Tightly-Secure Blind Signatures in Pairing-Free Groups

Nicholas Brandt, Dennis Hofheinz, Michael Klooß, and Michael Reichle

Department of Computer Science
ETH Zurich, Zurich, Switzerland
{nicholas.brandt,hofheinz, michael.klooss, michael.reichle}@inf.ethz.ch

Abstract. We construct the first blind signature scheme that achieves all of the following properties simultaneously:

- it is tightly secure under a standard (i.e., non-interactive, non-q-type) computational assumption,
- it does not require pairings,
- it does not rely on generic, non-black-box techniques (like generic NIZK proofs).

The third property enables a reasonably efficient solution, and in fact signatures in our scheme comprise 10 group elements and 29 $\mathbb{Z}_p$-elements. Our scheme starts from a pairing-based non-blind signature scheme (Abe et al., JoC 2023), and uses recent techniques of Chairattana-Apirom, Tessaro, and Zhu (CRYPTO 2024) to replace the pairings used in this scheme with non-interactive zero-knowledge proofs in the random oracle model. This conversion is not generic or straightforward (also because the mentioned previous works have converted only significantly simpler signature schemes), and we are required to improve upon and innovate existing techniques in several places.

As an interesting side note, and unlike previous works, our techniques only require a *non-programmable* random oracle, and our signature scheme achieves predicate blindness (which means that the user can prove statements about the signed message during the signing process).

Keywords. blind signatures, tight security, group-based cryptography.

1 Introduction

Blind signatures. Digital signatures allow a designated signer to *sign* messages in a way that the resulting signatures can be efficiently *verified* to be valid (for the corresponding message and relative to a signer-specific public key). For security, we require that *only* the signer can produce valid signatures, using a dedicated secret key. Digital signatures are one of the core cryptographic building blocks, and have a rich history with countless applications.

In this work, we are concerned with a variant of digital signatures, "blind signatures", in which the signer does not learn the signed message. This requirement immediately implies that signature generation must now be an interactive process that the signer and a user (who wants a particular message signed)

engage in. Originally, blind signatures were proposed as a technical tool to realize electronic cash [16]. Much later, a systematic investigation of this building block has started, with generic [42, 24] and direct [55, 13, 10, 53, 30, 29, 26, 32] constructions, as well as lower bounds and impossibility results [25, 54, 8]. Most of these "first-generation" blind signature schemes are however either very inefficient (e.g., because they rely on non-interactive zero-knowledge (NIZK) proofs for complex languages), they rely on somewhat nonstandard assumptions (such as interactive or even "one-more-type" assumptions), or they only offer a limited form of security (e.g., for a bounded number of issued signatures).

Recent constructions of blind signatures. More recently, a number of efficient blind signature schemes have been proposed in the "algebraic group model" (AGM, see [27]), e.g., [44, 57, 20, 28]. The AGM is a model of computation that lies in between the standard model and the generic group model (GGM, see [56, 52]), and it captures a very strong form of knowledge assumption. Another recent line of work trades (some) efficiency for security under a weaker assumption, and uses the cut-and-choose paradigm [35, 15] (building on prior works [36, 37, 43, 47]). Yet another line of work instantiates the generic NIZK-based construction of Fischlin [24] in algebraic ways (e.g., using lattices [21], pairings [46], or the strong RSA assumption [45]).

Tight security reductions. The other concept that is important for our work is the concept of tight security reductions. In cryptography, we are always interested in security *proofs*, which for many building blocks (including signatures and blind signatures) means security *reductions*. A security reduction argues that a certain cryptographic scheme S is secure by mapping every (hypothetical) successful adversary $\mathcal{A}$ on that scheme to a successful problem solver, i.e., an algorithm $\mathcal{B}$ that solves an assumed-to-be-hard computational problem P. Then, since P is assumed to be hard, no successful $\mathcal{B}$, and hence, no successful $\mathcal{A}$ can exist (and S is secure).

In typical reductions, however, $\mathcal{B}$ will be less effective than $\mathcal{A}$, in the sense that $\mathcal{B}$ success in solving P will be smaller than $\mathcal{A}$'s success in breaking S. Sometimes, also $\mathcal{B}$'s runtime will be greater than $\mathcal{A}$'s. In both cases, we say that the reduction is *non-tight*, since it may now be (quantitatively) easier to break S than to solve P. If we derive concrete keylength recommendations for a scheme S from the best known attacks on the underlying problem P, this means that we may need to increase keylengths to account for a non-tight reduction. This becomes particularly problematic in cases in which the reduction loss (i.e., degree of non-tightness) depends on the number of users or uses of the scheme. Indeed, in such cases, we may end up deploying a scheme (with concrete parameters) that becomes less and less secure the more popular it becomes.

Conversely, a *tight* reduction (in which $\mathcal{B}$'s success and runtime are close to $\mathcal{A}$'s) is preferable and leads to more efficient parameters. Tight reductions have been studied for many cryptographic building blocks like public-key encryption [9, 40, 7], digital signatures [18, 40, 12], or (interactive or non-interactive) key exchange [6, 7, 38].

Tightly secure blind signatures. The focus of this paper is a combination of the two above concepts: a blind signature scheme with a tight security reduction. Interestingly, there are few known examples of *tightly secure blind* signatures. Of course, it is always possible to implement a generic construction of blind signatures with tightly secure components. For instance, it is possible to implement Fischlin's generic construction [24] with a group-based NIZK proof system (such as the Groth-Sahai proof system [33]) and a tightly secure structure-preserving signature scheme [2, 31, 4, 3]. This leads to tightly secure blind signatures whose security relies on standard assumptions, but it requires pairings and is still somewhat inefficient.[1]

Our goal. In this work, we are interested in a blind signature scheme with the following two properties:

1. The scheme should be tightly secure under a standard (i.e., non-interactive and non-q-type) computational assumption.
2. The scheme should not require pairings.

Additionally, we would like our scheme to be as efficient as possible, both in signature size and computation times. In particular, we are not interested in generic constructions, such as the one obtained by implementing Fischlin's scheme [24] with generic NIZKs [28].

1.1 Our Contribution

We achieve our goal by combining two recent technical strategies. The first one, due to Chairattana-Apirom, Tessaro, and Zhu (CTZ [15]), starts from a blind signature scheme which uses pairings only for equality tests during verification. CTZ then replaces these pairing checks with NIZKs. (This is already technically quite delicate even for simple blind signatures used as a basis, since we cannot prove statements about a potentially used hash function in zero-knowledge.) The second strategy is to implement this CTZ strategy with a new (non-blind) tightly secure signature scheme based on the "adaptive partitioning" method [39, 2]. There are a number of technical obstacles (e.g., achieving blindness "along the way"), which we will detail in our technical overview below.

The resulting blind signature scheme relies on the Decisional Diffie-Hellman (DDH) assumption in pairing-free groups, but uses the Fiat-Shamir methodology [23] for the implicit NIZKs, and hence its analysis uses the random oracle model. Interestingly, however, we only require a *non-programmable* random oracle for most of our analysis.[2]

[1] For instance, [4] calculate that such a highly optimized blind signature scheme following from their structure-preserving signatures would have signatures comprised of 82 group elements.

[2] More specifically, our reduction only needs to program the (random) global parameters of our scheme. Hence, our scheme can be formulated in the common random string model (with trusted random parameters) with a completely non-programmable random oracle. Alternatively, the random oracle can be used to generate and program

Table 1: Comparison of Blind Signatures in the ROM secure under standard assumptions

Reference	Signature size	Communication size	Assumption	Advantage Bound
dK22 [21]	100 KB	850 KB	DSMR, MLWE, MSIS	$\mathsf{poly}(Q_M, Q_H) \cdot \epsilon^{1/\theta(1)}$
BFPV13 [11]	96 B	220 KB †	SXDH, CDH	$\mathcal{O}(Q \cdot \sqrt{\lambda}) \cdot \epsilon$ (cf. [58, 41])
AJOR18 [4]	5.5 KB	1 KB	SXDH	$\mathcal{O}(\log Q_S) \cdot \epsilon$
HLW23 [35]‡	5 KB 9 KB	72 KB 36 KB	CDH	$\mathcal{O}(Q_S)$
KRS23 [46, $\mathsf{BS}_{\mathsf{rnd}}$]	447 B	303 B	SXDH	$\mathcal{O}(Q_S^2) \cdot \epsilon$
KRS23 [46, $\mathsf{BS}_{\mathsf{bb}}$]	96 B	2.2 KB	DDH, CDH	$\mathcal{O}(Q_H) \cdot \epsilon$
CTZ24 [15, BS_3]	27.1 KB	10.6 KB	CDH	$\mathcal{O}(Q_S) \cdot \sqrt{Q_H \cdot \epsilon}$
KRW24 [50]	224 B	2.5 KB	DDH	$\mathcal{O}(Q_H) \cdot \epsilon$
KR24 [49]	192 B	608 B	DDH	$\mathcal{O}(Q_H) \cdot \epsilon$
Our scheme	1.3 KB	2.7 KB	DDH	$\mathcal{O}(\log Q_S) \cdot \epsilon$

We provide an overview of blind signatures that achieve full one-more unforgeability proven under standard assumptions (in the ROM). The first, second and third section depicts schemes based on lattices, pairings and pairing-free groups, respectively. Above, Q_S denotes the number of signing sessions, Q_H the number of random oracle queries and ϵ the advantage of the reduction. Note that the size for group based schemes ignores the reduction loss (i.e., we assume standard groups for security level $\lambda = 256$). A tight loss is highlighted in green.
(†): Communication of [11] scales linearly with the message size, and is given here for 256 bit messages.
(‡): [35] offers tradeoffs between signature and communication sizes.

Our scheme requires 41 (resp. 83) group and $\mathbb{Z}_p$ elements in signatures (resp. communication), and the signing process consists of four moves.[3] See Table 1 for comparisons with other blind signatures proven under standard assumptions in the ROM. We note that, as the to-be-signed message M is not hashed in our construction, the user can prove statements over M in the signing process (i.e., it achieves predicate blindness [28]).

1.2 Technical Overview

On our use of groups. Our blind signature scheme will be using (pairing-free) cyclic groups. This is a natural design choice, since the random self-reducibility of popular computational problems in cyclic groups is a very helpful property in achieving tight security. There are of course other examples of tightly secure cryptographic primitives, e.g., in the lattice setting [14]. However, current lattice-based strategies to achieve *blind* signatures do not seem particularly amenable to tight reductions (e.g., [21] use trapdoor sampling, while [5] rely on an interactive, one-more-type assumption).

First attempt: instantiate Fischlin's generic approach with tightly secure primitives. As explained above, it *is* possible to instantiate Fischlin's [24] generic blind signature scheme with tightly secure primitives (i.e., with tightly secure signatures and NIZK proofs). When trying to find a suitable (standard) signature scheme as a starting point, we are facing a dilemma, however. Namely, known (group-based and at least "somewhat efficient") tightly secure signature schemes either use pairings (e.g., [2, 31, 3]) or random oracles (e.g., [48, 1]). Using

these parameters as hash values, and we can formulate a parameter-free version of our scheme in the programmable random oracle model. See Remark 2.

[3] We note that concurrent work [49] also achieves constant communication.

pairings violates our goals, and using random oracles does not mesh well with Fischlin's approach.[4] Hence, we need to choose another strategy.

Second attempt: directly modify an existing scheme. Next, we can revisit existing pairing-free blind signature schemes, and try to modify them so that they become tightly secure. We discuss a few representative options:

- The recent scheme of Kastner, Nguyen, and Reichle [45] indeed implements a suitable variant of Fischlin's generic approach, but with an RSA-based signature scheme (that in itself requires neither pairings nor random oracles). The inherent use of the (strong) RSA assumption, however, makes tight security proofs very difficult.
- The recent CTZ [15] work in fact offers several pairing-free blind signature schemes in the random oracle model. Their BS_1 and BS_2 schemes require an interactive, one-more-type assumption, and it is not clear how to avoid this interactivity. (Besides, BS_1 and BS_2 achieve only a relately weak form of unforgeability.) At the cost of additional complexity, their BS_3 scheme (which in turn builds upon [35]) achieves stronger unforgeability without pairings, based on the relatively mild Computational Diffie-Hellman (CDH) assumption. Their proof, however, suffers from a large reduction loss due to the use of rewindings (as inherited from the Fiat-Shamir paradigm). Of course, it seems plausible that, say, a straight-line-extractable NIZK proof system could avoid this loss. However, both BS_3 additionally employs a seemingly inherent guessing argument inherited from the cut-and-choose approach of [35].
- Klooß, Reichle, and Wagner [50] give an improvement of BS_3 based on the pairing-based blind signature in [46]. Their scheme avoids rewindings by not relying on an explicit extraction of a forged underlying signature. However, their scheme still suffers from a reduction loss due to its reliance on a puncturing technique (that in turn requires guessing which unfinished session corresponds to a particular type of forgery). Also, concurrent work [49] builds on [50] to improve efficiency, but still relies on the puncturing technique, and in turn has a noticeable reduction loss.

Conversely, we can of course also try to start with an existing non-blind (but tightly secure) signature scheme such as the Katz-Wang scheme [48], and attempt to "add" blindness. However, "adding blindness" is a delicate process in general, since there must be a way to hide (or "blind") signatures across messages.

To describe what goes wrong in case of Katz-Wang signatures, recall that this scheme employs an additional bit b appended randomly to each message before signing. The reduction will set up things such that for every message m, it can either sign $m||0$ or $m||1$, but not both. This way, every message can be signed for an adversary $\mathcal{A}$, but with probability $1/2$, $\mathcal{A}$'s final forgery is *not* already known to the reduction. Hence, the reduction loses only a factor of 2. However, turning

[4] In the blind signature scheme from [24], signatures consist of a NIZK proof of knowledge of a valid signature of the underlying scheme. If the underlying scheme uses random oracles, this proof becomes problematic.

such a scheme and proof strategy into a *blind* signature runs into the problem that the reduction's ability to generate signatures is now closely tied to the full message $m||b$ itself. Since this message is hashed using a random oracle, it seems difficult to "forget" messages at any point during the reduction.

Our strategy: combine CTZ approach with adaptive partitioning. For our purposes, we will use certain ideas from CTZ [15, 50], but avoid the use of cut-and-choose or guessing techniques. Specifically, CTZ's approach can be thought of as proceeding in two phases: (1) start with a concrete pairing-based blind signature scheme that however uses pairings only during verification; (2) replace the pairings during verification with NIZK proofs of equality of suitable quadratic equations over the source group of the pairing. Such NIZK proof systems can be constructed using the Fiat-Shamir paradigm in the random oracle model.

This high-level strategy requires careful rerandomization steps at several points. For instance, blindness requires that a signer cannot link the signatures obtained by the user to a particular blind signing session. To achieve blindness, we need to be able to "blind" the low-level NIZK proofs for group equalities, *as well as* the corresponding proved equations. Fortunately (and as recognized and used already by [15, 50]), popular Schnorr-based proof systems have the necessary blindability.

The tightly secure signature of Abe et al. Next, we need to identify a suitable (pairing-based) blind signature scheme. Our starting point will be the tightly secure structure-preserving signature scheme of Abe et al. [2]. This scheme is very modular and structure-preserving, which means that it only relies on suitably algebraic operations. It uses the "adaptive partitioning" technique, which results in relatively complex (albeit tight) reductions. More specifically, the scheme of [2] has the following properties:

- Signatures consist of an ElGamal-style encryption of the secret key, along with a "consistency" proof that proves that encrypted values satisfy certain constraints. The secret contains some redundancy, so that the corresponding constraints are nontrivial. For instance, one constraint is an *OR* of two linear constraints, and the other constraint is a linear one that involves the signed message (but such that this constraint holds for all messages).
- Verification merely checks the corresponding proofs.

All verification constraints hold initially, but the unforgeability proof carefully plays with the redundancy in the secret key and sometimes violates (some of) these constraints. Soundness of the proof systems ensures that the adversary's forgery can still be held to the constraints that are not violated. The overall goal of this procedure is to partially randomize the values encrypted in signatures, so that finally, every signature is generated by (and the forgery is verified relative to) a fresh and independently random secret key, generated freshly for each message. At this point, the linear constraint which involves the signed message can be used to argue unforgeability based on the *one-time* security of that key.

Blindly issuing Abe et al. [2] signatures. Before invoking the CTZ strategy to achieve pairing-free blind signatures, we first have to turn the scheme of Abe et al. sketched above into a *blind* signature scheme. We do not directly follow Fischlin's approach [24], since this would lead to signatures that contain nested NIZK proofs.[5] Instead, like many previous works (e.g., [17, 13]), we design a "blind issuing protocol" between a user U and a signer S that allows U to obtain a signature σ for a message M without revealing it to S. The signature σ will have essentially the same structure as the one from Abe et al.'s scheme, but will contain additional proof parts that help replace pairing checks during verification.

As a first step in our blind issuing protocol, U will send *an encryption* ct_M of M to S. (Of course, U cannot send M directly, since this would violate blindness.) Already here, we deviate from previous works (including CTZ), which use an additive blinding of M or (usually unconditionally hiding) commitment instead. Looking ahead, ct_M facilitates extraction of the underlying message in our unforgeability proof. Then, S issues the group elements comprising the signature and the proofs required for verification following the CTZ strategy. As S proves statements over the message M, the proof cannot be computed in plain (as S does not know M). Instead, we can issue the desired proofs by homomorphically evaluating the Schnorr-prover over the encryption ct_M of M. As Schnorr-based proofs are linear, a simple linear homomorphic scheme (e.g., ElGamal) is sufficient. Finally, the group elements and Schnorr-based proofs are blinded by U.

To adapt the tight security proof of [2] to our blind signature, we need to very carefully manage information leaked about signatures and secret keys from *unfinished* signing sessions. To explain: one difficulty that previous works (in particular CTZ) faced, was achieving unforgeability in a setting in which a malicious user could abort arbitrarily many signing sessions prematurely. The difficulty that arises here is to prevent such a user from learning "half-finished signatures" that could help build a full signature. CTZ solved this issue by using a tailor-made commitment scheme with "special equivocation". This approach is not compatible with the security proof of [2], as their argument relies on *knowledge* soundness of the Schnorr-based proof. As CTZ's commitment is only computationally binding, arguing soundness requires rewinding which leads to a very loose security bound. Similarly, the strategy of [50, 49] relies on a guessing argument to randomize aborting sessions which we cannot afford either.

Our solution is to use *homomorphic dual-mode* commitments, which lets the user operate on the commitments, while keeping the committed values perfectly hidden. This ensures that unfinished sessions leak nothing. In its second message, the signer will send openings of the commitment to the user, and the signatures include rerandomized openings.

This additional commitment step does not impede the blinding operations of the user, because we can blind the committed values homomorphically over the

[5] Nested proofs are not problematic in the pairing setting, e.g., with Groth-Sahai proofs [33], but would make our life much harder when converting verification to a pairing-free setting.

commitments. Combining the above techniques, we obtain our blind signature. For more details on the security analysis, we refer to the main body.

1.3 Organization of this Paper

In Section 2, we provide the relevant cryptographic definitions (auxiliary preliminaries are given in Supplementary Material C). In Section 3, we sketch the pairing-free signature scheme that underlies our construction. In Section 3.2 we provide the concrete protocols of our scheme; followed by the security statements in Section 3.3. Formal proofs are given in Supplementary Material D.

We give a concrete instantiation (including a conrete efficiency overview) of our generic construction in Supplementary Material B.

2 Preliminaries

Let $\lambda \in \mathbb{N}$ be the security parameter. We use standard notations for probability, algorithms and distributions.[6] Throughout, we assume that any space is efficiently sampleable. We write $A(\mathsf{in}_A) \longleftrightarrow B(\mathsf{in}_B)$ for interactive protocols between parties A and B with input in_A and in_B, respectively. Within algorithmic descriptions, we denote by **req** C that the algorithm outputs $\perp$ if the condition C is false. When describing games, we denote by **abort if** C that the game outputs 0 if the condition C is false. Throughout, we denote by $\mathbb{G}$ a group of prime order p with generator $G \in \mathbb{G}$. We generally use additive notation for $\mathbb{G}$. Throughout, group elements G are capital, whereas elements x in $\mathbb{N}$ or $\mathbb{Z}_p$ are lowercase.

Assumptions. Although we assume many properties to hold perfectly for exposition, it is straightforward to relax them to be statistical (some even to be computational).

As is common, the group $\mathbb{G}$ should be understood as implicitly being a family of groups, i.e., $\mathbb{G} = \mathbb{G}_\lambda$ is implicitly parameterized by the security parameter λ. We briefly recall the DL, CDH and (Q-)DDH assumptions and refer to Supplementary Material A for formal definitions. Let a, b, c be uniformly random in $\mathbb{Z}_p$. The DL assumption states that given (G, aG) it is hard to compute a. The CDH assumption states that it is hard given (G, aG, bB) to compute $(ab)G$. The DDH assumption states that it is hard to distinguish a real Diffie-Hellman tuple $(G, aG, bB, (ab)G)$ from a random tuple (G, aG, bB, cG). The Q-DDH assumption states that it is hard to distinguish Q random Diffie-Hellman tuples from random Q tuples; it tightly reduces to DDH.

Random Oracle Model. By H, we always denote a random oracle. We sometimes leave the domain (usually $\{0,1\}^*$) and range (often challenge set $\mathcal{CH} = \mathbb{Z}_p$) of a random oracle unspecified when it is clear from the context. We call an algorithm Q_H-bounded, if it makes at most Q_H queries to H, where $\mathsf{Q}_\mathsf{H} = \mathsf{Q}_\mathsf{H}(\lambda)$.

[6] We use $x \coloneqq v$ for assignment of value v to x (and $x \leftarrow v$ if x is updated with value v), $x \leftarrow A(\mathsf{in})$ for (probabilistic) algorithms A on input in, and $x \leftarrow \mathcal{D}$ for sampling from distribution $\mathcal{D}$. (If $\mathcal{D}$ is a set, this denotes sampling from $\mathcal{D}$ uniformly and independently at random).

2.1 Blind Signatures

We define the primitive of interest, namely, blind signatures [16].

Definition 1 (Blind Signature Scheme). *A blind signature scheme with message space $\mathcal{M}$ in the ROM (with random oracle H) is a quadruple of PPT algorithms $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{S}, \mathsf{U}, \mathsf{Verify})$ with the following syntax:*

- $\mathsf{KeyGen}(1^\lambda)$*: outputs a pair of keys* $(\mathsf{vk}, \mathsf{sk})$*. We assume that* sk *includes* vk *implicitly.*
- $\mathsf{S}(\mathsf{sk}) \longleftrightarrow \mathsf{U}(\mathsf{vk}, m)$*:* S *takes as input a secret key* sk*.* U *takes as input a key* vk*, a message* $m \in \mathcal{M}$*. Both* S *and* U *have access to* H*. After the execution, the user* U *returns a signature* $\sigma \leftarrow \langle \mathsf{S}(\mathsf{sk}), \mathsf{U}(\mathsf{vk}, m) \rangle$ *(or* $\perp$*).*
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$ *is deterministic and takes as input public key* vk*, message* $m \in \mathcal{M}$*, and a signature* σ*, and outputs* $b \in \{0, 1\}$*.*

Due to space constraints, we provide the formal properties of blind signature schemes in Supplementary Material C.1. We give a brief inituition of the two security notions:

Unforgeability. Intuitively, a blind signature scheme should not allow any user to obtain signatures without interacting with the signer. This is modeled by the notion of one-more unforgeability, which states that after completing $\ell - 1$ signing sessions, an adversary can not output valid signatures on ℓ messages.

Blindness. To protect the privacy of users, blind signatures should satisfy blindness. Intuitively, blindness states that a malicious signer cannot link signing interactions to the message-signature pairs. We emphasize that we consider the malicious signer blindness, i.e., the malicious signer can freely choose the public key and arbitrarily deviate from the protocol.

2.2 Preimage Relations and Blindable Σ-protocols

In this section we define (blindable) Σ-protocols for a specific type of (linear) relations.

Linear relations Next, we define NP-relations and Σ-protocols for NP-relations. A preimage relation is a special type of NP-relation.

Definition 2 (NP-Relation and Language). *Let $\mathcal{X}$ be a statement space and $\mathcal{W}$ be a witness space. Let $\mathsf{R} \subseteq \mathcal{X} \times \mathcal{W}$ be a binary relation. We say that* R *is an* NP-relation*, if there exists a polynomial p such that* R *can efficiently be decided and for every $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$, we have $|\mathbb{w}| \leq p(|\mathbb{x}|)$. We denote by $\mathscr{L}_\mathsf{R} = \{\mathbb{x} \in \{0,1\}^* \mid \exists w \text{ s.t. } (\mathbb{x}, \mathbb{w}) \in \mathsf{R}\}$ the language induced by* R*.*

To model preimage relations, we consider a statement $\mathbb{x}$ and a pair of functions (ϕ, ξ) such that $\exists \mathbb{w} \colon \phi_\mathbb{x}(\mathbb{w}) = \xi(\mathbb{x})$. That is, the statement $\mathbb{x}$ specifies a function $\phi_\mathbb{x}$ and a target value $\xi(\mathbb{x})$, and the NP-relation asserts the existence of a preimage. Usually, the statment has the form $\mathbb{x} = (x, y)$, such that $\phi_\mathbb{x} = \phi_x$ and $\xi(\mathbb{x}) = y$.

Definition 3 (Linear Preimage Relation). *Let*

$$\phi\colon \mathcal{X} \times \mathcal{W} \to \mathcal{COM} \quad \textit{and} \quad \xi\colon \mathcal{X} \to \mathcal{COM}$$

be efficiently computable functions, where we write $\phi_{\mathbb{x}}(\mathbb{w}) \coloneqq \phi(\mathbb{x}, \mathbb{w})$ *for convenience. Define the NP-relation* $\mathsf{R}_{(\phi,\xi)} \subseteq \mathcal{X} \times \mathcal{W}$ *associated to the pair* (ϕ, ξ) *as*

$$\mathsf{R}_{(\phi,\xi)} = \{(\mathbb{x}, \mathbb{w}) \in \mathcal{X} \times \mathcal{W} \mid \phi_{\mathbb{x}}(\mathbb{w}) = \xi(\mathbb{x})\}.$$

We call such an $\mathsf{R}_{(\phi,\xi)}$ *a* preimage relation (for the pair (ϕ, ξ)).

Suppose $\mathcal{W}$ *and* $\mathcal{COM}$ *are* $\mathbb{Z}_p$*-vector spaces, and for every* $\mathbb{x}$ *the map* $\phi_{\mathbb{x}} : \mathcal{W} \to \mathcal{COM}$ *is* $\mathbb{Z}_p$*-linear. Then we call* $\mathsf{R}_{(\phi,\xi)}$ *a* linear relation *for short.*

All of our relations in Section 3 will be linear, and thus, there are canonical Σ-protocols (Definition 8) for proving them.

(Canonical) Blindable Σ-protocols Compared to the usual definition of Σ-protocol, we introduce a Setup (resp. $\mathsf{BlindSetup}$) algorithm, which intuitively samples the randomness. As a consequence, all other algorithms are deterministic (given the state), which will be convenient in our protocols and proofs. For consistency, we use "starred" variables to denote non-blind messages (i.e., variables which will be sent to or known by the signer).

Definition 4 (Blindable Σ-protocol). *Let* $\mathsf{R}_{(\phi,\xi)}$ *be an linear relation. A* Σ*-protocol for* $\mathsf{R}_{(\phi,\xi)}$ *with commitment space* $\mathcal{COM}$*, challenge space* $\mathcal{CH}$ *and response space* $\mathcal{RESP}$ *is a tuple of PPT algorithms* $\Sigma = (\mathsf{Setup}, \mathsf{Init}, \mathsf{Resp}, \mathsf{Verify}, \mathsf{BlindSetup}, \mathsf{BlindInit}, \mathsf{BlindChall}, \mathsf{BlindChall}^{-1}, \mathsf{BlindResp})$*, where except for* Setup *and* $\mathsf{BlindSetup}$ *all algorithms are deterministic. Moreover, we require that*

- $\mathsf{Setup}(1^\lambda)$*: outputs a state* st*.*
- $\mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})$*: given*[7] *state* st *and linear map* $\phi_{\mathbb{x}}$*, outputs a first flow message (i.e., commitment)* $A \in \mathcal{COM}$*.*
- $\mathsf{Resp}(\mathsf{st}, \gamma^*, \mathbb{w})$*: given a state* st*, a challenge* $\gamma^* \in \mathcal{CH}$ *and a witness* $\mathbb{w}$*, outputs a third flow message (i.e., response)* ζ^*
- $\mathsf{Verify}(\mathbb{x}, A^*, \gamma^*, \zeta^*)$*: given statement* $\mathbb{x} \in \mathscr{L}_{\mathsf{R}}$*, commitment* A^**, challenge* $\gamma^* \in \mathcal{CH}$*, and response* ζ^**, outputs a bit* $b \in \{0, 1\}$*.*
- $\mathsf{BlindSetup}(1^\lambda)$*: outputs a state* bst*.*
- $\mathsf{BlindInit}(\mathsf{bst}, \mathbb{x}, A^*)$*: given state* bst*, statement* $\mathbb{x}$*, commitment* A^**, outputs a commitment* A*.*
- $\mathsf{BlindChall}(\mathsf{bst}, \gamma^*)$*: given state* bst *and challenge* γ^**, outputs a challenge* γ*,*
- $\mathsf{BlindChall}^{-1}(\mathsf{bst}, \gamma)$*: given state* bst *and challenge* γ*, outputs a challenge* γ^**,*
- $\mathsf{BlindResp}(\mathsf{bst}, \zeta^*)$*: given state* bst *and response* ζ^**, outputs a response* ζ*,*

We call the tuple (A, γ, ζ) *the* transcript *and say that they are* valid for $\mathbb{x}$ *if* $\mathsf{Verify}(\mathbb{x}, A, \gamma, \zeta)$ *outputs 1. When the context is clear, we simply say it is valid and omit* $\mathbb{x}$*.*

[7] Conventionally, Init would take as input the entire statement $\mathbb{x}$. In contrast, we need to be able to generate a commitment even when the target value $\xi(\mathbb{x})$ is not yet fixed.

We define the standard notions of correctness, special honest-verifier zero-knowledge, and (2-)special soundness. We provide the formal definition of special honest-verifier zero-knowledge in Supplementary Material C.2. We state correctness explicitly to include the blind correctness.

Definition 5 (Correctness). *Let Σ be a (blindable) Σ-protocol for linear relation $\mathsf{R}_{(\phi,\xi)}$ as in Definition 4.*

- ***(Perfect) Correctness**: For all $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$, $\mathsf{st} \leftarrow \mathsf{Setup}(1^\lambda)$, $A \coloneqq \mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})$, $\gamma \in \mathcal{CH}$, and $\zeta \coloneqq \mathsf{Resp}(\mathsf{st}, \gamma)$, it holds that $\mathsf{Verify}(\mathbb{x}, A, \gamma, \zeta) = 1$.*
- ***(Perfect) Blind Correctness**: for all $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$, $\mathsf{bst} \leftarrow \mathsf{BlindSetup}(1^\lambda)$, $A^* \in \mathcal{COM}$, $\gamma^* \in \mathcal{CH}$, and $\zeta^* \in \mathcal{RESP}$ such that $\mathsf{Verify}(\mathbb{x}, A^*, \gamma^*, \zeta^*) = 1$, it holds that $\mathsf{Verify}(\mathbb{x}, A, \gamma, \zeta) = 1$ where $A \coloneqq \mathsf{BlindInit}(\mathsf{bst}, \mathbb{x}, A^*)$, $\gamma \coloneqq \mathsf{BlindChall}(\mathsf{bst}, \gamma^*)$ and $\zeta \coloneqq \mathsf{BlindResp}(\mathsf{bst}, \zeta^*)$.*
- ***Bijectivity of** $\mathsf{BlindChall}$: for all $\mathsf{bst} \leftarrow \mathsf{BlindSetup}(1^\lambda)$, the map $\mathcal{CH}^* \mapsto \mathsf{BlindChall}(\mathsf{bst}, \mathcal{CH}^*)$ is has inverse $\mathcal{CH} \mapsto \mathsf{BlindChall}^{-1}(\mathsf{bst}, \mathcal{CH})$.*

Definition 6 ((Transcript) Blindable Σ-protocol). *Let Σ be a blindable Σ-protocol for linear relation $\mathsf{R}_{(\phi,\xi)}$ as in Definition 4. Let $\mathcal{A}$ be a stateful adversary. The advantage of $\mathcal{A}$ against (transcript) blindness is*

$\mathsf{Exp}_{\mathsf{real}}(\lambda)$	$\mathsf{Exp}_{\mathsf{ideal}}(\lambda)$
$(\mathbb{x}, \mathbb{w}, (A^*, \gamma^*, \zeta^*)) \leftarrow \mathcal{A}(1^\lambda)$	$(\mathbb{x}, \mathbb{w}, (A^*, \gamma^*, \zeta^*)) \leftarrow \mathcal{A}(1^\lambda)$
abort if $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}_{(\phi,\xi)}$	**abort if** $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}_{(\phi,\xi)}$
abort if $\mathsf{Verify}(\mathbb{x}, (A^*, \gamma^*, \zeta^*)) = 0$	**abort if** $\mathsf{Verify}(\mathbb{x}, (A^*, \gamma^*, \zeta^*)) = 0$
// Blind the transcript	// Independent transcript
$\mathsf{bst} \leftarrow \mathsf{BlindSetup}(1^\lambda)$	$\mathsf{st} \leftarrow \mathsf{Setup}(1^\lambda)$
$A \leftarrow \mathsf{BlindInit}(\mathsf{bst}, \mathbb{x}, A^*)$	$A \leftarrow \mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})$
$\gamma \leftarrow \mathsf{BlindChall}(\mathsf{bst}, \gamma^*)$	$\gamma \leftarrow \mathcal{CH}$
$\zeta \leftarrow \mathsf{BlindResp}(\mathsf{bst}, \zeta^*)$	$\zeta \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma, \mathbb{w})$
$b \leftarrow \mathcal{A}(A, \gamma, \zeta)$	$b \leftarrow \mathcal{A}(A, \gamma, \zeta)$
return b	**return** b

Fig. 1: Blindability experiments.

$$\mathsf{AdvBlind}^{\Sigma}_{\mathcal{A}}(\lambda) \coloneqq |\Pr[\mathsf{Exp}_{\mathsf{real}}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathsf{ideal}}(\lambda) = 1]| \tag{1}$$

for the experiments defined in Figure 1. We say Σ is (perfectly) blindable if for any $\mathcal{A}$ the two distributions in Figure 1 are identical.

A final property which our protocol require is the translation of a transcript for one statement $\mathbb{x}^*$ into another related statement $\mathbb{x}$. Specifically, this will be used to rerandomize certain ciphertexts CT^*_i in $\mathbb{x}^*$.

Definition 7 ((Transcript) Translatable Σ-protocol). *Let Σ be a Σ-protocol for linear relation $\mathsf{R}_{(\phi,\xi)} \subseteq \mathcal{X} \times \mathcal{W}$. Let $\mathcal{V}$ be some set and let $\mathsf{TransStmt}\colon \mathcal{X} \times$*

$\mathcal{V} \to \mathcal{X}$ *and* $\mathsf{TransResp}\colon \mathcal{V} \times \mathcal{CH} \times \mathcal{RESP} \to \mathcal{RESP}$ *be efficiently computable maps such that for any statement* $\mathbb{x}^* \in \mathscr{L}_{\mathsf{R}_{(\phi,\xi)}}$ *and any* $v \in \mathcal{V}$*, it holds that* $\mathsf{TransStmt}(\mathbb{x}^*, v) \in \mathscr{L}_{\mathsf{R}_{(\phi,\xi)}}$*. We say* Σ *is (perfectly)* $\mathsf{TransStmt}$*-translatable if*

$\mathsf{Exp}_{\mathsf{fresh}}$	$\mathsf{Exp}_{\mathsf{trans}}$
$(\mathbb{x}, \mathbb{w}, \mathbb{x}^*, \mathbb{w}^*, v) \leftarrow \mathcal{A}(1^\lambda)$	$(\mathbb{x}, \mathbb{w}, \mathbb{x}^*, \mathbb{w}^*, v) \leftarrow \mathcal{A}(1^\lambda)$
abort if $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}_{(\phi,\xi)}$	**abort if** $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}_{(\phi,\xi)}$
abort if $(\mathbb{x}^*, \mathbb{w}^*) \notin \mathsf{R}_{(\phi,\xi)}$	**abort if** $(\mathbb{x}^*, \mathbb{w}^*) \notin \mathsf{R}_{(\phi,\xi)}$
abort if $\mathbb{x} \neq \mathsf{TransStmt}(\mathbb{x}^*, v)$	**abort if** $\mathbb{x} \neq \mathsf{TransStmt}(\mathbb{x}^*, v)$
// Generate fresh transcript for $\mathbb{x}$	// Translate a fresh transcript for $\mathbb{x}^*$
$\mathsf{st} \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathsf{st} \leftarrow \mathsf{Setup}(1^\lambda)$
$A \leftarrow \mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})$	$A \leftarrow \mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}^*})$
$\gamma \leftarrow \mathcal{CH}$	$\gamma \leftarrow \mathcal{CH}$
	$\widetilde{\zeta} \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma, \mathbb{w}^*)$
$\zeta \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma, \mathbb{w})$	$\zeta := \mathsf{TransResp}(v, \gamma, \widetilde{\zeta})$
$b \leftarrow \mathcal{A}(A, \gamma, \zeta)$	$b \leftarrow \mathcal{A}(A, \gamma, \zeta)$
return b	**return** b

Fig. 2: Translatability experiments. In both cases, if $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}_{(\phi,\xi)}$ or $(\mathbb{x}^*, \mathbb{w}^*) \notin \mathsf{R}_{(\phi,\xi)}$ the output is set to $\perp$.

there exists an efficiently computable map $\mathsf{TransResp}$ *such that for any* $\mathcal{A}$ *the two distributions in Figure 2 are identical.*

Allowing the adversary to choose v in Definition 7 is much stronger than required in our proofs, where v will be random. But it simplifies the definition and holds for our protocols (Definition 8).

Linear Σ-protocols Finally, we introduce the canonical Σ-protocol for a linear relation $\mathsf{R}_{(\phi,\xi)}$ for linear map ϕ [51]. It is well-known that this protocol is transcript-blindable.

Definition 8 (Canonical blindable Σ-protocol). *Let* ϕ, ξ *be as above and* R *be the associated linear NP-relation. We define the canonical blindable* Σ*-protocol* Σ *for* R *as follows:*

- $\mathsf{Setup}(1^\lambda)$*: samples randomness* $\mathsf{st} \leftarrow \mathcal{W}$*, outputs* st*.*
- $\mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})$*: outputs* $A^* := \phi_{\mathbb{x}}(\mathsf{st})$*.*
- $\mathsf{Resp}(\mathsf{st}, \gamma^*, \mathbb{w})$*: outputs the response* $\zeta^* := \mathsf{st} + \gamma^* \mathbb{w}$*.*
- $\mathsf{Verify}(\mathbb{x}, A^*, \gamma^*, \zeta^*)$*: outputs* 1 *iff* $A^* = \phi_{\mathbb{x}}(\zeta^*) + \gamma^* \xi(\mathbb{x})$*.*
- $\mathsf{BlindSetup}(1^\lambda)$*: samples* $(\gamma', \zeta') \leftarrow \mathcal{CH} \times \mathcal{W}$*, outputs* $\mathsf{bst} := (\gamma', \zeta')$*.*
- $\mathsf{BlindInit}(\mathsf{bst}, \mathbb{x}, A^*)$*: parses* $(A, \gamma', \zeta') := \mathsf{bst}$*, outputs* $A := A^* + \gamma' \xi(\mathbb{x}) + \phi_{\mathbb{x}}(\zeta')$*.*
- $\mathsf{BlindChall}(\mathsf{bst}, \gamma^*)$*: outputs* $\gamma := \gamma^* + \gamma'$*.*
- $\mathsf{BlindChall}^{-1}(\mathsf{bst}, \gamma)$*: outputs* $\gamma^* := \gamma - \gamma'$*.*
- $\mathsf{BlindResp}(\mathsf{bst}, \zeta^*)$*: outputs* $\zeta := \zeta^* + \zeta'$*.*

All variables which are superscripted with $*$ are message exchanged in the blinded protocol; the variables with superscript $'$ correspond to blinding terms, and finally the variables without superscript correspond to the blinded transcript.

Lemma 1. *The Σ-protocol in Definition 8 is a perfectly correct, SHVZK, and blindable.*

We give the proof in Supplementary Material D.

2.3 Non-Interactive Proof Systems

Here, we define (straightline-extractable) (zero-knowledge) non-interactive proof systems. Our definition is in the common *random* string (CRS) model combined with the (programmable) random oracle model. This is for simplicity: As we consider a common random string, we require no explicit setup algorithm; indeed, by domain separation, we can always generate the common random string through the random oracle.

Definition 9 (Non-Interactive Proof System). *A* non-interactive proof system Π *for NP-relation* R *in the common* random *string (CRS) model and random oracle model, is a pair* $\Pi = (\mathsf{Prove}, \mathsf{Ver})$ *of PPT algorithms, which have access to the CRS* $\mathsf{crs} \in \{0,1\}^{\ell(\lambda)}$ *and the random oracle* H*, where*

- $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \mathbb{w})$*: generates a proof* π *given a* crs *and* $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$.
- $\mathsf{Ver}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \pi)$*: verifies proof* π *for statement* $\mathbb{x}$ *given* crs *and outputs* 0 *or* 1.

We require four properties from a non-interactive proof system: correctness, zero-knowledge, straightline $\tilde{\mathsf{R}}$-extractability, and soundness. Due to space constraints, we give the formal definitions in Supplementary Material C.3. Correctness is straightforward, namely, every honestly generated proof verifies. Zero-Knowledge is guarantees the existence of a simulator that can produce proofs for valid statements with a witness (but it can program the CRS and random oracle). Straightline extractability means that for a given knowledge relation $\tilde{\mathsf{R}}$ there exists an extractor which programs the CRS with a trapdoor. Given a valid proof for a valid statement $\mathbb{x}$ the extractor produces a knowledge witness $\widetilde{\mathbb{w}}$ such that $(\mathbb{x}, \widetilde{\mathbb{w}}) \in \tilde{\mathsf{R}}$. It is called *straightline* because no rewinding is necessary. Finally, we require soundness which states it is infeasible to generate a proof (that verifies) for any invalid statement.

2.4 Public-Key Encryption

Definition 10 (Public-Key Encryption Scheme). *A public-key encryption (PKE) scheme* $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ *with message space* $\mathcal{M}_{\mathsf{RE}}$*, ciphertext space* $\mathcal{C}_{\mathsf{RE}}$*, public key space* $\mathcal{PK}_{\mathsf{RE}}$ *and randomness space* $\mathcal{R}_{\mathsf{RE}}$ *is a tuple of PPT algorithms defined as follows:*

- $\mathsf{PKE.Gen}(1^\lambda)$*: given security parameter* 1^λ*, outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$.

