Threshold Structure-Preserving Signatures: Strong and Adaptive Security under Standard Assumptions

Aikaterini Mitrokotsa[1], Sayantan Mukherjee[2], Mahdi Sedaghat[3], Daniel Slamanig[4], and Jenit Tomy[1]

[1] University of St. Gallen, St. Gallen, Switzerland
first.last@unisg.ch

[2] Indian Institute of Technology, Jammu, India
csayantan.mukherjee@gmail.com

[3] COSIC, KU Leuven, Leuven, Belgium
ssedagha@esat.kuleuven.be

[4] Research Institute CODE, Universität der Bundeswehr München, München, Germany
daniel.slamanig@unibw.de

Abstract. Structure-preserving signatures (SPS) have emerged as an important cryptographic building block, as their compatibility with the Groth-Sahai (GS) NIZK framework allows to construct protocols under standard assumptions with reasonable efficiency.

Over the last years there has been a significant interest in the design of threshold signature schemes. However, only very recently Crites et al. (ASIACRYPT 2023) have introduced threshold SPS (TSPS) along with a fully non-interactive construction. While this is an important step, their work comes with several limitations. With respect to the construction, they require the use of random oracles, interactive complexity assumptions and are restricted to so called indexed Diffie-Hellman message spaces. Latter limits the use of their construction as a drop-in replacement for SPS. When it comes to security, they only support static corruptions and do not allow partial signature queries for the forgery.

In this paper, we ask whether it is possible to construct TSPS without such restrictions. We start from an SPS from Kiltz, Pan and Wee (CRYPTO 2015) which has an interesting structure, but thresholdizing it requires some modifications. Interestingly, we can prove it secure in the strongest model (TS-UF-1) for fully non-interactive threshold signatures (Bellare et al., CRYPTO 2022) and even under fully adaptive corruptions. Surprisingly, we can show the latter under a standard assumption without requiring any idealized model. All known constructions of efficient threshold signatures in the discrete logarithm setting require interactive assumptions and idealized models.

Concretely, our scheme in type III bilinear groups under the SXDH assumption has signatures consisting of 7 group elements. Compared to the TSPS from Crites et al. (2 group elements), this comes at the cost of efficiency. However, our scheme is secure under standard assumptions, achieves strong and adaptive security guarantees and supports general message spaces, i.e., represents a drop-in replacement for many SPS applications. Given these features, the increase in the size of the signature seems acceptable even for practical applications.

1 Introduction

Structure-Preserving Signatures. Structure-preserving signature schemes (SPS for short) introduced by Abe et al. [AFG+10] are signatures defined over bilinear groups where the messages, public keys and signatures are required to be source group elements. Moreover, signature verification just consists of group membership testing and evaluating pairing product equations (PPE). SPS are very attractive as they can be combined with efficient pairing-based non-interactive zero-knowledge (NIZK) proofs due to Groth and Sahai (GS) [GS08]. This allows to construct many privacy-preserving cryptographic primitives and protocols under standard assumptions with reasonable practical efficiency.

SPS have been used in the literature to construct numerous cryptographic primitives and building blocks. Among them are many variants of signatures such as blind signatures [AFG+10, FHS15], group signatures [AFG+10, LPY15], traceable signatures [ACHO11], policy-compliant signatures [BMW21, BSW23], homomorphic and network coding signatures [LPJY13, ALP12] and protocols such as anonymous credentials [CDHK15], delegatable anonymous credentials [Fuc11], compact verifiable shuffles [CKLM12] or anonymous e-cash [BCF+11]. Due to their wide range of applications, SPS have attracted significant research interest. Looking ahead to the threshold setting (i.e., TSPS), we note that typical applications of SPS in privacy-preserving applications are as follows: a user obtains a signature from some entity and then prove possession of a valid signature without revealing it using GS NIZK. Consequently, thresholdizing the SPS signing process does not have any impact on the remaining protocol and thus, TSPS can be considered a drop-in replacement for SPS.

The first SPS scheme presented by Abe et al. in [AFG$^+$10] was followed by a line of research to obtain SPS with short signatures in the generic group model (GGM) [AGHO11, AGOT14, Gha16, Gha17], lower bounds [AGHO11, AGO11, AAOT18], security under standard assumptions [ACD$^+$12, CDH12, HJ12, KPW15, LPY15, JR17] as well as tight security reductions [AHN$^+$17, JOR18, GHKP18, AJOR18, AJO$^+$19, CH20].

Threshold Signatures. Motivated by real-world deployments in decentralized systems such as distributed ledger technologies, cryptocurrencies, and decentralized identity management, the use of threshold cryptography [DF90] and in particular threshold signatures has become a very active field of research in the last years with a main focus on ECDSA [GG18, CGG$^+$20, DOK$^+$20, ANO$^+$22, DMZ$^+$21, BS23, WMYC23], Schnorr [KG20, CKM23] and BLS [BL22] signatures. We recall that an (n, t) threshold signature allows a set of n potential signers to jointly compute a signature for a message m, which verifies under a single verification key, as long as at least a threshold t many signers participate.

There are different types of constructions in the literature; ones that require multiple rounds of interaction (e.g., ECDSA [GG18, CGG$^+$20]), ones that require a pre-processing round that does not depend on the message (often called non-interactive schemes), e.g,. FROST [KG20] and finally, ones that are fully non-interactive. The latter are schemes where all the participating signers can simply send a partial signature and the final signatures can then be combined from threshold many valid partial signatures, e.g., BLS [Bol03].

Security of Threshold Signatures. Although many works on threshold signatures were known in the literature, the rigorous study of security notions was done only very recently. In particular, Bellare et al. in [BCK$^+$22] studied a hierarchy of different notions of security for non-interactive schemes. As our work focuses on fully non-interactive schemes, we do not recall the entire hierarchy but only the ones relevant for this setting. In particular, the $\mathsf{TS\text{-}UF\text{-}0}$ notion is the weaker one and prohibits adversaries from querying the signing oracle for partial signatures on the challenge message, i.e., the message corresponding to the forged signature. The stronger $\mathsf{TS\text{-}UF\text{-}1}$ notion, which will be our main focus, allows adversaries to query the signing oracle up to $t - |\mathsf{CS}|$ times for partial signatures, even on the challenge message. Here CS with $|\mathsf{CS}| < t$ denotes the set of (statically corrupted) signers. Surprisingly, the majority of works on threshold signatures in the literature relied on weaker $\mathsf{TS\text{-}UF\text{-}0}$-style notions instead of the much more realistic $\mathsf{TS\text{-}UF\text{-}1}$ notion.

Another dimension in the security of threshold signatures is whether they support static or adaptive corruptions. In the case of static corruptions, the adversary has to declare the set of corrupted signers, CS, before seeing any parameters of the system apart from (n, t). In contrast, an adaptive adversary can choose the set of corrupted signers within a security game based on its view of the execution, which is a realistic assumption in the decentralized setting. All the notions in [BCK$^+$22] consider only a static setting and refer to a complexity leveraging argument for adaptive security. Precisely, it suggests that for small number of parties, a guessing argument can yield adaptive security for any statically secure scheme with a loss of $\binom{n}{t-1}$, i.e., guessing the set of corrupted parties and aborting if the guess is wrong. However, this exponential loss of security can become significant as the number of parties increases, e.g., supporting $n \geq 1024$ (cf. [CKM23]). While there are known generic techniques to lift statically secure schemes to adaptively secure ones [CGJ$^+$99, JL00, LP01], they all have undesirable side-effects such as relying on additional heavy tools, e.g., non-committing encryption [CFGN96], or relying on strong assumptions such as reliable erasure of secret states (cf. [CKM23]).

Apart from the adaptively secure threshold RSA signatures [ADN06], until recently there were no results on adaptively secure threshold signatures based on popular signature schemes in the discrete logarithm or pairing setting. Only very recently Bacho and Loss [BL22] as well as Crites et al. [CKM23] have shown tight adaptive security for threshold versions of the popular BLS [BLS01] and Schnorr schemes [Sch91], respectively. Interestingly, all these adaptive security proofs need to rely on interactive assumptions and in particular variants of the One-More Discrete Logarithm Assumption [BNPS03], which is known as a strong assumption. Only very recently and concurrent to this work, Bacho et al. [BLT$^+$23] as well as Das and Ren [DR23] present schemes from standard and non-interactive assumptions in the pairing-free discrete logarithm setting and pairing setting, respectively. It is interesting that only few of the existing works achieve adaptive security under the $\mathsf{TS\text{-}UF\text{-}1}$ notion, e.g., [LJY16, BL22, DR23], with [LJY16] being the only one from standard assumptions and without requiring idealized models.

Threshold SPS. Recently, Crites et al. [CKP+23] have extended the concept of threshold signatures to threshold SPS (TSPS). They introduce a definitional framework for fully non-interactive TSPS and provide a construction that is proven secure in the Random Oracle Model (ROM) [BR93] under the hardness of a new interactive assumption, called the GPS_3 assumption, which is analyzed in the Algebraic Group Model (AGM) [FKL18]. The authors start from an SPS proposed by Ghadafi [Gha16], that is secure in the Generic Group Model (GGM), and introduce a message indexing technique to avoid non-linear operations in the signature components and thus to obtain a fully non-interactive threshold version. While the TSPS proposed in [CKP+23] is highly efficient and compact (only 2 group elements), the defined message space is restricted to a so called indexed Diffie-Hellman message space. This prevents its use as a drop-in-replacement for SPS in arbitrary applications of SPS that are desired to be thresholdized. Additionally, the security of their proposed TSPS is only shown in the $\mathsf{TS\text{-}UF\text{-}0}$ model, i.e., under static corruptions.

1.1 Our Contributions

In this paper, we ask if it is possible to construct TSPS without the aforementioned restrictions and we answer this question affirmatively. We start with an observation that the SPS from Kiltz, Pan and Wee [KPW15] has an interesting structure that makes it amenable for thresholdizing although this process requires some modifications of the original scheme. While Crites et al. [CKP+23] prove security in the $\mathsf{TS\text{-}UF\text{-}0}$ model, i.e., under static corruptions, we are able to prove our construction is secure in the strongest model (TS-UF-1) for non-interactive threshold signatures [BCK+22] and even under fully adaptive corruptions (which we denote as $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ security). We provide a brief overview in Table 1 about our results.

Table 1. Overview of security notions and our results. t denotes the threshold, M^* the message corresponding to the forgery, S_1 the set recording signer indices of issued partial signatures and CS the set of corrupted signers.

Security Notion	Corruption Model	Winning Condition	Our Scheme (proof)
$\mathsf{TS\text{-}UF\text{-}0}$	static corruptions	$S_1(M^*) = \emptyset$	Theorem 1
$\mathsf{TS\text{-}UF\text{-}1}$	static corruptions	$\lvert S_1(M^*)\rvert < t - \lvert\mathsf{CS}\rvert$	Theorem 2
$\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$	adaptive corruptions	$\lvert S_1(M^*)\rvert < t - \lvert\mathsf{CS}\rvert$	Theorem 3

Interestingly, we can do so by relying on standard assumptions, i.e., the Matrix Diffie-Hellman (MDDH) assumption family [EHK+17, MRV16]. While this comes at some cost in concrete efficiency, as shown in Table 2, the overhead is still not significant. For instance, when instantiated in type III bilinear groups under the SXDH assumption ($k = 1$), then signatures consist of 7 group elements. When taking the popular `BLS12-381` curve giving around 110 bit of security, this amounts to signatures of size around 380 bytes. Compared to 256 bytes for an RSA signature with comparable security (2048 bit modulus), this gives an increase of around 50%. This seems perfectly tolerable for most practical applications.

As can be seen from Table 2, an important benefit of our TSPS over the one by Crites et al. [CKP+23] is that it is not limited to an indexed Diffie-Hellman message space, but works for arbitrary group message vectors. Thus, it represents a drop-in replacement for SPS when aiming to thresholdize its applications (such as anonymous credentials, e-cash, etc). Moreover, we prove the unforgeability of the proposed TSPS scheme against an adaptive adversary under a stronger $\mathsf{TS\text{-}UF\text{-}1}$ notion of security. We recall that in contrast, the TSPS proposed by Crites et al. in [CKP+23] only achieves $\mathsf{TS\text{-}UF\text{-}0}$ security against a static adversary based on an interactive assumption, called GPS_3, in the AGM and ROM.

1.2 Technical Overview

Considering the insights discussed in [CKP+23, Section 1], it can be deduced that a fully non-interactive TSPS scheme does not involve any non-linear operations during the partial signing phase. The use

Table 2. Comparison with the existing threshold structure-preserving signature by Crites et al. [CKP+23]. iDH refers to the indexed Diffie-Hellman message spaces. ℓ is the length of the message vector to be signed. $|\mathbb{G}_i|$ denote the bit-length of elements in groups $\mathbb{G}_i$ for $i \in \{1, 2\}$.

Scheme	Message Space	Signature Size	Number of Pairings	Security Notion	Security Model	Underlying Assumption
[CKP+23]	iDH	$2\lvert\mathbb{G}_1\rvert$	$\ell + 2$	$\mathsf{TS\text{-}UF\text{-}0}$ (Static)	AGM+ ROM	GPS_3 (Interactive)
Ours	$\mathbb{G}_1^\ell$	$(3k+3)\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$5k + \ell + 6$	$\mathsf{TS\text{-}UF\text{-}1}$ (Adaptive)	Standard Model	$\mathcal{D}_k$-MDDH (Non-Interactive)

of non-linear operations prevents the reconstruction of the final signature from the partial signatures via Lagrange interpolation. These non-linear operations include the inversion of secret share keys (i.e., $[1/\mathsf{sk}_i]$), performing multiplication of distinct randomness and secret shares (i.e., $[r_i\mathsf{sk}_i]$), as well as raising either secret shares or distinct randomness to a power (e.g., $[\mathsf{sk}_i^\zeta]$ or $[r_i^\zeta]$ for any $\zeta > 1$). By employing an indexing approach, the authors in [CKP+23] were able to circumvent the need for multiplying randomness and secret keys, as required by Ghadafi's SPS [Gha16]. In contrast, in our proposed TSPS scheme, we adopt a distinct perspective for avoiding the non-linear operations.

