Toothpicks: More Efficient Fork-Free Two-Round Multi-Signatures

Jiaxin Pan [1] Benedikt Wagner [2]

February 26, 2024

[1] University of Kassel, Kassel, Germany
jiaxin.pan@uni-kassel.de
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany,
Saarland University, Saarbrücken, Germany
benedikt.wagner@cispa.de

Abstract

Tightly secure cryptographic schemes can be implemented with standardized parameters, while still having a sufficiently high security level backed up by their analysis. In a recent work, Pan and Wagner (Eurocrypt 2023) presented the first tightly secure two-round multi-signature scheme without pairings, called Chopsticks. While this is an interesting first theoretical step, Chopsticks is much less efficient than its non-tight counterparts.

In this work, we close this gap by proposing a new tightly secure two-round multi-signature scheme that is as efficient as non-tight schemes. Our scheme is based on the DDH assumption without pairings. Compared to Chopsticks, we reduce the signature size by more than a factor of 3 and the communication complexity by more than a factor of 2.

Technically, we achieve this as follows: (1) We develop a new pseudorandom path technique, as opposed to the pseudorandom matching technique in Chopsticks. (2) We construct a more efficient commitment scheme with suitable properties, which is an important primitive in both our scheme and Chopsticks. Surprisingly, we observe that the commitment scheme does not have to be binding, enabling our efficient construction.

Keywords: Multi-Signatures, Tightness, Commitment Scheme, Lossy Identification

1 Introduction

A multi-signature scheme [IN83, BN06] allows a group of signers to jointly sign a message. Naively, every signer could sign the message locally, and we concatenate the resulting signatures. As the number of signers grows large, this results in impractical signature sizes, and so we aim for a more clever solution with compact signatures, potentially at the cost of introducing interaction. Early constructions of multi-signatures have been presented and analyzed in a variety of models [MOR01, Bol03, LOS+06, DEF+19, CKM21], mostly differing in how keys are generated, registered, and verified. Nowadays, the accepted de facto standard for multi-signatures is the plain public key model [BN06], where each signer generates his key pair independently. In this work, we focus on constructions in the said model, proven in the random oracle model [BR93] from assumptions over cyclic groups without pairings. We look at this problem from the perspective of concrete security, which we explain next.

Concrete Security. Cryptographic security proofs follow a common approach: Assuming the existence of an adversary with advantage ϵ_{S} against the security of a construction S, we construct a reduction with roughly the same running time that solves some hard underlying problem Π with probability ϵ_{Π}. Typically, ϵ_{Π} and ϵ_{S} are related via a bound of the form $\epsilon_{\mathsf{S}} \leq L \cdot \epsilon_{\Pi}$, where L is called the *security loss*.

On the one hand, this bound can be treated as a purely qualitative and asymptotic statement, and any polynomial (in the security parameter) L is sufficient to show security. On the other hand, interpreting

the bound as a quantitative statement about the concrete security level, it is desirable to minimize L. In the optimal case, L is a small constant, and we call the proof *tight*. There are two ways to interpret the security bound quantitatively: In the first interpretation, we want to achieve 128 bits of security for S. Then, we need to set our parameters such that cryptanalytic results suggest that Π is $128 + \log(L)$ bits hard. Such parameters include, for example, groups over which we implement the scheme. In the second interpretation, we fix parameters for which we believe that Π is 128-bit hard. Then, we are confident in having $128 - \log(L)$ bits of security for our scheme S, according to the concrete security bound. While the first interpretation compensates for the security loss and results in more secure schemes in theory, it is far from what is done in practice, where practitioners use standardized parameters to implement schemes, because these parameters are well-understood and there are highly optimized implementations for them. Using a different cyclic group for every scheme is just not feasible. For that reason, we stick to the second interpretation.

The Price of Tightness. A large body of research is centered around the concept of tightness, and many primitives have been studied in this regard. Prominent examples include public-key encryption [BBM00, HJ12, BJLS16, GHKW16, Hof17] and key exchange [BHJ+15, GJ18, LLGW20, HJK+21, DG21], as well as signatures [KW03, HJ12, AFLT12, BKKP15, BJLS16, BL16, KMP16, DGJL21, PW22] and related primitives [CW13, BKP14, GHKP18, LP20, PW23]. Unfortunately, tightness often comes at a price in terms of efficiency. This is particularly true for the first tightly secure constructions of some primitive. For instance, in the first public-key encryption scheme with tight security against chosen-ciphertext attacks, due to Hofheinz and Jager [HJ12], ciphertexts contain a linear (in the security parameter) number of group elements as overhead[1], while the respective non-tight scheme [KD04] has a constant ciphertext overhead. Clearly, such an overhead is not acceptable in practice, and so researchers strive for the holy grail of concrete security: Tightly secure constructions with minimal efficiency penalty.

Two-Round Multi-Signatures. In their seminal work [BN06], Bellare and Neven not only introduced the plain public key model but also presented constructions of three-round multi-signatures. Their first scheme is based on Schnorr identification [Sch91]. As typical for Schnorr-based constructions, a rewinding and guessing strategy is used to prove security from the Discrete Logarithm Assumption (DLOG), resulting in a highly non-tight scheme. Their second scheme is tightly secure based on the Decisional Diffie-Hellman (DDH) Assumption. In subsequent works, three-round schemes with so-called key aggregation have been proposed [MPSW19, BDN18, FH21]. With this extension, it is possible to compute a short aggregated key from a set of keys, which can later be used to verify signatures. More recent works concentrate on two-round signing protocols [NRSW20, NRS21, BD21, AB21, CKM21, DOTT21, BTT22, TZ23]. On the downside, many of these schemes require interactive assumptions [NRS21, CKM21, AB21], and all of them fail to provide meaningful concrete security guarantees due to the use of (double) rewinding[2].

In a recent result, Pan and Wagner [PW23] constructed the first tightly secure multi-signature scheme, called Chopsticks II. In particular, their scheme neither relies on rewinding, nor on any guessing argument. However, the price of tightness is high: Signatures and communication complexity in Chopsticks II are about 5 times and 3 times as large as in one of the most efficient non-tight two-round schemes, HBMS, respectively.

Our Goal. We aim to reduce the efficiency gap between tight and non-tight two-round multi-signatures. Concretely, we aim for two-round multi-signatures in the pairing-free discrete logarithm setting, based on well-studied assumptions in the random oracle model. Our constructions should have a minimal security loss, and be efficient in terms of signature size and communication complexity.

1.1 Our Contribution

We reach our goal by constructing a new two-round multi-signature scheme that achieves the best of two worlds:

- *Tightness.* Our scheme is tightly secure based on the DDH assumption. When instantiated over a standardized 128-bit secure group, its security guarantee is 126-bit, which is formally supported by our proofs. In contrast, non-tight schemes relying on rewinding do not guarantee any meaningful security level.

[1] The ciphertext overhead is the size of the ciphertext minus the size of the message.
[2] We do not consider proofs in the (idealized) algebraic group model [FKL18].

Scheme	Rounds	Key Aggregation	Assumption	Loss
BN [BN06]	3	✗	DLOG	$\Theta(Q_H/\epsilon)$
BN+ [BN06]	3	✗	DDH	$\Theta(1)$
Musig [MPSW19, BDN18]	3	✓	DLOG	$\Theta(Q_H^3/\epsilon^3)$
Musig+ [FH21]	3	✓	DDH	$\Theta(1)$
Musig2 [NRS21]	2	✓	AOMDL	$\Theta(Q_H^3/\epsilon^3)$
HBMS [BD21]	2	✓	DLOG	$\Theta(Q_S^4 Q_H^3/\epsilon^3)$
TZ [TZ23]	2	✓	DLOG	$\Theta(Q_H^3/\epsilon^3)$
TSSHO [TSS+23]	2	✓	DDH	$\Theta(Q_S)$
Chopsticks I [PW23]	2	✓	DDH	$\Theta(Q_S)$
Chopsticks II [PW23]	2	✗	DDH	$\Theta(1)$
Section 4.1	2	✓	DDH	$\Theta(Q_S)$
Section 4.2	2	✗	DDH	$\Theta(1)$

Table 1: Comparison of multi-signature schemes in the discrete logarithm setting without pairings in the plain public key model. We compare the number of rounds, whether the schemes support key aggregation, the assumption the schemes rely on, and the security loss, where Q_H, Q_S denote the number of random oracle and signing queries, respectively, and ϵ denotes the advantage of an adversary against the scheme. For the security loss, we do not consider proofs in the algebraic group model. We do not list [NRSW20] as it is prohibitively inefficient due to the use of heavy cryptographic machinery.

- *Efficiency.* Our scheme is as efficient as the state-of-the-art non-tight schemes. Concretely, the communication complexity per signer for our scheme is comparable to HBMS [BD21] and about 1.5 times smaller than TZ [TZ23] and Musig2 [NRS21]. The signature size is only about 1.5 times larger than for the non-tight schemes. Compared to Chopsticks II [PW23], this significantly reduces the efficiency cost of tightness. Concretely, our scheme outperforms Chopsticks II by a factor of more than 3 and 2 in terms of signature size and communication complexity, respectively.

In addition, we present a non-tight scheme with an acceptable security loss, namely, linear in the number of signing queries. The advantage of this scheme is that it supports key aggregation. A similar but much less efficient scheme, Chopsticks I, has been proposed in [PW23]. We compare our schemes with previous schemes in terms of security (see Table 1) and asymptotic (see Table 2) and concrete (see Table 3) efficiency.

From a technical perspective, our first contribution is a new pseudorandom path technique, as opposed to the pseudorandom matching technique in Chopsticks [PW23]. This new technique allows us to reduce the size of signatures and communication by a factor of two. Our second technical insight is that, somewhat surprisingly, we do not need a binding commitment as in Chopsticks [PW23]. Instead, we can significantly relax the binding property of our commitment to hold up to cosets of a certain subspace. This enables more efficient instantiations, further improving the efficiency of our schemes. To show that this relaxation does not introduce problems in terms of security, we identify a strong soundness property many natural lossy identification schemes [KW03, AFLT12, KMP16] have. We are confident that this combination of a weak commitment with lossy identification is of independent interest.

1.2 Technical Overview

In this paper, we introduce two major technical improvements to the Chopsticks schemes, namely, a novel overall construction strategy and a more efficient commitment scheme that can be used both in Chopsticks and in our schemes.

Background. We start with lossy identification schemes [KW03, AFLT12, KMP16]. In such a scheme, there are two ways to set up public keys pk. As usual, one can set up pk with a secret key sk. Alternatively, one can set up pk in lossy mode. In this mode, not even an unbounded prover can make the verifier accept, which is called lossy soundness. This paradigm turns out to be useful when constructing tighly secure Fiat-Shamir style signatures [KW03, AFLT12, KMP16] or three-round multi-signatures [BN06, FH21]. To apply it in the two-round setting, Pan and Wagner [PW23] leveraged a homomorphic dual-mode commitment. Concretely, such a commitment has two ways of setting up commitment keys ck. In the

Scheme	Rounds	Public Key	Communication	Signature
BN [BN06]	3	$1\langle\mathbb{G}\rangle$	$1\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle + 2\lambda$	$1\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle$
BN+ [BN06]	3	$2\langle\mathbb{G}\rangle$	$2\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle + 2\lambda$	$2\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle$
Musig [MPSW19, BDN18]	3	$1\langle\mathbb{G}\rangle$	$1\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle + 2\lambda$	$1\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle$
Musig+ [FH21]	3	$2\langle\mathbb{G}\rangle$	$2\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle + 2\lambda$	$2\langle\mathbb{Z}_p\rangle$
Musig2 [NRS21]	2	$1\langle\mathbb{G}\rangle$	$4\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle$	$1\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle$
HBMS [BD21]	2	$1\langle\mathbb{G}\rangle$	$1\langle\mathbb{G}\rangle + 2\langle\mathbb{Z}_p\rangle$	$1\langle\mathbb{G}\rangle + 2\langle\mathbb{Z}_p\rangle$
TZ [TZ23]	2	$1\langle\mathbb{G}\rangle$	$4\langle\mathbb{G}\rangle + 2\langle\mathbb{Z}_p\rangle$	$1\langle\mathbb{G}\rangle + 2\langle\mathbb{Z}_p\rangle$
TSSHO [TSS+23]	2	$2\langle\mathbb{G}\rangle$	$2\langle\mathbb{G}\rangle + 2\langle\mathbb{Z}_p\rangle$	$3\langle\mathbb{Z}_p\rangle$
Chopsticks I [PW23]	2	$2\langle\mathbb{G}\rangle$	$3\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle + \lambda$	$3\langle\mathbb{G}\rangle + 4\langle\mathbb{Z}_p\rangle$
Chopsticks II [PW23]	2	$4\langle\mathbb{G}\rangle$	$6\langle\mathbb{G}\rangle + 2\langle\mathbb{Z}_p\rangle + \lambda + 1$	$6\langle\mathbb{G}\rangle + 8\langle\mathbb{Z}_p\rangle + N$
Section 4.1	2	$2\langle\mathbb{G}\rangle$	$2\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle + \lambda$	$3\langle\mathbb{Z}_p\rangle + 2\lambda$
Section 4.2	2	$4\langle\mathbb{G}\rangle$	$2\langle\mathbb{G}\rangle + 1\langle\mathbb{Z}_p\rangle + \lambda + 1$	$3\langle\mathbb{Z}_p\rangle + 2\lambda + N$

Table 2: Asymptotic efficiency comparison of multi-signature schemes in the discrete logarithm setting without pairings in the plain public key model. We compare the number of rounds, the size of public keys, the communication complexity per signer, and the signature size. We denote the size of a group element by $\langle\mathbb{G}\rangle$ and the size of a field element by $\langle\mathbb{Z}_p\rangle$. Here, λ is a statistical security parameter, and N is the number of signers. Schemes below the line have two rounds and avoid rewinding, see Table 1. We do not list [NRSW20] as it is prohibitively inefficient due to the use of heavy cryptographic machinery.

Scheme	Security	Public Key	Communication	Signature
Musig2	9	33	164	65
HBMS	-11	33	97	97
TZ	8	33	196	97
TSSHO	106	66	130	96
Chopsticks I	106	66	147	227
Chopsticks II	126	132	278	470
Section 4.1	106	66	114	128
Section 4.2	125	132	114	144

Table 3: Concrete efficiency and security comparison of two-round multi-signature schemes in the discrete logarithm setting without pairings in the plain public key model. We compare the security level guaranteed by the security bound in the random oracle model assuming the underlying assumption is 128-bit hard, the size of public keys, the communication complexity per signer, and the signature size. Sizes are given in bytes and rounded. We do not list [NRSW20] as it is prohibitively inefficient due to the use of heavy cryptographic machinery.

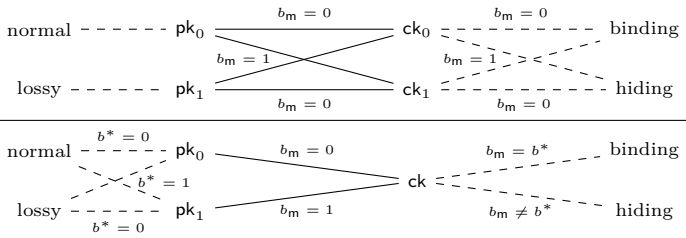

Figure 1: Visualization of the pseudorandom matching technique from Chopsticks [PW23] (top), and our new pseudorandom path technique (bottom). Here, b^* is a random bit sampled by the game, b_m is the pseudorandom bit that the signer chooses for message m, normal edges indicate how keys are paired in the scheme, and dotted edges indicate how keys are set up in the proof.

hiding mode, ck is generated in combination with a (weak) equivocation trapdoor. In the other mode, commitments are statistically binding. Given such a commitment and a lossy identification scheme, a signature for a message m contains transcripts of the lossy identification scheme, where some parts are given in a committed form. For these parts, the signature also contains the respective opening information. Importantly, the commitment key is derived from m, e.g., as $\mathsf{ck} := \mathsf{H}(\mathsf{m})$, where H is a random oracle. Abstractly, Pan and Wagner [PW23] identified the following properties:

- *Simulation via Secret Keys.* A reduction can simulate the signing oracle using the secret key if pk is in the normal mode. The mode of ck is not relevant.
- *Simulation via Trapdoors.* A reduction can simulate the signing oracle using the trapdoor if ck is in the hiding mode. The mode of pk is not relevant.
- *Forgery.* To show security without rewinding, the adversary must output a forgery with respect to a lossy pk and a binding ck.

The Chopsticks Approach: Pseudorandom Matching. In the proof of their first scheme, Pan and Wagner [PW23] use these properties by sampling all commitment keys with a trapdoor, allowing them to simulate signing even if the public key is lossy. Only for the forgery message the associated commitment key ck^* is set up to be binding. Then, the proof can be finished without rewinding. On the downside, this approach requires guessing the query defining ck^*, leading to a security loss.

A well-known trick to avoid such a guessing argument is the Katz-Wang approach [GJKW07]. Here, each message would specify two commitment keys $\mathsf{ck}_0 := \mathsf{H}(0, \mathsf{m})$ and $\mathsf{ck}_1 := \mathsf{H}(1, \mathsf{m})$, and a signer individually would pick a pseudorandom bit b_m for each message and then use $\mathsf{ck}_{b_\mathsf{m}}$. It turns out that this trick is not applicable here, as each signer has to use the same commitment key.

To overcome this obstacle, Pan and Wagner proposed the pseudorandom matching technique. Namely, each signer has two public keys $\mathsf{pk}_0, \mathsf{pk}_1$, and both message-dependent commitment keys ck_0 and ck_1 are used. That is, the protocol is run twice in parallel and the signature now contains two transcripts instead of one. Importantly, each signer uses the pseudorandom bit b_m to decide which public key to match with which commitment key. We illustrate the pseudorandom matching technique in Figure 1 (top). In the proof, one can set pk_1 to lossy and always match it with the trapdoor commitment key $\mathsf{ck}_{1-b_\mathsf{m}}$ for signing queries. In this way, it is possible to simulate the $(\mathsf{pk}_1, \mathsf{ck}_{1-b_\mathsf{m}})$ side via the trapdoor, and the $(\mathsf{pk}_0, \mathsf{ck}_{b_\mathsf{m}})$ side via the secret key. At the same time, with probability $1/2$, the adversary will match the lossy pk_1 with the binding $\mathsf{ck}_{b_\mathsf{m}}$ for the forgery, finishing the proof. While this is an elegant trick, it introduces a significant overhead for both signature size and communication complexity.

Our Approach: Pseudorandom Paths. We avoid this overhead by using our new pseudorandom path technique, as illustrated in Figure 1 (bottom). Our first observation is that for the argument used to finish the proof in Chopsticks, only one of the two paths, namely the $(\mathsf{pk}_1, \mathsf{ck}_{b_\mathsf{m}})$ path, is used. Instead of simply omitting one of the paths, which leads to problems similar to the naive Katz-Wang approach, let us see what happens if we go back to a solution in which there is only one commitment key ck per message m. If we also reduce the number of keys per signer back to one, we end up with the guessing-based solution again. So, we keep the two keys $\mathsf{pk}_0, \mathsf{pk}_1$ per signer, and let each signer pseudorandomly decide which key $\mathsf{pk}_{b_\mathsf{m}}$ to use in the signing interaction. In our proof, we can set up ck with a trapdoor if the lossy key

pk_1 is used, and we can set it up in binding mode if the normal key pk_0 is used. Unfortunately, without additional tricks, this strategy is doomed: The adversary could always use pk_0 in its forgery, which is not lossy. In our final solution, we therefore pick a bit b^* at random at the beginning of our simulation. Then, we set pk_{b^*} to normal and pk_{1-b^*} to lossy. We adapt the sampling of ck accordingly. By carrying out all arguments in the correct order, we can argue that in one of four cases, the adversary used the lossy key pk_{1-b^*} with a binding commitment key in its forgery. Let us explain the idea for that with our illustration (Figure 1, bottom) at hand. Every signature corresponds to a pseudorandom path from the left to the right. The bits are set up in a way that ensures the following:

- *Simulation of Signing.* If $\mathsf{pk}_{b_\mathsf{m}}$ is used, the path connects the lossy vertex to the trapdoor vertex, or the normal vertex to the binding vertex. In both cases, we can simulate.
- *Forgery.* The probability that the path associated with the forgery starts at the lossy vertex is $1/2$ and conditioned on that, the probability that the path ends at the binding vertex is also $1/2$.

With this technique, communication and signatures now only consist of one transcript of the lossy identification scheme, as opposed to two transcripts in the pseudorandom matching technique.

The Chopsticks Commitment. So far, we have reduced the size of signatures and communication by a factor of two. At this point, the size is mostly dominated by the size of commitments com and openings φ. It is therefore instructive to recall the commitment instantiation from [PW23] and see if we can optimize it. The instantiation from [PW23] allows to commit to pairs of group elements $(R_1, R_2) \in \mathbb{G}^2$ via the equation

$$\mathsf{com} = (C_0, C_1, C_2) \in \mathbb{G}^3, \text{ where } \begin{pmatrix} C_0 \\ C_1 \\ C_2 \end{pmatrix} := \begin{pmatrix} & A_{1,1}^{\alpha} \cdot A_{1,2}^{\beta} \cdot A_{1,3}^{\gamma} \\ R_1 \cdot & A_{2,1}^{\alpha} \cdot A_{2,2}^{\beta} \cdot A_{2,3}^{\gamma} \\ R_2 \cdot & A_{3,1}^{\alpha} \cdot A_{3,2}^{\beta} \cdot A_{3,3}^{\gamma} \end{pmatrix}.$$

Here, $\varphi = (\alpha, \beta, \gamma) \in \mathbb{Z}_p^3$ is the commitment randomness and the $A_{i,j} \in \mathbb{G}$ form the commitment key. In terms of exponents, the commitment has the form

$$\mathbf{E} \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} + \begin{pmatrix} 0 \\ r_1 \\ r_2 \end{pmatrix}$$

for a matrix $\mathbf{E} \in \mathbb{Z}_p^{3\times 3}$ that determines the commitment key. Now, one can prove that this is statistically hiding if $\mathbf{E}$ has full rank, and it is statistically binding if $\mathbf{E}$ has rank 1. It is one of the main technical contributions of [PW23] to introduce a weak equivocation trapdoor for this commitment, i.e., a trapdoor that allows to open commitments to messages (R_1, R_2) of a certain structure.