– $\mathsf{PKE.Enc}(\mathsf{pk}, m)$*: given public key* pk *and message* m*, outputs a ciphertext* ct*.*
– $\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct})$*: given secret key* sk *and ciphertext* ct*, outputs a message* m *or* $\perp$ *if the decryption fails.*

Notation. We use an implicit notation, for messages $m = (m_1, ..., m_n) \in \mathcal{M}_{\mathsf{PKE}}^n$ we write $\mathsf{ct} = (\mathsf{ct}_i = \mathsf{PKE.Enc}(\mathsf{pk}, m_i))_{i \in \{1,...,n\}}$.

For conceptual simplicity, we require the public key to be uniformly distributed.

Definition 11 (Uniform Public-Key). *A public-key encryption scheme* PKE *has a* uniform public-key *if the following distributions are equal:*

$$\left\{\mathsf{pk} \;\middle|\; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)\right\} \equiv \{\mathsf{pk} \mid \mathsf{pk} \leftarrow \mathcal{PK}_{\mathsf{PKE}}\} \ . \tag{2}$$

Definition 12 ((Perfectly) Linear Encryption). *Let* PKE *be as in Definition 10 and suppose that* $\mathcal{M}_{\mathsf{RE}}$*,* $\mathcal{R}_{\mathsf{RE}}$*,* $\mathcal{C}_{\mathsf{RE}}$ *are* $\mathbb{Z}_p$*-vector spaces. We say* PKE *has* linear encryption*, if for each public key* $\mathsf{pk}^{\mathsf{RE}} \in \mathcal{PK}$*, the encryption function* $\mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}, \cdot\,; \cdot) : m, \rho \mapsto \mathsf{CT}$ *is (*$\mathbb{Z}_p$*-)linear.*

Perfect linear encryption entails a number of natural properties, such as perfect correctness, and additional functionalities:

– **Perfect rerandomizability:** A rerandomizable encryption scheme RE is a PKE that also offers a rerandomization algorithm $\mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}; \rho')$ which outputs a rerandomized ciphertext. Any linear encryption scheme is randomizable via $\mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}; \rho') = \mathsf{CT} + \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}, 0; \rho')$, where $\rho' \leftarrow \mathcal{R}$. Moreover, this rerandomization is perfect and $\mathsf{CT} \mapsto \mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}; \rho')$ is linear in $\mathcal{C}_{\mathsf{RE}}$.
– **(Linear) Homomorphic evaluation:** A linear homomorphic encryption scheme LHE is a PKE that offers homomorphic evaluation of $\mathbb{Z}_p$-linear functions $f \colon \mathcal{M} \to \mathcal{M}^n$ by an algorithm $\mathsf{LHE.Eval}(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{CT}, f)$ which outputs ciphertexts encrypting $f(m)$ if CT encrypted m. Moreover, if LHE is rerandomizable, then by rerandomizing after a homomorphic linear evaluation, the applied function f is hidden; at most $f(m)$ is leaked. Any linear encryption scheme allows evaluation of linear functions and is rerandomizable.

In Supplementary Material C, we provide detailed definitions of the above mentioned properties and other common notions such as IND-CPA.

For future reference, we note the following.

Remark 1. Let PKE be a linear encryption scheme. Consider the linear relation $\mathsf{R}_{\mathsf{PKE}} \coloneqq \mathsf{R}_{\phi^{\mathsf{PKE}}, \xi^{\mathsf{PKE}}}$ for $\phi_{\mathbb{x}}^{\mathsf{PKE}}(\mathbb{w}) \coloneqq \mathsf{PKE.Enc}(\mathsf{pk}^{\mathsf{PKE}}, m; \rho)$ and $\xi^{\mathsf{PKE}}(\mathbb{x} = (\mathsf{pk}^{\mathsf{PKE}}, \mathsf{CT})) \coloneqq \mathsf{CT}$ be a target map, that is

$$\mathsf{R}_{\mathsf{PKE}} \coloneqq \left\{(\mathbb{x} = (\mathsf{pk}^{\mathsf{PKE}}, \mathsf{CT}), \mathbb{w} = (m, \rho)) \;\middle|\; \xi^{\mathsf{PKE}}(\mathbb{x}) = \phi_{\mathbb{x}}^{\mathsf{PKE}}(\mathbb{w})\right\} \tag{3}$$

$$\subseteq (\mathcal{X}_{\mathsf{PKE}} \coloneqq \mathcal{PK}_{\mathsf{PKE}} \times \mathcal{C}_{\mathsf{PKE}}) \times (\mathcal{W}_{\mathsf{PKE}} \coloneqq \mathcal{M}_{\mathsf{PKE}} \times \mathcal{R}_{\mathsf{PKE}}) \tag{4}$$

Then $\mathsf{R}_{\mathsf{PKE}}$ is a linear relation, since $\phi_{\mathbb{x}}^{\mathsf{PKE}}$ is linear for every choice of $\mathbb{x}$. Moreover, $\phi_{\mathbb{x}}^{\mathsf{PKE}}$ only depends on the component $\mathsf{pk}^{\mathsf{PKE}}$ of $\mathbb{x}$. This relation $\mathsf{R}_{\mathsf{PKE}}$ will be used to prove well-formedness of ciphertexts in our blind signature scheme.

2.5 Commitment Schemes

Definition 13 (Rerandomizable Linearly Homomorphic Dual-Mode Commitment Scheme). *Let $\mathcal{M}_{\mathsf{COM}}$ be an commutative group. A commitment scheme with message space $\mathcal{M}_{\mathsf{COM}}$, commitment space $\mathcal{C}_{\mathsf{COM}}$, parameter space $\mathcal{PP}_{\mathsf{COM}}$ and randomness space $\mathcal{R}_{\mathsf{COM}}$ is a tuple of algorithms*[8] $\mathsf{COM} = (\mathsf{COM.Setup}, \mathsf{COM.Commit})$ *defined as follows:*

- $\mathsf{COM.Setup}(1^\lambda, \mathsf{mode})$*: given security parameter λ and mode $\mathsf{mode} \in \{\mathsf{bind}, \mathsf{hide}\}$, outputs parameters pp.*
- $\mathsf{COM.Commit}(\mathsf{pp}, m; s)$*: given parameters $\mathsf{pp} \in \mathcal{PP}_{\mathsf{COM}}$ and message $m \in \mathcal{M}$ (and commitment randomness $s \in \mathcal{R}_{\mathsf{COM}}$), outputs a commitment CM.*
- $\mathsf{COM.Eval}(\mathsf{pp}, \mathsf{CM}, f)$*: given parameters $\mathsf{pp} \in \mathcal{PP}_{\mathsf{COM}}$, commitment $\mathsf{CM} = \mathsf{COM.Commit}(\mathsf{pp}, m; s)$, and linear function $f : \mathcal{M}_{\mathsf{COM}} \to \mathcal{M}_{\mathsf{COM}}^n$ for any $n \in \mathbb{N}$, outputs new commitments CM' to the plaintexts $f(m) \in \mathcal{M}_{\mathsf{COM}}^n$.*
- $\mathsf{COM.Rerand}(\mathsf{pp}, \mathsf{CM}; s')$*: given parameters $\mathsf{pp} \in \mathcal{PP}_{\mathsf{COM}}$, commitment $\mathsf{CM} = \mathsf{COM.Commit}(\mathsf{pp}, m; s)$, and randomness $s' \in \mathcal{R}_{\mathsf{COM}}$, outputs a fresh commitment CM' to the same message m.*
- $\mathsf{COM.RandEval}(\mathsf{pp}, m, s, f)$*: given parameters $\mathsf{pp} \in \mathcal{PP}_{\mathsf{COM}}$, message $m \in \mathcal{M}$, randomness $s \in \mathcal{R}_{\mathsf{COM}}$, linear function $f \colon \mathcal{M}_{\mathsf{COM}} \times \mathcal{R}_{\mathsf{COM}} \to \mathcal{M}_{\mathsf{COM}}^n$ for any $n \in \mathbb{N}$, outputs the randomness $\widetilde{s}$ of the homomorphically evaluated commitment.*
- $\mathsf{COM.RandRerand}(\mathsf{pp}, m, s, s')$*: given parameters $\mathsf{pp} \in \mathcal{PP}_{\mathsf{COM}}$, message $m \in \mathcal{M}$, randomness $s, s' \in \mathcal{R}_{\mathsf{COM}}$, outputs the randomness $\check{s}$ of the rerandomized commitment.*

Notation. We use an implicit notation for commitments for multiple messages. That is, for messages $m = (m_1, ..., m_n) \in \mathcal{M}_{\mathsf{COM}}^n$ we write $\mathsf{ct} = (\mathsf{ct}_i = \mathsf{COM.Commit}(\mathsf{pp}, m_i))_{i \in \{1,...,n\}}$.

For conceptual simplicity we require the following properties from an commitment scheme.

Definition 14 (Linear Commitment). *Let* Com *be as in Definition 13 and suppose that $\mathcal{M}_{\mathsf{RE}}$, $\mathcal{R}_{\mathsf{RE}}$, $\mathcal{C}_{\mathsf{RE}}$ are $\mathbb{Z}_p$-vector spaces. We say* Com *is a* linear commitment scheme*, if for each parameter $\mathsf{pp} \in \mathcal{PP}$ the commitment function $\mathsf{COM.Commit}(\mathsf{pp}, \cdot, \cdot) : m, s \mapsto \mathsf{CM}$ is linear.*

As with PKEs, given a linear commitment function, there are natural notions of $\mathsf{COM.Eval}$, $\mathsf{COM.RandEval}$, and $\mathsf{COM.Rerand}$, $\mathsf{COM.RandRerand}$. (Indeed, the only difference is that we handle message and randomness in separate algorithms, as this is required in our blind signature.) Moreover, similar to PKEs, we can see that the linear relation $\mathsf{R}_{\mathsf{COM}} = \mathsf{R}_{\phi^{\mathsf{COM}}, \xi^{\mathsf{COM}}}$

$$\mathsf{R}_{\mathsf{COM}} \coloneqq \left\{ (\mathbb{x} = (\mathsf{pp}, \mathsf{CM}), \mathbb{w} = (m, s)) \;\middle|\; \xi^{\mathsf{COM}}(\mathbb{x}) = \phi_{\mathbb{x}}^{\mathsf{COM}}(\mathbb{w}) \right\} \tag{5}$$

[8] We omit the typical "open" algorithm, because we don't need it in our construction.

Table 2: Extract of common protocol notation.

Com/Ctxt	Values	Rand	Scheme	Parameters/Keys/CRS
CMX_i	x_i	$\hat{s}_i$	COM_X	$\mathsf{pp}^\mathsf{X} = \mathsf{H}^\mathsf{X}_\mathsf{pp}(0)$
CMT_i	ct_i	s_i	COM_T	$\mathsf{pp}^\mathsf{T} = \mathsf{H}^\mathsf{T}_\mathsf{pp}(0)$
CMA_i	A_i	t_i	COM_T	
CT_i	Z_i	ρ_i	RE	$\mathsf{pk}^\mathsf{RE} = \mathsf{H}^\mathsf{RE}_\mathsf{pk}(0)$
ct	M	r	LHE	pk^LHE
vk	$x_i \in \mathbb{Z}_p$		BS	$(D_1, D_2, D_3) = \mathsf{H}^\mathsf{ddh}(0)$

The last row is irregular. It contains signature (secret) key and the public (non-)DDH parameters.

where $\phi^{\mathsf{COM}}_{\mathbb{x}}(\mathbb{w}) = \mathsf{COM.Commit}(\mathsf{pp}, m; s)$ and $\xi^{\mathsf{COM}}(\mathbb{x} = (\mathsf{pp}, \mathsf{CM})) \coloneqq \mathsf{CM}$, is linear for any linear commitment scheme.

For our construction we need several (fairly standard) properties: perfect hiding, perfect binding and rerandomization indistinguishability, uniform parameters, and parameter indistinguishability; listed in Supplementary Material C.5. We state unpredictability explicitly here, because it is used in Theorem 1.

Definition 15 (δ-Unpredictability). *A commitment scheme* COM *is* δ*-unpredictable, if for all* $\mathsf{pp} \in \mathcal{PP}_\mathsf{COM}$*, all* $m \in \mathcal{M}_\mathsf{COM}$*, and all* $\mathsf{CM} \in \mathcal{C}_\mathsf{COM}$*, we have that* $\Pr[\mathsf{COM.Commit}(\mathsf{pp}, m; s) = \mathsf{CM}] \leq \delta(\lambda)$*, with randomness is over* $s \leftarrow \mathcal{R}_\mathsf{COM}$*.*

3 Tight Signatures à la [2]

In this section, we define all primitives that are used in the construction of our blind signature scheme. We follow the conventions for group and group element notation in Section 2, in particular, $\mathbb{G}$ is a group of prime order p with generator G and group elements are denoted by capital letters. As noted in Section 2, we implicitly assume that an encryption/commitment scheme can be used to encrypt/commit to a vector of messages.

3.1 Primitives and Notation

We require following primitives:

- Let COM_X (resp. COM_T) be a commitment scheme (Definition 13) with message space $\mathcal{M}_{\mathsf{COM}_\mathsf{X}} \coloneqq \mathbb{Z}_p$ (resp. $\mathcal{M}_{\mathsf{COM}_\mathsf{T}} \coloneqq \mathbb{G}$) and public parameters pp^X (resp. pp^T). We use s_i (resp. t_i) for commitment randomness for COM_X (resp. COM_T).
- Let RE (resp. LHE) be a rerandomizable encryption scheme (resp. linearly homomorphic encryption scheme) (Definition 31) with message space $\mathcal{M}_\mathsf{RE} = \mathcal{M}_\mathsf{LHE} \coloneqq \mathbb{G}$ and public key pk^RE (resp. pk^LHE). We use $\hat{s}_i$ for commitment randomness for COM_X.
- In Table 2, we overview some frequent notation.

Linear Maps for Σ-protocols Now, we define the linear Σ-protocols for our protocol, which are derived from the (linear) verification maps $\phi^{\mathsf{COM_X}}$,$\phi^{\mathsf{COM_T}}$ and ϕ^{RE} be the maps associated with $\mathsf{COM_X}$, $\mathsf{COM_T}$ and RE, respectively. Let $\mathcal{CH} \coloneqq \mathbb{Z}_p$ be the common challenge space for all Σ-protocols.

For convenience, we define following shorthands for (partial) statements and witnesses, where $i \in \{0,1,2\}$:

$$\begin{aligned} \mathbb{x}_{\mathsf{CMX}_i} &= (\mathsf{pp}_i^{\mathsf{X}}, \mathsf{CMX}_i) & \mathbb{w}_{\mathsf{CMX}_i} &= (x_i, \hat{s}_i) \\ \mathbb{x}_{\mathsf{RE},i} &= (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CT}_i) & \mathbb{w}_{\mathsf{RE},i} &= (Z_i, \rho_i) \\ \mathbb{x}_{\mathsf{dh}} &= (G, D_1, D_2, D_3) & \mathbb{w}_{\mathsf{dh}} &\in \mathbb{Z}_p \end{aligned} \tag{6}$$

Next, we define the linear relation for Σ-protocols. For linear relation $\mathsf{R}_0 = \mathsf{R}_{(\phi^0,\xi^0)}$, we define the statement witness pairs and the function pair (ϕ^0, ξ^0) as follows:

$$\begin{aligned} \mathbb{x}_0 &= (\mathbb{x}_{\mathsf{CMX}_0}, \mathbb{x}_{\mathsf{CMX}_1}, \mathbb{x}_{\mathsf{RE},0}) = ((\mathsf{pp}_0^{\mathsf{X}}, \mathsf{CMX}_0), (\mathsf{pp}_1^{\mathsf{X}}, \mathsf{CMX}_1), (\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CT}_0)) \\ \mathbb{w}_0 &= (\mathbb{w}_{\mathsf{CMX}_0}, \mathbb{w}_{\mathsf{CMX}_1}, \mathbb{w}_{\mathsf{RE},0}) = ((x_0, \hat{s}_0), (x_1, \hat{s}_1), (Z_0, \rho_0)) \end{aligned} \tag{7}$$

$$\phi^0_{\mathbb{x}_0}(\mathbb{w}_0) \coloneqq \begin{pmatrix} \phi^{\mathsf{COM_X}}_{\mathbb{x}_{\mathsf{CMX}_0}}(\mathbb{w}_{\mathsf{CMX}_0}) \\ \phi^{\mathsf{COM_X}}_{\mathbb{x}_{\mathsf{CMX}_0}}(\mathbb{w}_{\mathsf{CMX}_1}) \\ \phi^{\mathsf{RE}}_{\mathbb{x}_{\mathsf{RE},0}}(\mathbb{w}_{\mathsf{RE},0}) \\ x_0 G + x_1 M - Z_0 \end{pmatrix} \overset{!}{=} \begin{pmatrix} \mathsf{CMX}_0 \\ \mathsf{CMX}_1 \\ \mathsf{CT}_0 \\ 0 \end{pmatrix} =: \xi^0(\mathbb{x}_0)$$

Relation R_0 states that there are scalars x_0, x_1 and commitment randomness $\hat{s}_0, \hat{s}_1$ for $\mathsf{CMX}_0, \mathsf{CMX}_1$, respectively (and under respective public parameters) so that CT_0 encrypts $Z_0 = x_0 G + x_1 M$. Note that for $x_1 = 0$, this means $Z_0 = x_0 G$. Here and in the following, we leave the (co)domain of functions implicitly specified by the types of the statement and witness. Moreover, here and in the following, it is clear that $\phi^0_{\mathbb{x}_0}$ is a linear (as a composition of linear maps).

The second relation $\mathsf{R}_1 = \mathsf{R}_{\phi^1,\xi^1}$ states that CT_0 and CT_1 encrypt the same message $Z_0 = Z_1$. It is defined by

$$\begin{aligned} \mathbb{x}_1 &\coloneqq (\mathbb{x}_{\mathsf{RE},0}, \mathbb{x}_{\mathsf{RE},1}) = ((\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CT}_0), (\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CT}_0)) \\ \mathbb{w}_1 &\coloneqq (\mathbb{w}_{\mathsf{RE},0}, \mathbb{w}_{\mathsf{RE},1}) = ((Z_0, \rho_0), (Z_1, \rho_1)) \end{aligned} \tag{8}$$

$$\phi^1_{\mathbb{x}_1}(\mathbb{w}_1) \coloneqq \begin{pmatrix} \phi^{\mathsf{RE}}_{\mathbb{x}_{\mathsf{RE},0}}(\mathbb{w}_{\mathsf{RE},0}) \\ \phi^{\mathsf{RE}}_{\mathbb{x}_{\mathsf{RE},1}}(\mathbb{w}_{\mathsf{RE},1}) \\ Z_0 - Z_1 \end{pmatrix} \overset{!}{=} \begin{pmatrix} \mathsf{CT}_0 \\ \mathsf{CT}_1 \\ 0 \end{pmatrix} =: \xi^1(\mathbb{x}_1)$$

The next relation $\mathsf{R}_2 = \mathsf{R}_{\phi^2,\xi^2}$ states that CMX_2 commits to the dlog encrypted in CT_2, i.e. $x_2 G = Z_2$. It is defined by

$$\begin{aligned} \mathbb{x}_2 &= (\mathbb{x}_{\mathsf{CMX}_2}, \mathbb{x}_{\mathsf{RE},2}) = ((\mathsf{pp}_2^{\mathsf{X}}, \mathsf{CMX}_2), (\mathsf{pk}_2^{\mathsf{RE}}, \mathsf{CT}_2)) \\ \mathbb{w}_2 &= (\mathbb{w}_{\mathsf{COM_X},2}, \mathbb{w}_{\mathsf{RE},2}) = ((x_2, \hat{s}_2), (Z_2, \rho_2)) \end{aligned} \tag{9}$$

$$\phi^2_{\mathbb{x}_2}(\mathbb{w}_2) \coloneqq \begin{pmatrix} \phi^{\mathsf{COM_X}}_{\mathbb{x}_{\mathsf{CMX}_2}}(\mathbb{w}_{\mathsf{CMX}_2}) \\ \phi^{\mathsf{RE}}_{\mathbb{x}_{\mathsf{RE},2}}(\mathbb{w}_{\mathsf{RE},2}) \\ x_2 G - Z_2 \end{pmatrix} \overset{!}{=} \begin{pmatrix} \mathsf{CMX}_2 \\ \mathsf{CT}_2 \\ 0 \end{pmatrix} =: \xi^2(\mathbb{x}_2)$$

Our last relation is the DDH relation $\mathsf{R}_{\mathsf{dh}} = \mathsf{R}_{\phi^{\mathsf{dh}},\xi^{\mathsf{dh}}}$

$$\mathbb{x}_{\mathsf{dh}} \coloneqq (G, D_1, D_2, D_3) \qquad \mathbb{w}_{\mathsf{dh}} \in \mathbb{Z}_p \tag{10}$$
$$\phi^{\mathsf{dh}}_{\mathbb{x}_{\mathsf{dh}}}(\mathbb{w}_{\mathsf{dh}}) \coloneqq \begin{pmatrix} \mathbb{w}_{\mathsf{dh}} G \\ \mathbb{w}_{\mathsf{dh}} D_1 \end{pmatrix} \overset{!}{=} \begin{pmatrix} D_2 \\ D_3 \end{pmatrix} =: \xi^{\mathsf{dh}}(\mathbb{x}_{\mathsf{dh}})$$

Finally, we write $\mathbf{\Sigma}_i$ for the canonical Σ-protocol for relations R_i ($i \in \{0,1,2,\mathsf{dh}\}$).

Translatability. To achieve blindness, the user must rerandomize the ciphertexts CT_i^* in $\mathbb{x}^*_{\mathsf{RE},i} = (\mathsf{pp}_i^{\mathsf{X}}, \mathsf{CT}_i^*)$ and adapt the Σ-protocol responses suitably (for $i \in \{0,1,2\}$). For this, we define translations $\mathsf{TransStmt}_i$ as

$$\mathsf{TransStmt}_0(\mathbb{x}_0^* = (\mathbb{x}_{\mathsf{CMX}_0}, \mathbb{x}_{\mathsf{CMX}_1}, \boxed{\mathbb{x}^*_{\mathsf{RE},0}}, M), \rho_0') \coloneqq (\mathbb{x}_{\mathsf{CMX}_0}, \mathbb{x}_{\mathsf{CMX}_1}, \boxed{\mathbb{x}_{\mathsf{RE},0}}, M)$$
$$\mathsf{TransStmt}_1(\mathbb{x}_1^* = (\boxed{\mathbb{x}^*_{\mathsf{RE},0}, \mathbb{x}^*_{\mathsf{RE},1}}), (\rho_0', \rho_1')) \coloneqq (\boxed{\mathbb{x}_{\mathsf{RE},0}, \mathbb{x}_{\mathsf{RE},1}})$$
$$\mathsf{TransStmt}_2(\mathbb{x}_2^* = (\mathbb{x}_{\mathsf{CMX}_2}, \boxed{\mathbb{x}^*_{\mathsf{RE},2}}), \rho_2') \coloneqq (\mathbb{x}_{\mathsf{CMX}_2}, \boxed{\mathbb{x}_{\mathsf{RE},2}})$$

where, for $i \in \{0,1,2\}$, we let

$$\mathsf{CT}_i \coloneqq \mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}_i^*; \rho_i') \quad \text{in} \quad \mathbb{x}_{\mathsf{RE},i} = (\mathsf{pp}_i^{\mathsf{X}}, \mathsf{CT}_i).$$

Lemma 2. *For each $i \in \{0,1,2\}$ the canonical Σ-protocol $\mathbf{\Sigma}_i$ from Definition 8 for the relation R_i is $\mathsf{TransStmt}_i$-translatable.*

The proof is included in Supplementary Material D.

Functions for Homomorphic Evaluation Within our protocol, the user and signer evaluate some functions homomorphically on ciphertexts or commitments. To reduce the syntactic overhead, we define these functions here. The function

$$\mathsf{InitZero}\colon \boxed{M} \mapsto A_0^* \coloneqq \Sigma_0.\mathsf{Init}(\mathsf{st}_0, \phi^0_{(\mathbb{x}_{\mathsf{CMX}_0}, \mathbb{x}_{\mathsf{CMX}_1}, \mathbb{x}^*_{\mathsf{RE},0}, \boxed{M})}) \tag{11}$$

allows to compute the Σ-commitment A_0^* for the first relation R_0 from the message M. At the time of (homomorphic) evaluation of $\mathsf{InitZero}$ the values $\mathsf{st}_0 \in \mathcal{W}_0$, $\mathbb{x}_{\mathsf{CMX}_0} \in \mathcal{X}_{\mathsf{COM}_{\mathsf{X}}}$, $\mathbb{x}_{\mathsf{CMX}_1} \in \mathcal{X}_{\mathsf{COM}_{\mathsf{X}}}$, $\mathbb{x}^*_{\mathsf{RE},0} \in \mathcal{X}_{\mathsf{RE}}$ are known to the signer who hardcodes them into $\mathsf{InitBlind}$. Note also, that the function $\mathsf{InitZero}$ is linear in the message M. In our protocol, the user will encrypt M in ct_M and the signer evaluates $\mathsf{InitZero}$ on ct_M to obtain an encryption of A_0^*. This ensures that the user can obtain A_0^* without revealing the message M to the signer.
In our protocol, the user initially obtains commitments CMT_i^* to the ciphertexts CT_i^*. The user then homomorphically computes the function

$$\mathsf{RandCT}_i\colon \boxed{\mathsf{CT}_i^*} \mapsto \mathsf{CT}_i \coloneqq \mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \boxed{\mathsf{CT}_i^*}; \rho_i') \ . \tag{12}$$

to obtain the rerandomized ciphertexts CT_i that are ultimately included in the signature. Analogously to the first function, the user knows the values $\mathsf{pk}^{\mathsf{RE}} \in \mathcal{PK}_{\mathsf{RE}}$

and $\rho'_i \in \mathcal{R}_{\mathsf{RE}}$ hardcoded in RandCT_i. Because RE is linearly rerandomizable, the function RandCT_i is linear in the ciphertext CT_i^*.
Finally, the signature contains blinded (and translated) transcripts of our Σ-protocols. For blindness of our signature scheme the user homomorphically computes commitments to the blinded Σ-commitments A_i by applying the functions

$$\mathsf{InitBlind}_i\colon \boxed{\mathsf{IN}_i^*} \mapsto A_i \coloneqq \Sigma_i.\mathsf{BlindInit}(\mathsf{bst}_i, \boxed{\mathbb{x}_i^*}, A_i^*), \tag{13}$$

to its commitments CMT_i^* where

$$\mathsf{IN}_0^* = \mathsf{CT}_0^*, \quad \mathsf{IN}_1^* = (\mathsf{CT}_0^*, \mathsf{CT}_1^*), \quad \mathsf{IN}_2^* = \mathsf{CT}_2^*$$

and $\mathbb{x}_i^*$ are as before, and in particular contain IN_i^*. Again, the user hardcodes the values $\mathsf{bst}_i \in \mathcal{CH}_i \times \mathcal{W}_i$, $A_i^* \in \mathcal{COM}_i$ and $\mathbb{x}_{\mathsf{CMX}_i} = (\mathsf{pp}_i^{\mathsf{X}}, \mathsf{CMX}_i)$ respectively into $\mathsf{InitBlind}_i$. Note here that the user obtains the values A_1^* and A_2^* directly from the signer, but it decrypts the ciphertext ct_0^* to obtain the value A_0^*.

Non-Interactive Proof Systems We require non-interactive proofs (Definition 9) for two relations. The first relation is

$$\mathsf{R}_M \coloneqq \left\{ (\mathbb{x} = (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}), \mathbb{w} = (\mathsf{sk}^{\mathsf{LHE}}, M, r_M)) \;\middle|\; \begin{array}{c} (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{sk}^{\mathsf{LHE}}) \in \mathsf{LHE.Gen}(1^\lambda) \\ \wedge\ \mathsf{ct} = \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}, M; r_M) \end{array} \right\} \tag{14}$$

$$\tilde{\mathsf{R}}_M \coloneqq \left\{ (\mathbb{x} = (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}), \widetilde{\mathbb{w}} = M) \;\middle|\; \exists \mathbb{w} = (\mathsf{sk}^{\mathsf{LHE}}, M, r_M) : (\mathbb{x}, \mathbb{w}) \in \mathsf{R}_M \right\} \tag{15}$$

where the correctness relation R_M asserts that the user's LHE public key $\mathsf{pk}^{\mathsf{LHE}}$ and encryption of M are honestly generated. Moreover, we require straightline extractability of M through the knowledge relation $\tilde{\mathsf{R}}_M$. We let Π_M w.r.t. random oracle H_M be a NIPS for the above.

The second relation for which we require non-interactive proofs is

$$\mathsf{R}_{\mathsf{CT}} \coloneqq \left\{ \left(\begin{array}{l} \mathbb{x} = \left(\begin{array}{c} \mathsf{pp}_0^{\mathsf{X}}, \mathsf{CMX}_0 \\ (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CT}_i^*)_{i\in\{0,1,2\}} \end{array} \right) \\ \mathbb{w} = ((\rho_i^*)_{i\in\{0,1,2\}}, x_0, \hat{s}_0) \end{array} \right) \;\middle|\; \begin{array}{r} \mathsf{CT}_0^* = \mathsf{RE.Enc}(\mathsf{pk}_0^{\mathsf{RE}}, x_0 G; \rho_0^*) \\ \mathsf{CT}_1^* = \mathsf{RE.Enc}(\mathsf{pk}_1^{\mathsf{RE}}, x_0 G; \rho_1^*) \\ \mathsf{CT}_2^* = \mathsf{RE.Enc}(\mathsf{pk}_2^{\mathsf{RE}}, 0; \rho_2^*) \\ \mathsf{CMX}_0 = \mathsf{COM}_{\mathsf{X}}.\mathsf{Commit}(\mathsf{pp}_0^{\mathsf{X}}, x_0; \hat{s}_0) \end{array} \right\} \tag{16}$$

where the (correctness and soundness) relation R_{CT} asserts an honest generation of the ciphertext CT_i^* which is in particular consistent with CMX_0. We let Π_{CT} w.r.t. random oracle H_{CT} be a NIPS for the above.

3.2 Construction

We generate several parameters as evaluations of hash functions. For $i \in \{0,1,2\}$, let $\mathsf{pp}_i^{\mathsf{X}} \coloneqq \mathsf{H}_{\mathsf{pp}}^{\mathsf{X}}(i)$, $\mathsf{pp}^{\mathsf{T}} \coloneqq \mathsf{H}_{\mathsf{pp}}^{\mathsf{T}}(0)$, $(D_1, D_2, D_3) \coloneqq \mathsf{H}^{\mathsf{ddh}}(0)$ and $\mathsf{pk}_i^{\mathsf{RE}} \coloneqq \mathsf{H}_{\mathsf{pk}}^{\mathsf{RE}}(i)$.

Also, let $\mathsf{crs}_{\mathsf{CT}} := \mathsf{H}^{\mathsf{CT}}_{\mathsf{crs}}(0)$ and $\mathsf{crs}_M := \mathsf{H}^{M}_{\mathsf{crs}}(0)$. We assume that these parameters are (implicitly) computed by signer, user and verifier during signing and verification.

High-level description. As our construction is involved, let us give a brief description and provide some intuition. Roughly, the goal of the protocol is that the user obtains ciphertexts CT_i together with Fiat-Shamir compiled proofs $(\pi_0, \pi_{\mathsf{dh}})$ and (π_1, π_2) for disjunctive relations $\mathsf{R}_0 \cup \mathsf{R}_{\mathsf{dh}}$ and $\mathsf{R}_1 \cup \mathsf{R}_2$ (cf. Section 3.1), respectively, where the commitments CMX_i are fixed in the verification key vk. Except for some minor modifications explained below, the signature consists of ciphertexts $(\mathsf{CT}_0, \mathsf{CT}_1, \mathsf{CT}_2)$ and proofs $(\pi_0, \pi_{\mathsf{dh}}), (\pi_1, \pi_2)$. This is also the conceptual structure of signatures in [2]. The aforementioned modifications are required for blindness and a tight proof of one-more unforgeability.

Generally, we follow the convention that values that are sent during the signing protocol are marked with $*$. Non-marked values are often randomized and part of the final signature or fixed by the verification key vk or random oracles.

KeyGen. The signer samples $x_0 \leftarrow \mathbb{Z}_p$ and sets $x_1 = x_2 = 0$. It commits to x_i in $\mathsf{CMX}_i = \mathsf{COM}_{\mathsf{X}}.\mathsf{Commit}(\mathsf{pp}^{\mathsf{X}}_i, x_i; \hat{s}_i)$ for some random $\hat{s}_i$. The verification key is $\mathsf{vk} = (\mathsf{CMX}_0, \mathsf{CMX}_1, \mathsf{CMX}_2)$ and the signing key is $\mathsf{sk} = (x_i, \hat{s}_i)_{i\in\{0,1,2\}}$.

Signing Protocol. The signing protocol proceeds in four moves.

1st message ($\mathsf{U} \to \mathsf{S}$). On input of verification key vk and message M, the user samples fresh keys $(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{sk}^{\mathsf{LHE}}) \leftarrow \mathsf{LHE}.\mathsf{Gen}(1^\lambda)$ for LHE and computes a ciphertext $\mathsf{ct}_M \leftarrow \mathsf{LHE}.\mathsf{Enc}(\mathsf{pk}^{\mathsf{LHE}}, M)$ to M. The user then computes a proof π_M via Π_M that certifies honest setup of $\mathsf{pk}^{\mathsf{LHE}}$ and ct_M. Then, the user outputs

$$\mathsf{smsg}_1 := (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}_M, \pi_M).$$

2nd message ($\mathsf{S} \to \mathsf{U}$). On input of signing key sk and $\mathsf{smsg}_1 = (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}_M, \pi_M)$, the signer checks that the proof π_M verifies and aborts otherwise.
If the check succeeds, the signer prepares ciphertexts CT^*_i to $Z_i := z_i G$ with $z_0 = x_0, z_1 = x_0$ and $z_2 = 0$. It also prepares the first flow for an interactive Σ-protocol proof via $\Sigma_i.\mathsf{Init}$. This interactive proof is blinded and compiled via Fiat-Shamir by the user to a non-interactive proof. More specifically, the signer proves that the disjunctive relations

$$\begin{aligned} x_0 G + x_1 M = Z_0 \quad &\vee \quad d_2 D_1 = D_3, && (\mathsf{R}_0 \cup \mathsf{R}_{\mathsf{dh}}) \\ Z_0 = Z_1 \quad &\vee \quad x_2 G = Z_2, && (\mathsf{R}_1 \cup \mathsf{R}_2) \end{aligned}$$

hold, where x_i are the values committed in vk. For this, the signer runs the OR-compilation of Σ-protocols $(\Sigma_0, \Sigma_{\mathsf{dh}})$ and (Σ_1, Σ_2) as in [19] (i.e., via additive secret sharing of the challenge and the HVZK simulator). Concretely, the first flow contains honestly computed A^*_i and simulated A^*_{dh}.
As relation R_0 contains message M which is not known to the signer, the first flow A^*_0 of Σ_0 is computed homomorphically by evaluating $\mathsf{InitZero}$ on ct_M. The signer obtains a ciphertext ct^*_0 which encrypts A^*_0.

For technical reasons, the signer cannot reveal CT_i^* to the user immediately, so it commits to CT_i^* in CMT_i^* via the dual-mode commitment COM_T instead. The signer outputs

$$\mathsf{smsg}_2 \coloneqq (\mathsf{ct}_0^*, (\mathsf{CMT}_i^*)_{i\in\{0,1,2\}}, A_1^*, A_2^*, A_{\mathsf{dh}}^*).$$

3rd message ($\mathsf{U} \to \mathsf{S}$). On input of $\mathsf{smsg}_2 = (\mathsf{ct}_0^*, (\mathsf{CMT}_i^*)_{i\in\{0,1,2\}}, A_1^*, A_2^*, A_{\mathsf{dh}}^*)$, the user recovers its state and decrypts ct_0^* via $\mathsf{sk}^{\mathsf{LHE}}$ to obtain A_0^*. To ensure blindness later, the user performs several randomization steps:

1. The user randomizes both the ciphertexts CT_i^* committed in CMT_i^* homomorphically and the commitments itself to obtain CMT_i. Note if the signer is honest,[9] then CMT_i is distributed like a fresh commitment to CT_i which is a fresh encryption to $z_i G$.
2. The user blinds the Σ-protocol commitments A_i^* and A_{dh} via $\mathsf{BlindInit}$. For A_i^* the function $\mathsf{BlindInit}$ expects CT_i^* in plain as input (which is not known by the user at this point), hence the user evaluates $\mathsf{BlindInit}$ homomorphically over the commitments CMT_i^* to CT_i^* to obtain commitments $\widetilde{\mathsf{CMA}}_i$ to blinded A_i.
3. The user randomizes the commitments $\widetilde{\mathsf{CMA}}_i$ to obtain commitments CMA_i to A_i.

Then, to compile the Σ-protocols via Fiat-Shamir, the user hashes (via $\mathsf{H}_0^{\mathcal{CH}}$ and $\mathsf{H}_1^{\mathcal{CH}}$) the first flow and statements (or commitments thereof if not available in plain) to obtain two challenges $\delta_{0,\mathsf{dh}}$ and $\delta_{1,2}$. Namely, it queries challenges $\delta_b \coloneqq \mathsf{H}_b^{\mathcal{CH}}(\mathsf{HIN}_b)$ for $b \in \{0,1\}$ where

$$\begin{aligned}\mathsf{HIN}_0 &\coloneqq \big((\mathsf{pp}_i^{\mathsf{X}}, \mathsf{CMX}_i)_{i\in\{0,1\}}, (\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CMT}_0), M, \mathsf{CMA}_0, \mathbb{x}_{\mathsf{dh}}, A_{\mathsf{dh}}\big)\\ \mathsf{HIN}_1 &\coloneqq \big((\mathsf{pp}_2^{\mathsf{X}}, \mathsf{CMX}_2), (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CMT}_i)_{i\in\{0,1,2\}}, (\mathsf{CMA}_i)_{i\in\{0,1,2\}}\big)\end{aligned} \tag{17}$$

The user prepares these challenges for blinding via $\mathsf{BlindChall}^{-1}$ to obtain $(\delta_{0,\mathsf{dh}}^*, \delta_{1,2}^*)$ and outputs

$$\mathsf{smsg}_3 \coloneqq (\delta_{0,\mathsf{dh}}^*, \delta_{1,2}^*).$$

4th message ($\mathsf{S} \to \mathsf{U}$). On input of $\mathsf{smsg}_3 = (\delta_{0,\mathsf{dh}}^*, \delta_{1,2}^*)$, the signer parses its state and computes the Σ-protocol responses $(\zeta_0^*, \zeta_1^*, \zeta_2^*, \zeta_{\mathsf{dh}}^*)$ via appropriate witnesses[10] for challenges $(\gamma_0^*, \gamma_{\mathsf{dh}}^*)$ and (γ_1^*, γ_2^*) which are additive sharings of $\delta_{0,\mathsf{dh}}^*$ and $\delta_{1,2}^*$, respectively. Finally, the signer outputs

$$\mathsf{smsg}_4 \coloneqq ((\gamma_i^*, \zeta_i^*, \mathsf{CT}_i^*, s_i^*)_{i\in\{0,1,2\}}, \gamma_{\mathsf{dh}}^*, \zeta_{\mathsf{dh}}^*, \pi_{\mathsf{CT}}^*),$$

where π_{CT}^* is a proof computed via Π_{CT} which certifies that ciphertexts CT_i^* are setup honestly, and s_i^* is an opening for CMT_i^*.

[9] Roughly, this is ensured by interactive execution of the Σ-protocols and appropriate NIZKs and verified by the user in U_3.

[10] As the Σ_{dh} transcript is simulated, the response ζ_{dh}^* is the simulated response that was created when HVZK was invoked.

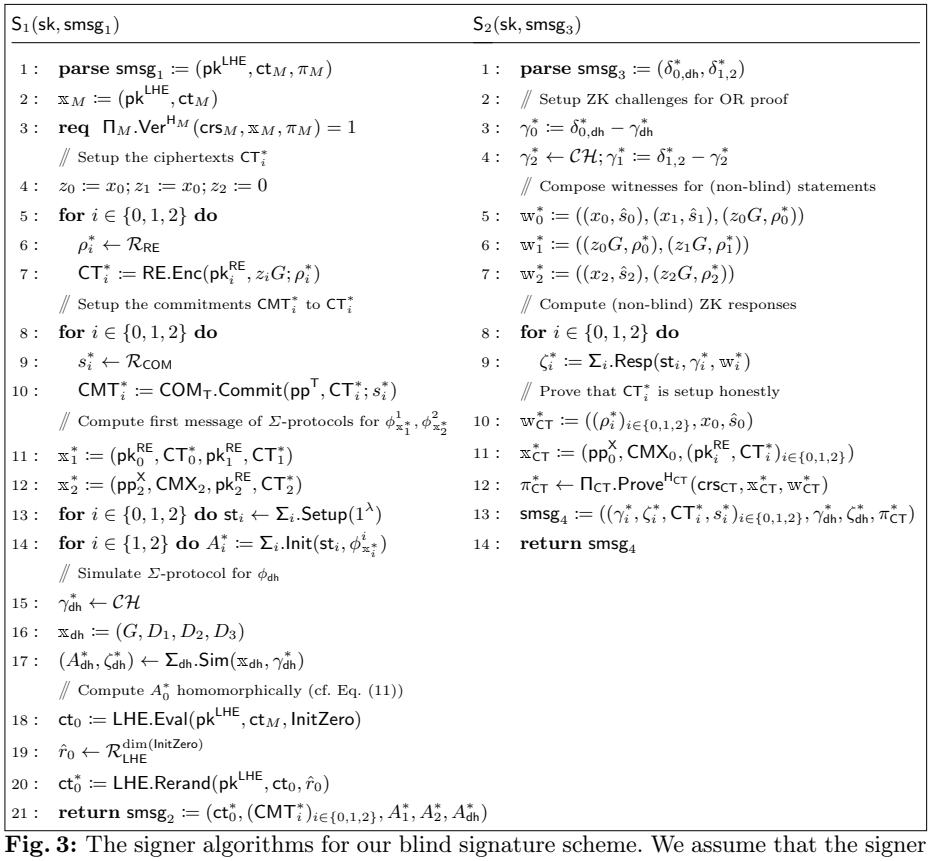

Fig. 3: The signer algorithms for our blind signature scheme. We assume that the signer is stateful and but omit its state for conciseness.