We start from an observation regarding the SPS construction of Kiltz *et al.* [KPW15] which computes the first and second components of signature on a message $[\mathbf{m}]_1 \in \mathbb{G}_1^\ell$ as:

$$\text{KPW15}: \ (\sigma_1, \sigma_2) := \left(\underbrace{\left[\left(1\ \mathbf{m}^\top\right)\right]_1 \mathbf{K}}_{\text{SP-OTS}} + \overbrace{\mathbf{r}^\top \left[\mathbf{B}^\top(\mathbf{U} + \tau \cdot \mathbf{V})\right]_1, \left[\mathbf{r}^\top \mathbf{B}^\top\right]_1}^{\text{randomized PRF}} \right) ,$$

where τ is a fresh random integer and $\mathbf{r}$ is a fresh random vector of proper size.[5] Additionally, the secret signing and verification keys are defined as follows:

$$\begin{aligned} \text{KPW15}: \mathsf{sk} &:= \left(\mathbf{K}, \left[\mathbf{B}^\top \mathbf{U}\right]_1, \left[\mathbf{B}^\top \mathbf{V}\right]_1, [\mathbf{B}]_1\right) , \\ \mathsf{vk} &:= \left([\mathbf{K}\mathbf{A}]_2, [\mathbf{U}\mathbf{A}]_2, [\mathbf{V}\mathbf{A}]_2, [\mathbf{A}]_2\right) , \end{aligned}$$

where $\mathbf{K}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{U}$ and $\mathbf{V}$ are random matrices of appropriate dimensions.

As noted by Kiltz *et al.* in their work [KPW15], their SPS is build based on two fundamental primitives: (i) a structure-preserving one-time signature (SP-OTS), $(\left[\left(1\ \mathbf{m}^\top\right)\right]_1 \mathbf{K})$, and ($ii$) a randomized pseudorandom function (PRF), $(\mathbf{r}^\top \left[\mathbf{B}^\top(\mathbf{U} + \tau \cdot \mathbf{V})\right]_1, \left[\mathbf{r}^\top \mathbf{B}^\top\right]_1)$. In their proof of security, we observe that both the building blocks are involved in a loose manner. In particular, in most of their proofs, the reduction samples the SP-OTS signing key $\mathbf{K}$. It is easy to verify that this observation still holds even when they are arguing about the security of the randomized PRF. Our approach in this work is motivated by this fact which further inspires us to modify Kiltz et al.'s SPS. This adjustment involves defining the secret key as $\mathsf{sk} := \mathbf{K}$ and transferring the remaining parameters to the set of public parameters, i.e., $\mathsf{pp} := ([\mathbf{A}]_2, [\mathbf{U}\mathbf{A}]_2, [\mathbf{V}\mathbf{A}]_2, [\mathbf{B}]_1, [\mathbf{B}^\top\mathbf{U}]_1, [\mathbf{B}^\top\mathbf{V}]_1)$ and the verification is defined as $\mathsf{vk} := [\mathbf{K}\mathbf{A}]_2$. This rather simple structure allows to obtain the first TSPS for general message spaces in the standard model that can withhold adaptive corruptions without the exponential degradation [BCK+22] and can be proven secure in the $\mathsf{TS\text{-}UF\text{-}1}$ model.

Consider the following setting. Imagine there are n signers, each equipped with their own signing key, either obtained through the involvement of a trusted dealer or by conducting a Distributed Key Generation (DKG). Their collective objective is to generate a signature for a given message $[\mathbf{m}]_1 \in \mathbb{G}_1^\ell$. It is clear that the linear structure of the SP-OTS $\{\left[\left(1\ \mathbf{m}^\top\right)\right]_1 \mathbf{K}_i\}_{i \in S}$ allows for effortless aggregation when dealing with a collection of them over any subset $S \subseteq [1, n]$. Since the random quantities τ_i and $\mathbf{r}_i$ are independently sampled from a uniform distribution by each signer $i \in [1, n]$, aggregating the PRF

[5] Here we follow the group notation by Escala *et al.* [EHK+17]. See Definition 2 for more details.

elements is still challenging. Consequently, we must explore potential modifications needed to enable the aggregation of these components in comparison to Kiltz et al.'s SPS. We choose to make the tag τ dependent on the message. Thus, the randomized PRF computed by every signer, while still being a random element in the respective space, now allows aggregation. Moreover, by establishing an injective mapping between $[\mathbf{m}]_1$ and τ, we can observe that the randomized PRF structure still guarantees the unforgeability in [KPW15] when attempting to forge a signature on a distinct message. We employ a collision-resistant hash function (CRHF), $\mathcal{H}(.)$, to derive τ from $[\mathbf{m}]_1$. This gives the basis of our construction, where each signer $i \in [1, n]$ computes a partial signature on $[\mathbf{m}]_1$ as

$$(\sigma_1, \sigma_2) = \left(\left[\left(1\ \mathbf{m}^\top\right)\right]_1 \mathbf{K}_i + \mathbf{r}_i^\top \left[\mathbf{B}^\top(\mathbf{U} + \tau \cdot \mathbf{V})\right]_1, \left[\mathbf{r}_i^\top \mathbf{B}^\top\right]_1\right) .$$

Here the signer i is holding the secret share $\mathbf{K}_i$ and chooses a random quantity $\mathbf{r}_i$ of appropriate size and uses $\tau = \mathcal{H}([\mathbf{m}]_1)$. It is easy to verify that this signature can be aggregated in a non-interactive manner. Looking ahead, as a first step we prove that this construction achieves $\mathsf{TS\text{-}UF\text{-}0}$ security, relying on the well-established and non-interactive standard assumption, i.e., the MDDH assumption.

In case of a $\mathsf{TS\text{-}UF\text{-}1}$ adversary, we need to deal with the fact that the adversary is allowed to obtain partial signatures on the forged message $[\mathbf{m}^*]_1$. Let us first consider the case of static corruptions. We cannot apply the unforgeability of [KPW15] here as it did not consider strong $\mathsf{Uf\text{-}CMA}$ security.[6] To overcome this problem, we introduce an information theoretic step to argue that given a number of partial signatures on the forged message $[\mathbf{m}^*]_1$ below the threshold, the adversary does not gather extra information. In particular, we use Shamir's secret reconstruction security to ensure that partial signatures do not really leak much information. In this argument, we implicitly use the "selective security" of Shamir's secret sharing where all the parties in the corrupted set are fixed at the start of the game.

In the case of adaptive corruptions, an $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ adversary not only is allowed to obtain partial signatures on the forged message $[\mathbf{m}^*]_1$, but also it can corrupt different users to get the corresponding secret keys within the security game, adaptively. We obviously could follow a standard guessing argument to achieve $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ security based on $\mathsf{TS\text{-}UF\text{-}1}$ security. However, that direction unfortunately induces a significant security loss. We critically look at our proof of $\mathsf{TS\text{-}UF\text{-}1}$ security we have briefly discussed above. To make our construction $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ secure, we show that it is sufficient to argue that the underlying secret sharing achieves "adaptive security". In this work, we indeed form an argument that Shamir's secret sharing achieves "adaptive security" which in turn makes our construction $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ secure.

Next, we provide a brief intuition of the formal argument for the "adaptive security" of Shamir's secret sharing. Informally speaking, we produce a reduction $\mathcal{B}$ to break the "selective security" of Shamir's secret sharing given an adaptive adversary $\mathcal{A}$ of the secret sharing. Being an information theoretic reduction, $\mathcal{B}$ basically runs the adaptive adversary $\mathcal{A}$ an exponential number of times. Since $\mathcal{B}$ chooses the target set S independently of $\mathcal{A}$'s run, the expected number of parallel runs of $\mathcal{A}$ required to ensure all the parties whose secrets $\mathcal{A}$ queried are indeed from S is upper bounded by exponential. Being an information theoretically secure secret sharing scheme, Shamir's secret sharing basically achieves "adaptive security" due to complexity leveraging but without any degradation in the advantage of the adversary. While we use Shamir secret sharing as our canonical choice, we believe that all information-theoretically secure Linear Secret Sharing schemes can be used instead.

2 Preliminaries

Notation. Throughout the paper, we let $\kappa \in \mathbb{N}$ denote the security parameter and 1^κ as its unary representation. Given a polynomial $p(\cdot)$, an efficient randomized algorithm, $\mathcal{A}$, is called *probabilistic polynomial time*, PPT in short, if its running time is bounded by a polynomial $p(|x|)$ for every input x. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}^+$ is called *negligible* if for every positive polynomial $f(x)$, there exists x_0 such that for all $x > x_0$: $\mathsf{negl}(\kappa) < 1/f(x)$. If clear from the context, we sometimes omit κ for improved readability. The set $\{1, \ldots, n\}$ is denoted as $[1, n]$ for a positive integer n. For the equality check of two elements, we use "=". The assign operator is denoted with ":=", whereas the randomized assignment is denoted by $a \leftarrow A$, with a randomized algorithm A and where the randomness is not explicit. We use $\mathcal{D}_1 \approx_c \mathcal{D}_2$ to show two distributions like $\mathcal{D}_1$ and $\mathcal{D}_2$ are computationally indistinguishable.

[6] A signature is called strongly unforgeable when the adversary is not only incapable of producing a valid signature for a fresh message but also, it cannot generate a new signature for a challenge message M^*, by observing a valid signature for the same message M^*.

Definition 1 (Secret Sharing). *For any two positive integers $n, t < n$, an $(n,t)_{\mathbb{Z}_p^{a\times b}}$-secret-sharing scheme over $\mathbb{Z}_p^{a\times b}$ for $a, b \in \mathbb{N}$ consists of two functions* Share *and* Rec. Share *is a randomized function that takes a secret $\mathbf{M} \in \mathbb{Z}_p^{a\times b}$ and outputs $(\mathbf{M}_1, \ldots, \mathbf{M}_n) \leftarrow \mathsf{Share}(\mathbf{M}, \mathbb{Z}_p^{a\times b}, n, t)$ where $\mathbf{M}_i \in \mathbb{Z}_p^{a\times b}$ $\forall i \in [1,n]$. The pair of functions* $(\mathsf{Share}, \mathsf{Rec})$ *satisfy the following requirements.*

– ***Correctness:*** *For any secret $\mathbf{M} \in \mathbb{Z}_p^{a\times b}$ and a set of parties $\{i_1, i_2, \ldots, i_k\} \subseteq [1,n]$ such that $k \geq t$, we have*

$$\Pr[\mathsf{Rec}(\mathbf{M}_{i_1}, \ldots, \mathbf{M}_{i_k} : (\mathbf{M}_1, \ldots, \mathbf{M}_n) \leftarrow \mathsf{Share}(\mathbf{M}, \mathbb{Z}_p^{a\times b}, n, t)) = \mathbf{M}] = 1 \ .$$

– ***Security:*** *For any secret $\mathbf{M} \in \mathbb{Z}_p^{a\times b}$ and a set of parties $S \subseteq [1,n]$ such that $|S| = k < t$, for all information-theoretic adversary $\mathcal{A}$ we have*

$$\Pr\left[S = \{i_i\}_{i\in[1,k]} \wedge \mathbf{M}^* = \mathbf{M} \,\middle|\, \begin{array}{l} (\mathbf{M}_1, \ldots, \mathbf{M}_n) \leftarrow \mathsf{Share}(\mathbf{M}, \mathbb{Z}_p^{a\times b}, n, t) \\ S \leftarrow \mathcal{A}() \\ \mathbf{M}^* \leftarrow \mathcal{A}(\mathbf{M}_{i_1}, \ldots, \mathbf{M}_{i_k}) \end{array} \right] = 1/p \ .$$

We follow standard nomenclature to call this "selective security". In case of "adaptive security", $\mathcal{A}$ adaptively chooses $i_j \in [1,n]$ to get $\mathbf{M}_{i_j}$ one at a time.

We briefly recall the well-known secret sharing scheme due to Shamir [Sha79]. In (n,t)-Shamir Secret Sharing, a secret s is shared to n parties via n evaluations of a polynomial of degree $(t-1)$. Reconstruction of the secret is essentially Lagrange interpolation where one computes Lagrange polynomials $\{\lambda_{i_j}(x)\}_{j\in S}$ and linearly combine them with the given polynomial evaluations. The degree of the original polynomial confirms that one needs at least $|S| = t$ many polynomial evaluations. In this work, we use Shamir Secret Sharing to secret share a matrix of size $a \times b$, i.e., we use ab-many parallel instances of Shamir Secret Sharing. To keep our exposition simpler, we however assume that we have an (n,t)-Shamir Secret Sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ which operates on matrices. Since, our work here uses Shamir Secret Sharing quite generically, it is convenient to make such abstraction without going into the details.

Definition 2 (Bilinear Groups). *Let an asymmetric bilinear group generator,* $\mathsf{ABSGen}(1^\kappa)$*, that returns a tuple $\mathcal{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{P}_1, \mathsf{P}_2, e)$, such that $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of the same prime order p such that there is no known homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. P_1 and P_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, where $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerate) bilinear map with the following properties:*

- $\forall\, a, b \in \mathbb{Z}_p,\ e([a]_1, [b]_2) = [ab]_T = e([b]_1, [a]_2)$,
- $\forall\, a, b \in \mathbb{Z}_p,\ e([a+b]_1, [1]_2) = e([a]_1, [1]_2)e([b]_1, [1]_2)$,

where we use an implicit representation of group elements, in which for $\zeta \in \{1, 2, T\}$ and an integer $\alpha \in \mathbb{Z}_p$, the implicit representation of integer α in group $\mathbb{G}_\zeta$ is defined by $[\alpha]_\zeta = \alpha \mathsf{P}_\zeta \in \mathbb{G}_\zeta$, where $\mathsf{P}_T = e(\mathsf{P}_1, \mathsf{P}_2)$. To be more general, the implicit representation of a matrix $\mathbf{A} = (\alpha_{ij}) \in \mathbb{Z}_p^{m\times n}$ in $\mathbb{G}_\zeta$ is defined by $[\mathbf{A}]_\zeta$ and we have:

$$[\mathbf{A}]_\zeta = \begin{pmatrix} \alpha_{1,1}\mathsf{P}_\zeta & \cdots & \alpha_{1,n}\mathsf{P}_\zeta \\ \alpha_{2,1}\mathsf{P}_\zeta & \cdots & \alpha_{2,n}\mathsf{P}_\zeta \\ \vdots & \ddots & \vdots \\ \alpha_{m,1}\mathsf{P}_\zeta & \cdots & \alpha_{m,n}\mathsf{P}_\zeta \end{pmatrix} .$$

For two matrices $\mathbf{A}$ and $\mathbf{B}$ with matching dimensions we define $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$.

Definition 3 (Matrix Distribution). *Let $k, \ell \in \mathbb{N}^*$ s.t. $k < \ell$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs matrices over $\mathbb{Z}_p^{\ell\times k}$ of full rank k in polynomial time. W.l.o.g, we assume the first k rows of matrix $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}$ form an invertible matrix. For $\ell = k+1$, we write $\mathcal{D}_k$ in short.*

Next, we recall the Matrix Decisional Diffie-Hellman assumption, which defines over $\mathbb{G}_\zeta$ for any $\zeta = \{1, 2\}$ and states two distributions $([\mathbf{A}]_\zeta, [\mathbf{Ar}]_\zeta)$ and $([\mathbf{A}]_\zeta, [\mathbf{u}]_\zeta)$, where $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}, \mathbf{r} \leftarrow \mathbb{Z}_p^k, \mathbf{u} \leftarrow \mathbb{Z}_p^\ell$ are computationally indistinguishable.