Strawman Commitment. In terms of efficiency, note that the 3×3 commitment key leads to three group elements per commitment and three exponents per opening. To improve it, our naive idea is to replace this 3×3 structure with a 2×2 structure, thereby saving one group element and exponent. Concretely, we could try to drop the first row of the commitment equation, leading to

$$\mathbf{E} \cdot \begin{pmatrix} \beta \\ \gamma \end{pmatrix} + \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

for a matrix $\mathbf{E} \in \mathbb{Z}_p^{2\times 2}$. Implemented carefully, this is still perfectly hiding with a weak equivocation trapdoor if $\mathbf{E}$ has full rank. Unfortunately, we fail when analyzing the statistically binding mode[3] if $\mathbf{E}$ has rank 1. Concretely, an (unbounded) adversary against binding could output (r_1, r_2) with opening (β, γ) on the one hand, and $(r_1, r_2) + (\beta, \gamma)\mathbf{E}^t$ with opening $(0, 0)$ on the other hand. The first row in the 3×3 scheme prevents this. To save our 2×2 construction without reintroducing such a first row, we thus need additional insights.

Coset Binding. As we have seen, our 2×2 scheme is not (statistically) binding, and as such it is not suitable to instantiate the multi-signature construction. However, we make the crucial observation that the scheme is binding *up to a difference in the span of* $\mathbf{E}$. In other words, if we interpret the commitment as a commitment to cosets of the span of $\mathbf{E}$, the scheme is binding. We call this property *coset binding*. It

[3] If we only have a computationally binding mode, the resulting multi-signature scheme needs to rely on rewinding. Therefore, we have to insist on a statistically binding mode.

is instructive to pinpoint where the overall multi-signature proof fails if we relax binding to coset binding: In the proof of our multi-signature construction, binding shows up in combination with lossy soundness in the very last proof step. To recall, lossy soundness states that even an unbounded prover can not make a verifier of the lossy identification scheme accept, given that pk is in lossy mode. An accepting transcript of the identification scheme has the form (R, c, s) and satisfies $\mathsf{F}(s) - c \cdot \mathsf{pk} = R$ for a linear function F[4]. Roughly, when constructing an unbounded reduction that breaks lossy soundness, we first guess[5] the random oracle query associated with the forgery. On this query, assume now that the adversary sends a commitment com for R with respect to a binding commitment key. In this case, the reduction would non-efficiently extract the committed pair of group elements $R = (R_1, R_2)$ from com and output it in the lossy soundness game. Further, it would appropriately embed the challenge c from the lossy soundness game. Finally, if the guess was correct, the adversary's forgery contains a valid response s, which the reduction can output. Now, it is clear why coset binding is not enough: The adversary is not bound to $R = (R_1, R_2)$, and it can output a response s that is valid for $R' = (R'_1, R'_2) \neq R$, which is of no use for the reduction.

Coset Lossy Soundness to the Rescue. While coset binding seems to be insufficient at first glance, it still gives us a guarantee we may leverage. Namely, by coset binding, $(R_1 = g^{r_1}, R_2 = g^{r_2})$ and $(R'_1 = g^{r'_1}, R'_2 = g^{r'_2})$ as above have to satisfy that $(r_1, r_2)^t - (r'_1, r'_2)^t$ is in the span of $\mathbf{E}$. We want to understand the impact of this guarantee on the lossy soundness game. For that, imagine a modified lossy soundness game where the final equation $\mathsf{F}(s) - c \cdot \mathsf{pk} = R$ only has to hold up to a difference in the span of $\mathbf{E}$. We call this stronger notion of lossy soundness *coset lossy soundness*. In fact, if we can argue that coset lossy soundness holds, then the reduction sketched above goes through assuming coset binding. For that, our main idea is to set up the binding commitment keys such that the span of $\mathbf{E}$ is always contained in the image of F. In this case, we observe that coset lossy soundness is implied by the original lossy soundness notion. This is because, roughly, if $\mathsf{F}(s) - c \cdot \mathsf{pk}$ equals R up to a difference in the span of $\mathbf{E}$, it means that $\mathsf{F}(s) - c \cdot \mathsf{pk} = R + \mathsf{F}(\delta)$ for some δ, and so one can just treat $s - \delta$ as the new s. To summarize our optimized commitment construction, we have seen that lossy soundness of the identification scheme at hand is strong enough to compensate for the relaxed binding notion. We are confident that this insight is applicable in other contexts as well.

2 Preliminaries

By $[L] := \{1, \ldots, L\} \subseteq \mathbb{N}$ we denote the set of the first L natural numbers. Let S be a finite set, $\mathcal{D}$ a distribution, and $\mathcal{A}$ be a probabilistic algorithm. The notation $s \xleftarrow{\$} S$ means that s is sampled uniformly at random from S, and $x \leftarrow \mathcal{D}$ means that x is sampled according to $\mathcal{D}$. The notation $s := \mathcal{A}(x; \rho)$ means that $\mathcal{A}$ outputs s on input x with random coins ρ, and when we write $s \leftarrow \mathcal{A}(x)$, we mean that ρ is sampled uniformly at random. We write $s \in \mathcal{A}(x)$ to indicate that there are coins ρ such that $\mathcal{A}$ outputs s on input x with these coins ρ. Throughout the paper, λ will denote the security parameter, and all algorithms get it (in unary) as input. We use standard cryptographic notions like PPT and negligible.

Multi-Signatures. We define the syntax and security of multi-signatures in the plain public key model [BN06]. Following previous works, e.g., [CKM21, DOTT21, PW23], we assume the public keys participating in the signing protocol are given by a set, and we assume that sets can be ordered canonically, e.g. lexicographically. Thus, we can uniquely encode sets $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$, and we denote such an encoding by $\langle \mathcal{P} \rangle$ throughout the paper. Further, we assume that the honest public key in our security definition is the entry pk_1 in such a set, which is without loss of generality and for simplicity of presentation. In terms of syntax and security, we use the definition from [PW23]. We postpone a formal definition of key aggregation to Appendix A.

Definition 1 (Multi-Signature Scheme)**.** A (two-round) multi-signature scheme is a tuple of PPT algorithms $\mathsf{MS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ with the following syntax:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{par}$ takes as input the security parameter 1^λ and outputs global system parameters par. We assume that par implicitly defines sets of public keys, secret keys, messages and signatures, respectively. All algorithms related to MS take par at least implicitly as input.

[4] We use additive notation when talking about lossy identification from such linear functions in general, and multiplicative notation for the concrete instantiation of the linear function and commitment.

[5] Recall that lossy soundness is a statistical notion, and so guessing is not a problem in terms of tightness at this point.

$\underline{\textbf{Alg } \mathsf{MS.Exec}(\mathcal{P}, \mathcal{S}, \mathsf{m})}$
01 **parse** $\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\} := \mathcal{P}$, $\{\mathsf{sk}_1, \ldots, \mathsf{sk}_N\} := \mathcal{S}$
02 **for** $i \in [N] : (\mathsf{pm}_{1,i}, St_{1,i}) \leftarrow \mathsf{Sig}_0(\mathcal{P}, \mathsf{sk}, \mathsf{m})$
03 $\mathcal{M}_1 := (\mathsf{pm}_{1,1}, \ldots, \mathsf{pm}_{1,N})$
04 **for** $i \in [N] : (\mathsf{pm}_{2,i}, St_{2,i}) \leftarrow \mathsf{Sig}_1(St_{1,i}, \mathcal{M}_1)$
05 $\mathcal{M}_2 := (\mathsf{pm}_{2,1}, \ldots, \mathsf{pm}_{2,N})$
06 **for** $i \in [N] : \sigma_i \leftarrow \mathsf{Sig}_2(St_{2,i}, \mathcal{M}_2)$
07 **if** $\exists i \neq j \in [N]$ s.t. $\sigma_i \neq \sigma_j$: **return** $\perp$
08 **return** $\sigma := \sigma_1$

Figure 2: Algorithm MS.Exec for a multi-signature scheme $\mathsf{MS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$. The algorithm specifies an honest execution of the signing protocol Sig among N signers with public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_N$ and secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$ for a message m.

- $\mathsf{Gen}(\mathsf{par}) \to (\mathsf{pk}, \mathsf{sk})$ takes as input system parameters par, and outputs a public key pk and a secret key sk.
- $\mathsf{Sig} = (\mathsf{Sig}_0, \mathsf{Sig}_1, \mathsf{Sig}_2)$ is split into three algorithms:
  - $\mathsf{Sig}_0(\mathcal{P}, \mathsf{sk}, \mathsf{m}) \to (\mathsf{pm}_1, St_1)$ takes as input a set $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ of public keys, a secret key sk, and a message m, and outputs a protocol message pm_1 and a state St_1.
  - $\mathsf{Sig}_1(St_1, \mathcal{M}_1) \to (\mathsf{pm}_2, St_2)$ takes as input a state St_1 and a tuple $\mathcal{M}_1 = (\mathsf{pm}_{1,1}, \ldots, \mathsf{pm}_{1,N})$ of protocol messages, and outputs a protocol message pm_2 and a state St_2.
  - $\mathsf{Sig}_2(St_2, \mathcal{M}_2) \to \sigma_i$ takes as input a state St_2 and a tuple $\mathcal{M}_2 = (\mathsf{pm}_{2,1}, \ldots, \mathsf{pm}_{2,N})$ of protocol messages, and outputs a signature σ.
- $\mathsf{Ver}(\mathcal{P}, \mathsf{m}, \sigma) \to b$ is deterministic, takes as input a set $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ of public keys, a message m, and a signature σ, and outputs a bit $b \in \{0, 1\}$.

We require that MS is complete in the following sense. For all $\mathsf{par} \in \mathsf{Setup}(1^\lambda)$, all $N = \mathsf{poly}(\lambda)$, all $(\mathsf{pk}_j, \mathsf{sk}_j) \in \mathsf{Gen}(\mathsf{par})$ for $j \in [N]$, and all messages m, we have

$$\Pr\left[\mathsf{Ver}(\mathcal{P}, \mathsf{m}, \sigma) = 1 \;\middle|\; \begin{array}{l} \mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}, \mathcal{S} = \{\mathsf{sk}_1, \ldots, \mathsf{sk}_N\}, \\ \sigma \leftarrow \mathsf{MS.Exec}(\mathcal{P}, \mathcal{S}, \mathsf{m}) \end{array}\right] = 1,$$

where algorithm MS.Exec is defined in Figure 2.

Definition 2 (MS-EUF-CMA Security)**.** Let $\mathsf{MS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a multi-signature scheme and consider the game **MS-EUF-CMA** defined in Figure 3. We say that MS is MS-EUF-CMA secure, if for all PPT adversaries $\mathcal{A}$, the following advantage is negligible:

$$\mathsf{Adv}^{\mathsf{MS\text{-}EUF\text{-}CMA}}_{\mathcal{A},\mathsf{MS}}(\lambda) := \Pr\left[\textbf{MS-EUF-CMA}^{\mathcal{A}}_{\mathsf{MS}}(\lambda) \Rightarrow 1\right].$$

Assumptions. In this work, we base our constructions on the well-known DDH assumption over a (family of) cyclic groups $\mathbb{G}$ of prime order p with generator g. To recall, the assumption states that it is infeasible to distinguish $(g, h, g^a, h^a) \in \mathbb{G}^4$ from $(g, h, u, v) \in \mathbb{G}^4$ for $h, u, v \xleftarrow{\$} \mathbb{G}$ and $a \xleftarrow{\$} \mathbb{Z}_p$. For convenience, we define its multi-instance variant Q-DDH. It is well-known that Q-DDH is tightly implied by DDH using random self-reducibility [EHK+13].

Definition 3 (DDH Assumption)**.** Let GGen be an algorithm that on input 1^λ outputs the description of a prime order group $\mathbb{G}$ of order p with generator g. We say that the DDH assumption holds relative to GGen, if for all PPT algorithms $\mathcal{A}$, the following advantage is negligible:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{A},\mathsf{GGen}}(\lambda) := \Bigg| &\Pr\left[\mathcal{A}(\mathbb{G}, p, g, h, g^a, h^a) = 1 \;\middle|\; \begin{array}{l}(\mathbb{G}, g, p) \leftarrow \mathsf{GGen}(1^\lambda), \\ h \xleftarrow{\$} \mathbb{G}, a \xleftarrow{\$} \mathbb{Z}_p\end{array}\right] \\ &- \Pr\left[\mathcal{A}(\mathbb{G}, p, g, h, u, v) = 1 \;\middle|\; \begin{array}{l}(\mathbb{G}, g, p) \leftarrow \mathsf{GGen}(1^\lambda), \\ h, u, v \xleftarrow{\$} \mathbb{G}\end{array}\right]\Bigg|.\end{aligned}$$

Game MS-EUF-CMA$^{\mathcal{A}}_{\mathsf{MS}}(\lambda)$

$\mathsf{par} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{par})$
$\mathrm{SIG} := (\mathrm{SIG}_0, \mathrm{SIG}_1, \mathrm{SIG}_2)$
$(\mathcal{P}^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathrm{SIG}}(\mathsf{par}, \mathsf{pk})$
if $\mathsf{pk} \notin \mathcal{P}^*$: **return** 0
if $(\mathcal{P}^*, \mathsf{m}^*) \in \mathsf{Queried}$: **return** 0
return $\mathsf{Ver}(\mathcal{P}^*, \mathsf{m}^*, \sigma^*)$

Oracle $\mathrm{SIG}_0(\mathcal{P}, \mathsf{m})$

parse $\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\} := \mathcal{P}$
if $\mathsf{pk}_1 \neq \mathsf{pk}$: **return** $\perp$
$\mathsf{Queried} := \mathsf{Queried} \cup \{(\mathcal{P}, \mathsf{m})\}$
$ctr := ctr + 1,\ sid := ctr$
$\mathsf{round}[sid] := 1$
$(\mathsf{pm}_1, St_1) \leftarrow \mathsf{Sig}_0(\mathcal{P}, \mathsf{sk}, \mathsf{m})$
$(\mathsf{pm}_1[sid], St_1[sid]) := (\mathsf{pm}_1, St_1)$
return $(\mathsf{pm}_1[sid], sid)$

Oracle $\mathrm{SIG}_1(sid, \mathcal{M}_1)$

if $\mathsf{round}[sid] \neq 1$: **return** $\perp$
parse $(\mathsf{pm}_{1,1}, \ldots, \mathsf{pm}_{1,N}) := \mathcal{M}_1$
if $\mathsf{pm}_1[sid] \neq \mathsf{pm}_{1,1}$: **return** $\perp$
$\mathsf{round}[sid] := \mathsf{round}[sid] + 1$
$(\mathsf{pm}_2, St_2) \leftarrow \mathsf{Sig}_1(St_1[sid], \mathcal{M}_1)$
$(\mathsf{pm}_2[sid], St_2[sid]) := (\mathsf{pm}_2, St_2)$
return $\mathsf{pm}_2[sid]$

Oracle $\mathrm{SIG}_2(sid, \mathcal{M}_2)$

if $\mathsf{round}[sid] \neq 2$: **return** $\perp$
parse $(\mathsf{pm}_{2,1}, \ldots, \mathsf{pm}_{2,N}) := \mathcal{M}_2$
if $\mathsf{pm}_2[sid] \neq \mathsf{pm}_{2,1}$: **return** $\perp$
$\mathsf{round}[sid] := \mathsf{round}[sid] + 1$
$\sigma \leftarrow \mathsf{Sig}_2(St_2[sid], \mathcal{M}_2)$
return σ

Figure 3: The game **MS-EUF-CMA** for a (two-round) multi-signature scheme MS and an adversary $\mathcal{A}$. To simplify the presentation, we assume that the canonical ordering of sets is chosen such that pk is always at the first position if it is included.

Definition 4 (Q-DDH Assumption)**.** Let GGen be an algorithm that on input 1^λ outputs the description of a prime order group $\mathbb{G}$ of order p with generator g. We say that the Q-DDH assumption holds relative to GGen, if for all PPT algorithms $\mathcal{A}$, the following advantage is negligible:

$$\mathsf{Adv}^{Q\text{-DDH}}_{\mathcal{A},\mathsf{GGen}}(\lambda) := \left| \Pr\left[\mathcal{A}\left(\mathbb{G}, p, g, h, (g^{a_i}, h^{a_i})_{i=1}^{Q}\right) = 1 \;\middle|\; \begin{array}{l} (\mathbb{G}, g, p) \leftarrow \mathsf{GGen}(1^\lambda), \\ h \xleftarrow{\$} \mathbb{G}, \\ \forall i \in [Q] :\ a_i \xleftarrow{\$} \mathbb{Z}_p \end{array} \right] - \Pr\left[\mathcal{A}\left(\mathbb{G}, p, g, h, (u_i, v_i)_{i=1}^{Q}\right) = 1 \;\middle|\; \begin{array}{l} (\mathbb{G}, g, p) \leftarrow \mathsf{GGen}(1^\lambda), \\ h \xleftarrow{\$} \mathbb{G}, \\ \forall i \in [Q] :\ u_i, v_i \xleftarrow{\$} \mathbb{G} \end{array} \right] \right|.$$

3 Our Building Blocks

In this section, we introduce two building blocks we will use in our constructions. Namely, we make use of linear function families and a special kind of commitment scheme. This is similar to what is done in [PW23]. However, compared to [PW23], a crucial observation is that we can weaken the requirements for the commitment scheme, thereby enabling a more efficient instantiation. To compensate, we strengthen the requirements for the linear function family in terms of soundness, which is for free in terms of efficiency.

3.1 Linear Functions

Our first building block is a family of linear functions with suitable properties. Linear function families are a widely used abstraction [HKL19, KLR21, CAHL+22, PW23, TZ23] that allows us to describe our protocols in a modular and uncluttered fashion. In terms of syntax, we use the definition given in [PW23].

Definition 5 (Linear Function Family)**.** A linear function family (LFF) is a tuple of PPT algorithms $\mathsf{LF} = (\mathsf{Gen}, \mathsf{F})$ with the following syntax:

- $\mathsf{Gen}(1^\lambda) \to \mathsf{par}$ takes as input the security parameter 1^λ and outputs parameters par. We assume that par implicitly defines the following sets:
  - A set of scalars $\mathcal{S}_{\mathsf{par}}$, which forms a field.
  - A domain $\mathcal{D}_{\mathsf{par}}$, which forms a vector space over $\mathcal{S}_{\mathsf{par}}$.

– A range $\mathcal{R}_{\mathsf{par}}$, which forms a vector space over $\mathcal{S}_{\mathsf{par}}$.

We omit the subscript par if it is clear from the context, and naturally denote the operations of these fields and vector spaces by $+$ and $\cdot$. We assume that these operations can be evaluated efficiently.

- $\mathsf{F}(\mathsf{par}, x) \to X$ is deterministic, takes as input parameters par, an element $x \in \mathcal{D}$, and outputs an element $X \in \mathcal{R}$. For all parameters par, $\mathsf{F}(\mathsf{par}, \cdot)$ realizes a homomorphism, i.e.

$$\forall s \in \mathcal{S}, x, y \in \mathcal{D} : \ \mathsf{F}(\mathsf{par}, s \cdot x + y) = s \cdot \mathsf{F}(\mathsf{par}, x) + \mathsf{F}(\mathsf{par}, y).$$

We omit the input par if it is clear from the context.

In [PW23], key indistinguishability and lossy soundness are defined to capture the set of properties that makes linear function families amenable for the use in lossy identification [AFLT12]. We recall these definitions from [PW23]. Further, we introduce strengthened definitions, which allow us to weaken the properties for other building blocks. Concretely, we relax the winning condition for lossy soundness, such that it has to hold up to an arbitrary shift in the image of the linear function, leading to coset lossy soundness. In Appendix A, we also adapt the definition of aggregation lossy soundness from [PW23] accordingly, leading to coset aggregation lossy soundness. Importantly, we show in Lemmata 1 and 8 that the strengthened definitions come for free.