Signature derivation. Finally, the user derives its signature as follows. It parses smsg_4 as above, and verifies the proof π_{CT}^* and that the Σ-protocol transcripts are valid via Verify. If either check fails, the user aborts. Else, the user recomputes CT_i from CT_i^* and openings s_i and t_i for CMT_i and CMA_i via s_i^*. This is possible via $\mathsf{RandEval}$ and $\mathsf{RandRerand}$, as the user knows the random coins that were used to randomize (homomorphically). Finally, it also recovers A_i^* in plain by reevaluating $\mathsf{BlindInit}$ with CT_i and blinds the challenges and responses via $\mathsf{BlindChall}$ and $\mathsf{BlindResp}$, and outputs

$$\sigma \coloneqq ((\mathsf{CT}_i, \pi_i, s_i, t_i)_{i\in\{0,1,2\}}, \pi_{\mathsf{dh}}),$$

where π_i and π_{dh} are the blinded Σ-protocol transcripts.

Verification. To verify a signature σ, the verifier parses σ as above. Then, it recomputes the commitments CMT_i and CMA_i via s_i and t_i, respectively. It outputs 1 iff the proofs π_i pass verification via Verify, and $\gamma_0 + \gamma_{\mathsf{dh}} = \mathsf{H}_0^{\mathcal{CH}}(\mathsf{HIN}_0)$ and $\gamma_1 + \gamma_2 = \mathsf{H}_1^{\mathcal{CH}}(\mathsf{HIN}_1)$. Here, HIN_0 and HIN_1 are as in Eq. (17)

A formal description is given below. We note here that, since an encryption of the to-be-signed message M is sent to the signer, it is efficiently possible to prove

statements over M in zero-knowledge. In the terminology of [28], we achieve *predicate blindness*.

BS: Pairing-free blind signature based on [2]

- $\mathsf{KeyGen}(1^\lambda)$:
  1. Sample $x_0 \leftarrow \mathbb{Z}_p$ and $\hat{s}_i \leftarrow \mathcal{R}_{\mathsf{COM}}$, set $x_1 = x_2 = 0$,
  2. For $i \in \{0,1,2\}$, set $\mathsf{CMX}_i := \mathsf{COM}_{\mathsf{X}}.\mathsf{Commit}(\mathsf{pp}_i^{\mathsf{X}}, x_i; \hat{s}_i)$.
  3. Output $\mathsf{vk} := (\mathsf{CMX}_0, \mathsf{CMX}_1, \mathsf{CMX}_2)$ and $\mathsf{sk} := (x_0, x_1, x_2, \hat{s}_1, \hat{s}_2, \hat{s}_3)$.
- $\mathsf{S}(\mathsf{sk}) \longleftrightarrow \mathsf{U}(\mathsf{vk}, M)$: Proceeds in 4 moves and is given in Figures 3 and 4. We assume that each signing session is implicitly identified by a session identifier and that user and signer keep appropriate states.
- $\mathsf{Verify}(\mathsf{vk}, M, \sigma)$:
  1. Parse $\sigma := ((\mathsf{CT}_i, \pi_i, s_i, t_i)_{i\in\{0,1,2\}}, \pi_{\mathsf{dh}})$.
  2. Set $\mathbb{x}_0 := ((\mathsf{pp}_0^{\mathsf{X}}, \mathsf{CMX}_0), (\mathsf{pp}_1^{\mathsf{X}}, \mathsf{CMX}_1), (\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CT}_0), M)$.
  3. Set $\mathbb{x}_1 := ((\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CT}_0), (\mathsf{pk}_1^{\mathsf{RE}}, \mathsf{CT}_1))$.
  4. Set $\mathbb{x}_2 := ((\mathsf{pp}_2^{\mathsf{X}}, \mathsf{CMX}_2), (\mathsf{pk}_2^{\mathsf{RE}}, \mathsf{CT}_2))$.
  5. Set $\mathbb{x}_{\mathsf{dh}} := (G, D_1, D_2, D_3)$.
  6. Parse $\pi_i := (A_i, \gamma_i, \zeta_i)$ for $i \in \{0,1,2\}$.
  7. Parse $\pi_{\mathsf{dh}} := (A_{\mathsf{dh}}, \gamma_{\mathsf{dh}}, \zeta_{\mathsf{dh}})$.
  8. Set $\mathsf{CMT}_i := \mathsf{COM}_{\mathsf{T}}.\mathsf{Commit}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}_i, s_i)$ for $i \in \{0,1,2\}$.
  9. Set $\mathsf{CMA}_i := \mathsf{COM}_{\mathsf{T}}.\mathsf{Commit}(\mathsf{pp}^{\mathsf{T}}, A_i, t_i)$ for $i \in \{0,1,2\}$.
  10. Set $\mathsf{HIN}_0 := \big((\mathsf{pp}_i^{\mathsf{X}}, \mathsf{CMX}_i)_{i\in\{0,1\}}, (\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CMT}_0), M, \mathsf{CMA}_0, \mathbb{x}_{\mathsf{dh}}, A_{\mathsf{dh}}\big)$
  11. Set $\mathsf{HIN}_1 := \big((\mathsf{pp}_2^{\mathsf{X}}, \mathsf{CMX}_2), (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CMT}_i)_{i\in\{0,1,2\}}, (\mathsf{CMA}_i)_{i\in\{0,1,2\}}\big)$
  12. Check that
  $$\gamma_0 + \gamma_{\mathsf{dh}} = \mathsf{H}_0^{\mathcal{CH}}(\mathsf{HIN}_0) \ \wedge \ \gamma_1 + \gamma_2 = \mathsf{H}_1^{\mathcal{CH}}(\mathsf{HIN}_1)$$
  13. Check for $i \in \{0,1,2\}$ that
  $$\Sigma_i.\mathsf{Verify}(\mathbb{x}_i, A_i, \gamma_i, \zeta_i) = 1 \ \wedge \ \Sigma_{\mathsf{dh}}.\mathsf{Verify}(\mathbb{x}_{\mathsf{dh}}, A_{\mathsf{dh}}, \gamma_{\mathsf{dh}}, \zeta_{\mathsf{dh}}) = 1$$
  14. Output 1 iff the above checks pass.

3.3 Security Analysis

Theorem 1 (Blindness). *For any PPT adversary $\mathcal{A}$ there exist reductions with running time roughly that of $\mathcal{A}$, such that for sufficiently large λ*

$$\begin{aligned}\mathsf{AdvBlind}_{\mathcal{A}}^{\mathsf{BS}}(\lambda)/2 \le\ & \mathsf{AdvDDH}^{\mathbb{G}}(\lambda) + 3\cdot\epsilon_{\mathsf{hide}}^{\mathsf{COM_X}}(\lambda) + \epsilon_{\mathsf{hide}}^{\mathsf{COM_T}}(\lambda) \\ & + 2(\mathsf{Q}_{\mathsf{H}_0^{\mathcal{CH}}} + \mathsf{Q}_{\mathsf{H}_1^{\mathcal{CH}}})\cdot\delta_{\mathsf{COM_T}}(\lambda) + 2\mathsf{AdvSnd}^{\Pi_{\mathsf{CT}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) \\ & + \mathsf{AdvZK}^{\Pi_{\mathsf{CT}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) + \mathsf{AdvINDCPA}^{\mathsf{LHE}}(\lambda) \\ & + 3\cdot\mathsf{AdvParamIND}^{\mathsf{COM_X}}(\lambda)\end{aligned}$$

where $\delta_{\mathsf{COM}_\mathsf{T}}(\lambda)$ is the unpredictability of COM_T, $\mathsf{Q}_{\mathsf{H}_\mathsf{CT}}, \mathsf{Q}_{\mathsf{H}_0^{\mathcal{CH}}}, \mathsf{Q}_{\mathsf{H}_1^{\mathcal{CH}}}$ are bounds on the number of resp. oracle calls made by $\mathcal{A}$, and $\epsilon_{\mathsf{hide}}^{\mathsf{COM}_\mathsf{T}}$ (resp. $\epsilon_{\mathsf{hide}}^{\mathsf{COM}_\mathsf{X}}$) is the (statistical) distance of COM_T's (resp. COM_X's) parameters from uniform.

$\mathsf{U}_1(\mathsf{vk}, M)$

$(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{sk}^{\mathsf{LHE}}) \leftarrow \mathsf{LHE.Gen}(1^\lambda)$
$r_M \leftarrow \mathcal{R}_{\mathsf{LHE}}$
$\mathsf{ct}_M := \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}, M; r_M)$
$\mathbb{x}_M := (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}_M); \mathbb{w}_M := (\mathsf{sk}^{\mathsf{LHE}}, M, r_M)$
// Prove that ct_M was setup honestly
$\pi_M \leftarrow \Pi_M.\mathsf{Prove}^{\mathsf{H}_M}(\mathsf{crs}_M, \mathbb{x}_M, \mathbb{w}_M)$
return $\mathsf{smsg}_1 := (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}_M, \pi_M)$

$\mathsf{U}_2(\mathsf{vk}, M, \mathsf{smsg}_2)$

parse $\mathsf{smsg}_2 := (\mathsf{ct}_0^*, (\mathsf{CMT}_i^*)_{i\in\{0,1,2\}}, A_1^*, A_2^*, A_{\mathsf{dh}}^*)$
// Randomize CT_i^* to CT_i homomorphically (cf. Eq. (12))
for $i \in \{0,1,2\}$ **do**
  $\rho_i' \leftarrow \mathcal{R}_{\mathsf{RE}}$ // Implicit parameter for RandCT_i
  $\widetilde{\mathsf{CMT}}_i := \mathsf{COM_T.Eval}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CMT}_i^*, \mathsf{RandCT}_i)$
  $s_i' \leftarrow \mathcal{R}_{\mathsf{COM_T}}$
  $\mathsf{CMT}_i := \mathsf{COM_T.Rerand}(\mathsf{pp}^{\mathsf{T}}, \widetilde{\mathsf{CMT}}_i; s_i')$
$A_0^* := \mathsf{LHE.Dec}(\mathsf{sk}^{\mathsf{LHE}}, \mathsf{ct}_0^*)$
// Blind A_i^* homomorphically via $\mathsf{InitBlind}$ (cf. Eq. (13))
$\mathsf{CIN}_0^* := \mathsf{CMT}_0^*$;
$\mathsf{CIN}_1^* := (\mathsf{CMT}_0^*, \mathsf{CMT}_1^*)$;
$\mathsf{CIN}_2^* := \mathsf{CMT}_2^*$
for $i \in \{0,1,2\}$ **do**
  // Implicit parameter for $\mathsf{InitBlind}_i$
  $\mathsf{bst}_i \leftarrow \Sigma_i.\mathsf{BlindSetup}(1^\lambda)$
  $\widetilde{\mathsf{CMA}}_i := \mathsf{COM_T.Eval}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CIN}_i^*, \mathsf{InitBlind}_i)$
  $t_i' \leftarrow \mathcal{R}_{\mathsf{COM_T}}$
  $\mathsf{CMA}_i := \mathsf{COM_T.Rerand}(\mathsf{pp}^{\mathsf{T}}, \widetilde{\mathsf{CMA}}_i; t_i')$
$\mathbb{x}_{\mathsf{dh}} := (G, D_1, D_2, D_3)$
// Blind ZK commitment A_{dh}^* in plain
$\mathsf{bst}_{\mathsf{dh}} \leftarrow \Sigma_{\mathsf{dh}}.\mathsf{Setup}(1^\lambda)$
$A_{\mathsf{dh}} \leftarrow \Sigma_{\mathsf{dh}}.\mathsf{BlindInit}(\mathsf{bst}_{\mathsf{dh}}, \mathbb{x}_{\mathsf{dh}}, A_{\mathsf{dh}}^*)$
// Prepare challenges for blinding
$\mathsf{HIN}_0 := ((\mathsf{pp}_i^{\mathsf{X}}, \mathsf{CMX}_i)_{i\in\{0,1\}}, (\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{CMT}_0), M, \mathsf{CMA}_0, \mathbb{x}_{\mathsf{dh}}, A_{\mathsf{dh}})$
$\delta_{0,\mathsf{dh}} := \mathsf{H}_0^{\mathcal{CH}}(\mathsf{HIN}_0)$
$\overline{\delta}_{0,\mathsf{dh}} := \Sigma_0.\mathsf{BlindChall}^{-1}(\mathsf{bst}_0, \delta_{0,\mathsf{dh}})$
$\delta_{0,\mathsf{dh}}^* := \Sigma_{\mathsf{dh}}.\mathsf{BlindChall}^{-1}(\mathsf{bst}_{\mathsf{dh}}, \overline{\delta}_{0,\mathsf{dh}})$
$\mathsf{HIN}_1 := ((\mathsf{pp}_2^{\mathsf{X}}, \mathsf{CMX}_2), (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CMT}_i)_{i\in\{0,1,2\}}, (\mathsf{CMA}_i)_{i\in\{0,1,2\}})$
$\delta_{1,2} := \mathsf{H}_1^{\mathcal{CH}}(\mathsf{HIN}_1)$
$\overline{\delta}_{1,2} := \Sigma_1.\mathsf{BlindChall}^{-1}(\mathsf{bst}_1, \delta_{1,2})$
$\delta_{1,2}^* := \Sigma_2.\mathsf{BlindChall}^{-1}(\mathsf{bst}_2, \overline{\delta}_{1,2})$
return $\mathsf{smsg}_3 := (\delta_{0,\mathsf{dh}}^*, \delta_{1,2}^*)$

$\mathsf{U}_3(\mathsf{vk}, M, \mathsf{smsg}_4)$

parse $\mathsf{smsg}_4 := ((\gamma_i^*, \zeta_i^*, \mathsf{CT}_i^*, s_i^*)_{i\in\{0,1,2\}}, \gamma_{\mathsf{dh}}^*, \zeta_{\mathsf{dh}}^*, \pi_{\mathsf{CT}}^*)$
req $\gamma_0^* + \gamma_{\mathsf{dh}}^* = \delta_{0,\mathsf{dh}}^* \wedge \gamma_1^* + \gamma_2^* = \delta_{1,2}^*$
$\mathbb{x}_{\mathsf{CT}}^* := (\mathsf{pp}_0^{\mathsf{X}}, \mathsf{CMX}_0, (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CT}_i)_{i\in\{0,1,2\}})$
req $\Pi_{\mathsf{CT}}.\mathsf{Ver}^{\mathsf{H}_{\mathsf{CT}}}(\mathsf{crs}_{\mathsf{CT}}, \mathbb{x}_{\mathsf{CT}}^*, \pi_{\mathsf{CT}}^*) = 1$
// Check transcripts and commitments
for $i \in \{0,1,2\}$ **do**
  req $\mathsf{CMT}_i^* = \mathsf{COM_T.Commit}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}_i^*, s_i^*)$
  req $\Sigma_i.\mathsf{Verify}(\mathbb{x}_i^*, A_i^*, \gamma_i^*, \zeta_i^*) = 1$
req $\Sigma_{\mathsf{dh}}.\mathsf{Verify}(\mathbb{x}_{\mathsf{dh}}^*, A_{\mathsf{dh}}^*, \gamma_{\mathsf{dh}}^*, \zeta_{\mathsf{dh}}^*) = 1$
// Recompute homomorphically computed values in plain
for $i \in \{0,1,2\}$ **do**
  $\mathsf{CT}_i := \mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}_i^*, \rho_i')$
$\mathsf{IN}_0^* := \mathsf{CT}_0^*$;
$\mathsf{IN}_1^* := (\mathsf{CT}_0^*, \mathsf{CT}_1^*)$;
$\mathsf{IN}_2^* := \mathsf{CT}_2^*$
for $i \in \{0,1,2\}$ **do**
  $A_i := \mathsf{InitBlind}_i(\mathsf{IN}_i^*)$
  $\gamma_i := \Sigma_i.\mathsf{BlindChall}(\mathsf{bst}_i, \gamma_i^*)$
  $\widetilde{\zeta}_i := \Sigma_i.\mathsf{BlindResp}(\mathsf{bst}_i, \zeta_i^*)$
$\gamma_{\mathsf{dh}} := \Sigma_{\mathsf{dh}}.\mathsf{BlindChall}(\mathsf{bst}_{\mathsf{dh}}, \gamma_{\mathsf{dh}}^*)$
$\zeta_{\mathsf{dh}} := \Sigma_{\mathsf{dh}}.\mathsf{BlindResp}(\mathsf{bst}_{\mathsf{dh}}, \zeta_{\mathsf{dh}}^*)$
// Translate the received ZK responses to blinded statements
$\mathsf{paramsTS}_0 := \rho_0'$;
$\mathsf{paramsTS}_1 := (\rho_0', \rho_1')$;
$\mathsf{paramsTS}_2 := \rho_2'$
for $i \in \{0,1,2\}$ **do**
  $\zeta_i := \mathsf{TransResp}_{i,\mathsf{paramsTS}_i}(\gamma_i, \widetilde{\zeta}_i)$
// Compose ZK transcripts
$\pi_{\mathsf{dh}} := (A_{\mathsf{dh}}, \gamma_{\mathsf{dh}}, \zeta_{\mathsf{dh}})$
for $i \in \{0,1,2\}$ **do**
  $\pi_i := (A_i, \gamma_i, \zeta_i)$
  // Recompute the commitment randomness
  $\widetilde{s}_i := \mathsf{COM_T.RandEval}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}_i^*, s_i^*, \mathsf{RandCT}_i)$
  $s_i := \mathsf{COM_T.RandRerand}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}_i, \widetilde{s}_i, s_i')$
  $\widetilde{t}_i := \mathsf{COM_T.RandEval}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}_i, s_i', \mathsf{InitBlind}_i)$
  $t_i := \mathsf{COM_T.RandRerand}(\mathsf{pp}^{\mathsf{T}}, A_i, \widetilde{t}_i, t_i')$
return $\sigma := ((\mathsf{CT}_i, \pi_i, s_i, t_i)_{i\in\{0,1,2\}}, \pi_{\mathsf{dh}})$

Fig. 4: The user algorithms for our blind signature scheme. We assume that the signer is stateful and but omit its state for conciseness.

Proof (Sketch). We give a very brief proof sketch. A full proof is given in Supplementary Material D.3. At a high level, we prove blindness by decoupling the interaction of the signing session from the final signature through a number of game hops. The main steps are:

- **Make all statements (trivially) true:** We switch $\mathbb{x}_{\mathsf{dh}}$ to a DDH tuple and set up pp^{X} in hiding mode. Together with the proof π^*_{CT} this ensures that all statements $\mathbb{x}^*_0, \mathbb{x}^*_1, \mathbb{x}^*_2, \mathbb{x}^*_{\mathsf{dh}}$ possess a witness.
- **Program the random oracle:** We pick $\delta^*_{0,\mathsf{dh}}, \delta^*_{1,2}$ and pick γ_i ahead of time, and retroactively program the random oracle. After this change, the user's entire computation in U_2 can be postponed to U_3. In particular, at this point the signer has revealed the (previously partially committed) transcripts of all Σ_i in the plain (for $i \in \{0, 1, 2, \mathsf{dh}\}$).
- **Switch to SHVZK simulation of transcripts:** By transcript blindness, translatability, and SHVZK of Σ_i, we can compute an accepting transcript for $\mathbb{x}_i$ through simulation. The view of the adversary is unchanged. (These steps require the *existence* of a witness, hence the first hop.)
- **Compute statements fresh and independent of M:** Next, we simulate the proof π_M and replace the ciphertexts CT_i by fresh encryptions, and the randomizations of commitments by fresh commitments. At this point, the signatures σ_b are completely independent from the interaction.

□

Theorem 2 (OMUF). *For any PPT adversary $\mathcal{A}$ there exist reductions with running time roughly that of $\mathcal{A}$, such that for sufficiently large λ*

$$\begin{aligned}\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) \leq\ & \mathsf{AdvZK}^{\Pi_{\mathsf{CT}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) + \mathsf{AdvCRS}^{\Pi_M, \mathsf{ExtSetup}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) \\ & + \mathsf{AdvExt}^{\Pi_M, \mathsf{Ext}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) + \epsilon^{\mathsf{COM_T}}_{\mathsf{hide}}(\lambda) \\ & + 2(\mathsf{Q}_{\mathsf{H}^{\mathcal{C}\mathcal{H}}_0} + 1)/p + 2\mathsf{AdvDDH}(\lambda) \\ & + \lceil \log Q_{\mathsf{S}} \rceil \begin{pmatrix} 4\mathsf{AdvINDCPA}^{\mathsf{RE}}(\lambda) \\ + 2(\mathsf{Q}_{\mathsf{H}^{\mathcal{C}\mathcal{H}}_1} + 1)/p \\ + 3\mathsf{AdvParamIND}^{\mathsf{COM_T}}_{\mathcal{R}^3_{\mathsf{COM_T}}}(\lambda) \\ + 7\mathsf{AdvParamIND}^{\mathsf{COM_X}}_{\mathcal{R}^3_{\mathsf{COM_X}}}(\lambda) \\ + 2\ell/p \end{pmatrix}\end{aligned}$$

where $\mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}, \mathsf{Q}_{\mathsf{H}_M}, \mathsf{Q}_{\mathsf{H}^{\mathcal{C}\mathcal{H}}_0}, \mathsf{Q}_{\mathsf{H}^{\mathcal{C}\mathcal{H}}_1}$ are bounds on the number of resp. oracle calls made by $\mathcal{A}$, Q_{S} is the number of signing sessions (started by $\mathcal{A}$), and $\epsilon^{\mathsf{COM_T}}_{\mathsf{hide}}(\lambda)$ is the (statistical) distance of $\mathsf{COM_T}$'s parameters from uniform.

Proof (Sketch). Let us provide a brief proof sketch. The formal proof is given in Supplementary Material D.4. On a high level, our proof follows the proof strategy of unforgeability in [2, Theorem 3.6]. There are two core challenges when adapting their proof technique to our setting:

1. The adversary $\mathcal{A}$ outputs $\ell+1$ forgeries on distinct messages M_k^+ for $k \in [\ell+1]$. Also, the signer does not learn which messages it signed. In contrast, in [2] the adversary outputs a single forgery on a fresh message and the game knows the messages it signed. Thus, to apply the proof strategy of [2], we also need to identify an unsigned message.
2. In contrast to [2], our signing phase proceeds in two steps. Importantly, the adversary $\mathcal{A}$ needs to provide a forgery for each *finished* signing session plus an additional forgery, but it is unrestricted in the number of opened sessions. Thus, we need to carefully control the information we leak to $\mathcal{A}$ in the first signing round.

To deal with the first point, we extract the messages $\mathcal{M}_{\mathsf{S}} := \{M_1, ..., M_{Q_{\mathsf{S}}}\}$ to be signed from the resp. proofs π_M, where Q_{S} is the number of opened signing sessions (including finished ones). Let us denote by $\mathcal{M}_{\mathsf{F}} \subseteq \mathcal{M}_{\mathsf{S}}$ the set of messages with a finished signing session. When the forgery is presented, there are $\ell + 1$ distinct forgeries $\mathcal{M}^+ := \{M_1^+, ..., M_{\ell+1}^+\}$. Because only at most $|\mathcal{M}_{\mathsf{F}}| \leq \ell$ signing sessions have been finished, there must be at least one fresh message $M^+ \in \mathcal{M}^+ \setminus \mathcal{M}_{\mathsf{F}}$. We interpret one such message M^+ as the forgery message.[11] Note that if the above approach is performed naively, it requires extracting a *full* witness $\mathbb{w} = (\mathsf{sk}^{\mathsf{LHE}}, M, r_M)$ for R_M. While extracting scalars (such as $\mathsf{sk}^{\mathsf{LHE}}, r_M$) induces a large overhead, we instead relax the requirement on the extractor to only extract M.[12] This is sufficient, although the proof requires some care (cf. Lemma 6).

The second point is more technical. Observe that even if we fix some message M^+ at the end of the OMUF game as described above, the game does not know M^+ during the simulation. In particular, it might be that M^+ is extracted in some *started* signing session that is never finished (i.e., only the S_1 oracle but not the S_2 oracle is accessed). This is in stark contrast to the setting in [2], where any message passed to the signing oracle cannot be the forgery's message. As we are interested in a tight reduction, we cannot simply guess the message M^+ as in previous works.

Let us observe what happens if we naively apply the proof strategy of [2]. Roughly, [2] introduces additional constraints on the forgery in a tight manner by employing the adaptive partitioning strategy [39]. Eventuelly, after a series of adding and removing constraints on the forgery, the adversary will only succeed if

$$Z_0^+ \in \{x_0 G + x_1 M_j\}_{j \in [Q_{\mathsf{S}}]},$$

where Z_0^+ is the message encrypted in the forgery's ciphertext CT_0^+ associated to message M^+, M_j is the extracted message in the j-th signing session, and x_0, x_1

[11] A similar argument is made by [50, 49] in order to identify an unsigned message. But their approach requires *guessing* the unsigned message in advance in order to puncture the verification key for the guessed message. Here, this message is not known until the adversary's success is evaluated. This is important for a tight security proof, but complicates the security argument. We give more details below.

[12] We note that this is a well-known optimization technique, e.g., used in [45].

are committed in $\mathsf{CMX}_0, \mathsf{CMX}_1$, respectively. This argument employs soundness of π_1 and π_2 as OR-compiled Fiat-Shamir proofs. To get to this point, we will go through a number of *partitionings* of the generated signatures. More specifically, we will consider two types of signatures: those with $z_{2,j} = 0$, and those with $z_{2,j} = 1$. About half of the generated signatures will have $z_{2,j} = 0$, and half will have $z_{2,j} = 1$. We will also guess the bit $\beta = z_2^+$ of the forged signature for M^+. If we play our cards right (and use the soundness of the involved proof systems), this setup enables us to (a) extract from $\mathcal{A}$'s signature for M^+, while (b) being able to change the encrypted values $Z_{0,j}$ in the generated signatures with $z_{2,j} \neq \beta$. This will eventually allow to randomize all $Z_{0,j}$, and the induced conditions on the $\mathcal{A}$'s forgery. As a technical complication, the invariants provided by the involved proof systems will only force $\mathcal{A}$ to *reuse* a previously used $Z_{0,j}$, but not more. We will need to deal with this complication next.

Further, soundness of π_0 and π_{dh} also guarantees that if $(G, D_1, D_2, D_3) \notin \mathscr{L}_{\mathsf{dh}}$, then

$$Z_0^+ = x_0 G + x_1 M^+.$$

Thus, the adversary $\mathcal{A}$ is forced to reuse some message M_j for its forgery which contradicts that M^+ was never signed. Unfortunately, in our context this is does *not* mean that $\mathcal{A}$ fails, as the session where the reused M_j was extracted could have never been finished by $\mathcal{A}$.

A more careful analysis is required. Roughly, for the proof to go through, we need that the signer's ciphertext CT_i^* is not leaked *statistically* within the first round. We ensure this property by only sending a commitment CMT_i^* to CT_i^* (rather than CT_i^* directly). While it is important that CT_i^* is *not* leaked statistically in the first round, we also rely on soundness of the Fiat-Shamir compiled NIPS π_i in the proof.

When the appropriate commitment is statistically hiding, then the former requirement is met, but adaptive soundness of the signature scheme is not guaranteed. On the other hand, if CMT_i^* is only computationally hiding, we can show statistical soundness of the NIPS π_i, but CT_i^* is leaked *computationally* too early to the adversary and the proof strategy fails.

Our key insight is that a dual-mode commitment suffices when combined with a careful analysis. More subtly, while the first message of the Σ-protocols $A_i^* \coloneqq \Sigma_i.\mathsf{Init}(\mathsf{st}_i, \phi^i_{\mathbb{x}_i^*})$ might reveal information about the statement $\mathbb{x}_i^*$, and thus about $\mathbb{x}_i^*$, the ciphertexts CT_i^* are exclusively part of the target vectors $\xi^i(\mathbb{x}_i^*)$ and not $\phi^i_{\mathbb{x}_i^*}$. The property in Remark 1 of Σ_i allows us to control the leaked information in the proof. □

Remark 2 (Security in the NPROM). Except for $\mathsf{H}^{\mathcal{CH}}$, random oracles are only used to generate parameters or within NIZKs. For parameters, we can replace them with CRSs. For the NIZKs, tight CRS-based simulation and extraction trapdoors along with an non-programmable RO are sufficient. The OMUF security proof never programs the $\mathsf{H}^{\mathcal{CH}}$. For blindness, we can avoid programming $\mathsf{H}^{\mathcal{CH}}$ by ensuring that the reduction can "honestly" prove the OR-claims $\mathsf{R}_0 \cup \mathsf{R}_{\mathsf{dh}}$ and $\mathsf{R}_1 \cup \mathsf{R}_2$, and thus generate (perfectly indistinguishable) π_i. For $\mathsf{R}_0 \cup \mathsf{R}_{\mathsf{dh}}$, we can

use the DDH witness. By adding a relaxed proof of knowledge to vk which allows extracting $Z_0 = x_0 G$, we also have a witness for R_1. For details, see Remark 8.

References

1. Abdalla, M., Fouque, P.A., Lyubashevsky, V., Tibouchi, M.: Tightly-secure signatures from lossy identification schemes. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 572–590. Springer, Berlin, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_34
2. Abe, M., Hofheinz, D., Nishimaki, R., Ohkubo, M., Pan, J.: Compact structure-preserving signatures with almost tight security. Journal of Cryptology **36**(4), 37 (Oct 2023). https://doi.org/10.1007/s00145-023-09477-z
3. Abe, M., Jutla, C.S., Ohkubo, M., Pan, J., Roy, A., Wang, Y.: Shorter QA-NIZK and SPS with tighter security. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 669–699. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_23
4. Abe, M., Jutla, C.S., Ohkubo, M., Roy, A.: Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 627–656. Springer, Cham (Dec 2018). https://doi.org/10.1007/978-3-030-03326-2_21
5. Agrawal, S., Kirshanova, E., Stehlé, D., Yadav, A.: Practical, round-optimal lattice-based blind signatures. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 39–53. ACM Press (Nov 2022). https://doi.org/10.1145/3548606.3560650
6. Bader, C., Hofheinz, D., Jager, T., Kiltz, E., Li, Y.: Tightly-secure authenticated key exchange. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 629–658. Springer, Berlin, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-46494-6_26
7. Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 273–304. Springer, Berlin, Heidelberg (May 2016). https://doi.org/10.1007/978-3-662-49896-5_10
8. Baldimtsi, F., Lysyanskaya, A.: On the security of one-witness blind signature schemes. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 82–99. Springer, Berlin, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-42045-0_5
9. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Berlin, Heidelberg (May 2000). https://doi.org/10.1007/3-540-45539-6_18
10. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology **16**(3), 185–215 (Jun 2003). https://doi.org/10.1007/s00145-002-0120-1
11. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Short blind signatures. Journal of computer security **21**(5), 627–661 (2013)
12. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 408–425. Springer, Berlin, Heidelberg (Aug 2014). https://doi.org/10.1007/978-3-662-44371-2_23

13. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In: Desmedt, Y. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Berlin, Heidelberg (Jan 2003). https://doi.org/10.1007/3-540-36288-6_3
14. Boyen, X., Li, Q.: Almost tight multi-instance multi-ciphertext identity-based encryption on lattices. In: Preneel, B., Vercauteren, F. (eds.) ACNS 18International Conference on Applied Cryptography and Network Security. LNCS, vol. 10892, pp. 535–553. Springer, Cham (Jul 2018). https://doi.org/10.1007/978-3-319-93387-0_28
15. Chairattana-Apirom, R., Tessaro, S., Zhu, C.: Pairing-free blind signatures from CDH assumptions. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part I. LNCS, vol. 14920, pp. 174–209. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68376-3_6
16. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO'82. pp. 199–203. Plenum Press, New York, USA (1982). https://doi.org/10.1007/978-1-4757-0602-4_18
17. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO'92. LNCS, vol. 740, pp. 89–105. Springer, Berlin, Heidelberg (Aug 1993). https://doi.org/10.1007/3-540-48071-4_7
18. Chevallier-Mames, B., Joye, M.: A practical and tightly secure signature scheme without hash function. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 339–356. Springer, Berlin, Heidelberg (Feb 2007). https://doi.org/10.1007/11967668_22
19. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 174–187. Springer, Berlin, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5_19
20. Crites, E.C., Komlo, C., Maller, M., Tessaro, S., Zhu, C.: Snowblind: A threshold blind signature in pairing-free groups. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part I. LNCS, vol. 14081, pp. 710–742. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_23
21. del Pino, R., Katsumata, S.: A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 306–336. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_11
22. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Berlin, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1_8
23. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. Journal of Cryptology **1**(2), 77–94 (Jun 1988). https://doi.org/10.1007/BF02351717
24. Fischlin, M.: Round-optimal composable blind signatures in the common reference string model. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 60–77. Springer, Berlin, Heidelberg (Aug 2006). https://doi.org/10.1007/11818175_4
25. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer, Berlin, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_10
26. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015,

Part II. LNCS, vol. 9216, pp. 233–253. Springer, Berlin, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-48000-7_12
27. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_2
28. Fuchsbauer, G., Wolf, M.: Concurrently secure blind schnorr signatures. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part II. LNCS, vol. 14652, pp. 124–160. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58723-8_5
29. Garg, S., Gupta, D.: Efficient round optimal blind signatures. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 477–495. Springer, Berlin, Heidelberg (May 2014). https://doi.org/10.1007/978-3-642-55220-5_27
30. Garg, S., Rao, V., Sahai, A., Schröder, D., Unruh, D.: Round optimal blind signatures. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 630–648. Springer, Berlin, Heidelberg (Aug 2011). https://doi.org/10.1007/978-3-642-22792-9_36
31. Gay, R., Hofheinz, D., Kohl, L., Pan, J.: More efficient (almost) tightly secure structure-preserving signatures. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 230–258. Springer, Cham (Apr / May 2018). https://doi.org/10.1007/978-3-319-78375-8_8
32. Ghadafi, E.: Efficient round-optimal blind signatures in the standard model. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 455–473. Springer, Cham (Apr 2017). https://doi.org/10.1007/978-3-319-70972-7_26
33. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Berlin, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_24
34. Groth, J., Sahai, A.: Efficient noninteractive proof systems for bilinear groups. SIAM J. Comput. **41**(5), 1193–1232 (2012). https://doi.org/10.1137/080725386, https://doi.org/10.1137/080725386
35. Hanzlik, L., Loss, J., Wagner, B.: Rai-choo! Evolving blind signatures to the next level. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 753–783. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_26
36. Hauck, E., Kiltz, E., Loss, J.: A modular treatment of blind signatures from identification schemes. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 345–375. Springer, Cham (May 2019). https://doi.org/10.1007/978-3-030-17659-4_12
37. Hauck, E., Kiltz, E., Loss, J., Nguyen, N.K.: Lattice-based blind signatures, revisited. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 500–529. Springer, Cham (Aug 2020). https://doi.org/10.1007/978-3-030-56880-1_18
38. Hesse, J., Hofheinz, D., Kohl, L.: On tightly secure non-interactive key exchange. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 65–94. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_3
39. Hofheinz, D.: Adaptive partitioning. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 489–518. Springer, Cham (Apr / May 2017). https://doi.org/10.1007/978-3-319-56617-7_17

40. Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 590–607. Springer, Berlin, Heidelberg (Aug 2012). https://doi.org/10.1007/978-3-642-32009-5_35
41. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Berlin, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85174-5_2
42. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 150–164. Springer, Berlin, Heidelberg (Aug 1997). https://doi.org/10.1007/BFb0052233
43. Kastner, J., Loss, J., Xu, J.: The Abe-Okamoto partially blind signature scheme revisited. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part IV. LNCS, vol. 13794, pp. 279–309. Springer, Cham (Dec 2022). https://doi.org/10.1007/978-3-031-22972-5_10
44. Kastner, J., Loss, J., Xu, J.: On pairing-free blind signature schemes in the algebraic group model. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part II. LNCS, vol. 13178, pp. 468–497. Springer, Cham (Mar 2022). https://doi.org/10.1007/978-3-030-97131-1_16
45. Kastner, J., Nguyen, K., Reichle, M.: Pairing-free blind signatures from standard assumptions in the ROM. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part I. LNCS, vol. 14920, pp. 210–245. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68376-3_7
46. Katsumata, S., Reichle, M., Sakai, Y.: Practical round-optimal blind signatures in the ROM from standard assumptions. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part II. LNCS, vol. 14439, pp. 383–417. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8724-5_12
47. Katz, J., Loss, J., Rosenberg, M.: Boosting the security of blind signature schemes. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 468–492. Springer, Cham (Dec 2021). https://doi.org/10.1007/978-3-030-92068-5_16
48. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003. pp. 155–164. ACM Press (Oct 2003). https://doi.org/10.1145/948109.948132
49. Klooß, M., Reichle, M.: Blind signatures from proofs of inequality (2024), https://eprint.iacr.org/2024/XXX
50. Klooß, M., Reichle, M., Wagner, B.: Practical blind signatures in pairing-free groups. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024 (to appear). LNCS (Dec 9–13, 2024)
51. Maurer, U.: Zero-knowledge proofs of knowledge for group homomorphisms. DCC **77**(2-3), 663–676 (2015). https://doi.org/10.1007/s10623-015-0103-5
52. Maurer, U.M.: Abstract models of computation in cryptography (invited paper). In: Smart, N.P. (ed.) 10th IMA International Conference on Cryptography and Coding. LNCS, vol. 3796, pp. 1–12. Springer, Berlin, Heidelberg (Dec 2005). https://doi.org/10.1007/11586821_1
53. Okamoto, T.: Efficient blind and partially blind signatures without random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 80–99. Springer, Berlin, Heidelberg (Mar 2006). https://doi.org/10.1007/11681878_5
54. Pass, R.: Limits of provable security from standard assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 109–118. ACM Press (Jun 2011). https://doi.org/10.1145/1993636.1993652

55. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology **13**(3), 361–396 (Jun 2000). https://doi.org/10.1007/s001450010003
56. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Berlin, Heidelberg (May 1997). https://doi.org/10.1007/3-540-69053-0_18
57. Tessaro, S., Zhu, C.: Short pairing-free blind signatures with exponential security. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 782–811. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_27
58. Waters, B.R.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Berlin, Heidelberg (May 2005). https://doi.org/10.1007/11426639_7

Supplementary Material

A Assumptions

Let $\mathbb{G}$ be a group of prime order p with generator $G \in \mathbb{G}$—implicitly parameterized by the security parameter (cf. Section 2). We formally define the DL, CDH and (Q-)DDH assumptions. Q-DDH is implied by DDH.

Definition 16 (DDH Assumption). *The decisional Diffie-Hellman (DDH) assumption holds in group $\mathbb{G}$ with generator G if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{AdvDDH}^{\mathbb{G}}_{\mathcal{A}}(\lambda) \coloneqq \Big| \Pr[a,b \leftarrow \mathbb{Z}_p : \mathcal{A}(G, aG, bG, (ab)G) = 1] - \Pr[a,b,c \leftarrow \mathbb{Z}_p : \mathcal{A}(G, aG, bG, cG) = 1] \Big| = \mathrm{negl}(\lambda).$$

Definition 17 (Q-DDH Assumption). *The Q-fold decisional Diffie-Hellman (Q-DDH) assumption holds in group $\mathbb{G}$ with generator g if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{AdvQDDH}^{\mathbb{G}}_{\mathcal{A}}(Q,\lambda) \coloneqq \Big| \Pr[a \leftarrow \mathbb{Z}_p, \vec{b} \leftarrow \mathbb{Z}_p^Q : \mathcal{A}(G, aG, (b_iG, (ab_i)G)_{i\in[Q]}) = 1] - \Pr[a \leftarrow \mathbb{Z}_p, \vec{b}, \vec{c} \leftarrow \mathbb{Z}_p^Q : \mathcal{A}(G, aG, (b_iG, c_iG)_{i\in[Q]}) = 1] \Big| = \mathrm{negl}(\lambda).$$

Remark 3. Q-DDH is tightly implied by DDH. Namely, for any PPT adversary $\mathcal{A}$ on Q-DDH, there is a PPT reduction $\mathcal{B}$ with running time roughly that of $\mathcal{A}$, such that $\mathsf{AdvQDDH}^{\mathbb{G}}_{\mathcal{A}}(Q,\lambda) \leq \mathsf{AdvDDH}^{\mathbb{G}}_{\mathcal{B}}(\lambda) + 1/(p-1)$ (cf. [22]).

B Instantiations

In this section, we provide the instantiations of our building blocks, recall the (tight) security of the Fiat–Shamir transformation and relaxed knowledge soundness, and finally, we provide the communication, signature and proof size estimates when instantiating our blind signature.

B.1 ElGamal Encryption: PKE, RE, LHE

Definition 18 (ElGamal encryption). *For a group $\mathbb{G} = \langle G \rangle$ of order p we define the ElGamal encryption scheme with message space $\mathcal{M} \coloneqq \mathbb{G}$, ciphertext space $\mathcal{C} \coloneqq \mathbb{G}^2$, randomness space $\mathcal{R} \coloneqq \mathbb{Z}_p$ as follows:*

- $\mathsf{PKE.Gen}(1^\lambda)$ *samples* $\mathsf{sk} \leftarrow \mathbb{Z}_p$*, computes* $\mathsf{pk} = \mathsf{sk}G$*, and outputs* $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{PKE.Enc}(\mathsf{pk}, M; r)$ *for* $M \in \mathbb{G}$*,* $r \in \mathbb{Z}_p$ *and outputs* $(rG, r\mathsf{pk} + M)$.
- $\mathsf{PKE.Dec}(\mathsf{sk}, (C_0, C_1))$ *outputs* $C_1 - \mathsf{sk}C_0$.

Observe that $(M, r) \mapsto \mathsf{Enc}(\mathsf{pk}, M; r)$ *is linear:* $\mathsf{Enc}(\mathsf{pk}, M; r) + \mathsf{Enc}(\mathsf{pk}, M'; r') = \mathsf{Enc}(\mathsf{pk}, M + M', r + r')$.

Lemma 3 (PKE, RE, LHE). *ElGamal encryption*

- *is perfectly correct, (Definition 29)*
- *has uniform public key, (Definition 11)*
- *has linear encryption, (Definition 12)*
- *is a randomizable encryption scheme with perfect rerandomization indistinguishability and linear randomizability, (Definitions 12, 31 and 33)*
- *is tightly IND-CPA secure under DDH,*[13] *(Definition 30)*
- *is a linearly homomorphic encryption scheme for $\mathcal{M}_{\mathsf{LHE}} = \mathbb{G}$, (Definition 34)*
- *is perfectly homomorphically correct. (Definition 35)*

Proof. All of these are straightforward to check or well-known. Many properties immediately follow from ElGamal encryption being a linear map $\mathsf{Enc}_{\mathsf{pk}} \colon \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ for an $\mathsf{pk} \in \mathcal{PK}$. □

[13] Namely, $\mathsf{AdvINDCPA}^{\mathsf{PKE}}_{\mathcal{A}}(1^\lambda) \leq \mathsf{AdvQDDH}^{\mathbb{G}}_{\mathcal{B}}(1^\lambda)$ by a straightforward reduction.