Definition 4 ($\mathcal{D}_{\ell,k}$-Matrix Decisional Diffie-Hellman ($\mathcal{D}_{\ell,k}$-MDDH) Assumption [EHK⁺17]). *For a given security parameter κ, let $k, \ell \in \mathbb{N}^*$ s.t. $k < \ell$ and $\mathcal{D}_{\ell,k}$ be a matrix distribution, defined in Definition 3. We say $\mathcal{D}_{\ell,k}$-MDDH assumption over $\mathbb{G}_\zeta$ for $\zeta = \{1, 2\}$ holds, if for all PPT adversaries $\mathcal{A}$ we have:*

$$Adv^{\mathsf{MDDH}}_{\mathcal{D}_{\ell,k},\mathbb{G}_\zeta,\mathcal{A}}(\kappa) = \Big| \Pr\left[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_\zeta, [\mathbf{Ar}]_\zeta) = 1\right] - \Pr\left[\mathcal{A}(\mathcal{G}, [\mathbf{A}]_\zeta, [\mathbf{u}]_\zeta) = 1\right] \Big| \leq \mathsf{negl}(\kappa) \ ,$$

where $\mathcal{G} \leftarrow \mathsf{ABSGen}(1^\kappa)$, $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}, \mathbf{r} \leftarrow \mathbb{Z}_p^k$ and $\mathbf{u} \leftarrow \mathbb{Z}_p^\ell$.

Definition 5 ($\mathcal{D}_k$-Kernel Matrix Diffie-Hellman ($\mathcal{D}_k$-KerMDH) Assumption [MRV16]). *For a given security parameter κ, let $k \in \mathbb{N}^*$ and $\mathcal{D}_k$ is a matrix distribution, defined in Definition 3. We say $\mathcal{D}_k$-KerMDH assumption over $\mathbb{G}_\zeta$ for $\zeta = \{1, 2\}$ holds, if for all PPT adversaries $\mathcal{A}$ we have:*

$$Adv^{\mathsf{KerMDH}}_{\mathcal{D}_k,\mathbb{G}_\zeta,\mathcal{A}}(\kappa) = \Pr\left[\mathbf{c} \in \mathsf{orth}(\mathbf{A}) \mid [\mathbf{c}]_{3-\zeta} \leftarrow \mathcal{A}(\mathcal{G}, [\mathbf{A}]_\zeta))\right] \leq \mathsf{negl}(\kappa) \cdot$$

The Kernel Matrix Diffie-Hellman assumption is a natural computational analog of the MDDH assumption. It is well-known that for all $k \geq 1$, $\mathcal{D}_k$-$\mathsf{MDDH} \Rightarrow \mathcal{D}_k$-$\mathsf{KerMDH}$ [KPW15, MRV16].

3 Threshold Structure-Preserving Signatures

In this section, we first present our security model for Threshold Structure-Preserving Signatures (TSPS) and then present our construction and prove its security.

3.1 TSPS: Syntax and Security Definitions

First, we recall the definition of the Threshold Structure-Preserving Signatures (TSPS) from [CKP⁺23] and their main security properties: correctness and threshold unforgeability. Informally, a threshold signature scheme enables a group of servers S of size n to collaboratively sign a message. In this paper, we assume the existence of a trusted dealer who shares the secret key among the signers. However, there are straightforward and well-known techniques in particular distributed key generation (DKG) protocols (e.g., [Ped92]) that eliminate this needed trust.

Definition 6 (Threshold Structure-Preserving Signatures [CKP⁺23]). *Over a security parameter κ and a bilinear group, an (n,t)-TSPS contains the following PPT algorithms:*

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$*: The setup algorithm takes the security parameter κ as input and returns the set of public parameters* pp *as output.*
- $(\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[1,n]}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, n, t)$*: The key generation algorithm takes the public parameters* pp *along with two integers n, t s.t. $1 \leq t \leq n$ as inputs. It then returns secret/verification keys* $(\mathsf{sk}_i, \mathsf{vk}_i)$ *for $i \in [1, n]$ along with a global verification key* vk *as output.*
- $\Sigma_i \leftarrow \mathsf{ParSign}(\mathsf{pp}, \mathsf{sk}_i, [\mathbf{m}])$*: The partial signing algorithm takes* pp*, the i^{th} party's secret key,* sk_i*, and a message $[\mathbf{m}] \in \mathcal{M}$ as inputs. It then returns a partial signature Σ_i as output.*
- $0/1 \leftarrow \mathsf{ParVerify}(\mathsf{pp}, \mathsf{vk}_i, [\mathbf{m}], \Sigma_i)$*: The partial verification algorithm as a deterministic algorithm, takes* pp*, the i^{th} verification key,* vk_i*, and a message $[\mathbf{m}] \in \mathcal{M}$ along with partial signature Σ_i as inputs. It then returns 1 (accept), if the partial signature is valid and 0 (reject), otherwise.*
- $\Sigma \leftarrow \mathsf{CombineSign}(\mathsf{pp}, T, \{\Sigma_i\}_{i\in T})$*: The combine algorithm takes a set of partial signatures Σ_i for $i \in T$ along with $T \subseteq [1, n]$ and then returns an aggregated signature Σ as output.*
- $0/1 \leftarrow \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}], \Sigma)$*: The verification algorithm as a deterministic algorithm, takes* pp*, the global verification key,* vk*, and message $[\mathbf{m}] \in \mathcal{M}$ along with an aggregated signature Σ as inputs. It then returns 1 (accept), if the aggregated signature is valid and 0 (reject), otherwise.*

Correctness. Correctness guarantees that a signature obtained from a set $T \subseteq [1, n]$ s.t. $|T| \geq t$ of honest signers always verifies.

Definition 7 (Correctness). *An (n,t)-TSPS scheme is called correct if we have:*

$$\Pr\begin{bmatrix} \forall\ \mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa), (\{\mathsf{sk}_i, \mathsf{vk}_i\}_{i\in[1,n]}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, n, t), [\mathbf{m}] \in \mathcal{M}, \\ \Sigma_i \leftarrow \mathsf{ParSign}(\mathsf{pp}, \mathsf{sk}_i, [\mathbf{m}]) \text{ for } i \in [1, n], \forall\ T \subseteq [1, n], |T| \geq t, \\ \Sigma \leftarrow \mathsf{CombineSign}\left(\mathsf{pp}, T, \{\Sigma_i\}_{i\in T}\right) : \mathsf{Verify}\left(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}], \Sigma\right) = 1 \end{bmatrix} = 1 \ .$$

Unforgeability. Our security model for threshold unforgeability extends the one from Crites et al. [CKP+23]. Therefore, we need to recall a recent work by Bellare et al. [BCK+22], which investigates existing security notions and proposes stronger and more realistic security notions for threshold signatures under static corruptions. In particular, the authors in [BCK+22] present a hierarchy of different notions of security for non-interactive schemes. We focus on fully non-interactive schemes, i.e., ones that do not require one round of pre-processing, and thus in this paper only the TS-UF-0 and TS-UF-1 notions are relevant. The TS-UF-0 notion is a less stringent notion of unforgeability. In this context, if the adversary has previously seen a partial signature on a challenge message $[\mathbf{m}^*]$, the act of forging a signature for that specific message is considered as a trivial forgery. The security of the original TSPS is proved under this notion of unforgeability.

The stronger TS-UF-1 notion, which is our main focus, allows adversaries to query the signing oracle up to $t - |\mathsf{CS}|$ times for partial signatures, even on the challenge message. Here CS with $|\mathsf{CS}| < t$ denotes the set of (statically corrupted) signers. Moreover, the model in [BCK+22] as well as the TSPS construction in [CKP+23] only considers static corruptions. But we also integrate the core elements of the model introduced in the recent work by Crites et al. [CKM23], adapted to fully non-interactive schemes, to support fully adaptive corruptions. Our model is depicted in Figure 1. The dashed box as well as the solid white box in the winning condition apply to the TS-UF-0 and TS-UF-1 notions, respectively. Grey boxes are only present in the adaptive version of the game, i.e., adp-TS-UF-0 and adp-TS-UF-1.

Definition 8 (Threshold Unforgeability). *Let* $\mathsf{TSPS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{ParSign}, \mathsf{ParVerify}, \mathsf{CombineSign}, \mathsf{Verify})$ *be an* (n, t)*-TSPS scheme over message space* $\mathcal{M}$ *and let* $\mathsf{prop} \in \{\mathsf{TS\text{-}UF\text{-}b}, \mathsf{adp\text{-}TS\text{-}UF\text{-}b}\}_{\mathsf{b}\in\{0,1\}}$. *The advantage of a PPT adversary* $\mathcal{A}$ *playing described security games in Figure 1, is defined as,*

$$\mathbf{Adv}^{\mathsf{prop}}_{\mathsf{TSPS},\mathcal{A}}(\kappa) = \Pr\left[\mathbf{G}^{\mathsf{prop}}_{\mathsf{TS},\mathcal{A}}(\kappa) = 1\right] \quad .$$

A TSPS achieves prop-security if we have, $\mathbf{Adv}^{\mathsf{prop}}_{\mathsf{TSPS},\mathcal{A}}(\kappa) \le \mathsf{negl}(\kappa)$.

```
$G^{\mathsf{TS\text{-}UF\text{-}0}}_{\mathsf{TS},\mathcal{A}}(\kappa)$, $\boxed{G^{\mathsf{TS\text{-}UF\text{-}1}}_{\mathsf{TS},\mathcal{A}}(\kappa)}$, $G^{\mathsf{adp\text{-}TS\text{-}UF\text{-}0}}_{\mathsf{TS},\mathcal{A}}(\kappa)$, $\boxed{G^{\mathsf{adp\text{-}TS\text{-}UF\text{-}1}}_{\mathsf{TS},\mathcal{A}}(\kappa)}$:
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$
$(n, t, \mathsf{CS}, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathsf{pp})$
$\mathsf{HS} := [1, n] \setminus \mathsf{CS}$
$(\mathsf{vk}, \{\mathsf{sk}_i\}_{i\in[1,n]}, \{\mathsf{vk}_i\}_{i\in[1,n]}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, n, t)$
$([\mathbf{m}^*], \Sigma^*, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{PSign}}(.),\ \mathcal{O}^{\mathsf{Corrupt}}(.)}(\mathsf{st}_0, \mathsf{vk}, \{\mathsf{sk}_i\}_{i\in\mathsf{CS}}, \{\mathsf{vk}_i\}_{i\in[1,n]})$
return $\Big(\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*], \Sigma^*) \wedge |\mathsf{CS}| < t \wedge$
        $(S_1([\mathbf{m}^*]) = \emptyset \vee \boxed{|S_1([\mathbf{m}^*])| < t - |\mathsf{CS}|})\Big)$

$\mathcal{O}^{\mathsf{PSign}}(i, [\mathbf{m}])$:
Assert $([\mathbf{m}] \in \mathcal{M} \wedge i \in \mathsf{HS})$
$\Sigma_i \leftarrow \mathsf{ParSign}(\mathsf{pp}, \mathsf{sk}_i, [\mathbf{m}])$
if $\Sigma_i \neq \bot$ :
    $S_1([\mathbf{m}]) \leftarrow S_1([\mathbf{m}]) \cup \{i\}$
return $(\Sigma_i)$

$\mathcal{O}^{\mathsf{Corrupt}}(k)$:
if $k \in \mathsf{CS}$ :
    return $\bot$
else : $\mathsf{CS} \leftarrow \mathsf{CS} \cup \{k\}$
    $\mathsf{HS} \leftarrow \mathsf{HS} \setminus \{k\}$
    return $(\mathsf{sk}_k)$
```

Fig. 1. Games defining the TS-UF-0, $\boxed{\text{TS-UF-1}}$, adp-TS-UF-0, and $\boxed{\text{adp-TS-UF-1}}$ unforgeability notions of threshold signatures.

3.2 Core Lemma

Prior to introducing our construction, we first present the core lemma that forms a basis in the proofs of our proposed TSPS. It extends the core lemmas from [KW15, KPW15], however it is important to note that both of these schemes are standard SPS, where there was no need to simulate signatures on forged messages. In contrast, both the TS-UF-1 and adp-TS-UF-1 security models necessitate the simulation of partial signature queries on forged messages. Thus we define our core lemma with a key difference being the introduction of a new oracle, denoted as $\mathcal{O}^{**}(\cdot)$.

Lemma 1 (Core Lemma). *Let the game $\mathbf{G}^{\mathsf{Core}}_{\mathcal{D}_k,\mathsf{ABSGen}}(\kappa)$ be defined as Figure 2. For any adversary $\mathcal{A}$ with the advantage of $Adv^{\mathsf{Core}}_{\mathcal{D}_k,\mathsf{ABSGen},\mathcal{A}}(\kappa) := |\Pr[\mathbf{G}^{\mathsf{Core}}_{\mathcal{D}_k,\mathsf{ABSGen}}(\kappa)] - 1/2|$, there exists an adversary $\mathcal{B}$ against the $\mathcal{D}_k$-MDDH assumption such that with the running time $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B})$ it holds that*

$$Adv^{\mathsf{Core}}_{\mathcal{D}_k,\mathsf{ABSGen},\mathcal{A}}(\kappa) \leq 2q Adv^{\mathsf{MDDH}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}}(\kappa) + q/p ,$$

where q is a bound on the number of queries requested by adversary $\mathcal{A}$ for oracle $\mathcal{O}_b(\cdot)$. Note that $\mathcal{A}$ can only query the other oracles only once.

```
Init():
    $\mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k,\ \mathbf{U}, \mathbf{V} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$
    $\mathsf{vk} := (\mathbf{A}, \mathbf{UA}, \mathbf{VA}, [\mathbf{B}]_1, [\mathbf{B}^\top\mathbf{U}]_1, [\mathbf{B}^\top\mathbf{V}]_1)$
    $b \leftarrow \{0,1\}$
    Let $\mathbf{a}^\perp \leftarrow \mathbb{Z}_p^{1\times(k+1)}$ such that $\mathbf{a}^\perp\mathbf{A} = \mathbf{0}$
    $q := 0,\ \mathcal{Q}_{\mathsf{tag}} := \emptyset$
    return vk

$\mathcal{O}^*([\tau^*]_2)$:
    return $[\mathbf{U} + \tau^*\mathbf{V}]_2$

$\mathcal{O}^{**}([\tau^*]_1)$:
    return $\left[\mathbf{B}^\top(\mathbf{U} + \tau^*\mathbf{V})\right]_1$

$\mathcal{O}_b([\tau]_1)$:
    $\mu \leftarrow \mathbb{Z}_p, \mathbf{r} \leftarrow \mathbb{Z}_p^k,\ q := q + 1$
    $\mathcal{Q}_{\mathsf{tag}} := \mathcal{Q}_{\mathsf{tag}} \cup \{\tau\}$
    return $\left(\left[b\mu\mathbf{a}^\perp + \mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})\right]_1, \left[\mathbf{r}^\top\mathbf{B}^\top\right]_1\right)$
```

Fig. 2. Game defining the core lemma, $\mathbf{G}^{\mathsf{Core}}_{\mathcal{D}_k,\mathsf{ABSGen}}(\kappa)$.