Definition 6 (Key Indistinguishability)**.** Let $\mathsf{LF} = (\mathsf{Gen}, \mathsf{F})$ be a linear function family. We say that LF satisfies key indistinguishability, if for any PPT algorithm $\mathcal{A}$, the following advantage is negligible:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{A},\mathsf{LF}}(\lambda) := |\Pr\left[\mathcal{A}(\mathsf{par}, X) = 1 \middle| \ \mathsf{par} \leftarrow \mathsf{Gen}(1^\lambda),\ x \xleftarrow{\$} \mathcal{D},\ X := \mathsf{F}(x)\ \right] \\ - \Pr\left[\mathcal{A}(\mathsf{par}, X) = 1 \middle| \ \mathsf{par} \leftarrow \mathsf{Gen}(1^\lambda),\ X \xleftarrow{\$} \mathcal{R}\ \right]|.\end{aligned}$$

Definition 7 (Lossy Soundness)**.** Let $\mathsf{LF} = (\mathsf{Gen}, \mathsf{F})$ be a linear function family. We say that LF satisfies ε_{l}-lossy soundness, if for any unbounded algorithm $\mathcal{A}$, the following probability is at most ε_{l}:

$$\Pr\left[\mathsf{F}(s) - c \cdot X = R \middle| \begin{array}{l}\mathsf{par} \leftarrow \mathsf{Gen}(1^\lambda),\ X \xleftarrow{\$} \mathcal{R}, \\ (St, R) \leftarrow \mathcal{A}(\mathsf{par}, X), \\ c \xleftarrow{\$} \mathcal{S},\ s \leftarrow \mathcal{A}(St, c)\end{array}\right].$$

Definition 8 (Coset Lossy Soundness)**.** Let $\mathsf{LF} = (\mathsf{Gen}, \mathsf{F})$ be a linear function family. We say that LF satisfies ε_{l}-coset lossy soundness, if for any unbounded algorithm $\mathcal{A}$, the following probability is at most ε_{l}:

$$\Pr\left[\mathsf{F}(s) - c \cdot X \in R + \mathsf{F}(\mathcal{D}) \middle| \begin{array}{l}\mathsf{par} \leftarrow \mathsf{Gen}(1^\lambda),\ X \xleftarrow{\$} \mathcal{R}, \\ (St, R) \leftarrow \mathcal{A}(\mathsf{par}, X), \\ c \xleftarrow{\$} \mathcal{S},\ s \leftarrow \mathcal{A}(St, c)\end{array}\right].$$

Lemma 1. *Let* LF *be a linear function family, such that for any* $\mathsf{par} \in \mathsf{Gen}(1^\lambda)$*, the domain* $\mathcal{D}_{\mathsf{par}}$ *can be enumerated. Then, if* LF *satisfies* ε_{l}*-lossy soundness, it also satisfies* ε_{l}*-coset lossy soundness.*

Proof. To prove the claim, it is sufficient to describe an (unbounded) reduction $\mathcal{B}$, that turns any algorithm $\mathcal{A}$ running in the coset lossy soundness game into an algorithm in the lossy soundness game. The reduction $\mathcal{B}$ gets as input parameters par and an element $X \in \mathcal{R}$ from the lossy soundness game. It runs $\mathcal{A}$ on input par and X and gets an output R in return, which it passes to the lossy soundness game. In return, it receives $c \in \mathcal{S}$, and forwards it to $\mathcal{A}$, which outputs $s \in \mathcal{D}$. If $\mathcal{A}$ breaks coset lossy soundness, i.e, $\mathsf{F}(s) - c \cdot X \in R + \mathsf{F}(\mathcal{D})$, then there is a $\delta \in \mathcal{D}$ such that $\mathsf{F}(s) - c \cdot X = R + \mathsf{F}(\delta)$. By enumerating $\mathcal{D}$, the reduction $\mathcal{B}$ finds such a δ and returns $s - \delta$ to the lossy soundness game. As we have $\mathsf{F}(s - \delta) - c \cdot X = R$, $\mathcal{B}$ breaks lossy soundness with the same probability as $\mathcal{A}$ breaks coset lossy soundness, and the claim follows. □

3.2 Weaker Commitments

In this section, we formally define the syntax and properties of the commitment scheme we require for our construction. Namely, we weaken the commitment definition given in [PW23]. To recall, the commitment scheme in [PW23] allows to homomorphically commit to elements in the range of a linear function family. In addition to a statistically binding mode, there is an indistinguishable way of generating commitment keys together with a weak equivocation trapdoor. This trapdoor allows to open commitments to all messages of a certain structure. In comparison to [PW23], we now also weaken the binding property of the scheme. Concretely, in the binding mode, we only require the commitment to be binding up to any shift in the image of the linear function. Except for this change, we take the definition of [PW23] verbatim.

Game Q-$\mathbf{KEYDIST}^{\mathcal{A}}_{0,\mathsf{CMT}}(\lambda)$	**Game** Q-$\mathbf{KEYDIST}^{\mathcal{A}}_{1,\mathsf{CMT}}(\lambda)$
01 $\mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda),\ x \xleftarrow{\$} \mathcal{D}$	06 $\mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda),\ x \xleftarrow{\$} \mathcal{D}$
02 **if** $(\mathsf{par}, x) \notin \mathsf{Good}$: **return** 0	07 **if** $(\mathsf{par}, x) \notin \mathsf{Good}$: **return** 0
03 **for** $i \in [Q]$: $\mathsf{ck}_i \leftarrow \mathsf{BGen}(\mathsf{par})$	08 **for** $i \in [Q]$: $\mathsf{ck}_i \xleftarrow{\$} \mathcal{K}_{\mathsf{par}}$
04 $\beta \leftarrow \mathcal{A}(\mathsf{par}, x, (\mathsf{ck}_i)_{i\in[Q]})$	09 $\beta \leftarrow \mathcal{A}(\mathsf{par}, x, (\mathsf{ck}_i)_{i\in[Q]})$
05 **return** β	10 **return** β

Figure 4: The games $\mathbf{KEYDIST}_0, \mathbf{KEYDIST}_1$ for the definition of a weakly equivocable coset commitment Scheme CMT and an adversary $\mathcal{A}$.

Definition 9 (Weakly Equivocable Coset Commitment Scheme)**.** Let $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{F})$ be a linear function family and $\mathcal{G} = \{\mathcal{G}_{\mathsf{par}}\}, \mathcal{H} = \{\mathcal{H}_{\mathsf{par}}\}$ be families of subsets of abelian groups with efficiently computable group operations $\oplus$ and $\otimes$, respectively. Let $\mathcal{K} = \{\mathcal{K}_{\mathsf{par}}\}$ be a family of sets. An $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$-weakly equivocable coset commitment scheme for LF with key space $\mathcal{K}$, randomness space $\mathcal{G}$ and commitment space $\mathcal{H}$ is a tuple of PPT algorithms $\mathsf{CMT} = (\mathsf{BGen}, \mathsf{TGen}, \mathsf{Com}, \mathsf{TCom}, \mathsf{TCol})$ with the following syntax:

- $\mathsf{BGen}(\mathsf{par}) \to \mathsf{ck}$ takes as input parameters par, and outputs a key $\mathsf{ck} \in \mathcal{K}_{\mathsf{par}}$.
- $\mathsf{TGen}(\mathsf{par}, X) \to (\mathsf{ck}, \mathsf{td})$ takes as input parameters par, and an element $X \in \mathcal{R}$, and outputs a key $\mathsf{ck} \in \mathcal{K}_{\mathsf{par}}$ and a trapdoor td.
- $\mathsf{Com}(\mathsf{ck}, R; \varphi) \to \mathsf{com}$ takes as input a key ck, an element $R \in \mathcal{R}$, and a randomness $\varphi \in \mathcal{G}_{\mathsf{par}}$, and outputs a commitment $\mathsf{com} \in \mathcal{H}_{\mathsf{par}}$.
- $\mathsf{TCom}(\mathsf{ck}, \mathsf{td}) \to (\mathsf{com}, St)$ takes as input a key ck and a trapdoor td, and outputs a commitment $\mathsf{com} \in \mathcal{H}_{\mathsf{par}}$ and a state St.
- $\mathsf{TCol}(St, c) \to (\varphi, R, s)$ takes as input a state St, and an element $c \in \mathcal{S}$, and outputs randomness $\varphi \in \mathcal{G}_{\mathsf{par}}$, and elements $R \in \mathcal{R}, s \in \mathcal{D}$.

We omit the subscript par if it is clear from the context. Further, the algorithms are required to satisfy the following properties:

- **Homomorphism.** For all $\mathsf{par} \in \mathsf{LF.Gen}(1^\lambda), \mathsf{ck} \in \mathcal{K}_{\mathsf{par}}, R_0, R_1 \in \mathcal{R}$ and $\varphi_0, \varphi_1 \in \mathcal{G}$, the following holds:
$$\mathsf{Com}(\mathsf{ck}, R_0; \varphi_0) \otimes \mathsf{Com}(\mathsf{ck}, R_1; \varphi_1) = \mathsf{Com}(\mathsf{ck}, R_0 + R_1; \varphi_0 \oplus \varphi_1).$$
- **Good Parameters.** There is a set Good, such that membership to Good can be decided in polynomial time, and
$$\Pr\left[(\mathsf{par}, x) \notin \mathsf{Good} \mid \mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda),\ x \xleftarrow{\$} \mathcal{D}\right] \leq \varepsilon_{\mathsf{g}},$$
- **Uniform Keys.** For all $(\mathsf{par}, x) \in \mathsf{Good}$, the following distributions are identical:
$$\{(\mathsf{par}, x, \mathsf{ck}) \mid \mathsf{ck} \xleftarrow{\$} \mathcal{K}_{\mathsf{par}}\} \text{ and } \{(\mathsf{par}, x, \mathsf{ck}) \mid (\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{TGen}(\mathsf{par}, \mathsf{F}(x))\}.$$

- **Weak Trapdoor Property.** For all $(\mathsf{par}, x) \in \mathsf{Good}$, and all $c \in \mathcal{S}$, the following distributions $\mathcal{T}_0$ and $\mathcal{T}_1$ have statistical distance at most ε_{t}:

$$\mathcal{T}_0 := \left\{ (\mathsf{par}, \mathsf{ck}, \mathsf{td}, x, c, \mathsf{com}, \mathsf{tr}) \;\middle|\; \begin{array}{l} (\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{TGen}(\mathsf{par}, \mathsf{F}(x)) \\ r \xleftarrow{\$} \mathcal{D},\ R := \mathsf{F}(r),\ \varphi \xleftarrow{\$} \mathcal{G}, \\ \mathsf{com} := \mathsf{Com}(\mathsf{ck}, R; \varphi), \\ s := c \cdot x + r,\ \mathsf{tr} := (\varphi, R, s) \end{array} \right\},$$

$$\mathcal{T}_1 := \left\{ (\mathsf{par}, \mathsf{ck}, \mathsf{td}, x, c, \mathsf{com}, \mathsf{tr}) \;\middle|\; \begin{array}{l} (\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{TGen}(\mathsf{par}, \mathsf{F}(x)) \\ (\mathsf{com}, St) \leftarrow \mathsf{TCom}(\mathsf{ck}, \mathsf{td}), \\ \mathsf{tr} \leftarrow \mathsf{TCol}(St, c) \end{array} \right\}.$$

- **Multi-Key Indistinguishability.** For every $Q = \mathsf{poly}(\lambda)$ and any PPT algorithm $\mathcal{A}$, the following advantage is negligible:

$$\mathsf{Adv}^{Q\text{-}\mathsf{keydist}}_{\mathcal{A},\mathsf{CMT}}(\lambda) := \left| \Pr\left[Q\text{-}\mathbf{KEYDIST}^{\mathcal{A}}_{0,\mathsf{CMT}}(\lambda) \Rightarrow 1 \right] - \Pr\left[Q\text{-}\mathbf{KEYDIST}^{\mathcal{A}}_{1,\mathsf{CMT}}(\lambda) \Rightarrow 1 \right] \right|,$$

  where games $\mathbf{KEYDIST}_0, \mathbf{KEYDIST}_1$ are defined in Figure 4.

- **Statistical Coset Binding.** There exists some (potentially unbounded) algorithm Ext, such that for every (potentially unbounded) algorithm $\mathcal{A}$ the following probability is at most ε_{b}:

$$\Pr\left[\begin{array}{rl} & \mathsf{Com}(\mathsf{ck}, R'; \varphi') = \mathsf{com} \\ \wedge & R' \notin R + \mathsf{F}(\mathcal{D}) \end{array} \;\middle|\; \begin{array}{l} \mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda), \\ \mathsf{ck} \leftarrow \mathsf{BGen}(\mathsf{par}),\ (\mathsf{com}, St) \leftarrow \mathcal{A}(\mathsf{ck}), \\ R \leftarrow \mathsf{Ext}(\mathsf{ck}, \mathsf{com}),\ (R', \varphi') \leftarrow \mathcal{A}(St) \end{array} \right].$$

4 Our Constructions

In this section, we present two constructions of efficient two-round multi-signatures that do not rely on rewinding. Both constructions rely on the building blocks introduced before.

4.1 Our Construction with Key Aggregation

In [PW23], a multi-signature scheme $\mathsf{Chopsticks\ I}$ supporting key aggregation is presented, with a security loss proportional to the number of signing queries and without rewinding. In Appendix B, we show that if we instantiate $\mathsf{Chopsticks\ I}$ with our new building blocks, we get the same properties while improving efficiency.

4.2 Our Tight Construction

Here, we present our construction of a tightly secure two-round multi-signature scheme. For that, let $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{F})$ be a linear function family. Let $\mathsf{CMT} = (\mathsf{BGen}, \mathsf{TGen}, \mathsf{Com}, \mathsf{TCom}, \mathsf{TCol})$ be an $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$-weakly equivocable coset commitment scheme for LF with key space $\mathcal{K}$, randomness space $\mathcal{G}$ and commitment space $\mathcal{H}$. Finally, let $\mathsf{H}\colon \{0,1\}^* \to \mathcal{K}$, $\mathsf{H}_b\colon \{0,1\}^* \to \{0,1\}$, and $\mathsf{H}_c\colon \{0,1\}^* \to \mathcal{S}$ be random oracles. We give a verbal description of our scheme $\mathsf{Tooth}[\mathsf{LF}, \mathsf{CMT}] = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$. In addition, we present it as pseudocode in Figure 6.

Setup and Key Generation. Our scheme makes use of public parameters $\mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda)$, which define the linear function $\mathsf{F} = \mathsf{F}(\mathsf{par}, \cdot)$. Keys are generated by sampling elements $x_0, x_1 \xleftarrow{\$} \mathcal{D}$ and a seed $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda$. Then, the keys are

$$\mathsf{sk} := (x_0, x_1, \mathsf{seed}), \quad \mathsf{pk} := (X_0, X_1) := (\mathsf{F}(x_0), \mathsf{F}(x_1)).$$

Signing Protocol. We consider the setting of a set of N signers with public keys $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$. Let $\mathsf{m} \in \{0,1\}^*$ denote the message that should be signed. In the following, we describe the signing protocol, i.e., algorithms $\mathsf{Sig}_0, \mathsf{Sig}_1, \mathsf{Sig}_2$, from the perspective of the first signer. This signer holds a secret key $\mathsf{sk}_1 = (x_{1,0}, x_{1,1}, \mathsf{seed}_1)$ for public key $\mathsf{pk}_1 = (X_{1,0}, X_{1,1})$.

1. *Commitment Phase.* First, a commitment key $\mathsf{ck} := \mathsf{H}(\langle\mathcal{P}\rangle, \mathsf{m})$ is derived from the set of public keys and the message. Further, the signer computes a bit $b_1 := \mathsf{H}_b(\mathsf{seed}_1, \langle\mathcal{P}\rangle, \mathsf{m})$. The signer computes

$$r_1 \xleftarrow{\$} \mathcal{D}, \quad R_1 := \mathsf{F}(r_1).$$

Then, the signer commits to R_1 using the commitment key ck, i.e., it computes

$$\varphi_1 \xleftarrow{\$} \mathcal{G}, \quad \mathsf{com}_1 := \mathsf{Com}(\mathsf{ck}, R_1; \varphi_1).$$

Finally, it sends $\mathsf{pm}_{1,1} := (b_1, \mathsf{com}_1)$ as its first message of the protocol to all signers.

2. *Response Phase.* Let $\mathcal{M}_1 = (\mathsf{pm}_{1,1}, \ldots, \mathsf{pm}_{1,N})$ be the list of messages output by the signers in the commitment phase. That is, the message $\mathsf{pm}_{1,i}$ is sent by signer i and has the form $\mathsf{pm}_{1,i} = (b_i, \mathsf{com}_i)$. The signer aggregates these messages by setting

$$B := b_1 \ldots b_N \in \{0,1\}^N, \quad \mathsf{com} := \bigotimes_{i\in[N]} \mathsf{com}_i.$$

Next, a signer specific challenge c_1 is derived and a response s_1 is computed. This is done via

$$c_1 := \mathsf{H}_c(\mathsf{pk}_1, \mathsf{com}, \mathsf{m}, \langle\mathcal{P}\rangle, B), \quad s_1 := c_1 \cdot x_{1,b_1} + r_1.$$

Observe that the signer uses bit b_1 to determine which part of the secret key is used. Finally, the signer sends $\mathsf{pm}_{2,1} := (s_1, \varphi_1)$ as its second message of the protocol to all signers.

3. *Aggregation Phase.* Let $\mathcal{M}_2 = (\mathsf{pm}_{2,1}, \ldots, \mathsf{pm}_{2,N})$ be the list of messages output by the signers in the response phase. That is, the message $\mathsf{pm}_{2,i}$ is sent by signer i and has the form $\mathsf{pm}_{2,i} = (s_i, \varphi_i)$. The signers aggregate the responses and commitment randomness received in the previous messages via

$$s := \sum_{i\in[N]} s_i, \;\; \varphi := \bigoplus_{i\in[N]} \varphi_i.$$

Finally, the signature is defined as $\sigma := (\mathsf{com}, \varphi, s, B)$.

Verification. Assume we have a set of public keys $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$, a message $\mathsf{m} \in \{0,1\}^*$, and a signature $\sigma := (\mathsf{com}, \varphi, s, B)$. To verify σ, write $B = b_1 \ldots b_N \in \{0,1\}^N$ and each public key pk_i as $\mathsf{pk}_i = (X_{i,0}, X_{i,1})$. Then, reconstruct the commitment key $\mathsf{ck} := \mathsf{H}(\langle\mathcal{P}\rangle, \mathsf{m})$ and the signer specific challenges $c_i := \mathsf{H}_c(\mathsf{pk}_i, \mathsf{com}, \mathsf{m}, \langle\mathcal{P}\rangle, B)$ for each $i \in [N]$. The signature is valid, i.e., the verification outputs 1, if and only if

$$\mathsf{com} = \mathsf{Com}\left(\mathsf{ck}, \mathsf{F}(s) - \sum_{i=1}^{N} c_i \cdot X_{i,b_i}; \varphi\right).$$

Lemma 2. *Let* LF *be a linear function family. Let* CMT *be an* $(\varepsilon_\mathsf{b}, \varepsilon_\mathsf{g}, \varepsilon_\mathsf{t})$*-weakly equivocable coset commitment scheme for* LF*. Then* $\mathsf{Tooth}[\mathsf{LF}, \mathsf{CMT}]$ *is complete.*

Proof. To show completeness of $\mathsf{Tooth}[\mathsf{LF}, \mathsf{CMT}]$, consider N users and let $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ be the set of their public keys, where $\mathsf{pk}_i = (X_{i,0}, X_{i,1}) = (\mathsf{F}(x_{i,0}), \mathsf{F}(x_{i,1}))$ for each $i \in [N]$. Let $\mathsf{m} \in \{0,1\}^*$ be a message, and let $\sigma = (\mathsf{com}, \varphi, s, B)$ for $B = b_1 \ldots b_N \in \{0,1\}^N$ be a signature computed honestly in the signing protocol. We have to show that verification outputs 1 on input $\mathcal{P}, \mathsf{m}, \sigma$. For that, let $\mathsf{ck} = \mathsf{H}(\langle\mathcal{P}\rangle, \mathsf{m})$ and $c_i = \mathsf{H}_c(\mathsf{pk}_i, \mathsf{com}, \mathsf{m}, \langle\mathcal{P}\rangle, B)$ for each $i \in [N]$ be as in the verification algorithm. We have to show that

$$\mathsf{com} = \mathsf{Com}\left(\mathsf{ck}, \mathsf{F}(s) - \sum_{i=1}^{N} c_i \cdot X_{i,b_i}; \varphi\right).$$

Using definition of s and the X_{i,b_i}, and linearity of F, we get

$$\mathsf{F}(s) - \sum_{i=1}^{N} c_i \cdot X_{i,b_i} = \mathsf{F}\left(\sum_{i=1}^{N} s_i\right) - \sum_{i=1}^{N} c_i \cdot \mathsf{F}(x_{i,b_i}) = \sum_{i=1}^{N} \mathsf{F}(s_i - c_i \cdot x_{i,b_i}).$$

Now, we use the definition of the s_i as $s_i = c_i \cdot x_{i,b_i} + r_i$, where $r_i \in \mathcal{D}$ is the element that the ith signer samples in the first step, and get

$$\sum_{i=1}^{N} \mathsf{F}(s_i - c_i \cdot x_{i,b_i}) = \sum_{i=1}^{N} \mathsf{F}(r_i) = \sum_{i=1}^{N} R_i,$$

where $R_i = \mathsf{F}(r_i)$ is the element to which each signer commits in the first step. In combination, we get

$$\mathsf{Com}\left(\mathsf{ck}, \mathsf{F}(s) - \sum_{i=1}^{N} c_i \cdot X_{i,b_i}; \varphi\right) = \mathsf{Com}\left(\mathsf{ck}, \sum_{i=1}^{N} R_i; \bigoplus_{i\in[N]} \varphi_i\right),$$

where we used the definition of φ. We can now apply the homomorphism property of the commitment and get

$$\mathsf{Com}\left(\mathsf{ck}, \sum_{i=1}^{N} R_i; \bigoplus_{i\in[N]} \varphi_i\right) = \bigotimes_{i=1}^{N} \mathsf{Com}\left(\mathsf{ck}, R_i; \varphi_i\right) = \bigotimes_{i=1}^{N} \mathsf{com}_i = \mathsf{com},$$

where the com_i are what each signer sends in the first message. This proves the claim. □

Theorem 1. *Let* LF *be a linear function family that satisfies key indistinguishability and* ε_{l}*-coset lossy soundness. Let* CMT *be an* $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$*-weakly equivocable coset commitment scheme for* LF*. Further, let* $\mathsf{H}\colon \{0,1\}^* \to \mathcal{K}, \mathsf{H}_b\colon \{0,1\}^* \to \{0,1\}, \mathsf{H}_c\colon \{0,1\}^* \to \mathcal{S}$ *be random oracles. Then* $\mathsf{Tooth}[\mathsf{LF}, \mathsf{CMT}]$ *is* $\mathsf{MS\text{-}EUF\text{-}CMA}$ *secure.*

Concretely, for any PPT algorithm $\mathcal{A}$ *that makes at most* $Q_{\mathsf{H}}, Q_{\mathsf{H}_b}, Q_{\mathsf{H}_c}, Q_S$ *queries to oracles* $\mathsf{H}, \mathsf{H}_b, \mathsf{H}_c, \textsc{Sig}_0$*, respectively, there are PPT algorithms* $\mathcal{B}, \mathcal{B}'$ *with* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}), \mathbf{T}(\mathcal{B}') \approx \mathbf{T}(\mathcal{A})$ *and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{MS\text{-}EUF\text{-}CMA}}_{\mathcal{A},\mathsf{Tooth}[\mathsf{LF},\mathsf{CMT}]}(\lambda) \leq \frac{Q_{\mathsf{H}_b}}{2^{\lambda}} + 8\varepsilon_{\mathsf{g}} + 4Q_S\varepsilon_{\mathsf{t}} + 4Q_{\mathsf{H}}Q_{\mathsf{H}_c}\varepsilon_{\mathsf{b}} + 4Q_{\mathsf{H}_c}\varepsilon_{\mathsf{l}} \\ + 4 \cdot \mathsf{Adv}^{Q_{\mathsf{H}}\text{-}\mathsf{keydist}}_{\mathcal{B},\mathsf{CMT}}(\lambda) + 4 \cdot \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}',\mathsf{LF}}(\lambda).\end{aligned}$$

Proof. Let $\mathcal{A}$ be an adversary against the security of $\mathsf{Tooth}[\mathsf{LF}, \mathsf{CMT}]$. To prove the statement, we give a sequence of games $\mathbf{G}_0, \ldots, \mathbf{G}_8$. We present the games formally in Figures 7 to 9, and we verbally describe and analyze them here.

