denote $h^a \in \mathbb{H}$, $[a]_T$ to denote $e(g, h)^a \in \mathbb{G}_T$. This obviously extends to vectors and matrices. We define $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{A}^\top \mathbf{B}]_T$.

Definition 2 (Composite-order Group). *The group order N is the product of two distinct primes p_1 and p_2. We denote by $\mathbb{G}_{p_i}$ the subgroup of $\mathbb{G}$, where $p_i \in \{p_1, p_2\}$. We use g_1, g_2 denote random generators of the subgroups $\mathbb{G}_{p_i}$ respectively. The same representation also applies in group $\mathbb{H}$.*

2.2 Dual System Groups

Dual system groups [46,47,16] contain a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$, a non-generate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, and consist of the following properties: orthogonality, non-degeneracy, associativity, α-privacy, and $\boldsymbol{w}$-hiding. See more details in Appendix A.

2.3 Complexity Assumptions

Assumption 1 (k-Lin: the k-Linear assumption in $\mathbb{G}$) *Let $\mathcal{D}_k$ be an efficiently samplable distribution of matrices $(\mathbf{A}, \mathbf{a}^\perp)$ over $\mathbb{Z}_p^{(k+1)\times k} \times \mathbb{Z}_p^{k+1}$ so that $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$ and $\mathbf{a}^\perp \neq \mathbf{0}$. This distribution captures the k-Linear assumption, which stipulates that*

$$([\mathbf{A}], [\mathbf{As}]) \approx_c ([\mathbf{A}], [\mathbf{z}]),$$

where $\mathbf{s} \leftarrow_R \mathbb{Z}_p^k$, $\mathbf{z} \leftarrow_R \mathbb{Z}_p^{k+1}$ in both $\mathbb{G}$ and $\mathbb{H}$. For any adversary $\mathcal{A}$, we define the advantage function:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{k\text{-}\mathsf{Lin}} :=& |\Pr[\mathcal{A}((p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, e); [\mathbf{A}]_1, [\mathbf{As}]_1) = 1] - \\ & \Pr[\mathcal{A}((p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, e); [\mathbf{A}]_1, [\mathbf{z}]_1) = 1]|,\end{aligned}$$

where $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, e) \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{s} \leftarrow_R \mathbb{Z}_p^k$, $\mathbf{z} \leftarrow_R \mathbb{Z}_p^{k+1}$, $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$.

We will slightly abuse notation and also use $\mathsf{Adv}_{\mathcal{A}}^{k\text{-}\mathsf{Lin}}$ to denote the corresponding advantage function for $\mathbb{H}$. Our constructions in composite-order bilinear groups rely on the following two decisional subgroup assumptions [6,17,47].

Assumption 2 (Decisional left subgroup indistinguishability in $\mathbb{G}$, DS1) *Given a group generator $\mathcal{G}(\cdot)$, we assume that for any PPT adversary $\mathcal{A}$*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DS1}}(1^\lambda) := |\mathsf{Pr}[\mathcal{A}(D, T_0)] = 1] - \mathsf{Pr}[\mathcal{A}(D, T_1) = 1]|,$$

where,

$$\begin{aligned}&(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, e) \leftarrow \mathcal{G}(1^\lambda) \text{ where } N = p_1 p_2 \text{ for distinct primes } p_1, p_2 \\ &D := (N, \mathbb{G} = \mathbb{G}_{p_1}\mathbb{G}_{p_2}, e, g_1, g), \quad g_1 \leftarrow_R \mathbb{G}_{p_1}, \quad g \leftarrow_R \mathbb{G} \\ &T_0 \leftarrow_R \mathbb{G}_{p_1}, \quad T_1 \leftarrow_R \mathbb{G}\end{aligned}$$

Assumption 3 (Decisional right subgroup indistinguishability in $\mathbb{H}$, DS2) *This assumption (denoted as DS2) is completely symmetric to Assumption 2 on group $\mathbb{H}$. The challenger generates $D := (N, \mathbb{H} = \mathbb{H}_{p_1}\mathbb{H}_{p_2}, e, h_1, h)$ and the tuples T_0, T_1 are selected from $\mathbb{H}_{p_1}$ and $\mathbb{H}$, respectively.*

Assumption 2 (resp. 3) asserts that for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DS1}}(1^\lambda)$ (resp. $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DS2}}(1^\lambda)$) is a negligible function in 1^λ.

2.4 Threshold IBE

Definition 3 (Threshold IBE [10]). *A threshold identity-based encryption scheme consists of several algorithms as follows:*

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pp}, \mathsf{mpk}, \mathsf{msk})$*: Takes as input the security parameter* λ *and outputs system parameters* pp*, master public key* mpk*, and master secret key* msk*.*
- $\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}) \to (\{\mathsf{pk}_i\}_{i\in[n]}, \{\mathsf{msk}_i\}_{i\in[n]})$*: Takes as input the system parameters* pp *and master secret key* msk*. It outputs public key shares* $\{\mathsf{pk}_i\}_{i\in[n]}$ *is called a public verification key shares, and* $\{\mathsf{msk}_i\}_{i\in[n]}$ *is a vector of master key shares. Each authority* i *is given the master key share* (i, msk_i)*.*
- $\mathsf{Enc}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_x, m) \to \mathsf{ct}$*: Takes as input* pp*, master public key* mpk*, an identity* id_x*, and a message* m*, and outputs a ciphertext* ct*.*
- $\mathsf{ValidateCT}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_x, \mathsf{ct})$*: Takes as input* pp*, master public key* mpk*, an identity* id_x*, and ciphertext* ct*. It outputs* valid *or* $\mathsf{invalid}$*. If* valid*, we say that* ct *is* valid *encryption under* id_x*.*
- $\mathsf{ShareKeyGen}(\mathsf{pp}, \mathsf{mpk}, i, \mathsf{msk}_i, \mathsf{id}_y) \to \mathsf{sk}_{i,\mathsf{id}_y}$*: Takes as input* pp*, master public key* mpk*, an identity* id_y*, and a master key share* msk_i*. It outputs a secret key share* $\mathsf{sk}_{i,\mathsf{id}_y}$ *for* id_y*.*
- $\mathsf{ShareVerify}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_y, \mathsf{pk}_i, \mathsf{sk}_{i,\mathsf{id}_y})$*: Takes as input* pp*, master public key* mpk*, an identity* id_y*, and a secret key share* $\mathsf{sk}_{i,\mathsf{id}_y}$*. It outputs* valid *or* $\mathsf{invalid}$*.*
- $\mathsf{Combine}(\mathsf{pp}, \mathsf{id}_y, \{\mathsf{sk}_{i,\mathsf{id}_y}\}_{i\in S}) \to \mathsf{sk}_{\mathsf{id}_y}$*: Takes as input* pp*, an identity* id_y*, and* $|S|$ *secret key shares* $\{\mathsf{sk}_{1,\mathsf{id}_y}, \cdots, \mathsf{sk}_{|S|,\mathsf{id}_y}\}$ *from a subset of authorities* $S \subset [n]$ *with* $|S| \geq t$*, . It outputs a secret key* $\mathsf{sk}_{\mathsf{id}_y}$ *or* $\perp$*.*
- $\mathsf{Dec}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_y, \mathsf{sk}_{\mathsf{id}_y}, \mathsf{ct}) \to m/\perp$*: Takes as input* pp*, master public key* mpk*, an identity* id_y*, secret key* sk_{id_y}*, and ciphertext* ct*. It outputs a message* m *or* $\perp$*.*

Security of a TIBE scheme is defined using two properties: security against chosen identity attacks and consistency of key generation. We first define the security of the TIBE scheme against adaptively chosen identity attacks (or fully secure) and adaptive corruption, and then the definition of consistency of key generation.

Fully secure TIBE against adaptive corruption. The security game is defined by the following experiment, played by challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

- Setup. The challenger $\mathcal{C}$ runs the $\mathsf{Setup}(1^\lambda)$ algorithm to obtain the public system parameters pp, a vector of master secret key shares $\{\mathsf{msk}_i\}_{i\in[n]}$ and a vector of public key shares $\{\mathsf{pk}_i\}_{i\in[n]}$ as verification keys. It gives pp and $\{\mathsf{pk}_i\}_{i\in[n]}$ to the adversary $\mathcal{A}$ and keeps $\{\mathsf{msk}_i\}_{i\in[n]}$ to itself.
- $\mathsf{Phase\ 1}$. The adversary $\mathcal{A}$ adaptively makes the following kinds of queries:
  - Corruption query: $\mathcal{A}$ selects authority $i \in \{1, \cdots, n\}$ and obtains its msk_i. No more than $t-1$ authorities' master key shares can be obtained by $\mathcal{A}$ in the whole game.

- Secret key share query: $\mathcal{A}$ requests secret key share of any identity (i, id_y), where $i \in \{1, \cdots, n\}$. The challenger responds with the corresponding secret key $\mathsf{sk}_{i,\mathsf{id}_y}$, which it generates by running the key share generation algorithm $\mathsf{ShareKeyGen}(\mathsf{pp}, \mathsf{mpk}, i, \mathsf{msk}_i, \mathsf{id}_y)$.

– $\mathsf{Challenge}$. The adversary $\mathcal{A}$ submits two messages m_0 and m_1 of equal length and a challenge identity $\mathsf{id}_y^\star$ with the restriction that the sum of the number of secret key share queries of $\mathsf{id}_y^\star$ and corruption queries is at most $t-1$. The challenger picks $b \leftarrow \{0,1\}$, and encrypts m_b under identity $\mathsf{id}_y^\star$ by running the encryption algorithm. It sends the ciphertext ct_b to the adversary $\mathcal{A}$.
– $\mathsf{Phase\ 2}$. $\mathcal{A}$ adaptively makes further queries as in $\mathsf{Phase\ 1}$ with the restriction that the sum of the number of secret key share queries of $\mathsf{id}_y^\star$ and corruption queries is at most $t-1$ in the whole game.
– Guess. The adversary $\mathcal{A}$ must output a guess b' for b.

The advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{TIBE}}(\lambda)$ of an adversary $\mathcal{A}$ is defined to be $|\Pr[b' = b] - 1/2|$.

Key Generation Consistency. Consistency of key generation (and decryption) is defined using the following game, which starts with the Setup, and $\mathsf{Phase\ 1}$ steps as in the game above. The adversary then outputs an identity id, ciphertext ct, and two sets of secret key shares $S = (\mathsf{sk}_{1,\mathsf{id}_y}, \cdots, \mathsf{sk}_{t,\mathsf{id}_y})$ and $S' = (\mathsf{sk}'_{1,\mathsf{id}_y}, \cdots, \mathsf{sk}'_{t,\mathsf{id}_y})$ each of size t. Let pp and $\mathbf{pk}$ be the system parameters and the public verification key generated in the Setup step. The adversary wins if:

1. the shares in S and S' are valid secret key shares for id under $\mathbf{pk}$;
2. S and S' each contain secret key shares from t distinct authorities;
3. ct is valid for the given id, i.e. $\mathsf{ValidateCT}(\mathsf{pp}, \mathsf{id}, \mathsf{ct})$ is valid;
4. the keys $\mathsf{sk}_{\mathsf{id}_y} = \mathsf{Combine}(\mathsf{pp}, \mathsf{id}_y, S)$ and $\mathsf{sk}'_{\mathsf{id}_y} = \mathsf{Combine}(\mathsf{pp}, \mathsf{id}_y, S')$ are such that $\perp \neq \mathsf{sk}_{\mathsf{id}_y} \neq \mathsf{sk}'_{\mathsf{id}_y} \neq \perp$;
5. $\mathsf{Dec}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}) \neq \mathsf{Dec}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}, \mathsf{sk}'_{\mathsf{id}}, \mathsf{ct})$.

Let $\mathsf{Adv\text{-}CD}_{\mathcal{A}}^{\mathsf{TIBE}}(\lambda)$ be the adversary's advantage in winning the game.

Definition 4. *A TIBE scheme is* fully secure *and against* adaptive corruption *if all PPT adversaries $\mathcal{A}$, its advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{TIBE}}(\lambda)$ and $\mathsf{Adv\text{-}CD}_{\mathcal{A}}^{\mathsf{TIBE}}(\lambda)$ are a negligible function in λ.*

2.5 Threshold Signatures

We introduce the syntax and security definitions for threshold signatures. We focus on schemes with non-interactive signing and deterministic verification. Our security definitions are from those of [13,24].