Proof Sketch. The proof of this lemma uses the proof of core lemma in [KW15, KPW15]. The fundamental concept of these proofs is primarily an information-theoretic argument that $(\mathbf{t}^\top(\mathbf{U} + \tau\mathbf{V}), \mathbf{U} + \tau^*\mathbf{V})$ is identically distributed to $(\mu{\mathbf{a}^\perp}^\top + \mathbf{t}^\top(\mathbf{U} + \tau\mathbf{V}), \mathbf{U} + \tau^*\mathbf{V})$ for $\mu \leftarrow \mathbb{Z}_p$, $\mathbf{a}^\perp, \mathbf{t} \leftarrow \mathbb{Z}_p^{k+1}$ and $\tau \neq \tau^*$. We use $\left[b\mu{\mathbf{a}^\perp}^\top + \mathbf{t}^\top(\mathbf{U} + \tau\mathbf{V})\right]_1$ to simulate $\mathcal{O}_b([\tau]_1)$, $[\mathbf{U} + \tau^*\mathbf{V}]_2$ to simulate $\mathcal{O}^*([\tau^*]_2)$ and $\left[\mathbf{B}^\top(\mathbf{U} + \tau^*\mathbf{V})\right]_1$ to simulate $\mathcal{O}^{**}([\tau^*]_1)$. The detailed proof can be found in Section 3.5. □

3.3 Our Threshold SPS Construction

Given a collision resistant hash function, $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$, and message space $\mathcal{M} := \mathbb{G}_1^\ell$, we present our (n,t)-TSPS construction in Figure 3. This consists of six main PPT algorithms – Setup, KeyGen, ParSign, ParVerify, CombineSign and Verify, as defined in Definition 6. Similar to the settings of Bellare *et al.* [BCK$^+$22], we also assume there is a dealer who is responsible for generating key pairs for all signers and a general verification key.

3.4 Security

Theorem 1. *Under the $\mathcal{D}_k$-MDDH Assumption in $\mathbb{G}_1$ and $\mathcal{D}_k$-KerMDH Assumption in $\mathbb{G}_2$, the proposed Threshold Structure-Preserving Signature construction in Figure 3 achieves TS-UF-0 security against an efficient adversary making at most q partial signature queries.*

```
$\underline{\mathsf{Setup}(1^\kappa)}$:
    $\mathcal{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{P}_1, \mathsf{P}_2, e) \leftarrow \mathsf{ABSGen}(1^\kappa)$.
    $\mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k$, $\mathbf{U}, \mathbf{V} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$.
    $\mathsf{pp} := \left([\mathbf{A}]_2, [\mathbf{UA}]_2, [\mathbf{VA}]_2, [\mathbf{B}]_1, \left[\mathbf{B}^\top\mathbf{U}\right]_1, \left[\mathbf{B}^\top\mathbf{V}\right]_1\right)$.
$\underline{\mathsf{KeyGen}(\mathsf{pp}, n, t)}$:
    $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times(k+1)}$.
    $\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)$.
    Set $\mathsf{vk} := [\mathbf{KA}]_2$ and $(\mathsf{sk}_i, \mathsf{vk}_i) := (\mathbf{K}_i, [\mathbf{K}_i\mathbf{A}]_2)$.
$\underline{\mathsf{ParSign}(\mathsf{pp}, \mathsf{sk}_i, [\mathbf{m}]_1)}$:
    $\mathbf{r}_i \leftarrow \mathbb{Z}_p^k$.
    $\tau := \mathcal{H}([\mathbf{m}]_1)$.
    Output $\Sigma_i := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ s.t.
    $\sigma_1 := \left[\begin{pmatrix} 1 & \mathbf{m}^\top \end{pmatrix}\right]_1 \mathbf{K}_i + \mathbf{r}_i^\top \left[\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})\right]_1$ ,
    $\sigma_2 := \left[\mathbf{r}_i^\top \mathbf{B}^\top\right]_1$ ,
    $\sigma_3 := \left[\tau\mathbf{r}_i^\top \mathbf{B}^\top\right]_1$ ,
    $\sigma_4 := [\tau]_2$ .
$\underline{\mathsf{ParVerify}(\mathsf{pp}, \mathsf{vk}_i, [\mathbf{m}]_1, \Sigma_i)}$: Output 1 if the following checks hold; else output 0.
    $e(\sigma_1, [\mathbf{A}]_2) = e\left(\left[\begin{pmatrix} 1 & \mathbf{m}^\top \end{pmatrix}\right]_1, \mathsf{vk}_i\right) \cdot\ e(\sigma_2, [\mathbf{UA}]_2) \cdot\ e(\sigma_3, [\mathbf{VA}]_2)$ .
    $e(\sigma_2, \sigma_4) = e(\sigma_3, [1]_2)$.
$\underline{\mathsf{CombineSign}(\mathsf{pp}, S, \{\Sigma_i\}_{i\in S})}$:
    Parse $\Sigma_i = (\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}, \sigma_4)$ for all $i \in S$.
    Compute Lagrange polynomials $\lambda_i$ for $i \in S$.
    Output $\Sigma := (\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4)$ s.t.
    $\widehat{\sigma}_1 := \prod_{i\in S} \sigma_{i,1}^{\lambda_i} = \left[\begin{pmatrix} 1 & \mathbf{m}^\top \end{pmatrix} \sum_{i\in S} \lambda_i \mathbf{K}_i\right]_1 + \sum_{i\in S} \lambda_i \mathbf{r}_i^\top \left[\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})\right]_1 = \left[\begin{pmatrix} 1 & \mathbf{m}^\top \end{pmatrix} \mathbf{K}\right]_1 +$
    $\mathbf{r}^\top \left[\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})\right]_1$ ,
    $\widehat{\sigma}_2 := \prod_{i\in S} \sigma_{i,2}^{\lambda_i} = \left[\sum_{i\in S} \lambda_i \mathbf{r}_i^\top \mathbf{B}^\top\right]_1 = \left[\mathbf{r}^\top \mathbf{B}^\top\right]_1$ ,
    $\widehat{\sigma}_3 := \prod_{i\in S} \sigma_{i,3}^{\lambda_i} = \left[\sum_{i\in S} \tau\lambda_i \mathbf{r}_i^\top \mathbf{B}^\top\right]_1 = \left[\tau\mathbf{r}^\top \mathbf{B}^\top\right]_1$ ,
    $\widehat{\sigma}_4 := \sigma_4$ .
$\underline{\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}]_1, \Sigma)}$: Output 1 if the following checks satisfy; else output 0.
    $e(\widehat{\sigma}_1, [\mathbf{A}]_2) = e\left(\left[\begin{pmatrix} 1 & \mathbf{m}^\top \end{pmatrix}\right]_1, \mathsf{vk}\right) \cdot\ e(\widehat{\sigma}_2, [\mathbf{UA}]_2) \cdot\ e(\widehat{\sigma}_3, [\mathbf{VA}]_2)$ .
    $e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)$ .
```

Fig. 3. Our proposed TSPS construction.

Proof. We prove the above theorem through a series of games and we use $\mathbf{Adv}_i$ to denote the advantage of the adversary $\mathcal{A}$ in winning the Game i. The games are described below.

Game 0. This is the TS-UF-0 security game described in Definition 8. As shown in Figure 4, an adversary $\mathcal{A}$ after receiving the set of public parameters, pp, returns (n, t, CS), where n, t and CS represents the total number of signers, the threshold, and the set of corrupted signers, respectively. The adversary can query the partial signing oracle $\mathcal{O}^{\mathsf{PSign}}(\cdot)$ to receive partial signatures and q represents the total number of these queries. In the end, the adversary outputs a message $[\mathbf{m}^*]_1$ and a forged signature Σ^*.

```
$\boldsymbol{G}_0(\kappa)$:
$\mathcal{G} \leftarrow \mathsf{ABSGen}(1^\kappa)$,
$\mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k$,
$\mathbf{U}, \mathbf{V} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$.
$\mathsf{pp} := ([\mathbf{A}]_2, [\mathbf{UA}]_2, [\mathbf{VA}]_2, [\mathbf{B}]_1, [\mathbf{B}^\top\mathbf{U}]_1, [\mathbf{B}^\top\mathbf{V}]_1)$.
$(n, t, \mathsf{CS}, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathsf{pp})$.
Assert $\mathsf{CS} \subset [1, n]$.
Sample $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times(k+1)}$.
$(\mathbf{K}_1, \ldots, \mathbf{K}_n) \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)$.
$\mathsf{vk} := [\mathbf{KA}]_2$.
for $i \in [1, n]$:
    $\mathsf{sk}_i := \mathbf{K}_i$, $\mathsf{vk}_i := [\mathbf{K}_i\mathbf{A}]_2$.
$([\mathbf{m}^*]_1, \Sigma^*, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{PSign}(.)}}\big(\mathsf{st}_0, \mathsf{vk}, \{\mathsf{sk}_i\}_{i\in\mathsf{CS}}, \{\mathsf{vk}_i\}_{i\in[1,n]}\big)$.
return $(\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \Sigma^*) \wedge |\mathsf{CS}| < t \wedge S_1([\mathbf{m}^*]_1) = \emptyset)$

$\mathcal{O}^{\mathsf{PSign}}(i, [\mathbf{m}]_1)$:
Assert $\big([\mathbf{m}]_1 \in \mathcal{M} \wedge i \in \mathsf{HS}\big)$.
$\mathbf{r}_i \leftarrow \mathbb{Z}_p^k$.
$\tau := \mathcal{H}([\mathbf{m}]_1)$.
$\sigma_1 := \left[\begin{pmatrix}1 & \mathbf{m}^\top\end{pmatrix}\mathbf{K}_i + \mathbf{r}_i^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})\right]_1$,
$\sigma_2 := [\mathbf{r}_i^\top\mathbf{B}^\top]_1$,
$\sigma_3 := [\tau\mathbf{r}_i^\top\mathbf{B}^\top]_1$,
$\sigma_4 := [\tau]_2$.
$\Sigma_i := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.
if $\Sigma_i \neq \bot$ :
    $S_1([\mathbf{m}]_1) := S_1([\mathbf{m}]_1) \cup \{i\}$.
return $\Sigma_i$

$\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \Sigma^*)$ :
Parse $\Sigma^*$ as $(\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4)$.
return $\Big(e(\widehat{\sigma}_1, [\mathbf{A}]_2) = e\Big(\Big[\begin{pmatrix}1 & \mathbf{m}^{*\top}\end{pmatrix}\Big]_1, [\mathbf{KA}]_2\Big) \cdot e(\widehat{\sigma}_2, [\mathbf{UA}]_2) \cdot e(\widehat{\sigma}_3, [\mathbf{VA}]_2)$
        $\wedge\ e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)\Big)$
```

Fig. 4. Game_0.

Game 1. We modify the verification procedure to the one described in Figure 5. Consider any forged message/signature pair $([\mathbf{m}^*]_1, \Sigma^* = (\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4))$, where $e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)$, $|\mathsf{CS}| < t$ and

$S_1([\mathbf{m}^*]_1) = \emptyset$. It is easy to observe that if the pair $([\mathbf{m}^*]_1, \varSigma^*)$ meets the $\mathsf{Verify}^*(\cdot)$ criteria, outlined in Figure 5, it also satisfies $\mathsf{Verify}(\cdot)$ procedure, described in Figure 4. This is primarily due to the fact that:

$$\begin{aligned} & e(\widehat{\sigma}_1, [\mathbf{A}]_2) = e\left([\left(1\ \mathbf{m}^{*\top}\right)]_1, [\mathbf{K}\mathbf{A}]_2\right) \cdot e(\widehat{\sigma}_2, [\mathbf{U}\mathbf{A}]_2) \cdot e(\widehat{\sigma}_3, [\mathbf{V}\mathbf{A}]_2) \\ \Longleftarrow\ & e(\widehat{\sigma}_1, [1]_2) = e([\left(1\ \mathbf{m}^{*\top}\right)]_1, [\mathbf{K}]_2) \cdot e(\widehat{\sigma}_2, [\mathbf{U}]_2) \cdot e(\widehat{\sigma}_3, [\mathbf{V}]_2) \\ \Longleftrightarrow & e(\widehat{\sigma}_1, [1]_2) = e([\left(1\ \mathbf{m}^{*\top}\right)\mathbf{K}]_1, [1]_2) \cdot e(\widehat{\sigma}_2, [\mathbf{U} + \tau^*\mathbf{V}]_2) \cdot \end{aligned}$$

Assume there exists a message/signature pair like $([\mathbf{m}^*]_1, \varSigma^* = (\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4))$ that satisifies $\mathsf{Verify}(\cdot)$ and not $\mathsf{Verify}^*(\cdot)$, then we can compute a non-zero vector $\mathbf{c}$ in the kernal of $\mathbf{A}$ as follows:

$$\mathbf{c} := \widehat{\sigma}_1 - ([\left(1\ \mathbf{m}^{*\top}\right)\mathbf{K}]_1 + \widehat{\sigma}_2\mathbf{U} + \widehat{\sigma}_3\mathbf{V}) \in \mathbb{G}_1^{1\times(k+1)} \cdot$$

According to $\mathcal{D}_k$-KerMDH assumption over $\mathbb{G}_2$ described in Definition 5, computing such a vector $\mathbf{c}$ is considered computationally hard. Thus,

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq Adv^{\mathsf{KerMDH}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_0}(\kappa) \cdot$$

```
$\mathsf{Verify}^*(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \varSigma^*)$:
Parse $\varSigma^*$ as $(\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4 = [\tau^*]_2)$.
return $\Big(e(\widehat{\sigma}_1, [1]_2) = e\left([\left(1\ \mathbf{m}^{*\top}\right)\mathbf{K}]_1, [1]_2\right) \cdot e(\widehat{\sigma}_2, [\mathbf{U} + \tau^*\mathbf{V}]_2) \wedge$
          $e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)\Big)$
```

Fig. 5. Modifications in Game_1.