B.2 Dual-Mode Commitments

We use the same dual-mode commitments as used for Groth–Sahai proofs [34]. For simplicity, we commit scalars $x \in \mathbb{Z}_p$ as xG, and thus only define the commitments for group elements.

Definition 19 (Dual-mode ElGamal encryption). *For a group $\mathbb{G} = \langle G \rangle$ of order p we define dual-mode ElGamal encryption with message space $\mathcal{M} := \mathbb{G}$, ciphertext space $\mathcal{C} := \mathbb{G}^2$, randomness space $\mathcal{R} := \mathbb{Z}_p^2$ as follows:*

- $\mathsf{PKE.Gen}(1^\lambda)$*: security parameter 1^λ, samples $a, b, c \leftarrow \mathbb{Z}_p$, computes* $\mathsf{pk} = \begin{pmatrix} aG & bG \\ acG & abcG \end{pmatrix}$*, outputs* $(\mathsf{pk}, \mathsf{sk} := b)$.
- $\mathsf{PKE.Enc}(\mathsf{pk}, M; r)$*: given message $M \in \mathbb{G}$ and randomness $r \in \mathbb{Z}_p^2$, outputs $C := (0, M) + r^\intercal \mathsf{pk}$.*
- $\mathsf{PKE.Dec}(\mathsf{sk}, C = (C_0, C_1))$*: outputs $M = C^\intercal(-b, 1)$.*

Observe that $(M, r) \mapsto \mathsf{Enc}(\mathsf{pk}, M; r)$ is linear: $\mathsf{Enc}(\mathsf{pk}, M; r) + \mathsf{Enc}(\mathsf{pk}, M'; r') = \mathsf{Enc}(\mathsf{pk}, M + M', r + r')$.

Remark 4. Dual-mode ElGamal encryption PKE can be used as a dual-mode commitment scheme COM for message in $\mathbb{G}$ by setting:

- $\mathsf{COM.Setup}(1^\lambda, \mathsf{hide})$ samples $X \leftarrow \mathbb{Z}_p^{2\times 2}$ s.t. $\det(X) \neq 0$ and $\mathsf{pp} \leftarrow G^X$.
- $\mathsf{COM.Setup}(1^\lambda, \mathsf{bind})$ samples $\mathsf{pp} \leftarrow \mathsf{PKE.Gen}(1^\lambda)$.
- $\mathsf{COM.Commit}(\mathsf{pp}, M) = \mathsf{PKE.Enc}(\mathsf{pp}, M)$.

Lemma 4. *Dual-mode ElGamal encryption used as a commitment for $\mathcal{M}_{\mathsf{COM}} = \mathbb{G}$:*

- *has uniform public key in hiding mode, (Definition 40)*
- *has linear encryption, (Definition 14)*
- *is randomizable as a commitment scheme (and an encryption scheme), with perfect rerandomization indistinguishability and linear randomizability, (Definition 39)*
- *is a linearly homomorphic (as commitment and encryption) scheme for $\mathcal{M}_{\mathsf{LHE}} = \mathbb{G}$,*
- *is perfectly homomorphically correct, (Definition 38)*
- *is a dual-mode commitment scheme (resp. lossy encryption scheme). It is parameter indistinguishable (Definition 41) with advantage $\mathsf{2AdvDDH}^{\mathbb{G}}_{\mathcal{A}}$ by a straightforward reduction. In hiding mode, "encryptions" are uniformly random tuples in $\mathbb{G}^2$. (In particular, the commitment is perfectly hiding (Definition 36) in hiding mode.)*
- *in binding mode is perfectly binding (Definition 37). (Moreover, it is efficiently extractable for $\mathcal{M}_{\mathsf{COM}} = \mathbb{G}$ with extraction trapdoor $\mathsf{td} = \mathsf{sk}$.)*

When $\mathcal{M}_{\mathsf{COM}} = \mathbb{Z}_p$, by committing to mG for message $m \in \mathbb{Z}_p$, then the same properties apply (except that extractability is lost).

Proof. All of these are straightforward to check or well-known. □

B.3 Non-Interactive Proof Systems

We instantiate our proof systems as simple Fiat–Shamir transformations of a respective linear Σ-protocol. For completeness, we recall the proofs of zero-knowledge and soundness, observing that both are tight in the ROM. Moreover, we recall the folklore transformation to achieve straightline extractability for (relaxed) knowledge soundness.

Definition 20 (Fiat–Shamir transformation). *Let $\Sigma = (\mathsf{Setup}, \mathsf{Init}, \mathsf{Resp}, \mathsf{Verify})$ be a Σ-protocol for linear relation $\mathsf{R}_{\phi,\xi}$ and with challenge space $\mathcal{CH}$. Let $\mathsf{H}\colon \{0,1\}^* \to \mathcal{CH}$ a random oracle. The Fiat–Shamir transformation $\mathsf{FS}^{\mathsf{H}}[\Sigma] = (\mathsf{Prove}^{\mathsf{H}}, \mathsf{Ver}^{\mathsf{H}})$ of Σ constructs a non-interactive proof system defined as follows:*

- $\mathsf{Prove}^{\mathsf{H}}(\mathbb{x}, \mathbb{w})$*: given statement–witness pair $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$, computes* $\mathsf{st} \leftarrow \mathsf{Setup}(1^\lambda)$*, $A := \mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})$, $\gamma := \mathsf{H}(\mathbb{x}, A)$, and $\zeta := \mathsf{Resp}(\mathsf{st}, \gamma)$, outputs $\pi := (A, \gamma, \zeta)$.*
- $\mathsf{Ver}^{\mathsf{H}}(\mathbb{x}, (A, \gamma, \zeta))$*: given a purported statement $\mathbb{x}$ and transcript (A, γ, ζ), outputs 1 if* $\mathsf{H}(\mathbb{x}, A) = \gamma$ *and* $\Sigma.\mathsf{Verify}(\mathbb{x}, (A, \gamma, \zeta)) = 1$*, else outputs 0.*

Lemma 5. *Let* $\Sigma = (\mathsf{Setup}, \mathsf{Init}, \mathsf{Resp}, \mathsf{Verify})$ *be a* Σ*-protocol for linear relation* $\mathsf{R}_{\phi,\xi}$ *and with challenge space* $\mathcal{CH}$ *and* $\Pi = \mathsf{FS}^{\mathsf{H}}[\Sigma] = (\mathsf{Prove}^{\mathsf{H}}, \mathsf{Ver}^{\mathsf{H}})$ *its Fiat–Shamir transformation w.r.t. random oracle* H.

1. *Suppose* Σ *is 2-special sound. Then for every (unbounded) adversary* $\mathcal{A}$ *against soundness of* $\mathsf{FS}^{\mathsf{H}}[\Sigma]$*, it holds that*
$$\mathsf{AdvSnd}^{\Pi}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) \leq \frac{\mathsf{Q}_{\mathsf{H}}}{|\mathcal{CH}|}.$$
2. *Suppose* Σ *is perfectly SHVZK with simulator* $\Sigma.\mathsf{Sim}$ *and for every* $\mathbb{x} \in \mathscr{L}_{\mathsf{R}_{(\phi,\xi)}}$ *the output of* $\mathsf{Init}(\phi_{\mathbb{x}})$ *is* δ*-unpredictable, i.e., for all* $x \in \{0,1\}^*$*:* $\Pr[x = A \mid \mathsf{st} \leftarrow \mathsf{Setup}(1^\lambda), A := \mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})] \leq \delta(\lambda)$*. Then there exists a simulator* $\mathsf{SimSetup}, \mathsf{Sim}$ *such that for every adversary* $\mathcal{A}$ *against zero-knowledge of* $\mathsf{FS}^{\mathsf{H}}[\Sigma]$ *which makes at most* Q_{H} *queries to* H *it holds that*
$$\mathsf{AdvZK}^{\Pi,\mathsf{SimSetup},\mathsf{Sim}}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) \leq \mathsf{AdvSHVZK}^{\Sigma,\Sigma.\mathsf{Sim}}(\lambda) + \mathsf{Q}_{\mathsf{H}} \cdot \delta(\lambda) = \mathsf{Q}_{\mathsf{H}} \cdot \delta(\lambda).$$

Proof (Sketch). For Item 1, it suffices to observe that by 2-special soundness, if for any statement $\mathbb{x}$ and first flow message A there exist 2 (or more) challenges $\gamma_1 \neq \gamma_2$ which can be completed into accepting transcripts, then $\mathbb{x} \in \mathscr{L}_{\mathsf{R}_{(\phi,\xi)}}$ (because a witness could be extracted, hence exists). Consequently, for every false statement, for every pair $(\mathbb{x}, A)$ there is (at most) one unique challenge which the verifier could accept. As H is a random oracle, this bad challenge is hit with probability $1/|\mathcal{CH}|$. By a union bound, given Q_{H} queries, the bad event occurs with probability at most $\mathsf{Q}_{\mathsf{H}}/|\mathcal{CH}|$

For Item 2, it suffices to observe that by δ-unpredictability, the adversary $\mathcal{A}$ will not have queried $\mathsf{H}(\mathbb{x}, A)$, except with probability at most $\mathsf{Q}_{\mathsf{H}} \cdot \delta(\lambda)$. Thus, we can instead program $\mathsf{H}(\mathbb{x}, A)$ to $\gamma \leftarrow \mathcal{CH}$ and use the SHVZK simulator to perfectly simulate the respective transcript. Since simulation is perfect, $\mathsf{AdvSHVZK}^{\Sigma,\Sigma.\mathsf{Sim}}(\lambda) = 0$. □

Straightline Extractability via Encryption Straightline extractability (Definition 27) requires to extract a witness $\mathbb{w}$ for $\mathbb{x}$ merely from the proof π for $\mathbb{x}$ and extraction trapdoor td (and perhaps all random oracle queries $\mathcal{Q}$). There is a folklore transformation to achieve this strong guarantee from a sound proof system. We first sketch the idea in a general setting, and then present an instantiation for linear relations.

Remark 5 (The generic transformation). Let R_0 be an arbitrary relation, let PKE be a PKE scheme, and let $\mathsf{R}_{\mathsf{PKE}}$ be the relation for valid ciphertexts under PKE. The idea is to transform a given relation R_0 into a new relation R^*. A transformed statement $\mathbb{x}^*$ consists of the original statement $\mathbb{x}_0$ plus an encryption CT of the to-be-extracted part $\widetilde{\mathbb{w}}_0$ of the (original) witness $\mathbb{w}_0$ in a ciphertext. The CRS for the transformed relation consists of a public key pk plus the CRS for a NIPS for R^*. The trapdoor for the straightline extractor is the secret key sk corresponding to pk. This allows extracting the relevant part $\widetilde{\mathbb{w}}_0$ of the original witness from the transformed statement. In the extreme case, which is the classic transformation, we can let $\widetilde{\mathbb{w}}_0 = \mathbb{w}$ and extract the complete witness. But our notion of relaxed knowledge soundness (Definition 27) allows us to restrict to only a partial witness $\widetilde{\mathbb{w}}_0$

Now, we sketch how relaxed knowledge soundness for certain linear relations is generically achieved by the above template. We do not provide an abstract definition nor a fully detailed proof, and limit our attention to $\mathbb{Z}_p$-linear maps from $\mathbb{Z}_p^m \times \mathbb{G}^n \to \mathbb{Z}_p^{m'} \times \mathbb{G}^{n'}$ and the extraction of witness components in $\mathbb{G}$.[14] Moreover, we concentrate solely on Fiat–Shamir transformation of (canonical) Σ-protocols for linear relations.

Instantiating the transformation.. Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE with uniform public keys, *linear encryption*, and linear relation $\mathsf{R}_{\mathsf{PKE}}$. For simplicity, let $\mathcal{CH} := \mathbb{Z}_p$ be the challenge space for all Σ-protocols in this section.

- Let $\mathsf{R}_0 := \mathsf{R}_{(\phi_0,\xi_0)}$ be linear relation.
- Let $\mathsf{R}_{\mathsf{PKE}} := \mathsf{R}_{(\phi^{\mathsf{PKE}},\xi^{\mathsf{PKE}})}$ be the linear encryption relation for valid ciphertexts.
- Let $\Sigma_0 := (\mathsf{Setup}_0, \mathsf{Init}_0, \mathsf{Resp}_0, \mathsf{Verify}_0)$ be a Σ-protocol for R_0 and with challenge space $\mathcal{CH}$.
- Let $\Sigma_{\mathsf{PKE}} := (\mathsf{Setup}_{\mathsf{PKE}}, \mathsf{Init}, \mathsf{Resp}_{\mathsf{PKE}}, \mathsf{Verify}_{\mathsf{PKE}})$ be a Σ-protocol for linear relation $\mathsf{R}_{\phi^{\mathsf{PKE}},\xi^{\mathsf{PKE}}}$ and with challenge space $\mathcal{CH}$.

[14] Indeed, for *straightline* extraction of scalars, i.e., elements in $\mathbb{Z}_p$, we are not aware of a proof system that is constant size (in terms of group elements, i.e. $\mathcal{O}(\lambda)$) bits for elliptic curves.

– Let

$$\mathsf{R}^* := \left\{ (\mathbb{x}^*, \mathbb{w}^*) \;\middle|\; \begin{array}{r} \mathbb{x}^* = (\mathbb{x}_0, \mathbb{x}_{\mathsf{PKE}}) \\ \mathbb{w}^* = (\mathbb{w}_0, \mathbb{w}_{\mathsf{PKE}}) \\ (\mathbb{x}_0, \mathbb{w}_0) \in \mathsf{R}_0 \\ \mathbb{w}_0 = (\mathbb{w}'_0, \widetilde{\mathbb{w}}_0) \\ \mathbb{x}_{\mathsf{PKE}} = (\mathsf{pk}, \mathsf{CT}) \\ \mathsf{CT} = \mathsf{PKE.Enc}(\mathsf{pk}, \widetilde{\mathbb{w}}_0; r) \\ \mathbb{w}_{\mathsf{PKE}} = (\widetilde{\mathbb{w}}_0, r) \\ \mathsf{pk} \in \mathcal{PK}_{\mathsf{PKE}} \\ r \in \mathcal{R}_{\mathsf{PKE}} \end{array} \right\} \tag{18}$$

where $\widetilde{\mathbb{w}}_0$ is the to-be-extracted part of the witness.

Note that to define R^*, we assume that the vector $\mathbb{w}_0$ can be split into the to-be-extracted part of the witness $\widetilde{\mathbb{w}}_0$ and the remaining part $\mathbb{w}'_0$. Also observe that R^* is an AND-claim which asserts that $\widetilde{\mathbb{w}}_0$ is used *both* in the witness for R_0 and $\mathsf{R}_{\mathsf{PKE}}$. Fortunately, by our assumption that we can split $\mathbb{w}_0$ as $\mathbb{w}'_0, \widetilde{\mathbb{w}}_0$, such an AND-relation can again be expressed by a linear relation, namely by $\mathsf{R}^* = \mathsf{R}_{(\phi^*_{\mathbb{x}}, \xi^*)}$ where

$$\phi^*_{\mathbb{x}}(\mathbb{w}^* = (\mathbb{w}_0, \mathbb{w}_{\mathsf{PKE}})) = \begin{pmatrix} \phi^0_{\mathbb{x}_0}(\mathbb{w}_0 = (\mathbb{w}'_0, \widetilde{\mathbb{w}}_0)) \\ \phi^{\mathsf{PKE}}_{\mathbb{x}_{\mathsf{PKE}}}(\mathbb{w}_{\mathsf{PKE}} = (\widetilde{\mathbb{w}}_0, r)) \end{pmatrix} \quad \text{and} \quad \xi^*(\mathbb{x}^* = (\mathbb{x}_0, \mathbb{x}_{\mathsf{PKE}})) = \begin{pmatrix} \xi_0(\mathbb{x}_0) \\ \xi^{\mathsf{PKE}}(\mathbb{x}_{\mathsf{PKE}}) \end{pmatrix} .$$

Hence, we have a canonical Σ-protocol $\mathbf{\Sigma}^* = \mathbf{\Sigma}_{\phi^*, \xi^*}$ and therefore a NIPS $\mathsf{\Pi}^* = \mathsf{FS}^{\mathsf{H}}[\mathbf{\Sigma}^*]$ for R^*.

Now, following the template, we define a straightline-extractable NIPS $\mathsf{\Pi}_0$ for the relation R_0 as follows:

$\mathsf{\Pi}_0.\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}_0, \mathbb{w}_0)$	$\mathsf{\Pi}_0.\mathsf{Ver}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}_0, \pi_0)$
1 : $\mathsf{pk} := \mathsf{crs}$	1 : $\mathsf{pk} := \mathsf{crs}$
2 : $r \leftarrow \mathcal{R}_{\mathsf{PKE}}$	2 : $(\pi^*, \mathsf{CT}) := \pi_0$
3 : $\mathsf{CT} := \mathsf{PKE.Enc}(\mathsf{pk}, \widetilde{\mathbb{w}}_0; r)$	3 : $\mathbb{x}_{\mathsf{PKE}} := (\mathsf{pk}, \mathsf{CT})$
4 : $\mathbb{x}_{\mathsf{PKE}} := (\mathsf{pk}, \mathsf{CT})$	4 : $\mathbb{x}^* := (\mathbb{x}_0, \mathbb{x}_{\mathsf{PKE}})$
5 : $\mathbb{w}_{\mathsf{PKE}} := (\widetilde{\mathbb{w}}_0, r)$	5 : **return** $\mathsf{\Pi}^*.\mathsf{Ver}^{\mathsf{H}}(\mathbb{x}^*, \pi^*)$
6 : $\mathbb{x}^* := (\mathbb{x}_0, \mathbb{x}_{\mathsf{PKE}})$	
7 : $\mathbb{w}^* := (\mathbb{w}_0, \mathbb{w}_{\mathsf{PKE}})$	
8 : $\pi^* \leftarrow \mathsf{\Pi}^*.\mathsf{Prove}^{\mathsf{H}}(\mathbb{x}^*, \mathbb{w}^*)$	
9 : **return** $\pi_0 := (\pi^*, \mathsf{CT})$	

The above NIPS $\mathsf{\Pi}_0$ is *not* a Fiat–Shamir transformation of a canonical Σ-protocol anymore. Hence, it needs a separate analysis. As a first step, let us recall the properties of $\mathsf{\Pi}^* = \mathsf{FS}^{\mathsf{H}}[\mathbf{\Sigma}^*]$ which were established in Lemma 5: Since $\mathbf{\Sigma}$ is the canonical Σ-protocol for linear relation $\mathsf{R}_{(\phi^*, \xi^*)}$, it is 2-special sound and SHVZK, and by Lemma 5 $\mathsf{\Pi}^*$ is

– **zero-knowledge** with advantage stated in Lemma 5;
– **sound for** R^* with advantage stated in Lemma 5.

Now, let us turn to the NIPS $\mathsf{\Pi}_0$, for we we conclude:

– **Computational Zero-knowledge:** The actual transformation requires that $\mathsf{\Pi}_0.\mathsf{Prove}$ *encrypts* $\widetilde{\mathbb{w}}_0$ under pk and proves well-formedness of the statement as part of R^*. Clearly, this cannot be perfect/statistical zero-knowledge anymore.[15] Assuming $\delta^*(\lambda)$-unpredictability Σ^*, we can bound the advantage of an adversary $\mathcal{A}$ against zero-knowledge of $\mathsf{\Pi}_0$ which makes at most Q_{H} queries to H by

$$\mathsf{AdvZK}^{\mathsf{\Pi}, \mathsf{SimSetup}, \mathsf{Sim}}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) \leq \mathsf{AdvINDCPA}^{\mathsf{PKE}}_{\mathcal{D}}(\lambda) + \mathsf{Q}_{\mathsf{H}} \cdot \delta^*(\lambda) \tag{19}$$

where the running time of $\mathcal{D}$ is roughly that the zero-knowledge experiment with $\mathcal{A}$. This follows by first simulating $\mathsf{\Pi}^*$, using Lemma 5, and then appling IND-CPA to replace the ciphertexts encrypting $\widetilde{\mathbb{w}}_0$ with ciphertext encrypting 0.

[15] We could use extractable dual-mode commitments or lossy encryption to achieve a dual-mode notion of zero-knowledge and relaxed straightline extractability.

– **Extractability:** Let

$$\widetilde{\mathsf{R}}_0 := \{(\mathbb{x}_0, \widetilde{\mathbb{w}}_0) \mid \exists \mathbb{w}_0' \colon (\mathbb{x}_0, \mathbb{w}_0 := (\mathbb{w}_0', \widetilde{\mathbb{w}}_0)) \in \mathsf{R}_0\} \tag{20}$$

be the knowledge relation for R_0. Then, the NIPS Π_0 is *straightline* $\widetilde{\mathsf{R}}_0$*-extractable*. The extractor $(\mathsf{ExtSetup}, \mathsf{Ext})$ is defined as follows:

$\mathsf{ExtSetup}(1^\lambda)$	$\mathsf{Ext}((\mathsf{td}, \mathsf{Q}_\mathsf{H}), \mathbb{x}_0, \pi_0)$
1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$	1 : $\mathsf{sk} := \mathsf{td}$
2 : **return** $(\mathsf{crs} := \mathsf{pk}, \mathsf{td} := \mathsf{sk})$	2 : $(\pi^*, \mathsf{CT}) := \pi_0$
	3 : **return** $\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{CT})$

Since we assume PKE is perfectly correct (by Definition 10), an abort in the extractability experiment for $\tilde{\mathsf{R}}_0$ (see Definition 27) only is equivalent to a soundness break for R^*. Hence, we inherit the tight bound for soundness shown in Lemma 5. We omit a formal proof.

Finally, we note that IND-CPA security of ElGamal encryption tightly reduces to DDH. Thus, we have shown the existence of tightly secure straightline extractable and computationally zero-knowledge non-interactive proof systems for suitable linear relations. Instantiations with ElGamal encryption are limited to extracting witness parts in $\mathcal{M}_{\mathsf{PKE}} = \mathbb{G}$. This suffices for our construction.

B.4 Communication, Signature and Proof Sizes

In Table 3 we overview the (co)domains of our linear maps, which correspond to sizes of commitments and ciphertexts, as well as sizes of witnesses in our relations and Σ-protocols. From these and the protocol description in Figures 3 and 4, we compute the communication and signature sizes in Table 4. For comprehensibility, we list the intermediate sizes in Table 4 used to compute the final proof sizes as well.

Table 3: Linear map codomains and domains for our instantiations.

$\phi^{\mathsf{COM_X}}_{\mathbb{x}_{\mathsf{CMX}_i}}$	$\mathbb{Z}_p^3$	$\to$	$\mathbb{G}^2$	dual-mode ElGamal for $\mathcal{M} = \mathbb{Z}_p$
$\phi^{\mathsf{COM_T}}$	$\mathbb{Z}_p^2 \times \mathbb{G}$	$\to$	$\mathbb{G}^2$	dual-mode ElGamal for $\mathcal{M} = \mathbb{G}$
$\phi^{\mathsf{RE}}_{\mathbb{x}_{\mathsf{RE},i}}$	$\mathbb{Z}_p^2 \times \mathbb{G}$	$\to$	$\mathbb{G}^2$	ElGamal encryption
$\phi^0_{\mathbb{x}_0^*}$	$\mathbb{Z}_p^7 \times \mathbb{G}$	$\to$	$\mathbb{G}^7$	
$\phi^1_{\mathbb{x}_1^*}$	$\mathbb{Z}_p^2 \times \mathbb{G}^2$	$\to$	$\mathbb{G}^5$	
$\phi^2_{\mathbb{x}_2^*}$	$\mathbb{Z}_p^4 \times \mathbb{G}$	$\to$	$\mathbb{G}^5$	
$\phi^{\mathsf{dh}}_{\mathbb{x}_{\mathsf{dh}}}$	$\mathbb{Z}_p$	$\to$	$\mathbb{G}^2$	
ϕ_M	$\mathbb{Z}_p^2 \times \mathbb{G}$	$\to$	$\mathbb{G}^4$	relation R_M
$\phi^{\mathsf{CT}}_{\mathbb{x}}$	$\mathbb{Z}_p^5$	$\to$	$\mathbb{G}^8$	relation R_{CT}

Let us take a look at the final signature in Table 4. For our optimizations, we ignore their positive effects on communication, but note that it provides similar savings.

– The unoptimized signature contains 60 $\mathbb{Z}_p$-elements, 46 of which are opening randomness for the dual-mode commitments $\mathsf{CMT}_i, \mathsf{CMA}_i$ to CT_i and A_i.
– The signature size for $\sigma_{\mathsf{opt-simple}}$ consider the straightforward optimization of not committing all elements in A_i, but only those related to CT_i. More specifically, we can compute the Σ-commitment A_i with fixed randomness 0 for all other parts, and still apply the homomorphic evaluation on those "commitments". Since much of the randomness is now fixed, we only require the optimized costs for $t_{\mathsf{opt},0}, t_{\mathsf{opt},1}, t_{\mathsf{opt},2}$ listed in Table 4 which are much lower than naive ones. Moreover, we can apply a standard trick where we don't include the Σ-commitments A_i in the signature but instead recompute it given the challenge γ_i and the response ζ_i and check its hash against the challenge.
– Finally, we can apply the more fancy optimization of only committing to the message-dependent group element of CT_i. This is in fact sufficient for the security proof of OMUF, as the message of CT_i remains perfectly hidden (and is irrelevant for blindness). This optimization further halves the openings sizes for $(s_i, t_i)_{i \in \{0,1,2\}}$ from 28 to 14 $\mathbb{Z}_p$-elements, as now, only a single group element is

Table 4: Communication in rounds or proof sizes. We count $\mathcal{CH} = \mathbb{Z}_p$ separately for better overview.

Object	$\mathcal{CH}$	$\mathbb{Z}_p$	$\mathbb{G}$	Comment
π_M	1	2	1	
π^*_{CT}	1	4	0	
$\mathsf{pk}^{\mathsf{LHE}}$	0	0	1	ElGamal-LHE with fixed generator
$\mathsf{ct}_M, \mathsf{ct}_0, \mathsf{CT}_i$	0	0	2	Individual size
$(\mathsf{CMT}^*_i)_{i\in\{0,1,2\}}$	0	0	12	Commitments to $\mathsf{CT}_i \in \mathbb{G}^2$
$(s_i)_{i\in\{0,1,2\}}, (s^*_i)_{i\in\{0,1,2\}}$	0	12	0	Randomness for all CMT^*_i
$(t_i)_{i\in\{0,1,2\}}$	0	34	0	Naive randomness for all CMA_i
t_0	0	7	0	Naive randomness for CMA_0
t_1	0	5	0	Naive randomness for CMA_1
t_2	0	5	0	Naive randomness for CMA_2
A_0, A^*_0	0	0	7	
A_1, A^*_1	0	0	5	
A_2, A^*_2	0	0	5	
$A_{\mathsf{dh}}, A^*_{\mathsf{dh}}$	0	0	2	
γ_i, γ^*_i	1	0	0	
ζ_0, ζ^*_0	0	7	1	
ζ_1, ζ^*_1	0	2	2	
ζ_2, ζ^*_2	0	4	1	
$\zeta_{\mathsf{dh}}, \zeta^*_{\mathsf{dh}}$	0	1	0	
smsg_1	1	2	4	$\mathsf{smsg}_1 = (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}_M, \pi_M)$
smsg_2	0	0	26	$\mathsf{smsg}_2 = (\mathsf{ct}^*_0, (\mathsf{CMT}^*_i)_{i\in\{0,1,2\}}, A^*_1, A^*_2, A^*_{\mathsf{dh}})$
smsg_3	2	0	0	$\mathsf{smsg}_3 = (\delta^*_{0,\mathsf{dh}}, \delta^*_{1,2})$
smsg_4	5	30	10	$\mathsf{smsg}_4 = ((\gamma^*_i, \zeta^*_i, \mathsf{CT}^*_i, s^*_i)_{i\in\{0,1,2\}}, \gamma^*_{\mathsf{dh}}, \zeta^*_{\mathsf{dh}}, \pi^*_{\mathsf{CT}})$
σ	4	60	29	$\sigma = ((\mathsf{CT}_i, \pi_i, s_i, t_i)_{i\in\{0,1,2\}}, \pi_{\mathsf{dh}})$
$(\mathsf{smsg}_i)_{i\in\{0,1,2\}}$	8	32	40	Total communication
$t_{\mathsf{opt},0}$	0	4	0	Optimized randomess for CMA^*_0
$t_{\mathsf{opt},1}$	0	8	0	Optimized randomess for CMA^*_1
$t_{\mathsf{opt},2}$	0	4	0	Optimized randomess for CMA^*_2
$\mathsf{smsg}_{\mathsf{opt},1}$	1	2	4	$\mathsf{smsg}_1 = (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}_M, \pi_M)$
$\mathsf{smsg}_{\mathsf{opt},2}$	0	0	26	$\mathsf{smsg}_2 = (\mathsf{ct}^*_0, (\mathsf{CMT}^*_i)_{i\in\{0,1,2\}}, A^*_1, A^*_2, A^*_{\mathsf{dh}})$
$\mathsf{smsg}_{\mathsf{opt},3}$	2	0	0	$\mathsf{smsg}_3 = (\delta^*_{0,\mathsf{dh}}, \delta^*_{1,2})$
$\mathsf{smsg}_{\mathsf{opt},4}$	5	30	10	$\mathsf{smsg}_4 = ((\gamma^*_i, \zeta^*_i, \mathsf{CT}^*_i, s^*_i)_{i\in\{0,1,2\}}, \gamma^*_{\mathsf{dh}}, \zeta^*_{\mathsf{dh}}, \pi^*_{\mathsf{CT}})$
$\sigma_{\mathsf{opt-simple}}$	4	42	10	Optimized $t_{\mathsf{opt},i}$
$(\mathsf{smsg}_i)_{i\in\{0,1,2\}}$	8	32	37	Total communication
$\sigma_{\mathsf{opt-fancy}}$	4	25	10	Optimized $t_{\mathsf{opt},i}$, half-commit to CT_i, use better $\mathsf{COM}_{\mathsf{X}}$

committed per CT_i instead of two. Moreover, we can switch to a better linear dual-mode commitment schemes for $\mathbb{Z}_p$-elements which has randomness in $\mathbb{Z}_p$ instead of $\mathbb{Z}_p^2$, see e.g. [34].[16] This saves another 3 $\mathbb{Z}_p$-elements in witness and thus signature size. This this level of optimization we find that the signature contains 10 group elements and 29 $\mathbb{Z}_p$-elements, and the total communication cost is 37 group elements and 40 $\mathbb{Z}_p$-elements.

– We note there are less trivial optimizations which could be used to further reduce the signature size, but we reach diminishing returns.

C Auxiliary Preliminaries

C.1 Properties of Blind Signatures

In this section we define formal properties of a blind signature scheme.

Definition 21 (Correctness). *A blind signature* BS *is perfectly correct if for all* $(\mathsf{vk}, \mathsf{sk}) \in \mathsf{KeyGen}(1^\lambda)$*, all* H*, and all* $m \in \mathcal{M}$*, it holds that*

$$\Pr[\sigma \leftarrow \langle \mathsf{S}(\mathsf{sk}), \mathsf{U}(\mathsf{vk}, m)\rangle : \mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1] = 1$$

[16] Essentially, we use our dual-mode ElGamal encryption with randomness $(r_0, r_1) \in \mathbb{Z}_p^2$, and commit to $m \in \mathbb{Z}_p$ by setting $r_0 = m$ and choosing $r_1 \leftarrow \mathbb{Z}_p$.

Intuitively, a blind signature scheme should not allow any user to obtain signatures without interacting with the signer. This is modeled by the notion of one-more unforgeability, which states that after completing $\ell - 1$ signing sessions, an adversary can not output valid signatures on ℓ messages. For simplicity, we consider 4-move schemes.

Definition 22 (One-More Unforgeability (OMUF)). *Let* $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{S}, \mathsf{U}, \mathsf{Verify})$ *be a BS scheme. Let* $\mathcal{A}$ *be a PPT adversary. We define* $\mathcal{A}$*'s advantage against the one-more unforgeability (OMUF) of* BS *as*

$$\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) \coloneqq \Pr[\mathsf{Exp}^{\mathsf{OMUF}}_{\mathcal{A}}(\lambda) = 1] \tag{21}$$

with the OMUF experiment

$\mathsf{Exp}^{\mathsf{OMUF}}_{\mathcal{A}}(\lambda)$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$((m_1, \sigma_1), ..., (m_{\ell+1}, \sigma_{\ell+1})) \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda, \mathsf{vk})$
req $c_{\mathsf{F}} \leq \ell \wedge |\{m_1, ..., m_{\ell+1}\}| = \ell + 1$
return $\forall i \in [\ell + 1] : \mathsf{BS.Verify}(\mathsf{vk}, m_i, \sigma_i) = 1$

$\mathcal{O}(j, \mathsf{smsg})$

if $\varrho_j = 3$ **return** $\bot$
$\mathsf{smsg}' \leftarrow \mathsf{S}^j_{\varrho_j}(\mathsf{sk}, \mathsf{smsg})$
// Session finished if $\varrho_j = 2$
if $\varrho_j = 2 \wedge \mathsf{smsg}' \neq \bot$
then $c_{\mathsf{F}} \coloneqq c_{\mathsf{F}} + 1$
$\varrho_j \coloneqq \varrho_j + 1$
return smsg'

where for each session j *the algorithm* S^j *is a fresh stateful instance of the signer,* $\varrho_j \coloneqq 1$ *is initialized as the first round, and* $c_{\mathsf{F}} \coloneqq 0$ *counts the number of successfully finished sessions. Here,* $\varrho_j \in \{1, 2\}$ *is the signer's phase in the* j*-th session.*

Remark 6. Definition 22 is a strong notion of security, also called OMUF-2. A weaker version (OMUF-1) is defined analogously, except that each *started* session is counted towards c_{F}, instead of each *finished* session.

To protect the privacy of users, blind signatures should satisfy blindness. Intuitively, blindness states that a malicious signer cannot link signing interactions to the message-signature pairs. We emphasize that we consider the malicious signer blindness, i.e., the malicious signer can freely choose the public key and arbitrarily deviate from the protocol.

Definition 23 (Blindness). *Let* $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{S}, \mathsf{U}, \mathsf{Verify})$ *be a BS scheme. Let* $\mathcal{A}$ *be a (stateful) PPT adversary. We define* $\mathcal{A}$*'s advantage against the blindness of* BS *as*

$$\mathsf{AdvBlind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) \coloneqq 2\left|\Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{A}}(\lambda) = 1] - 1/2\right| \tag{22}$$

with the blindness experiment

$\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{A}}(\lambda)$

$(\mathsf{vk}, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda)$
$b \leftarrow \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathsf{U}'_1, \mathsf{U}'_2, \mathsf{U}'_3, \mathsf{Fin}}()$
return $b = b'$

$\mathsf{Fin}()$

if $\sigma_0 = \bot \vee \sigma_1 = \bot$
return $(\bot, \bot)$
return $(\sigma_b, \sigma_{b\oplus 1})$

$\mathsf{U}'_1(\hat{b})$

if $\mathsf{smsg}_1^{\hat{b}} \neq \bot$ **return** $\bot$
return $\mathsf{smsg}_1^{\hat{b}} \leftarrow \mathsf{U}_1(\mathsf{vk}, m_{\hat{b}\oplus b})$

$\mathsf{U}'_2(\hat{b}, \mathsf{smsg}_2)$

if $\mathsf{smsg}_1^{\hat{b}} = \bot \vee \mathsf{smsg}_3^{\hat{b}} \neq \bot$ **return** $\bot$
return $\mathsf{smsg}_3^{\hat{b}} \leftarrow \mathsf{U}_1(\mathsf{vk}, m_{\hat{b}\oplus b}, \mathsf{smsg}_2)$

$\mathsf{U}'_3(\hat{b}, \mathsf{smsg}_4)$

if $\mathsf{smsg}_3^{\hat{b}} = \bot \vee \sigma_{\hat{b}} \neq \bot$ **return** $\bot$
$\sigma_{\hat{b}} \leftarrow \mathsf{U}_3(\mathsf{vk}, m_{\hat{b}\oplus b}, \mathsf{smsg}_4)$ *// No output to* $\mathcal{A}$

C.2 Properties of Σ-protocols

Definition 24 (Special honest-verifier zero-knowledge (SHVZK)). *Let Σ be a (blindable) Σ-protocol linear relation $\mathsf{R}_{(\phi,\xi)}$ as in Definition 4. For a given simulator Sim, define experiments*

$\mathsf{Exp}_{\mathsf{real}}(\lambda)$	$\mathsf{Exp}_{\mathsf{ideal}}(\lambda)$
$(\mathbb{x}, \mathbb{w}) \leftarrow \mathcal{A}(1^\lambda)$	$(\mathbb{x}, \mathbb{w}) \leftarrow \mathcal{A}(1^\lambda)$
abort if $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}_{(\phi,\xi)}$	**abort if** $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}_{(\phi,\xi)}$
// Generate fresh transcript	*// Simulate transcript*
$\mathsf{st} \leftarrow \mathsf{Setup}(1^\lambda)$	$\gamma \leftarrow \mathcal{CH}$
$(A, \mathsf{st}) \leftarrow \mathsf{Init}(\mathsf{st}, \phi_{\mathbb{x}})$	$(A, \zeta) \leftarrow \mathsf{Sim}(\mathbb{x}, \gamma)$
$\gamma \leftarrow \mathcal{CH}$	
$\zeta \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma, \mathbb{w})$	
$b \leftarrow \mathcal{A}(A, \gamma, \zeta)$	$b \leftarrow \mathcal{A}(A, \gamma, \zeta)$
return b	**return** b

The distinguishing advantage of $\mathcal{A}$ is

$$\mathsf{AdvSHVZK}^{\Sigma,\mathsf{Sim}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Exp}_{\mathsf{real}}(\lambda) = 1] - \Pr[\mathsf{Exp}_{\mathsf{ideal}}(\lambda) = 1]\right| \ . \tag{23}$$

We say Σ is (perfectly) SHVZK if there exists a PPT Sim such that the two distributions are equal.

C.3 Properties of Non-Interactive Proof Systems

Definition 25 (Correctness). *Let $\Pi = (\mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for a relation R. We call Π* perfectly correct *if for all λ and $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$, it holds that*

$$\Pr\left[\, \pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \mathbb{w}) \ : \ \mathsf{Ver}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \pi) = 1 \,\right] = 1,$$

where the probability is over the choice of crs, H and the randomness of $\mathsf{Prove}, \mathsf{Ver}$.

Our definition of zero-knowledge is by definition straightline and allows the simulator to program crs and H. Moreover, we use a multi-proof variant to avoid (non-tight) hybrid arguments in our reductions.

Definition 26 (Zero-Knowledge). *Let $\Pi = (\mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for a relation R. Let $\mathsf{SimSetup}, \mathsf{Sim}$ be PPT algorithms. Let $\mathcal{A}$ be an Q_{H}-bounded algorithm and let*

$$\mathsf{Real}^{\Pi}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) := \Pr\left[b = 1 \mid \mathsf{crs} \leftarrow \{0,1\}^{\ell(\lambda)}; b \leftarrow \mathcal{A}^{\mathsf{H}, \mathcal{O}_{\mathsf{Prove}}}(1^\lambda, \mathsf{crs})\right]$$
$$\mathsf{Ideal}^{\Pi}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) := \Pr\left[b = 1 \mid (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SimSetup}(1^\lambda); b \leftarrow \mathcal{A}^{\mathsf{H}, \mathcal{O}_{\mathsf{Sim}}}(1^\lambda, \mathsf{crs})\right]$$

Here, $\mathcal{A}$ has (black-box) access to the random oracle H and to an oracle $\mathcal{O}_{\mathsf{Prove}}$ or $\mathcal{O}_{\mathsf{Sim}}$, which are as follows:

- *$\mathcal{O}_{\mathsf{Prove}}(\mathbb{x}, \mathbb{w})$: Return $\perp$ if $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}$. Else, output $\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \mathbb{w})$.*
- *$\mathcal{O}_{\mathsf{Sim}}(\mathbb{x}, \mathbb{w})$: Return $\perp$ if $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}$. Else, output $\pi \leftarrow \mathsf{Sim}^{\mathsf{H}}(\mathsf{td}, \mathbb{x})$.*

Furthermore, the simulator Sim is allowed to program *the random oracle H on any fresh input to H, i.e. if $\mathsf{H}(m)$ has not been queried before, then Sim is free to choose $\mathsf{H}(m)$, else, programming fails. The advantage of Q_{H}-bounded $\mathcal{A}$ against Π and Sim is*

$$\mathsf{AdvZK}^{\Pi,\mathsf{SimSetup},\mathsf{Sim}}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) = \left|\mathsf{Real}^{\Pi}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) - \mathsf{Ideal}^{\Pi}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}})\right| \ . \tag{24}$$

We say that Π is (straightline) AdvZK^{Π}-zero-knowledge *if there exists a simulator $\mathsf{SimSetup}, \mathsf{Sim}$ such that for any Q_{H}-bounded $\mathcal{A}$ its advantage is bounded by $\mathsf{AdvZK}^{\Pi,\mathsf{SimSetup},\mathsf{Sim}}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}}) \leq \mathsf{AdvZK}^{\Pi}_{\mathcal{A}}(\lambda, \mathsf{Q}_{\mathsf{H}})$.*

We define straightline knowledge extractability w.r.t. a separate (not necessarily efficient) knowledge relation $\tilde{\mathsf{R}}$, which is sufficient in our schemes. Importantly, we will avoid extraction of group exponents. Our definition is an indistinguishability-based notion with multi-proof extractability.