Game $\mathbf{G}_0$: This is defined to be the original security game $\mathbf{MS\text{-}EUF\text{-}CMA}^{\mathcal{A}}_{\mathsf{Tooth}[\mathsf{LF},\mathsf{CMT}]}$, but we omit the oracle $\textsc{Sig}_2$ from the game. Observe that this is without loss of generality for the scheme at hand, as this oracle can be run publicly based on the outputs of the other oracles and does not make use of any secret state or key. More concretely, for any adversary $\mathcal{A}$ that calls this oracle, we can build a wrapper adversary that internally simulates the game including oracle $\textsc{Sig}_2$ for $\mathcal{A}$ and forwards everything else to $\mathbf{G}_0$. This wrapper adversary has the same advantage and running time as $\mathcal{A}$. We now recall the game to fix notation. First, system parameters $\mathsf{par} \leftarrow \mathsf{LF.Gen}(1^{\lambda})$ are generated. In addition, the secret and public key of an honest user are generated. Namely, the game samples $\mathsf{seed}_1 \xleftarrow{\$} \{0,1\}^{\lambda}$ and $x_{1,0}, x_{1,1} \xleftarrow{\$} \mathcal{D}$ and sets $X_{1,0} := \mathsf{F}(x_{1,0})$ and $X_{1,1} := \mathsf{F}(x_{1,1})$. It sets $\mathsf{pk}^* := (X_{1,0}, X_{1,1})$ and runs $\mathcal{A}$ on input $\mathsf{par}, \mathsf{pk}^*$, with access to the following oracles

- Signing oracles $\textsc{Sig}_0$ and $\textsc{Sig}_1$: The signing oracles simulate an honest signer in a signing interaction. More precisely, if $\mathcal{A}$ queries $\textsc{Sig}_0(\mathcal{P}, \mathsf{m})$, a new signing interaction for message m with respect to $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ is started, where we assume that $\mathsf{pk}_1 = \mathsf{pk}^*$. For that, first $(\mathcal{P}, \mathsf{m})$ is added to list $\mathsf{Queried}$. Then, the game runs algorithm Sig_0 in the natural way and outputs the result to the adversary. Similarly, when $\mathcal{A}$ calls $\textsc{Sig}_1$, algorithm Sig_1 is run.
- Random oracles H and H_c: The game simulates random oracles H and H_c for $\mathcal{A}$ by standard lazy sampling. For that, it holds maps h and h_c which map the inputs to their outputs. For example, if $\mathcal{A}$ queries $\mathsf{H}(x)$, the game checks if $h[x]$ is defined. If it is not yet defined, it is sampled at random from the output domain of H, i.e., from $\mathcal{K}$. Then, the game returns $h[x]$.

- Random oracle H_b: For H_b, we additionally introduce a level of indirection. This will allow us to distinguish queries to H_b that the game itself issues from the queries that $\mathcal{A}$ issues directly. Concretely, when H_b is queried, the game forwards the query to a random oracle $\bar{\mathsf{H}}_b$ with the same interface. Oracle $\bar{\mathsf{H}}_b$ is simulated using a map $\bar{h}_b$ via lazy sampling. We emphasize that this oracle $\bar{\mathsf{H}}_b$ is not provided to $\mathcal{A}$. Further, the convention for all games will be that the game itself only queries $\bar{\mathsf{H}}_b$ and not H_b, for example in oracle $\textsc{Sig}_0$.

Finally, $\mathcal{A}$ outputs a forgery $(\mathcal{P}^*, \mathsf{m}^*, \sigma^*)$. Write $\mathcal{P}^*$ as $\mathcal{P}^* = \{\mathsf{pk}_1, \dots, \mathsf{pk}_N\}$ and $\sigma^* = (\mathsf{com}^*, \varphi^*, s^*, B^*)$. Further, write $B^* = b_1^* \dots b_N^* \in \{0,1\}^N$. Then, the game outputs 0 if $\mathsf{pk}^* \notin \mathcal{P}^*$ or $(\mathcal{P}^*, \mathsf{m}^*) \in \mathsf{Queried}$. Otherwise, we assume that $\mathsf{pk}^* = \mathsf{pk}_1$, and the game outputs 1 if and only if $\mathsf{Ver}(\mathcal{P}^*, \mathsf{m}^*, \sigma^*) = 1$. By definition, we have

$$\mathsf{Adv}^{\mathsf{MS\text{-}EUF\text{-}CMA}}_{\mathcal{A},\mathsf{Tooth}[\mathsf{LF},\mathsf{CMT}]}(\lambda) = \Pr[\mathbf{G}_0 \Rightarrow 1].$$

Before we continue, we give an overview of the remaining games and our strategy. In our first step (games $\mathbf{G}_1$ and $\mathbf{G}_2$), we will ensure that for the forgery it holds that $b_1^* = 1 - b^*$ and $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*) = b^*$, for a random bit b^*. Once this is established, we change how we simulate the signing oracles (games $\mathbf{G}_3$ to $\mathbf{G}_6$). Namely, in the case $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P} \rangle, \mathsf{m}) = b^*$, we embed a binding commitment key and simulate signing for $(\mathcal{P}, \mathsf{m})$ honestly, whereas for the other case, we embed a commitment key with a trapdoor and simulate signing by using the trapdoor. The result is that we no longer need $x_{1,1-b^*}$. Now, we switch $X_{1,1-b^*}$ to lossy mode and use the binding property to reduce to lossy soundness (games $\mathbf{G}_7$ and $\mathbf{G}_8$). This works, because the forgery is with respect to a lossy key and a binding commitment key.

Game $\mathbf{G}_1$: This game is the same as $\mathbf{G}_0$, but we introduce a bad event on which the game aborts. Namely, the game sets $\mathsf{bad} := 1$ if $\mathcal{A}$ queries $\mathsf{H}_b(\mathsf{seed}_1, x)$ for any $x \in \{0,1\}^*$. Once $\mathcal{A}$ terminates, the game outputs 0 if $\mathsf{bad} = 1$. Otherwise, it behaves as $\mathbf{G}_0$. It is clear that games $\mathbf{G}_0$ and $\mathbf{G}_1$ only differ if $\mathcal{A}$ makes such a query. Further, the only information about seed_1 that $\mathcal{A}$ gets are the values of $\mathsf{H}_b(\mathsf{seed}_1, \cdot)$. As seed_1 is sampled uniformly at random from $\{0,1\}^\lambda$, we can bound the probability that a fixed query of $\mathcal{A}$ has the form $\mathsf{H}_b(\mathsf{seed}_1, x)$ by $1/2^\lambda$. With a union bound over the queries of $\mathcal{A}$ we obtain

$$|\Pr[\mathbf{G}_0 \Rightarrow 1] - \Pr[\mathbf{G}_1 \Rightarrow 1]| \leq \frac{Q_{\mathsf{H}_b}}{2^\lambda}.$$

Game $\mathbf{G}_2$: In this game, we introduce a random bit $b^* \xleftarrow{\$} \{0,1\}$ that is sampled at the beginning of the game. Further, we change the winning condition as follows. When $\mathcal{A}$ outputs the forgery, the game outputs 0, if $b_1^* = b^*$ or $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*) = 1 - b^*$. Otherwise, it continues as $\mathbf{G}_1$ does. In other words, game $\mathbf{G}_2$ outputs 1 if $\mathbf{G}_1$ outputs 1 and the following event occurs:

- Event $\mathsf{RightBits}$: This event occurs, if for $\mathcal{A}$'s final output $(\mathcal{P}^*, \mathsf{m}^*, \sigma^*)$ with $\mathcal{P}^* = \{\mathsf{pk}_1 = \mathsf{pk}^*, \dots, \mathsf{pk}_N\}, \sigma^* = (\mathsf{com}^*, \varphi^*, s^*, B^*)$, and $B^* = b_1^* \dots b_N^* \in \{0,1\}^N$, it holds that $b_1^* = 1 - b^*$ and $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*) = b^*$.

If we condition on $\mathbf{G}_1 \Rightarrow 1$, then we claim that b^* and $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*)$ are uniformly random and independent, and independent of $\mathcal{A}$'s view. In particular, they are independent of b_1^*. This is because bit b^* is hidden from $\mathcal{A}$ by construction, and $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*)$ is hidden from $\mathcal{A}$ due to $(\mathcal{P}^*, \mathsf{m}^*) \notin \mathsf{Queried}$ and the change introduced in $\mathbf{G}_1$. Therefore, we have

$$\Pr[\mathsf{RightBits} \mid \mathbf{G}_1 \Rightarrow 1] = \Pr_{b, b^* \xleftarrow{\$} \{0,1\}}[b_1^* = 1 - b^* \wedge b = b^*] = \frac{1}{4}.$$

With this, we obtain

$$\begin{aligned}\Pr[\mathbf{G}_2 \Rightarrow 1] &= \Pr[\mathsf{RightBits} \wedge \mathbf{G}_1 \Rightarrow 1] \\ &= \Pr[\mathsf{RightBits} \mid \mathbf{G}_1 \Rightarrow 1] \cdot \Pr[\mathbf{G}_1 \Rightarrow 1] = \frac{1}{4} \cdot \Pr[\mathbf{G}_1 \Rightarrow 1].\end{aligned}$$

Game $\mathbf{G}_3$: This game is the same as $\mathbf{G}_2$, but we add another abort. Namely, once the game sampled par and $x_{1,0}, x_{1,1}$ at the beginning of the game, it returns 0 and terminates if $(\mathsf{par}, x_{1,1-b^*}) \notin \mathsf{Good}$, where Good is as in the definition of the weakly equivocable coset commitment scheme. Otherwise, it continues as in $\mathbf{G}_2$ does. By the good parameters property of CMT, we have

$$|\Pr[\mathbf{G}_2 \Rightarrow 1] - \Pr[\mathbf{G}_3 \Rightarrow 1]| \leq \Pr[(\mathsf{par}, x_{1,1-b^*}) \notin \mathsf{Good}] \leq \varepsilon_{\mathsf{g}}.$$

Game $\mathbf{G}_4$: In this game, we change how random oracle H is simulated. Recall that until now, when H is queried on an input $(\langle\mathcal{P}\rangle, \mathsf{m})$ and the output of H is not yet defined, it samples a random commitment key $\mathsf{ck} \xleftarrow{\$} \mathcal{K}$ uniformly at random and defines the output to be this key. From now on, we sample ck differently, distinguishing two cases depending on the bit $b := \bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle\mathcal{P}\rangle, \mathsf{m})$ and the bit b^*. Namely, if $b = 1 - b^*$, then ck is sampled in hiding mode with a trapdoor, i.e., $(\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{TGen}(\mathsf{par}, X_{1,1-b^*})$. Further, the trapdoor td is stored in a map tr by setting $tr[\langle\mathcal{P}\rangle, \mathsf{m}] := \mathsf{td}$. On the other hand, if $b = b^*$, then ck is sampled in binding mode, i.e., $\mathsf{ck} \leftarrow \mathsf{BGen}(\mathsf{par})$. We now show indistinguishability of $\mathbf{G}_3$ and $\mathbf{G}_4$. First, note that keys sampled in the first case are distributed identically in $\mathbf{G}_3$ and $\mathbf{G}_4$. This follows from the uniform keys property of CMT, which we can apply due to the previous change that ensures that $(\mathsf{par}, x_{1,1-b^*}) \in \mathsf{Good}$. Second, keys sampled in the second case are indistinguishable by the multi-key indistinguishability property of CMT. More precisely, there is a reduction $\mathcal{B}$ that gets as input $\mathsf{par}, x_{1,1-b^*}$, and commitment keys $\mathsf{ck}_1, \ldots, \mathsf{ck}_{Q_\mathsf{H}}$. It then simulates game $\mathbf{G}_4$ for $\mathcal{A}$, but embedding the commitment keys ck_i whenever random oracle H needs to be simulated and $b = b^*$ as above. In the end, $\mathcal{B}$ outputs whatever the game outputs. Clearly, $\mathcal{B}$'s running time is determined by the running time of $\mathcal{A}$ and it perfectly simulates $\mathbf{G}_4$ if the keys $\mathsf{ck}_1, \ldots, \mathsf{ck}_{Q_\mathsf{H}}$ are generated via algorithm BGen. Otherwise, if the keys are sampled uniformly at random, it perfectly simulates $\mathbf{G}_3$ for $\mathcal{A}$. For this, it was important that we introduced the indirection via oracle $\bar{\mathsf{H}}_b$ as otherwise the simulation would not be perfect. Concretely, if $\mathcal{B}$ itself had queried H_b instead of $\bar{\mathsf{H}}_b$, then the game would always have output 0, see the change in $\mathbf{G}_2$. We have

$$|\Pr[\mathbf{G}_3 \Rightarrow 1] - \Pr[\mathbf{G}_4 \Rightarrow 1]| \leq \mathsf{Adv}^{Q_\mathsf{H}\text{-}\mathsf{keydist}}_{\mathcal{B},\mathsf{CMT}}(\lambda).$$

Game $\mathbf{G}_5$: In this game, we change the signing oracle. The result will be that we can simulate the signing oracle without $x_{1,1-b^*}$, but using trapdoors for commitment keys instead. First, we explain how we change oracle $\textsc{Sig}_0$, which runs Sig_0 in previous games. Recall from the definition of Sig_0, that this means that the oracle on input $\mathcal{P}$ and m samples $r_1 \xleftarrow{\$} \mathcal{D}$, defines $R_1 := \mathsf{F}(r_1)$, computes a bit $b_1 := \mathsf{H}_b(\mathsf{seed}_1, \langle\mathcal{P}\rangle, \mathsf{m})$ and a commitment key $\mathsf{ck} := \mathsf{H}(\langle\mathcal{P}\rangle, \mathsf{m})$, and commits to R_1 by sampling $\varphi_1 \xleftarrow{\$} \mathcal{G}$ and setting $\mathsf{com}_1 := \mathsf{Com}(\mathsf{ck}, R_1; \varphi_1)$. Now, if $b_1 = b^*$, we don't change anything and $\mathbf{G}_5$ behaves as previous games do. However, if $b_1 = 1 - b^*$, the game computes com_1 differently. It computes it as $(\mathsf{com}_1, St_1) \leftarrow \mathsf{TCom}(\mathsf{ck}, tr[\langle\mathcal{P}\rangle, \mathsf{m}])$. Here, recall that if $b_1 = 1 - b^*$, then ck has been generated with a trapdoor that is stored in tr, see $\mathbf{G}_4$. Next, we explain how we change oracle $\textsc{Sig}_1$, which runs Sig_1 in previous games. To recall, this means that first, a challenge c_1 is computed using the random oracle H_c and all messages of the first round. Then, a response $s_1 := c_1 \cdot x_{1,b_1} + r_1$ is computed, and s_1 and φ_1 is returned to $\mathcal{A}$. Again, we only change the case where $b_1 = 1 - b^*$. Namely, in this case, the game runs $(\varphi_1, R_1, s_1) \leftarrow \mathsf{TCol}(St_1, c_1)$ to compute s_1 and φ_1 instead. Due to the change introduced in $\mathbf{G}_3$, we know that $(\mathsf{par}, x_{1,1-b^*}) \in \mathsf{Good}$, and thus we can apply the weak trapdoor property of CMT for every signing query. We get

$$|\Pr[\mathbf{G}_4 \Rightarrow 1] - \Pr[\mathbf{G}_5 \Rightarrow 1]| \leq Q_S \varepsilon_\mathsf{t}.$$

Game $\mathbf{G}_6$: In this game, we undo the change from $\mathbf{G}_3$, namely, we no longer require that $(\mathsf{par}, x_{1,1-b^*}) \in \mathsf{Good}$. With a similar argument as in $\mathbf{G}_3$, we get

$$|\Pr[\mathbf{G}_5 \Rightarrow 1] - \Pr[\mathbf{G}_6 \Rightarrow 1]| \leq \varepsilon_\mathsf{g}.$$

Game $\mathbf{G}_7$: In this game, we change how $X_{1,1-b^*}$ is generated. Recall that until now, it is generated by sampling $x_{1,1-b^*} \xleftarrow{\$} \mathcal{D}$ and setting $X_{1,1-b^*} := \mathsf{F}(x_{1,1-b^*})$. From now on, we sample it in lossy mode, i.e., as $X_{1,1-b^*} \xleftarrow{\$} \mathcal{R}$. Observe that $x_{1,1-b^*}$ is used nowhere else during the game, due to our previous changes. Therefore, we can easily bound the distinguishing advantage between $\mathbf{G}_6$ and $\mathbf{G}_7$ by a reduction $\mathcal{B}'$ that runs in the key indistinguishability game of LF and embeds its input in $X_{1,1-b^*}$. We have

$$|\Pr[\mathbf{G}_6 \Rightarrow 1] - \Pr[\mathbf{G}_7 \Rightarrow 1]| \leq \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}',\mathsf{LF}}(\lambda).$$

Game $\mathbf{G}_8$: In game $\mathbf{G}_8$, we make use of the statistical coset binding property of CMT. Concretely, we change oracle H_c and the winning condition. Recall that until now, a query $\mathsf{H}_c(\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle\mathcal{P}\rangle, B)$ is answered in the standard way using lazy sampling. In game $\mathbf{G}_8$, this is still the case, but additionally the extractor Ext for the statistical coset binding property of CMT is run in certain cases. Namely, write $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ and $B = b_1 \ldots b_N$. Further, set $b := \bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle\mathcal{P}\rangle, \mathsf{m})$. If pk^* is part of $\mathcal{P}$, i.e., $\mathsf{pk}^* = \mathsf{pk}_1$, and $b = b^*$, then we know that the commitment key $\mathsf{ck} := \mathsf{H}(\langle\mathcal{P}\rangle, \mathsf{m})$ is generated in binding mode by algorithm BGen. This is due to the change in $\mathbf{G}_4$. Now, game $\mathbf{G}_8$ runs $R \leftarrow \mathsf{Ext}(\mathsf{H}(\langle\mathcal{P}\rangle, \mathsf{m}), \mathsf{com})$

and stores R in a map $r[\cdot]$ as $r[\mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] := R$. Other than that, the oracle H_c does not change. Next, we describe how the winning condition is changed. For that, assume that $\mathcal{A}$ outputs a forgery $\mathcal{A}$ outputs a forgery $(\mathcal{P}^*, \mathsf{m}^*, \sigma^*)$ with $\mathcal{P}^* = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$, $\sigma^* = (\mathsf{com}^*, \varphi^*, s^*, B^*)$, and $B^* = b_1^* \ldots b_N^* \in \{0,1\}^N$. Assume that game $\mathbf{G}_7$ does not return 0. Especially, we have $\mathsf{pk}_1 = \mathsf{pk}^*$ and $(\mathcal{P}^*, \mathsf{m}^*) \notin \mathsf{Queried}$, and $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*) = b^*$ (see $\mathbf{G}_2$). Further, the game parses $\mathsf{pk}_i = (X_{i,0}, X_{i,1})$ for every key pk_i in $\mathcal{P}^*$ and defines challenges $c_i^* := \mathsf{H}_c(\mathsf{pk}_i, \mathsf{com}_0^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*)$ for all $i \in [N]$ as the verification algorithm does. In particular, now we know, due to $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*) = b^*$, that $r[\mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*]$ is defined. Next, the game defines $R^* := \mathsf{F}(s^*) - \sum_{i=1}^N c_i^* \cdot X_{i,b_i^*}$ as the verification algorithm does. The game outputs 0 if $R^* \notin r[\mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*] + \mathsf{F}(\mathcal{D})$. Otherwise, it behaves as $\mathbf{G}_7$ does. Note that if $\mathbf{G}_7$ outputs 1, but $\mathbf{G}_8$ does not, then we know that $\mathsf{com}^* = \mathsf{Com}(\mathsf{ck}, R^*; \varphi^*)$, where $\mathsf{ck} := \mathsf{H}(\langle \mathcal{P} \rangle^*, \mathsf{m}^*)$. In other words, $\mathbf{G}_7$ and $\mathbf{G}_8$ only differ, if for the forgery, the value $r[\mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*]$ that the extractor extracted from commitment com^* is in a different coset than the value to which $\mathcal{A}$ successfully opens com^* in its forgery. We can easily bound the probability of this using the statistical coset binding property of CMT. For that, we sketch an (unbounded) reduction that gets as input the parameters par of the linear function, and a commitment key $\mathsf{ck} \leftarrow \mathsf{BGen}(\mathsf{par})$. Then, it first samples indices $i_{\mathsf{H}} \xleftarrow{\$} [Q_{\mathsf{H}}]$ and $i_{\mathsf{H}_c} \xleftarrow{\$} [Q_{\mathsf{H}_c}]$ uniformly at random, and then simulates the game $\mathbf{G}_8$ honestly for $\mathcal{A}$, except the i_{H}th query to H and the i_{H_c}th query to H_c. In the i_{H}th query to H, if it has to sample a binding key, it embeds ck. Otherwise, it aborts. In the i_{H_c}th query to H_c, if it had to run Ext, it instead outputs the commitment com and its state to the statistical coset binding game. Otherwise, it aborts. Finally, when $\mathcal{A}$ outputs its forgery, and the i_{H}th query to H and the i_{H_c}th query to H_c are the queries of interest, and $R^* \notin r[\mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*] + \mathsf{F}(\mathcal{D})$, the reduction outputs R^* and φ^*, thereby winning the statistical coset binding game. It is easy to see that this shows

$$|\Pr[\mathbf{G}_7 \Rightarrow 1] - \Pr[\mathbf{G}_8 \Rightarrow 1]| \leq Q_{\mathsf{H}} Q_{\mathsf{H}_c} \varepsilon_{\mathsf{b}}.$$