Game 2. On receiving a partial signature query on a message $[\mathbf{m}_i]_1$, the query list is updated to include the message $[\mathbf{m}_i]_1$ along with its corresponding tag, $\tau_i := \mathcal{H}([\mathbf{m}_i]_1)$. The challenger aborts if an adversary can generate two tuples $([\mathbf{m}_i]_1, \tau_i)$, $([\mathbf{m}_j]_1, \tau_j)$ with $[\mathbf{m}_i]_1 \neq [\mathbf{m}_j]_1$ and $\tau_i = \tau_j$. By the collision resistance property of the underlying hash function we have,

$$|\mathbf{Adv}_1 - \mathbf{Adv}_2| \leq Adv^{\mathsf{CRHF}}_{\mathcal{H}}(\kappa) \cdot$$

Game 3. In this game, we introduce randomness to the partial signatures by adding $\mu\mathbf{a}^\perp$ to each partial signature, where μ is chosen uniformly at random and the vector $\mathbf{a}^\perp$ is a non-zero vector in the kernel of $\mathbf{A}$. The new partial signatures satisfy the verification procedure as $\mathbf{a}^\perp\mathbf{A} = \mathbf{0}$. Figure 6 describes the new partial signing oracle, $\mathcal{O}^{\mathsf{PSign}^*}(.)$.

```
$\mathcal{O}^{\mathsf{PSign}^*}(i, [\mathbf{m}]_1)$:
Assert $\left([\mathbf{m}]_1 \in \mathcal{M}\ \wedge\ i \in \mathsf{HS}\right)$.
$\mathbf{r}_i \leftarrow \mathbb{Z}_p^k, \tau := \mathcal{H}([\mathbf{m}]_1), \mu \leftarrow \mathbb{Z}_p$.
$\sigma_1 := [\left(1\ \mathbf{m}^\top\right)\mathbf{K}_i + \mu\mathbf{a}^\perp + \mathbf{r}_i^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1,$
$\sigma_2 := [\mathbf{r}_i^\top\mathbf{B}^\top]_1$ ,
$\sigma_3 := [\tau\mathbf{r}_i^\top\mathbf{B}^\top]_1$ ,
$\sigma_4 := [\tau]_2$ .
$\varSigma_i := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.
if $\varSigma_i \neq \perp$ :
    $S_1([\mathbf{m}]_1) := S_1([\mathbf{m}]_1) \cup \{i\}$.
return $\varSigma_i$
```

Fig. 6. Modifications in Game_3.

```
$\mathcal{B}_1^{\mathsf{Init}(\cdot),\mathcal{O}_b(\cdot),\mathcal{O}^*(\cdot),\mathcal{O}^{**}(\cdot)}$:
  Assert $\big([\mathbf{m}]_1 \in \mathcal{M} \ \wedge \ i \in \mathsf{HS}\big)$.
  $(\mathbf{A}, \mathbf{UA}, \mathbf{VA}, [\mathbf{B}]_1, [\mathbf{B}^\top\mathbf{U}]_1, [\mathbf{B}^\top\mathbf{V}]_1) \leftarrow \mathsf{Init}()$.
  $\mathsf{pp} := ([\mathbf{A}]_2, [\mathbf{UA}]_2, [\mathbf{VA}]_2, [\mathbf{B}]_1, [\mathbf{B}^\top\mathbf{U}]_1, [\mathbf{B}^\top\mathbf{V}]_1)$.
  $(n, t, \mathsf{CS}, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathsf{pp})$.
  Assert $\mathsf{CS} \subset [1, n]$.
  Sample $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times(k+1)}$.
  $(\mathbf{K}_1, \ldots, \mathbf{K}_n) \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)$.
  $\mathsf{vk} := [\mathbf{KA}]_2$.
  for $i \in [1, n]$:
      $\mathsf{sk}_i := \mathbf{K}_i$, $\mathsf{vk}_i := [\mathbf{K}_i\mathbf{A}]_2$.
  $(\mathbf{m}^*, \varSigma^*, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{PSign}^*}(\cdot)}(\mathsf{st}_0, \mathsf{vk}, \{\mathsf{sk}_i\}_{i\in\mathsf{CS}}, \{\mathsf{vk}_i\}_{i\in[1,n]})$.
  Parse $\varSigma^*$ as $(\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4)$
  if $(\mathsf{Verify}^*(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \varSigma^*) \ \wedge \ |\mathsf{CS}| < t \ \wedge \ S_1([\mathbf{m}^*]_1) = \emptyset)$ :
      $\mathsf{result} := \mathsf{true}$
  else : $\mathsf{result} := \mathsf{false}$
  return $\tilde{b} \leftarrow \mathcal{A}(\mathsf{result})$

$\mathcal{O}^{\mathsf{PSign}^*}(i, [\mathbf{m}]_1)$:
  $\tau := \mathcal{H}([\mathbf{m}]_1)$.
  $(val_1, val_2) \leftarrow \mathcal{O}_b(\tau)$.
  $\sigma_1 := \left[\begin{pmatrix}1 & \mathbf{m}^\top\end{pmatrix}\mathbf{K}_i\right]_1 \cdot val_1$.
  $\sigma_2 := val_2$,
  $\sigma_3 := [\tau]_1 \cdot val_2$,
  $\sigma_4 := [\tau]_2$.
  $\varSigma_i := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.
  if $\varSigma_i \neq \perp$ :
      $S_1([\mathbf{m}]_1) := S_1([\mathbf{m}]_1) \cup \{i\}$.
  return $\varSigma_i$

$\mathsf{Verify}^*(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \varSigma^*)$ :
  Parse $\varSigma^*$ as $(\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4)$.
  return $\Big(e(\widehat{\sigma}_1, [1]_2) = e\Big(\left[\begin{pmatrix}1 & \mathbf{m}^{*\top}\end{pmatrix}\mathbf{K}\right]_1, [1]_2\Big) \cdot \ e(\widehat{\sigma}_2, \mathcal{O}^*(\widehat{\sigma}_4))$
         $\wedge \ e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)\Big)$
```

Fig. 7. Reduction to the core lemma in Lemma 1.

Lemma 2. $|\mathbf{Adv}_2 - \mathbf{Adv}_3| \leq 2q Adv^{\mathsf{MDDH}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}_1}(\kappa) + q/p$.

Proof. We prove this lemma through a reduction to the core lemma, Lemma 1. Let us assume there exists an adversary $\mathcal{A}$ that can distinguish the games Game_2 and Game_3, we can use it to build an adversary $\mathcal{B}_1$, defined in Figure 7, which breaks the core lemma, Lemma 1. The adversary $\mathcal{B}_1$ has access to four oracles, $\mathsf{Init}(\cdot), \mathcal{O}_b(\cdot), \mathcal{O}^*(\cdot), \mathcal{O}^{**}(\cdot)$, however in this reduction, we only use the first three oracles, defined as follows:

Oracle $\mathsf{Init}(\cdot)$: The oracle Init provides the set of public parameters pp.
Oracle $\mathcal{O}_b(\cdot)$: On the i-th query to this oracle on $[\tau]_1$, it outputs $\left([b\mu\mathbf{a}^\perp + \mathbf{r}_i^\top\mathbf{B}^\top(\mathbf{U} + \tau\cdot\mathbf{V})]_1, [\mathbf{r}_i^\top\mathbf{B}^\top]_1\right)$ depending on a random bit b.
Oracle $\mathcal{O}^*(\cdot)$: On input $[\tau^*]_2$, it returns $[\mathbf{U} + \tau^*\mathbf{V}]_2$.

When the lemma challenger selects the challenge bit as $b = 0$, it leads to the game Game_2, and when $b = 1$, it results in the game Game_3. All the other values are simulated perfectly. Thus, $|\mathbf{Adv}_2 - \mathbf{Adv}_3| \leq Adv^{\mathsf{Core}}_{\mathcal{D}_k,\mathsf{ABSGen},\mathcal{B}_1}(\kappa)$ holds and therefore we have,

$$|\mathbf{Adv}_2 - \mathbf{Adv}_3| \leq 2q Adv^{\mathsf{MDDH}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}}(\kappa) + q/p \cdot \qquad \square$$

Game 4. In this game, we apply the modifications described in Figure 8. Shamir secret sharing (see Definition 1) ensures that $(\mathbf{K}_1, \ldots, \mathbf{K}_n)$ in Game_3 and $(\widetilde{\mathbf{K}}_1, \ldots, \widetilde{\mathbf{K}}_n)$ in Game_4 have identical distributions. W.l.o.g, $\mathbf{K}_i$ in Game_3 and $\widetilde{\mathbf{K}}_i$ in Game_4 are identically distributed. In Game_4, on the other hand, $\widetilde{\mathbf{K}}_i$ and $\mathbf{K}_i = \widetilde{\mathbf{K}}_i - \mathbf{u}_i\mathbf{a}^\perp$ are identically distributed. Combining these observations, it follows that $\mathbf{K}_i$ in Game_3 and $\mathbf{K}_i$ in Game_4 are identically distributed for all $i \in [1, n]$. Consequently, it can be deduced that $\mathbf{K}$ in Game_3 and $\mathbf{K} + \mathbf{u}_0\mathbf{a}^\perp$ in Game_4 are identically distributed. Therefore, this change is just a conceptual change and we have,

$$|\mathbf{Adv}_3 - \mathbf{Adv}_4| = 0 \cdot$$

Now, we give a bound on $\mathbf{Adv}_4$ via an information-theoretic argument. We first consider the information about $\mathbf{u}_0$ (and subsequently $\{\mathbf{u}_i\}_{i\in[1,n]\setminus\mathsf{CS}}$) leaked from vk (and subsequently $\{\mathsf{vk}_i\}_{i\in[1,n]}$) and partial signing queries:

- $\mathsf{vk} := [\mathbf{K}\mathbf{A}]_2 = \left[\widetilde{\mathbf{K}}\mathbf{A}\right]_2$ and $\mathsf{vk}_i := [\mathbf{K}_i\mathbf{A}]_2 = \left[\widetilde{\mathbf{K}}_i\mathbf{A}\right]_2$ for all $i \in [1, n]$.
- The output of the j^{th} partial signature query on $(i, [\mathbf{m}]_1)$ for $[\mathbf{m}]_1 \neq [\mathbf{m}^*]_1$ completely hides $\{\mathbf{u}_i\}_{i\in[1,n]\setminus\mathsf{CS}}$ (and subsequently $\mathbf{u}_0$ as the adversary has only $|\mathsf{CS}|$ many $\mathbf{u}_i$ with $|\mathsf{CS}| < t$), since

$$\left(1\ \mathbf{m}^\top\right)\mathbf{K}_i + \mu_j\mathbf{a}^\perp = \left(1\ \mathbf{m}^\top\right)\widetilde{\mathbf{K}}_i + \left(1\ \mathbf{m}^\top\right)\mathbf{u}_i\mathbf{a}^\perp + \mu_j\mathbf{a}^\perp\ .$$

  distributed identically to $\left(1\ \mathbf{m}^\top\right)\widetilde{\mathbf{K}}_i + \mu_j\mathbf{a}^\perp$. This is because $\mu_j\mathbf{a}^\perp$ already hides $\left(1\ \mathbf{m}^\top\right)\mathbf{u}_i\mathbf{a}^\perp$ for uniformly random $\mu_j \leftarrow \mathbb{Z}_p$.

The only way to successfully convince the verification to accept a signature Σ^* on $\mathbf{m}^*$, the adversary must correctly compute $\left(1\ \mathbf{m}^{*\top}\right)(\mathbf{K} + \mathbf{u}_0\mathbf{a}^\perp)$ and thus $\left(1\ \mathbf{m}^{*\top}\right)\mathbf{u}_0$. Observe that, $\{\mathbf{u}_i\}_{i\in[1,n]\setminus\mathsf{CS}}$ (and thereby $\mathbf{u}_0$) are completely hidden to the adversary, $\left(1\ \mathbf{m}^{*\top}\right)\mathbf{u}_0$ is uniformly random from $\mathbb{Z}_p$ from the adversary's viewpoint. Therefore, $\mathbf{Adv}_4 = 1/p$.

□

Theorem 2. *Under the $\mathcal{D}_k$-MDDH Assumption in $\mathbb{G}_1$ and $\mathcal{D}_k$-KerMDH Assumption in $\mathbb{G}_2$, our Threshold Structure-Preserving Signature construction achieves $\mathsf{TS\text{-}UF\text{-}1}$ security against an efficient adversary making at most q partial signature queries.*

Proof Sketch. The difference between $\mathsf{TS\text{-}UF\text{-}0}$ and $\mathsf{TS\text{-}UF\text{-}1}$ lies in the fact that, in the latter model, an adversary can request $\mathcal{O}^{\mathsf{PSign}}(\cdot)$ queries on $[\mathbf{m}^*]_1$ for which it aims to forge a signature. The natural restriction in Figure 1 is expressed as $|S_1([\mathbf{m}^*]_1)| < t - |\mathsf{CS}|$, where t is the threshold value and the corrupted parties CS are fixed at the beginning of the game. As this security model allows partial signature oracle queries on $[\mathbf{m}^*]_1$, we next explore the changes we need to make on the proof of Theorem 1.

```
$G_3(\kappa)$: $\boxed{G_4(\kappa):}$
$\mathcal{G} \leftarrow \mathsf{ABSGen}(1^\kappa)$,
$\mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k$,
$\mathbf{U}, \mathbf{V} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$.
$\mathsf{pp} := ([\mathbf{A}]_2, [\mathbf{UA}]_2, [\mathbf{VA}]_2, [\mathbf{B}]_1, [\mathbf{B}^\top \mathbf{U}]_1, [\mathbf{B}^\top \mathbf{V}]_1)$.
$(n, t, \mathsf{CS}, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathsf{pp})$.
Assert $\mathsf{CS} \subset [1, n]$.
Sample $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times(k+1)}$.
$(\mathbf{K}_1, \ldots, \mathbf{K}_n) \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)$
$\boxed{\text{Sample } \mathbf{u}_0 \leftarrow \mathbf{Z}_p^{\ell+1}}$
$\boxed{(\mathbf{u}_1, \ldots, \mathbf{u}_n) \leftarrow \mathsf{Share}(\mathbf{u}_0, \mathbb{Z}_p^{(\ell+1)}, n, t)}$
$\boxed{(\widetilde{\mathbf{K}}_1, \ldots, \widetilde{\mathbf{K}}_n) \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)}$
$\boxed{\mathbf{K}_i := \widetilde{\mathbf{K}}_i + \mathbf{u}_i \mathbf{a}^\perp, \forall i \in [1, n]}$
$\mathsf{vk} := [\mathbf{KA}]_2$.
for $i \in [1, n]$:
    $\mathsf{sk}_i := \mathbf{K}_i$, $\mathsf{vk}_i := [\mathbf{K}_i \mathbf{A}]_2$.
$([\mathbf{m}^*]_1, \Sigma^*, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{PSign}(.)}}(\mathsf{st}_0, \mathsf{vk}, \{\mathsf{sk}_i\}_{i \in \mathsf{CS}}, \{\mathsf{vk}_i\}_{i \in [1,n]})$ .
return $\left(\mathsf{Verify}^*(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \Sigma^*) \wedge |\mathsf{CS}| < t \wedge S_1([\mathbf{m}^*]_1) = \emptyset\right)$

$\mathsf{Verify}^*(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \Sigma^*)$ :
Parse $\Sigma^*$ as $(\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4 = [\tau^*]_2)$.
return $\Big(e(\widehat{\sigma}_1, [1]_2) = e\left(\left[\left(1\ \mathbf{m}^{*\top}\right)(\mathbf{K} + \boxed{\mathbf{u}_0 \mathbf{a}^\perp})\right]_1, [1]_2\right) e\left(\widehat{\sigma}_2, [\mathbf{U} + \tau^* \mathbf{V}]_2\right)$
        $\wedge\ e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)\Big)$
```

Fig. 8. Modification from Game_3 to Game_4.