Definition 27 (Straightline $\tilde{\mathsf{R}}$-Extractable). *Let $\Pi = (\mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for some relation. Let $\mathsf{ExtSetup}, \mathsf{Ext}$ be PPT algorithms and $\tilde{\mathsf{R}}$ be an associated knowledge relation. Let $\mathcal{A}$ be a Q_H-bounded algorithm and let*

$$\begin{aligned}\mathsf{AdvCRS}^{\Pi,\mathsf{ExtSetup}}_{\mathcal{A}}(\lambda, \mathsf{Q}_\mathsf{H}) \coloneqq &\Pr\left[b = 1 \mid \mathsf{crs} \leftarrow \{0,1\}^{\ell(\lambda)}; b \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda, \mathsf{crs})\right] \\ &- \Pr\left[b = 1 \mid (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda); b \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda, \mathsf{crs})\right]\end{aligned}$$

$$\mathsf{AdvExt}^{\Pi,\mathsf{Ext}}_{\mathcal{A}}(\lambda, \mathsf{Q}_\mathsf{H}) \coloneqq \Pr[\mathsf{ExpExt}^{\Pi,\mathsf{Ext}}_{\mathcal{A}}(\lambda, \mathsf{Q}_\mathsf{H}) = 1]$$

where $\mathsf{ExpExt}^{\Pi,\mathsf{Ext}}_{\mathcal{A}}(\lambda, \mathsf{Q}_\mathsf{H})$ is defined as follows:

1. *Sample $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda)$*
2. *Run $\mathcal{A}^{\mathcal{O}_{\mathsf{Ver}}}(\mathsf{crs})$, where the oracle $\mathcal{O}_{\mathsf{Ver}}$ when queried on $(\mathbb{x}, \pi)$ acts as follows*
   - *If $\mathsf{Ver}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \pi) = 0$, do nothing.*
   - *Else run $u \leftarrow \mathsf{Ext}((\mathsf{td}, \mathcal{Q}), (\mathbb{x}, \pi))$, where $\mathcal{Q}$ is a list containing all random oracle queries of $\mathcal{A}$. If $(\mathbb{x}, u) \notin \tilde{\mathsf{R}}$, then the game returns 1. Else do nothing.*
3. *If $\mathcal{O}_{\mathsf{Ver}}$ has not caused a return of 1, return 0.*

We say that Π is (straightline) $\tilde{\mathsf{R}}$-extractable *if there exists an extractor $(\mathsf{ExtSetup}, \mathsf{Ext})$ such that for any Q_H-bounded $\mathcal{A}$ the CRS distinguishing advantage $\mathsf{AdvCRS}^{\Pi,\mathsf{ExtSetup}}_{\mathcal{A}}(\lambda, \mathsf{Q}_\mathsf{H})$ and extraction failure $\mathsf{AdvExt}^{\Pi,\mathsf{Ext}}_{\mathcal{A}}(\lambda, \mathsf{Q}_\mathsf{H})$ are negligible. If extraction failure is bounded by $\kappa = \kappa(\lambda, \mathsf{Q}_\mathsf{H})$, we call κ the knowledge error and say Π is $(\tilde{\mathsf{R}}, \kappa)$-extractable.*

Note that $\tilde{\mathsf{R}}$-extractability allows us to replace the random CRS with a trapdoored CRS, and then use the trapdoor to obtain a witness for any accepting proof, except with negligible probability. Also observe that we did not require $\tilde{\mathsf{R}}$ to be an NP-relation. Indeed, in our instantiations, $\tilde{\mathsf{R}}$ will not be an efficient relation.

For conciseness, we define soundness through $\tilde{\mathsf{R}}$-extractability.

Definition 28 (Soundness). *Let $\Pi = (\mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for relation $\mathsf{R} \subseteq \mathcal{X} \times \mathcal{W}$. Let $\tilde{\mathsf{R}} = \mathscr{L}_\mathsf{R} \times \{0,1\}^*$ (that is, $\tilde{\mathsf{R}} = \{(\mathbb{x}, \_) \mid \exists \mathbb{w} \colon (\mathbb{x}, \mathbb{w}) \in \mathsf{R}\}$). Then Π is sound for R if it is $\tilde{\mathsf{R}}$-extractable. For concreteness, we also write AdvSnd instead of AdvExt for the respective advantage.*

Note that an "extractor" in Definition 28 is trivial: It can switch to a sound crs and have Ext output any string u as a "witness". So this is indeed implied by typical (multi-proof) notions of soundness.

C.4 Properties of Public-Key Encryption

The following are additional properties we require in our constructions.

Definition 29 (Correctness). *A public-key encryption scheme PKE is* perfectly correct *if for every $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$ and any message m, it holds that $\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m)) = m$.*

Definition 30 (IND-CPA Security). *A public-key encryption scheme PKE is* IND-CPA-secure *if for any PPT adversary $\mathcal{A}$ there exists a negligible function negl such that*

$$\mathsf{AdvINDCPA}^{\mathsf{PKE}}_{\mathcal{A}}(\lambda) \coloneqq |\Pr[\mathsf{Exp}_0(\lambda) = 0] - \Pr[\mathsf{Exp}_1(\lambda) = 0]| \leq \mathrm{negl}(\lambda) \tag{25}$$

with the experiments

$\mathsf{Exp}^{\mathsf{PKE}}_b$	$\mathcal{O}_b(m_0, m_1)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$	**return** $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_b}(1^\lambda, \mathsf{pk})$	
return $b = b'$	

Definition 31 (Randomizable Encryption Scheme). *Let $\mathcal{C}_\mathsf{RE}$ and $\mathcal{R}_\mathsf{RE}$ be vector spaces over $\mathbb{Z}_p$. A rerandomizable encryption (RE) scheme with message space $\mathcal{M}_\mathsf{RE}$, ciphertext space $\mathcal{C}_\mathsf{RE}$, public key space $\mathcal{PK}_\mathsf{RE}$ and randomness space $\mathcal{R}_\mathsf{RE}$ is a tuple of algorithms $\mathsf{RE} = (\mathsf{RE.Gen}, \mathsf{RE.Enc}, \mathsf{RE.Dec}, \mathsf{RE.Rerand})$ defined as follows:*

– $(\mathsf{RE.Gen}, \mathsf{RE.Enc}, \mathsf{RE.Dec})$ *is a PKE scheme according to Definition 10.*
– $\mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}; \rho')$ *takes as input a public key* $\mathsf{pk}^{\mathsf{RE}}$*, a ciphertext* $\mathsf{CT} = \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}, m; \rho)$*, and randomness* $\rho' \in \mathcal{R}_{\mathsf{RE}}$*, outputs the rerandomized ciphertext* $\mathsf{CT}' = \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}, m; \rho + \rho')$*.*

Definition 32 (Linear Rerandomizability). *In the setting of Definition* 31*, an RE* RE *is* linearly rerandomizable *iff for any public key* $\mathsf{pk}^{\mathsf{RE}}$ *and any randomness* $\rho' \in \mathcal{R}_{\mathsf{RE}}$ *there exists some (efficiently computable) linear function* $\psi_{\rho'} : \mathcal{C}_{\mathsf{RE}} \mapsto \mathcal{C}_{\mathsf{RE}}$ *such that* $\mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}; \rho') = \psi_{\rho'}(\mathsf{CT})$*.*

Definition 33 (Rerandomization Indistinguishability). *An RE is* rerandomization indistinguishable *if for any stateful (unbounded) adversary* $\mathcal{A}$ *it holds that*

$$\Pr[\mathsf{Exp}_{\mathsf{fresh}}(\lambda) = 0] = \Pr[\mathsf{Exp}_{\mathsf{rerand}}(\lambda) = 0] \tag{26}$$

where

$\mathsf{Exp}^{\mathsf{PKE}}_{\mathsf{fresh}}$	$\mathsf{Exp}^{\mathsf{PKE}}_{\mathsf{rerand}}$
$(\mathsf{pk}^{\mathsf{RE}}, m^*, \rho^*) \leftarrow \mathcal{A}(1^\lambda)$	$(\mathsf{pk}^{\mathsf{RE}}, m^*, \rho^*) \leftarrow \mathcal{A}(1^\lambda)$
	$\mathsf{CT}^* \coloneqq \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}, m^*; \rho^*)$
$\rho \leftarrow \mathcal{R}_{\mathsf{RE}}$	$\rho' \leftarrow \mathcal{R}_{\mathsf{RE}}$
$\mathsf{CT} \coloneqq \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}, m; \rho)$	$\mathsf{CT} \coloneqq \mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}, \mathsf{CT}^*, \rho')$
$b \leftarrow \mathcal{A}(\mathsf{CT})$	$b \leftarrow \mathcal{A}(\mathsf{CT})$
return b	**return** b

Definition 34 (Linearly Homomorphic Encryption Scheme). *Let* $\mathcal{M}_{\mathsf{LHE}}$ *be a vector space over* $\mathbb{Z}_p$*. A linearly homomorphic encryption (LHE) scheme with message space* $\mathcal{M}_{\mathsf{LHE}}$*, ciphertext space* $\mathcal{C}_{\mathsf{LHE}}$ *and randomness space* $\mathcal{R}_{\mathsf{LHE}}$*is a tuple of algorithms* $\mathsf{LHE} = (\mathsf{LHE.Gen}, \mathsf{LHE.Enc}, \mathsf{LHE.Dec}, \mathsf{LHE.Eval})$ *defined as follows:*

– $(\mathsf{LHE.Gen}, \mathsf{LHE.Enc}, \mathsf{LHE.Dec}, \mathsf{LHE.Rerand})$ *is a RE scheme according to Definition 31.*
– $\mathsf{LHE.Eval}(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}, f) \rightarrow \mathsf{ct}'$ *takes as input a public key* $\mathsf{pk}^{\mathsf{LHE}}$*, ciphertext* $\mathsf{ct} = \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}, m; r)$*, and a linear function* $f : \mathcal{M}_{\mathsf{LHE}} \rightarrow \mathcal{M}^n_{\mathsf{LHE}}$ *for any* $n \in \mathbb{N}$*, and outputs new ciphertexts* $\widetilde{\mathsf{ct}}$ *of the plaintexts* $f(m) \in \mathcal{M}^n_{\mathsf{LHE}}$*.*

Definition 35 (Homomorphic Correctness). *A LHE* LHE *is* homomorphically correct *iff for any keys* $(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{sk}^{\mathsf{LHE}}) \in \mathsf{LHE.Gen}(1^\lambda)$*, any ciphertext* $\mathsf{ct} = \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}, m; r)$*, any linear function* $f : \mathcal{M}_{\mathsf{LHE}} \rightarrow \mathcal{M}^n_{\mathsf{LHE}}$ *for any* $n \in \mathbb{N}$ *it holds that there exists randomness* $\widetilde{r} \in \mathcal{R}^n_{\mathsf{LHE}}$ *such that* $\mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}, f(m); \widetilde{r}) = \widetilde{\mathsf{ct}} \coloneqq \mathsf{LHE.Eval}(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}, f)$*.*

C.5 Properties of Commitments

Definition 36 (Perfect Hiding). *A commitment scheme* COM *is* perfectly hiding *iff for any (unbounded) adversary* $\mathcal{A}$ *it holds that* $\Pr[\mathsf{Exp}^{\mathsf{COM},\mathcal{A}}_{\mathsf{hide}}(\lambda) = 1] = 1/2$ *where*

$\mathsf{Exp}^{\mathsf{COM},\mathcal{A}}_{\mathsf{hide}}$	$\mathcal{O}(m_0, m_1)$
$\mathsf{pp} \leftarrow \mathsf{COM.Setup}(1^\lambda, \mathsf{hide})$	**return** $\mathsf{CM} \leftarrow \mathsf{COM.Commit}(\mathsf{pp}, m_b)$
$b \leftarrow \{0, 1\}$	
$b' \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda, \mathsf{pp})$	
return $b = b'$	

Definition 37 (Perfect Binding). *A commitment scheme* COM *is* perfectly binding *iff for all* $\mathsf{pp} \in \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{bind})$ *for all* $m_0 \neq m_1 \in \mathcal{M}_{\mathsf{COM}}$ *for all* $s_0, s_1 \in \mathcal{R}_{\mathsf{COM}}$ *it holds that* $\mathsf{COM.Commit}(\mathsf{pp}, m_0; s_0) \neq \mathsf{COM.Commit}(\mathsf{pp}, m_1; s_1)$*.*

Definition 38 (Homomorphic Correctness). *Let $\mathcal{M}_{\mathsf{COM}}$ be a commutative group. A COM* COM *is* homomorphically correct *iff for any parameters* $\mathsf{pp} \in \mathcal{PP}_{\mathsf{COM}}$*, any message* $m \in \mathcal{M}_{\mathsf{COM}}$*, any randomness* $s \in \mathcal{R}_{\mathsf{COM}}$*, and any linear function* $f : \mathcal{M}_{\mathsf{COM}} \to \mathcal{M}^n_{\mathsf{COM}}$ *for any* $n \in \mathbb{N}$ *it holds that there exists randomness* $\widetilde{s} \in \mathcal{R}^n_{\mathsf{COM}}$ *such that*

$$\mathsf{COM.Eval}(\mathsf{pp}, \mathsf{CM}, f) = \mathsf{COM.Commit}(\mathsf{pp}, f(m); \widetilde{s}) \ . \tag{27}$$

Definition 39 (Rerandomization Indistinguishability). *A commitment scheme* COM *is* rerandomization indistinguishable *if for any stateful (unbounded) adversary $\mathcal{A}$ it holds that* $\Pr[\mathsf{Exp}^{}_{\mathsf{fresh}}(\lambda) = 0] = \Pr[\mathsf{Exp}^{}_{\mathsf{rerand}}(\lambda) = 0]$ *where*

$\mathsf{Exp}^{\mathsf{COM}}_{\mathsf{fresh}}$	$\mathsf{Exp}^{\mathsf{COM}}_{\mathsf{rerand}}$
$(\mathsf{pp}, m, s^*) \leftarrow \mathcal{A}(1^\lambda)$	$(\mathsf{pp}, m, s^*) \leftarrow \mathcal{A}(1^\lambda)$
	$s' \leftarrow \mathcal{R}_{\mathsf{COM}}$
$s \leftarrow \mathcal{R}_{\mathsf{COM}}$	$s \coloneqq \mathsf{COM.RandRerand}(\mathsf{pp}, m, s^*, s')$
$b \leftarrow \mathcal{A}(s)$	$b \leftarrow \mathcal{A}(s)$
return b	**return** b

Definition 40 (Uniform Parameters). *A commitment scheme* COM *has* uniform parameters *(in hiding mode) if the following distributions are statistically close, i.e., there exists a negligible function* ϵ_{hide} *such that for any unbounded adversary $\mathcal{A}$ it holds that*

$$\sum_{\mathsf{pp}' \in \mathcal{PP}_{\mathsf{COM}}} \left| \Pr[\mathsf{pp} \leftarrow \mathsf{COM.Setup}(1^\lambda, \mathsf{bind}) : \mathsf{pp} = \mathsf{pp}'] - \frac{1}{|\mathcal{PP}_{\mathsf{COM}}|} \right| \leq \epsilon_{\mathsf{hide}}(\lambda) \ . \tag{28}$$

Definition 41 (Parameter Indistinguishability). *Let* COM *be a commitment scheme. For any PPT adversary $\mathcal{A}$ we define the parameter distinguishing advantage as*

$$\mathsf{AdvParamIND}^{\mathsf{COM}}_{\mathcal{A}}(\lambda) \coloneqq \left| \begin{matrix} \Pr[\mathsf{pp} \leftarrow \mathsf{COM.Setup}(1^\lambda, \mathsf{hide}) : \mathcal{A}(\mathsf{pp}) = 1] \\ - \Pr[\mathsf{pp} \leftarrow \mathsf{COM.Setup}(1^\lambda, \mathsf{bind}) : \mathcal{A}(\mathsf{pp}) = 1] \end{matrix} \right| \ . \tag{29}$$

D Supplementary Proofs

In this section, we provide proofs that were omitted from the main body.

D.1 Proof of Lemma 1

Proof. All claims are straightforward and well-known. For completeness, we show blindability: First, observe that A is determined by the values of γ and ζ. Moreover, γ is distributed uniformly in $\mathcal{CH}$ because γ' is uniform, and ζ is distributed uniformly in $\mathcal{W}$ because ζ' is uniform. For the original transcript it holds $A^* = \phi_{\mathbb{x}}(\zeta^*) + \gamma^* \cdot \xi(\mathbb{x})$, and the blinded transcript verifies because

$$\begin{aligned} A &= A^* + \gamma' \cdot \xi(\mathbb{x}) + \phi_{\mathbb{x}}(\zeta') && (30) \\ &= \phi_{\mathbb{x}}(\zeta^*) + \gamma^* \cdot \xi(\mathbb{x}) + \gamma' \cdot \xi(\mathbb{x}) + \phi_{\mathbb{x}}(\zeta') = \phi_{\mathbb{x}}(\zeta) + \gamma \cdot \xi(\mathbb{x}) \ . && (31) \end{aligned}$$

□

D.2 Proof of Lemma 2

Proof (Lemma 2). First, we argue that the validity of statements is invariant under translation, i.e., for any statement $\mathbb{x}^*_i \in \mathscr{L}_{\mathsf{R}_i}$ and any $v \in \mathcal{V}_i$, it holds that $\mathsf{TransStmt}(\mathbb{x}^*, v_i) \in \mathscr{L}_{\mathsf{R}_i}$.
Translating a ciphertext statement with a witness by some (rerandomization) randomness v_i yields a new statement whose witness is the old witness translated by the same randomness. More formally, let $(\mathbb{x}^*_{\mathsf{RE},i} = (\mathsf{pk}^{\mathsf{RE}}_i, \mathsf{CT}^*_i), \mathbb{w}^*_i = (Z_i, \rho^*_i)) \in \mathsf{R}_{\mathsf{RE}_i}$ be some statement and corresponding witness. Let

$v_i \in \mathcal{R}_{\mathsf{RE}_i}$ be some translation (rerandomization randomness). Let $\mathbb{x}_{\mathsf{RE},i} \coloneqq (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CT}_i)$ be the translated statement. Let $\mathbb{w}_i \coloneqq (Z_i, \rho_i^* + v_i)$ be the translated witness. Because of Definition 31 it holds that

$$\mathsf{CT}_i \coloneqq \mathsf{RE.Rerand}(\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CT}_i^*, v_i) = \mathsf{RE.Enc}(\mathsf{pk}_i^{\mathsf{RE}}, Z_i, \rho_i^* + v_i) \tag{32}$$

and thus $(\mathbb{x}_{\mathsf{RE},i}, \mathbb{w}_i) \in \mathsf{R}_{\mathsf{RE}_i}$. We omit the straightforward extension to $\mathsf{TransStmt}_0, \mathsf{TransStmt}_1, \mathsf{TransStmt}_2$.

Now, we show that the two distributions in Definition 7 are equal. First, we show that accepting transcripts are mapped to accepting transcripts. For a given $\rho_0', \rho_1' \in \mathcal{R}_{\mathsf{RE}}$ let

$$\begin{aligned}
\mathsf{TransResp}_{0,\rho_0'}(\gamma, \widetilde{\zeta}) &\coloneqq \widetilde{\zeta} - \gamma \overbrace{(0,0,0,0,0,\rho_0')}^{=v_0} \\
\mathsf{TransResp}_{1,\rho_0',\rho_1'}(\gamma, \widetilde{\zeta}) &\coloneqq \widetilde{\zeta} - \gamma(0,\rho_0',0,\rho_1') \\
\mathsf{TransResp}_{2,\rho_2'}(\gamma, \widetilde{\zeta}) &\coloneqq \widetilde{\zeta} - \gamma(0,0,0,\rho_2')
\end{aligned}$$

be response translation maps. For each language we can verify for each randomness ρ_0', statement $\mathbb{x}_0$ and the translated Σ-response $\zeta_0 \coloneqq \mathsf{TransResp}_{0,\rho_0'}(\gamma_0, \widetilde{\zeta}_0)$ that the respective verification equations pass

$$\begin{aligned}
A_0 = \phi^0_{\mathbb{x}_0}(\zeta_0) + \gamma_0 \xi^0(\mathbb{x}_0) &= \phi^0_{\mathbb{x}_0}(\widetilde{\zeta}_0 - \gamma_0(0,0,0,0,0,\rho_0')) + \gamma_0 \xi^0(\mathbb{x}_0) \\
&= \phi^0_{\mathbb{x}_0}(\widetilde{\zeta}_0) - \gamma_0 \phi^0_{\mathbb{x}_0}((0,0,0,0,0,\rho_0')) + \gamma_0 \xi^0(\mathbb{x}_0) \\
&= \phi^0_{\mathbb{x}_0}(\widetilde{\zeta}_0) - \gamma_0 \phi^0_{\mathbb{x}_0}(0,0,0,0,0,\rho_0') + \gamma_0 \phi^0_{\mathbb{x}_0}(x_0, s_0, x_1, s_1, Z_0, \rho_0) \\
&= \phi^0_{\mathbb{x}_0}(\widetilde{\zeta}_0) + \gamma_0 \phi^0_{\mathbb{x}_0}(x_0, s_0, x_1, s_1, Z_0, \rho_0 - \rho_0') \\
&= \phi^0_{\mathbb{x}_0}(\widetilde{\zeta}_0) + \gamma_0 \phi^0_{\mathbb{x}_0}(x_0, s_0, x_1, s_1, Z_0, \rho_0^*) \\
&= \phi^0_{\mathbb{x}_0}(\widetilde{\zeta}_0) + \gamma_0 \xi^0(\mathbb{x}_0^*) \ .
\end{aligned}$$

For the second language we can verify for each randomness ρ_0', ρ_1', statement $\mathbb{x}_1$ and the translated Σ-response $\zeta_1 \coloneqq \mathsf{TransResp}_{1,\rho_0',\rho_1'}(\gamma_1, \widetilde{\zeta}_1)$ that the respective verification equations pass

$$\begin{aligned}
A_1 = \phi^1_{\mathbb{x}_1}(\zeta_1) + \gamma_1 \xi^1(\mathbb{x}_1) &= \phi^1_{\mathbb{x}_1}(\widetilde{\zeta}_1 - \gamma_1(0,\rho_0',0,\rho_1')) + \gamma_1 \xi^1(\mathbb{x}_1) \\
&= \phi^1_{\mathbb{x}_1}(\widetilde{\zeta}_1) - \gamma_1 \phi^1_{\mathbb{x}_1}(0,\rho_0',0,\rho_1') + \gamma_1 \phi^1_{\mathbb{x}_1}(Z_0, \rho_0, Z_1, \rho_1) \\
&= \phi^1_{\mathbb{x}_1}(\widetilde{\zeta}_1) - \gamma_1 \phi^1_{\mathbb{x}_1}(Z_0, \rho_0^*, Z_1, \rho_1^*) \\
&= \phi^1_{\mathbb{x}_1}(\widetilde{\zeta}_1) + \gamma_1 \xi^1(\mathbb{x}_1^*) \ .
\end{aligned}$$

For the third language we can verify for each randomness ρ_2', statement $\mathbb{x}_2$ and the translated Σ-response $\zeta_2 \coloneqq \mathsf{TransResp}_{2,\rho_2'}(\gamma_2, \widetilde{\zeta}_2)$ that the respective verification equations pass

$$\begin{aligned}
A_2 = \phi^2_{\mathbb{x}_2}(\zeta_2) + \gamma_2 \xi^2(\mathbb{x}_2) &= \phi^2_{\mathbb{x}_2}(\widetilde{\zeta}_2 - \gamma_2(0,0,0,\rho_2')) + \gamma_2 \xi^2(\mathbb{x}_2) && (33) \\
&= \phi^2_{\mathbb{x}_2}(\widetilde{\zeta}_2) - \gamma_2 \phi^2_{\mathbb{x}_2}(0,0,0,\rho_2') + \gamma_2 \phi^2_{\mathbb{x}_2}(x_2, s_2, Z_2, \rho_2) && (34) \\
&= \phi^2_{\mathbb{x}_2}(\widetilde{\zeta}_2) - \gamma_2 \phi^2_{\mathbb{x}_2}(x_2, s_2, Z_2, \rho_2 - \rho_2') && (35) \\
&= \phi^2_{\mathbb{x}_2}(\widetilde{\zeta}_2) + \gamma_2 \phi^2_{\mathbb{x}_2}(x_2, s_2, Z_2, \rho_2^*) && (36) \\
&= \phi^2_{\mathbb{x}_2}(\widetilde{\zeta}_2) + \gamma_2 \xi^2(\mathbb{x}_2^*) \ . && (37)
\end{aligned}$$

Finally, since ζ_i^* is uniformly distributed, so is $\zeta_i^* - v_i$. The challenge distribution γ_i is unchanged. Lastly, there is a unique accepting choice of A_i. Hence, the distribution of a transcript (A_i, γ_i, ζ_i) is identical to a fresh transcript. □

D.3 Proof of Blindness (Theorem 1)

Theorem 1 (Blindness). *For any PPT adversary $\mathcal{A}$ there exist reductions with running time roughly that of $\mathcal{A}$, such that for sufficiently large λ*

$$\begin{aligned}
\mathsf{AdvBlind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)/2 \leq\ & \mathsf{AdvDDH}^{\mathbb{G}}(\lambda) + 3 \cdot \epsilon^{\mathsf{COM_X}}_{\mathsf{hide}}(\lambda) + \epsilon^{\mathsf{COM_T}}_{\mathsf{hide}}(\lambda) \\
& + 2(\mathsf{Q}_{\mathsf{H}_0^{\mathcal{CH}}} + \mathsf{Q}_{\mathsf{H}_1^{\mathcal{CH}}}) \cdot \delta_{\mathsf{COM_T}}(\lambda) + 2\mathsf{AdvSnd}^{\Pi_{\mathsf{CT}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}})
\end{aligned}$$

$$+ \mathsf{AdvZK}^{\Pi_{\mathsf{CT}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) + \mathsf{AdvINDCPA}^{\mathsf{LHE}}(\lambda)$$
$$+ 3 \cdot \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}(\lambda)$$

where $\delta_{\mathsf{COM}_{\mathsf{T}}}(\lambda)$ is the unpredictability of $\mathsf{COM}_{\mathsf{T}}$, $\mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}, \mathsf{Q}_{\mathsf{H}_0^{\mathcal{CH}}}, \mathsf{Q}_{\mathsf{H}_1^{\mathcal{CH}}}$ are bounds on the number of resp. oracle calls made by $\mathcal{A}$, and $\epsilon_{\mathsf{hide}}^{\mathsf{COM}_{\mathsf{T}}}$ (resp. $\epsilon_{\mathsf{hide}}^{\mathsf{COM}_{\mathsf{X}}}$) is the (statistical) distance of $\mathsf{COM}_{\mathsf{T}}$'s (resp. $\mathsf{COM}_{\mathsf{X}}$'s) parameters from uniform.

We first give sketch the proof idea. Informally, to argue blindness, we need to decouple the transcripts of the signing sessions from the messages and final signatures. There are three sources from which information about the signing session could leak:

- The first signing message smsg_1 contains a ciphertext ct_M of the message M.
- The final signature contains (rerandomized) ciphertexts CT_i of some exponents z_i which could be linked to the signing session,
- As per Fiat-Shamir the Σ-challenges γ_0, γ_1 are computed as a hash of (commitments to blinded) Σ-commitments A_i which in turn depend on parts A_i^* of the signing message smsg_2.

We note that the signing transcript contains the message M information-theoretically, hence we can only achieve blindness against computationally bounded adversaries.

At a high level, the proof of blindness proceeds as follows:

- **Make all statements (trivially) true:** First, we ensure that $\mathbb{x}_{\mathsf{dh}}$ is a DDH tuple and pp^{X} is in hiding mode. Together with the proof π_{CT}^* this ensures that all statements $\mathbb{x}_0^*, \mathbb{x}_1^*, \mathbb{x}_2^*, \mathbb{x}_{\mathsf{dh}}^*$ possess a witness (or the user rejects and returns $\perp$).
- **Program the random oracle:** We send random challenges $\delta_{0,\mathsf{dh}}^*, \delta_{1,2}^*$, and pick γ_i ahead of time, and then retroactively program the random oracle. This brings us into a situation where the user's entire computation in U_2 can be postponed to U_3. In particular, at this point the signer reveals the (previously partially committed) transcripts of all Σ_i in the plain (for $i \in \{0, 1, 2, \mathsf{dh}\}$).
- **Switch to SHVZK simulation of transcripts:** Next, we can apply the transcript blindness of the Σ_i, to compute a fresh transcripts (instead of blinding the interaction), and then translatability, to change the statement from $\mathbb{x}_i^*$ to $\mathbb{x}_i$. To apply the notions to some $\mathbb{x}_i^*$, we crucially require a witness to exist, which is ensured by the very first hop we made. Finally, we apply SHVZK simulation to efficiently simulate the transcripts instead of bruteforcing a witness. We note that the inefficient intermediate steps are perfectly indistinguishable, hence we can afford an inefficient reduction which bruteforces the witnesses.
- **Compute statements fresh and independent of M:** At this point, we simulate all the Σ-protocol outputs of the user, and only need to decouple the statements. Now, we simulate π_M, and replace $\mathsf{Enc}(\mathsf{pk}^{\mathsf{LHE}}, M_b)$ by $\mathsf{Enc}(\mathsf{pk}^{\mathsf{LHE}}, 0)$. Then we switch pp^{X} into binding mode and use the well-formedness proof π_{CT}^*, so that we can extract $Z_0 = x_0 G$ and, by perfect rerandomization, replace $\mathsf{CT}_i := \mathsf{Rerand}(\mathsf{pk}, \mathsf{CT}_i^*; \rho_i')$ with fresh encryptions $\mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, x_0 G)$ for $i = 0, 1$ and $\mathsf{CT}_2 \leftarrow \mathsf{Enc}(\mathsf{pk}, 0)$. Similarly, we can replace the randomized commitment openings s_i and t_i by freshly sampled (commitment) openings. At this point, the signatures are completely independent from the interactions.

Remark 7 (Blindness in CRS + NPROM model). If we knew a witness for both OR-proofs, we could honestly prove them instead by programming the challenges and relying on perfect SHVZK. By using the DDH witness for $\mathbb{x}_{\mathsf{dh}}$, the first OR-proof is already efficiently provable. However, the proof for $\mathbb{x}_1$ or $\mathbb{x}_2$ has no known witness. This can be remedied by modifying the protocol, e.g., a trivial modification is to add an additional $\mathbb{x}_{\mathsf{dh},2}$ there. (We stress that this cannot share $\mathbb{x}_{\mathsf{dh}}$ due to the OMUF proof.) For such a modification, blindness then holds in the non-programmable random oracle model if we setup the parameters $\mathsf{pp}^{\mathsf{X}}, \mathsf{pp}^{\mathsf{T}}$ via a CRS, and use Π_{CT}, Π_M in the CRS + NPROM model (which are easy to obtain based from the current NIPS).

Proof (Theorem 1). Let $\mathcal{A}$ be a adversary against the blindness of the signature scheme BS that makes at most $\mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}$ queries to the random oracle H_{CT}, at most $\mathsf{Q}_{\mathsf{H}_M}$ queries to the random oracle H_M, and at most $\mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}}$ queries to $\mathsf{H}_0^{\mathcal{CH}}$ and $\mathsf{H}_1^{\mathcal{CH}}$ combined. We gradually modify the oracles $\mathcal{O}_0, \mathcal{O}_1$ of the signature's blindness game such that in the end the adversary cannot win the blindness game information-theoretically. Inefficient games are marked with *. Let ϵ_i denote the adversary's success probability in the i-th game. We assume that adversarial responses are well-formed, as otherwise the user would abort and output $\perp$ as its signature.

Game 0 (Original game): This is the blindness game with a random bit $b \leftarrow \{0,1\}$ according to Definition 23. We let $\epsilon_0(\lambda) \coloneqq \Pr[\mathsf{Exp}^{\mathsf{blind}}_{\mathcal{A}}(\lambda) = 1]$ denote the adversary's original success probability.
Game 1 (DDH parameters): In this game we set up the parameters $(D_1, D_2, D_3) \coloneqq (aG, bG, abG)$ as a DDH triple. This change is justified by the DDH assumption (Definition 16). Thus

$$|\epsilon_1 - \epsilon_0| \leq \mathsf{AdvDDH}^{\mathbb{G}}_{\mathcal{R}_{\mathsf{DDH}}}(\lambda) \ .$$

Game 2 (Switch $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} to hiding mode): In this game we set up $\mathsf{pp}^{\mathsf{X}}_i \leftarrow \mathsf{COM}_{\mathsf{X}}.\mathsf{Setup}(1^\lambda, \mathsf{hide})$ and $\mathsf{pp}^{\mathsf{T}} \leftarrow \mathsf{COM}_{\mathsf{X}}.\mathsf{Setup}(1^\lambda, \mathsf{hide})$. Because of the uniform parameters in hiding mode (Definition 40), we have that

$$|\epsilon_2 - \epsilon_1| \leq 3 \cdot \epsilon^{\mathsf{COM}_{\mathsf{X}}}_{\mathsf{hide}}(\lambda) + \epsilon^{\mathsf{COM}_{\mathsf{T}}}_{\mathsf{hide}}(\lambda) \ .$$

Game 3 (Abort on RO queries): So far, in phase U_2 the commitment CMA_i is computed as a rerandomization $\mathsf{CMA}_i \coloneqq \mathsf{COM}.\mathsf{Rerand}(\mathsf{pp}^{\mathsf{T}}, \widetilde{\mathsf{CMA}}_i; t'_i)$. At the end we abort (output a random bit) if the adversary queried the RO oracle on input HIN_0 and HIN_1 before obtaining the final signatures from the FIN oracle. Recall that the values $\mathsf{HIN}_0, \mathsf{HIN}_1$ contain the rerandomized commitments CMA_i which are $\delta_{\mathsf{COM}_{\mathsf{T}}}$-unpredictable. By a simple union bound[17] it follows that the adversary's probability to query the RO on HIN_0 or HIN_1 is at most $2\mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}}\delta_{\mathsf{COM}_{\mathsf{T}}}(\lambda)$. Consequently,

$$|\epsilon_3 - \epsilon_2| \leq 2 \cdot \mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}} \cdot \delta_{\mathsf{COM}_{\mathsf{T}}}(\lambda) \ .$$

Game 4 (Sample challenges uniformly and program RO): In phase U_2 instead of sampling $\delta_{0,\mathsf{dh}} \coloneqq \mathsf{H}^{\mathcal{CH}}_0(\mathsf{HIN}_0)$ and $\delta_{1,2} \coloneqq \mathsf{H}^{\mathcal{CH}}_1(\mathsf{HIN}_1)$ from the random oracle and *un*blinding them to obtain $\delta^*_{0,\mathsf{dh}}, \delta^*_{1,2}$, we instead sample $\delta^*_{0,\mathsf{dh}}, \delta^*_{1,2} \leftarrow \mathcal{CH}$ uniformly and blind them to obtain the values $\delta_{0,\mathsf{dh}}, \delta_{1,2}$ which are programmed into the random oracle in phase U_3. Recall from Definition 5 that the blinding, $\Sigma.\mathsf{BlindChall}(\mathsf{bst}, \cdot)$, is the inverse of the unblinding, $\Sigma.\mathsf{BlindChall}^{-1}(\mathsf{bst}, \cdot)$. Thus, all values have the same distribution as in the previous game, i.e.,

$$\epsilon_4 = \epsilon_3 \ .$$

More formally,

$$\begin{aligned}
\overline{\delta}_{0,\mathsf{dh}} &\coloneqq \Sigma_{\mathsf{dh}}.\mathsf{BlindChall}(\mathsf{bst}_{\mathsf{dh}}, \delta^*_{0,\mathsf{dh}}) & \overline{\delta}_{1,2} &\coloneqq \Sigma_2.\mathsf{BlindChall}(\mathsf{bst}_2, \delta^*_{1,2}) \\
\delta_{0,\mathsf{dh}} &\coloneqq \Sigma_0.\mathsf{BlindChall}(\mathsf{bst}_0, \overline{\delta}_{0,\mathsf{dh}}) & \delta_{1,2} &\coloneqq \Sigma_1.\mathsf{BlindChall}(\mathsf{bst}_1, \overline{\delta}_{1,2}) \\
\mathsf{H}^{\mathcal{CH}}_0(\mathsf{HIN}_0) &\coloneqq \delta_{0,\mathsf{dh}} & \mathsf{H}^{\mathcal{CH}}_1(\mathsf{HIN}_1) &\coloneqq \delta_{1,2} \ .
\end{aligned}$$

Note also, that $\delta_{0,\mathsf{dh}} = \gamma_0 + \gamma_{\mathsf{dh}}$ and $\delta_{1,2} = \gamma_1 + \gamma_2$ where γ_i is computed as a blinding of γ^*_i in phase U_3. Alternatively, we sample $\gamma_i \leftarrow \mathcal{CH}$ (and as before $\delta^*_{0,\mathsf{dh}}, \delta^*_{1,2} \leftarrow \mathcal{CH}$), and set

$$\delta_{0,\mathsf{dh}} \coloneqq \gamma_0 + \gamma_{\mathsf{dh}} \quad \text{and} \quad \delta_{1,2} \coloneqq \gamma_1 + \gamma_2 \ .$$

Note that it is not necessary to generate a blind state bst_i for the Σ_i anymore.
Game* 5 (Abort if $\mathbb{x}^*_{\mathsf{CT}} \notin \mathscr{L}_{\mathsf{R}_{\mathsf{CT}}}$): We abort if the adversary (in phase S_2) submits π^*_{CT} such that $\Pi_{\mathsf{CT}}.\mathsf{Ver}^{\mathsf{H}_{\mathsf{CT}}}(\mathsf{crs}_{\mathsf{CT}}, \mathbb{x}^*_{\mathsf{CT}}, \pi^*_{\mathsf{CT}}) = 1$ yet $\mathbb{x}^*_{\mathsf{CT}} \notin \mathscr{L}_{\mathsf{R}_{\mathsf{CT}}}$ (checking this is inefficient). This happens with small probability because of the statistical soundness of Π_{CT} and perfect correctness of RE and $\mathsf{COM}_{\mathsf{X}}$. Suppose the adversary $\mathcal{A}_2$ produces such invalid CT^*_i with a valid proof π^*_{CT}, then we can define an inefficient reduction $\mathcal{R}$ that wins the statistical soundness game of Π_{CT} with the same probability. Thus,

$$|\epsilon_5 - \epsilon_4| \leq \mathsf{AdvSnd}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}_{\mathsf{snd}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}})$$

for a straightforward reduction $\mathcal{R}_{\mathsf{snd}}$ that runs in approximately the same time as $\mathcal{A}$.
Game* 6 (Abort if $\mathbb{x}^*_i \notin \mathscr{L}_{\mathsf{R}_i}$ for $i \in \{0,1,2,\mathsf{dh}\}$): In this game we abort if $\mathbb{x}^*_i \notin \mathscr{L}_{\mathsf{R}_i}$ for any $i \in \{0,1,2,\mathsf{dh}\}$. We argue that this never happens. First, consider R_{dh}. Because the DDH tuple (D_1, D_2, D_3) is a valid DDH tuple since Game 1, the statement $\mathbb{x}_{\mathsf{dh}}$ is always in $\mathscr{L}_{\mathsf{R}_{\mathsf{dh}}}$.
Next, consider R_0. For any statement $\mathbb{x}^*_0$ let Z_0 be the plaintext encrypted by CT^*_0, i.e, there exists some encryption randomness ρ^*_0 such that $\mathsf{CT}^*_0 = \mathsf{RE}.\mathsf{Enc}(\mathsf{pk}^{\mathsf{RE}}, Z_0; \rho^*_0)$. Let $x_1 \coloneqq 0$ and let $x_0 \coloneqq \mathsf{DLOG}(Z_0)$.

[17] The factor of 2 accounts for the fact that there are two messages for which it has to abort on corresponding HIN_o and HIN_1.