To finish the proof, we bound the probability that $\mathbf{G}_8$ outputs 1 using coset lossy soundness of LF. For that consider the following unbounded reduction:

- The reduction gets as input parameters par and an element $X \in \mathcal{R}$.
- It picks a random index $\hat{i} \xleftarrow{\$} [Q_{\mathsf{H}_c}]$. It then simulates game $\mathbf{G}_8$ for $\mathcal{A}$ until $\mathcal{A}$ outputs its forgery, using parameters par and defining $X_{1,1-b^*} := X$ instead of picking it randomly from $\mathcal{R}$. Further, the reduction handles the $\hat{i}$th query to H_c differently.
- Let $\mathsf{H}_c(\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B)$ be the $\hat{i}$th query to H_c. If the hash value for this query is already defined, the reduction continues as $\mathbf{G}_8$ would do. Otherwise, let $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$, $B = b_1 \ldots b_N$, and $b := \bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P} \rangle, \mathsf{m})$. The reduction also continues as $\mathbf{G}_8$ would do if pk^* is not in $\mathcal{P}$. Otherwise, assume that $\mathsf{pk}^* = \mathsf{pk}_1$ as usual. If $\mathsf{pk} \neq \mathsf{pk}^*$ or $b \neq b^*$, the reduction also continues as $\mathbf{G}_8$ would do. In other words, the reduction only differs in the case in which $\mathbf{G}_8$ would run the extractor Ext. In this case, the reduction runs Ext as $\mathbf{G}_8$ would do, i.e., it runs $\hat{R} \leftarrow \mathsf{Ext}(\mathsf{H}(\langle \mathcal{P} \rangle, \mathsf{m}), \mathsf{com})$ and sets $r[\mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] := \hat{R}$. In addition, the reduction sets $c_i := \mathsf{H}_c(\mathsf{pk}_i, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B)$ for each $i \in [N] \setminus \{1\}$, and computes

  $$R := \hat{R} + \sum_{i=2}^{N} c_i \cdot X_{i,b_i}.$$

  Then, the reduction outputs R to the coset lossy soundness game, and in return it receives a challenge $c \in \mathcal{S}$. Finally, the reduction programs $h_c[\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] := c$ and returns this hash value.
- When $\mathcal{A}$ outputs its forgery $(\mathcal{P}^*, \mathsf{m}^*, \sigma^*)$, the reduction does all the verification checks as in $\mathbf{G}_8$. Assuming all of these checks pass, write $\mathcal{P}^* = \{\mathsf{pk}_1 = \mathsf{pk}^*, \ldots, \mathsf{pk}_N\}$, $\sigma^* = (\mathsf{com}^*, \varphi^*, s^*, B^*)$, and $B^* = b_1^* \ldots b_N^* \in \{0,1\}^N$. Additionally, the reduction aborts if the hash value $\mathsf{H}_c(\mathsf{pk}_1, \mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*)$ has not been defined during the $\hat{i}$th query to H_c. Otherwise, the reduction returns $s := s^*$ to the coset lossy soundness game.

One can easily see that the view of $\mathcal{A}$ is independent of the index $\hat{i}$ until a potential abort, and that, assuming the reduction does not abort, the simulation of $\mathbf{G}_8$ is perfect. Now, we want to argue that the reduction breaks coset lossy soundness if $\mathbf{G}_8$ outputs 1, and the index $\hat{i}$ is guessed correctly. Once this is shown, we can conclude with

$$\Pr[\mathbf{G}_8 \Rightarrow 1] \leq Q_{\mathsf{H}_c} \varepsilon_{\mathsf{l}}.$$

To show this claim, we assume that $\mathbf{G}_8$ outputs 1 and the index $\hat{i}$ is guessed correctly. Now, it follows from the condition $\bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*) = b^*$ introduced in $\mathbf{G}_2$ that the reduction output R as above to the coset lossy soundness game, received c, and programmed $\mathsf{H}_c(\mathsf{pk}_1, \mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*)$ to be c. It remains to argue that $\mathsf{F}(s) - c \cdot X \in R + \mathsf{F}(\mathcal{D})$. For that, first recall that the change introduced in $\mathbf{G}_8$ ensures that

$$\mathsf{F}(s^*) - \sum_{i=1}^{N} c_i^* \cdot X_{i,b_i^*} \in r[\mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*] + \mathsf{F}(\mathcal{D}).$$

Using the assumption that the index $\hat{i}$ is guessed correctly, this implies

$$\mathsf{F}(s^*) - c \cdot X_{1,b_1^*} - \sum_{i=2}^{N} c_i \cdot X_{i,b_i^*} \in \hat{R} + \mathsf{F}(\mathcal{D}).$$

Now, we rearrange terms and use the condition $b_1^* = 1 - b^*$ introduced in $\mathbf{G}_2$, and get

$$\mathsf{F}(s^*) - c \cdot X_{1,1-b^*} \in \hat{R} + \sum_{i=2}^{N} c_i \cdot X_{i,b_i^*} + \mathsf{F}(\mathcal{D}).$$

If we recall the definition of $s^* = s$, $X_{1,1-b^*} = X$, and the definition of R, then this is exactly the statement we want to show. Concluded. □

5 Our Instantiations

In this section, we instantiate the building blocks introduced in Section 3. We present a linear function family and a commitment scheme, both based on DDH.

5.1 Linear Function Family

We use the same linear function family as in [PW23], which is a linear function family $\mathsf{LF}_{\mathsf{DDH}} = (\mathsf{Gen}, \mathsf{F})$ based on the DDH assumption. For that, we assume an algorithm GGen, that outputs the description of a prime order group $\mathbb{G}$ of order p with generator g on input 1^λ. Algorithm Gen runs GGen, samples $h \xleftarrow{\$} \mathbb{G}$, and outputs parameters $\mathsf{par} := (g, h) \in \mathbb{G}^2$. The description of $\mathbb{G}$ is also contained in par and left implicit for the sake of a concise presentation. These parameters define the set of scalars, domain, range, and the function $\mathsf{F}(\mathsf{par}, \cdot)$, which are as follows:

$$\mathcal{S} := \mathbb{Z}_p, \quad \mathcal{D} := \mathbb{Z}_p, \quad \mathcal{R} := \mathbb{G} \times \mathbb{G}, \quad \mathsf{F}(\mathsf{par}, x) := (g^x, h^x).$$

One can easily verify that this is a linear function family. Further, it is shown in [PW23] that $\mathsf{LF}_{\mathsf{DDH}}$ satisfies key indistinguishability, lossy soundness, and aggregation lossy soundness. Using Lemmata 1 and 8, we conclude that $\mathsf{LF}_{\mathsf{DDH}}$ satisfies coset lossy soundness and coset aggregation lossy soundness. The following lemma summarizes this.

Lemma 3. *Assuming that the* DDH *assumption holds relative to* GGen*, the linear function family* $\mathsf{LF}_{\mathsf{DDH}}$ *satisfies key indistinguishability. Concretely, for any PPT algorithm* $\mathcal{A}$ *there is a PPT algorithm* $\mathcal{B}$ *with* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and*

$$\mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{A},\mathsf{LF}_{\mathsf{DDH}}}(\lambda) \leq \mathsf{Adv}^{\mathsf{DDH}}_{\mathcal{B},\mathsf{GGen}}(\lambda).$$

Further, the linear function family $\mathsf{LF}_{\mathsf{DDH}}$ *satisfies* ε_{l}*-coset lossy soundness and* $\varepsilon_{\mathsf{al}}$*-coset aggregation lossy soundness for*

$$\varepsilon_{\mathsf{l}} \leq 3/p, \quad \varepsilon_{\mathsf{al}} \leq 4/p.$$

5.2 Commitment Scheme

In this section, we present our instantiation of the weakly equivocable coset commitment scheme for the linear function family $\mathsf{LF}_{\mathsf{DDH}}$ introduced before. Our commitment scheme shares similarities with the commitment scheme from [PW23], which uses a 3×3 matrix of group elements as a commitment key. Our crucial observation is that if we replace this 3×3 structure with a more efficient 2×2 structure, we obtain a scheme that is still binding on cosets. We now describe our commitment scheme $\mathsf{CMT}_{\mathsf{DDH}} = (\mathsf{BGen}, \mathsf{TGen}, \mathsf{Com}, \mathsf{TCom}, \mathsf{TCol})$ for $\mathsf{LF}_{\mathsf{DDH}}$. Assume parameters of $\mathsf{LF}_{\mathsf{DDH}}$ are given, specifying a group $\mathbb{G}$. Then, the commitment scheme has key space $\mathcal{K} := \mathbb{G}^{2\times 2}$, message space $\mathcal{D} = \mathbb{G} \times \mathbb{G}$, randomness space $\mathcal{G} = \mathbb{Z}_p^2$, and commitment space $\mathcal{H} = \mathbb{G}^2$. The spaces $\mathcal{D}, \mathcal{G}$, and $\mathcal{H}$ are associated with the natural componentwise group operations. Next, we describe the algorithms of the commitment scheme verbally.

- $\mathsf{BGen}(\mathsf{par}) \to \mathsf{ck}$: Parse $\mathsf{par} = (g, h)$. Sample $a, b \xleftarrow{\$} \mathbb{Z}_p$ and set

$$\mathsf{ck} := \mathbf{A} := \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} := \begin{pmatrix} g^a & g^b \\ h^a & h^b \end{pmatrix} \in \mathbb{G}^{2\times 2}.$$

- $\mathsf{TGen}(\mathsf{par}, X = (X_1, X_2)) \to (\mathsf{ck}, \mathsf{td})$: Sample exponents $d_{i,j} \xleftarrow{\$} \mathbb{Z}_p$ for all $(i, j) \in [2] \times [2]$. Set

$$\mathsf{ck} := \mathbf{A} := \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} := \begin{pmatrix} X_1^{d_{1,1}} & X_1^{d_{1,2}} \\ X_2^{d_{2,1}} & X_2^{d_{2,2}} \end{pmatrix} \in \mathbb{G}^{2\times 2}.$$

  Further, set $\mathsf{td} := (\mathbf{D}, X_1, X_2)$ for

$$\mathbf{D} := \begin{pmatrix} d_{1,1} & d_{1,2} \\ d_{2,1} & d_{2,2} \end{pmatrix} \in \mathbb{Z}_p^{2\times 2}.$$

- $\mathsf{Com}(\mathsf{ck}, R = (R_1, R_2); \varphi) \to \mathsf{com}$: Let $\varphi = (\alpha, \beta) \in \mathbb{Z}_p^2$. Compute $\mathsf{com} := (C_1, C_2)$ for

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} := \begin{pmatrix} R_1 \cdot & A_{1,1}^{\alpha} \cdot A_{1,2}^{\beta} \\ R_2 \cdot & A_{2,1}^{\alpha} \cdot A_{2,2}^{\beta} \end{pmatrix}.$$

- $\mathsf{TCom}(\mathsf{ck}, \mathsf{td}) \to (\mathsf{com}, St)$: Sample $\rho_1, \rho_2, s \xleftarrow{\$} \mathbb{Z}_p$. Set $St := (\mathsf{td}, \tau, \rho_1, \rho_2, s)$ and compute $\mathsf{com} := (C_1, C_2)$ for

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} := \begin{pmatrix} X_1^{\rho_1} \cdot g^s \\ X_2^{\rho_2} \cdot h^s \end{pmatrix}.$$

- $\mathsf{TCol}(St, c) \to (\varphi, R, s)$: Set $R := (R_1, R_2) := \left(g^s \cdot X_1^{-c}, h^s \cdot X_2^{-c}\right)$. Then, if $\mathbf{D}$ is not invertible, return $\perp$. Otherwise, compute $\varphi := (\alpha, \beta)$ for

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \mathbf{D}^{-1} \cdot \begin{pmatrix} \rho_1 + c \\ \rho_2 + c \end{pmatrix}.$$

Theorem 2. *If the* DDH *assumption holds relative to* GGen*, then* $\mathsf{CMT}_{\mathsf{DDH}}$ *is an* $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$*-weakly equivocable coset commitment scheme for* $\mathsf{LF}_{\mathsf{DDH}}$*, with*

$$\varepsilon_{\mathsf{b}} = 0, \quad \varepsilon_{\mathsf{g}} \leq 2/p, \quad \varepsilon_{\mathsf{t}} \leq 2/p.$$

Concretely, for any PPT algorithm $\mathcal{A}$*, there is a PPT algorithm* $\mathcal{B}$ *with* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and*

$$\mathsf{Adv}^{Q\text{-keydist}}_{\mathcal{A},\mathsf{CMT}_{\mathsf{DDH}}}(\lambda) \leq \mathsf{Adv}^{2Q\text{-DDH}}_{\mathcal{A},\mathsf{GGen}}(\lambda)..$$

The proof of Theorem 2 is split into a sequence of lemmas, showing the required properties of a weakly equivocable coset commitment scheme separately. The homomorphism property is easy to verify. For the

remaining properties, we first have to define a set Good. We define it as the set of non-zero parameters and domain elements, i.e.,

$$\mathsf{Good} = \left\{((g,h),x) \in \mathbb{G}^2 \times \mathbb{Z}_p \;\middle|\; (g,h) \in \mathsf{LF}_{\mathsf{DDH}}.\mathsf{Gen}(1^\lambda) \wedge h \neq g^0 \wedge x \neq 0\right\}.$$

It is clear that membership in Good can be decided efficiently. Further, for $(g,h) \leftarrow \mathsf{LF}_{\mathsf{DDH}}.\mathsf{Gen}(1^\lambda)$ and $x \xleftarrow{\$} \mathbb{Z}_p$, the probability that $((g,h),x) \notin \mathsf{Good}$ is at most $2/p$ by a union bound over the two events $h = g^0$ and $x = 0$. This shows $\varepsilon_{\mathsf{g}} \leq 2/p$. We proceed by showing that the commitment scheme satisfies the uniform keys property, the weak trapdoor property, multi-key indistinguishability, and is statistically coset binding. The proofs of the uniform keys property and the weak trapdoor property are similar to the proofs of the corresponding statement in [PW23].

Lemma 4. *The scheme $\mathsf{CMT}_{\mathsf{DDH}}$ satisfies the uniform keys property of an $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$-weakly equivocable coset commitment scheme for $\mathsf{LF}_{\mathsf{DDH}}$.*

Proof. Let $((g,h),x) \in \mathsf{Good}$ and define $(X_1, X_2) \in \mathbb{G}^2$ to be the image of x under F, i.e., $X_1 = g^x$ and $X_2 = h^x$. We have to argue that the distribution of $((g,h),x,\mathbf{A})$ for a commitment key $\mathbf{A}$ as output by algorithm TGen is the same as for a random $\mathbf{A} \xleftarrow{\$} \mathbb{G}^{2\times 2}$. Recall that $\mathbf{A}$ output by TGen has the form

$$\mathbf{A} = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} = \begin{pmatrix} X_1^{d_{1,1}} & X_1^{d_{1,2}} \\ X_2^{d_{2,1}} & X_2^{d_{2,2}} \end{pmatrix} \in \mathbb{G}^{2\times 2},$$

where the $d_{i,j}$ are sampled uniformly at random from $\mathbb{Z}_p$. As $((g,h),x) \in \mathsf{Good}$, we know that X_1 and X_2 are generators of $\mathbb{G}$, and therefore $\mathbf{A}$ is uniformly random over $\mathbb{G}^{2\times 2}$. □

Lemma 5. *The scheme $\mathsf{CMT}_{\mathsf{DDH}}$ satisfies the weak trapdoor property of an $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$-weakly equivocable coset commitment scheme for $\mathsf{LF}_{\mathsf{DDH}}$, where $\varepsilon_{\mathsf{t}} \leq 2/p$.*

Proof. Fix parameters $g, h \in \mathbb{G}$ and $x \in \mathbb{Z}_p$ and a challenge $c \in \mathbb{Z}_p$ such that $((g,h),x) \in \mathsf{Good}$. Define $(X_1, X_2) \in \mathbb{G}^2$ to be the image of x under F, i.e., $X_1 = g^x$ and $X_2 = h^x$. According to the definition of the weak trapdoor property, we have to consider two different distributions $\mathcal{T}_0$ and $\mathcal{T}_1$ of tuples

$$((g,h),\mathbf{A},(\mathbf{D},X_1,X_2),x,c,(C_1,C_2),((\alpha,\beta),(R_1,R_2),s))$$

Here, g, h, x, c, X_1, X_2 are fixed as above and $\mathbf{A}, \mathbf{D}, X_1, X_2$ are output by TGen in both distributions $\mathcal{T}_0$ and $\mathcal{T}_1$. In distribution $\mathcal{T}_1$, the remaining components $(C_1,C_2),((\alpha,\beta),(R_1,R_2),s$ are sampled via

$$((C_1,C_2),St) \leftarrow \mathsf{TCom}(\mathsf{ck},\mathsf{td}),\ ((\alpha,\beta),(R_1,R_2),s) \leftarrow \mathsf{TCol}(St,c).$$

In distribution $\mathcal{T}_0$, the remaining components $(C_1,C_2),((\alpha,\beta),(R_1,R_2),s$ are sampled as

$$\begin{aligned} r \xleftarrow{\$} \mathbb{Z}_p,\ R_1 := g^r,\ R_2 := h^r,\ s := c\cdot x + r,\\ \alpha,\beta \xleftarrow{\$} \mathbb{Z}_p,\ (C_1,C_2) := \mathsf{Com}(\mathbf{A},(R_1,R_2);(\alpha,\beta)). \end{aligned}$$

We will now gradually change this distribution $\mathcal{T}_0$ until we arrive at the distribution $\mathcal{T}_1$. Before we do that, we assume that in both distributions $\mathcal{T}_0$ and $\mathcal{T}_1$, the matrix $\mathbf{D} \in \mathbb{Z}_p^{2\times 2}$ has full rank. As $\mathbf{D}$ is sampled uniformly at random over $\mathbb{Z}_p^{2\times 2}$, we know that it has full rank except with probability $1/p$. Thus, this assumption will add $2/p$ to our final bound.

In our first step, we remove the dependence on r and x. That is, we define a new distribution, in which the remaining components $(C_1,C_2),((\alpha,\beta),(R_1,R_2),s$ are samples as

$$\begin{aligned} s \xleftarrow{\$} \mathbb{Z}_p,\ R_1 := g^s \cdot X_1^{-c},\ R_2 := h^s \cdot X_2^{-c},\\ \alpha,\beta \xleftarrow{\$} \mathbb{Z}_p,\ (C_1,C_2) := \mathsf{Com}(\mathbf{A},(R_1,R_2);(\alpha,\beta)). \end{aligned}$$

It is clear that $\mathcal{T}_0$ and this new distribution are identical[6]. Next, we want to make (C_1,C_2) independent of (R_1,R_2). For that, we consider the mapping from (α,β) to (C_1,C_2) induced by Com, namely,

$$\Psi\colon \mathbb{Z}_p^2 \to \mathbb{G}^2,\quad (\alpha,\beta) \mapsto (C_1,C_2) = (R_1 \cdot A_{1,1}^{\alpha} \cdot A_{1,2}^{\beta}, R_2 \cdot A_{2,1}^{\alpha} \cdot A_{2,2}^{\beta}).$$

[6]Essentially, we used the special honest-verifier zero-knowledge property of the Chaum-Pedersen identification scheme [CP93, KW03, KMP16] here.