Game_0, Game_1 and Game_2 stay the same. To handle $\mathsf{TS\text{-}UF\text{-}1}$ adversaries, we introduce an additional game Game_2' to handle partial signature queries on the forged message. In Game_2', the challenger makes a list of all the partial signature queries and guesses the message on which forgery will be done. However, the guess will be made on the list of partial signature queries. More precisely, let $\mathcal{A}$ make partial signature queries on $[\mathbf{m}_1]_1, \ldots, [\mathbf{m}_{\mathcal{Q}}]_1$ s.t. $\mathcal{Q} \leq q$, the challenger of Game_2' rightly guesses the forged message with $1/\mathcal{Q}$ probability which introduces a degradation in the advantage. This small yet powerful modification allows the challenger in Game_3 to add a uniformly random quantity μ to partial signature oracle queries on $[\mathbf{m}]_1 \neq [\mathbf{m}^*]_1$. This concept is formulated by adding an additional line between lines number 2 and 3 in Figure 6. In particular, the new Game_3' (See Figure 9) would set $\mu = 0$ if $[\mathbf{m}]_1 = [\mathbf{m}^*]_1$. Next, we give an intuitive explanation of the indistinguishability of Game_2' and Game_3' which basically is a modification of the proof of Lemma 2.

```
$\mathcal{O}^{\mathsf{PSign}^*}(i, [\mathbf{m}]_1)$:
1: assert $\left([\mathbf{m}]_1 \in \mathcal{M} \ \wedge \ i \in \mathsf{HS}\right)$
2: $\mathbf{r}_i \leftarrow \mathbb{Z}_p^k, \tau := \mathcal{H}([\mathbf{m}]_1), \mu \leftarrow \mathbb{Z}_p$ $\boxed{\text{If } [\mathbf{m}]_1 = [\mathbf{m}^*]_1, \text{ set } \mu := 0}$
3: $\sigma_1 := \left[\left(1\ \mathbf{m}^\top\right)\mathbf{K}_i + \mu\mathbf{a}^\perp + \mathbf{r}_i^\top \mathbf{B}^\top(\mathbf{U} + \tau \cdot \mathbf{V})\right]_1,$
   $\sigma_2 := [\mathbf{r}_i^\top \mathbf{B}^\top]_1,$
   $\sigma_3 := [\tau \mathbf{r}_i^\top \mathbf{B}^\top]_1,$
   $\sigma_4 := [\tau]_2$
4: $\Sigma_i := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$
5: if $\Sigma_i \neq \perp$ :
6:     $S_1([\mathbf{m}]_1) := S_1([\mathbf{m}]_1) \cup \{i\}$
7: return $\Sigma_i$
```

Fig. 9. Game_3' in the proof of Theorem 2.

The novelty of this research lies in the need to simulate partial signature queries on the forged message $[\mathbf{m}^*]_1$, a challenge not addressed in previous works like [KW15, KPW15] upon which this study is based. It's important to mention that an extra oracle, termed $\mathcal{O}^{**}(\cdot)$, is sufficient for our objectives. On any partial signature query on the forged message $[\mathbf{m}^*]_1$, the reduction calls $\mathcal{O}^{**}([\tau^*]_1)$ for $\tau^* \leftarrow \mathcal{H}([\mathbf{m}^*]_1)$. Next we see that a single query to $\mathcal{O}^{**}([\tau^*]_1)$ is sufficient to handle multiple partial signature queries on $[\mathbf{m}^*]_1$. In particular, given a partial signature oracle query on $(i, [\mathbf{m}^*]_1)$, the reduction uses $\mathcal{O}^{**}(\cdot)$ of the so-called core-lemma (in Lemma 1) to get $\mathbf{X} = \left[\mathbf{B}^\top(\mathbf{U} + \tau^*\mathbf{V})\right]_1$, where $\tau^* = \mathcal{H}([\mathbf{m}^*]_1)$. The reduction then replies with $\left(\left[\left(1\ \mathbf{m}^{*\top}\right)\right]_1 \mathbf{K}_i + \mathbf{r}^\top \cdot \mathbf{X},\ \left[\mathbf{r}^\top\mathbf{B}^\top\right]_1,\ \left[\tau^*\mathbf{r}^\top\mathbf{B}^\top\right]_1,\ [\tau^*]_2\right)$ as a partial signature response to $\mathcal{A}$. Thus, a single call to $\mathcal{O}^{**}(\cdot)$ suffices to handle all partial signature queries on $[\mathbf{m}^*]_1$.

We define Game_4 as being identical to the proof of Theorem 1. In fact, the argument for the indistinguishability of Game_3 and Game_4 from the proof of Theorem 1 applies here as well. The argument that $\mathbf{Adv}_4$ is negligible however requires a small modification. Similar to the proof of Theorem 1, we can show that all verification keys vk and $\{\mathsf{vk}_i\}_{i \in [1,n]}$ stay the same. Furthermore, all partial signature queries on $[\mathbf{m}]_1 \neq [\mathbf{m}^*]_1$ do not leak any information about $\{\mathbf{u}_i\}_{i \in [1,n] \setminus \mathsf{CS}}$. Since, partial signature oracle queries are allowed on $[\mathbf{m}^*]_1$, observe that at most $\{\mathbf{u}_i\}_{i \in S_1([\mathbf{m}^*]_1)}$ are leaked to the adversary. To summarise, an adversary in $\mathsf{TS\text{-}UF\text{-}1}$ gets at most $\{\mathbf{u}_i\}_{i \in S_1([\mathbf{m}^*]_1) \sqcup \mathsf{CS}}$ even when it is unbounded. Due to the natural restriction, $|S_1([\mathbf{m}^*]_1)| + |\mathsf{CS}| < t$ ensures that $\mathbf{u}_0$ stays completely hidden to the adversary. Thus, $\left(1\ \mathbf{m}^{*\top}\right)\mathbf{u}_0$ is uniformly random from $\mathbb{Z}_p$ from the adversary's viewpoint. Therefore, $\mathbf{Adv}_4 \leq 1/p$. □

Theorem 3. *Under the $\mathcal{D}_k$-MDDH Assumption in $\mathbb{G}_1$ and $\mathcal{D}_k$-KerMDH Assumption in $\mathbb{G}_2$, the proposed Threshold Structure-Preserving Signature construction in Figure 3 achieves $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ security against an efficient adversary making at most q partial signature queries.*

Proof. The difference between $\mathsf{TS\text{-}UF\text{-}1}$ and $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ is that an adversary of the later model has access to $\mathcal{O}^{\mathsf{Corrupt}}(.)$ oracle and can corrupt the honest signers, adaptively. As per Figure 1, an $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ adversary proposes a corrupted set CS at the start of the game which it updates incrementally as the game progresses. At the time of forgery, the natural restriction in Figure 1 formulates as $|S_1([\mathbf{m}^*]_1)| < t - |\mathsf{CS}|$, where t is the threshold value and CS contains the list of corrupted signers at the forgery phase. Given that this security model permits an adversary to obtain the secret keys of users it may have queried using the $\mathcal{O}^{\mathsf{PSign}}(.)$ oracle in the past, our next step involves investigating the main modifications required for the proof in Theorem 2.

Game_0, Game_1, Game_2, and Game_2' stay the same. In the proof of Theorem 2, we also have showed that Game_2' and Game_3' to be indistinguishable due to the so-called core lemma, Lemma 1. We reuse the reduction in Figure 7 towards this purpose. The reduction in Figure 7 samples $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times(k+1)}$ and generates $(\mathbf{K}_1, \dots, \mathbf{K}_n) \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)$. Recall that, the $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ adversary $\mathcal{A}$ of Lemma 2 corrupts a party $i \in [1, n]$ adaptively. Since the reduction of Lemma 2 already knows $\mathbf{K}_i$ in plain, it can handle the $\mathcal{O}^{\mathsf{Corrupt}}(.)$ oracle queries quite naturally.

The indistinguishability of Game_3 and Game_4 are argued exactly the same as in Theorem 2. We now focus on $\mathbf{Adv}_4$. In Game_4, the adversary gets to update CS adaptively. Intuitively, all $\mathbf{K}_i$ are independently sampled. Giving out a few of them to the adversary does not change the adversary's view. In the proof of Theorem 2, we already have managed to address partial signature queries on forged message. Except a few details, this ensures our proof will work out. We next give a formal argument.

We prove this theorem through a series of games and we use $\mathbf{Adv}_i$ to denote the advantage of the adversary $\mathcal{A}$ in winning the Game i. The games are described below.

Game 0. This is the $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ security game described in Definition 8. As shown in Figure 10, an adversary $\mathcal{A}$ after receiving the set of public parameters, pp, returns $(\mathrm{n}, t, \mathsf{CS})$, where n, t and CS represents the total number of signers, the threshold, and the set of corrupted signers, respectively. The adversary can query the partial signing oracle $\mathcal{O}^{\mathsf{PSign}}(\cdot)$ to receive partial signatures. Let $\mathcal{Q}$ represent the number of distinct messages where partial signing queries are made. In the end, the adversary outputs a message $[\mathbf{m}^*]_1$ and a forged signature Σ^*.

Game 1. We modify the verification procedure to the one described in Figure 11. Consider any forged message/signature pair $([\mathbf{m}^*]_1, \Sigma^* = (\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4))$ where $e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)$, $|\mathsf{CS}| < t$ and $S_1([\mathbf{m}^*]_1) = \emptyset$. Note that if the pair $([\mathbf{m}^*]_1, \Sigma^*)$ meets the $\mathsf{Verify}^*(\cdot)$ conditions, outlined in Figure 11, it also satisfies $\mathsf{Verify}(\cdot)$ procedure, described in Figure 10. This is primarily due to the fact that:

$$\begin{aligned} & e(\widehat{\sigma}_1, [\mathbf{A}]_2) = e\left([\left(1\ \mathbf{m}^{*\top}\right)]_1, [\mathbf{KA}]_2\right) \cdot e(\widehat{\sigma}_2, [\mathbf{UA}]_2) \cdot e(\widehat{\sigma}_3, [\mathbf{VA}]_2) \\ \Longleftarrow\ & e(\widehat{\sigma}_1, [1]_2) = e([\left(1\ \mathbf{m}^{*\top}\right)]_1, [\mathbf{K}]_2) \cdot e(\widehat{\sigma}_2, [\mathbf{U}]_2) \cdot e(\widehat{\sigma}_3, [\mathbf{V}]_2) \\ \Longleftrightarrow\ & e(\widehat{\sigma}_1, [1]_2) = e([\left(1\ \mathbf{m}^{*\top}\right)\mathbf{K}]_1, [1]_2) \cdot e(\widehat{\sigma}_2, [\mathbf{U} + \tau^*\mathbf{V}]_2) \cdot \end{aligned}$$

Assume there exists a message/signature pair $([\mathbf{m}^*]_1, \Sigma^* = (\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4))$ that satisfies $\mathsf{Verify}(.)$ and not $\mathsf{Verify}^*(.)$, then we can compute a non-zero vector $\mathbf{c}$ in the kernel of $\mathbf{A}$ as follows:

$$\mathbf{c} := \widehat{\sigma}_1 - ([\left(1\ \mathbf{m}^{*\top}\right)\mathbf{K}]_1 + \widehat{\sigma}_2\mathbf{U} + \widehat{\sigma}_3\mathbf{V}) \in \mathbb{G}_1^{1\times(k+1)} \cdot$$

According to $\mathcal{D}_k$-KerMDH assumption over $\mathbb{G}_2$ described in Definition 5, such a vector $\mathbf{c}$ is hard to compute. Thus,

$$|\mathbf{Adv}_0 - \mathbf{Adv}_1| \leq Adv^{\mathsf{KerMDH}}_{\mathcal{D}_k, \mathbb{G}_2, \mathcal{B}_0}(\kappa) \cdot$$

Game 2. On receiving a partial signature query on a message $[\mathbf{m}_i]_1$, a list is updated with the message $[\mathbf{m}_i]_1$ and the corresponding tag $\tau_i := \mathcal{H}([\mathbf{m}_i]_1)$. The challenger aborts if an adversary can generate two tuples $([\mathbf{m}_i]_1, \tau_i)$, $([\mathbf{m}_j]_1, \tau_j)$ with $[\mathbf{m}_i]_1 \neq [\mathbf{m}_j]_1$ and $\tau_i = \tau_j$. By the collision resistance property of the underlying hash function we have:

$$|\mathbf{Adv}_1 - \mathbf{Adv}_2| \leq Adv^{\mathsf{CRHF}}_{\mathcal{H}}(\kappa) \cdot$$

```
Verify*(pp, vk, $[\mathbf{m}^*]_1, \Sigma^*$):
  Parse $\Sigma^*$ as $(\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4 = [\tau^*]_2)$.
  return $\Big(e(\widehat{\sigma}_1, [1]_2) = e\left([\left(1\ \mathbf{m}^{*\top}\right)\mathbf{K}]_1, [1]_2\right) \cdot e(\widehat{\sigma}_2, [\mathbf{U} + \tau^*\mathbf{V}]_2) \wedge$
         $e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)\Big)$
```