Definition 5 (Non-interactive threshold signatures). *Let t, n be natural numbers with $t < n$. A non-interactive (n,t)-threshold signature schemes for a finite message space $\mathcal{M}$ is a tuple of polynomial time algorithms $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PSign}, \mathsf{PVer}, \mathsf{Combine}, \mathsf{Ver})$ defined as follows:*

$\mathsf{Setup}(1^\lambda) \to (\mathsf{pp}, \mathsf{mpk}, \mathsf{msk})$: *Takes as input the security parameter* λ *and outputs system parameters* pp*, master public key* mpk*, and master secret key* msk*.*

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}) \to (\{\mathsf{pk}_i, \mathsf{msk}_i\}_{i\in[n]})$: *Takes as input master sect key* msk*. The key generation algorithm outputs a vector of threshold public key shares* $\{\mathsf{pk}_i\}_{i\in[n]}$ *and a vector of master secret key shares* $\{\mathsf{msk}_i\}_{i\in[n]}$*.*

$\mathsf{PSign}(\mathsf{pp}, \mathsf{mpk}, \mathsf{msk}_i, m) \to \sigma_i$: *The partial signing takes as input master public key* mpk*, master secret key shares* msk_i*, and a message* $m \in \mathcal{M}$*. It outputs a signature share* σ_i*.*

$\mathsf{PVer}(\mathsf{pp}, \mathsf{mpk}, \mathsf{pk}_i, m, \sigma_i) \to 0/1$: *The partial signature verification takes as input master public key* mpk*, a threshold public key share* pk_i*, a message* m*, and a signature share* σ_i*. It outputs 1 (accept) or 0 (reject).*

$\mathsf{Combine}(\mathsf{pp}, m, \{\sigma_i\}_{i\in S}) \to \sigma/\perp$: *The combine algorithm takes as input a message* m*, and signature shares* $\{\sigma_i\}_{i\in S}$ *of signers from a subset of authorities* $S \subset [n]$ *with* $|S| \geq t$*, . It outputs either a signature* σ *or* $\perp$*.*

$\mathsf{Ver}(\mathsf{pp}, \mathsf{mpk}, m, \sigma) \to 0/1$*. The signature verification algorithm takes as input public parameters* pp*, master public key* mpk*, a message* m*, and a signature* σ*. It outputs 1 (accept) or 0 (reject).*

We require a non-interactive (n, t)-threshold signature scheme to satisfy the Unforgeability and Robustness properties we describe next.

Security Model of UF-CMA. The Unforgeability Under Chosen Message Attack (UF-CMA) with adaptive corruption security game is defined by the following experiment, played by a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

- $\mathsf{Setup}(1^\lambda)$ The challenger runs the $\mathsf{Setup}(n, t, 1^\lambda)$ algorithm to obtain a public system parameters pp, and run KeyGen algorithm to obtain vector of public key shares $\mathbf{pk} = (\mathsf{pk}_1, \cdots, \mathsf{pk}_n)$ and a vector of master secret key shares $\mathbf{msk} = (\mathsf{msk}_1, \cdots, \mathsf{msk}_n)$. It gives pp and $\mathbf{pk}$ to the adversary $\mathcal{A}$ and keeps $\mathbf{msk}$ to itself. Initialize sets $\mathcal{H} := \{1, \cdots, n\}, \mathcal{CR} := \emptyset$.
- $\mathsf{Query\ Phase}$ The adversary $\mathcal{A}$ adaptively makes the following kinds of queries:
  - Corruption query: $\mathcal{A}$ choose $i \in \{1, \cdots, n\}$ and obtains signing key msk_i. In this case, set $\mathcal{CR} = \mathcal{CR} \cup \{i\}$ is the malicious set and $\mathcal{H} = \mathcal{H} \backslash \{i\}$ is the honest set. No more than $t-1$ master secret key shares can be obtained by $\mathcal{A}$ in the whole game. The game also maintains a list Q to store the subset of parties $\mathcal{A}$ has queried for partial signatures. Initially, $Q[m] = \{\}$ for every message m.
  - Signature query: The adversary $\mathcal{A}$ submits a pair (i, m) for $i \in \mathcal{H}$. The challenger responds with the corresponding signature σ, which it generates by running the PSign algorithm.
- $\mathsf{Forgery}$ The adversary $\mathcal{A}$ outputs a message $m^\star$ and a signature $\sigma^\star$, let $\mathcal{S} \subset \{1, \cdots, n\}$ denote the subset of parties for which $\mathcal{A}$ made a signing query of the form $(i, m^\star)$. Output 1 if $|\mathcal{CR} \cup \mathcal{S}| < t$ (The adversary did not have enough shares to construct the signature legally) and $\mathsf{Ver}(\mathsf{pp}, m^\star, \sigma^\star) = 1$. Otherwise, output 0.

Definition 6 (Unforgeability Under Chosen Message Attack, UF-CMA). *Let* $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PSign}, \mathsf{PVer}, \mathsf{Combine}, \mathsf{Ver})$ *be a* (n, t)*-threshold signature scheme. Consider the game* $\mathsf{UF\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}$ *defined above. We say that the scheme TS is* $\mathsf{UF\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}$ *secure, if for all PPT adversaries* $\mathcal{A}$*, the following advantage is negligible, i.e.*

$$\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{A},\mathsf{TS}}(\lambda) := \Pr[\mathsf{UF\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}(\lambda) = 1] = \mathsf{negl}(\lambda).$$

Security Model of RB-CMA. The Robustness Under Chosen Message Attack (RB-CMA) with adaptive corruption security game is defined by the following experiment, played by a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

- $\mathsf{Setup}(1^\lambda)$ The challenger runs the $\mathsf{Setup}(n, t, 1^\lambda)$ algorithm to obtain a public system parameters pp, and run KeyGen algorithm to obtain vector of public key shares $\mathbf{pk} = (\mathsf{pk}_1, \cdots, \mathsf{pk}_n)$ and a vector of master secret key shares $\mathbf{msk} = (\mathsf{msk}_1, \cdots, \mathsf{msk}_n)$. It gives pp and $\mathbf{pk}$ to the adversary $\mathcal{A}$ and keeps $\mathbf{msk}$ to itself. Initialize sets $\mathcal{H} := \{1, \cdots, n\}, \mathcal{CR} := \emptyset$.
- $\mathsf{Query\ Phase}$ The adversary $\mathcal{A}$ adaptively makes the following kinds of queries:
  - Corruption query: $\mathcal{A}$ choose $i \in \{1, \cdots, n\}$ and obtains msk_i. No more than $t - 1$ master secret key shares can be obtained by $\mathcal{A}$ in the whole game.
  - Signature query: The adversary $\mathcal{A}$ adaptively requests i-th signature query, where $i \in \{1, \cdots, n\}$. The challenger responds with the corresponding signature σ, which it generates by running the PSign algorithm.
- $\mathsf{Forgery}$ There are two cases to be considered: (i). $\mathsf{Partial\ Forgery}$: The adversary $\mathcal{A}$ submits $(i^\star, m^\star, \sigma^\star)$. The challenger checks for $i^\star \in \mathcal{H}$, the partial verification algorithm $\mathsf{PVer}(\mathsf{pk}_{i^\star}, m^\star, \sigma^\star_{i^\star}) \neq 1$, where $m^\star$ was never queried. If all checks pass, output 1. (ii). $\mathsf{Threshold\ Forgery}$: The adversary $\mathcal{A}$ submits $(m^\star, \{\sigma_i\}_{i \in S})$. The challenger checks $\mathsf{PVer}(\mathsf{pk}_i, m^\star, \sigma_i) = 1$ for all $i \in S$, where $m^\star$ was never fully signed via $\mathsf{Combine}$. If checks pass, compute $\sigma \leftarrow \mathsf{Combine}(m^\star, \{\sigma_i\}_{i \in S})$ and verify $\mathsf{Ver}(\mathsf{pp}, m^\star, \sigma) \neq 1$. If verification passes, output 1; else output 0.

Intuitively, the robustness property ensures that the protocol behaves as expected for honest parties, even in the presence of an adaptive adversary that corrupts up to t parties. More precisely, it says that: (i) PVer should always accept honestly generated partial signatures; and (ii) if we combine t valid partial signatures (accepted by PVer) using the $\mathsf{Combine}$ algorithm, the output of $\mathsf{Combine}$ should be accepted by Ver, except with a negligible probability. The latter requirement ensures that maliciously generated partial signatures cannot prevent an honest aggregator from efficiently computing a threshold signature except with a negligible probability.

Definition 7 (Robustness Under Chosen Message Attack, RB-CMA). *Let* $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PSign}, \mathsf{PVer}, \mathsf{Combine}, \mathsf{Ver})$ *be a* (n, t)*-threshold signature scheme. Consider the game* $\mathsf{RB\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}$ *defined above. We say that the scheme TS is* $\mathsf{RB\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}$ *secure, if for all PPT adversaries* $\mathcal{A}$*, the following advantage is negligible, i.e.*

$$\mathsf{Adv}^{\mathsf{RB\text{-}CMA}}_{\mathcal{A},\mathsf{TS}}(\lambda) := \Pr[\mathsf{RB\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}(\lambda) = 1] = \mathsf{negl}(\lambda).$$

3 Threshold IBE in Composite-Order Groups

We now present the construction of threshold IBE in composite-order bilinear groups [9,38], following the dual system groups framework of Chen and Wee [18]. Let $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, g, h, e) \leftarrow \mathcal{G}(1^\lambda)$ be pairing groups of composite-order N and bilinear pairing operation $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$.

3.1 Construction of Fully Adaptive Threshold IBE

$\mathsf{Setup}(1^\lambda)$: Run $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, e) \leftarrow \mathcal{G}(1^\lambda)$ to obtain the composite-order groups description and generators $g \in \mathbb{G}, g_1 \in \mathbb{G}_{p_1}, g_2 \in \mathbb{G}_{p_2}, h \in \mathbb{H}, h_1 \in \mathbb{H}_{p_1}, h_2 \in \mathbb{H}_{p_2}$. Define the hash function $\mathsf{H} : \mathbb{G}_T \to \mathbb{G}_{p_2}$. Then, randomly picks $\boldsymbol{w} = (a, b) \leftarrow_R \mathbb{Z}_N^2$ and $\alpha \in \mathbb{Z}_N$. Let public key be $\mathsf{pk} = e(g_1, h_1)^\alpha$. Output public parameters $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, g, h, g_1, h_1, \mathsf{H})$ and

$$\mathsf{mpk} = (g_1^a, g_1^b, h_1^a, h_1^b, \mathsf{pk}) \quad \mathsf{msk} = (\alpha, \boldsymbol{w})$$

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk})$: Sample a uniformly random polynomial $\alpha(x) = \alpha + \sum_{i=1}^{t-1} a_i x^i$ of degree $t-1$ with coefficient $a_i \in \mathbb{Z}_N$. We define α_i as the shares of $\alpha(x)$ and constant term $\alpha_0 = \alpha$. For each authority $i \in [n]$, its public key shares and master secret key shares pair $(i, \mathsf{pk}_i, \mathsf{msk}_i)_{i\in[n]}$ are $\mathsf{pk}_i = e(g_1, h_1)^{\alpha_i}, \mathsf{msk}_i = (h^{\alpha_i}, a, b)$, respectively.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_x, m)$: On input the identity $\mathsf{id}_x \in \mathcal{X}$ and message $m \in \mathbb{G}_{p_2}$. The encryption algorithm chooses $s \in \mathbb{Z}_N$ randomly and produces the ciphertext $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2)$ as:

$$\mathsf{c}_0 = \mathsf{H}(e(g_1, h_1)^{\alpha \cdot s}) \cdot m, \quad \mathsf{c}_1 = g_1^s, \quad \mathsf{c}_2 = g_1^{s(a\mathsf{id}_x + b)}.$$

$\mathsf{ValidateCT}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_x, \mathsf{ct})$: Parse $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2)$ and verify the equation

$$e(\mathsf{c}_2, h_1) = e(\mathsf{c}_1, h_1^{a\mathsf{id}_x + b}).$$

$\mathsf{ShareKeyGen}(\mathsf{pp}, \mathsf{mpk}, i, \mathsf{msk}_i, \mathsf{id}_y)$: On input the identity $\mathsf{id}_y \in \mathcal{Y}$ and master secret key shares msk_i for each authority i, the secret key $(i, \mathsf{sk}_{i,\mathsf{id}_y})$

$$\mathsf{sk}_{i,\mathsf{id}_y} = (\mathsf{k}_{1,i}, \mathsf{k}_{2,i}) = \left(h^{\alpha_i} \cdot h_1^{r_i(a\mathsf{id}_y + b)}, \quad h_1^{r_i} \right),$$

where $r_i \leftarrow_R \mathbb{Z}_N$ is randomly selected by authority i.

$\mathsf{ShareVerify}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_y, \mathsf{pk}_i, \mathsf{sk}_{i,\mathsf{id}_y})$: On input the secret share $\mathsf{sk}_{i,\mathsf{id}_y}$ for each authority i, the public key shares pk_i, and identity id_y. Then, verify the following equations:

$$e\left(g_1, \mathsf{k}_{1,i}\right) = \mathsf{pk}_i \cdot e(g_1^{a\mathsf{id}_y + b}, \mathsf{k}_{2,i}).$$

$\mathsf{Combine}(\mathsf{pp}, \mathsf{id}_y, \{\mathsf{sk}_{i,\mathsf{id}_y}\}_{i\in S})$: On input the identity id_y and using a set of shares $\{\mathsf{sk}_{i,\mathsf{id}_y}\}_{i\in S}$ from a subset of authorities $S \subset [n]$ with $|S| \geq t$, aggregate them into one $\mathsf{sk}_{\mathsf{id}_y} = (\mathsf{k}_1, \mathsf{k}_2)$ as follows.

$$\mathsf{k}_1 = \prod_{i \in S} \mathsf{k}_{1,i}^{L_i} = h^{\alpha} \cdot h_1^{r(a\mathsf{id}_y + b)}, \quad \mathsf{k}_2 = \prod_{i \in S} \mathsf{k}_{2,i}^{L_i} = h_1^{r}.$$

where, $L_i = \prod_{i \in S} \frac{j}{j-i}$ is the Lagrange interpolation for the authority i and we define $r = \sum_{i \in S} L_i \cdot r_i$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}, \mathsf{sk}_{\mathsf{id}_y}, \mathsf{ct})$: If the identities id's of the ciphertext and secret key are equal, the decryption algorithm computes the blinding factor as:

$$\frac{e(\mathsf{c}_1, \mathsf{k}_1)}{e(\mathsf{c}_2, \mathsf{k}_2)} = \frac{e(g_1^s, h^{\alpha} \cdot h_1^{r(a\mathsf{id}_y + b)})}{e(g_1^{s(a\mathsf{id}_x + b)}, h_1^r)} = e(g_1, h_1)^{\alpha s}. \tag{3.1}$$

Due to the orthogonality of the subgroups in composite-order groups and the fact that $h \in \mathbb{H}, g_1 \in \mathbb{G}_{p_1}$, the above Eq. 3.1 holds.