Using the assumption that $\mathbf{D}$ has full rank, that g, h, X_1, X_2 are generators of $\mathbb{G}$, and the definition of the $A_{i,j}$, we see that Ψ is a bijection, and (C_1, C_2) is uniformly random over $\mathbb{G}^2$. Therefore, we can equivalently write the distribution as

$$\begin{gathered} s \xleftarrow{\$} \mathbb{Z}_p,\ R_1 := g^s \cdot X_1^{-c},\ R_2 := h^s \cdot X_2^{-c}, \\ \rho_1, \rho_2 \xleftarrow{\$} \mathbb{Z}_p,\ (C_1, C_2) := (X_1^{\rho_1} \cdot g^s, X_2^{\rho_2} \cdot h^s),\ \ (\alpha, \beta) := \Psi^{-1}(C_1, C_2). \end{gathered}$$

We claim that this is exactly the distribution $\mathcal{T}_1$. For that, it is sufficient to argue that the way algorithm TCol computes (α, β), i.e., as $(\alpha, \beta)^t := \mathbf{D}^{-1}(\rho_1 + c, \rho_2 + c)^t$, is identical to Ψ^{-1}. By definition of Ψ, the expression $(\alpha, \beta) = \Psi^{-1}(C_1, C_2)$ is equivalent to

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} R_1 \cdot & A_{1,1}^{\alpha} \cdot & A_{1,2}^{\beta} \\ R_2 \cdot & A_{2,1}^{\alpha} \cdot & A_{2,2}^{\beta} \end{pmatrix}.$$

Using our definition of $\mathbf{A}$, C_1, C_2, and R_1, R_2, this is equivalent to

$$\begin{pmatrix} X_1^{\rho_1} \cdot g^s \\ X_2^{\rho_2} \cdot h^s \end{pmatrix} = \begin{pmatrix} g^s \cdot X_1^{-c} \cdot & X_1^{d_{1,1}\alpha} \cdot & X_1^{d_{1,2}\beta} \\ h^s \cdot X_2^{-c} \cdot & X_2^{d_{2,1}\alpha} \cdot & X_2^{d_{2,2}\beta} \end{pmatrix}.$$

The g^s and h^s terms cancel out, and this is equivalent to

$$\begin{pmatrix} \rho_1 + c \\ \rho_2 + c \end{pmatrix} = \mathbf{D} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

which concludes the proof. □

Lemma 6. *The scheme* $\mathsf{CMT}_{\mathsf{DDH}}$ *satisfies the statistical coset binding property of an* $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$*-weakly equivocable coset commitment scheme for* $\mathsf{LF}_{\mathsf{DDH}}$*, where* $\varepsilon_{\mathsf{b}} = 0$*.*

Proof. To show that the scheme is statistically coset binding, we have to present an algorithm Ext that outputs (R_1, R_2) on input $\mathbf{A} \in \mathbb{Z}_p^{2\times 2}$ and $\mathsf{com} = (C_1, C_2) \in \mathbb{G}^2$. Algorithm Ext just outputs $(R_1, R_2) := (C_1, C_2)$. It remainst to show that Ext satisfies the conditions of the statistical coset binding property. Concretely, we have to consider the following experiment for any adversary $\mathcal{A}$: First, parameters $\mathsf{par} := (g, h) \leftarrow \mathsf{LF}_{\mathsf{DDH}}.\mathsf{Gen}(1^\lambda)$ and a commitment key $\mathsf{ck} := \mathbf{A} \leftarrow \mathsf{BGen}(\mathsf{par})$ are generated. Then, $\mathcal{A}$ gets par and ck and outputs a commitment $\mathsf{com} = (C_1, C_2) \in \mathbb{G}^2$. The extractor Ext is run, outputting $R_1 = C_1, R_2 = C_2$. Finally, $\mathcal{A}$ outputs $(R_1', R_2') \in \mathbb{G}^2$ and $(\alpha', \beta') \in \mathbb{Z}_p^2$. We have to bound the probability of the event

$$\mathsf{Com}(\mathsf{ck}, (R_1', R_2'); (\alpha', \beta')) = \mathsf{com} \ \wedge\ \forall \delta \in \mathbb{Z}_p : \begin{pmatrix} R_1' \\ R_2' \end{pmatrix} \neq \begin{pmatrix} R_1 \cdot g^\delta \\ R_2 \cdot h^\delta \end{pmatrix}.$$

Now, we rewrite the first condition according to the definition of Com, taking into account the definition of $\mathbf{A}$ in algorithm BGen. Further, we use $(R_1, R_2) = (C_1, C_2)$. Then, this is equivalent to

$$\begin{pmatrix} R_1' \cdot & g^{a\cdot\alpha'} \cdot g^{b\cdot\beta'} \\ R_2' \cdot & h^{a\cdot\alpha'} \cdot h^{b\cdot\beta'} \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} \ \wedge\ \forall \delta \in \mathbb{Z}_p : \begin{pmatrix} R_1' \\ R_2' \end{pmatrix} \neq \begin{pmatrix} R_1 \cdot g^\delta \\ R_2 \cdot h^\delta \end{pmatrix}.$$

Now, by taking $\delta = -(a \cdot \alpha' + b \cdot \beta')$, it is easy to see that this can never hold, and therefore the event we have to bound can never occur. □

Lemma 7. *For any PPT algorithm* $\mathcal{A}$*, there is a PPT algorithm* $\mathcal{B}$ *with* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and*

$$\mathsf{Adv}^{Q\text{-}\mathsf{keydist}}_{\mathcal{A},\mathsf{CMT}_{\mathsf{DDH}}}(\lambda) \leq \mathsf{Adv}^{2Q\text{-}\mathsf{DDH}}_{\mathcal{A},\mathsf{GGen}}(\lambda).$$

Proof. To prove multi-key indistinguishability based on the $2Q$-DDH assumption, we assume the existence of an adversary $\mathcal{A}$ against the multi-key indistinguishability of $\mathsf{CMT}_{\mathsf{DDH}}$, and turn it into an algorithm that breaks the $2Q$-DDH assumption. That is, we construct a reduction $\mathcal{B}$ that simulates the multi-key indistinguishability game for $\mathcal{A}$ and runs in the $2Q$-DDH game. Reduction $\mathcal{B}$ is as follows:

1. $\mathcal{B}$ gets as input $\mathbb{G}, p, g, h$ and $2Q$ group elements $(u_i, v_i)_{i=1}^{2Q}$.

2. $\mathcal{B}$ defines $\mathsf{par} := (g, h)$ and samples $x \xleftarrow{\$} \mathbb{Z}_p$. If $((g, h), x) \notin \mathsf{Good}$, $\mathcal{B}$ returns 0 and terminates.

3. Otherwise, $\mathcal{B}$ defines Q commitment keys $\mathsf{ck}_1, \ldots, \mathsf{ck}_Q \in \mathbb{G}^{2\times 2}$ via

$$\mathsf{ck}_i := \begin{pmatrix} u_{2i-1} & u_{2i} \\ v_{2i-1} & v_{2i-1} \end{pmatrix} \text{ for all } i \in [Q].$$

4. $\mathcal{B}$ runs $\mathcal{A}$ on input par, x and $(\mathsf{ck}_i)_{i\in[Q]}$. It returns whatever $\mathcal{A}$ returns.

It is clear that the running time of $\mathcal{B}$ is dominated by the running time of $\mathcal{A}$. Further, assume that $\mathcal{B}$'s input satisfies $u_i = g^{a_i}, v_i = g^{a_i}$ for random $a_i \in \mathbb{Z}_p$. In this case, the commitment keys are distributed as if they are generated by BGen, and $\mathcal{B}$ perfectly simulates game Q-$\mathbf{KEYDIST}^{\mathcal{A}}_{0,\mathsf{CMT}}(\lambda)$ for $\mathcal{A}$. On the other hand, if $u_i, v_i \xleftarrow{\$} \mathbb{G}$ for all $i \in [2Q]$, then the resulting commitment keys ck_i are distributed uniformly over $\mathbb{G}^{2\times 2}$, and $\mathcal{B}$ perfectly simulates game Q-$\mathbf{KEYDIST}^{\mathcal{A}}_{1,\mathsf{CMT}}(\lambda)$ for $\mathcal{A}$. This shows the claim. □

6 Efficiency

Here, we analyze the efficiency of our schemes. We first explain minor optimizations that improve signature size and communication complexity. Then, we focus on the asymptotic and concrete efficiency of our schemes.

Further Optimizations. We describe the optimizations for our tight construction in Section 4.2, but they also apply to our other scheme. Our first optimization is to reduce the communication complexity by deriving the commitment randomness φ, which consists of two field elements, from a short seed of length λ bit using a random oracle $\bar{\mathsf{H}}\colon \{0,1\}^* \to \mathbb{Z}_p^2$. Then, instead of sending φ in the second round, each signer sends its seed, and the signers locally derive all φ's and aggregate them. By the unpredictability of the random oracle, the scheme stays secure. Our second optimization allows us to remove the commitment com, which consists of two group elements, from the final signature. The idea is to replace it by a hash of com of length 2λ using another random oracle $\hat{\mathsf{H}}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$. Concretely, the signers first complete the signing protocol as before, but to define signer specific challenges, they compute $c_i := \mathsf{H}_c(\mathsf{pk}_i, h, \mathsf{m}, \langle\mathcal{P}\rangle, B)$ for all $i \in [N]$, where $h := \hat{\mathsf{H}}(\mathsf{com})$. In the end, the signature is $\sigma := (h, \varphi, s, B)$. The signature is verified by first recomputing the c_i's as before (using h instead of com) and by checking the commitment after hashing, i.e., using the equation

$$h = \hat{\mathsf{H}}\left(\mathsf{Com}\left(\mathsf{ck}, \mathsf{F}(s) - \sum_{i=1}^{N} c_i \cdot X_{i,b_i}; \varphi\right)\right).$$

In a security proof, we would use the collision-resistance and observability of $\hat{\mathsf{H}}$ to reduce this equation to the original verification equation.

Asymptotics. In Table 2, we compare the asymptotic sizes of public keys and signatures and the communication complexity per signer for our schemes with previous schemes in the pairing-free setting. We see that our schemes are much more efficient than the schemes in [PW23]. Especially, the asymptotic efficiency of our schemes is comparable with most non-tight or three-round schemes.

Concrete Parameters. We compare the concrete efficiency and security level of two-round multi-signatures in the pairing-free setting in Table 3. The numbers are computed using a Python script, see Appendix C. Our comparison assumes that all constructions are instantiated with the secp256k1 curve, and we assume security parameter $\lambda = 128$. We compute the concrete security level based on the security bounds (see Table 1) assuming the underlying assumption is 128 bit hard, and assuming $Q_H = 2^{30}$ hash queries and $Q_S = 2^{20}$ signing queries. We see that our tightly secure scheme outperforms Chopsticks II, the only other two-round scheme with comparable security level, by a factor of more than 3 in a signature size, and more than 2 in the communication complexity. Finally, we also remark that our scheme is at least twice as efficient as Chopsticks II in terms of computation.

Acknowledgments. Benedikt Wagner was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 507237585.

References

[AB21] Handan Kilinç Alper and Jeffrey Burdges. Two-round trip schnorr multi-signatures via delinearized witnesses. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 157–188, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 2.)

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 572–590. Springer, Heidelberg, April 2012. (Cited on page 2, 3, 10.)

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000. (Cited on page 2.)

[BD21] Mihir Bellare and Wei Dai. Chain reductions for multi-signatures and the HBMS scheme. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 650–678. Springer, Heidelberg, December 2021. (Cited on page 2, 3, 4.)

[BDN18] Dan Boneh, Manu Drijvers, and Gregory Neven. Compact multi-signatures for smaller blockchains. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 435–464. Springer, Heidelberg, December 2018. (Cited on page 2, 3, 4.)

[BHJ+15] Christoph Bader, Dennis Hofheinz, Tibor Jager, Eike Kiltz, and Yong Li. Tightly-secure authenticated key exchange. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 629–658. Springer, Heidelberg, March 2015. (Cited on page 2.)

[BJLS16] Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 273–304. Springer, Heidelberg, May 2016. (Cited on page 2.)

[BKKP15] Olivier Blazy, Saqib A. Kakvi, Eike Kiltz, and Jiaxin Pan. Tightly-secure signatures from chameleon hash functions. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 256–279. Springer, Heidelberg, March / April 2015. (Cited on page 2.)

[BKP14] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) identity-based encryption from affine message authentication. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, Heidelberg, August 2014. (Cited on page 2.)

[BL16] Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and id-based encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 404–434. Springer, Heidelberg, December 2016. (Cited on page 2.)

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006. (Cited on page 1, 2, 3, 4, 7.)

[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. (Cited on page 1.)

[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. (Cited on page 1.)

[BTT22] Cecilia Boschini, Akira Takahashi, and Mehdi Tibouchi. MuSig-L: Lattice-based multi-signature with single-round online phase. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 276–305. Springer, Heidelberg, August 2022. (Cited on page 2.)

[CAHL+22] Rutchathon Chairattana-Apirom, Lucjan Hanzlik, Julian Loss, Anna Lysyanskaya, and Benedikt Wagner. PI-cut-choo and friends: Compact blind signatures via parallel instance cut-and-choose and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 3–31. Springer, Heidelberg, August 2022. (Cited on page 9.)

[CKM21] Elizabeth Crites, Chelsea Komlo, and Mary Maller. How to prove schnorr assuming schnorr: Security of multi- and threshold signatures. Cryptology ePrint Archive, Report 2021/1375, 2021. https://eprint.iacr.org/2021/1375. (Cited on page 1, 2, 7.)

[CP93] David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 89–105. Springer, Heidelberg, August 1993. (Cited on page 20.)

[CW13] Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, Heidelberg, August 2013. (Cited on page 2.)

[DEF+19] Manu Drijvers, Kasra Edalatnejad, Bryan Ford, Eike Kiltz, Julian Loss, Gregory Neven, and Igors Stepanovs. On the security of two-round multi-signatures. In *2019 IEEE Symposium on Security and Privacy*, pages 1084–1101. IEEE Computer Society Press, May 2019. (Cited on page 1.)

[DG21] Hannah Davis and Felix Günther. Tighter proofs for the SIGMA and TLS 1.3 key exchange protocols. In Kazue Sako and Nils Ole Tippenhauer, editors, *ACNS 21, Part II*, volume 12727 of *LNCS*, pages 448–479. Springer, Heidelberg, June 2021. (Cited on page 2.)

[DGJL21] Denis Diemert, Kai Gellert, Tibor Jager, and Lin Lyu. More efficient digital signatures with tight multi-user security. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 1–31. Springer, Heidelberg, May 2021. (Cited on page 2.)

[DOTT21] Ivan Damgård, Claudio Orlandi, Akira Takahashi, and Mehdi Tibouchi. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 99–130. Springer, Heidelberg, May 2021. (Cited on page 2, 7.)

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013. (Cited on page 8.)

[FH21] Masayuki Fukumitsu and Shingo Hasegawa. A tightly secure ddh-based multisignature with public-key aggregation. *Int. J. Netw. Comput.*, 11(2):319–337, 2021. (Cited on page 2, 3, 4.)

[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. (Cited on page 2.)

[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. (Cited on page 2.)

[GHKW16] Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Heidelberg, May 2016. (Cited on page 2.)

[GJ18] Kristian Gjøsteen and Tibor Jager. Practical and tightly-secure digital signatures and authenticated key exchange. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 95–125. Springer, Heidelberg, August 2018. (Cited on page 2.)

[GJKW07] Eu-Jin Goh, Stanislaw Jarecki, Jonathan Katz, and Nan Wang. Efficient signature schemes with tight reductions to the Diffie-Hellman problems. *Journal of Cryptology*, 20(4):493–514, October 2007. (Cited on page 5.)

[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. (Cited on page 2.)

[HJK+21] Shuai Han, Tibor Jager, Eike Kiltz, Shengli Liu, Jiaxin Pan, Doreen Riepel, and Sven Schäge. Authenticated key exchange and signatures with tight security in the standard model. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 670–700, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 2.)

[HKL19] Eduard Hauck, Eike Kiltz, and Julian Loss. A modular treatment of blind signatures from identification schemes. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 345–375. Springer, Heidelberg, May 2019. (Cited on page 9.)

[Hof17] Dennis Hofheinz. Adaptive partitioning. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 489–518. Springer, Heidelberg, April / May 2017. (Cited on page 2.)

[IN83] Kazuharu Itakura and Katsuhiro Nakamura. A public-key cryptosystem suitable for digital multisignatures. *NEC Research & Development*, (71):1–8, 1983. (Cited on page 1.)

[KD04] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer, Heidelberg, August 2004. (Cited on page 2.)

[KLR21] Jonathan Katz, Julian Loss, and Michael Rosenberg. Boosting the security of blind signature schemes. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 468–492. Springer, Heidelberg, December 2021. (Cited on page 9.)

[KMP16] Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal security proofs for signatures from identification schemes. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 33–61. Springer, Heidelberg, August 2016. (Cited on page 2, 3, 20.)

[KW03] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM CCS 2003*, pages 155–164. ACM Press, October 2003. (Cited on page 2, 3, 20.)

[LLGW20] Xiangyu Liu, Shengli Liu, Dawu Gu, and Jian Weng. Two-pass authenticated key exchange with explicit authentication and tight security. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 785–814. Springer, Heidelberg, December 2020. (Cited on page 2.)

[LOS+06] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, and Brent Waters. Sequential aggregate signatures and multisignatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 465–485. Springer, Heidelberg, May / June 2006. (Cited on page 1.)

[LP20] Roman Langrehr and Jiaxin Pan. Unbounded HIBE with tight security. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 129–159. Springer, Heidelberg, December 2020. (Cited on page 2.)

[MOR01] Silvio Micali, Kazuo Ohta, and Leonid Reyzin. Accountable-subgroup multisignatures: Extended abstract. In Michael K. Reiter and Pierangela Samarati, editors, *ACM CCS 2001*, pages 245–254. ACM Press, November 2001. (Cited on page 1.)

[MPSW19] Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple schnorr multi-signatures with applications to bitcoin. *Des. Codes Cryptogr.*, 87(9):2139–2164, 2019. (Cited on page 2, 3, 4.)

[NRS21] Jonas Nick, Tim Ruffing, and Yannick Seurin. MuSig2: Simple two-round Schnorr multi-signatures. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 189–221, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 2, 3, 4.)

[NRSW20] Jonas Nick, Tim Ruffing, Yannick Seurin, and Pieter Wuille. MuSig-DN: Schnorr multi-signatures with verifiably deterministic nonces. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1717–1731. ACM Press, November 2020. (Cited on page 2, 3, 4.)

[PW22] Jiaxin Pan and Benedikt Wagner. Lattice-based signatures with tight adaptive corruptions and more. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 347–378. Springer, Heidelberg, March 2022. (Cited on page 2.)

[PW23] Jiaxin Pan and Benedikt Wagner. Chopsticks: Fork-free two-round multi-signatures from non-interactive assumptions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 597–627. Springer, Heidelberg, April 2023. (Cited on page 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 18, 19, 20, 22, 27, 29, 30.)

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991. (Cited on page 2.)

[TSS+23] Kaoru Takemure, Yusuke Sakai, Bagus Santoso, Goichiro Hanaoka, and Kazuo Ohta. More efficient two-round multi-signature scheme with provably secure parameters. Cryptology ePrint Archive, Report 2023/155, 2023. https://eprint.iacr.org/2023/155. (Cited on page 3, 4.)

[TZ23] Stefano Tessaro and Chenzhi Zhu. Threshold and multi-signature schemes from linear hash functions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 628–658. Springer, Heidelberg, April 2023. (Cited on page 2, 3, 4, 9.)

Appendix

A Omitted Background on Key Aggregation

In this section, we give some background needed to understand the construction with key aggregation, formally presented in Appendix B. For the definition of key aggregation, we follow [PW23].

Definition 10 (Key Aggregation)**.** A multi-signature scheme $\mathsf{MS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is said to support key aggregation, if the algorithm Ver can be split into two deterministic polynomial time algorithms $\mathsf{Agg}, \mathsf{VerAgg}$ with the following syntax:

- $\mathsf{Agg}(\mathcal{P}) \to \tilde{\mathsf{pk}}$ takes as input a set $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ of public keys and outputs an aggregated key $\tilde{\mathsf{pk}}$.
- $\mathsf{VerAgg}(\tilde{\mathsf{pk}}, \mathsf{m}, \sigma) \to b$ is deterministic, takes as input an aggregated key $\tilde{\mathsf{pk}}$, a message m, and a signature σ, and outputs a bit $b \in \{0,1\}$.

That is, algorithm $\mathsf{Ver}(\mathcal{P}, \mathsf{m}, \sigma)$ can be written as $\mathsf{VerAgg}(\mathsf{Agg}(\mathcal{P}), \mathsf{m}, \sigma)$.

In addition to this definition, we need to adjust the definition of lossy soundness to the setting of aggregated keys. For this, we first recall the definition of aggregation lossy soundness from [PW23], and then define our new notion of coset aggregation lossy soundness.