Fig. 11. Modifications in Game_1.

```
$\boldsymbol{G}_0(\kappa)$:
  $\mathcal{G} \leftarrow \mathsf{ABSGen}(1^\kappa)$,
  $\mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k$,
  $\mathbf{U}, \mathbf{V} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$.
  $\mathsf{pp} := ([\mathbf{A}]_2, [\mathbf{UA}]_2, [\mathbf{VA}]_2, [\mathbf{B}]_1, [\mathbf{B}^\top\mathbf{U}]_1, [\mathbf{B}^\top\mathbf{V}]_1)$.
  $(n, t, \mathsf{CS}, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathsf{pp})$.
  Assert $\mathsf{CS} \subset [1, n]$.
  Sample $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times(k+1)}$.
  $(\mathbf{K}_1, \dots, \mathbf{K}_n) \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)$.
  $\mathsf{vk} := [\mathbf{KA}]_2$.
  $\mathsf{sk}_i := \mathbf{K}_i$, $\mathsf{vk}_i := [\mathbf{K}_i\mathbf{A}]_2$ for $i \in [1, n]$.
  $([\mathbf{m}^*]_1, \Sigma^*, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{PSign}(\cdot)}, \mathcal{O}^{\mathsf{Corrupt}}(\cdot)}(\mathsf{st}_0, \mathsf{vk}, \{\mathsf{sk}_i\}_{i\in\mathsf{CS}}, \{\mathsf{vk}_i\}_{i\in[1,n]})$.
  return $\Big(\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \Sigma^*) \ \wedge\ |\mathsf{CS}| < t \ \wedge\ S_1([\mathbf{m}^*]_1) = \emptyset\Big)$

$\mathcal{O}^{\mathsf{PSign}}(i, [\mathbf{m}]_1)$:
  Assert $\big([\mathbf{m}]_1 \in \mathcal{M} \ \wedge\ i \in \mathsf{HS}\big)$.
  $\mathbf{r}_i \leftarrow \mathbb{Z}_p^k$.
  $\tau := \mathcal{H}([\mathbf{m}]_1)$.
  $\sigma_1 := \left[\begin{pmatrix}1 & \mathbf{m}^\top\end{pmatrix}\mathbf{K}_i + \mathbf{r}_i^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})\right]_1$.
  $\sigma_2 := [\mathbf{r}_i^\top\mathbf{B}^\top]_1$,
  $\sigma_3 := [\tau\mathbf{r}_i^\top\mathbf{B}^\top]_1$,
  $\sigma_4 := [\tau]_2$.
  $\Sigma_i := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.
  if $\Sigma_i \neq \perp$ :
      $S_1([\mathbf{m}]_1) := S_1([\mathbf{m}]_1) \cup \{i\}$.
  return $\Sigma_i$

$\mathcal{O}^{\mathsf{Corrupt}}(j)$:
  $\mathsf{CS} \leftarrow \mathsf{CS} \cup \{j\}$
  $\mathsf{HS} \leftarrow \mathsf{CS} \setminus \{j\}$
  return $\mathsf{sk}_j$

$\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, [\mathbf{m}^*]_1, \Sigma^*)$ :
  Parse $\Sigma^*$ as $(\widehat{\sigma}_1, \widehat{\sigma}_2, \widehat{\sigma}_3, \widehat{\sigma}_4)$.
  return $\Big(e(\widehat{\sigma}_1, [\mathbf{A}]_2) = e\left(\left[\begin{pmatrix}1 & \mathbf{m}^{*\top}\end{pmatrix}\right]_1, [\mathbf{KA}]_2\right) \cdot e(\widehat{\sigma}_2, [\mathbf{UA}]_2) \cdot e(\widehat{\sigma}_3, [\mathbf{VA}]_2)$
              $\wedge\ e(\widehat{\sigma}_2, \widehat{\sigma}_4) = e(\widehat{\sigma}_3, [1]_2)\Big)$
```

Fig. 10. Game$_0$.

Game 2′. In Game_2', the challenger randomly chooses an index $j^* \leftarrow [1, Q]$ as its guess of the message on which the forgery will be done. This game is the same as Game 2 except that the challenger aborts the game immediately if forged message $[\mathbf{m}^*]_1 \neq [\mathbf{m}_{j^*}]_1$.
The challenger of Game_2' rightly guesses the forged message $[\mathbf{m}^*]_1$ with $1/\mathcal{Q}$ probability which introduces a degradation in the advantage of Game_2': $\mathbf{Adv}_{2'} = \frac{1}{\mathcal{Q}}\mathbf{Adv}_2$.

Game 3′. This game is same as Game_2' except we introduce randomness to the partial signatures by adding $\mu\mathbf{a}^\perp$ to each partial signature query on all messages $[\mathbf{m}]_1$ except $[\mathbf{m}]_1^*$ on which the forgery is done.

```
$\mathcal{O}^{\mathsf{PSign}^*}(i, [\mathbf{m}]_1)$:
assert $\left([\mathbf{m}]_1 \in \mathcal{M} \ \wedge \ i \in \mathsf{HS}\right)$
$\mathbf{r}_i \leftarrow \mathbb{Z}_p^k, \tau := \mathcal{H}([\mathbf{m}]_1), \mu \leftarrow \mathbb{Z}_p$ $\boxed{\text{If } [\mathbf{m}]_1 = [\mathbf{m}^*]_1, \text{ set } \mu := 0}$
$\sigma_1 := \left[\left(1\ \mathbf{m}^\top\right)\mathbf{K}_i + \mu\mathbf{a}^\perp + \mathbf{r}_i^\top\mathbf{B}^\top(\mathbf{U} + \tau\cdot\mathbf{V})\right]_1,$
$\sigma_2 := [\mathbf{r}_i^\top\mathbf{B}^\top]_1,$
$\sigma_3 := [\tau\mathbf{r}_i^\top\mathbf{B}^\top]_1,$
$\sigma_4 := [\tau]_2$
$\Sigma_i := (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$
if $\Sigma_i \neq \perp$ :
    $S_1([\mathbf{m}]_1) := S_1([\mathbf{m}]_1) \cup \{i\}$
return $\Sigma_i$
```

Fig. 12. Game_3' in the proof of Theorem 3.

We show that, we can make a reduction algorithm $\mathcal{B}$ for the so-called core-lemma (in Lemma 1) using $\mathcal{A}$. At the start of the game, $\mathcal{B}$ randomly chooses an index $j^* \leftarrow [1, Q]$ as its guess of the message on which forgery will be done. If $[\mathbf{m}^*]_1 \neq [\mathbf{m}_{j^*}]_1 = [\mathbf{m}^*]_1$, $\mathcal{B}$ aborts. Otherwise, B outputs A's output as it is. In particular, $\mathcal{B}$ does the following:

1. $\mathcal{B}$ receives pp from the challenger.
2. $\mathcal{B}$ samples $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times(k+1)}$.
3. $\mathcal{B}$ then secret shares $\mathbf{K}$ into $(\mathbf{K}_1, \ldots, \mathbf{K}_n) \leftarrow \mathsf{Share}(\mathbf{K}, \mathbb{Z}_p^{(\ell+1)\times(k+1)}, n, t)$.
4. On a $\mathcal{O}^{\mathsf{Corrupt}}(.)$ oracle query on $j \in [1, n]$, $\mathcal{B}$ returns $\mathbf{K}_j$.
5. $\mathcal{B}$ simulates the partial signature query on $(i, [\mathbf{m}]_1)$ as following:
   - If $[\mathbf{m}]_1 = [\mathbf{m}^*]_1$, it makes a query (i, τ^*) on $\mathcal{O}^{**}(.)$ where $\tau^* \leftarrow \mathcal{H}([\mathbf{m}^*]_1)$.
     - Let $\mathcal{B}$ receives val as the response of the above queries.
     - $\mathcal{B}$ samples $\mathbf{r}_i \leftarrow \mathbb{Z}_p^k$ and returns $\Sigma_i := (\left[\left(1\ \mathbf{m}^\top\right)\mathbf{K}_i\right]_1 \cdot \mathbf{r}_i^\top \cdot val, \mathbf{r}_i^\top \cdot val, \tau \cdot \mathbf{r}_i^\top \cdot val, [\tau]_2)$ to $\mathcal{A}$ as the partial signature.
   - If $[\mathbf{m}]_1 \neq [\mathbf{m}^*]_1$, it makes a query (i, τ) on $\mathcal{O}^b(\cdot)$, where $\tau \leftarrow \mathcal{H}([\mathbf{m}]_1)$.
     - Let $\mathcal{B}$ receives (val_1, val_2) as the response of the above queries.
     - It returns $\Sigma_i := \left(\left[\left(1\ \mathbf{m}^\top\right)\mathbf{K}_i\right]_1 \cdot val_1, val_2, \tau\cdot val_2, [\tau]_2\right)$ to $\mathcal{A}$ as the partial signature.
6. On $\mathsf{Verify}^*(.)$ on $(\mathsf{vk}, [\mathbf{m}^*]_1, \Sigma^*)$, $\mathcal{B}$ queries on $\mathcal{O}^*(\cdot)$ on $[\tau^*]_2$ where $\tau^* \leftarrow \mathcal{H}([\mathbf{m}^*]_1)$.
   - Let Σ^* is $(\sigma_1, \sigma_2, \sigma_3, \sigma_4 = [\tau^*]_2)$.
   - Let $\mathcal{B}$ receives val as the response of the above query.
   - $\mathcal{B}$ verifies the signature: $e(\sigma_1, [1]_2) = e\left(\left[\left(1\ \mathbf{m}^{*\top}\right)\mathbf{K}\right]_1, [1]_2\right) \cdot e(\sigma_2, val) \wedge e(\sigma_2, \sigma_4) = e(\sigma_3, [1]_2)$.

Game_2' and Game_3' are indistinguishable due to the so-called core-lemma (in Lemma 1), then we have:

$$|\mathbf{Adv}_{2'} - \mathbf{Adv}_{3'}| \leq 2\mathcal{Q}Adv^{\mathsf{MDDH}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}_1}(\kappa) + \mathcal{Q}/p\ \cdot$$

Game 4. This game is same as Game_3' except that $\{\mathbf{K}_i\}_{i\in[n]}$ are sampled. In particular, we sample $\mathbf{K}_i = \widetilde{\mathbf{K}}_i + \mathbf{u}_i\mathbf{a}^\perp$ for $i \in [1, n]$.
Shamir secret sharing (see Definition 1) ensures that $(\mathbf{K}_1, \ldots, \mathbf{K}_n)$ in Game_3 and $(\widetilde{\mathbf{K}}_1, \ldots, \widetilde{\mathbf{K}}_n)$ in Game_4 are identically distributed. W.l.o.g, $\mathbf{K}_i$ in Game_3' and $\widetilde{\mathbf{K}}_i$ in Game_4 are identically distributed. In Game_4, on the other hand, $\widetilde{\mathbf{K}}_i$ and $\mathbf{K}_i = \widetilde{\mathbf{K}}_i - \mathbf{u}_i\mathbf{a}^\perp$ are identically distributed. Considering both

together, $\mathbf{K}_i$ is Game_3' and $\mathbf{K}_i$ in Game_4 are identically distributed for all $i \in [1, n]$. Thus further ensures that $\mathbf{K}$ in Game_3' and $\mathbf{K} + \mathbf{u}_0\mathbf{a}^\perp$ in Game_4 are identically distributed. Therefore, this change is just a conceptual change and $\mathbf{Adv}_{3'} - \mathbf{Adv}_4 = 0$.

Finally, we argue that $\mathbf{Adv}_4 = 1/p$. Notice that, the adversary gets to update CS adaptively. To complete the argument, we have to ensure that even after getting $\mathbf{K}_i = \widetilde{\mathbf{K}}_i + \mathbf{u}_i\mathbf{a}^\perp$ for $i \in [\mathsf{CS}]$ chosen adaptively and even after having several partial signatures (possibly on the corrupted keys too), $\mathbf{u}_0$ is still hidden to the adversary.

– Firstly, vk and $\{\mathsf{vk}_i\}_{i\in[1,n]}$ do not leak anything about $\mathbf{u}_0$ and $\{\mathbf{u}_i\}_{i\in[1,n]}$ respectively. Note that, $\mathcal{A}$ gets $\mathsf{sk}_i = \mathbf{K}_i = \widetilde{\mathbf{K}}_i + \mathbf{u}_i\mathbf{a}^\perp$ for $i \in [\mathsf{CS}]$ as a part of Input.

– The output of j^{th} partial signature query on $(i, [\mathbf{m}]_1)$ for $[\mathbf{m}]_1 \neq [\mathbf{m}^*]_1$ completely hides $\{\mathbf{u}_i\}_{i\in[1,n]\setminus\mathsf{CS}}$ (and subsequently $\mathbf{u}_0$ as the adversary has only $|\mathsf{CS}|$ many $\mathbf{u}_i$ where $|\mathsf{CS}| < t$), since

$$\left(1\ \mathbf{m}^\top\right)\mathbf{K}_i + \mu_j\mathbf{a}^\perp = \left(1\ \mathbf{m}^\top\right)\widetilde{\mathbf{K}}_i + \left(1\ \mathbf{m}^\top\right)\mathbf{u}_i\mathbf{a}^\perp + \mu_j\mathbf{a}^\perp \cdot$$

distributed identically to $\left(1\ \mathbf{m}^\top\right)\widetilde{\mathbf{K}}_i + \mu_j\mathbf{a}^\perp$. This is because $\mu_j\mathbf{a}^\perp$ already hides $\left(1\ \mathbf{m}^\top\right)\mathbf{u}_i\mathbf{a}^\perp$ for uniformly random $\mu_j \leftarrow \mathbb{Z}_p$.

– In case of the j^{th} partial signature query on $(i, [\mathbf{m}^*]_1)$, observe that at most $\{\mathbf{u}_i\}_{i\in S_1([\mathbf{m}^*]_1)}$ are leaked to the adversary. To summarise, an $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ adversary gets at most $\{\mathbf{u}_i\}_{i\in S_1([\mathbf{m}^*]_1)}$ even when it is unbounded.

– Finally, we take a look at the corrupted set CS. We emphasize that this set was updated through out the game adaptively.

From the above discussion, it is clear that the information theoretically adversary can at most gets hold of $\{\mathbf{u}_i\}_{i\in S_1([\mathbf{m}^*]_1)\sqcup\mathsf{CS}}$ adaptively. Note that, the only way to sucessfuly convince the verification to accept a signature Σ^* on $\mathbf{m}^*$, the adversary must correctly compute $\left(1\ \mathbf{m}^{*\top}\right)(\mathbf{K} + \mathbf{u}_0\mathbf{a}^\perp)$ and thus $\left(1\ \mathbf{m}^{*\top}\right)\mathbf{u}_0$. So the question now reduces to if the adversary can compute $\mathbf{u}_0$ from $\{\mathbf{u}_i\}_{i\in S_1([\mathbf{m}^*]_1)\sqcup\mathsf{CS}}$ which it got adaptively. Since Shamir secret sharing is information theoretically secure, the advantage of an adversary in case of selective corruption of users is same as the advantage of an adversary in case of adaptive corruption of users. Thus, $\mathbf{u}_0$ is completely hidden to the adaptive adversary, $\left(1\ \mathbf{m}^{*\top}\right)\mathbf{u}_0$ is uniformly random from $\mathbb{Z}_p$ from its viewpoint. Therefore, $\mathbf{Adv}_4 = 1/p$.