3.2 Proof of security

Theorem 1 (Full security with adaptive corruptions). *The threshold IBE scheme is fully secure against adaptive corruptions, assuming that Assumptions 2 and 3 all hold. More precisely, for any adversary $\mathcal{A}$ that makes at most q key queries against the TIBE scheme, there exist adversaries $\mathcal{A}_1, \mathcal{A}_1, \mathcal{A}_3$ such that*

$$\mathsf{Adv}^{\mathsf{TIBE}}_{\mathsf{composite}}(1^\lambda) \leq \mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{A}_1}(1^\lambda) + q \cdot \mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{A}_2}(1^\lambda) + q \cdot \mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{A}_3}(1^\lambda) + 2^{-\Omega(\lambda)}.$$

The proof proceeds via a sequence of games, following the pairing encoding framework of Wee [47] and the dual system encryption methodology of Waters [46]. In the dual system paradigm, ciphertexts and secret keys can be either *normal* or *semi-functional* (SF). Normal objects are used in the real system, while SF objects are introduced exclusively within the proof to facilitate the security reduction.

Hybrid Transition Strategy. Our proof relies on a hybrid argument over the identities queried by the adversary. Without loss of generality, we assume the adversary queries secret key shares for at most q distinct identities, denoted as $\mathsf{id}_1, \ldots, \mathsf{id}_q$, and subsequently submits a challenge identity $\mathsf{id}^\star$ such that $\mathsf{id}^\star \notin \{\mathsf{id}_1, \ldots, \mathsf{id}_q\}$ (which is distinct from all queried identities). We transform the secret key shares associated with these q distinct identities from normal to SF form, following the order in which the identities first appear. Specifically, for any identity id queried to different authorities, we respond with the same type of secret key share (either normal or SF) corresponding to the phase of that identity. The type of response to a secret key query for an identity id depends solely on the order of id among the distinct queries, and is independent of the specific authority being queried.

Simulation of Corruption. For corruption queries on an authority i, we always return the normal master secret key share msk_i. Consequently, the adversary can only generate (pseudo-)normal secret key shares for any identity id using the obtained msk_i. Due to the indistinguishability of the key distributions in the

dual system groups, the adversary cannot distinguish these shares from the SF secret key shares returned by the simulator for other identities.

Distributions of SF Objects. To instantiate this strategy, we define the forms of SF master secret key shares, normal/SF ciphertexts, and normal/pseudo-normal/pseudo-SF/SF secret key shares. Let $(\mathsf{pp}, \mathsf{pk}_i, \mathsf{msk}_i, \alpha_i, \boldsymbol{w} = (a, b))$ be sampled as in the Setup algorithm. The distributions of these objects follow the definitions in Wee [47]. Specifically, normal ciphertexts and secret key shares correspond exactly to those in our real scheme. It is important to note that normal ciphertexts, SF ciphertexts, and (pseudo-)normal secret key shares are generated using the normal master secret key msk_i, whereas (pseudo-)SF secret key shares are generated using the semi-functional master secret key shares $\widehat{\mathsf{msk}_i}$.

- SF master secret key shares : Picks randomly $\alpha_i' \in \mathbb{Z}_N$ and forms the SF master secret key shares as: $\widehat{\mathsf{msk}_i} = \left(h^{\alpha_i} h_2^{\alpha_i'}, a, b\right)$.
- normal ciphertext : $\mathsf{ct} = \left(\mathsf{H}(e(g_1, h_1)^{\alpha s}) \cdot m, \quad g_1^s, \quad g_1^{s(a\mathsf{id}_x + b)}\right)$.
- SF ciphertext : $\mathsf{ct} = \left(\mathsf{H}(e(g, h)^{\alpha s}) \cdot m, \quad g^s, \quad g^{s(a\mathsf{id}_x + b)}\right)$.
- normal secret key shares : $\mathsf{sk}_{i,\mathsf{id}_y} = \left(h^{\alpha_i} \cdot h_1^{r_i(a\mathsf{id}_y + b)}, \quad h_1^{r_i}\right)$.
- pseudo-normal secret key shares : $\mathsf{sk}_{i,\mathsf{id}_y} = \left(h^{\alpha_i} \cdot h^{r_i(a\mathsf{id}_y + b)}, \quad h^{r_i}\right)$.
- pseudo-SF secret key shares : $\mathsf{sk}_{i,\mathsf{id}_y} = \left(h^{\alpha_i} h_2^{\alpha_i'} \cdot h^{r_i(a\mathsf{id}_y + b)}, \quad h^{r_i}\right)$.
- SF secret key shares : $\mathsf{sk}_{i,\mathsf{id}_y} = (h^{\alpha_i} h_2^{\alpha_i'} \cdot h_1^{r_i(a\mathsf{id}_y + b)}, \quad h_1^{r_i})$.

Observe that all types of secret keys can decrypt a normal ciphertext. In addition, only normal and pseudo-normal secret keys can decrypt an SF ciphertext, whereas pseudo-SF and SF keys cannot. The latter is consistent with the fact that we exploit $\boldsymbol{w}$-hiding and the $\mathsf{id}_x \neq \mathsf{id}_y$ when we switch from pseudo-normal to pseudo-SF keys, which is precisely why we lose decryption capabilities in the $\mathbb{G}_{p_2}$ subgroup components.

Proof. We present a series of games. We write Adv_{xx} to denote the advantage of $\mathcal{A}$ in Game_{xx}.

Game_0: is the real security game.
Game_1: is the same as Game_0 except that the challenge ciphertext is semi-functional. We also modify the distribution of $\mathbb{G}_T$ accordingly.
$\mathsf{Game}_{2,\kappa,0}$: is identical to Game_1.
$\mathsf{Game}_{2,\kappa}$ for $\kappa = 1, 2, \cdots, q$: is the same as Game_1, except the first $\kappa - 1$ keys are semi-functional, and the last $q - \kappa$ keys are normal. There are 4 sub-games, where the κ-th secret key transitions from normal in $\mathsf{Game}_{2,\kappa,0}$ to pseudo-normal in $\mathsf{Game}_{2,\kappa,1}$, to pseudo-SF in $\mathsf{Game}_{2,\kappa,2}$, to SF in $\mathsf{Game}_{2,\kappa,3}$.
Game_3: is the same as $\mathsf{Game}_{2,q,3}$, except that $\kappa_0 \leftarrow_R \{0,1\}^\lambda$.

In Game_3, the view of the adversary $\mathcal{A}$ is statistically independent of the challenge bit b. Hence, $\mathsf{Adv}_3 = 0$. We complete the proof by establishing the following sequence of lemmas.

Lemma 1 (normal to SF ciphertexts). *There exists $\mathcal{A}_1$ whose running time is roughly that of $\mathcal{A}$ such that*

$$|\mathsf{Adv}_0 - \mathsf{Adv}_1| \leq \mathsf{Adv}^{\mathsf{DS1}}_{\mathcal{G},\mathcal{A}_1}(1^\lambda).$$

Proof. We will rely on Assumption 2. On input $D = (\mathbb{G}, g_1, g)$ and distribution $T \in \{T_0, T_1\}$ where $T_0 = g_1 \in \mathbb{G}_{p_1}, T_1 = g \in \mathbb{G}$, the adversary $\mathcal{A}_1$ simulates $\mathcal{A}$ as follows:

- **Setup**. Sample $\alpha(x), \mathsf{H}, \boldsymbol{w} = (a, b)$ as in Setup, and output the public parameter $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g_1, h_1, g, h, g_1^a, g_1^b, h_1^a, h_1^b)$. For each authority, set $\mathsf{pk} = e(T,h)^\alpha, \mathsf{msk}_i = (h^{\alpha_i}, \boldsymbol{w})$, and $\mathsf{pk}_i = e(T,h)^{\alpha_i}$.
- **Ciphertexts**. Computes the ciphertexts with T and a randomly selected $s \leftarrow \mathbb{Z}_N$ outputs $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2) = \left(\mathsf{H}(e(T,h)^{\alpha s}) \cdot m, \quad T^s, \quad T^{s(a\mathsf{id}_x + b)}\right)$.
- **Key Queries**. On input the j-th secret key queries id_{y_j}, for each authority $i \in [n]$, output the secret key $\mathsf{sk}_{i,\mathsf{id}_{y_j}}$. Randomly select a randomness r_i for every secret key query id_{y_j},

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = (\mathsf{k}_{1,i}, \mathsf{k}_{2,i}) = \left(h^{\alpha_i} \cdot h_1^{r_i(a\mathsf{id}_{y_j} + b)}, h_1^{r_i}\right).$$

- **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = (h^{\alpha_i}, \boldsymbol{w})$.
- **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

Observe that when $T = T_0 = g_1$, the output is identical to that in Game_0, and when $T = T_1 = g$, the output is identical to that in Game_1.

Lemma 2 (normal to pseudo-normal secret key shares). *There exists $\mathcal{A}_2$ whose running time is roughly that of $\mathcal{A}$ such that*

$$|\mathsf{Adv}_{2,\kappa,1} - \mathsf{Adv}_{2,\kappa,0}| \leq \mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{G},\mathcal{A}_2}(1^\lambda).$$

Proof. We will rely on Assumption 3. On input $D = (\mathbb{H}, h_1, h)$ and distribution $T \in \{T_0, T_1\}$ where $T_0 = h_1 \in \mathbb{H}_{p_1}, T_1 = h \in \mathbb{H}$, the adversary $\mathcal{A}_2$ simulates $\mathcal{A}$ as follows:

- **Setup**. Sample $\alpha(x), \mathsf{H}, \boldsymbol{w} = (a, b)$ as in Setup, and output the public parameter $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g_1, h_1, g, h, g_1^a, g_1^b, h_1^a, h_1^b)$. For each authority, set $\mathsf{pk} = e(g,h)^\alpha, \mathsf{msk}_i = (h^{\alpha_i}, \boldsymbol{w})$, and $\mathsf{pk}_i = e(g,h)^{\alpha_i}$.
- **Ciphertexts**. Computes the SF ciphertexts with a randomly selected $s \leftarrow \mathbb{Z}_N$ as: $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2) = \left(\mathsf{H}(e(g,h)^{\alpha s}) \cdot m, \quad g^s, \quad g^{s(a\mathsf{id}_x + b)}\right)$.
- **Key Queries**. On input the j-th key queries id_{y_j}, for each authority $i \in [n]$, outputs $\mathsf{sk}_{i,\mathsf{id}_{y_j}}$ as follows:

  If $j = \kappa$, output $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} \cdot T^{r_i(a\mathsf{id}_{y_j} + b)}, \quad T^{r_i}\right)$.

  If $j > \kappa$, output $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} \cdot h_1^{r_i(a\mathsf{id}_{y_j} + b)}, \quad h_1^{r_i}\right)$.

  If $j < \kappa$, output $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} h_2^{\alpha'_i} \cdot h_1^{r_i(a\mathsf{id}_{y_j} + b)}, \quad h_1^{r_i}\right)$.

– **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = (h^{\alpha_i}, \boldsymbol{w})$.
– **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

Observe that when $T = T_0 = h_1$,the output is identical to that in $\mathsf{Game}_{2,\kappa,0}$, and when $T = T_1 = h$, the output is identical to that in $\mathsf{Game}_{2,\kappa,1}$.

Lemma 3 (pseudo-normal to pseudo-SF secret key shares). *There exists $\mathcal{A}_3$ whose running time is roughly that of $\mathcal{A}$ such that for $\kappa = 1, 2, \cdots, q$*

$$|\mathsf{Adv}_{2,\kappa,2} - \mathsf{Adv}_{2,\kappa,1}| = 0.$$

Proof. We rely on the α-privacy and $\boldsymbol{w}$-hiding [47]. The only difference between $\mathsf{Game}_{2,\kappa,1}$ and $\mathsf{Game}_{2,\kappa,2}$ lies in the distribution of $\mathsf{sk}_{\mathsf{id}_y}$, which we sample using msk_i and $\widehat{\mathsf{msk}_i}$ respectively.