Because the parameters $\mathsf{pp}_i^{\mathsf{X}}$ are set up in hiding mode, for any commitment $\mathsf{CMX}_0^*, \mathsf{CMX}_1^*$ and any plaintexts x_0, x_1 there exists commitment randomness s_0^*, s_1^* such that $\mathsf{CMX}_0^* = \mathsf{COM.Commit}(\mathsf{pp}_0^{\mathsf{X}}, x_0; s_0^*)$ and $\mathsf{CMX}_1^* = \mathsf{COM.Commit}(\mathsf{pp}_1^{\mathsf{X}}, x_1; s_1^*)$. Thus, for any $\mathbb{x}_0^*$ there exists a witness $\mathbb{w}_0^* := (x_0, s_0^*, x_1, s_1^*, Z_0; \rho_0^*)$ such that $(\mathbb{x}_0^*, \mathbb{w}_0^*) \in \mathsf{R}_0$.
A similar argument implies that for any $\mathbb{x}_2^*$ there exists a witness $\mathbb{w}_2^* = (x_2, s_2^*, Z_2; \rho_2^*)$ such that $(\mathbb{x}_2^*, \mathbb{w}_2^*) \in \mathsf{R}_2$.
Finally, consider R_1. Due to Game* 5 we can assume that $\mathbb{x}_{\mathsf{CT}}^* \notin \mathscr{L}_{\mathsf{R}_{\mathsf{CT}}}$ which implies that the ciphertexts $\mathsf{CT}_0^*, \mathsf{CT}_1^*$ contain the same plaintext $x_0 G$. Hence, for any $\mathbb{x}_1^*$ there exists a witness $\mathbb{w}_1^* = (x_0 G, \rho_0^*, x_0 G, \rho_1^*)$ such that $(\mathbb{x}_1^*, \mathbb{w}_1^*) \in \mathsf{R}_1$. Thus,

$$\epsilon_6 = \epsilon_5 \ .$$

Game* 7 (Compute fresh $(A_i, \gamma_i, \widetilde{\zeta}_i)$ for $\mathbb{x}_i^*$): Now, in phase U_3 we (inefficiently) bruteforce a valid witness $\mathbb{w}_i^*$ such that $(\mathbb{x}_i^*, \mathbb{w}_i^*) \in \mathsf{R}_i$. This is possible because we have shown in Game* 6 that $\mathbb{x}_i^* \in \mathscr{L}_{\mathsf{R}_i}$ for all $i \in \{0, 1, 2, \mathsf{dh}\}$.
Instead of computing the (blinded) transcript (in phase U_3) as $A_i := \Sigma_i.\mathsf{BlindInit}(\mathsf{bst}_i, \mathbb{x}_i^*, A_i^*)$, $\gamma_i := \Sigma_i.\mathsf{BlindChall}(\mathsf{bst}_i, \gamma_i^*)$ and $\widetilde{\zeta}_i := \Sigma_i.\mathsf{BlindResp}(\mathsf{bst}_i, \zeta_i^*)$, we compute a fresh ZK commitment $A_i := \Sigma_i.\mathsf{Init}(\mathsf{st}_i, \phi^i_{\mathbb{x}_i^*})$, the challenge $\gamma_i \leftarrow \mathcal{CH}$ and the response $\widetilde{\zeta}_i := \Sigma_i.\mathsf{Resp}(\mathsf{st}_i, \gamma_i, \mathbb{w}_i^*)$. This step is justified by the perfect blindness of the Σ-protocol (Definition 6). Let $\mathcal{R}$ be a reduction that simulates the blindness challenger and plays the blindness game with $\mathcal{A}$. After obtaining the signing messages $\mathsf{smsg}_2, \mathsf{smsg}_4$ (containing $A_i^*, \gamma_i^*, \zeta_i^*$) the reduction forwards $A_i^*, \gamma_i^*, \zeta_i^*$ to the Σ-protocol blindness challenger to obtain either a blinded transcript or a fresh transcript $(A_i, \gamma_i, \widetilde{\zeta}_i)$. Thus,

$$\epsilon_7 = \epsilon_6 \ .$$

Game* 8 (Compute fresh (A_i, γ_i, ζ_i) for $\mathbb{x}_i$): Recall that in phase U_3 the final statements $\mathbb{x}_i$ are fixed. Now (in phase U_3) we bruteforce a valid witness $\mathbb{w}_i$ s.t. $(\mathbb{x}_i, \mathbb{w}_i) \in \mathsf{R}_i$. This is possible because we have shown in Game* 6 that $\mathbb{x}_i^* \in \mathscr{L}_{\mathsf{R}_i}$ and hence $\mathbb{x}_i \in \mathscr{L}_{\mathsf{R}_i}$ for all $i \in \{0, 1, 2, \mathsf{dh}\}$ (cf. Definition 7).
In phase U_3 instead of computing the ZK commitment $A_i \leftarrow \Sigma_i.\mathsf{Init}(\mathsf{st}_i, \phi^i_{\mathbb{x}_i^*})$, the challenge $\gamma_i \leftarrow \mathcal{CH}$ and the response $\widetilde{\zeta}_i := \Sigma_i.\mathsf{Resp}(\mathsf{st}_i, \gamma_i, \mathbb{w}_i^*)$, we compute the ZK commitment $A_i \leftarrow \Sigma_i.\mathsf{Init}(\mathsf{st}_i, \phi^i_{\mathbb{x}_i})$, the challenge $\gamma_i \leftarrow \mathcal{CH}$ and the response $\zeta_i := \Sigma_i.\mathsf{Resp}(\mathsf{st}_i, \gamma_i, \mathbb{w}_i)$. This step is justified because Σ_i is (perfectly) translatable (Lemma 2). Let $\mathcal{R}$ be a reduction that simulates the blindness challenger and plays the blindness game with $\mathcal{A}$. In phase U_2 the reduction assembles $\mathbb{x}_i, \mathbb{w}_i, \mathbb{x}_i^*, \mathbb{w}_i^*, \rho_i^*$ and submits it to the translatability challenger to obtain either a translated transcript or a fresh transcript for $\mathbb{x}_i$. Thus,

$$\epsilon_8 = \epsilon_7 \ .$$

Game* 9 (Compute (A_i, γ_i, ζ_i) for $\mathbb{x}_i$ via SHVZK simulation): Instead of computing the ZK commitment $A_i := \Sigma_i.\mathsf{Init}(\mathsf{st}_i, \phi^i_{\mathbb{x}_i})$, the challenge $\gamma_i \leftarrow \mathcal{CH}$ and the response $\zeta_i := \Sigma_i.\mathsf{Resp}(\mathsf{st}_i, \gamma_i, \mathbb{w}_i)$ in phase U_3, we simulate the ZK proof using the SHVZK property of Σ_i (Definition 24) as $\gamma_i \leftarrow \mathcal{CH}$ and $(A_i, \zeta_i) \leftarrow \Sigma_i.\mathsf{Sim}(\mathsf{st}_i, \mathbb{x}_i)$. Thus,

$$\epsilon_9 = \epsilon_8 \ .$$

Game 10 (Undo aborts from Game 5 and 6): Note that the abort from Game 6 still never occurs. Removing the abort from Game 5 incurs the same loss as in Game 5. Thus,

$$|\epsilon_{10} - \epsilon_9| \leq \mathsf{AdvSnd}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}_{\mathsf{snd}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) \ .$$

Note that this game is efficient again.
Game 11 (Simulate π_M): Instead of generating the CRS $\mathsf{crs}_M := \mathsf{H}^M_{\mathsf{crs}}(0)$, we generate $(\mathsf{crs}_M, \mathsf{td}) \leftarrow \Pi_M.\mathsf{SimSetup}(1^\lambda)$. Now, instead of generating the proof $\pi_M \leftarrow \Pi_M.\mathsf{Prove}^{\mathsf{H}_M}(\mathsf{crs}_M, \mathbb{x}_M, \mathbb{w}_M)$ in phase U_1, we simulate the NIPS proof for the ciphertext ct_M as $\pi_M \leftarrow \mathsf{Sim}^{\mathsf{H}_M}(\mathsf{crs}_M, \mathsf{ct}_M)$. This step is justified by the zero-knowledge property of Π_M (Definition 26). Hence,

$$|\epsilon_{11} - \epsilon_{10}| \leq \mathsf{AdvZK}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}_{\mathsf{zk}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}})$$

for a straightforward reduction $\mathcal{R}_{\mathsf{zk}}$ that runs in approximately the same time as $\mathcal{A}$. Note that we no longer use the secret key of LHE nor the encryption randomness r_M.

Game 12 (Encrypt $\mathsf{ct}_M \leftarrow \mathsf{Enc}(\mathsf{pk}^{\mathsf{LHE}}, 0)$): Instead of computing $\mathsf{ct}_M \coloneqq \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{RE}}, M; r_M)$ in phase U_1, we compute $\mathsf{ct}_M \coloneqq \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{RE}}, 0; r_M)$. This step is justified by the IND-CPA security of LHE. Thus,

$$|\epsilon_{12} - \epsilon_{11}| \leq \mathsf{AdvINDCPA}^{\mathsf{LHE}}_{\mathcal{R}_{\mathsf{indcpa}}}(\lambda)$$

for a straightforward reduction $\mathcal{R}_{\mathsf{indcpa}}$ that runs in approximately the same time as $\mathcal{A}$.[18]

Game 13 (Switch $\mathsf{pp}^{\mathsf{X}}_i$ to binding mode): In this game we set up $\mathsf{pp}^{\mathsf{X}}_i \leftarrow \mathsf{COM_X.Setup}(1^\lambda, \mathsf{bind})$. Because of the parameter indistinguishability (Definition 41), we have that

$$|\epsilon_{13} - \epsilon_{12}| \leq 3 \cdot \mathsf{AdvParamIND}^{\mathsf{COM_X}}_{\mathcal{R}_{\mathsf{COM_X}}}(\lambda)$$

for straightforward reductions $\mathcal{R}_{\mathsf{COM_T}}, \mathcal{R}_{\mathsf{COM_X}}$ that run in approximately the same time as $\mathcal{A}$.

Game* 14 (Abort if $\mathbb{x}^*_{\mathsf{CT}} \notin \mathscr{L}_{\mathsf{R_{CT}}}$): We reintroduce the abort from Game 5, i.e., we abort if the adversary (in phase S_2) submits π^*_{CT} such that $\Pi_{\mathsf{CT}}.\mathsf{Ver}^{\mathsf{H_{CT}}}(\mathsf{crs}_{\mathsf{CT}}, \mathbb{x}^*_{\mathsf{CT}}, \pi^*_{\mathsf{CT}}) = 1$ yet $\mathbb{x}^*_{\mathsf{CT}} \notin \mathscr{L}_{\mathsf{R_{CT}}}$. Thus,

$$|\epsilon_{14} - \epsilon_{13}| \leq \mathsf{AdvSnd}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}'_{\mathsf{snd}}}(\lambda, \mathsf{Q}_{\mathsf{H_{CT}}})$$

for a straightforward reduction $\mathcal{R}'_{\mathsf{snd}}$ that runs in approximately the same time as $\mathcal{A}$.

Game* 15 (Compute CT_i as fresh encryption): Since the parameters of CMX_i are in binding mode, we can inefficiently extract the value $x_0 G$ from the commitment CMX_0. Notice that since Game 14 it holds that $\mathbb{x}^*_{\mathsf{CT}} \in \mathscr{L}_{\mathsf{R_{CT}}}$, hence the ciphertexts $\mathsf{CT}^*_0, \mathsf{CT}^*_1, \mathsf{CT}^*_2$ contains the plaintexts $x_0 G, x_0 G, 0$. Instead of computing the ciphertexts $\mathsf{CT}_i \coloneqq \mathsf{RE.Rerand}(\mathsf{pk}^{\mathsf{RE}}_i, \mathsf{CT}^*_i; \rho'_i)$ in phase U_3, we compute $\mathsf{CT}_0 \coloneqq \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}_0, x_0 G; \rho_0)$, $\mathsf{CT}_1 \coloneqq \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}_1, x_0 G; \rho_1)$ and $\mathsf{CT}_2 \coloneqq \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}_2, 0; \rho_2)$ where $\rho'_i, \rho_i \leftarrow \mathcal{R}_{\mathsf{RE}}$. This step is justified by the perfect rerandomization indistinguishability of the encryption scheme. Moreover, note that because if π^*_{CT} verifies, then the ciphertexts $\mathsf{CT}^*_0, \mathsf{CT}^*_1, \mathsf{CT}^*_2$ contain the plaintexts $x_0 G, x_0 G, 0$. Thus,

$$\epsilon_{15} = \epsilon_{14} \ .$$

Game* 16 (Sample fresh openings s_i and t_i for $\mathsf{COM_T}$): Recall that since Game 2 the parameters pp^{T} are set up in hiding mode. Instead of rerandomizing the commitments CMT_i and CMA_i in phase U_2, we sample fresh randomness $s_i, t_i \leftarrow \mathcal{R}_{\mathsf{COM_T}}$ and compute fresh commitments $\mathsf{CMT}_i \coloneqq \mathsf{COM_T.Commit}(\mathsf{pp}^{\mathsf{T}}, 0; s_i)$ and $\mathsf{CMA}_i \coloneqq \mathsf{COM_T.Commit}(\mathsf{pp}^{\mathsf{T}}, 0; t_i)$. This step is justified by the perfect rerandomization indistinguishability and perfect hiding of $\mathsf{COM_T}$. Thus,

$$\epsilon_{16} = \epsilon_{15} \ .$$

Notice that the behavior of the oracles $\mathsf{U}'_1, \mathsf{U}'_2, \mathsf{U}'_3$ in this final game is completely independent of the challenger's bit b. Thus, $\epsilon_{16} = 1/2$.

Consequently,

$$\begin{aligned}
\mathsf{AdvBlind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)/2 &= |\epsilon_0 - \epsilon_{16}| && (38)\\
&\leq \mathsf{AdvDDH}^{\mathbb{G}}_{\mathcal{R}_{\mathsf{DDH}}}(\lambda) + 3 \cdot \epsilon^{\mathsf{COM_X}}_{\mathsf{hide}}(\lambda) + \epsilon^{\mathsf{COM_T}}_{\mathsf{hide}}(\lambda) && (39)\\
&+ 2\mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}} \cdot \delta_{\mathsf{COM_T}}(\lambda) + \mathsf{AdvSnd}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}_{\mathsf{snd}}}(\lambda, \mathsf{Q}_{\mathsf{H_{CT}}}) && (40)\\
&+ \mathsf{AdvZK}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}_{\mathsf{zk}}}(\lambda, \mathsf{Q}_{\mathsf{H_{CT}}}) + \mathsf{AdvINDCPA}^{\mathsf{LHE}}_{\mathcal{R}_{\mathsf{indcpa}}}(\lambda) && (41)\\
&+ 3 \cdot \mathsf{AdvParamIND}^{\mathsf{COM_X}}_{\mathcal{R}_{\mathsf{COM_X}}}(\lambda) + \mathsf{AdvSnd}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}'_{\mathsf{snd}}}(\lambda, \mathsf{Q}_{\mathsf{H_{CT}}}) \ . && (42)
\end{aligned}$$

□

Remark 8 (Blindness without programming $\mathsf{H}^{\mathcal{CH}}$). The above proof requires programming the random oracle $\mathsf{H}^{\mathcal{CH}}$ to switch to SHVZK simulation of transcripts in the signature in Game* 9. To attain blindness without programming $\mathsf{H}^{\mathcal{CH}}$, we need a witness for $\mathsf{R_{dh}}$ and R_1. This can be achieved by remembering $\mathbb{w}_{\mathsf{dh}}$ in the reduction, and by augmenting the verification key vk with a zero-knowledge proof with relaxed knowledge soundness for relation

$$\mathsf{R_{vk}} \coloneqq \left\{ \begin{pmatrix} \mathbb{x}_{\mathsf{vk}} = \begin{pmatrix} \mathsf{pp}^{\mathsf{X}}_0, \mathsf{CMX}_0 \\ \mathsf{pp}^{\mathsf{X}}_1, \mathsf{CMX}_1 \\ \mathsf{pp}^{\mathsf{X}}_2, \mathsf{CMX}_2 \end{pmatrix} \\ \mathbb{w}_{\mathsf{vk}} = (x_0, \hat{s}_0, \hat{s}_1, \hat{s}_2) \end{pmatrix} \;\middle|\; \begin{array}{l} \mathsf{CMX}_0 = \mathsf{COM_X.Commit}(\mathsf{pp}^{\mathsf{X}}_0, x_0; \hat{s}_0) \\ \mathsf{CMX}_1 = \mathsf{COM_X.Commit}(\mathsf{pp}^{\mathsf{X}}_0, x_0; \hat{s}_1) \\ \mathsf{CMX}_2 = \mathsf{COM_X.Commit}(\mathsf{pp}^{\mathsf{X}}_0, 0; \hat{s}_2) \end{array} \right\}$$

[18] Here, we have a factor of two because we invoke IND-CPA security once for each message.

$$\tilde{\mathsf{R}}_{\mathsf{vk}} \coloneqq \{(\mathbb{x}_{\mathsf{vk}}, \widetilde{\mathbb{w}}_{\mathsf{vk}} = Z_0) \mid \exists \mathbb{w}_{\mathsf{vk}} = (x_0, \hat{s}_0, \hat{s}_1, \hat{s}_2) \colon Z_0 = x_0 G \wedge (\mathbb{x}_{\mathsf{vk}}, \mathbb{w}_{\mathsf{vk}}) \in \mathsf{R}_{\mathsf{COM_X}}\} \ .$$

Then the reduction is in possession of a valid witness for OR-relations $\mathsf{R}_0 \cup \mathsf{R}_{\mathsf{dh}}$ and $\mathsf{R}_1 \cup \mathsf{R}_2$. Hence, we are able to generate the transcripts by honestly using the OR-technique: Since the reduction can honestly run Σ_1 (resp. Σ_{dh}) for R_1 (resp. R_{dh}), and answer any challenge γ_1 and γ_{dh}, it can pick γ_0, γ_2 uniformly and use SHVZK simulation for $\mathsf{R}_0, \mathsf{R}_1$. Importantly, fresh encryptions of $x_0 G$ for R_1 are consistent with encryptions used by the (malicious) signer due to π^*_{CT}. By perfect indistinguishability of SHVZK simulation and honest transcript generation (given that a witness exists), making this change after Game* 9 is perfectly indistinguishable.

Importantly, after the above change, we can generate the challenges by querying $\mathsf{H}^{\mathcal{CH}}$ and do not need to program it anymore. Moreover, note that all steps[19] from Game* 5 to Game* 9 are statistical, including this additional change. Hence, while we programmed the random oracle during these proof steps, this is only used to establish the statistical indistinguishability between the first and last game. Since neither of them requires programming the random oracle (and it is also not required in the later proof steps), this indistinguishability still holds in the non-programmable ROM.

D.4 Proof of One-more Unforgeability (Theorem 2)

.

Theorem 2 (OMUF). *For any PPT adversary $\mathcal{A}$ there exist reductions with running time roughly that of $\mathcal{A}$, such that for sufficiently large λ*

$$\begin{aligned}
\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) \leq\ & \mathsf{AdvZK}^{\Pi_{\mathsf{CT}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) + \mathsf{AdvCRS}^{\Pi_M, \mathsf{ExtSetup}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) \\
& + \mathsf{AdvExt}^{\Pi_M, \mathsf{Ext}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) + \epsilon^{\mathsf{COM_T}}_{\mathsf{hide}}(\lambda) \\
& + 2(\mathsf{Q}_{\mathsf{H}_0^{\mathcal{CH}}} + 1)/p + 2\mathsf{AdvDDH}(\lambda) \\
& + \lceil \log Q_{\mathsf{S}} \rceil \begin{pmatrix} 4\mathsf{AdvINDCPA}^{\mathsf{RE}}(\lambda) \\ + 2(\mathsf{Q}_{\mathsf{H}_1^{\mathcal{CH}}} + 1)/p \\ + 3\mathsf{AdvParamIND}^{\mathsf{COM_T}}_{\mathcal{R}^3_{\mathsf{COM_T}}}(\lambda) \\ + 7\mathsf{AdvParamIND}^{\mathsf{COM_X}}_{\mathcal{R}^3_{\mathsf{COM_X}}}(\lambda) \\ + 2\ell/p \end{pmatrix}
\end{aligned}$$

where $\mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}, \mathsf{Q}_{\mathsf{H}_M}, \mathsf{Q}_{\mathsf{H}_0^{\mathcal{CH}}}, \mathsf{Q}_{\mathsf{H}_1^{\mathcal{CH}}}$ are bounds on the number of resp. oracle calls made by $\mathcal{A}$, Q_{S} is the number of signing sessions (started by $\mathcal{A}$), and $\epsilon^{\mathsf{COM_T}}_{\mathsf{hide}}(\lambda)$ is the (statistical) distance of $\mathsf{COM_T}$'s parameters from uniform.

Proof (Theorem 2). Let $\mathcal{A}$ be a PPT adversary on OMUF. We argue by game hops. As in the proof of blindness, inefficient games are marked by *. We denote by ϵ_i the adversary's success probability in Game i. An overview of the game hops is given in Tables 5 to 7. Before we proceed, let us establish some conventions.

- We assume without loss of generality that $\mathcal{A}$ finishes at least one session.[20]
- It can easily be checked that runtimes of reductions $\mathcal{R}$ in the proof are roughly that of $\mathcal{A}$. We omit further details on runtimes.
- We introduce some notation. For some bit $\beta \in \{0,1\}$, we denote by $\overline{\beta} \coloneqq 1 - \beta$ its negation. For some $j \in \mathbb{N}$, we denote by $j[k]$ the k-th bit of j in binary representation. Also, we denote by $j|_k$ the k-bit prefix of j in binary representation.

Game 0 (Original game): This is the OMUF game according to Definition 22. Let us recall the game explicitly and establish some notation. Note that hash functions are modeled as random oracles. Denote by Q_{H} the number of queries to some random oracle H made by $\mathcal{A}$ throughout the game and by Q_{S} the total number of signing sessions.

[19] If the NIPS is statistically sound, as it is for our instantiation in Supplementary Material B.

[20] If $\mathcal{A}$ successfully forges a signature σ for message M without finishing any session, then $\mathcal{A}$ can be transformed to an adversary $\mathcal{A}'$ that finishes one session for some message $M' \neq M$ with the same advantage. We stress that this assumption is purely for readability as it simplifies the argument in Game 7.

Recall that public parameters $(\mathsf{pp}_i^{\mathsf{X}})_{i\in\{0,1,2\}}$ for $\mathsf{COM}_{\mathsf{X}}$, public parameters pp^{T} for $\mathsf{COM}_{\mathsf{T}}$, common reference strings $\mathsf{crs}_{\mathsf{CT}}$ for Π_{CT}, common reference string crs_M for Π_M, public keys $\mathsf{pk}_i^{\mathsf{RE}}$ for RE, and tuple $(D_1, D_2, D_3) \in \mathbb{G}^3$ are setup by appropriate random oracles. First, the challenger samples $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$. That is, it samples $x_0 \leftarrow \mathbb{Z}_p$ and sets $x_1 = x_2 = 0$, and commits to x_i in $\mathsf{CMX}_i = \mathsf{COM}_{\mathsf{X}}.\mathsf{Commit}(\mathsf{pp}_i^{\mathsf{X}}, x_i, \hat{s}_i)$. It sends $\mathsf{vk} = (\mathsf{CMX}_0, \mathsf{CMX}_1, \mathsf{CMX}_2)$ to $\mathcal{A}$. Then, the challenger provides $\mathcal{A}$ access to signing oracles S_1 and S_2 (cf. Figure 3). We briefly recap how the challenger answers the oracles on the j-th signing session below.

- In $\mathsf{S}_1(\mathsf{smsg}_{1,j})$, the challenger parses $\mathsf{smsg}_{1,j} = (\mathsf{pk}_j^{\mathsf{LHE}}, \mathsf{ct}_{M,j}, \pi_{M,j})$, and verifies that $\pi_{M,j}$ is valid with respect to statement $\mathbb{x}_{M,j} = (\mathsf{pk}_j^{\mathsf{LHE}}, \mathsf{ct}_{M,j})$. If not, the challenger aborts, else it sets $z_{0,j} = x_0, z_{1,j} = x_0$, and $z_{2,j} = 0$. Then, the challenger encrypts $Z_{i,j} := z_{i,j}G$ in $\mathsf{CT}_{i,j}^* = \mathsf{RE}.\mathsf{Enc}(\mathsf{pk}_i^{\mathsf{RE}}, Z_{i,j}; \rho_{i,j}^*)$, and also commits to $\mathsf{CT}_{i,j}^*$ in $\mathsf{CMT}_{i,j}^* := \mathsf{COM}_{\mathsf{T}}.\mathsf{Commit}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}_{i,j}^*; s_{i,j}^*)$ Next, after sampling states $\mathsf{st}_{i,j} \leftarrow \Sigma_i.\mathsf{Setup}(1^\lambda)$, the challenger prepares the first message of the Σ-protocols. Then, the challenger encrypts $Z_{i,j} := z_{i,j}G$ in $\mathsf{CT}_{i,j}^* := \mathsf{RE}.\mathsf{Enc}(\mathsf{pk}_i^{\mathsf{RE}}, Z_{i,j}; \rho_{i,j}^*)$, and also commits to $\mathsf{CT}_{i,j}^*$ in $\mathsf{CMT}_{i,j}^* := \mathsf{COM}_{\mathsf{T}}.\mathsf{Commit}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}_{i,j}^*; s_{i,j}^*)$. Next, after sampling states $\mathsf{st}_{i,j} \leftarrow \Sigma_i.\mathsf{Setup}(1^\lambda)$, the challenger prepares the first message of the Σ-protocols. That is, it sets $\mathbb{x}_{1,j}^* := (\mathsf{pk}_0^{\mathsf{RE}}, \mathsf{pk}_1^{\mathsf{RE}}, \mathsf{CT}_{0,j}^*, \mathsf{CT}_{1,j}^*)$ and $\mathbb{x}_{2,j}^* := (\mathsf{pp}_2^{\mathsf{X}}, \mathsf{pk}_2^{\mathsf{RE}}, \mathsf{CMX}_2, \mathsf{CT}_{2,j}^*)$, and samples $A_{i,j}^* := \Sigma_i.\mathsf{Init}(\mathsf{st}_{i,j}, \phi^i_{\mathbb{x}_{i,j}^*})$ for $i \in \{1,2\}$. It also simulates $(A_{\mathsf{dh},j}^*, \zeta_{\mathsf{dh},j}^*) \leftarrow \Sigma_{\mathsf{dh}}.\mathsf{Sim}(\mathbb{x}_{\mathsf{dh}}, \gamma_{\mathsf{dh},j}^*)$ for $\mathbb{x}_{\mathsf{dh}} = (G, D_1, D_2, D_3)$ and $\gamma_{\mathsf{dh},j}^* \leftarrow \mathcal{CH}$. Then, the challenger computes $\mathsf{InitZero}$ homomorphically on $\mathsf{ct}_{M,j}$ to obtain $\mathsf{ct}_{0,j} := \mathsf{LHE}.\mathsf{Eval}(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{ct}_{M,j}, \mathsf{InitZero})$, which it rerandomizes to a ciphertext $\mathsf{ct}_{0,j}^* := \mathsf{LHE}.\mathsf{Rerand}(\mathsf{pk}_j^{\mathsf{LHE}}, \mathsf{ct}_{0,j}; \hat{r}_{0,j})$. Finally, the challenger outputs
$$\mathsf{smsg}_{2,j} := (\mathsf{ct}_{0,j}^*, (\mathsf{CMT}_{i,j}^*), A_1^*, A_2^*, A_{\mathsf{dh},j}^*).$$
- In $\mathsf{S}_2(\mathsf{smsg}_{3,j})$, the challenger parses $\mathsf{smsg}_{3,j} = (\delta_{0,j}^*, \delta_{1,j}^*)$. The challenger then sets $\gamma_{0,j}^* := \delta_{0,j}^* - \gamma_{\mathsf{dh},j}^*$ and $\gamma_{1,j}^* := \delta_{1,j}^* - \gamma_{2,j}^*$ for $\gamma_{2,j}^* \leftarrow \mathcal{CH}$. Then, it sets $\mathbb{w}_{0,j}^* := ((x_0, \hat{s}_1), (x_1, \hat{s}_1), (Z_{0,j}, \rho_{0,j}^*))$, $\mathbb{w}_{1,j}^* := ((Z_{0,j}, \rho_{0,j}^*), (Z_{1,j}, \rho_{1,j}^*))$, and $\mathbb{w}_{2,j}^* := ((x_2, \hat{s}_2), (Z_{2,j}, \rho_{2,j}^*))$. Next, computes the responses $\zeta_{i,j}^* = \Sigma_i.\mathsf{Resp}(\mathsf{st}_{i,j}, \gamma_{i,j}^*, \mathbb{w}_{i,j}^*)$. Then, the challenger proves that it set up the ciphertexts $\mathsf{CT}_{i,j}^*$ honestly via $\pi_{\mathsf{CT},j}^*$. That is, it sets $\mathbb{x}_{\mathsf{CT},j} := (\mathsf{pp}_0^{\mathsf{X}}, \mathsf{CMX}_0, (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CT}_{i,j}^*)_{i\in\{0,1,2\}})$ and $\mathbb{w}_{\mathsf{CT},j} := ((Z_{i,j}, \rho_{i,j}^*)_{i\in\{0,1,2\}}, x_0, \hat{s}_0)$, and computes the proof $\pi_{\mathsf{CT},j}^* \leftarrow \Pi_{\mathsf{CT}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{CT}}}(\mathsf{crs}_{\mathsf{CT}}, \mathbb{x}_{\mathsf{CT},j}, \mathbb{w}_{\mathsf{CT},j})$. It outputs
$$\mathsf{smsg}_{4,j} := ((\gamma_{i,j}^*, \zeta_{i,j}^*, \mathsf{CT}_{i,j}^*, s_{i,j}^*)_{i\in\{0,1,2\}}, \gamma_{\mathsf{dh},j}^*, \zeta_{\mathsf{dh},j}^*, \pi_{\mathsf{CT},j}^*).$$

Denote by ℓ the number of finished signing sessions. In the end, adversary $\mathcal{A}$ outputs $\ell + 1$ signature-message pairs $(M_\kappa^+, \sigma_\kappa^+)_{\kappa\in[\ell+1]}$. Throughout, we denote by $\mathcal{M}^+ := \{M_1^+, ..., M_{\ell+1}^+\}$ the set of "forged" messages, by $\mathcal{M}_{\mathsf{S}} := \{M_1, ..., M_{Q_{\mathsf{S}}}\} \subseteq \mathcal{M}$ the set of message for which a signing session has been started, and by $\mathcal{M}_{\mathsf{F}} \subseteq \mathcal{M}_{\mathsf{S}}$ the set of messages for which a signing session has been finished ($1 \le |\mathcal{M}_{\mathsf{F}}| \le \ell$). For convenience, we also define the set of indices of finished sessions as $\mathcal{J}_{\mathsf{F}} := \{j \mid M_j \in \mathcal{M}_{\mathsf{F}}\}$.

Game 1 (Sample $\mathsf{pk}_i^{\mathsf{RE}}$ with known $\mathsf{sk}_i^{\mathsf{RE}}$): In this game, we sample $(\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{sk}_i^{\mathsf{RE}}) \leftarrow \mathsf{RE}.\mathsf{KeyGen}(1^\lambda)$ and program $\mathsf{H}_{\mathsf{pk}}^{\mathsf{RE}}(i) := \mathsf{pk}_i^{\mathsf{RE}}$. Since honest RE public keys are distributed uniformly at random by assumption (cf. Definition 11), this change does not affect the success probability of $\mathcal{A}$.
$$\epsilon_1 = \epsilon_0 \ .$$

Game 2 (Simulate π_{CT}^*): In this game, we simulate the proofs $\pi_{\mathsf{CT},j}^*$ via the zero-knowledge simulator $\Pi_{\mathsf{CT}}.\mathsf{Sim}$ of Π_{CT} in S_2. That is, after the challenger defines $\mathbb{x}_{\mathsf{CT},j} = (\mathsf{pp}_0^{\mathsf{X}}, \mathsf{CMX}_0, (\mathsf{pk}_i^{\mathsf{RE}}, \mathsf{CT}_{i,j}^*)_{i\in\{0,1,2\}})$ and $\mathbb{w}_{\mathsf{CT},j} = ((Z_{i,j}, \rho_{i,j}^*)_{i\in\{0,1,2\}}, x_0, \hat{s}_0)$, instead of computing $\pi_{\mathsf{CT},j}^* \leftarrow \Pi_{\mathsf{CT}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{CT}}}(\mathsf{crs}_{\mathsf{CT}}, \mathbb{x}_{\mathsf{CT},j}, \mathbb{w}_{\mathsf{CT},j})$, it generates $(\mathsf{crs}_{\mathsf{CT}}, \mathsf{td}_{\mathsf{CT}}) \leftarrow \mathsf{SimSetup}(1^\lambda)$ and sets $\pi_{\mathsf{CT},j}^* \leftarrow \mathsf{Sim}(\mathsf{td}_{\mathsf{CT}}, \mathbb{x}_{\mathsf{CT},j})$.
Note that by construction, we have $(\mathbb{x}_{\mathsf{CT},j}, \mathbb{w}_{\mathsf{CT},j}) \in \mathsf{R}_{\mathsf{CT}}$ (cf. Eq. (16)). Also, the common reference string $\mathsf{crs}_{\mathsf{CT}}$ is chosen by random oracle $\mathsf{H}_{\mathsf{crs}}^{\mathsf{CT}}$. Thus,
$$|\epsilon_2 - \epsilon_1| \le \mathsf{AdvZK}_{\mathcal{R}_{\mathsf{zk}}}^{\Pi_{\mathsf{CT}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) \ .$$
for a straightforward reduction $\mathcal{R}_{\mathsf{zk}}$.

Game 3 (Embed td in crs_M): In this game, the challenger sets up crs_M in extraction mode. That is, denote by $(\mathsf{ExtSetup}, \mathsf{Ext})$ the knowledge extractor of Π_M. Before interacting with $\mathcal{A}$, the challenger sets $(\mathsf{crs}_M, \mathsf{td}_M) \leftarrow \mathsf{ExtSetup}(1^\lambda)$. The game programs $\mathsf{H}_{\mathsf{crs}}^M$ such that $\mathsf{H}_{\mathsf{crs}}^M(0) := \mathsf{crs}_M$ and then proceeds as before.

This step is justified by the CRS indistinguishability of Π_M (cf. Definition 27) such that

$$|\epsilon_3 - \epsilon_2| \leq \mathsf{AdvCRS}^{\Pi_M,\mathsf{ExtSetup}}_{\mathcal{R}_{\mathsf{extcrs}}}(\lambda, \mathsf{Q}_{\mathsf{H}_M})$$

for a straightforward reduction $\mathcal{R}_{\mathsf{extcrs}}$.

Game 4 (Extract M_j from $\pi_{M,j}$ and compute $\mathsf{ct}^*_{0,j}$ via M_j): In this game, the challenger extracts M_j from $\pi_{M,j}$ and computes $\mathsf{ct}^*_{0,j}$ as a fresh ciphertext of $A^*_{0,j}$, where $A^*_{0,j}$ is setup via $\Sigma_0.\mathsf{Init}$. In more detail, in the j-th signing session in S_1 the challenger sets $M_j \leftarrow \mathsf{Ext}((\mathsf{td}_M, \mathcal{Q}), (\mathbb{x}_{M,j}, \pi_j))$, where $\mathcal{Q}$ is a list containing all H_M queries and $\mathbb{x}_{M,j} := (\mathsf{pk}^{\mathsf{LHE}}_j, \mathsf{ct}_{M,j})$. Then, after the challenger samples $\mathsf{st}_{0,j} \leftarrow \Sigma_0.\mathsf{Setup}(1^\lambda)$, instead of computing $\mathsf{InitZero}$ homomorphically on $\mathsf{ct}_{M,j}$ and rerandomizing the resulting ciphertext, it sets

$$\begin{aligned}
\mathbb{x}^*_{0,j} &:= ((\mathsf{pp}^{\mathsf{X}}_0, \mathsf{CMX}_0), (\mathsf{pp}^{\mathsf{X}}_1, \mathsf{CMX}_1), (\mathsf{pk}^{\mathsf{RE}}_0, \mathsf{CT}^*_0), M_j),\\
A^*_{0,j} &\leftarrow \Sigma_0.\mathsf{Init}(\mathsf{st}_{0,j}, \phi^0_{\mathbb{x}^*_{0,j}}),\\
\mathsf{ct}^*_{M,j} &\leftarrow \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}_j, A^*_{0,j}).
\end{aligned}$$

Otherwise, the challenger proceeds as before.

In Lemma 6, we show via two intermediate hybrids that there is some adversary $\mathcal{R}_{\mathsf{ext}}$ on $\tilde{\mathsf{R}}_M$-extractability (cf. Eq. (15) and Definition 27) of Π_M such that

$$|\epsilon_4 - \epsilon_3| \leq \mathsf{AdvExt}^{\Pi_M,\mathsf{Ext}}_{\mathcal{R}_{\mathsf{ext}}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) \ .$$

Game 5 (Define M^+): In this game, after $\mathcal{A}$ outputs its forgeries, the challenger chooses index $\kappa^* := \min\{\kappa \mid M^+_\kappa \notin \mathcal{M}_{\mathsf{F}}\}$. Recall that M^+_κ are the $\ell + 1$ messages associated to $\mathcal{A}$'s forgeries, whereas there are at most ℓ "forged" messages in $\mathcal{M}_{\mathsf{F}}$. Thus, κ^* is always well-defined. Since this change is syntactical we have

$$\epsilon_5 = \epsilon_4 \ .$$

We refer to $M^+ := M^+_{\kappa^*}$ as the forgery's message and parse the corresponding signature $\sigma^+_{\kappa^*}$ as $((\mathsf{CT}^+_i, \pi^+_i, s^+_i, t^+_i)_{i\in\{0,1,2\}}, \pi^+_{\mathsf{dh}})$. Further, parse $\pi^+_i := (A^+_i, \gamma^+_i, \zeta^+_i)$ for $i \in \{0,1,2\}$ and $\pi^+_{\mathsf{dh}} := (A^+_{\mathsf{dh}}, \gamma^+_{\mathsf{dh}}, \zeta^+_{\mathsf{dh}})$. Also, set $\mathsf{CMT}^+_i := \mathsf{COM_T.Commit}(\mathsf{pp}^{\mathsf{T}}, \mathsf{CT}^+_i; s^+_i)$ for $i \in \{0,1,2\}$ and $\mathsf{CMA}^+_i := \mathsf{COM_T.Commit}(\mathsf{pp}^{\mathsf{T}}, A^+_i, t^+_i)$ for $i \in \{0,1,2\}$, $\mathsf{HIN}^+_0 := \big((\mathsf{pp}^{\mathsf{X}}_i, \mathsf{CMX}_i)_{i\in\{0,1\}}, (\mathsf{pk}^{\mathsf{RE}}_0, \mathsf{CMT}^+_0), M^+, \mathsf{CMA}_0, \mathbb{x}_{\mathsf{dh}}, A^+_{\mathsf{dh}}\big)$ and $\mathsf{HIN}^+_1 := \big((\mathsf{pp}^{\mathsf{X}}_2, \mathsf{CMX}_2), (\mathsf{pk}^{\mathsf{RE}}_i, \mathsf{CM}$ Recall that if $\sigma^+_{\kappa^*}$ is valid, then we have for $i \in \{0,1,2\}$ that

$$\begin{aligned}
\gamma^+_0 + \gamma^+_{\mathsf{dh}} &= \mathsf{H}^{\mathcal{CH}}_0(\mathsf{HIN}^+_0)\\
\gamma_1 + \gamma_2 &= \mathsf{H}^{\mathcal{CH}}_1(\mathsf{HIN}^+_1)\\
\Sigma_i.\mathsf{Verify}(\mathbb{x}^+_i, A^+_i, \gamma^+_i, \zeta^+_i) &= 1\\
\Sigma_{\mathsf{dh}}.\mathsf{Verify}(\mathbb{x}_{\mathsf{dh}}, A^+_{\mathsf{dh}}, \gamma^+_{\mathsf{dh}}, \zeta^+_{\mathsf{dh}}) &= 1
\end{aligned}$$

where $\mathbb{x}^+_0 := ((\mathsf{pp}^{\mathsf{X}}_0, \mathsf{CMX}_0), (\mathsf{pp}^{\mathsf{X}}_1, \mathsf{CMX}_1), (\mathsf{pk}^{\mathsf{RE}}_0, \mathsf{CT}^+_0), M^+), \mathbb{x}^+_1 := ((\mathsf{pk}^{\mathsf{RE}}_0, \mathsf{CT}^+_0), (\mathsf{pk}^{\mathsf{RE}}_1, \mathsf{CT}^+_1)), \mathbb{x}^+_2 := ((\mathsf{pp}^{\mathsf{X}}_2, \mathsf{CMX}_2), (\mathsf{pk}^{\mathsf{RE}}_2, \mathsf{CT}^+_2)), \mathbb{x}_{\mathsf{dh}} := (G, D_1, D_2, D_3)$. In the following, the game will sometimes know $\mathsf{sk}^{\mathsf{RE}}_i$, sometimes not. If $\mathsf{sk}^{\mathsf{RE}}_i$ is known to the game, we denote by $Z^+_i := \mathsf{RE.Dec}(\mathsf{sk}^{\mathsf{RE}}_i, \mathsf{CT}^+_i)$ the message encrypted in CT^+_i. Otherwise, Z^+_i remains undefined.

Game 6 (Setup $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} in binding mode): In this game, we setup the public parameters of $\mathsf{COM_X}$ and $\mathsf{COM_T}$ in binding mode. That is, before interacting with $\mathcal{A}$, the challenger sets $\mathsf{pp}^{\mathsf{X}}_i \leftarrow \mathsf{COM_X.Setup}(1^\lambda, \mathsf{bind})$ and $\mathsf{pp}^{\mathsf{T}} \leftarrow \mathsf{COM_T.Setup}(1^\lambda, \mathsf{bind})$. Then, it programs $\mathsf{H}^{\mathsf{X}}_{\mathsf{pp}}$ and $\mathsf{H}^{\mathsf{T}}_{\mathsf{pp}}$ such that $\mathsf{H}^{\mathsf{X}}_{\mathsf{pp}}(i) := \mathsf{pp}^{\mathsf{X}}_i$ and $\mathsf{H}^{\mathsf{T}}_{\mathsf{pp}}(0) := \mathsf{pp}^{\mathsf{T}}$.

As both $\mathsf{COM_T}$ and $\mathsf{COM_X}$ have uniform parameters in *hiding* mode (cf. Definition 40), we have

$$|\epsilon_6 - \epsilon_5| \leq \epsilon^{\mathsf{COM_T}}_{\mathsf{hide}}(\lambda) + \mathsf{AdvParamIND}^{\mathsf{COM_T}}_{\mathcal{R}^1_{\mathsf{COM_T}}}(\lambda) + 3 \cdot (\epsilon^{\mathsf{COM_X}}_{\mathsf{hide}}(\lambda) + \mathsf{AdvParamIND}^{\mathsf{COM_X}}_{\mathcal{R}^1_{\mathsf{COM_X}}}(\lambda))$$

for straightforward reductions $\mathcal{R}^1_{\mathsf{COM_T}}, \mathcal{R}^1_{\mathsf{COM_X}}$ that run in approximately the same time as $\mathcal{A}$.

Game 7 (Abort if $Z^+_0 \notin \{x_0 G\}_{j\in\mathcal{J}_{\mathsf{F}}}$): In this game, we abort if $Z^+_0 := \mathsf{RE.Dec}(\mathsf{sk}^{\mathsf{RE}}_i, \mathsf{CT}^+_0)$ (i.e., the message encrypted in CT^+_0) is not in the set $\{x_0 G\}_{j\in\mathcal{J}_{\mathsf{F}}}$.