□

3.5 Proof of Core Lemma

Proof of Lemma 1. We proceed through a series of games from Game_0 to Game_q. Note that, Init outputs the same in all the games. In Game_i, the first i queries to the oracle $\mathcal{O}_b(.)$ are responded with $([\mu\mathbf{a}^\perp + \mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{r}^\top\mathbf{B}^\top]_1)$ and the next $q - i$ queries are responded with $([\mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{r}^\top\mathbf{B}^\top]_1)$. The intermediate games Game_i and Game_{i+1} respond differently to the $i + 1$-th query to $\mathcal{O}_b(.)$. The Game_i responds with $([\mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{r}^\top\mathbf{B}^\top]_1)$ whereas Game_{i+1} responds with $([\mu\mathbf{a}^\perp + \mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{r}^\top\mathbf{B}^\top]_1)$. We compute the advantage of the adversary in differentiating the two games below. The advantage of the adversary in Game_i is denoted by $\mathbf{Adv}_i$ for $i = 0, \ldots, q$. On querying $\mathcal{O}_b(\cdot)$, Game_i responds to $i + 1$-th query with

$$([\mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{r}^\top\mathbf{B}^\top]_1)\ ,$$

where $\mathbf{r} \leftarrow \mathbb{Z}_p^k$.

We define a sub-game $\mathsf{Game}_{i.1}$ where $[\mathbf{Br}]_1$ is replaced with $[\mathbf{w}]_1$, $[\mathbf{w}]_1 \leftarrow \mathbb{G}_1^{k+1}$. From the MDDH assumption, a MDDH adversary cannot distinguish between the distributions $([\mathbf{B}]_1, [\mathbf{Br}]_1)$ and $([\mathbf{B}]_1, [\mathbf{w}]_1)$. Thus,

$$([\mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{r}^\top\mathbf{B}^\top]_1) \approx_c ([\mathbf{w}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{w}]_1)\ \cdot$$

All the other values can be perfectly simulated in the reduction by choosing $\mathbf{U}$ and $\mathbf{V}$ from the appropriate distributions. In the next sub-game $\mathsf{Game}_{i.2}$, we introduce the randomness $\mu\mathbf{a}^\perp$ to $[\mathbf{w}^\top(\mathbf{U} + \tau\mathbf{V})]_1$ and proceed to use an information-theoretic argument to bound the advantage in this experiment. As shown in [KW15], for every $\mathbf{A}, \mathbf{B} \leftarrow \mathcal{D}_k$, $\tau \neq \tau^*$, the following distributions are identically distributed

$$(\mathsf{vk}, [\mathbf{w}^\top(\mathbf{U} + \tau\mathbf{V})]_1, \mathbf{U} + \tau^*\mathbf{V}) \text{ and } (\mathsf{vk}, [\mu\mathbf{a}^\perp + \mathbf{w}^\top(\mathbf{U} + \tau\mathbf{V})]_1, \mathbf{U} + \tau^*\mathbf{V})\ \cdot$$

with probability $1 - 1/p$ over $\mathbf{w}$. The values $[\mathbf{B}^\top \mathbf{U}]_1$ and $[\mathbf{B}^\top \mathbf{V}]_1$ are part of the public values $\mathsf{vk} := (\mathbf{A}, \mathbf{UA}, \mathbf{VA}, [\mathbf{B}]_1, [\mathbf{B}^\top \mathbf{U}]_1, [\mathbf{B}^\top \mathbf{V}]_1)$ and anyone can compute $[\mathbf{B}^\top(\mathbf{U} + \tau^*\mathbf{V})]_1$ corresponding to a τ^*. Thus, for $\tau \neq \tau^*$, we have the two following identical distributions:

$$\begin{aligned}&(\mathsf{vk}, [\mathbf{w}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{U} + \tau^*\mathbf{V}]_2, [\mathbf{B}^\top(\mathbf{U} + \tau^*\mathbf{V})]_1) \text{ and} \\ &(\mathsf{vk}, [\mu\mathbf{a}^\perp + \mathbf{w}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{U} + \tau^*\mathbf{V}]_2, [\mathbf{B}^\top(\mathbf{U} + \tau^*\mathbf{V})]_1) \cdot\end{aligned} \tag{1}$$

From Equation (1), the subgames $\mathsf{Game}_{i.1}$ and $\mathsf{Game}_{i.2}$ are statistically close. We use the MDDH assumption again in the next sub-game $\mathsf{Game}_{i.3}$ and replace $[\mathbf{w}]_1$ with $[\mathbf{Br}]_1$. The resulting distribution is

$$(\mathsf{vk}, [\mu\mathbf{a}^\perp + \mathbf{r}^\top\mathbf{B}^\top(\mathbf{U} + \tau\mathbf{V})]_1, [\mathbf{U} + \tau^*\mathbf{V}]_2, [\mathbf{B}^\top(\mathbf{U} + \tau^*\mathbf{V})]_1) ,$$

which is same as Game_{i+1}. Thus, from the two MDDH instances as well as the information-theoretic argument,

$$|\mathbf{Adv}_i - \mathbf{Adv}_{i+1}| \leq 2Adv^{\mathsf{MDDH}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}}(\kappa) + 1/p \cdot$$

□

4 Conclusion

In this paper, we give the first construction of a non-interactive threshold structure-preserving signature (TSPS) scheme from standard assumptions. We prove our construction secure in the $\mathsf{adp\text{-}TS\text{-}UF\text{-}1}$ security model where the adversary is allowed to obtain partial signatures on the forged message and additionally allow the adversary to adaptively corrupt parties. Although the signatures are constant-size (and in fact quite small), we consider improving the efficiency of TSPS under standard assumptions as an interesting future work.

Acknowledgements. Mahdi Sedaghat was supported in part by the Research Council KU Leuven C1 on Security and Privacy for Cyber-Physical Systems and the Internet of Things with contract number C16/15/058 and by CyberSecurity Research Flanders with reference number VR20192203. Daniel Slamanig was supported by the Austrian Science Fund (FWF) and netidee SCIENCE PROFET (GA No. P31621-N38). His work was done while being affiliated with AIT Austrian Institute of Technology.

References

AAOT18. Masayuki Abe, Miguel Ambrona, Miyako Ohkubo, and Mehdi Tibouchi. Lower bounds on structure-preserving signatures for bilateral messages. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 3–22. Springer, Heidelberg, September 2018.

ACD$^+$12. Masayuki Abe, Melissa Chase, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 4–24. Springer, Heidelberg, December 2012.

ACHO11. Masayuki Abe, Sherman S. M. Chow, Kristiyan Haralambiev, and Miyako Ohkubo. Double-trapdoor anonymous tags for traceable signatures. In Javier Lopez and Gene Tsudik, editors, *ACNS 11*, volume 6715 of *LNCS*, pages 183–200. Springer, Heidelberg, June 2011.

ADN06. Jesus F. Almansa, Ivan Damgard, and Jesper Buus Nielsen. Simplified threshold RSA with adaptive and proactive security. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 593–611. Springer, 2006.

AFG$^+$10. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, Heidelberg, August 2010.

AGHO11. Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 649–666. Springer, Heidelberg, August 2011.

AGO11. Masayuki Abe, Jens Groth, and Miyako Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 628–646. Springer, Heidelberg, December 2011.

AGOT14. Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 688–712. Springer, Heidelberg, February 2014.

AHN$^+$17. Masayuki Abe, Dennis Hofheinz, Ryo Nishimaki, Miyako Ohkubo, and Jiaxin Pan. Compact structure-preserving signatures with almost tight security. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 548–580. Springer, Heidelberg, August 2017.

AJO$^+$19. Masayuki Abe, Charanjit S. Jutla, Miyako Ohkubo, Jiaxin Pan, Arnab Roy, and Yuyu Wang. Shorter QA-NIZK and SPS with tighter security. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 669–699. Springer, Heidelberg, December 2019.

AJOR18. Masayuki Abe, Charanjit S. Jutla, Miyako Ohkubo, and Arnab Roy. Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 627–656. Springer, Heidelberg, December 2018.

ALP12. Nuttapong Attrapadung, Benoît Libert, and Thomas Peters. Computing on authenticated data: New privacy definitions and constructions. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 367–385. Springer, Heidelberg, December 2012.

ANO$^+$22. Damiano Abram, Ariel Nof, Claudio Orlandi, Peter Scholl, and Omer Shlomovits. Low-bandwidth threshold ECDSA via pseudorandom correlation generators. In *2022 IEEE Symposium on Security and Privacy*, pages 2554–2572. IEEE Computer Society Press, May 2022.

BCF$^+$11. Olivier Blazy, Sébastien Canard, Georg Fuchsbauer, Aline Gouget, Hervé Sibert, and Jacques Traoré. Achieving optimal anonymity in transferable e-cash with a judge. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 206–223. Springer, Heidelberg, July 2011.

BCK$^+$22. Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Better than advertised security for non-interactive threshold signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 517–550. Springer, Heidelberg, August 2022.

BL22. Renas Bacho and Julian Loss. On the adaptive security of the threshold BLS signature scheme. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 193–207. ACM Press, November 2022.

BLS01. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Heidelberg, December 2001.

BLT$^+$23. Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold signatures from ddh with full adaptive security. Cryptology ePrint Archive, Paper 2023/1482 (To appear at EUROCRYPT 2024), 2023. https://eprint.iacr.org/2023/1482.

BMW21. Christian Badertscher, Christian Matt, and Hendrik Waldner. Policy-compliant signatures. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part III*, volume 13044 of *Lecture Notes in Computer Science*, pages 350–381. Springer, 2021.

BNPS03. Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003.

Bol03. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Heidelberg, January 2003.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

BS23. Alexandre Bouez and Kalpana Singh. One round threshold ECDSA without roll call. In Mike Rosulek, editor, *Topics in Cryptology - CT-RSA 2023 - Cryptographers' Track at the RSA Conference 2023, San Francisco, CA, USA, April 24-27, 2023, Proceedings*, volume 13871 of *Lecture Notes in Computer Science*, pages 389–414. Springer, 2023.

BSW23. Christian Badertscher, Mahdi Sedaghat, and Hendrik Waldner. Fine-grained accountable privacy via unlinkable policy-compliant signatures. *Cryptology ePrint Archive, Paper 2023/1070*, 2023. https://eprint.iacr.org/2023/1070.

CDH12. Jan Camenisch, Maria Dubovitskaya, and Kristiyan Haralambiev. Efficient structure-preserving signature scheme from standard assumptions. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 76–94. Springer, Heidelberg, September 2012.

CDHK15. Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 262–288. Springer, Heidelberg, November / December 2015.

CFGN96. Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996.

CGG$^+$20. Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. UC non-interactive, proactive, threshold ECDSA with identifiable aborts. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1769–1787. ACM Press, November 2020.

CGJ$^+$99. Ran Canetti, Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Adaptive security for threshold cryptosystems. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 98–115. Springer, Heidelberg, August 1999.

CH20. Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, August 2020.

CKLM12. Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable proof systems and applications. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 281–300. Springer, Heidelberg, April 2012.

CKM23. Elizabeth Crites, Chelsea Komlo, and Mary Maller. Fully adaptive schnorr threshold signatures. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 678–709, Cham, 2023. Springer Nature Switzerland.

CKP$^+$23. Elizabeth Crites, Markulf Kohlweiss, Bart Preneel, Mahdi Sedaghat, and Daniel Slamanig. Threshold Structure-Preserving Signatures. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023*, pages 348–382, Singapore, 2023. Springer Nature Singapore.

DF90. Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 307–315. Springer, Heidelberg, August 1990.

DMZ$^+$21. Yi Deng, Shunli Ma, Xinxuan Zhang, Hailong Wang, Xuyang Song, and Xiang Xie. Promise Σ-protocol: How to construct efficient threshold ECDSA from encryptions based on class groups. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 557–586. Springer, Heidelberg, December 2021.

DOK$^+$20. Anders P. K. Dalskov, Claudio Orlandi, Marcel Keller, Kris Shrishak, and Haya Shulman. Securing DNSSEC keys via threshold ECDSA from generic MPC. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *ESORICS 2020, Part II*, volume 12309 of *LNCS*, pages 654–673. Springer, Heidelberg, September 2020.

DR23. Sourav Das and Ling Ren. Adaptively Secure BLS Threshold Signatures from DDH and co-CDH. Cryptology ePrint Archive, Paper 2023/1553, 2023. https://eprint.iacr.org/2023/1553.

EHK$^+$17. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Luis Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, January 2017.

FHS15. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 233–253. Springer, Heidelberg, August 2015.

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

Fuc11. Georg Fuchsbauer. Commuting signatures and verifiable encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 224–245. Springer, Heidelberg, May 2011.

GG18. Rosario Gennaro and Steven Goldfeder. Fast multiparty threshold ECDSA with fast trustless setup. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 1179–1194. ACM Press, October 2018.

Gha16. Essam Ghadafi. Short structure-preserving signatures. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 305–321. Springer, Heidelberg, February / March 2016.

Gha17. Essam Ghadafi. More efficient structure-preserving signatures - or: Bypassing the type-III lower bounds. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *ESORICS 2017, Part II*, volume 10493 of *LNCS*, pages 43–61. Springer, Heidelberg, September 2017.

GHKP18. Romain Gay, Dennis Hofheinz, Lisa Kohl, and Jiaxin Pan. More efficient (almost) tightly secure structure-preserving signatures. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 230–258. Springer, Heidelberg, April / May 2018.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

HJ12. Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Heidelberg, August 2012.

JL00. Stanislaw Jarecki and Anna Lysyanskaya. Adaptively secure threshold cryptography: Introducing concurrency, removing erasures. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 221–242. Springer, Heidelberg, May 2000.

JOR18. Charanjit S. Jutla, Miyako Ohkubo, and Arnab Roy. Improved (almost) tightly-secure structure-preserving signatures. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 123–152. Springer, Heidelberg, March 2018.

JR17. Charanjit S. Jutla and Arnab Roy. Improved structure preserving signatures under standard bilinear assumptions. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 183–209. Springer, Heidelberg, March 2017.

KG20. Chelsea Komlo and Ian Goldberg. FROST: Flexible round-optimized Schnorr threshold signatures. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 34–65. Springer, Heidelberg, October 2020.

KPW15. Eike Kiltz, Jiaxin Pan, and Hoeteck Wee. Structure-preserving signatures from standard assumptions, revisited. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 275–295. Springer, Heidelberg, August 2015.

KW15. Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015.

LJY16. Benoît Libert, Marc Joye, and Moti Yung. Born and raised distributively: Fully distributed non-interactive adaptively-secure threshold signatures with short shares. *Theor. Comput. Sci.*, 645:1–24, 2016.

LP01. Anna Lysyanskaya and Chris Peikert. Adaptive security in the threshold setting: From cryptosystems to signature schemes. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 331–350. Springer, Heidelberg, December 2001.

LPJY13. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Linearly homomorphic structure-preserving signatures and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 289–307. Springer, Heidelberg, August 2013.

LPY15. Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 296–316. Springer, Heidelberg, August 2015.

MRV16. Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Heidelberg, December 2016.

Ped92. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, Heidelberg, August 1992.

Sch91. Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991.

Sha79. Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

WMYC23. Harry W. H. Wong, Jack P. K. Ma, Hoover H. F. Yin, and Sherman S. M. Chow. Real threshold ECDSA. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023.