– **Setup**. Sample $\alpha(x), \mathsf{H}, \boldsymbol{w}$ as in Setup, set $\widehat{\mathsf{msk}_i} = \left(h^{\alpha_i} h_2^{\alpha_i'}, \boldsymbol{w}\right), \boldsymbol{w} = (a, b)$ and output $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g_1, h_1, g, h, g_1^a, g_1^b, h_1^a, h_1^b), \mathsf{pk}_i = (e(g,h)^{\alpha_i}, g^a, g^b)$.
– **Ciphertexts**. Computes the SF ciphertexts with a randomly selected $s \leftarrow \mathbb{Z}_N$ as: $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2) = \left(\mathsf{H}(e(g,h)^{\alpha s}) \cdot m, \quad g^s, \quad g^{s(a\mathsf{id}_x + b)}\right)$.
– **Key Queries**. On input the j-th key queries id_{y_j}, for each authority $i \in [n]$, output the secret key $\mathsf{sk}_{i,\mathsf{id}_{y_j}}$. We consider the condition $j \neq \kappa$. Observe that for all $j \neq \kappa$, the $\mathbb{H}_{p_2}$-component of $\mathsf{sk}_{\mathsf{id}_{y_j}}$ is given by:

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} h_2^{\alpha_i'} \cdot h^{r_i(a\mathsf{id}_{y_j} + b)}, h^{r_i}\right).$$

If $j = \kappa$, assuming the challenge identity is $\mathsf{id}_{y_j}^\star$, we use the SF master secret key shares $\widehat{\mathsf{msk}_i} = \left(h^{\alpha_i} h_2^{\alpha_i'}, a, b\right)$ to simulate the secret key shares

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} h_2^{\alpha_i'} \cdot h^{r_i(a\mathsf{id}_{y_j}^\star + b)}, h^{r_i}\right).$$

According to α-privacy, $\boldsymbol{w}$-hiding, and $\mathsf{id}_x \neq \mathsf{id}_{y_j}^\star$, the secret key shares

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} \cdot h^{r_i(a\mathsf{id}_{y_j}^\star + b)}, h^{r_i}\right). \tag{3.2}$$

and

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} h_2^{\alpha_i'} \cdot h^{r_i(a\mathsf{id}_{y_j}^\star + b)}, h^{r_i}\right). \tag{3.3}$$

is statistically indistinguishable. If the id_x of the ciphertext equals $\mathsf{id}_{y_j}^\star$ of secret keys, according to secret sharing, the α's privacy can also be proved.
– **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = (h^{\alpha_i}, \boldsymbol{w})$.
– **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

They are identical from the viewpoint of the adversary. The pseudo-normal secret key shares in Eq. 3.2 is in $\mathsf{Game}_{2,\kappa,1}$ and the pseudo-SF secret key shares in Eq. 3.3 is in $\mathsf{Game}_{2,\kappa,2}$.

Lemma 4 (pseudo-SF to SF secret key shares). *There exists $\mathcal{A}_3$ whose running time is roughly that of $\mathcal{A}$ such that for $\kappa = 1, 2, \cdots, q$*

$$|\mathsf{Adv}_{2,\kappa,3} - \mathsf{Adv}_{2,\kappa,2}| \leq \mathsf{Adv}^{\mathsf{DS2}}_{\mathcal{G},\mathcal{A}_2}(1^\lambda).$$

Proof. We will rely on Assumption 3. The proof is completely analogous to $\mathsf{Game}_{2,\kappa,1}$, except $\mathcal{A}_3$ uses $\widehat{\mathsf{msk}_i}$ instead of msk_i to sample $\mathsf{sk}_{\mathsf{id}_y}$. On input the distribution $T = (T_0, T_1)$ where $T_0 \leftarrow_R \mathbb{H}, T_1 \leftarrow_R \mathbb{H}_{p_1}$, the adversary $\mathcal{A}_3$ simulates $\mathcal{A}$ as follows:

- **Setup**. Sample $\alpha(x), \mathsf{H}, \boldsymbol{w}$ as in Setup, set $\widehat{\mathsf{msk}_i} = \left(h^{\alpha_i} h_2^{\alpha_i'}, \boldsymbol{w}\right), \quad \boldsymbol{w} = (a, b)$ and output $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g_1, h_1, g, h, g_1^a, g_1^b, h_1^a, h_1^b), \mathsf{pk}_i = (e(g,h)^{\alpha_i}, g^a, g^b)$.
- **Ciphertexts**. Computes the SF ciphertexts and outputs

$$\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2) = \left(\mathsf{H}(e(g,h)^{\alpha s}) \cdot m, \quad g^s, \quad g^{s(a\mathsf{id}_x + b)}\right).$$

- **Key Queries**. On input the j-th key queries id_{y_j}, for each authority $i \in [n]$, sk_{i,y_j} as follows:
  If $j = \kappa$, output $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} h_2^{\alpha_i'} \cdot T^{r_i(a\mathsf{id}_{y_j}+b)}, T^{r_i}\right)$.
  If $j < \kappa$, output $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} h_2^{\alpha_i'} \cdot h_1^{r_i(a\mathsf{id}_{y_j}+b)}, h_1^{r_i}\right)$.
  If $j > \kappa$, output $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left(h^{\alpha_i} \cdot h_1^{r_i(a\mathsf{id}_{y_j}+b)}, h_1^{r_i}\right)$.
- **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = (h^{\alpha_i}, \boldsymbol{w})$.
- **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

Observe that when $T = T_0 = h$, the output is identical to that in $\mathsf{Game}_{2,\kappa,2}$, and when $T = T_1 = h_1$, the output is identical to that in $\mathsf{Game}_{2,\kappa,3}$.

Lemma 5 (final transition). $|\mathsf{Adv}_{2,q,3} - \mathsf{Adv}_3| \leq 2^{-\Omega(\lambda)}$.

Proof. In $\mathsf{Game}_{2,q,3}$, all the secret key shares are semi-functional, which means they leak no information whatsoever about α. For the (SF) challenge ciphertext, $\kappa_0 = \mathsf{H}(e(g, h_1^\alpha) \cdot e(g, h_2^\alpha))$ by the left-over hash lemma is $2^{-\Omega(\lambda)}$-close to the uniform distribution over $\{0,1\}^\lambda$.

Consistency of key generation. To complete the proof of Theorem 1, it remains to prove the consistency of key generation, which is defined by the Sec. 2.4. We argue that for any adversary $\mathcal{A}$, we have $\mathsf{Adv\text{-}CD}^{\mathsf{TIBE}}_{\mathcal{A}}(\lambda) = 0$. To see this, observe that the two tests performed during decryption ensure that $\mathsf{Dec}(\mathsf{pp}, \mathsf{id}, \mathsf{sk}_{\mathsf{id}_y}, \mathsf{ct})$ outputs the same value for all reconstituted keys $\mathsf{sk}_{\mathsf{id}_y}$ that pass the tests. Furthermore, conditions (1)–(4) needed for the adversary to win the consistency of key generation game ensure that both tests succeed. Hence, the decryption algorithm will output the same value no matter which key $\mathsf{sk}_{\mathsf{id}_y}$ is given as input, and thus $\mathsf{Adv\text{-}CD}^{\mathsf{TIBE}}_{\mathcal{A}}(\lambda) = 0$.

4 Threshold IBE in Prime-Order Groups

We present our threshold IBE scheme in prime-order groups under the k-Linear assumption, following the dual system group encodings in [16].

4.1 Construction

$\mathsf{Setup}(1^\lambda)$: We define the hash function $\mathsf{H} : \mathbb{G}_T \to \mathbb{G}_p^{k+1}$, which maps the pairing element $e(g,h)$ to elements in $\mathbb{G}_p^{k+1}$. Concretely, for the pairing element $e(g,h) \in \mathbb{G}_T$, we define $\mathsf{H}(e(g,h)) = [\mathbf{e}]_1$, where $[\mathbf{e}]_1 \leftarrow_R \mathbb{G}_p^{k\times 1}$. Samples $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, along with $\mathbf{W}_1, \mathbf{W}_2 \leftarrow_R \mathbb{Z}_p^{(k+1)\times(k+1)}$, where $\mathbf{a}^{\perp\top}\mathbf{A} = \mathbf{b}^{\perp\top}\mathbf{B} = \mathbf{0}$ and $\mathbf{a}^{\perp\top}\mathbf{b}^\perp \neq 0$. Picks randomly $\boldsymbol{\alpha} \in \mathbb{Z}_p^k$ and public key $\mathsf{pk} = e([\mathbf{A}]_1, [\boldsymbol{\alpha}]_2)$. Then, output public parameters $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, [\mathbf{A}]_1, [\mathbf{B}]_2, \mathsf{H})$ and

$$\mathsf{mpk} = \left([\mathbf{W}_1^\top \mathbf{A}]_1, [\mathbf{W}_2^\top \mathbf{A}]_1, [\mathbf{W}_1\mathbf{B}]_2, [\mathbf{W}_2\mathbf{B}]_2, \mathsf{pk}\right), \mathsf{msk} = (\boldsymbol{\alpha}, \mathbf{W}_1, \mathbf{W}_2).$$

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk})$: Picks randomly $\boldsymbol{\alpha}(\mathbf{x}) = \boldsymbol{\alpha} + \sum_{i=1}^{t-1} a_i \mathbf{x}^i$ of degree $t-1$ with coefficient $a_i \in \mathbb{Z}_N$. We define $\boldsymbol{\alpha}_i$ as the shares of $\boldsymbol{\alpha}(\mathbf{x})$ and constant term $\boldsymbol{\alpha}_0 = \boldsymbol{\alpha}$. For each authority $i \in [n]$, its public and secret key shares pair $(i, \mathsf{pk}_i, \mathsf{msk}_i)$ is: $\mathsf{pk}_i = e([\mathbf{A}]_1, [\boldsymbol{\alpha}_i]_2), \quad \mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_x, \boldsymbol{m})$: On input the identity $\mathsf{id}_x \in \mathcal{X}$ and $\boldsymbol{m} \in \mathbb{G}_{p_2}^{k+1}$. The encryption algorithm chooses $\mathbf{s} \in \mathbb{Z}_p^k$ randomly and outputs the ciphertext $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2)$ as:

$$\mathsf{c}_0 = \mathsf{H}\left(e\left([\mathbf{As}]_1, [\boldsymbol{\alpha}]_2\right)\right) \cdot \boldsymbol{m}, \quad \mathsf{c}_1 = [\mathbf{As}]_1, \quad \mathsf{c}_2 = [(\mathbf{W}_1^\top \cdot \mathsf{id}_x + \mathbf{W}_2^\top)\mathbf{As}]_1.$$

$\mathsf{ValidateCT}(\mathsf{pp}, \mathsf{mpk}, \mathsf{ct}, \mathsf{id}_x)$: Parse $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2)$ and verify the equation

$$e([\mathbf{B}]_2, \mathsf{c}_2) = e(\mathsf{c}_1, [\mathbf{W}_1\mathbf{B} \cdot \mathsf{id}_x + \mathbf{W}_2\mathbf{B}]_2).$$

$\mathsf{ShareKeyGen}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_y, i, \mathsf{msk}_i)$: On input the identity $\mathsf{id}_y \in \mathcal{Y}$ and master secret key shares msk_i for each authority $i \in [n]$, the secret key $(i, \mathsf{sk}_{i,\mathsf{id}_y})$ is formed as:

$$\mathsf{sk}_{i,\mathsf{id}_y} = (\mathsf{k}_{1,i}, \mathsf{k}_{2,i}) = ([\boldsymbol{\alpha}_i]_2[(\mathbf{W}_1 \cdot \mathsf{id}_y + \mathbf{W}_2)\mathbf{Br}_i]_2, [\mathbf{Br}_i]_2),$$

where $\mathbf{r}_i \leftarrow_R \mathbb{Z}_p^k$ is randomly selected by authority i.

$\mathsf{ShareVerify}(\mathsf{pp}, \mathsf{mpk}, \mathsf{sk}_{i,\mathsf{id}_y}, \mathsf{pk}_i, \mathsf{id}_y)$: On input the secret key share $\mathsf{sk}_{i,\mathsf{id}_y}$ for each authority i, the public key shares pk_i, and identity id_y. Then, verify the following equations:

$$e\left(\mathsf{k}_{1,i}, [\mathbf{A}]_1\right) = \mathsf{pk}_i \cdot e([\mathbf{W}_1^\top \mathbf{A} \cdot \mathsf{id}_y + \mathbf{W}_2^\top \mathbf{A}]_1, \mathsf{k}_{2,i}).$$

$\mathsf{Combine}(\mathsf{pp}, \mathsf{id}_y, \{\mathsf{sk}_{i,\mathsf{id}_y}\}_{i\in S})$: On input the identity id_y and using a set of shares $\{\mathsf{sk}_{i,\mathsf{id}_y}\}_{i\in S}$ from a subset of authorities $S \subset [n]$ with $|S| \geq t$, aggregate them into one $\mathsf{sk}_{\mathsf{id}_y} = (\mathsf{k}_1, \mathsf{k}_2)$ as follows.

$$\mathsf{k}_1 = \prod_{i\in S} \mathsf{k}_{1,i}^{L_i} = [\boldsymbol{\alpha}]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_y + \mathbf{W}_2)\mathbf{Br}]_2, \quad \mathsf{k}_2 = \prod_{i\in S} \mathsf{k}_{2,i}^{L_i} = [\mathbf{Br}]_2,$$

where, $L_i = \prod_{i \in S} \frac{j}{j-i}$ is the Lagrange interpolation for the authority i and we define $r = \sum_{i \in S} L_i \cdot \mathbf{r}_i$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}, \mathsf{sk}_{\mathsf{id}_y}, \mathsf{ct})$: If the identities id's of the ciphertext and secret key are equal, the decryption algorithm computes the blinding factor as:

$$e\left([\mathbf{As}]_1, [\boldsymbol{\alpha}]_2\right) = \frac{e(\mathsf{c}_1, \mathsf{k}_1)}{e(\mathsf{c}_2, \mathsf{k}_2)} = \frac{e([\mathbf{As}]_1, [\boldsymbol{\alpha}]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_y + \mathbf{W}_2)\mathbf{Br}]_2)}{e([(\mathbf{W}_1^\top \cdot \mathsf{id}_x + \mathbf{W}_2^\top)\mathbf{As}]_1, [\mathbf{Br}]_2)}.$$