Recall that we assume that at least one signing session is finished, i.e., $\mathcal{J}_{\mathsf{F}} \neq \emptyset$. By soundness of π^+_0 for relation R_0, we have that $Z^+_0 = x_0 G$ because $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} are setup in binding mode. In more detail,

if $\mathbb{x}_0^+ \in \mathscr{L}_{\mathsf{R}_0}$, there exists some witness $\mathbb{w}_{0,j}^+$ such that $(\mathbb{x}_0^+, \mathbb{w}_{0,j}^+) \in \mathsf{R}_0$. The values Z_0^+, x_0, x_1 (part of $\mathbb{w}_{0,j}^+$) are uniquely determined because CMX is set up in binding mode and RE is perfectly correct. By Eq. (7) we have that $x_0 G + x_1 M^+ - Z_0^+ = 0$. Since $x_1 = 0$ by construction, it follows that $x_0 G = Z_0^+$. It remains to show that indeed $\mathbb{x}_0^+ \in \mathscr{L}_{\mathsf{R}_0}$.
Assume for the sake of contradiction that $\mathbb{x}_0^+ \notin \mathscr{L}_{\mathsf{R}_0}$. Notice that since (D_1, D_2, D_3) are sampled at random, the probability that they form a DDH tuple is at most $1/p$, i.e., $\mathbb{x}_{\mathsf{dh}} \notin \mathscr{L}_{\mathsf{R}_{\mathsf{dh}}}$. Due to special soundness of Σ_0 and Σ_{dh}, there is at most a single choice of γ_0 and γ_1 such that there is a valid response ζ_0 and ζ_{dh}, that is

$$\Sigma_0.\mathsf{Verify}(\mathbb{x}_0^+, A_0^+, \gamma_0, \zeta_0) = 1$$
$$\Sigma_{\mathsf{dh}}.\mathsf{Verify}(\mathbb{x}_{\mathsf{dh}}, A_{\mathsf{dh}}^+, \gamma_{\mathsf{dh}}, \zeta_{\mathsf{dh}}) = 1$$

Since COM_T is perfectly binding, HIN_0^+ determines $\mathbb{x}_0^+, \mathbb{x}_{\mathsf{dh}}, A_0^+$ and A_{dh}^+. Thus, the pair (γ_0, γ_1) is already determined by HIN_0^+. Consequently, for a given $\mathsf{H}_0^{\mathcal{CH}}$ query, the probability that $\mathsf{H}_0^{\mathcal{CH}}(\mathsf{HIN}_0^+) = \gamma_0 + \gamma_1$ is at most $1/p$. A union bound over all possible $\mathsf{H}_0^{\mathcal{CH}}$ queries yields

$$|\epsilon_6 - \epsilon_7| \leq \frac{\mathsf{Q}_{\mathsf{H}_0^{\mathcal{CH}}} + 1}{p} \ .$$

Game 8 (Setup (D_1, D_2, D_3) as DDH tuple): In this game, we setup $\mathbb{x}_{\mathsf{dh}}$ as valid DDH tuple. That is, before interacting with $\mathcal{A}$, the challenger samples $D_1 \leftarrow \mathbb{G} \setminus \{0\}$, $d_2 \leftarrow \mathbb{Z}_p \setminus \{0\}$ and sets $D_2 \coloneqq d_2 \cdot G$ and $D_3 \coloneqq d_2 \cdot D_1$. Then, it embeds the values into $\mathsf{H}^{\mathsf{ddh}}$, i.e., sets $\mathsf{H}^{\mathsf{ddh}}(0) \coloneqq (D_1, D_2, D_3)$.
Thus

$$|\epsilon_7 - \epsilon_8| \leq \mathsf{AdvDDH}_{\mathcal{R}_{\mathsf{dh}}^1}(\lambda)$$

for a straightforward reduction $\mathcal{R}_{\mathsf{dh}}^1$ that runs in approximately the same time as $\mathcal{A}$.
Game 9 (Use DDH witness for Σ_{dh}): In this game, the challenger computes the Σ_{dh} transcript $(A_{\mathsf{dh},j}^*, \gamma_{\mathsf{dh},j}^*, \zeta_{\mathsf{dh},j}^*)$ via the witness $\mathbb{w}_{\mathsf{dh}} \coloneqq d_2$ (where d_2 is defined as in Game 8). That is, in S_1 the challenger samples $\mathsf{st}_{\mathsf{dh},j} \leftarrow \Sigma_{\mathsf{dh}}.\mathsf{Setup}(1^\lambda)$, and then sets $A_{\mathsf{dh},j}^* \coloneqq \Sigma_{\mathsf{dh}}.\mathsf{Init}(\mathsf{st}_{\mathsf{dh},j}, \phi_{\mathbb{x}_{\mathsf{dh}}}^{\mathsf{dh}})$, where $\mathbb{x}_{\mathsf{dh}} = (G, D_1, D_2, D_3)$. It also already samples $\gamma_{0,j}^* \leftarrow \mathbb{Z}_p$. In S_2 the challenger sets $\gamma_{\mathsf{dh},j}^* \coloneqq \delta_{0,\mathsf{dh}}^* - \gamma_{0,j}^*$ (instead of $\gamma_0^* \coloneqq \delta_{0,\mathsf{dh}}^* - \gamma_{\mathsf{dh},j}^*$ as in Game 8) and computes $\zeta_{\mathsf{dh},j} \coloneqq \Sigma_{\mathsf{dh}}.\mathsf{Resp}(\mathsf{st}_{\mathsf{dh},j}, \gamma_{\mathsf{dh},j}^*, \mathbb{w}_{\mathsf{dh}})$.
Clearly, the distribution of $\gamma_{\mathsf{dh},j}^*$ and $\gamma_{0,j}^*$ are identical in Game 8 and Game 9. Also, it follows from HVZK of Σ_{dh} that the Σ_{dh} transcripts $(A_{\mathsf{dh},j}^*, \gamma_{\mathsf{dh},j}^*, \zeta_{\mathsf{dh},j}^*)$ are distributed identically in Game 8 and Game 9. In conclusion, we have

$$\epsilon_9 = \epsilon_8 \ .$$

Game 10 (Simulate for Σ_0): In this game, the challenger simulates the Σ_0 transcript $(A_{0,j}^*, \gamma_{0,j}^*, \zeta_{0,j}^*)$ via the HVZK simulator $\Sigma_0.\mathsf{Sim}$. That is, in S_1 the challenger samples $\gamma_{0,j}^* \leftarrow \mathbb{Z}_p$ as in Game 9, and then sets $(A_{0,j}^*, \zeta_{0,j}^*) \leftarrow \Sigma_0.\mathsf{Sim}(\mathbb{x}_{0,j}^*, \gamma_{0,j}^*)$. In S_2 the challenger uses $\zeta_{0,j}^*$ sampled in S_1.
It follows from HVZK of Σ_0 that the transcripts $(A_{0,j}^*, \gamma_{0,j}^*, \zeta_{0,j}^*)$ are distributed identically in Game 9 and Game 10. Thus, we have

$$\epsilon_{10} = \epsilon_9 \ .$$

Game 11 (Setup pp_i^X and pp^T in hiding mode): In this game, we setup the public parameters of COM_X and COM_T in hiding mode. That is, before interacting with $\mathcal{A}$, the challenger sets $\mathsf{pp}_i^\mathsf{X} \leftarrow \mathsf{COM}_\mathsf{X}.\mathsf{Setup}(1^\lambda, \mathsf{hide})$ and $\mathsf{pp}^\mathsf{T} \leftarrow \mathsf{COM}_\mathsf{T}.\mathsf{Setup}(1^\lambda, \mathsf{hide})$. Then, it programs $\mathsf{H}_{\mathsf{pp}}^\mathsf{X}$ and $\mathsf{H}_{\mathsf{pp}}^\mathsf{T}$ such that $\mathsf{H}_{\mathsf{pp}}^\mathsf{X}(i) \coloneqq \mathsf{pp}_i^\mathsf{X}$ and $\mathsf{H}_{\mathsf{pp}}^\mathsf{T}(i) \coloneqq \mathsf{pp}^\mathsf{T}$.
This step is justified by the parameter indistinguishability (cf. Definition 41) such that

$$|\epsilon_{11} - \epsilon_{10}| \leq \mathsf{AdvParamIND}_{\mathcal{R}_{\mathsf{COM}_\mathsf{T}}^2}^{\mathsf{COM}_\mathsf{T}}(\lambda) + 3 \cdot \mathsf{AdvParamIND}_{\mathcal{R}_{\mathsf{COM}_\mathsf{X}}^2}^{\mathsf{COM}_\mathsf{X}}(\lambda)$$

for straightforward reductions $\mathcal{R}_{\mathsf{COM}_\mathsf{T}}^2, \mathcal{R}_{\mathsf{COM}_\mathsf{X}}^2$ that run in approximately the same time as $\mathcal{A}$.
Game 12 (Simulate for Σ_2): In this game, the challenger simulates the Σ_2 transcript $(A_{2,j}^*, \gamma_{2,j}^*, \zeta_{2,j}^*)$ via the HVZK simulator $\Sigma_2.\mathsf{Sim}$. That is, in S_1 the challenger samples $\gamma_{2,j}^* \leftarrow \mathbb{Z}_p$ (instead of doing so in S_2), and then sets $(A_{2,j}^*, \zeta_{2,j}^*) \leftarrow \Sigma_2.\mathsf{Sim}(\mathbb{x}_{2,j}^*, \gamma_{2,j}^*)$. In S_2 the challenger uses $\zeta_{2,j}^*$ sampled in S_1.
It follows from HVZK of Σ_2 that the Σ_2 transcripts $(A_{2,j}^*, \gamma_{2,j}^*, \zeta_{2,j}^*)$ are distributed identically in Game 11 and Game 12. Thus, we have

$$\epsilon_{12} = \epsilon_{11} \ .$$

Partitioning Before we start with the core argument of the proof, let us give a brief overview of the partitioning technique [39, 2]. The goal is to move to a game, where in the j-th signing session, the challenger encrypts $\mathbf{F}(j)$ in CT_0^* and CT_1^*, where $\mathbf{F}$ is a random function mapping into $\mathbb{Z}_p$, while keeping the guarantee that the adversary reuses some value $\mathbf{F}(j)$ for its forgery, i.e., $Z_0^+ \in \{\mathbf{F}(j)G\}_{j\in\mathcal{J}_\mathsf{F}}$. Note that only values from *finished* sessions are accepted in the forgery which is vital to later argue unforgeability. This random function is introduced iteratively in $\lceil\log(Q_\mathsf{S})\rceil$ conceptual steps in a tight manner.

At the beginning of the k-th step, some random function $\mathbf{RF}_k$ is evaluated only on the k-bit prefix $j|_k$ of j, and it holds that $Z_0^+ \in \{\mathbf{RF}_{k(j|_k)}G\}_{j\in\mathcal{J}_\mathsf{F}}$. Note that for $k = 0$, this is identical to encrypting a random value $\mathbf{RF}_0(\epsilon) \coloneqq x_0 \leftarrow \mathbb{Z}_p$ in CT_0^* and CT_1^* and checking that $Z_0^+ = x_0G$ as in Game 12. Then, the signing sessions are *partitioned* into two parts: Depending on a random bit-guess β, either a fresh random function $\mathbf{RF}'_k$ or the old random function $\mathbf{RF}_k$ is evaluated on the k-bit prefix $j|_k$ of j. Observe that the next function $\mathbf{RF}_{k+1}$ defined via

$$\mathbf{RF}_{k+1}(j|_{k+1}) \coloneqq \begin{cases} \mathbf{RF}_k(j|_k), & j[k+1] = \beta \\ \mathbf{RF}'_k(j|_k), & j[k+1] = \overline{\beta} \end{cases} \tag{43}$$

is again a random function, and after a logarithmic number of steps, the encrypted values are fully randomized. The forgery's check is adapted accordingly. We stress that as we only accept evaluations of $Z_0^+ \in \{\mathbf{RF}_{k(j|_k)}G\}_{j\in\mathcal{J}_\mathsf{F}}$, we need to make sure that the adversary learns no information about $\mathbf{RF}_k$ evaluations from unfinished sessions. For convenience, we define $k[0] \coloneqq k \bmod 2$.

Game 13.k.0 (Begin of partitioning loop): In this game, the challenger proceeds as in Game 12, except that it sets $z_0 \coloneqq z_1 \coloneqq \mathbf{RF}_k(j|_k)$ and checks that $Z_{k[0]}^+ \in \mathcal{Z}_k$, where $\mathbf{RF}_k$ denotes a random function mapping into $\mathbb{Z}_p$ and

$$\mathcal{Z}_k \coloneqq \{\mathbf{RF}_k(j|_k) \cdot G\}_{j\in\mathcal{J}_\mathsf{F}}. \tag{44}$$

For $k = 0$, this Game is identical to Game 12, and we have

$$\epsilon_{13.k.0} = \epsilon_{12} \ .$$

Game 13.k.1 (Sample β and set $x_2 = \overline{\beta}$): In this game, the challenger samples a random bit $\beta \leftarrow \{0,1\}$ and sets $x_2 \coloneqq \overline{\beta}$ (recall that the commitment CMX_2 to x_2 is part of the verification key). As pp_2^X is setup in hiding mode and x_2 is not used elsewhere,[21] Game 13.k.0 and Game 13.k.1 are identically distributed, and we have

$$\epsilon_{13.k.1} = \epsilon_{13.k.0} \ .$$

Game 13.k.2 (Set $z_{2,j} \coloneqq j[k+1]$): In this game, the challenger sets $z_{2,j} \coloneqq j[k+1]$ in the j-th signing session, i.e., $z_{2,j}$ is the $(k+1)$-th bit of $j \in [Q_\mathsf{S}]$.

Note that based on $z_{2,j}$, the challenger sets $\mathsf{CT}_{2,j}^* \coloneqq \mathsf{RE.Enc}(\mathsf{pk}_2^\mathsf{RE}, Z_{2,j}; \rho_{2,j}^*)$, where $Z_{2,j} \coloneqq z_{2,j}G$. As the proof $\pi_{\mathsf{CT},j}^*$ and the transcript $(A_{2,j}^*, \gamma_{2,j}^*, \zeta_{2,j}^*)$ are simulated, the values $(Z_{2,j}, \rho_{2,j}^*)$ are exclusively used to set up $\mathsf{CT}_{2,j}^*$. This step is justified by the IND-CPA security of RE such that

$$|\epsilon_{13.k.2} - \epsilon_{13.k.1}| \leq \mathsf{AdvINDCPA}^\mathsf{RE}_{\mathcal{R}^1_\mathsf{indcpa}}(\lambda)$$

for a straightforward reduction $\mathcal{R}^1_\mathsf{indcpa}$.

Game 13.k.3 (Abort if $Z_2^+ \neq \beta G$): In this game, after $\mathcal{A}$ outputs its forgeries $(M_\kappa^+, \sigma_\kappa^+)_{\kappa\in[\ell+1]}$, the challenger samples another bit $b \leftarrow \{0,1\}$ and aborts if (1) $Z_2^+ \in \{1, G\}$ and $Z_2^+ = (1-\beta)G$ or if (2) $Z_2^+ \notin \{1, G\}$ and $b = 0$.

Again, as pp_2^X is setup in hiding mode, the values β and b are independent of $\mathcal{A}$'s view. Also, note that both β and b are bits chosen uniformly at random. Thus, the abort probability is

$$\begin{aligned} \Pr[\mathsf{abort}] &= \Pr[Z_2^+ \in \{1, G\} \wedge Z_2^+ = (1-\beta)G] + \Pr[Z_2^+ \notin \{1, G\} \wedge b = 0] \\ &= \frac{1}{2} \cdot \Pr[Z_2^+ \in \{1, G\}] + \frac{1}{2} \cdot (1 - \Pr[Z_2^+ \in \{1, G\}]) = \frac{1}{2}. \end{aligned}$$

Thus, we have

$$\epsilon_{13.k.3} = \epsilon_{13.k.2}/2 \ .$$

[21] Recall that the transcripts $(A_{2,j}^*, \gamma_{2,j}^*, \zeta_{2,j}^*)$ are simulated.

Game 13.k.4 (Use witness for Σ_2 if $\beta \neq j[k+1]$): In this game, the challenger computes the transcript $(A^*_{2,j}, \gamma^*_{2,j}, \zeta^*_{2,j})$ via the witness $\mathbb{w}^*_{2,j} = ((x_2, \hat{s}_2), (Z_{2,j}, \rho^*_{2,j}))$ in the j-th session, if $\beta \neq j[k+1]$. That is, in S_1 the challenger first checks if $\beta \neq j[k+1]$. If not, it proceeds as in the previous game, else it sets $A^*_{2,j} := \Sigma_2.\mathsf{Init}(\mathsf{st}_{2,j}, \phi^2_{\mathbb{x}^*_{2,j}})$. In S_2 the challenger again checks if $\beta \neq j[k+1]$, and proceeds as in the previous game if the check fails. Else, it samples $\gamma^*_{2,j} := \delta^*_{1,2} - \gamma^*_{1,j}$, where $\gamma^*_{1,j} \leftarrow \mathbb{Z}_p$, and sets $\zeta^*_{2,j} := \Sigma_2.\mathsf{Resp}(\mathsf{st}_{2,j}, \gamma^*_{2,j}, \mathbb{w}^*_{2,j})$.
Observe that we have $x_2 G = Z_{2,j}$ since $x_2 = \overline{\beta} = j[k+1]$, and $Z_{2,j} = j[k+1]G$, thus $(\mathbb{x}^*_{2,j}, \mathbb{w}^*_{2,j}) \in \mathsf{R}_2$. Consequently, we have under HVZK of Σ_2 that

$$\epsilon_{13.k.4} = \epsilon_{13.k.3} \ .$$

Game 13.k.5 (Simulate for Σ_1 if $\beta \neq j[k+1]$): In this game, the challenger simulates the Σ_1 transcript $(A^*_{1,j}, \gamma^*_{1,j}, \zeta^*_{1,j})$ via the HVZK simulator $\Sigma_1.\mathsf{Sim}$ if $\beta \neq j[k+1]$. That is, in S_1 if $\beta \neq j[k+1]$, the challenger samples $\gamma^*_{1,j} \leftarrow \mathbb{Z}_p$ as before, and then sets $(A^*_{1,j}, \zeta^*_{1,j}) \leftarrow \Sigma_1.\mathsf{Sim}(\mathbb{x}^*_{1,j}, \gamma^*_{1,j})$. In S_2 the challenger uses $\zeta^*_{1,j}$ sampled in S_1 if $\beta \neq j[k+1]$.
It follows from HVZK of Σ_1 that the simulated Σ_1 transcripts $(A^*_{1,j}, \gamma^*_{1,j}, \zeta^*_{1,j})$ are distributed identically in Game 13.k.4 and Game 13.k.5. Thus, we have

$$\epsilon_{13.k.5} = \epsilon_{13.k.4} \ .$$

Game 13.k.6 (Set $z_{1-k[0],j} := \mathbf{RF}'_k[j|_k]$ if $\beta \neq j[k+1]$): Recall $k[0] := k \bmod 2$. In this game, if $\beta \neq j[k+1]$ in the j-th session, the challenger sets $z_{1-k[0],j} := \mathbf{RF}'_k[j|_k]$, where $\mathbf{RF}'_k$ is a fresh function mapping into $\mathbb{Z}_p$. Then, it sets $\mathsf{CT}^*_{1-k[0],j} := \mathsf{RE}.\mathsf{Enc}(\mathsf{pk}^{\mathsf{RE}}_{1-k[0]}, Z_{1-k[0],j}; \rho^*_{1-k[0],j})$, where $Z_{1-k[0],j} := z_{1-k[0],j} G$ as before.
Observe that as Σ_0 and Σ_1 are simulated, the values $z_{1-k[0],j}$ and $\rho^*_{1-k[0],j}$ are only used to initialize $\mathsf{CT}^*_{1-k[0],j}$. Also, the secret key $\mathsf{sk}_{1-k[0]}$ associated to $\mathsf{pk}^{\mathsf{RE}}_{1-k[0]}$ is not required for the simulation (as the forgery checks for M^+ is performed with $\mathsf{sk}_{k[0]}$ and the abort condition is evaluated with sk_2). This step is justified by the IND-CPA security of RE such

$$|\epsilon_{13.k.6} - \epsilon_{13.k.5}| \leq \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^2_{\mathsf{indcpa}}}(\lambda)$$

for a straightforward reduction $\mathcal{R}^2_{\mathsf{indcpa}}$.
Game 13.k.7 (Setup $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} in binding mode): In this game, the challenger sets up the public parameters of $\mathsf{COM}_{\mathsf{X}}$ and $\mathsf{COM}_{\mathsf{T}}$ in binding mode. That is, before interacting with $\mathcal{A}$, the challenger sets $\mathsf{pp}^{\mathsf{X}}_i \leftarrow \mathsf{COM}_{\mathsf{X}}.\mathsf{Setup}(1^\lambda, \mathsf{bind})$ and $\mathsf{pp}^{\mathsf{T}} \leftarrow \mathsf{COM}_{\mathsf{T}}.\mathsf{Setup}(1^\lambda, \mathsf{bind})$. Then, it programs $\mathsf{H}^{\mathsf{X}}_{\mathsf{pp}}$ and $\mathsf{H}^{\mathsf{T}}_{\mathsf{pp}}$ such that $\mathsf{H}^{\mathsf{X}}_{\mathsf{pp}}(i) := \mathsf{pp}^{\mathsf{X}}_i$ and $\mathsf{H}^{\mathsf{T}}_{\mathsf{pp}}(0) := \mathsf{pp}^{\mathsf{T}}$.
This step is justified by the parameter indistinguishability (cf. Definition 41) such that

$$|\epsilon_{13.k.7} - \epsilon_{13.k.6}| \leq \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}^3_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) + 3\mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^3_{\mathsf{COM}_{\mathsf{X}}}}(\lambda)$$

for straightforward reductions $\mathcal{R}^3_{\mathsf{COM}_{\mathsf{T}}}, \mathcal{R}^3_{\mathsf{COM}_{\mathsf{X}}}$ that run in approximately the same time as $\mathcal{A}$.
Game 13.k.8 (Abort if $Z^+_{1-k[0]} \notin \mathcal{Z}_k$): In this game, we change the forgery check that was introduced in Game 7. That is, instead of checking that $Z^+_{k[0]} \in \mathcal{Z}_k = \{\mathbf{RF}_k(j|_k) \cdot G\}_{j \in \mathcal{J}_{\mathsf{F}}}$, the challenger aborts if the value $Z^+_{1-k[0]}$ encrypted in the ciphertext $\mathsf{CT}^+_{1-k[0]}$ in $\mathcal{A}$'s forgery associated to M^+ does not lie in $\mathcal{Z}_k$.
Because $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} are setup in binding mode, this follows by soundness of π^+_1 for relation R_1. In more detail, if $\mathbb{x}^+_1 \in \mathscr{L}_{\mathsf{R}_1}$, there exists some witness $\mathbb{w}^+_{1,j}$ such that $(\mathbb{x}^+_1, \mathbb{w}^+_{1,j}) \in \mathsf{R}_1$. The values Z^+_0, Z^+_1 (part of $\mathbb{w}^+_{1,j}$) are uniquely determined because RE is perfectly correct. By Eq. (8) we have that $Z^+_0 = Z^+_1$. On the other hand, observe that $\mathbb{x}^+_2 \notin \mathscr{L}_{\mathsf{R}_2}$ as CMX_2 is setup in binding mode, so $x_2 = \overline{\beta}$ is uniquely determined, and due to the abort condition, it holds that $Z^+_2 = \beta G$ (i.e., $x_2 G - Z^+_2 \neq 0$). It remains to show that indeed $\mathbb{x}^+_1 \in \mathscr{L}_{\mathsf{R}_1}$ which follows by special soundness of Σ_0 and Σ_1. The formal argument is as in Game 7, and we omit details. We obtain

$$|\epsilon_{13.k.8} - \epsilon_{13.k.7}| \leq \frac{\mathsf{Q}_{\mathsf{H}^{\mathcal{C}\mathcal{H}}_1} + 1}{p} \ .$$

Game 13.k.9 (Setup $\mathsf{pp}_i^{\mathsf{X}}$ and pp^{T} in hiding mode): In this game, the challenger sets up the public parameters of $\mathsf{COM}_{\mathsf{X}}$ and $\mathsf{COM}_{\mathsf{T}}$ in hiding mode. That is, before interacting with $\mathcal{A}$, the challenger sets $\mathsf{pp}_i^{\mathsf{X}} \leftarrow \mathsf{COM}_{\mathsf{X}}.\mathsf{Setup}(1^\lambda, \mathsf{hide})$ and $\mathsf{pp}^{\mathsf{T}} \leftarrow \mathsf{COM}_{\mathsf{T}}.\mathsf{Setup}(1^\lambda, \mathsf{hide})$. Then, it programs $\mathsf{H}_{\mathsf{pp}}^{\mathsf{X}}$ and $\mathsf{H}_{\mathsf{pp}}^{\mathsf{T}}$ such that $\mathsf{H}_{\mathsf{pp}}^{\mathsf{X}}(i) \coloneqq \mathsf{pp}_i^{\mathsf{X}}$ and $\mathsf{H}_{\mathsf{pp}}^{\mathsf{T}}(0) \coloneqq \mathsf{pp}^{\mathsf{T}}$.
This step is justified by the parameter indistinguishability (cf. Definition 41) such that

$$|\epsilon_{13.k.9} - \epsilon_{13.k.8}| \leq \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}^4_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) + 3\mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^4_{\mathsf{COM}_{\mathsf{X}}}}(\lambda)$$

for straightforward reductions $\mathcal{R}^4_{\mathsf{COM}_{\mathsf{T}}}, \mathcal{R}^4_{\mathsf{COM}_{\mathsf{X}}}$ that run in approximately the same time as $\mathcal{A}$.
Game 13.k.10 (Set $z_{k[0],j} = \mathbf{RF}'_k[j|_k]$ if $\beta \neq j[k+1]$): Let $k[0] \coloneqq k \bmod 2$. In this game, if $\beta \neq j[k+1]$ in the j-th session, the challenger sets $z_{k[0],j} \coloneqq \mathbf{RF}'_k[j|_k]$, where $\mathbf{RF}'_k$ is the random function introduced in Game 13.k.6. Then, it sets $\mathsf{CT}^*_{k[0],j} \coloneqq \mathsf{RE}.\mathsf{Enc}(\mathsf{pk}^{\mathsf{RE}}_{k[0]}, Z^*_{k[0]}; \rho^*_{k[0],j})$, where $Z^*_{k[0]} \coloneqq z_{k[0]}G$ as before.
Observe that as Σ_0 and Σ_1 are simulated, the values $z_{k[0],j}$ and $\rho_{k[0],j}$ are only used to initialize $\mathsf{CT}^*_{k[0],j}$. Also, the secret key $\mathsf{sk}_{k[0]}$ associated to $\mathsf{pk}^{\mathsf{RE}}_{k[0]}$ is not required for the simulation (as the forgery checks for M^+ is performed with $\mathsf{sk}_{1-k[0]}$ due to the previous hybrid, and the abort condition is evaluated with sk_2). This step is justified by the IND-CPA security of RE such

$$|\epsilon_{13.k.10} - \epsilon_{13.k.9}| \leq \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^3_{\mathsf{indcpa}}}(\lambda)\ .$$

for a straightforward reduction $\mathcal{R}^3_{\mathsf{indcpa}}$.
In the next two games, we revert the changes made in Game 13.k.4 and Game 13.k.5 with respect to the computation of proofs via Σ_1 and Σ_2.
Game 13.k.11 (Use witness for Σ_1 if $\beta \neq j[k+1]$): In this game, the challenger computes the transcript $(A^*_{1,j}, \gamma^*_{1,j}, \zeta^*_{1,j})$ via the witness $\mathbb{w}^*_{1,j} = ((Z_{0,j}, \rho^*_{0,j}), (Z_{1,j}, \rho^*_{1,j}))$ in the j-th session, if $\beta \neq j[k+1]$.
Observe that we have $Z_{0,j} = Z_{1,j} = \mathbf{RF}'_k[j|_k]$ if $\beta \neq j[k+1]$, thus $(\mathbb{x}^*_{1,j}, \mathbb{w}^*_{1,j}) \in \mathsf{R}_1$. Consequently, we have under HVZK of Σ_1 that

$$\epsilon_{13.k.11} = \epsilon_{13.k.10}\ .$$

Game 13.k.12 (Simulate for Σ_2 if $\beta \neq j[k+1]$): In this game, the challenger simulates the Σ_2 transcript $(A^*_{2,j}, \gamma^*_{2,j}, \zeta^*_{2,j})$ via the HVZK simulator $\Sigma_2.\mathsf{Sim}$ if $\beta \neq j[k+1]$. That is, in S_1 if $\beta \neq j[k+1]$, the challenger samples $\gamma^*_{2,j} \leftarrow \mathbb{Z}_p$, and then sets $(A^*_{2,j}, \zeta^*_{2,j}) \leftarrow \Sigma_2.\mathsf{Sim}(\mathbb{x}^*_{2,j}, \gamma^*_{2,j})$. In S_2 the challenger uses $\zeta^*_{2,j}$ sampled in S_1 if $\beta \neq j[k+1]$.
It follows from HVZK of Σ_2 that the simulated Σ_2 transcripts $(A^*_{2,j}, \gamma^*_{2,j}, \zeta^*_{2,j})$ are distributed identically in Game 13.k.4 and Game 13.k.5. Thus, we have

$$\epsilon_{13.k.12} = \epsilon_{13.k.11}\ .$$

Game 13.k.13 (Rewrite $z_{0,j}$ and $z_{1,j}$ in terms of $\mathbf{RF}_{k+1}$): This is a purely conceptual game. In particular, observe that the challenger sets $z_{0,j} \coloneqq z_{1,j} \coloneqq \mathbf{RF}_{k+1}(j|_{k+1})$ in Game 13.k.12, where $\mathbf{RF}_{k+1}$ is defined as in Eq. (43). We have

$$\epsilon_{13.k.13} = \epsilon_{13.k.12}\ .$$

Game 13.k.14 (Abort if $Z^+_{1-k[0]} \notin \mathcal{Z}_{k+1}$): In this game, we change the forgery check again. That is, instead of checking that $Z^+_{1-k[0]} \in \mathcal{Z}_k$, the challenger checks that $Z^+_{1-k[0]} \in \mathcal{Z}_{k+1}$, where $\mathcal{Z}_k$ and $\mathcal{Z}_{k+1}$ are defined by Eq. (44) with respect to random functions $\mathbf{RF}_k$ and $\mathbf{RF}_{k+1}$, respectively. That is, recall that

$$\begin{aligned}
\mathcal{Z}_k &= \{\mathbf{RF}_k(j|_k) \cdot G\}_{j \in \mathcal{J}_{\mathsf{F}}} \\
&= \underbrace{\{\mathbf{RF}_k(j|_k) \cdot G \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \beta\}}_{\coloneqq \mathcal{S}_{\mathsf{k.both}}} \cup \underbrace{\{\mathbf{RF}_k(j|_k) \cdot G \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \overline{\beta}\}}_{\coloneqq \mathcal{S}_{k.13}}, \\
\mathcal{Z}_{k+1} &= \{\mathbf{RF}_{k+1}(j|_{k+1}) \cdot G\}_{j \in \mathcal{J}_{\mathsf{F}}} \\
&= \underbrace{\{\mathbf{RF}_k(j|_k) \cdot G \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \beta\}}_{= \mathcal{S}_{\mathsf{k.both}}} \cup \underbrace{\{\mathbf{RF}'_k(j|_k) \cdot G \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \overline{\beta}\}}_{\coloneqq \mathcal{S}_{k.14}},
\end{aligned}$$

where the last equality follows by definition of $\mathbf{RF}_{k+1}$ (cf. Eq. (43)).

Observe that in Game 13.k.13, the challenger also accepts the values $Z_{1-k[0]} \in \mathcal{S}_{k.13}$, whereas not all such values are accepted in Game 13.k.14. Below, we show that this does not considerably decrease the advantage of $\mathcal{A}$ in Game 13.k.14. Roughly, this is because the values that are not accepted in $\mathcal{S}_{k.13}$ are statistically hidden, and thus hard to predict. This argument is quite subtle in our case because we only index over finished session $j \in \mathcal{J}_{\mathsf{F}}$, and we need to make sure that unfinished sessions do not leak information about non-accepted values in $\mathcal{S}_{k.13}$. On the other hand, in Game 13.k.14, the challenger accepts forgeries $Z_{1-k[0]} \in \mathcal{S}_{k.14}$, but this at most improves the advantage of $\mathcal{A}$ in Game 13.k.14.
Before we proceed with the proof, let us be more precise in our identity for $\mathcal{Z}_k$. In particular, observe that the values $\mathbf{RF}_k(j|_k) \in \mathcal{S}_{k.13}$ such that $j|_k \in \mathcal{J}^{\beta}_{\mathsf{pre}} \coloneqq \{j|_k \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \beta\}$ are also included in $\mathcal{S}_{\mathsf{k.both}}$, and thus also accepted in Game 13.k.14. In conclusion, we can write

$$\mathcal{Z}_k = \mathcal{S}_{\mathsf{k.both}} \cup \underbrace{\{\mathbf{RF}_k(j|_k) \cdot G \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \overline{\beta}, j|_k \notin \mathcal{J}^{\beta}_{\mathsf{pre}}\}}_{\coloneqq \mathcal{S}^*_{k.13}},$$

By the above discussion, it suffices to show that $\Pr[Z^+_{1-k[0]} \in \mathcal{S}^*_{k.13}]$ is sufficiently small in Game 13.k.13. Let $Z \in \mathcal{S}^*_{k.13}$ be arbitrary. That is, there is $j_Z \in \mathcal{J}_{\mathsf{F}}$ such that $Z = \mathbf{RF}_k(j_Z|_k) \cdot G$ with $j_Z[k+1] = \overline{\beta}$ and $j_Z|_k \notin \mathcal{J}^{\beta}_{\mathsf{pre}}$. We show that the value Z is information-theoretically hidden from $\mathcal{A}$ in Game 13.k.13. For this, let us recap the sources from which $\mathcal{A}$ obtains information about evaluations of $\mathbf{RF}_k$, including Z. Observe that these are the ciphertexts $\mathsf{CT}^*_{0,j}$ and $\mathsf{CT}^*_{1,j}$ for $j \in [Q_{\mathsf{S}}]$, and values that are computed based on $\mathsf{CT}^*_{0,j}$ and $\mathsf{CT}^*_{1,j}$. In particular, for $i \in \{0,1\}$:

i. Ciphertext $\mathsf{CT}^*_{i,j}$ encrypts $\mathbf{RF}_k(j|_k)$.
ii. Commitment $\mathsf{CMT}_{i,j}$ commits to $\mathsf{CT}^*_{i,j}$.
iii. Σ-protocol commitment $A^*_{i,j} = \Sigma_i.\mathsf{Init}(\mathsf{st}_{i,j}, \phi^i_{\mathbb{x}^*_{i,j}})$, where $\mathbb{x}^*_{i,j}$ includes the ciphertexts.
iv. Σ-protocol responses $\zeta^*_{i,j} = \Sigma_i.\mathsf{Resp}(\mathsf{st}_{i,j}, \gamma^*_{i,j}, \mathbb{w}^*_{i,j})$, where $\mathbb{w}^*_{i,j}$ contains the message and randomness of the ciphertexts.
v. Ciphertext $\mathsf{ct}^*_{0,j}$, which is an encryption of $A^*_{0,j}$.

First, observe that if we have in the j-th session that $j[k+1] = \overline{\beta}$, then the evaluations of the random function $\mathbf{RF}'_k$ are encrypted. Since $\mathbf{RF}'_k$ and $\mathbf{RF}_k$ are independent, these evaluations leak no information about $Z = \mathbf{RF}_k(j_Z|_k) \cdot G$. Let us inspect the case $j[k+1] = \beta$. If the session is finished (i.e., $j \in \mathcal{J}_{\mathsf{F}}$) or $j|_k \in \mathcal{J}^{\beta}_{\mathsf{pre}}$, then values in $\mathcal{S}_{\mathsf{k.both}}$ are encrypted which are distributed independently of Z by definition of $\mathcal{S}^*_{k.13}$.
On the other hand, if $j \notin \mathcal{J}_{\mathsf{F}}$ and $j[k+1] = \beta$, then $\mathsf{CT}^*_{0,j}$ and $\mathsf{CT}^*_{1,j}$ are not necessarily independently distributed from Z. Let us inspect this case in more detail, for $i \in \{0,1\}$:

i. Ciphertext $\mathsf{CT}^*_{i,j}$ is only sent to $\mathcal{A}$ in S_2, and because $j \notin \mathcal{J}_{\mathsf{F}}$, $\mathcal{A}$ does not learn $\mathsf{CT}^*_{i,j}$ in plain.
ii. While $\mathsf{CMT}^*_{i,j}$ is a commitment to $\mathsf{CT}^*_{i,j}$, the public parameters pp^{T} are setup in hiding mode. Thus, the commitments are distributed independently of Z.
iii. As $A^*_{i,j} = \Sigma_i.\mathsf{Init}(\mathsf{st}_{i,j}, \phi^i_{\mathbb{x}^*_{i,j}})$, the only value that might leak information about Z is the description of $\phi^i_{\mathbb{x}^*_{i,j}}$. In particular, by definition of $\phi^i_{\mathbb{x}^*_{i,j}}$ (cf. Section 3.1). The values that depend on $\mathsf{CT}^*_{i,j}$ are the maps $\phi^{\mathsf{RE}}_{\mathbb{x}^*_{\mathsf{RE},i,j}}$, where $\mathbb{x}^*_{\mathsf{RE},i,j} = (\mathsf{pk}^{\mathsf{RE}}_i, \mathsf{CT}^*_{i,j})$. By Remark 1, the map $\phi^{\mathsf{RE}}_{\mathbb{x}^*_{\mathsf{RE},i,j}}$ only depends on $\mathsf{pk}^{\mathsf{RE}}_i$. Thus, the distribution of $A^*_{i,j}$ is independent of Z.
iv. The Σ-protocol responses $\zeta^*_{i,j}$ are only sent in the S_2, but the j-th signing session is not finished ($j \notin \mathcal{J}_{\mathsf{F}}$).
v. As the distribution of ciphertext $A^*_{0,j}$ is independent of Z, so is the distribution of $\mathsf{ct}^*_{0,j}$.

In conclusion, the view of $\mathcal{A}$ is independent of the distribution of $Z \in \mathcal{S}^*_{k.13}$, and we have $\Pr[Z^+_{1-k[0]} = Z] = 1/p$. Finally, since there are at most $\ell = |\mathcal{J}_{\mathsf{F}}|$ values in $\mathcal{S}^*_{k.13}$, a union bound yields

$$|\epsilon_{13.k.14} - \epsilon_{13.k.13}| \leq \frac{\ell}{p} \ .$$

Game 13.k.15 (Do not abort if $Z^+_2 \neq \beta G$): Recall that in Game 13.k.14, after $\mathcal{A}$ outputs its forgeries $(M^+_\kappa, \sigma^+_\kappa)_{\kappa \in [\ell+1]}$, the challenger samples a bit $b \leftarrow \{0,1\}$ and aborts if (1) $Z^+_2 \in \{1, G\}$ and $Z^+_2 = (1-\beta)G$ or if (2) $Z^+_2 \notin \{1, G\}$ and $b = 0$. In this game, abort conditions (1) and (2) are removed. As $\mathsf{pp}^{\mathsf{X}}_2$ is setup in hiding mode, the bits β and b are independent of $\mathcal{A}$'s view. Importantly, the advantage of $\mathcal{A}$ is independent of the bits β and b. As in Game 13.k.3, we can show that the abort probability in Game 13.k.14 is 1/2. Thus, by removing the abort condition, the advantage of $\mathcal{A}$ doubles. That is, we have

$$\epsilon_{13.k.15} = 2\epsilon_{13.k.14} \ .$$

Game 13.k.16 (Set $z_{2,j} \coloneqq 0$): In this game, the challenger sets $z_{2,j} = 0$ in the j-th signing session, i.e., we revert the change made in Game 13.k.2.
We can argue that Game 13.k.15 and 13.k.16 are indistinguishable under IND-CPA security of RE as in Game 13.k.2. That is, observe that based on $z_{2,j}$, the challenger sets $\mathsf{CT}^*_{2,j} \coloneqq \mathsf{RE.Enc}(\mathsf{pk}_2^{\mathsf{RE}}, Z_{2,j}; \rho^*_{2,j})$, where $Z_{2,j} \coloneqq z_{2,j}G$. As the proof $\pi^*_{\mathsf{CT},j}$ and the transcript $(A^*_{2,j}, \gamma^*_{2,j}, \zeta^*_{2,j})$ are simulated, the values $(Z_{2,j}, \rho^*_{2,j})$ are exclusively used to set up $\mathsf{CT}^*_{2,j}$. This step is justified by the IND-CPA security of RE such that

$$|\epsilon_{13.k.16} - \epsilon_{13.k.15}| \leq \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^4_{\mathsf{indcpa}}}(\lambda)$$

for a straightforward reduction $\mathcal{R}^4_{\mathsf{indcpa}}$.
Game 13.k.17 (Forget β and set $x_2 \coloneqq 0$): In this game, the challenger does not sample the bit β anymore. Also, it sets $x_2 \coloneqq 0$. As $\mathsf{pp}_2^{\mathsf{X}}$ is setup in hiding mode and the values x_2, β are not used elsewhere, Game 13.k.16 and Game 13.k.17 are identically distributed, and we have

$$\epsilon_{13.k.17} = \epsilon_{13.k.16} \ .$$

Moreover, observe that the last game of one iteration is equal to the first game of the next iteration, i.e.,

$$\epsilon_{13.(k+1).0} = \epsilon_{13.k.17}$$

because both games perform the same check $Z^+_{1-k[0]} \in \mathcal{Z}_{k+1} \iff Z^+_{(k+1)[0]} \in \mathcal{Z}_{k+1}$ since $(k+1)[0] = 1 - k[0]$.
Game 14 (End of partitioning loop): This game is identical to Game 13.N.17 for $N \coloneqq \lceil \log Q_{\mathsf{S}} \rceil$, i.e.,

$$\epsilon_{14} = \epsilon_{13.k.17} \ .$$

Without loss of generality, let us assume that N is even, so the forgery check in Game 14 is performed with Z^+_0 and not Z^+_1. As we have $j|_N = j$, the random function $\mathbf{RF}_N$ is evaluated on its entire input and we simply set $\mathbf{F} \coloneqq \mathbf{RF}_N$ and write $z_0 = z_1 = \mathbf{F}(j)$. Also, note that the challenger now accepts the forgeries if $\mathbb{Z}^+_0 \in \{\mathbf{F}(j) \cdot G\}_{j \in \mathcal{J}_{\mathsf{F}}}$. Over a single iteration of the loop, we find that

$$|\epsilon_{13.k.17} - \epsilon_{13.k.0}| \leq \epsilon_{\mathsf{loop}}(\lambda) \ .$$

We delay ϵ_{loop} until Eq. (46). Over all iterations of the loop, we find that

$$|\epsilon_{14} - \epsilon_{13}| \leq \lceil \log Q_{\mathsf{S}} \rceil \cdot \epsilon_{\mathsf{loop}}(\lambda) \ .$$

Game 15 (Set $Z_{0,j} \coloneqq Z_{1,j} \coloneqq \mathbf{F}(j)G + M_j$ and $x_1 \coloneqq 1$): In this game, the challenger sets $Z_{0,j} \coloneqq Z_{1,j} \coloneqq \mathbf{F}(j)G + M_j$ and $x_1 \coloneqq 1$. The forgery check is also adapted accordingly, i.e., the challenger checks that $Z^+_0 \in \{\mathbf{F}(j)G + M_j\}_{j \in \mathcal{J}_{\mathsf{F}}}$.
Note that the challenger can *not* efficiently compute the DLOGs $z_{0,j}$ and $z_{1,j}$ anymore as this requires the DLOG of M_j. But as these DLOGs are not required for simulation,[22] the simulation of Game 15 is efficient.
As $\mathsf{pp}_1^{\mathsf{X}}$ is setup in hiding mode and as $\mathbf{F}$ is a random function, the distribution of $Z_{0,j}$ and $Z_{1,j}$ are identical in Game 14 and Game 15. Thus, we have

$$\epsilon_{15} = \epsilon_{14} \ .$$

Game 16 (Replace $\mathbf{F}(j)$ with x_0 via partitioning): In this game, the challenger sets $Z_{0,j} \coloneqq Z_{1,j} \coloneqq x_0 G + M_j$ and adapts the forgery check accordingly, i.e., checks that $Z^+_0 \in \{x_0 G + M_j\}_{j \in \mathcal{J}_{\mathsf{F}}}$. By reversing the transitions from Game 13.k.0 to Game 13.k.17 we can show (Lemma 7) that

$$|\epsilon_{16} - \epsilon_{15}| \leq \lceil \log Q_{\mathsf{S}} \rceil \cdot \epsilon_{\mathsf{loop}}(\lambda) \ .$$

Game 17 (Setup $\mathsf{pp}_i^{\mathsf{X}}$ and pp^{T} in binding mode): In this game, the challenger sets up the public parameters of $\mathsf{COM}_{\mathsf{X}}$ and $\mathsf{COM}_{\mathsf{T}}$ in binding mode. That is, before interacting with $\mathcal{A}$, the challenger sets $\mathsf{pp}_i^{\mathsf{X}} \leftarrow \mathsf{COM}_{\mathsf{X}}.\mathsf{Setup}(1^\lambda, \mathsf{bind})$ and $\mathsf{pp}^{\mathsf{T}} \leftarrow \mathsf{COM}_{\mathsf{T}}.\mathsf{Setup}(1^\lambda, \mathsf{bind})$. Then, it programs $\mathsf{H}^{\mathsf{X}}_{\mathsf{pp}}$ and $\mathsf{H}^{\mathsf{T}}_{\mathsf{pp}}$ such that $\mathsf{H}^{\mathsf{X}}_{\mathsf{pp}}(i) \coloneqq \mathsf{pp}_i^{\mathsf{X}}$ and $\mathsf{H}^{\mathsf{T}}_{\mathsf{pp}}(0) \coloneqq \mathsf{pp}^{\mathsf{T}}$.
As before, this step is justified by the parameter indistinguishability such that

$$|\epsilon_{17} - \epsilon_{16}| \leq \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) + 3\mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^3_{\mathsf{COM}_{\mathsf{X}}}}(\lambda)$$

for straightforward reductions $\mathcal{R}^3_{\mathsf{COM}_{\mathsf{T}}}, \mathcal{R}^3_{\mathsf{COM}_{\mathsf{X}}}$ that run in approximately the same time as $\mathcal{A}$.