Definition 11 (Aggregation Lossy Soundness)**.** Let $\mathsf{LF} = (\mathsf{Gen}, \mathsf{F})$ be a linear function family. We say that LF satisfies $\varepsilon_{\mathsf{al}}$-aggregation lossy soundness, if for any unbounded algorithm $\mathcal{A}$, the following probability is at most $\varepsilon_{\mathsf{al}}$:

$$\Pr\left[\mathsf{F}(s) - c \cdot \tilde{X} = R \;\middle|\; \begin{array}{l} \mathsf{par} \leftarrow \mathsf{Gen}(1^\lambda),\ X_1 \xleftarrow{\$} \mathcal{R}, \\ (St, (X_2, a_2), \ldots, (X_N, a_N)) \leftarrow \mathcal{A}(\mathsf{par}, X_1), \\ a_1 \xleftarrow{\$} \mathcal{S},\ (St', R) \leftarrow \mathcal{A}(St, a_1), \\ c \xleftarrow{\$} \mathcal{S},\ s \leftarrow \mathcal{A}(St', c), \\ \tilde{X} := \sum_{i=1}^N a_i X_i \end{array}\right].$$

Definition 12 (Coset Aggregation Lossy Soundness)**.** Let $\mathsf{LF} = (\mathsf{Gen}, \mathsf{F})$ be a linear function family. We say that LF satisfies $\varepsilon_{\mathsf{al}}$-coset aggregation lossy soundness, if for any unbounded algorithm $\mathcal{A}$, the following probability is at most $\varepsilon_{\mathsf{al}}$:

$$\Pr\left[\mathsf{F}(s) - c \cdot \tilde{X} \in R + \mathsf{F}(\mathcal{D}) \;\middle|\; \begin{array}{l} \mathsf{par} \leftarrow \mathsf{Gen}(1^\lambda),\ X_1 \xleftarrow{\$} \mathcal{R}, \\ (St, (X_2, a_2), \ldots, (X_N, a_N)) \leftarrow \mathcal{A}(\mathsf{par}, X_1), \\ a_1 \xleftarrow{\$} \mathcal{S},\ (St', R) \leftarrow \mathcal{A}(St, a_1), \\ c \xleftarrow{\$} \mathcal{S},\ s \leftarrow \mathcal{A}(St', c), \\ \tilde{X} := \sum_{i=1}^N a_i X_i \end{array}\right].$$

Lemma 8. *Let* LF *be a linear function family, such that for any* $\mathsf{par} \in \mathsf{Gen}(1^\lambda)$*, the domain* $\mathcal{D}_{\mathsf{par}}$ *can be enumerated. Then, if* LF *satisfies* $\varepsilon_{\mathsf{al}}$*-aggregation lossy soundness, it also satisfies* ε_{l}*-coset aggregation lossy soundness.*

Proof. The proof is almost identical to the proof of Lemma 1, and details are left to the reader. □

B Construction with Key Aggregation

Here, we present the construction of two-round multi-signatures with key aggregation. Essentially, the construction is the same as in [PW23], but we show that it can be instantiated with our weaker notion of commitments. Before looking at the construction, we encourage the reader to consult Appendix A for necessary definitions regarding key aggregation. Let $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{F})$ be a linear function family and $\mathsf{CMT} = (\mathsf{BGen}, \mathsf{TGen}, \mathsf{Com}, \mathsf{TCom}, \mathsf{TCol})$ be an $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$-weakly equivocable coset commitment scheme for LF with key space $\mathcal{K}$, randomness space $\mathcal{G}$ and commitment space $\mathcal{H}$. Let $\mathsf{H}\colon \{0,1\}^* \to \mathcal{K}$, $\mathsf{H}_a\colon \{0,1\}^* \to \mathcal{S}$, and $\mathsf{H}_c\colon \{0,1\}^* \to \mathcal{S}$ be random oracles. We verbally describe the multi-signature scheme $\mathsf{ToothKA}[\mathsf{LF}, \mathsf{CMT}] = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ here, and give a more formal description in Figure 5.

Setup and Key Generation. The public parameters of the scheme are $\mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda)$. These parameters specify the linear function $\mathsf{F} = \mathsf{F}(\mathsf{par}, \cdot)$. To generate a pair of keys, a signer samples a secret key $\mathsf{sk} := x \xleftarrow{\$} \mathcal{D}$ and computes its public key as $\mathsf{pk} := X := \mathsf{F}(x)$.

Key Aggregation. Given a set $\mathcal{P} = \{\mathsf{pk}_1 = X_1, \ldots, \mathsf{pk}_N = X_N\}$ of public keys, the aggregate public key $\tilde{\mathsf{pk}}$ is computed as $\tilde{\mathsf{pk}} := \tilde{X} := \sum_{i=1}^{N} a_i \cdot X_i$, where $a_i := \mathsf{H}_a(\langle \mathcal{P} \rangle, \mathsf{pk}_i)$ for each $i \in [N]$.

Signing Protocol. Let $\mathsf{m} \in \{0,1\}^*$ be the message to be signed by a set of N signers. Each signer $i \in [N]$ holds a secret key $\mathsf{sk}_i = x_i \in \mathbb{Z}_p$. Let $\mathcal{P} = \{\mathsf{pk}_1 = X_1, \ldots, \mathsf{pk}_N = X_N\}$ be the set of respective public keys. We describe the signing protocol, i.e., algorithms $\mathsf{Sig}_0, \mathsf{Sig}_1, \mathsf{Sig}_2$, from the perspective of the first signer, holding secret key $\mathsf{sk}_1 = x_1$.

1. *Commitment Phase.* The signer computes the aggregated key $\tilde{\mathsf{pk}} := \mathsf{Agg}(\mathcal{P})$ and a commitment key $\mathsf{ck} := \mathsf{H}(\tilde{\mathsf{pk}}, \mathsf{m})$. The signer samples an element $r_1 \xleftarrow{\$} \mathcal{D}$ and computes $R_1 := \mathsf{F}(r_1)$. The signer commits to R_1 by sampling $\varphi_1 \xleftarrow{\$} \mathcal{G}$ and computing $\mathsf{com}_1 := \mathsf{Com}(\mathsf{ck}, R_1; \varphi_1)$. Then, it sends $\mathsf{pm}_{1,1} := \mathsf{com}_1$ to all other signers.

2. *Response Phase.* Let $\mathcal{M}_1 = (\mathsf{pm}_{1,1}, \ldots, \mathsf{pm}_{1,N})$ be the list of messages output by the signers in the commitment phase. Here, the message $\mathsf{pm}_{1,i}$ is sent by signer i and has the form $\mathsf{pm}_{1,i} = \mathsf{com}_i$. The signer computes an aggregated commitment $\mathsf{com} := \bigotimes_{i \in [N]} \mathsf{com}_i$ and a challenge $c := \mathsf{H}_c(\tilde{\mathsf{pk}}, \mathsf{com}, \mathsf{m})$. It computes a response $s_1 := c \cdot a_1 \cdot x_1 + r_1$, where $a_1 := \mathsf{H}_a(\langle \mathcal{P} \rangle, \mathsf{pk}_1)$ is as in the key aggregation algorithm. It sends $\mathsf{pm}_{2,1} := (s_1, \varphi_1)$ to all other signers.

3. *Aggregation Phase.* Let $\mathcal{M}_2 = (\mathsf{pm}_{2,1}, \ldots, \mathsf{pm}_{2,N})$ be the list of messages output by the signers in the response phase, where message $\mathsf{pm}_{2,i}$ is sent by signer i and has the form $\mathsf{pm}_{2,i} = (s_i, \varphi_i)$. The signers aggregate the responses and commitment randomness received in the previous messages via

$$s := \sum_{i \in [N]} s_i, \;\; \varphi := \bigoplus_{i \in [N]} \varphi_i.$$

   The final signature is $\sigma := (\mathsf{com}, s, \varphi)$.

Verification. Let $\mathsf{m} \in \{0,1\}^*$ be a message, $\mathcal{P} = \{\mathsf{pk}_1 = X_1, \ldots, \mathsf{pk}_N = X_N\}$ be a set of public keys, and $\sigma = (\mathsf{com}, s, \varphi)$ be a signature. To verify σ with respect to m and $\mathcal{P}$, one first computes the aggregated key $\tilde{\mathsf{pk}} = \tilde{X}$ as above. Then, one computes the commitment key $\mathsf{ck} := \mathsf{H}(\tilde{\mathsf{pk}}, \mathsf{m})$ and $c := \mathsf{H}_c(\tilde{\mathsf{pk}}, \mathsf{com}, \mathsf{m})$. The signature is valid, i.e., the verification outputs 1 if and only if the following equation holds:

$$\mathsf{com} = \mathsf{Com}\left(\mathsf{ck}, \mathsf{F}(s) - c \cdot \tilde{X}; \varphi\right).$$

Lemma 9. *Let* LF *be a linear function family. Let* CMT *be an* $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$*-weakly equivocable coset commitment scheme for* LF *Then* $\mathsf{ToothKA}[\mathsf{LF}, \mathsf{CMT}]$ *is complete.*

The proof of Lemma 9 is a similar calculation as in the proof of Lemma 2, and we leave it to the reader.

Theorem 3. *Let* LF *be a linear function family that satisfies key indistinguishability and* $\varepsilon_{\mathsf{al}}$*-coset aggregation lossy soundness. Let* CMT *be an* $(\varepsilon_{\mathsf{b}}, \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$*-weakly equivocable coset commitment scheme for* LF*. Further, let* $\mathsf{H}\colon \{0,1\}^* \to \mathcal{K}$*,* $\mathsf{H}_a\colon \{0,1\}^* \to \mathcal{S}$*, and* $\mathsf{H}_c\colon \{0,1\}^* \to \mathcal{S}$ *be random oracles. Then, the scheme* $\mathsf{ToothKA}[\mathsf{LF}, \mathsf{CMT}]$ *is* $\mathsf{MS\text{-}EUF\text{-}CMA}$ *secure.*

Concretely, for any PPT algorithm $\mathcal{A}$ *that makes at most* $Q_{\mathsf{H}}, Q_{\mathsf{H}_a}, Q_{\mathsf{H}_c}, Q_S$ *queries to oracles* $\mathsf{H}, \mathsf{H}_a, \mathsf{H}_c, \textsc{Sig}_0$*, respectively, there are PPT algorithms* $\mathcal{B}, \mathcal{B}'$ *with* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}), \mathbf{T}(\mathcal{B}') \approx \mathbf{T}(\mathcal{A})$ *and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{MS\text{-}EUF\text{-}CMA}}_{\mathcal{A},\mathsf{ToothKA}[\mathsf{LF},\mathsf{CMT}]}(\lambda) \leq\; & \varepsilon_{\mathsf{g}} + 4Q_S^2\varepsilon_{\mathsf{t}} + 4Q_S\varepsilon_{\mathsf{g}} + 4Q_S Q_{\mathsf{H}} Q_{\mathsf{H}_c}\varepsilon_{\mathsf{b}} \\ & + \frac{4Q_S}{|\mathcal{R}|} + \frac{4Q_S Q_{\mathsf{H}_a} Q_{\mathsf{H}_c}}{|\mathcal{S}|} + 4Q_S Q_{\mathsf{H}_a} Q_{\mathsf{H}_c}\varepsilon_{\mathsf{al}} \\ & + 4Q_S\left(\mathsf{Adv}^{Q_{\mathsf{H}}\text{-}\mathsf{keydist}}_{\mathcal{B},\mathsf{CMT}}(\lambda) + \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}',\mathsf{LF}}(\lambda)\right).\end{aligned}$$

Alg $\mathsf{Setup}(1^\lambda)$
01 **return** $\mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda)$

Alg $\mathsf{Gen}(\mathsf{par})$
02 $\mathsf{sk} := x \xleftarrow{\$} \mathcal{D}$
03 $\mathsf{pk} := X := \mathsf{F}(x)$
04 **return** $(\mathsf{pk}, \mathsf{sk})$

Alg $\mathsf{Agg}(\mathcal{P})$
05 **parse** $\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\} := \mathcal{P}$
06 **for** $i \in [N]$: **parse** $X_i := \mathsf{pk}_i$
07 **for** $i \in [N] : a_i := \mathsf{H}_a(\langle \mathcal{P} \rangle, \mathsf{pk}_i)$
08 **return** $\tilde{\mathsf{pk}} := \tilde{X} := \sum_{i=1}^{N} a_i \cdot X_i$

Alg $\mathsf{VerAgg}(\tilde{\mathsf{pk}}, \mathsf{m}, \sigma)$
09 **parse** $\tilde{X} := \tilde{\mathsf{pk}},\ (\mathsf{com}, s, \varphi) := \sigma$
10 $c := \mathsf{H}_c(\tilde{\mathsf{pk}}, \mathsf{com}, \mathsf{m})$
11 $R := \mathsf{F}(s) - c \cdot \tilde{X}$
12 $\mathsf{ck} := \mathsf{H}(\tilde{\mathsf{pk}}, \mathsf{m})$
13 **if** $\mathsf{com} \neq \mathsf{Com}(\mathsf{ck}, R; \varphi)$: **return** 0
14 **return** 1

Alg $\mathsf{Ver}(\mathcal{P}, \mathsf{m}, \sigma)$
15 $\tilde{\mathsf{pk}} := \mathsf{Agg}(\mathcal{P})$
16 **return** $\mathsf{VerAgg}(\tilde{\mathsf{pk}}, \mathsf{m}, \sigma)$

Alg $\mathsf{Sig}_0(\mathcal{P}, \mathsf{sk}_1, \mathsf{m})$
17 **parse** $x_1 := \mathsf{sk}_1$
18 $\tilde{\mathsf{pk}} := \mathsf{Agg}(\mathcal{P}),\ \mathsf{ck} := \mathsf{H}(\tilde{\mathsf{pk}}, \mathsf{m})$
19 $r_1 \xleftarrow{\$} \mathcal{D},\ R_1 := \mathsf{F}(r_1),\ \varphi_1 \xleftarrow{\$} \mathcal{G}$
20 $\mathsf{pm}_{1,1} := \mathsf{com}_1 := \mathsf{Com}(\mathsf{ck}, R_1; \varphi_1)$
21 $St_1 := (\tilde{\mathsf{pk}}, x_1, r_1, \varphi_1, \mathsf{m})$
22 **return** $(\mathsf{pm}_{1,1}, St_1)$

Alg $\mathsf{Sig}_1(St_1, \mathcal{M}_1)$
23 **parse** $(\mathsf{pm}_{1,1}, \ldots, \mathsf{pm}_{1,N}) := \mathcal{M}_1$
24 **parse** $(\tilde{\mathsf{pk}}, x_1, r_1, \varphi_1, \mathsf{m}) := St_1$
25 **for** $i \in [N]$: **parse** $\mathsf{com}_i := \mathsf{pm}_{1,i}$
26 $\mathsf{com} := \bigotimes_{i \in [N]} \mathsf{com}_i$
27 $c := \mathsf{H}_c(\tilde{\mathsf{pk}}, \mathsf{com}, \mathsf{m})$
28 $a_1 := \mathsf{H}_a(\langle \mathcal{P} \rangle, \mathsf{pk}_1)$
29 $s_1 := c \cdot a_1 \cdot x_1 + r_1$
30 $\mathsf{pm}_{2,1} := (s_1, \varphi_1)$
31 **return** $(\mathsf{pm}_{2,1}, St_2 := \mathsf{com})$

Alg $\mathsf{Sig}_2(St_2, \mathcal{M}_2)$
32 **parse** $\mathsf{com} := St_2$
33 **parse** $(\mathsf{pm}_{2,1}, \ldots, \mathsf{pm}_{2,N}) := \mathcal{M}_2$
34 **for** $i \in [N]$: **parse** $(s_i, \varphi_i) := \mathsf{pm}_{2,i}$
35 $s := \sum_{i=1}^{N} s_i,\ \varphi := \bigoplus_{i=1}^{N} \varphi_i$
36 **return** $\sigma := (\mathsf{com}, s, \varphi)$

Figure 5: The multi-signature scheme $\mathsf{ToothKA}[\mathsf{LF}, \mathsf{CMT}] = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ with key aggregation for a linear function family $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{F})$ and a weakly equivocable coset commitment scheme $\mathsf{CMT} = (\mathsf{BGen}, \mathsf{TGen}, \mathsf{Com}, \mathsf{TCom}, \mathsf{TCol})$ for LF. The construction is the same as in [PW23], instantiated with our new building blocks.

Proof. The proof is an adaptation of the proof of Theorem 1 in [PW23] to our new building blocks. At the same time, we can think of the proof as a simplification of the proof of Theorem 1. For these reasons, we only sketch the proof informally. The proof is identical to the proof of Theorem 1 in [PW23] except for $\mathbf{G}_7$ and the fact that the final reduction reduces to coset aggregation lossy soundness instead of aggregation lossy soundness.

Game $\mathbf{G}_0$: This is the original security game $\mathbf{MS\text{-}EUF\text{-}CMA}^{\mathcal{A}}_{\mathsf{ToothKA}[\mathsf{LF},\mathsf{CMT}]}$. As in the proof of Theorem 1, we can omit signing oracle $\textsc{Sig}_2$ without loss of generality. We have

$$\mathsf{Adv}^{\mathsf{MS\text{-}EUF\text{-}CMA}}_{\mathcal{A},\mathsf{ToothKA}[\mathsf{LF},\mathsf{CMT}]}(\lambda) = \Pr[\mathbf{G}_0 \Rightarrow 1].$$

Game $\mathbf{G}_1$: We let the game abort if $(\mathsf{par}, x_1) \notin \mathsf{Good}$, where x_1 is the secret key of the signer that is simulated by the game. As in the proof of Theorem 1 (game $\mathbf{G}_3$), we can argue that

$$|\Pr[\mathbf{G}_0 \Rightarrow 1] - \Pr[\mathbf{G}_1 \Rightarrow 1]| \leq \Pr[(\mathsf{par}, x_1) \notin \mathsf{Good}] \leq \varepsilon_{\mathbf{g}}.$$

Game $\mathbf{G}_2$: We introduce a map b mapping inputs $(\tilde{\mathsf{pk}}, \mathsf{m})$ to random oracle H to bits. For each such input for which $b[\tilde{\mathsf{pk}}, \mathsf{m}]$ is not yet defined, we set it to 1 with probability $1/(Q_S + 1)$, and to 0 otherwise. The game additionally aborts if $b[\tilde{\mathsf{pk}}, \mathsf{m}] = 1$ for a signing query or $b[\tilde{\mathsf{pk}}, \mathsf{m}] = 0$ for the forgery. We can argue that

$$\Pr[\mathbf{G}_2 \Rightarrow 1] \geq \frac{1}{4Q_S} \cdot \Pr[\mathbf{G}_1 \Rightarrow 1].$$

Game $\mathbf{G}_3$: We change how commitment keys ck output by random oracle H on inputs $(\tilde{\mathsf{pk}}, \mathsf{m})$ are sampled. Namely, if $b[\tilde{\mathsf{pk}}, \mathsf{m}] = 0$, then we sample ck with a trapdoor via $(\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{TGen}(\mathsf{par}, X_1)$, where X_1 is the public key of the signer simulated by the game. Otherwise, if $b[\tilde{\mathsf{pk}}, \mathsf{m}] = 1$, we sample ck in

binding mode via $\mathsf{ck} \leftarrow \mathsf{BGen}(\mathsf{par})$. Indistinguishability can be argued using the uniform keys property of CMT and the multi-key indistinguishability property of CMT. We get a reduction $\mathcal{B}$ with

$$|\Pr[\mathbf{G}_2 \Rightarrow 1] - \Pr[\mathbf{G}_3 \Rightarrow 1]| \leq \mathsf{Adv}^{Q_\mathsf{H}\text{-}\mathsf{keydist}}_{\mathcal{B},\mathsf{CMT}}(\lambda).$$

Game $\mathbf{G}_4$: In this game, we use the trapdoor td generated in random oracle H to simulate the signing oracle. This works out because due to the changes in $\mathbf{G}_2$ and $\mathbf{G}_3$, the commitment key ck used in signing queries has been sampled with a trapdoor. We can argue that

$$|\Pr[\mathbf{G}_3 \Rightarrow 1] - \Pr[\mathbf{G}_4 \Rightarrow 1]| \leq Q_S \varepsilon_\mathsf{t}.$$

Game $\mathbf{G}_5$: We undo the change from $\mathbf{G}_1$ and no longer require that $(\mathsf{par}, x_1) \in \mathsf{Good}$. As before, we get

$$|\Pr[\mathbf{G}_4 \Rightarrow 1] - \Pr[\mathbf{G}_5 \Rightarrow 1]| \leq \Pr[(\mathsf{par}, x_1) \notin \mathsf{Good}] \leq \varepsilon_\mathsf{g}.$$

Game $\mathbf{G}_6$: We change how public key X_1 is generated. Before, it was generated by sampling the secret key $x_1 \xleftarrow{\$} \mathcal{D}$ and setting $X_1 := \mathsf{F}(x_1)$. Now, we sample $X_1 \xleftarrow{\$} \mathcal{R}$. Note that the secret key x_1 is not used anymore, due to the previous changes. Now, we can use the key indistinguishability of LF and get a reduction $\mathcal{B}'$ with

$$|\Pr[\mathbf{G}_5 \Rightarrow 1] - \Pr[\mathbf{G}_6 \Rightarrow 1]| \leq \mathsf{Adv}^{\mathsf{keydist}}_{\mathcal{B}',\mathsf{LF}}(\lambda).$$