4.2 Proof of security

Theorem 2 (Full security with adaptive corruptions). *Under the k-Linear assumptions 1, the TIBE scheme in Sec. 4.1 is adaptively secure. More precisely, for any adversary $\mathcal{A}$ that makes at most q key queries against the TIBE scheme, there exist adversaries $\mathcal{A}_1, \mathcal{A}_1, \mathcal{A}_3$ such that*

$$\mathsf{Adv}^{\mathsf{TIBE}}_{\mathsf{prime}} \leq \mathsf{Adv}^{\mathsf{k\text{-}Lin}}_{\mathcal{A}_1} + q \cdot \mathsf{Adv}^{\mathsf{k\text{-}Lin}}_{\mathcal{A}_2} + q \cdot \mathsf{Adv}^{\mathsf{k\text{-}Lin}}_{\mathcal{A}_3} + 2^{-\Omega(\lambda)}.$$

Proof. This proof is analogous to the TIBE in composite-order groups, except that the generic transformation on the technique of [16]. The proof follows via a series of games. We first form SF master secret key shares, the normal ciphertexts, SF ciphertexts, normal secret key shares, pseudo-normal secret key shares, pseudo-SF secret key shares, and SF secret key shares as follows:

- SF master secret key shares. Picks $\mathbf{a}^\perp \in \mathbb{Z}_p^{k+1}, \alpha \leftarrow_R \mathbb{Z}_p$ and forms the SF master secret key shares as: $\widehat{\mathsf{msk}_i} = \left([\boldsymbol{\alpha}_i + \mathbf{a}^\perp \alpha]_2, \mathbf{W}_1, \mathbf{W}_2\right)$.
- normal ciphertext. The normal ciphertext $\mathsf{ct} = (\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2)$ is as follows:

$$\mathsf{c}_0 = \mathsf{H}\left(e\left([\mathbf{As}]_1, [\boldsymbol{\alpha}]_2\right)\right) \cdot \boldsymbol{m}, \quad \mathsf{c}_1 = [\mathbf{As}]_1, \quad \mathsf{c}_2 = [(\mathbf{W}_1^\top \cdot \mathsf{id}_x + \mathbf{W}_2^\top)\mathbf{As}]_1.$$

- SF ciphertext. Randomly picks $\hat{s} \leftarrow_R \mathbb{Z}_p$. The SF ciphertext ct is as follows:

$$\mathsf{c}_1 = [\mathbf{As} + \mathbf{b}^\perp \hat{s}]_1, \quad \mathsf{c}_2 = [(\mathbf{W}_1^\top \cdot \mathsf{id}_x + \mathbf{W}_2^\top)(\mathbf{As} + \mathbf{b}^\perp \hat{s})]_1,$$

$$\mathsf{c}_0 = \mathsf{H}(e([\mathbf{As} + \mathbf{b}^\perp \hat{s}]_1, [\boldsymbol{\alpha}]_2) \cdot \boldsymbol{m}.$$

- normal secret key shares. For each authority i, the normal secret key shares $\mathsf{sk}_{i,\mathsf{id}_y} = (\mathsf{k}_{1,i}, \mathsf{k}_{2,i})$ is as follows:

$$\mathsf{sk}_{i,\mathsf{id}_y} = \left([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_y + \mathbf{W}_2)\mathbf{Br}_i]_2, \quad [\mathbf{Br}_i]_2\right).$$

- pseudo-normal secret key shares. Randomly pick $\hat{r} \leftarrow \mathbb{Z}_p$. The pseudo-normal secret key shares are as follows:

$$\mathsf{sk}_{i,\mathsf{id}_y} = \left([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_y + \mathbf{W}_2)(\mathbf{Br}_i + \mathbf{a}^\perp \hat{r})]_2, \quad [\mathbf{Br}_i + \mathbf{a}^\perp \hat{r}]_2\right).$$

- pseudo-SF secret key shares. The pseudo-SF secret key shares are as follows:

$$\mathsf{sk}_{i,\mathsf{id}_y} = \left([\boldsymbol{\alpha}_i + \mathbf{a}^\perp \alpha]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_y + \mathbf{W}_2)(\mathbf{Br}_i + \mathbf{a}^\perp \hat{r})]_2, \quad [\mathbf{Br}_i + \mathbf{a}^\perp \hat{r}]_2\right).$$

– SF secret key shares. The SF secret key shares are as follows:

$$\mathsf{sk}_{i,\mathsf{id}_y} = \left([\boldsymbol{\alpha}_i + \mathbf{a}^{\perp}\alpha]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_y + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, \quad [\mathbf{B}\mathbf{r}_i]_2\right).$$

Game sequence. We present a series of games. The games are identical to the games in Theorem 1. Thus, we omit it here. We complete the proof by establishing the following sequence of lemmas.

Lemma 6 (normal to SF ciphertexts). *There exists $\mathcal{A}_1$ whose running time is roughly that of $\mathcal{A}$ such that*

$$|\mathsf{Adv}_0 - \mathsf{Adv}_1| \leq \mathsf{Adv}^{\mathsf{k-Lin}}_{\mathcal{G},\mathcal{A}_1}(1^\lambda).$$

Proof. We will rely on Assumption 1. On input $D = (\mathbb{G}, \mathbb{H}, [\mathbf{A}]_1, [\mathbf{B}]_2, [\mathbf{b}^{\perp}]_1, [\boldsymbol{\alpha}_i]_2)$ and the distribution $T = \{T_0, T_1\}$, where $T_0 : \mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s}, \hat{s} = 0; \quad T_1 : \mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s}, \hat{s} \leftarrow_R \mathbb{Z}_p$, the adversary $\mathcal{A}_1$ simulates $\mathcal{A}$ as follows:

– **Setup**. Sample $\boldsymbol{\alpha}(\mathbf{x}), \mathsf{H}, \mathbf{W}_1, \mathbf{W}_2$ as in Setup, set $\mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$ and output $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g, h), \quad \mathsf{pk}_i = e([\mathbf{A}]_1, [\boldsymbol{\alpha}_i]_2)$.
– **Ciphertexts.** Computes the ciphertexts and a randomly selected $\hat{s} \leftarrow \mathbb{Z}_N$ outputs

$$\mathsf{c}_1 = [T]_1, \quad \mathsf{c}_2 = [(\mathbf{W}_1^{\top} \cdot \mathsf{id}_x + \mathbf{W}_2^{\top})T]_1, \quad \mathsf{c}_0 = \mathsf{H}(e([T]_1, [\boldsymbol{\alpha}]_2) \cdot \boldsymbol{m}.$$

– **Key Queries**. On input the j-th key queries id_{y_j}, for each authority $i \in [n]$, output the secret key $\mathsf{sk}_{i,\mathsf{id}_{y_j}}$. Randomly select a randomness $\mathbf{r}_i \leftarrow_R \mathbb{Z}_p^k$ for every key query id_{y_j},

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, \quad [\mathbf{B}\mathbf{r}_i]_2\right).$$

– **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$.
– **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

Observe that when $T_0 : \mathbf{A}\mathbf{s}+\mathbf{b}^{\perp}\hat{s}, \hat{s} = 0$, the output is identical to that in Game_0, and when $T_1 : \mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s}, \hat{s} \leftarrow_R \mathbb{Z}_p^{\star}$, the output is identical to that in Game_1.

Lemma 7 (normal to pseudo-normal secret key shares). *There exists $\mathcal{A}_2$ whose running time is roughly that of $\mathcal{A}$ such that*

$$|\mathsf{Adv}_{2,\kappa,1} - \mathsf{Adv}_{2,\kappa,0}| \leq \mathsf{Adv}^{\mathsf{k-Lin}}_{\mathcal{G},\mathcal{A}_2}(1^\lambda).$$

Proof. We will rely on Assumption 1. On input $D = (\mathbb{G}, \mathbb{H}, [\mathbf{A}]_1, [\mathbf{B}]_2, [\mathbf{b}^{\perp}]_1, [\mathbf{a}^{\perp}]_2, [\boldsymbol{\alpha}_i]_2)$ and the distribution $T = (T_0, T_1)$ where $T_0 : \mathbf{B}\mathbf{r}_i + \mathbf{a}^{\perp}\hat{r}, \hat{r} = 0; T_1 : \mathbf{B}\mathbf{r}_i + \mathbf{a}^{\perp}\hat{r}, \hat{r} \leftarrow_R \mathbb{Z}_p$, the adversary $\mathcal{A}_2$ simulates the adversary $\mathcal{A}$ as follows:

– **Setup**. Sample $\boldsymbol{\alpha}(\mathbf{x}), \mathsf{H}, \mathbf{W}_1, \mathbf{W}_2$ as in Setup, output public parameters $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g, h)$, $\mathsf{pk} = [\boldsymbol{\alpha}^{\top}\mathbf{A}]_T$ and set $\mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$, $\mathsf{pk}_i = e([\mathbf{A}]_1, [\boldsymbol{\alpha}_i]_2)$.

– **Ciphertexts.** Computes the ciphertexts and a randomly selected $\hat{s} \leftarrow \mathbb{Z}_N$ outputs

$$\mathsf{c}_1 = [\mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s}]_1, \quad \mathsf{c}_2 = [(\mathbf{W}_1^{\top} \cdot \mathsf{id}_x + \mathbf{W}_2^{\top})(\mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s})]_1,$$

$$\mathsf{c}_0 = \mathsf{H}(e([\mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s}]_1, [\boldsymbol{\alpha}]_2) \cdot \boldsymbol{m}.$$

– **Key Queries**. On input the j-th key queries id_{y_j}, for each authority $i \in [n]$, output the secret key $\mathsf{sk}_{i,\mathsf{id}_{y_j}}$. Randomly select a randomness $\mathbf{r}_i \leftarrow_R \mathbb{Z}_p^k$ for every key query id_{y_j},
If $j = \kappa$, outputs $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \big([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2) \cdot T]_2, \quad [T]_2\big)$.
If $j > \kappa$, outputs $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \big([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2)(\mathbf{B}\mathbf{r}_i)]_2, \quad [\mathbf{B}\mathbf{r}_i]_2\big)$.
If $j < \kappa$, outputs $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \big([\boldsymbol{\alpha}_i + \mathbf{a}^{\perp}\alpha]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, \quad [\mathbf{B}\mathbf{r}_i]_2\big)$.
– **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$.
– **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

Observe that when $T = T_0 : \mathbf{B}\mathbf{r}_i + \mathbf{a}^{\perp}\hat{r}, \hat{r} = 0$, the output is identical to that in $\mathsf{Game}_{2,\kappa,0}$, and when $T = T_1 : \mathbf{B}\mathbf{r}_i + \mathbf{a}^{\perp}\hat{r}, \hat{r} \leftarrow_R \mathbb{Z}_p^{\star}$, the output is identical to that in $\mathsf{Game}_{2.\kappa.1}$.

Lemma 8 (pseudo-normal to pseudo-SF secret sahred keys). *There exists $\mathcal{A}_3$ whose running time is roughly that of $\mathcal{A}$ such that for $\kappa = 1, 2, \cdots, q$*

$$|\mathsf{Adv}_{2,\kappa,2} - \mathsf{Adv}_{2,\kappa,1}| = 0.$$

Proof. We rely on the α-privacy and $\boldsymbol{w}$-hiding. The only difference between $\mathsf{Game}_{2,\kappa,1}$ and $\mathsf{Game}_{2,\kappa,2}$ lies in the distribution of $\mathsf{sk}_{\mathsf{id}_y}$, which we sample using msk_i and $\widehat{\mathsf{msk}_i}$ respectively.

– **Setup**. Sample $\boldsymbol{\alpha}(\mathbf{x}), \mathsf{H}, \mathbf{W}_1, \mathbf{W}_2$ as in Setup, set $\widehat{\mathsf{msk}_i} = ([\boldsymbol{\alpha}_i + \mathbf{a}^{\perp}\alpha]_2, \mathbf{W}_1, \mathbf{W}_2)$ and output $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g, h), \quad \mathsf{pk}_i = e([\mathbf{A}]_1, [\boldsymbol{\alpha}_i]_2)$.
– **Ciphertexts.** Computes the ciphertexts and a randomly selected $\hat{s} \leftarrow \mathbb{Z}_N$ outputs

$$\mathsf{c}_1 = [\mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s}]_1, \quad \mathsf{c}_2 = [(\mathbf{W}_1^{\top} \cdot \mathsf{id}_x + \mathbf{W}_2^{\top})(\mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s})]_1,$$

$$\mathsf{c}_0 = \mathsf{H}(e([\mathbf{A}\mathbf{s} + \mathbf{b}^{\perp}\hat{s}]_1, [\boldsymbol{\alpha}]_2) \cdot \boldsymbol{m}.$$

– **Key Queries**. On input the j-th key queries id_{y_j}, for each authority $i \in [n]$, output the secret key sk_{i,y_j}. Randomly select a randomness $\mathbf{r}_i \leftarrow_R \mathbb{Z}_p^k$ for every key query id_{y_j},

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \big([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2)(\mathbf{B}\mathbf{r}_i + \mathbf{a}^{\perp}\hat{r})]_2, [\mathbf{B}\mathbf{r}_i + \mathbf{a}^{\perp}\hat{r}]_2\big)\,.$$

If $j = \kappa$, assuming the challenge identity is $\mathsf{id}_{y_j}^{\star}$, we use the SF master secret key shares $\widehat{\mathsf{msk}_i} = ([\boldsymbol{\alpha}_i + \mathbf{a}^{\perp}\alpha]_2, \mathbf{W}_1, \mathbf{W}_2)$ to simulate the secret key shares:

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \Big([\boldsymbol{\alpha}_i + \mathbf{a}^{\perp}\alpha]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j}^{\star} + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, [\mathbf{B}\mathbf{r}_i]_2\Big)\,.$$

According to the α-privacy, $\mathbf{W}$-hiding, and $\mathsf{id}_x \neq \mathsf{id}^\star_{y_j}$, the secret key shares: If $j > \kappa$, outputs

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}^\star_{y_j} + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, [\mathbf{B}\mathbf{r}_i]_2\right). \tag{4.1}$$

If $j < \kappa$, outputs

$$\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \left([\boldsymbol{\alpha}_i + \mathbf{a}^\perp\alpha]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}^\star_{y_j} + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, [\mathbf{B}\mathbf{r}_i]_2\right). \tag{4.2}$$

is statistically indistinguishable. If the id_x of the ciphertext equals $\mathsf{id}^\star_{y_j}$ of the secret keys, according to secret sharing, the α's privacy can also be proved.

- **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$.
- **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

They are identical from the viewpoint of the adversary. The pseudo-normal secret key shares in Eq. 4.2 is in $\mathsf{Game}_{2,\kappa,1}$ and the pseudo-SF secret key shares in Eq. 4.1 is in $\mathsf{Game}_{2,\kappa,2}$.

Lemma 9 (pseudo-SF to SF secret key shares). *There exists $\mathcal{A}_3$ whose running time is roughly that of $\mathcal{A}$ such that for $\kappa = 1, 2, \cdots, q$*

$$|\mathsf{Adv}_{2,\kappa,3} - \mathsf{Adv}_{2,\kappa,2}| \leq \mathsf{Adv}^{\mathsf{k-Lin}}_{\mathcal{G},\mathcal{A}_3}(\lambda).$$

Proof. We will rely on Assumption 1. On input $D = (\mathbb{G}, \mathbb{H}, [\mathbf{A}]_1, [\mathbf{B}]_2, [\mathbf{b}^\perp]_1, [\mathbf{a}^\perp]_2, [\boldsymbol{\alpha}_i]_2)$ and the distribution $T = (T_0, T_1)$, where $T_0 : \mathbf{B}\mathbf{r}_i + \mathbf{a}^\perp\hat{r}, \hat{r} \leftarrow_R \mathbb{Z}_p; T_1 : \mathbf{B}\mathbf{r}_i + \mathbf{a}^\perp\hat{r}, \hat{r} = 0$, the adversary $\mathcal{A}_3$ simulates the adversary $\mathcal{A}$ as follows:

- **Setup**. Sample $\boldsymbol{\alpha}(\mathbf{x}), \mathsf{H}, \mathbf{W}_1, \mathbf{W}_2$ as in Setup, set $\widehat{\mathsf{msk}_i} = ([\boldsymbol{\alpha}_i + \mathbf{a}^\perp\alpha]_2, \mathbf{W}_1, \mathbf{W}_2)$ and output $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathsf{H}, g, h)$, $\mathsf{pk}_i = e([\mathbf{A}]_1, [\boldsymbol{\alpha}_i]_2)$.
- **Ciphertexts.** Computes the ciphertexts and a randomly selected $\hat{s} \leftarrow \mathbb{Z}_N$ outputs

$$\mathsf{c}_1 = [\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s}]_1, \quad \mathsf{c}_2 = [(\mathbf{W}_1^\top \cdot \mathsf{id}_x + \mathbf{W}_2^\top)(\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s})]_1,$$

$$\mathsf{c}_0 = \mathsf{H}(e([\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s}]_1, [\boldsymbol{\alpha}]_2) \cdot \boldsymbol{m}.$$

- **Key Queries**. On input the j-th key queries id_{y_j}, for each authority $i \in [n]$, output the secret key $\mathsf{sk}_{i,\mathsf{id}_{y_j}}$. Randomly select a randomness $\mathbf{r}_i \leftarrow_R \mathbb{Z}_p^k$ for every key query id_{y_j},
  If $j = \kappa$, outputs $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \big([\boldsymbol{\alpha}_i + \mathbf{a}^\perp\alpha]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2) \cdot T]_2, \quad [T]_2\big)$.
  If $j < \kappa$, outputs $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \big([\boldsymbol{\alpha}_i + \mathbf{a}^\perp\alpha]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, \quad [\mathbf{B}\mathbf{r}_i]_2\big)$.
  If $j > \kappa$, outputs $\mathsf{sk}_{i,\mathsf{id}_{y_j}} = \big([\boldsymbol{\alpha}_i]_2 \cdot [(\mathbf{W}_1 \cdot \mathsf{id}_{y_j} + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, \quad [\mathbf{B}\mathbf{r}_i]_2\big)$.
- **Corruption Queries**. Return the corresponding authority's normal master secret key shares $\mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$.
- **Outputs**. Outputs whatever $\mathcal{A}$ outputs.

Observe that when $T = T_0 : \mathbf{Br}_i + \mathbf{a}^{\perp}\hat{r}, \hat{r} \leftarrow_R \mathbb{Z}_p^{\star}$,the output is identical to that in $\mathsf{Game}_{2,\kappa,2}$, and when $T = T_1 : \mathbf{Br}_i + \mathbf{a}^{\perp}\hat{r}, \hat{r} = 0$, the output is identical to that in $\mathsf{Game}_{2.\kappa.3}$.

Lemma 10 (final transition). $|\mathsf{Adv}_{2,q,3} - \mathsf{Adv}_3| \leq 2^{-\Omega(\lambda)}$.

Proof. In $\mathsf{Game}_{2,q,3}$, all the secret key shares are semi-functional, which means they leak no information whatsoever about $\boldsymbol{\alpha}$. For the (SF) challenge ciphertext, $\kappa_0 = \mathsf{H}(e([\mathbf{As} + \mathbf{b}^{\perp}\hat{s}]_1, [\boldsymbol{\alpha}]_2)$ by the left-over hash lemma is $2^{-\Omega(\lambda)}$-close to the uniform distribution over $\{0,1\}^{\lambda}$.

Consistency of key generation To complete the proof of Theorem 2, it remains to prove the consistency of key generation in the prime-order groups, which is defined by the subsection 2.4. This proof is analogous to the proof of composite-order groups in Sec. 3, so we omit it here.

5 Threshold Signatures from Threshold IBE

In this section, we present a generic construction of threshold signatures from threshold IBE, achieved by an extension of Naor's transform [11]. The generic construction $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Psign}, \mathsf{Pver}, \mathsf{Combine}, \mathsf{Ver})$.

5.1 Generic Construction

$\mathsf{Setup}(1^{\lambda})$: Let λ be the security parameters, the public parameters pp, master public key mpk, and master secret key msk:

1. $\mathsf{TIBE.pp}, \mathsf{TIBE.mpk}, \mathsf{TIBE.msk} \leftarrow \mathsf{TIBE.Setup}(1^{\lambda})$
2. $\mathsf{pp}, \mathsf{mpk}, \mathsf{msk} = \mathsf{TIBE.pp}, \mathsf{TIBE.mpk}, \mathsf{TIBE.msk}$
3. return $\mathsf{pp}, \mathsf{mpk.msk}$.

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk})$: For each authority i, public key shares pk_i and master secret key shares $(\mathsf{pk}, \mathsf{pk}_i, \mathsf{msk}_i)$ are:

1. $(\mathsf{TIBE.pk}_i, \mathsf{TIBE.msk}_i) \leftarrow \mathsf{TIBE.KeyGen}(\mathsf{pp}, \mathsf{msk})$
2. $\mathsf{pk}_i = \mathsf{TIBE.pk}_i, \quad \mathsf{msk}_i = \mathsf{TIBE.msk}_i$
3. return $(\mathsf{pk}_i, \mathsf{msk}_i)$

$\mathsf{PSign}(\mathsf{pp}, \mathsf{mpk}, m, i, \mathsf{msk}_i)$: For each authority i, the partial signature σ_i is:

1. $\mathsf{id}_y = m$
2. $\mathsf{sk}_{i,\mathsf{id}_y} \leftarrow \mathsf{TIBE.ShareKeyGen}(\mathsf{pp}, \mathsf{mpk}, i, \mathsf{msk}_i, m||r_i), r_i \leftarrow_R \mathbb{Z}_N$
3. $\sigma_i = \mathsf{sk}_{i,\mathsf{id}_y}$
4. return σ_i.

$\mathsf{PVer}(\mathsf{pp}, \mathsf{mpk}, \sigma_i, \mathsf{pk}_i, m)$: For each authority i, the partial signature verification algorithm is as follows:

1. $\mathsf{id}_y = m$
2. return $0/1 \leftarrow \mathsf{TIBE.ShareVerify}(\mathsf{pp}, \mathsf{mpk}, \mathsf{pk}_i, m, \sigma_i)$

$\mathsf{Combine}(\mathsf{pp}, m, \{\sigma_i\}_{i \in S})$: On input the identity id_y and using a set of shares $\{\mathsf{sk}_{i,\mathsf{id}_y}\}_{i \in S}$ from a subset of authorities $S \subset [n]$ with $|S| \geq t$, the aggregated signature algorithm is as follows:

1. $\mathsf{sk}_{\mathsf{id}_y} \leftarrow \mathsf{TIBE.Combine}(\mathsf{pp}, m, \{\sigma_i\}_{i \in S})$
2. $\sigma = \mathsf{sk}_{\mathsf{id}_y}$
3. return σ.

$\mathsf{Ver}(\mathsf{pp}, \mathsf{mpk}, m, \sigma)$: To verify a signature, we can choose a random message m', encrypt it to get the ciphertext, and then attempt to decrypt it.

1. $\mathsf{id}_x = \mathsf{id}_y = m$
2. $\mathsf{TIBE.ct} \leftarrow \mathsf{TIBE.Enc}(\mathsf{pp}, m', \mathsf{id}_x)$, $m' \leftarrow_R \mathbb{G}_2$
3. $0/1 \leftarrow m' \stackrel{?}{=} \mathsf{TIBE.Dec}(\mathsf{pp}, \mathsf{mpk}, \mathsf{id}_y, \sigma, \mathsf{TIBE.ct})$
4. If valid, outputs 1; otherwise, outputs 0.

5.2 Proof of Security

The threshold signature scheme's security consists of two parts: robustness and unforgettability with adaptive corruption.

Theorem 3 (Robustness). *The non-interactive (n, t)-threshold signatures $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PSign}, \mathsf{PVer}, \mathsf{Combine}, \mathsf{Ver})$ is $\mathsf{RB\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}$ secure for any PPT adversary $\mathcal{A}$.*

Proof. There are two possible winning cases for an adversary $\mathcal{A}$ in the $\mathsf{RB\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}$ game: (1) honestly computed partial signatures do not satisfy the validation check PVer, and (2) every partial signature passes PVer but the honestly aggregated full signature does not satisfy the validation check Ver.

Firstly, let us consider the first case: honestly computed partial signatures do not satisfy the validation check PVer. Note that our construction's PVer is built by the partial signatures and public parameters. If the signer is honest, this implies that the winning condition never occurs for our construction. Secondly, let us consider the second case: every partial signature passes PVer, but the honestly aggregated full signature does not satisfy the validation check Ver. Assuming the hardness of the discrete logarithm in $\mathbb{H}$, the aggregator only aggregates well-formed partial signatures. Thus, we get $\sigma = \prod_{i \in S} \sigma_i^{L_i}$. Note that σ always satisfies the final verification check Ver.

Theorem 4 (Unforgeability with the Adaptive Security). *Non-interactive (n, t)-threshold signature scheme $\mathsf{TS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PSign}, \mathsf{PVer}, \mathsf{Combine}, \mathsf{Ver})$ is $\mathsf{UF\text{-}CMA}^{\mathcal{A}}_{\mathsf{TS}}$ secure for all PPT adversaries $\mathcal{A}$.*

Proof. Here, we reduce the security of the threshold signatures to threshold IBE. If the adversary can forge a valid signature for a message $m^\star$, then in the threshold IBE setting, they can obtain a valid secret key for the identity $\mathsf{id}^\star = m^\star$, which can then be used to distinguish between the challenge ciphertexts of m_0, m_1.