Game $\mathbf{G}_7$: We use the statistical coset binding property of CMT as follows. For oracle queries of the form $\mathsf{H}_c(\tilde{\mathsf{pk}}, \mathsf{com}, \mathsf{m})$, we set $\mathsf{ck} := \mathsf{H}(\tilde{\mathsf{pk}}, \mathsf{m})$, and extract from the commitment com using the extractor Ext from the statistical coset binding property of CMT if ck has been sampled in binding mode, i.e., if $b[\tilde{\mathsf{pk}}, \mathsf{m}] = 1$. Concretely, we run $R \leftarrow \mathsf{Ext}(\mathsf{ck}, \mathsf{com})$ and store R in another map r as $r[\tilde{\mathsf{pk}}, \mathsf{com}, \mathsf{m}] := R$. We continue the simulation of H_c as before. Later, when $\mathcal{A}$ outputs its forgery $(\mathcal{P}^*, \mathsf{m}^*, \sigma^*)$ for a signature $\sigma^* = (\mathsf{com}^*, s^*, \varphi^*)$, we compute the aggregated key $\tilde{\mathsf{pk}} := \tilde{X} := \mathsf{Agg}(\mathcal{P}^*)$ and $c^* := \mathsf{H}_c(\tilde{\mathsf{pk}}, \mathsf{com}^*, \mathsf{m}^*)$ and $R^* := \mathsf{F}(s^*) - c^* \cdot \tilde{X}$ as in the verification algorithm. Then, the game outputs 0 if $R^* \notin r[\tilde{\mathsf{pk}}, \mathsf{com}^*, \mathsf{m}^*] + \mathsf{F}(\mathcal{D})$. Otherwise, it continues as before. Observe that compared to $\mathbf{G}_7$ in the proof of Theorem 1 in [PW23], we weaken this new winning condition by allowing a difference in the span of F. This is necessary as we only have statistical coset binding, and not statistical binding as in [PW23]. We will see that this is compensated by coset aggregation lossy soundness. Using a reduction as given in $\mathbf{G}_7$ in the proof of Theorem 1 in [PW23], we can argue that

$$|\Pr[\mathbf{G}_6 \Rightarrow 1] - \Pr[\mathbf{G}_7 \Rightarrow 1]| \leq Q_\mathsf{H} Q_{\mathsf{H}_c} \varepsilon_\mathsf{b}.$$

Game $\mathbf{G}_8$: We ensure that H_a and H_c are always queried in the correct order. Namely, if there is a query $\mathsf{H}_a(\langle \mathcal{P} \rangle, \mathsf{pk})$ for $\mathsf{pk} = X_1$ and the hash value is not yet defined, but for $\tilde{\mathsf{pk}} := \mathsf{Agg}(\mathcal{P})$ the hash value $\mathsf{H}_c(\tilde{\mathsf{pk}}, \mathsf{com}, \mathsf{m})$ is already defined for some com, m, then the game aborts. As in $\mathbf{G}_8$ in the proof of Theorem 1 in [PW23], we get that

$$|\Pr[\mathbf{G}_7 \Rightarrow 1] - \Pr[\mathbf{G}_8 \Rightarrow 1]| \leq \frac{1}{|\mathcal{R}|} + \frac{Q_{\mathsf{H}_a} Q_{\mathsf{H}_c}}{|\mathcal{S}|}.$$

Finally, we reduce from coset aggregation lossy soundness to finish the proof. In contrast to the final reduction in the proof of Theorem 1 in [PW23], it is important to use coset aggregation lossy soundness and not aggregation lossy soundness due to the modified change in $\mathbf{G}_7$. We get

$$\Pr[\mathbf{G}_8 \Rightarrow 1] \leq Q_{\mathsf{H}_a} Q_{\mathsf{H}_c} \varepsilon_\mathsf{al}.$$

□

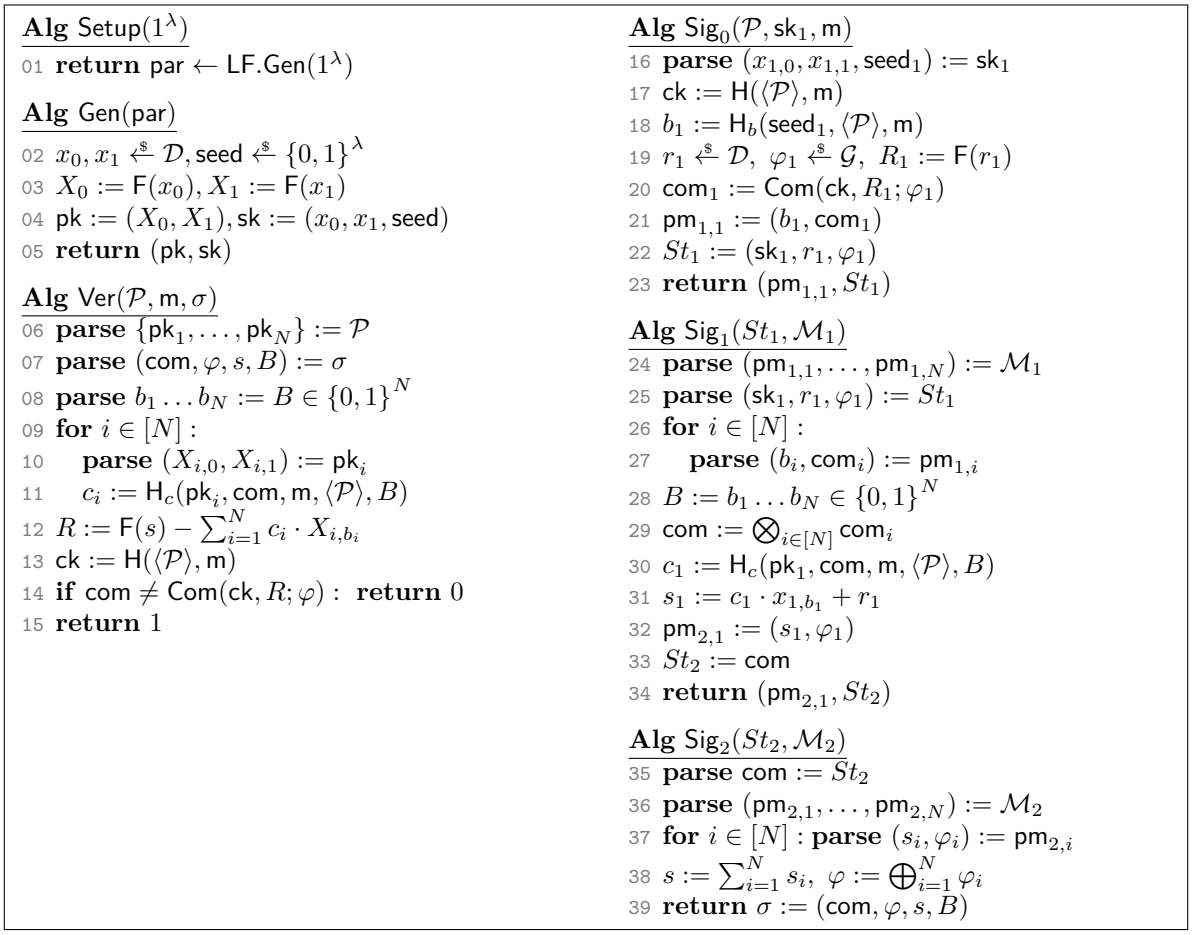

Figure 6: The multi-signature scheme $\mathsf{Tooth}[\mathsf{LF}, \mathsf{CMT}] = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ for a linear function family $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{F})$ and a weakly equivocable coset commitment scheme $\mathsf{CMT} = (\mathsf{TGen}, \mathsf{Com}, \mathsf{TCom}, \mathsf{TCol})$.

Game $\mathbf{G}_0$-$\mathbf{G}_8$

$b^* \xleftarrow{\$} \{0,1\}$ // $\mathbf{G}_2$-$\mathbf{G}_8$
$\mathsf{par} \leftarrow \mathsf{LF.Gen}(1^\lambda)$
$\mathsf{seed}_1 \xleftarrow{\$} \{0,1\}^\lambda$
$x_{1,0}, x_{1,1} \xleftarrow{\$} \mathcal{D},\ X_{1,0} := \mathsf{F}(x_{1,0}),\ X_{1,1} := \mathsf{F}(x_{1,1})$ // $\mathbf{G}_0$-$\mathbf{G}_6$
$x_{1,b^*} \xleftarrow{\$} \mathcal{D},\ X_{1,b^*} := \mathsf{F}(x_{1,b^*}),\ X_{1,1-b^*} \xleftarrow{\$} \mathcal{R}$ // $\mathbf{G}_7$-$\mathbf{G}_8$
if $(\mathsf{par}, x_{1,1-b^*}) \notin \mathsf{Good}$: **return** 0 // $\mathbf{G}_3$-$\mathbf{G}_5$
$\mathsf{pk}^* := (X_{1,0}, X_{1,1})$
$(\mathcal{P}^*, \mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{H},\mathsf{H}_b,\mathsf{H}_c,\mathrm{SIG}_0,\mathrm{SIG}_1}(\mathsf{par}, \mathsf{pk}^*)$
if $\mathsf{pk}^* \notin \mathcal{P}^* \vee (\mathcal{P}^*, \mathsf{m}^*) \in \mathsf{Queried}$: **return** 0
parse $(\mathsf{com}^*, \varphi^*, s^*, B^*) := \sigma^*,\quad b_1^* \dots b_N^* := B^* \in \{0,1\}^N$
if $\mathsf{bad} = 1$: **return** 0 // $\mathbf{G}_1$-$\mathbf{G}_8$
if $b^* = b_1^* \vee \bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P}^* \rangle, \mathsf{m}^*) = 1 - b^*$: **return** 0 // $\mathbf{G}_2$-$\mathbf{G}_8$
parse $\{\mathsf{pk}_1 = \mathsf{pk}^*, \dots, \mathsf{pk}_N\} := \mathcal{P}^*$ // $\mathbf{G}_8$
for $i \in [N]$: // $\mathbf{G}_8$
  parse $(X_{i,0}, X_{i,1}) := \mathsf{pk}_i$ // $\mathbf{G}_8$
  $c_i^* := \mathsf{H}_c(\mathsf{pk}_i, \mathsf{com}_0^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*)$ // $\mathbf{G}_8$
$R^* := \mathsf{F}(s^*) - \sum_{i=1}^N c_i^* \cdot X_{i,b_i^*}$ // $\mathbf{G}_8$
if $R^* \notin r[\mathsf{com}^*, \mathsf{m}^*, \langle \mathcal{P}^* \rangle, B^*] + \mathsf{F}(\mathcal{D})$: **return** 0 // $\mathbf{G}_8$
return $\mathsf{Ver}(\mathcal{P}^*, \mathsf{m}^*, \sigma^*)$

Figure 7: The games $\mathbf{G}_0$-$\mathbf{G}_8$ used in the proof of Theorem 1. Lines with highlighted comments are only executed in the respective games. The signing oracles are defined in Figure 8, and the random oracles are defined in Figure 9.

Oracle $\textsc{Sig}_0(\mathcal{P}, \mathsf{m})$

$\mathbf{parse}\ \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\} := \mathcal{P}$
$\mathbf{if}\ \mathsf{pk}_1 \neq \mathsf{pk}^* : \mathbf{return}\ \bot$
$\mathsf{Queried} := \mathsf{Queried} \cup \{(\mathcal{P}, \mathsf{m})\},\ ctr := ctr + 1,\ sid := ctr,\ \mathsf{round}[sid] := 1$
$\mathsf{ck} := \mathsf{H}(\langle \mathcal{P} \rangle, \mathsf{m})$
$b_1 := \bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P} \rangle, \mathsf{m})$
$r_1 \xleftarrow{\$} \mathcal{D},\ \varphi_1 \xleftarrow{\$} \mathcal{G},\ R_1 := \mathsf{F}(r_{1,b_1})$
$\mathsf{com}_1 := \mathsf{Com}(\mathsf{ck}, R_1; \varphi_1)$
$St_1 := (r_1, \varphi_1)$
$\mathbf{if}\ b_1 = 1 - b^* : \ (\mathsf{com}_1, St_1) \leftarrow \mathsf{TCom}(\mathsf{ck}, tr[\langle \mathcal{P} \rangle, \mathsf{m}])$ // $\mathbf{G}_5$-$\mathbf{G}_8$
$(\mathsf{pm}_1[sid], St_1[sid]) := (\mathsf{pm}_{1,1} := (b_1, \mathsf{com}_1), St_1)$
$\mathbf{return}\ (\mathsf{pm}_1[sid], sid)$

Oracle $\textsc{Sig}_1(sid, \mathcal{M}_1)$

$\mathbf{if}\ \mathsf{round}[sid] \neq 1 : \mathbf{return}\ \bot$
$\mathbf{parse}\ (\mathsf{pm}_{1,1}, \ldots, \mathsf{pm}_{1,N}) := \mathcal{M}_1$
$\mathbf{if}\ \mathsf{pm}_1[sid] \neq \mathsf{pm}_{1,1} : \mathbf{return}\ \bot$
$\mathsf{round}[sid] := \mathsf{round}[sid] + 1,\ St_1 := St_1[sid]$
$\mathbf{parse}\ (r_1, \varphi_1) := St_1$ // $\mathbf{G}_0$-$\mathbf{G}_4$
$\mathbf{for}\ i \in [N] : \mathbf{parse}\ (b_i, \mathsf{com}_i) := \mathsf{pm}_{1,i}$
$B := b_1 \ldots b_N \in \{0,1\}^N$
$\mathsf{com} := \bigotimes_{i \in [N]} \mathsf{com}_i$
$c_1 := \mathsf{H}_c(\mathsf{pk}_1, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B)$
$s_1 := c_1 \cdot x_{1,b_1} + r_1$ // $\mathbf{G}_0$-$\mathbf{G}_4$
$\mathbf{if}\ b_1 = 1 - b^* : \ (\varphi_1, R_1, s_1) \leftarrow \mathsf{TCol}(St_1, c_1)$ // $\mathbf{G}_5$-$\mathbf{G}_8$
$\mathbf{if}\ b_1 = b^* : \ s_1 := c_1 \cdot x_{1,b_1} + r_1$ // $\mathbf{G}_5$-$\mathbf{G}_8$
$St_2 := \mathsf{com}$
$(\mathsf{pm}_2[sid], St_2[sid]) := (\mathsf{pm}_{2,1} := (s_1, \varphi_1), St_2)$
$\mathbf{return}\ \mathsf{pm}_2[sid]$

Figure 8: The signing oracles that are used in the proof of Theorem 1. Lines with highlighted comments are only executed in the respective games.

Oracle $\mathsf{H}(\langle \mathcal{P} \rangle, \mathsf{m})$ // $\mathbf{G}_0$-$\mathbf{G}_3$

$\mathbf{if}\ h[\langle \mathcal{P} \rangle, \mathsf{m}] = \bot :$
  $h[\langle \mathcal{P} \rangle, \mathsf{m}] \xleftarrow{\$} \mathcal{K}$
$\mathbf{return}\ h[\langle \mathcal{P} \rangle, \mathsf{m}]$

Oracle $\mathsf{H}(\langle \mathcal{P} \rangle, \mathsf{m})$ // $\mathbf{G}_4$-$\mathbf{G}_8$

$\mathbf{if}\ h[\langle \mathcal{P} \rangle, \mathsf{m}] = \bot :$
  $b := \bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P} \rangle, \mathsf{m})$
  $\mathbf{if}\ b = 1 - b^* :$
    $(\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{TGen}(\mathsf{par}, X_{1,1-b^*})$
    $tr[\langle \mathcal{P} \rangle, \mathsf{m}] := \mathsf{td}$
  $\mathbf{if}\ b = b^* :$
    $\mathsf{ck} \leftarrow \mathsf{BGen}(\mathsf{par})$
  $h[\langle \mathcal{P} \rangle, \mathsf{m}] := \mathsf{ck}$
$\mathbf{return}\ h[\langle \mathcal{P} \rangle, \mathsf{m}]$

Oracle $\bar{\mathsf{H}}_b(\mathsf{seed}, \langle \mathcal{P} \rangle, \mathsf{m})$

$\mathbf{if}\ \bar{h}_b[\mathsf{seed}, \langle \mathcal{P} \rangle, \mathsf{m}] = \bot :$
  $\bar{h}_b[\mathsf{seed}, \langle \mathcal{P} \rangle, \mathsf{m}] \xleftarrow{\$} \{0,1\}$
$\mathbf{return}\ \bar{h}_b[\mathsf{seed}, \langle \mathcal{P} \rangle, \mathsf{m}]$

Oracle $\mathsf{H}_c(\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B)$ // $\mathbf{G}_0$-$\mathbf{G}_7$

$\mathbf{if}\ h_c[\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] = \bot :$
  $h_c[\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] \xleftarrow{\$} \mathcal{S}$
$\mathbf{return}\ h_c[\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B]$

Oracle $\mathsf{H}_c(\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B)$ // $\mathbf{G}_8$

$\mathbf{if}\ h_c[\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] = \bot :$
  $\mathbf{parse}\ \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\} := \mathcal{P}$
  $\mathbf{parse}\ b_1 \ldots b_N := B$
  $b := \bar{\mathsf{H}}_b(\mathsf{seed}_1, \langle \mathcal{P} \rangle, \mathsf{m})$
  $\mathbf{if}\ \mathsf{pk} = \mathsf{pk}^* = \mathsf{pk}_1 \wedge b = b^* :$
    $R \leftarrow \mathsf{Ext}(\mathsf{H}(\langle \mathcal{P} \rangle, \mathsf{m}), \mathsf{com})$
    $r[\mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] := R$
  $h_c[\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B] \xleftarrow{\$} \mathcal{S}$
$\mathbf{return}\ h_c[\mathsf{pk}, \mathsf{com}, \mathsf{m}, \langle \mathcal{P} \rangle, B]$

Oracle $\mathsf{H}_b(\mathsf{seed}, \langle \mathcal{P} \rangle, \mathsf{m})$

$\mathbf{if}\ \mathsf{seed} = \mathsf{seed}_1 : \ \mathsf{bad} := 1$ // $\mathbf{G}_1$-$\mathbf{G}_8$
$\mathbf{return}\ \bar{\mathsf{H}}_b(\mathsf{seed}, \langle \mathcal{P} \rangle, \mathsf{m})$

Figure 9: The random oracles that are used in the proof of Theorem 1. Lines with highlighted comments are only executed in the respective games. Algorithm Ext is the (unbounded) extractor for the statistical coset binding property of CMT.

C Scripts for Parameter Computation

Listing 1: Python Script to compute concrete security levels and efficiency of two-round multi-signatures. More explanation is given in Section 6.

```
#!/usr/bin/env python



################PURPOSE#OF#THIS#SCRIPT##################
# For each scheme, we estimate the security level      #
# that is guaranteed by the security bound.            #
# We also compute the concrete sizes of public keys,   #
# signatures, and communication complexity per signer. #
# We assume a certain number of hash and signing       #
# queries; we assume that secp256k1 is used and the    #
# underlying assumptions/problems offer a security     #
# level of 128 bit.                                    #
########################################################

import math
from tabulate import tabulate



#number of hash queries and signing queries
log_q_h = 30
log_q_s = 20

#hardness of the underlying assumption
kappa = 128

#sizes of group elements, exponents, and statistical security parameter+
#we assume secp256k1 and all sizes in bits
secpar = 128
sizege = 33*8
sizefe = 256


#################Define Schemes#######################################
# Note: we estimate an upper bound on epsilon, assuming unit time    #
# One can see that this favors schemes with rewinding due to the sqrt#
########################################################################


musigtwo = {
        "name": "Musig2",
        "level": 0.25 * (kappa-2-3*log_q_h),
        "pk": sizege,
        "comm": 4*sizege+sizefe,
        "sig": sizege+sizefe,
}

hbms = {
        "name": "HBMS",
        "level": 0.25 * (kappa-2-3*log_q_h - 4*log_q_s),
        "pk": sizege,
        "comm": sizege+2*sizefe,
        "sig": sizege+2*sizefe,
}

tz = {
        "name": "TZ",
        "level": 0.25 * (kappa-3-3*log_q_h),
        "pk": sizege,
        "comm": 4*sizege+2*sizefe,
        "sig": sizege+2*sizefe,
}

tssho = {
        "name": "TSSHO",
        "level": kappa - 2 - log_q_s,
        "pk": 2*sizege,
        "comm": 2*sizege+2*sizefe,
        "sig": 3*sizefe,
}

chopsticksone = {
        "name": "Chopsticks 1",
        "level": kappa - 2 - log_q_s,
        "pk": 2*sizege,
        "comm": 3*sizege+1*sizefe+secpar,
        "sig": 3*sizege+4*sizefe,
}

chopstickstwo = {
        "name": "Chopsticks 2",
        "level": kappa - 2,
        "pk": 4*sizege,
        "comm": 6*sizege+2*sizefe+secpar+1,
        "sig": 6*sizege+8*sizefe+secpar, #assuming number of signers <= secpar
}

toothone = {
        "name": "Toothpicks 1",
        "level": kappa - 2 - log_q_s,
        "pk": 2*sizege,
        "comm": 2*sizege+1*sizefe+secpar,
        "sig": 3*sizefe+2*secpar,
}

toothtwo = {
        "name": "Toothpicks 2",
        "level": kappa - 3,
        "pk": 4*sizege,
        "comm": 2*sizege+1*sizefe+secpar+1,
        "sig": 3*sizefe+2*secpar+secpar, #assuming number of signers <= secpar
```

```
}

schemes = [musigtwo,hbms,tz,tssho,chopsticksone,chopstickstwo,toothone,toothtwo]



#####################Main Part########################

def bytes(x):
        return int(round(x/8.0,0))

data = [["Scheme", "Security Level", "Pk", "Communicaton", "Signature"]]

for s in schemes:
        data.append([s["name"],int(s["level"]),bytes(s["pk"]),bytes(s["comm"]),bytes(s["sig"])])


#print(tabulate(data,headers='firstrow',tablefmt='fancy_grid'))
#print(tabulate(data,headers='firstrow'))
print(tabulate(data,headers='firstrow',tablefmt='latex_raw',disable_numparse=True))
```