Assume that there is a PPT adversary $\mathcal{A}$ which breaks the security of threshold signatures with advantage ϵ, we then employ it to build another PPT algorithm $\mathcal{B}$ to break a fully secure with adaptively corruption threshold IBE system in Sec. 3 which consists of TIBE= ($\mathsf{TIBE.Setup}$, $\mathsf{TIBE.KeyGen}$, $\mathsf{TIBE.Enc}$,

TIBE.ValidateCT, TIBE.ShareKeyGen, TIBE.ShareVerify, TIBE.Comb, TIBE.Dec). Oracles $\mathsf{O}_1, \mathsf{O}_2$ are implemented by adaptive corruption query and signature query. We simulated by $\mathcal{B}$ as follows:

- *Corruption query:* $\mathcal{A}$ launches a master secret query for signer i to O_1, then $\mathcal{B}$ transfers this query for corruption authority i to the TIBE for generating the corresponding master secret key shares msk_i. It uses the $\mathsf{TIBE.KeyGen}$ algorithm's output to answer this query and returns the msk_i to $\mathcal{B}$. Finally, $\mathcal{B}$ uses this msk_i from TIBE to answer $\mathcal{A}$'s query for the adaptive corruptions.
- *Signature query:* $\mathcal{A}$ launches a signature share query of message m for the signer i to O_2, then $\mathcal{B}$ transfers the query for identity $\mathsf{id}_y = m$ to the TIBE system for generating corresponding secret key share $\mathsf{sk}_{i,\mathsf{id}_y}$. It uses the $\mathsf{TIBE.ShareKeyGen}$ algorithm's output to answer this query and returns the secret key share to $\mathcal{B}$. Finally, $\mathcal{B}$ uses this share from TIBE to answer $\mathcal{A}$'s query for signature share.

Suppose that $\mathsf{Adv}^{\mathsf{TS}}_{\mathcal{A}}(\lambda) = \epsilon$, where ϵ is a non-negligible value. Then, we can build an algorithm $\mathcal{B}$ whose $\mathsf{Adv}^{\mathsf{TIBE}}_{\mathcal{B}}(\lambda) = \epsilon$ as follow: Upon $\mathcal{A}$ making the query, $\mathcal{B}$ generates the master secret key shares and signature share to answer this query by simulating $\mathsf{O}_1, \mathsf{O}_2$, and sends $(\mathsf{msk}_i, \sigma_i)$ to the adversary $\mathcal{A}$. Then $\mathcal{A}$ can forge a valid message-signature pair $(m^\star, \sigma^\star)$ with the restriction that the total number of signature share queries for $m^\star$ and corruption queries is at most t. The master secret key shares msk_i in TS is identical to that in TIBE and the message-signature pair $(m^\star, \sigma^\star)$ is identical to $(\mathsf{id}^\star, \mathsf{sk}_{\mathsf{id}^\star})$ in TIBE. Therefore, the $\mathcal{B}$ can submit a challenger ideneity $\mathsf{id}^\star = m^\star$ and two messages (m_0, m_1). Since $\mathcal{A}$'s successful forgery, in effect, $\mathcal{B}$ has been given the combined secret key $\mathsf{sk}_{\mathsf{id}^\star}$. Then $\mathcal{B}$ can distinguish the challenge ciphertexts easily and break this TIBE system. This means that by this simulation, we have successfully reduced the unforgeability of TS to the security of this TIBE system. And we have

$$\mathsf{Adv}^{\mathsf{TS}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{TIBE}}_{\mathcal{B}}(\lambda).$$

That is to say, if an adversary cannot successfully break the TIBE system we constructed, it cannot forge a valid signature in our TS scheme either. Thus, for any PPT adversary $\mathcal{A}$, its advantage of breaking the adaptive security of our TS scheme is negligible.

6 Instantiations of Threshold Signatures

Here, we present two TS instantiations from threshold IBE in composite-order groups in Sec. 3 and in prime-order groups in Sec. 4, respectively.

6.1 Instantiation in composite-order groups

$\mathsf{Setup}(1^\lambda)$: Run $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, e) \leftarrow \mathcal{G}(1^\lambda)$ to obtain the composite-order groups description and generators $g \in \mathbb{G}, g_1 \in \mathbb{G}_{p_1}, g_2 \in \mathbb{G}_{p_2}, h \in \mathbb{H}, h_1 \in \mathbb{H}_{p_1}, h_2 \in$

$\mathbb{H}_{p_2}$. We define the hash function $\mathsf{H} : \mathbb{G}_T \to \mathbb{G}_{p_2}$. Then, randomly picks $a, b \leftarrow_R \mathbb{Z}_N$ and $\alpha \in \mathbb{Z}_N$. Let public key be $\mathsf{pk} = e(g_1, h_1)^\alpha$. Output public parameter $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, g, h, g_1, h_1, \mathsf{H})$ and

$$\mathsf{mpk} = \left(g_1^a, g_1^b, h_1^a, h_1^b, \mathsf{pk}\right), \mathsf{msk} = (\alpha, a, b)$$

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk})$: Sample a uniformly random polynomial $\alpha(x) = \alpha + \sum_{i=1}^{t-1} a_i x^i$ of degree $t-1$ with coefficient $a_i \in \mathbb{Z}_N$. We define α_i as the shares of $\alpha(x)$ and constant term $\alpha_0 = \alpha$. For each signer $i \in [n]$, its public and secret key shares pair $(i, \mathsf{pk}_i, \mathsf{msk}_i)$ are $\mathsf{pk}_i = e(g_1, h_1)^{\alpha_i}, \quad \mathsf{msk}_i = (h^{\alpha_i}, a, b)$.

$\mathsf{PSign}(\mathsf{pp}, \mathsf{mpk}, m, i, \mathsf{msk}_i)$: On input the message $m \in \mathcal{M}$ and master secret key shares msk_i for each signer $i \in [n]$, the partial signature σ_i

$$\sigma_i = \left(\sigma_{1,i} = h^{\alpha_i} \cdot h_1^{r_i(a \cdot m + b)}, \quad \sigma_{2,i} = h_1^{r_i}\right).$$

where $r_i \leftarrow_R \mathbb{Z}_N$ is randomly selected by signer i. The partial signature form as $\sigma_i = (\sigma_{1,i}, \sigma_{2,i})$.

$\mathsf{PVer}(\mathsf{pp}, \mathsf{mpk}, \sigma_i, \mathsf{pk}_i, m)$: On input threshold public key shares pk_i, partial signature share σ_i, and message m. The algorithm accepts if and only if the following equation holds.

$$e\left(\sigma_{1,i}, g_1\right) = \mathsf{pk}_i \cdot e(g_1^{a \cdot m + b}, \sigma_{2,i}). \tag{6.1}$$

$\mathsf{Combine}(\mathsf{pp}, m, \{\sigma_i\}_{i \in S})$: Upon receiving a message m, and the corresponding threshold public key shares and partial signatures tuples $\{\mathsf{pk}_i, \sigma_i\}_{i \in S}$ from a subset of authorities $S \subset [n]$ with $|S| \geq t$. Then, validates each of the partial signatures using Pver. If any partial signatures verification fails, i.e., returns 0, the $\mathsf{Combine}$ algorithm returns $\perp$. Otherwise, the $\mathsf{Combine}$ algorithm computes the signature σ as: Using the σ_i generated from the t signers to aggregate into signature $\sigma = (\sigma_1, \sigma_2)$ as follows.

$$\sigma_1 = \prod_{i \in S} \sigma_{1,i}^{L_i} = h^\alpha \cdot h_1^{r(a \cdot m + b)}, \quad \sigma_2 = \prod_{i \in S} \sigma_{2,i}^{L_i} = h_1^r,$$

where, $L_i = \prod_{i \in S} \frac{j}{j - i}$ is the Lagrange interpolation for the signer i and we define $r = \sum_{i \in S} L_i \cdot r_i$.

$\mathsf{Ver}(\mathsf{pp}, \mathsf{mpk}, m, \sigma)$: The verification procedure of our scheme is identical to that of the threshold signature. On input pp and the signature $\sigma = (\sigma_1, \sigma_2)$ on a message m, a verifier accepts if $e(g_1, h_1)^\alpha = \frac{e(g_1, \sigma_1)}{e(g_1^{a \cdot m + b}, \sigma_2)}$.

6.2 Instantiation in prime-order groups

$\mathsf{Setup}(1^\lambda)$: We define the hash function $\mathsf{H} : \mathbb{G}_T \to \mathbb{G}_p^{k+1}$, which maps the pairing element $e(g, h)$ to elements in $\mathbb{G}_p^{k+1}$. Concretely, for the pairing element $e(g, h) \in \mathbb{G}_T$, $\mathsf{H}(e(g, h)) = [\mathbf{e}]_1$, where $[\mathbf{e}]_1 \leftarrow_R \mathbb{G}_p^{k \times 1}$. Sample $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, along with $\mathbf{W}_1, \mathbf{W}_2 \leftarrow_R \mathbb{Z}_p^{(k+1) \times (k+1)}$, where $\mathbf{a}^{\perp^\top} \mathbf{A} =$

$\mathbf{b}^{\perp\top}\mathbf{B} = \mathbf{0}$ and $\mathbf{a}^{\perp\top}\mathbf{b}^{\perp} \neq 0$; Picks randomly $\boldsymbol{\alpha} \in \mathbb{Z}_p^k$ and let public key be $\mathsf{pk} = e([\mathbf{A}]_1, [\boldsymbol{\alpha}]_2)$. Output public parameters $\mathsf{pp} = (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, [\mathbf{A}]_1, [\mathbf{B}]_2, \mathsf{H})$ and

$$\mathsf{mpk} = \left([\mathbf{W}_1^\top \mathbf{A}]_1, [\mathbf{W}_2^\top \mathbf{A}]_1, [\mathbf{W}_1\mathbf{B}]_2, [\mathbf{W}_2\mathbf{B}]_2, \mathsf{pk}\right), \mathsf{msk} = (\boldsymbol{\alpha}, \mathbf{W}_1, \mathbf{W}_2).$$

$\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk})$: Picks randomly $\boldsymbol{\alpha}(x) = \boldsymbol{\alpha} + \sum_{i=1}^{t-1} a_i \mathbf{x}^i$ of degree $t-1$ with coefficient $a_i \in \mathbb{Z}_N$. We define $\boldsymbol{\alpha}_i$ as the shares of $\boldsymbol{\alpha}(x)$ and constant term $\boldsymbol{\alpha}_0 = \boldsymbol{\alpha}$. For each signer $i \in [n]$, its public and secret key shares pair $(i, \mathsf{pk}_i, \mathsf{msk}_i)$ is: $\mathsf{pk}_i = e([\mathbf{A}]_1, [\boldsymbol{\alpha}_i]_2), \quad \mathsf{msk}_i = ([\boldsymbol{\alpha}_i]_2, \mathbf{W}_1, \mathbf{W}_2)$.

$\mathsf{PSign}(\mathsf{pp}, \mathsf{mpk}, m, i, \mathsf{msk}_i)$: On input the message $m \in \mathcal{M}$ and master secret key shares msk_i for each signer $i \in [n]$, the partial signature (i, σ_i)

$$\sigma_i = (\sigma_{1,i} = [\boldsymbol{\alpha}_i]_2[(\mathbf{W}_1 \cdot m + \mathbf{W}_2)\mathbf{B}\mathbf{r}_i]_2, \quad \sigma_{2,i} = [\mathbf{B}\mathbf{r}_i]_2),$$

where $\mathbf{r}_i \leftarrow_R \mathbb{Z}_p^k$ is randomly selected by authority i, not by secret share. Then, the secret key is as follows: $\sigma_i = (\sigma_{1,i}, \sigma_{2,i})$.

$\mathsf{PVer}(\mathsf{pp}, \mathsf{mpk}, \sigma_i, \mathsf{pk}_i, m)$: On input the threshold public key shares pk_i, the partial signature tuple σ_i, and the message m validates σ_i. The algorithm accepts if and only if the equation holds.

$$e(\sigma_{1,i}, [\mathbf{A}]_1) = \mathsf{pk}_i \cdot e([\mathbf{W}_1^\top \mathbf{A} \cdot m + \mathbf{W}_2^\top \mathbf{A}]_1, \sigma_{2,i}).$$

$\mathsf{Combine}(\mathsf{pp}, m, \{\sigma_i\}_{i \in S})$: Upon receiving a message m, and the corresponding threshold public key shares and partial signatures tuples $\{\mathsf{pk}_i, \sigma_i\}_{i \in S}$ from a subset of authorities $S \subset [n]$ with $|S| \geq t$. Then, validates each of the partial signatures using Pver. If any partial signatures verification fails, i.e., returns 0, the $\mathsf{Combine}$ algorithm returns $\perp$. Otherwise, the $\mathsf{Combine}$ algorithm computes threshold signature σ as: Using the σ_i generated from t signers to aggregate into one $\sigma = (\sigma_1, \sigma_2)$ as follows.

$$\sigma_1 = \prod_{i \in S} \sigma_{1,i}^{L_i} = [\boldsymbol{\alpha}]_2 \cdot [(\mathbf{W}_1 \cdot m + \mathbf{W}_2)\mathbf{B}\mathbf{r}]_2, \quad \sigma_2 = \prod_{i \in S} \sigma_{2,i}^{L_i} = [\mathbf{B}\mathbf{r}]_2,$$

where, $L_i = \prod_{i \in S} \frac{j}{j-i}$ is the Lagrange interpolation for the signer i and we define $\mathbf{r} = \sum_{i \in S} L_i \cdot \mathbf{r}_i$.

$\mathsf{Ver}(\mathsf{pp}, \mathsf{mpk}, m, \sigma)$: The verification procedure of our scheme is identical to that of the threshold signature: on input the public parameters pp and the signature $\sigma = (\sigma_1, \sigma_2)$ on a message m, a verifier accepts if

$$\begin{aligned}\frac{e([\mathbf{A}]_1, \sigma_1)}{e([\mathbf{W}_1^\top \mathbf{A} \cdot m + \mathbf{W}_2^\top \mathbf{A}]_1, \sigma_2)} &= \frac{e([\mathbf{A}]_1, [\boldsymbol{\alpha}]_2 \cdot [(\mathbf{W}_1 \cdot m + \mathbf{W}_2)\mathbf{B}\mathbf{r}]_2)}{e([\mathbf{W}_1^\top \mathbf{A} \cdot m + \mathbf{W}_2^\top \mathbf{A}]_1, [\mathbf{B}\mathbf{r}]_2)} \\ &= e([\mathbf{A}]_1, [\boldsymbol{\alpha}]_2).\end{aligned}$$

Acknowledgements. We want to thank Hoeteck Wee for helpful advice. We would also like to thank anonymous reviewers for their helpful suggestions on how to improve the paper presentation.

References

1. Aumasson, J., Hamelink, A., Shlomovits, O.: A survey of ECDSA threshold signing. IACR Cryptol. ePrint Arch. p. 1390 (2020), https://eprint.iacr.org/2020/1390
2. Bacho, R., Das, S., Loss, J., Ren, L.: Glacius: Threshold schnorr signatures from DDH with full adaptive security. IACR Cryptol. ePrint Arch. p. 1628 (2024), https://eprint.iacr.org/2024/1628
3. Bacho, R., Das, S., Loss, J., Ren, L.: Adaptively secure three-round threshold schnorr signatures from DDH. In: Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 16005, pp. 390–422. Springer (2025). https://doi.org/10.1007/978-3-032-01887-8_13
4. Bacho, R., Loss, J.: On the adaptive security of the threshold BLS signature scheme. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022. pp. 193–207. ACM (2022). https://doi.org/10.1145/3548606.3560656
5. Bacho, R., Loss, J., Tessaro, S., Wagner, B., Zhu, C.: Twinkle: Threshold signatures from DDH with full adaptive security. In: Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14651, pp. 429–459. Springer (2024). https://doi.org/10.1007/978-3-031-58716-0_15
6. Bellare, M., Waters, B., Yilek, S.: Identity-based encryption secure against selective opening attack. In: Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6597, pp. 235–252. Springer (2011). https://doi.org/10.1007/978-3-642-19571-6_15
7. Benhamouda, F., Halevi, S., Krawczyk, H., Ma, Y., Rabin, T.: SPRINT: high-throughput robust distributed schnorr signatures. In: Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14655, pp. 62–91. Springer (2024). https://doi.org/10.1007/978-3-031-58740-5_3
8. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2567, pp. 31–46. Springer (2003). https://doi.org/10.1007/3-540-36288-6_3
9. Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3027, pp. 223–238. Springer (2004). https://doi.org/10.1007/978-3-540-24676-3_14
10. Boneh, D., Boyen, X., Halevi, S.: Chosen ciphertext secure public key threshold encryption without random oracles. In: Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings. Lecture Notes in Computer Science, vol. 3860, pp. 226–243. Springer (2006). https://doi.org/10.1007/11605805_15

11. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2139, pp. 213–229. Springer (2001). https://doi.org/10.1007/3-540-44647-8_13
12. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2248, pp. 514–532. Springer (2001). https://doi.org/10.1007/3-540-45682-1_30
13. Boneh, D., Shoup, V.: A graduate course in applied cryptography. Draft 0.6 (2023), https://toc.cryptobook.us/
14. Canetti, R., Gennaro, R., Goldfeder, S., Makriyannis, N., Peled, U.: UC non-interactive, proactive, threshold ECDSA with identifiable aborts. In: CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020. pp. 1769–1787. ACM (2020). https://doi.org/10.1145/3372297.3423367
15. Canetti, R., Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Adaptive security for threshold cryptosystems. In: Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 98–115. Springer (1999). https://doi.org/10.1007/3-540-48405-1_7
16. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9057, pp. 595–624. Springer (2015). https://doi.org/10.1007/978-3-662-46803-6_20
17. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II. Lecture Notes in Computer Science, vol. 8043, pp. 435–460. Springer (2013). https://doi.org/10.1007/978-3-642-40084-1_25
18. Chen, J., Wee, H.: Dual system groups and its applications - compact HIBE and more. IACR Cryptol. ePrint Arch. p. 265 (2014), http://eprint.iacr.org/2014/265
19. Chu, H., Gerhart, P., Ruffing, T., Schröder, D.: Practical schnorr threshold signatures without the algebraic group model. In: Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14081, pp. 743–773. Springer (2023). https://doi.org/10.1007/978-3-031-38557-5_24
20. Crites, E.C., Kohlweiss, M., Preneel, B., Sedaghat, M., Slamanig, D.: Threshold structure-preserving signatures. In: Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14439, pp. 348–382. Springer (2023). https://doi.org/10.1007/978-981-99-8724-5_11

21. Crites, E.C., Komlo, C., Maller, M.: How to prove schnorr assuming schnorr: Security of multi- and threshold signatures. IACR Cryptol. ePrint Arch. p. 1375 (2021), https://eprint.iacr.org/2021/1375
22. Crites, E.C., Komlo, C., Maller, M.: Fully adaptive schnorr threshold signatures. In: Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14081, pp. 678–709. Springer (2023). https://doi.org/10.1007/978-3-031-38557-5_22
23. Das, S., Camacho, P., Xiang, Z., Nieto, J., Bünz, B., Ren, L.: Threshold signatures from inner product argument: Succinct, weighted, and multi-threshold. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023. pp. 356–370. ACM (2023). https://doi.org/10.1145/3576915.3623096
24. Das, S., Ren, L.: Adaptively secure BLS threshold signatures from DDH and co-cdh. In: Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VII. Lecture Notes in Computer Science, vol. 14926, pp. 251–284. Springer (2024). https://doi.org/10.1007/978-3-031-68394-7_9
25. Desmedt, Y.: Society and group oriented cryptography: A new concept. In: Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings. Lecture Notes in Computer Science, vol. 293, pp. 120–127. Springer (1987). https://doi.org/10.1007/3-540-48184-2_8
26. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings. Lecture Notes in Computer Science, vol. 435, pp. 307–315. Springer (1989). https://doi.org/10.1007/0-387-34805-0_28
27. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.L.: An algebraic framework for diffie-hellman assumptions. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II. Lecture Notes in Computer Science, vol. 8043, pp. 129–147. Springer (2013). https://doi.org/10.1007/978-3-642-40084-1_8
28. Fouque, P., Pointcheval, D.: Threshold cryptosystems secure against chosen-ciphertext attacks. In: Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2248, pp. 351–368. Springer (2001). https://doi.org/10.1007/3-540-45682-1_21
29. Gagol, A., Kula, J., Straszak, D., Swietek, M.: Threshold ECDSA for decentralized asset custody. IACR Cryptol. ePrint Arch. p. 498 (2020), https://eprint.iacr.org/2020/498
30. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust threshold DSS signatures. In: Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding. Lecture Notes in Computer Science, vol. 1070, pp. 354–371. Springer (1996). https://doi.org/10.1007/3-540-68339-9_31
31. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. J. Cryptol. **20**(1), 51–83 (2007). https://doi.org/10.1007/S00145-006-0347-3

32. Gong, J., Waters, B., Wee, H., Wu, D.J.: Threshold batched identity-based encryption from pairings in the plain model. Cryptology ePrint Archive, Paper 2025/2103 (2025), https://eprint.iacr.org/2025/2103
33. Groth, J., Shoup, V.: Fast batched asynchronous distributed key generation. In: Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14655, pp. 370–400. Springer (2024). https://doi.org/10.1007/978-3-031-58740-5_13
34. Jarecki, S.: Efficient threshold cryptosystems. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2001), https://hdl.handle.net/1721.1/8370
35. Jarecki, S., Lysyanskaya, A.: Adaptively secure threshold cryptography: Introducing concurrency, removing erasures. In: Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 221–242. Springer (2000). https://doi.org/10.1007/3-540-45539-6_16
36. Jarecki, S., Nazarian, P.: Adaptively secure threshold blind BLS signatures and threshold oblivious PRF. IACR Cryptol. ePrint Arch. p. 483 (2025), https://eprint.iacr.org/2025/483
37. Katz, J., Yung, M.: Threshold cryptosystems based on factoring. In: Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2501, pp. 192–205. Springer (2002). https://doi.org/10.1007/3-540-36178-2_12
38. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings. vol. 5978, pp. 455–479. Springer (2010). https://doi.org/10.1007/978-3-642-11799-2_27
39. Libert, B., Joye, M., Yung, M.: Born and raised distributively: fully distributed non-interactive adaptively-secure threshold signatures with short shares. In: ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014. pp. 303–312. ACM (2014). https://doi.org/10.1145/2611462.2611498
40. Libert, B., Yung, M.: Adaptively secure non-interactive threshold cryptosystems. Theor. Comput. Sci. **478**, 76–100 (2013). https://doi.org/10.1016/J.TCS.2013.01.001
41. Lysyanskaya, A., Peikert, C.: Adaptive security in the threshold setting: From cryptosystems to signature schemes. In: Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2248, pp. 331–350. Springer (2001). https://doi.org/10.1007/3-540-45682-1_20
42. Rabin, T.: A simplified approach to threshold and proactive RSA. In: Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 89–104. Springer (1998). https://doi.org/10.1007/BFB0055722
43. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979). https://doi.org/10.1145/359168.359176

44. Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. J. Cryptol. **15**(2), 75–96 (2002). https://doi.org/10.1007/S00145-001-0020-9
45. Tessaro, S., Zhu, C.: Threshold and multi-signature schemes from linear hash functions. In: Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 628–658. Springer (2023). https://doi.org/10.1007/978-3-031-30589-4_22
46. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5677, pp. 619–636. Springer (2009). https://doi.org/10.1007/978-3-642-03356-8_36
47. Wee, H.: Dual system encryption via predicate encodings. In: Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8349, pp. 616–637. Springer (2014). https://doi.org/10.1007/978-3-642-54242-8_26

A Dual System Groups

The dual system group is a powerful proof technique introduced by Waters [46], which aims to prove adaptive security in the standard model. The core idea is to introduce a semi-functional form in parallel with the real scheme in the security proof. We introduce the dual system groups [46,47,16] as follows.

A.1 Definitions

(Orthogonality). For an element $g_1 \in \mathbb{G}_{p_1}$ and $h_2 \in \mathbb{H}_{p_2}$ in the semi-functional space, we have $e(g_1, h_2) = 1_{\mathbb{G}_T}$. Similarly, for an element $g_2 \in \mathbb{G}_{p_2}$ and $h_1 \in \mathbb{H}_{p_1}$, we have $e(g_2, h_1) = 1_{\mathbb{G}_T}$.

(Non-degeneracy.) For an element $g_2^s \in \mathbb{G}_{p_2}$, we have $e(g_2^s, h_2) = e(g_2, h_2)^s \neq 1$ for $s \xleftarrow{R} \mathbb{Z}_N$.

(Associativity). For all $(g_1, g_2) \in \mathbb{G}^2$ and all $(h_1, h_2) \in \mathbb{H}^2$ drawn from the respective normal distributions, for all $i \in 1, 2$, we have that $e(g_1, h_i) = e(g_i, h_1)$. We require this property for correctness. In particular, for all $(g_1, g_2) \in \mathbb{G}^2$ and all $(h_1, h_2) \in \mathbb{H}^2$ drawn from the respective normal distributions, we have that for all $i = 1, 2$, $e(g_j, h_i) = e(g_i, h_j) = 1, \forall j \neq i$ along with additional analogous requirements amongst the SF components.

(α-privacy.) A predicate encoding for P is a pair of algorithms [47]. If $\mathsf{P}(x, y) = 0$ and $\alpha \in \mathcal{D}$, the joint distribution $g_1^{s(a\mathsf{id}_x+b)}$ and $(h^\alpha \cdot h_1^{r(a\mathsf{id}_y+b)}, h_1^r)$ hide α perfectly. That is, for all $\alpha, \alpha' \in \mathcal{D}$, the following joint distributions are identically distributed:

$$\{(g_1^s, g_1^{s(a\mathsf{id}_x+b)}), (h^\alpha \cdot h_1^{r(a\mathsf{id}_y+b)}, h_1^r)\} \quad \text{and} \quad \{(g_1^s, g_1^{s(a\mathsf{id}_x+b)}), (h^{\alpha'} \cdot h_1^{r(a\mathsf{id}_y+b)}, h_1^r)\},$$

where the randomness is taken over $(\boldsymbol{w} = (a, b), r) \leftarrow_R \mathcal{W} \times \mathcal{R}$.

($\boldsymbol{w}$-hiding.) There exists some element $0 \in \mathcal{R}$ such that for all $(\alpha, y, \boldsymbol{w}) \in \mathcal{D} \times \mathcal{Y} \times \mathcal{W}$, the encoding $h^\alpha \cdot h_1^{r(a\mathsf{id}_y+b)}$ is statistically independent of $\boldsymbol{w} = (a, b)$, that is, for $\boldsymbol{w}' = (a', b') \in \mathcal{W}$: the distribution $h^\alpha \cdot h_1^{r(a\mathsf{id}_y+b)} \approx h_1^{r(a'\mathsf{id}_y+b')}$. We rely on the fact that α is perfectly hidden in the proof of security, so that non-adaptive indistinguishability implies adaptive indistinguishability. Concretely, we claim that α-privacy implies that even if y is chosen adaptively after seeing $(h_1^{r(a\mathsf{id}_y+b)}, h_1^r)$, the distributions $(h^\alpha \cdot h_1^{r(a\mathsf{id}_y+b)}, h_1^r)$ and $(h_1^{r(a'\mathsf{id}_y+b')}, h_1^r)$ are perfectly indistinguishable.