[22] Recall that the witness for the Σ-protocols are the group elements $Z_{i,j}$ and not their DLOGs.

Game 18 (Use witness for Σ_0): In this game, the challenger computes the Σ_0-transcript $(A^*_{0,j}, \gamma^*_{0,j}, \zeta^*_{0,j})$ via the witness $\mathbb{w}^*_{0,j} = ((x_0, \hat{s}_0), (x_1, \hat{s}_1), Z_{0,j}, \rho^*_{0,j})$. That is, in S_1 the challenger samples $\mathsf{st}_{0,j} \leftarrow \Sigma_0.\mathsf{Setup}(1^\lambda)$, and then sets $A^*_{0,j} := \Sigma_0.\mathsf{Init}(\mathsf{st}_{0,j}, \phi^0_{\mathbb{x}^*_{0,j}})$, where $\mathbb{x}^*_{0,j} = ((\mathsf{pp}^{\mathsf{X}}_0, \mathsf{CMX}_0), (\mathsf{pp}^{\mathsf{X}}_1, \mathsf{CMX}_1), (\mathsf{pk}^{\mathsf{RE}}_0, \mathsf{CT}^*_{0,j}), M_j)$. It also already samples $\gamma^*_{\mathsf{dh},j} \leftarrow \mathbb{Z}_p$. In S_2 the challenger sets $\gamma^*_{0,j} := \delta^*_{0,\mathsf{dh}} - \gamma^*_{\mathsf{dh},j}$ and computes $\zeta^*_{0,j} := \Sigma_0.\mathsf{Resp}(\mathsf{st}_{0,j}, \gamma^*_{0,j}, \mathbb{w}^*_{0,j})$.
As in previous games, we can show via HVZK of Σ_0 that

$$\epsilon_{18} = \epsilon_{17} \ .$$

Game 19 (Simulate for Σ_{dh}): In this game, the challenger simulates the Σ_{dh} transcript $(A^*_{\mathsf{dh},j}, \gamma^*_{\mathsf{dh},j}, \zeta^*_{\mathsf{dh},j})$ via the HVZK simulator $\Sigma_{\mathsf{dh}}.\mathsf{Sim}$. That is, in S_1 the challenger samples $\gamma^*_{\mathsf{dh},j} \leftarrow \mathbb{Z}_p$ as in Game 18, and then sets $(A^*_{\mathsf{dh},j}, \zeta^*_{\mathsf{dh},j}) \leftarrow \Sigma_{\mathsf{dh}}.\mathsf{Sim}(\mathbb{x}_{\mathsf{dh}}, \gamma^*_{\mathsf{dh},j})$, where $\mathbb{x}_{\mathsf{dh}} = (G, D_1, D_2, D_3)$. In S_2 the challenger uses $\zeta^*_{\mathsf{dh},j}$ sampled in S_1.
As before, we can show via HVZK of Σ_{dh} that

$$\epsilon_{19} = \epsilon_{18} \ .$$

Game 20 (Setup D_i at random): In this game, we setup $\mathbb{x}_{\mathsf{dh}}$ at random. That is, initially $D_1, D_2, D_3 \leftarrow \mathbb{G}$ are drawn at random and $\mathsf{H}^{\mathsf{ddh}}(0) := (D_1, D_2, D_3)$ is programmed accordingly. This step is justified by the DDH assumption such that

$$|\epsilon_{20} - \epsilon_{19}| \leq \mathsf{AdvDDH}_{\mathcal{R}^2_{\mathsf{dh}}}(\lambda)$$

for a straightforward reduction $\mathcal{R}^2_{\mathsf{dh}}$.
Finally, let us upper bound the advantage of $\mathcal{A}$ in Game 20. Recall that CMX_i and CMT are setup in binding mode and that except with probability $1/p$, it holds that $(G, D_1, D_2, D_3) \notin \mathscr{L}_{\mathsf{R}_{\mathsf{dh}}}$. As in Game 7, we can show via special soundness of Σ_0 and Σ_{dh} that $\mathbb{x}^+_0 \in \mathscr{L}_{\mathsf{R}_0}$ except with probability $(\mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}_0} + 1)/p$. By definition of R_0 (cf. Eq. (7)), if $\mathbb{x}^+_0 \in \mathscr{L}_{\mathsf{R}_0}$, then we have that $x_0 G + x_1 M^+ - Z^+_0 = 0$, where x_0 and $x_1 = 1$ are uniquely determined by the verification key vk. That is, we have

$$x_0 G + M^+ = Z^+_0 \tag{45}$$

Further, recall that $\mathcal{A}$ is only successful if $Z^+_0 \in \{x_0 G + M_j\}_{j \in \mathcal{J}_{\mathsf{F}}}$, that is there is some $j^+ \in \mathcal{J}_{\mathsf{F}}$ such that $x_0 G + M_{j^+} = Z^+_0$. Together with Eq. (45), this yields that

$$M^+ = M_{j^+}$$

But by definition of M^+ (cf. Game 5), we have that $M^+ \notin \{M_j | j \in \mathcal{J}_{\mathsf{F}}\}$. In conclusion, we have that

$$\epsilon_{20} \leq \frac{\mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}_0} + 1}{p} \ .$$

Overall, we obtain the bound

$$\begin{aligned}
\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) \leq\ & \mathsf{AdvZK}^{\Pi_{\mathsf{CT}}}_{\mathcal{R}_{\mathsf{zk}}}(\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{CT}}}) + \mathsf{AdvCRS}^{\Pi_M, \mathsf{ExtSetup}}_{\mathcal{R}_{\mathsf{extcrs}}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) \\
& + \mathsf{AdvExt}^{\Pi_M, \mathsf{Ext}}_{\mathcal{R}_{\mathsf{ext}}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) + \epsilon^{\mathsf{COM}_{\mathsf{T}}}_{\mathsf{hide}}(\lambda) \\
& + \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}^1_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) \\
& + 3 \cdot (\epsilon^{\mathsf{COM}_{\mathsf{X}}}_{\mathsf{hide}}(\lambda) + \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^1_{\mathsf{COM}_{\mathsf{X}}}}(\lambda)) \\
& + (\mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}_0} + 1)/p + \mathsf{AdvDDH}_{\mathcal{R}^1_{\mathsf{dh}}}(\lambda) \\
& + \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}^2_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) + 3 \cdot \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^2_{\mathsf{COM}_{\mathsf{X}}}}(\lambda) \\
& + 2 \lceil \log Q_{\mathsf{S}} \rceil \, \epsilon_{\mathsf{loop}}(\lambda) \\
& + \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}^3_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) + 3 \cdot \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^3_{\mathsf{COM}_{\mathsf{X}}}}(\lambda) \\
& + \mathsf{AdvDDH}_{\mathcal{R}^2_{\mathsf{dh}}}(\lambda) + (\mathsf{Q}_{\mathsf{H}^{\mathcal{CH}}_0} + 1)/p
\end{aligned}$$

where

$$
\begin{aligned}
\epsilon_{\mathsf{loop}}(\lambda) \leq\ & \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^1_{\mathsf{indcpa}}}(\lambda) \\
& +2\mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^2_{\mathsf{indcpa}}}(\lambda) \\
& +2\mathsf{AdvParamIND}^{\mathsf{COM_T}}_{\mathcal{R}^2_{\mathsf{COM_T}}}(\lambda) \\
& +6\mathsf{AdvParamIND}^{\mathsf{COM_X}}_{\mathcal{R}^2_{\mathsf{COM_X}}}(\lambda) \\
& +2(\mathsf{Q}_{\mathsf{H}_1^{\mathcal{C}\mathcal{H}}})/p \\
& +2\mathsf{AdvParamIND}^{\mathsf{COM_T}}_{\mathcal{R}^3_{\mathsf{COM_T}}}(\lambda) \\
& +6\mathsf{AdvParamIND}^{\mathsf{COM_X}}_{\mathcal{R}^3_{\mathsf{COM_X}}}(\lambda) \\
& +2\mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^3_{\mathsf{indcpa}}}(\lambda) \\
& +2\ell/p \\
& +\mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^4_{\mathsf{indcpa}}}(\lambda)\ .
\end{aligned}
\tag{46}
$$

Table 5: Transitions from Game 0 to Game 11.

Game	$\mathsf{pk}_i^{\mathsf{RE}}$	M_j	M^+	crs_M	$\pi^*_{\mathsf{CT},j}$	D_i	$\mathsf{pp}_i^{\mathsf{X}}$	pp^{T}	Σ_{dh}	$\mathsf{ct}^*_{0,j}$	Σ_0	Forgery check for M^+	Reduction
0	\$	-	-	\$	$\mathrm{w}_{\mathsf{CT},j}$	\$	\$	\$	Sim	Eval, Rerand	$\mathrm{w}^*_{0,j}$	-	OMUF Game
1	$\mathsf{sk}_i^{\mathsf{RE}}$												Uniform public-key
2					Sim								Zero-knowledge
3				Ext									CRS Indistinguishability
4		from $\pi_{M,j}$								Init, Enc			$\tilde{\mathsf{R}}_M$-Extractability Hom. correctness, Rerand. Ind.
5			$\mathcal{M}^+ \setminus \mathcal{M}_{\mathsf{F}}$										Syntax / Notation
6							B	B					Uniform Parameters
7												$Z_0^+ \in \{x_0 G\}_{j \in \mathcal{J}_{\mathsf{F}}}$	Soundness of Σ_0 and Σ_{dh}
8						DDH							DDH
9									w_{dh}				HVZK of Σ_{dh}
10											Sim		HVZK of Σ_0
11	$\mathsf{sk}_i^{\mathsf{RE}}$	from $\pi_{M,j}$	$\mathcal{M}^+ \setminus \mathcal{M}_{\mathsf{F}}$	Ext	Sim	DDH	H	H	w_{dh}	Init, Enc	Sim	$Z_0^+ \in \{x_0 G\}_{j \in \mathcal{J}_{\mathsf{F}}}$	Parameter indistinguishability

In column $\mathsf{pk}_i^{\mathsf{RE}}$, "\$" (resp. "$\mathsf{sk}_i^{\mathsf{RE}}$") means that $\mathsf{pk}_i^{\mathsf{RE}} := \mathsf{H}_{\mathsf{pk}}^{\mathsf{RE}}(i)$ is sampled at random (resp. sampled via $\mathsf{RE.Gen}(1^\lambda)$ with known secret key $\mathsf{sk}_i^{\mathsf{RE}}$) for $i \in \{0,1,2\}$. In column M_j, "-" means that M_j is undefined and "from π_M" means that M_j is extracted from $\pi_{M,j}$ in the j-th session in phase S_1^j. In column M^+, "-" means that the message M^+ is undefined, and else it means that M^+ is chosen from the set $\mathcal{M}^+ \setminus \mathcal{M}_{\mathsf{F}}$ when $\mathcal{A}$ provides its forgeries with messages $\mathcal{M}^+$. In column crs_M, "\$" (resp. "Ext") means that $\mathsf{crs}_M := \mathsf{H}_{\mathsf{crs}}^M(0)$ is sampled at random (resp. sampled via ExtSetup). In column $\pi^*_{\mathsf{CT},j}$, "$\mathrm{w}_{\mathsf{CT},j}$" (resp. "Sim") means that the proof $\pi^*_{\mathsf{CT},j}$ is computed via witness $\mathrm{w}_{\mathsf{CT},j}$ honestly (resp. simulated via the zero-knowledge simulator of Π_{CT}). In column D_i, "\$" (resp. "DDH") means that $(D_1, D_2, D_3) := \mathsf{H}^{\mathsf{ddh}}(0)$ is setup at random (resp. as valid DDH tuple with witness $\mathrm{w}_{\mathsf{dh}} := \mathsf{DLOG}(D_2)$ for $\mathscr{L}_{\mathsf{dh}}$). In column $\mathsf{pp}_i^{\mathsf{X}}$, "\$" means that parameters $\mathsf{pp}_i^{\mathsf{X}} := \mathsf{H}_{\mathsf{pp}}^{\mathsf{X}}(i)$ for $\mathsf{COM}_{\mathsf{X}}$ are chosen at random, and "H" (resp. "B") means that $\mathsf{pp}_i^{\mathsf{X}}$ is setup in hiding (resp. binding) mode. The column pp^{T} is analogous to the column $\mathsf{pp}_i^{\mathsf{X}}$ except for $\mathsf{COM}_{\mathsf{T}}$ instead of $\mathsf{COM}_{\mathsf{X}}$. In column Σ_{dh}, "Sim" (resp. "w_{dh}") means that the transcript $(A^*_{\mathsf{dh},j}, \gamma^*_{\mathsf{dh},j}, \zeta^*_{\mathsf{dh},j})$ is simulated via HVZK (resp. computed via w_{dh} honestly). In column $\mathsf{ct}^*_{0,j}$, "Eval, Rerand" means that $\mathsf{ct}^*_{0,j}$ is setup as in the construction (i.e., by evaluating InitZero on ct_M homomorphically via Eval and rerandomizing the obtained ciphertext via Rerand), and "Init, Enc" means that $\mathsf{ct}^*_{0,j}$ is setup by encrypting A_0^* obtained by evaluating Σ_0.Init. In column Σ_0, "$\mathrm{w}^*_{0,j}$" means that ζ_0^* is computed via $\mathrm{w}^*_{0,j}$ and "Sim" means that the transcript $(A^*_{0,j}, \gamma^*_{0,j}, \zeta^*_{0,j})$ is simulated via HVZK. Recall that $\mathcal{J}_{\mathsf{F}}$ denotes the set of finished sessions. In column "Forgery check for M^+", "-" means that no additional check is performed and "$Z_0^+ \in \{x_0 G\}_{j \in \mathcal{J}_{\mathsf{F}}}$" means that the game aborts if $Z_0^+ \in \{x_0 G\}_{j \in \mathcal{J}_{\mathsf{F}}}$, where $Z_0^+ := \mathsf{RE.Dec}(\mathsf{sk}_i^{\mathsf{RE}}, \mathsf{CT}_0^+)$ is the messsage encrypted in the ciphertext CT_0^+ in the signature associated to M^+. Finally, in the column "Reduction", we give a brief justification for the game hop.

Table 6: Transitions from Game 11 to Game 14. We apply the adaptive partitioning technique from [39] in Game 13.k.0 to Game 13.k.17.

							If $\beta \neq j[k+1]$				If $\beta = j[k+1]$				
Game	$\mathsf{pp}_i^{\mathsf{X}}, \mathsf{pp}^{\mathsf{T}}$	guess	x_2	$z_{2,j}$	Σ_1	Σ_2	$z_{k[0],j}$	$z_{1-k[0],j}$	Σ_1	Σ_2	$z_{0,j} = z_{1,j}$	Forgery check for M^+	Abort cond.	Reduction
Game 11	H	-	0	0	$\mathrm{w}^*_{1,j}$	$\mathrm{w}^*_{2,j}$	x_0	x_0	$\mathrm{w}^*_{1,j}$	$\mathrm{w}^*_{2,j}$	x_0	$Z_0^+ \in \{x_0 G\}_{j \in \mathcal{J}_{\mathsf{F}}}$	-	-
Game 12						$\boxed{\mathsf{Sim}}$				$\boxed{\mathsf{Sim}}$				HVZK of Σ_2
Game 13.0.0							$\mathbf{RF}_0(\varepsilon) \coloneqq x_0$	$\mathbf{RF}_0(\varepsilon) \coloneqq x_0$			$\mathbf{RF}_0(\varepsilon) \coloneqq x_0$	$Z_0^+ \in \{\mathbf{RF}_0(\varepsilon) G\}_{j \in \mathcal{J}_{\mathsf{F}}}$		Notation
Game 13.k.0							$\mathbf{RF}_k(j\|_k)$	$\mathbf{RF}_k(j\|_k)$			$\mathbf{RF}_k(j\|_k)$	$Z^+_{k[0]} \in \mathcal{Z}_k$		Begin Loop
Game 13.k.1		$\boxed{\beta}$	$\boxed{\overline{\beta}}$											Hiding
Game 13.k.2				$\boxed{j[k+1]}$										IND-CPA
Game 13.k.3													$\boxed{Z_2^+ \neq \beta G}$	Loses factor 2
Game 13.k.4						$\boxed{\mathrm{w}^*_{2,j}}$								HVZK of Σ_2
Game 13.k.5					$\boxed{\mathsf{Sim}}$									HVZK of Σ_1
Game 13.k.6								$\boxed{\mathbf{RF}_{1-k[0]}[j\|_k]}$						IND-CPA
Game 13.k.7	$\boxed{B}$													Parameter ind.
Game 13.k.8												$\boxed{Z^+_{1-k[0]} \in \mathcal{Z}_k}$		Soundness of Σ_1 and Σ_2
Game 13.k.9	$\boxed{H}$													Parameter ind.
Game 13.k.10							$\boxed{\mathbf{RF}_{1-k[0]}[j\|_k]}$							IND-CPA
Game 13.k.11					$\boxed{\mathrm{w}^*_{1,j}}$									HVZK of Σ_1
Game 13.k.12						$\boxed{\mathsf{Sim}}$								HVZK of Σ_2
Game 13.k.13							$\boxed{\mathbf{RF}_{k+1}[j\|_{k+1}]}$	$\boxed{\mathbf{RF}_{k+1}[j\|_{k+1}]}$			$\boxed{\mathbf{RF}_{k+1}[j\|_{k+1}]}$			Notation
Game 13.k.14												$\boxed{Z^+_{1-k[0]} \in \mathcal{Z}_{k+1}}$		Additive loss ℓ/p
Game 13.k.15													$\boxed{-}$	Gains factor 2
Game 13.k.16				$\boxed{0}$										IND-CPA
Game 13.k.17		$\boxed{-}$	$\boxed{0}$											Hiding
Game 14	H	-	0	0	$\mathrm{w}^*_{1,j}$	Sim	$\mathbf{F}(j)$	$\mathbf{F}(j)$	$\mathrm{w}^*_{1,j}$	Sim	$\mathbf{F}(j)$	$Z_0^+ \in \{\mathbf{F}(j) G\}_{j \in \mathcal{J}_{\mathsf{F}}}$	-	-

As before, $\mathcal{J}_{\mathsf{F}}$ denotes the indices of finished sessions. In column $\mathsf{pp}_i^{\mathsf{X}}, \mathsf{pp}^{\mathsf{T}}$, "$H$" (resp. "$B$") means that the parameters for $\mathsf{COM}_{\mathsf{X}}$ and $\mathsf{COM}_{\mathsf{T}}$ are setup in hiding (resp. binding mode). In column "guess", $\beta \leftarrow \{0,1\}$ denotes a random guess made by the game. In column x_2, the value committed in CMX_2 is given. In columns $z_{2,j}$, the DLOG of the value encrypted in $\mathsf{CT}^*_{2,j}$ given in the j-th session in phase S_1^j. Note that depending on the guess β and the $(k+1)$-th bit $j[k+1]$ of the j-th signing session, the game simulates the signing oracles in different manners. In column Σ_1, "$\mathrm{w}^*_{1,j}$" means that $\zeta^*_{1,j}$ is computed via $\mathrm{w}^*_{1,j}$ and "Sim" means that the transcript $(A^*_{1,j}, \gamma^*_{1,j}, \zeta^*_{1,j})$ is simulated via HVZK. In columns $z_{b,j}$ for $b \in \{0,1\}$, the DLOG of the value encrypted in $\mathsf{CT}^*_{b,j}$ given in the j-th session in phase S_1^j. Note that $\mathbf{RF}_k$ denotes a random function that are inductively defined by $\mathbf{RF}_{k+1}(j|_{k+1}) \coloneqq \mathbf{RF}_k(i|_k)$ if $j[k+1] = \beta$ and $\mathbf{RF}_{k+1}(j|_{k+1}) \coloneqq \mathbf{RF}'_k(i|_k)$ if $j[k+1] = 1-\beta$, where $\mathbf{RF}_0$ and $\mathbf{RF}_{1-k[0]}$ are fresh random functions. In column "Forgery check for M^+", an additional forgery check is described and the game aborts if the check fails. Here, the sets $\mathcal{Z}_k$ and $\mathcal{Z}_{k+1}$ are defined by $\mathcal{Z}_k \coloneqq \{\mathbf{RF}_k(j|_k) G\}_{j \in \mathcal{J}_{\mathsf{F}}}$ and $\mathcal{Z}_{k+1} \coloneqq \{\mathbf{RF}_{k+1}(j|_{k+1}) G\}_{j \in \mathcal{J}_{\mathsf{F}}}$, respectively. In column "Abort cond.", an additional abort condition is introduced that depends on the game's guess β. Finally, in the column "Reduction", we give a brief justification for the game hop.

Table 7: Transitions from Game 14 to Game 20.

Game	D_i	$\mathsf{pp}_i^{\mathsf{X}}$	pp^{T}	x_1	$Z_{0,j} = Z_{1,j}$	Σ_{dh}	Σ_0	Forgery check for M^+	Reduction
Game 14	DDH	H	H	0	$\mathbf{F}(j)G$	w_{dh}	Sim	$Z_0^+ \in \{\mathbf{F}(j)G\}_{j\in\mathcal{J}_{\mathsf{F}}}$	-
Game 15				1	$\mathbf{F}(j)G + M_j$			$Z_0^+ \in \{\mathbf{F}(j)G + M_j\}_{j\in\mathcal{J}_{\mathsf{F}}}$	Statistical
Game 16					$x_0G + M_j$			$Z_0^+ \in \{x_0G + M_j\}_{j\in\mathcal{J}_{\mathsf{F}}}$	Backward partitioning (cf. Table 6)
Game 17		B	B						Parameter ind.
Game 18							$\mathrm{w}^*_{0,j}$		HVZK of Σ_0
Game 19						Sim			HVZK of Σ_{dh}
Game 20	\$								DDH

In column D_i, "\$" (resp. "DDH") means that $(D_1, D_2, D_3) \coloneqq \mathsf{H}^{\mathsf{ddh}}(0)$ is setup at random (resp. as valid DDH tuple with witness $\mathrm{w}_{\mathsf{dh}} = \mathsf{DLOG}(D_2)$ for $\mathscr{L}_{\mathsf{dh}}$). In column $\mathsf{pp}_i^{\mathsf{X}}$, "\$" means that parameters $\mathsf{pp}_i^{\mathsf{X}} \coloneqq \mathsf{H}_{\mathsf{pp}}^{\mathsf{X}}(i)$ for $\mathsf{COM}_{\mathsf{X}}$ are chosen at random and "H" (resp. "B") means that $\mathsf{pp}_i^{\mathsf{X}}$ is setup in hiding (resp. binding) mode. The column pp^{T} is analogous to the column $\mathsf{pp}_i^{\mathsf{X}}$ except for $\mathsf{COM}_{\mathsf{T}}$ instead of $\mathsf{COM}_{\mathsf{X}}$. In column x_1, the value committed in CMX_1 is given. In column $Z_{0,j} = Z_{1,j}$, the value encrypted in ciphertexts $\mathsf{CT}_{0,j}$ and $\mathsf{CT}_{1,j}$ in the j-th session is given. In column Σ_{dh}, "Sim" (resp. "w_{dh}") means that the transcript $(A^*_{\mathsf{dh},j}, \gamma^*_{\mathsf{dh},j}, \zeta^*_{\mathsf{dh},j})$ is simulated via HVZK (resp. computed via w_{dh} honestly). The column Σ_0 is interpreted analogously. In column "Forgery check for M^+", an additional forgery check is described and the game aborts if the check fails. Finally, in the column "Reduction", we give a brief justification for the game hop.

This concludes the one-more unforgeability proof. □

Below, we provide the missing proofs of aforementioned lemmata.

Lemma 6. *This step is justified by the $\tilde{\mathsf{R}}_M$-extractability of Π_M such that*

$$|\epsilon_4 - \epsilon_3| \leq \mathsf{AdvExt}^{\Pi_M,\mathsf{Ext}}_{\mathcal{R}_{\mathsf{ext}}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}) \ .$$

Proof (of Lemma 6). We show this by introducing intermediate games between Game 3 and Game 4. Roughly, we first extract M_j from $\pi_{M,j}$ and abort if the extracted witness is not in the desired relation. Because extraction is not efficient for M_j, the game becomes inefficient at this point. Then, we use M_j to construct $\mathsf{ct}^*_{0,j}$ as a fresh ciphertext (indepdenent on ct_0). In the last step, we remove the abort condition again and obtain Game 4.

Game* 3.1: Let us define an intermediate hybrid Game 3.1. In this game, the challenger extracts M_j from $\pi_{M,j}$ in S_1 as described in Game 4, but the encryption $\mathsf{ct}^*_{0,j}$ is still computed as in Game 3 (i.e., via Eval and Rerand). After extraction, the challenger aborts if there are no $(\mathsf{sk}^{\mathsf{LHE}}, r_{M,j})$ such that

$$\mathsf{ct}_{M,j} = \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}, M_j; r_{M,j}) \wedge (\mathsf{pk}^{\mathsf{LHE}}, \mathsf{sk}^{\mathsf{LHE}}) \in \mathsf{LHE.Gen}(1^\lambda). \tag{47}$$

Note that this abort condition is inefficient.
This step is justified by the straighline $\tilde{\mathsf{R}}_M$-extractability (cf. Definition 27) such that

$$\epsilon_{3.1} - \epsilon_3 \leq \mathsf{AdvExt}^{\Pi_M,\mathsf{Ext}}_{\mathcal{R}_{\mathsf{ext}}}(\lambda, \mathsf{Q}_{\mathsf{H}_M}).$$

Game* 3.2: In this game, the challenger also computes $\mathsf{ct}^*_{0,j}$ as in Game 4. By Eq. (47) and by homomorphic correctness, we have that there exists some $r_{\mathsf{eval},j}$ such that $\mathsf{ct}_{0,j} = \mathsf{LHE.Enc}(\mathsf{pk}^{\mathsf{LHE}}, \mathsf{InitZero}(M); r_{\mathsf{eval},j})$. By definition of InitZero (cf. Eq. (11)), we have $\mathsf{InitZero}(M_j) = A^*_{0,j}$, where $A^*_{0,j} \leftarrow \Sigma_0.\mathsf{Init}(\mathsf{st}_{0,j}, \phi^0_{\mathrm{x}^*_0})$. Thus, rerandomization indistinguishability of LHE yields that

$$\epsilon_{3.2} = \epsilon_{3.1}$$

Finally, observe that the only difference between Game 4 and Game 3.3 is the added abort condition. Removing the abort condition at most improves the advantage of $\mathcal{A}$ in Game 4, and we have

$$\epsilon_4 \leq \epsilon_{3.2}$$

Note that Game 4 is efficient again.

□

Lemma 7. *There are appropriate reductions such that* $|\epsilon_{16} - \epsilon_{15}|(\lambda) \leq \epsilon_{\mathsf{loop}}(\lambda)$ *as defined in Eq.* (46).

Proof (of Lemma 7). Let us argue that $\mathcal{A}$'s advantage in Game 15 and Game 16 is close. Roughly, we start with Game 16 and perform the game transitions from Game 13.k.0 to Game 13.k.17 for $k \in [\lceil \log Q_{\mathsf{S}} \rceil]$ in reverse order. That is, we "deconstruct" $\mathbf{RF}_{k+1}$ into $\mathbf{RF}_k$. Below, $\mathbf{RF}_k$ is a random function defined as in Eq. (43). We obtain Game 16 after $\lceil \log Q_{\mathsf{S}} \rceil$ steps. Most arguments are identical to before and we use matching notation for readability. That is, we denote sets and values with similar meaning in the same manner, even if their definition slightly differs.

Game R.0 ($\hat{=}$13.k.17; Start of partitioning loop): This game is identical Game 13.k.17 except that $Z_{0,j} = Z_{1,j} = \mathbf{RF}_{k+1}[j|_{k+1}] \cdot G + M_j$ is encrypted in $\mathsf{CT}^*_{0,j}$ and $\mathsf{CT}^*_{1,j}$. Also, the forgery check $Z^+_{1-k[0]} \in \mathcal{Z}_{k+1}$ is performed, where $\mathcal{Z}_{k+1}$ is defined as $\mathcal{Z}_{k+1} = \{\mathbf{RF}_{k+1}(j|_{k+1}) \cdot G + M_j\}_{j \in \mathcal{J}_{\mathsf{F}}}$.
By setting $\mathbf{F} = \mathbf{RF}_{k+1}$ and observing that $j|_{k+1} = j$, we have for $k = \lceil \log \mathsf{Q}_{\mathsf{S}} \rceil - 1$ that

$$\epsilon_{\mathrm{R}.0} = \epsilon_{13.k.17} \ .$$

Game R.1 ($\hat{=}$13.k.16; Sample β and set $x_2 \coloneqq \overline{\beta}$): In this game, the challenger initially samples a random bit $\beta \leftarrow \{0,1\}$ and sets $x_2 \coloneqq \overline{\beta}$.
As $\mathsf{pp}^{\mathsf{X}}_2$ is setup in hiding mode and the values x_2, β are not used elsewhere, we have

$$\epsilon_{\mathrm{R}.1} = \epsilon_{\mathrm{R}.0} \ .$$

Game R.2 ($\hat{=}$13.k.15; Set $z_{2,j} \coloneqq j[k+1]$): In this game, the challenger sets $z_{2,j} = j[k+1]$ in the j-th signing session.
As in Game 13.k.16, this step is justified by the IND-CPA security of RE such that

$$|\epsilon_{\mathrm{R}.2} - \epsilon_{\mathrm{R}.1}| \leq \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^1_{\mathsf{indcpa}}}(\lambda) \ .$$

Game R.3 ($\hat{=}$13.k.14; Abort if $Z^+_2 \neq \beta G$): After $\mathcal{A}$ outputs its forgeries $(M^+_\kappa, \sigma^+_\kappa)_{\kappa \in [\ell+1]}$, the challenger samples another bit $b \leftarrow \{0,1\}$ and aborts if (1) $Z^+_2 \in \{1, G\}$ and $Z^+_2 = (1-\beta)G$ or if (2) $Z^+_2 \notin \{1, G\}$ and $b = 0$.
As $\mathsf{pp}^{\mathsf{X}}_2$ is setup in hiding mode, the bits β and b are independent of $\mathcal{A}$'s view. As in Game 13.k.3 through 13.k.14 the abort probability in this game is $1/2$. Thus, we have

$$\epsilon_{\mathrm{R}.3} = \epsilon_{\mathrm{R}.2}/2 \ .$$

Game R.4 ($\hat{=}$13.k.13; Abort if $Z^+_{1-k[0]} \notin \mathcal{Z}_k$): In this game, we change the forgery check. That is, instead of checking that $Z^+_{1-k[0]} \in \mathcal{Z}_{k+1}$, the challenger checks that $Z^+_{1-k[0]} \in \mathcal{Z}_k$, where $\mathcal{Z}_k$ and $\mathcal{Z}_{k+1}$ are defined as

$$\begin{aligned}
\mathcal{Z}_{k+1} &= \{\mathbf{RF}_{k+1}(j|_{k+1}) \cdot G + M_j\}_{j \in \mathcal{J}_{\mathsf{F}}} \\
&= \underbrace{\{\mathbf{RF}_k(j|_k) \cdot G + M_j \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \beta\}}_{=\mathcal{S}_{\mathsf{k.both}}} \cup \underbrace{\{\mathbf{RF}'_k(j|_k) \cdot G + M_j \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \overline{\beta}\}}_{\coloneqq \mathcal{S}_{k.15}}, \\
\mathcal{Z}_k &= \{\mathbf{RF}_k(j|_k) \cdot G + M_j\}_{j \in \mathcal{J}_{\mathsf{F}}} \\
&= \underbrace{\{\mathbf{RF}_k(j|_k) \cdot G + M_j \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \beta\}}_{\coloneqq \mathcal{S}_{\mathsf{k.both}}} \cup \underbrace{\{\mathbf{RF}_k(j|_k) \cdot G + M_j \mid j \in \mathcal{J}_{\mathsf{F}}, j[k+1] = \overline{\beta}\}}_{\coloneqq \mathcal{S}_{k.14}},
\end{aligned}$$

where the last equality follows by definition of $\mathbf{RF}_{k+1}$ (cf. Eq. (43)).
Recall that in Game 13.k.15, the challenger also accepts the values $Z_{1-k[0]} \in \mathcal{S}_{k.15}$, whereas not all such values are accepted in Game 13.k.14. As before, this impacts the advantage of $\mathcal{A}$ by at most an additive loss ℓ/p because the values that are not accepted in $\mathcal{S}_{k.15}$ are statistically hidden, and thus hard to predict. Analogously, in this game the challenger accepts forgeries $Z_{1-k[0]} \in \mathcal{S}_{k.15}$, but this at most improves the advantage of $\mathcal{A}$ compared to Game R.3. Thus, we have

$$|\epsilon_{\mathrm{R}.4} - \epsilon_{\mathrm{R}.3}| \leq \frac{\ell}{p} \ .$$

Game R.5 ($\hat{=}13.k.12$; Rewrite $Z_{0,j}$ and $Z_{1,j}$ in terms of $\mathbf{RF}_k$ and $\mathbf{RF}'_k$): This is a purely conceptual game. As the challenger sets $Z_{0,j} = Z_{1,j} = \mathbf{RF}_{k+1}(j|_{k+1})G + M_j$ in Game R.4, where $\mathbf{RF}_{k+1}$ is defined as in Eq. (43), we can write $Z_{0,j}$ in terms of $\mathbf{RF}'_k$ (resp. $\mathbf{RF}_k$) in the j-th signing session if $\beta \neq j[k+1]$ (resp. $\beta = j[k+1]$). We have

$$\epsilon_{\mathrm{R}.5} = \epsilon_{\mathrm{R}.4} \ .$$

Game R.6 ($\hat{=}13.k.11$; Use witness for Σ_2 if $\beta \neq j[k+1]$): In this game, the challenger computes the Σ_2 transcript $(A^*_{2,j}, \gamma^*_{2,j}, \zeta^*_{2,j})$ with witness $\mathbb{w}^*_{2,j} = ((x_2, \hat{s}_2), (Z_{2,j}, \rho^*_{2,j}))$ if $\beta \neq j[k+1]$. By HVZK of Σ_1, we have

$$\epsilon_{\mathrm{R}.6} = \epsilon_{\mathrm{R}.5} \ .$$

Game R.7 ($\hat{=}13.k.10$; Simulate for Σ_1 if $\beta \neq j[k+1]$): In this game, the challenger simulates the transcript $(A^*_{1,j}, \gamma^*_{1,j}, \zeta^*_{1,j})$ via the HVZK simulator Sim_1 in the j-th session and computes $\gamma^*_{2,j}$ and $\gamma^*_{1,j}$ accordingly, if $\beta \neq j[k+1]$. Under HVZK of Σ_1, we have that

$$\epsilon_{\mathrm{R}.7} = \epsilon_{\mathrm{R}.6} \ .$$

Game R.8 ($\hat{=}13.k.9$; Set $z_{k[0],j} = \mathbf{RF}_k[j|_k]$ if $\beta \neq j[k+1]$): Let $k[0] \coloneqq k \bmod 2$. In this game, if $\beta \neq j[k+1]$ in the j-th session, the challenger sets $z_{k[0],j} \coloneqq \mathbf{RF}'_k[j|_k]$. Then, it sets $\mathsf{CT}^*_{k[0],j} \coloneqq \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}_{k[0]}, Z^*_{k[0]}; \rho^*_{k[0],j})$, where $Z^*_{k[0]} \coloneqq z_{k[0]}G + M_j$ as before.
As Σ_0 and Σ_1 are simulated, the values $z_{k[0],j}$ and $\rho_{k[0],j}$ are only used to initialize $\mathsf{CT}^*_{k[0],j}$. Also, the secret key $\mathsf{sk}_{k[0]}$ associated to $\mathsf{pk}^{\mathsf{RE}}_{k[0]}$ is not required for the simulation (as the forgery checks for M^+ is performed with $\mathsf{sk}_{1-k[0]}$, and the abort condition is evaluated with sk_2). This step is justified by the IND-CPA security of RE such

$$|\epsilon_{\mathrm{R}.8} - \epsilon_{\mathrm{R}.7}| \leq \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^2_{\mathsf{indcpa}}}(\lambda) \ .$$

Game R.9 ($\hat{=}13.k.8$; Setup $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} in binding mode): In this game, the challenger sets up the public parameters of $\mathsf{COM}_{\mathsf{X}}$ and $\mathsf{COM}_{\mathsf{T}}$ in binding mode. This step is justified by the parameter indistinguishability such that

$$|\epsilon_{\mathrm{R}.9} - \epsilon_{\mathrm{R}.8}| \leq \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}^1_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) + 3 \cdot \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^1_{\mathsf{COM}_{\mathsf{X}}}}(\lambda) \ .$$

Game R.10 ($\hat{=}13.k.7$; Abort if $Z^+_{k[0]} \notin \mathcal{Z}_k$): In this game, instead of checking that $Z^+_{1-k[0]} \in \mathcal{Z}_k$, the challenger aborts if the value $Z^+_{k[0]}$ encrypted in the ciphertext $\mathsf{CT}^+_{k[0]}$ in $\mathcal{A}$'s forgery associated to M^+ does not lie in $\mathcal{Z}_k$.
Because $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} are setup in binding mode, this follows by soundness of π^+_1 for relation R_1 as in Game R.9, and we obtain

$$|\epsilon_{\mathrm{R}.10} - \epsilon_{\mathrm{R}.9}| \leq \frac{\mathsf{Q}_{\mathsf{H}^{\mathcal{C}\mathcal{H}}_1}}{p} \ .$$

Game R.11 ($\hat{=}13.k.6$; Setup $\mathsf{pp}^{\mathsf{X}}_i$ and pp^{T} in hiding mode): In this game, the challenger sets up the public parameters of $\mathsf{COM}_{\mathsf{X}}$ and $\mathsf{COM}_{\mathsf{T}}$ in hiding mode. This step is justified by the parameter indistinguishability such that

$$|\epsilon_{\mathrm{R}.11} - \epsilon_{\mathrm{R}.10}| \leq \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{T}}}_{\mathcal{R}^2_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) + 3 \cdot \mathsf{AdvParamIND}^{\mathsf{COM}_{\mathsf{X}}}_{\mathcal{R}^2_{\mathsf{COM}_{\mathsf{T}}}}(\lambda) \ .$$

Game R.12 ($\hat{=}13.k.5$; Set $z_{1-k[0],j} \coloneqq \mathbf{RF}_k[j|_k]$ if $\beta \neq j[k+1]$): In this game, if $\beta \neq j[k+1]$ in the j-th session, the challenger sets $z_{1-k[0],j} \coloneqq \mathbf{RF}_k[j|_k]$. Then, it sets $\mathsf{CT}^*_{1-k[0],j} \coloneqq \mathsf{RE.Enc}(\mathsf{pk}^{\mathsf{RE}}_{1-k[0]}, Z_{1-k[0],j}; \rho^*_{1-k[0],j})$, where $Z_{1-k[0],j} \coloneqq z_{1-k[0],j}G + M_j$ as before.
Observe that as Σ_0 and Σ_1 are simulated, the values $z_{1-k[0],j}$ and $\rho^*_{1-k[0],j}$ are only used to initialize $\mathsf{CT}^*_{1-k[0],j}$. Also, the secret key $\mathsf{sk}_{1-k[0]}$ associated to $\mathsf{pk}^{\mathsf{RE}}_{1-k[0]}$ is not required for the simulation (as the forgery checks for M^+ is performed with $\mathsf{sk}_{k[0]}$ and the abort condition is evaluated with sk_2). This step is justified by the IND-CPA security of RE such

$$|\epsilon_{\mathrm{R}.12} - \epsilon_{\mathrm{R}.11}| \leq \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^3_{\mathsf{indcpa}}}(\lambda) \ .$$

Game R.13 ($\hat{=}$13.k.4; Use witness for Σ_1 if $\beta \neq j[k+1]$): In this game, the challenger computes the Σ_1 transcript $(A^*_{1,j}, \gamma^*_{1,j}, \zeta^*_{1,j})$ via the witness $\mathrm{w}^*_{1,j} = ((Z_{0,j}, \rho^*_{0,j}), (Z_{1,j}, \rho^*_{1,j}))$ if $\beta \neq j[k+1]$. By HVZK of Σ_1, we have

$$\epsilon_{\mathrm{R.13}} = \epsilon_{\mathrm{R.12}} \ .$$

Game R.14 ($\hat{=}$13.k.3; Simulate Σ_2 if $\beta \neq j[k+1]$): In this game, the challenger simulates the transcript $(A^*_{2,j}, \gamma^*_{2,j}, \zeta^*_{2,j})$ via the Σ_2's HVZK simulator and computes $\gamma^*_{2,j}$ and $\gamma^*_{1,j}$ accordingly, if $\beta \neq j[k+1]$ in the j-th signing session. By HVZK, we have under HVZK of Σ_2 that

$$\epsilon_{\mathrm{R.14}} = \epsilon_{\mathrm{R.13}} \ .$$

Game R.15 ($\hat{=}$13.k.2; Do not abort if $Z_2^+ \neq \beta G$): Recall that in Game R.14, after $\mathcal{A}$ outputs its forgeries $(M^+_\kappa, \sigma^+_\kappa)_{\kappa \in [\ell+1]}$, the challenger samples a bit $b \leftarrow \{0,1\}$ and aborts if (1) $Z_2^+ \in \{1, G\}$ and $Z_2^+ = (1-\beta)G$ or if (2) $Z_2^+ \notin \{1, G\}$ and $b = 0$.
As $\mathsf{pp}_2^{\mathsf{X}}$ is setup in hiding mode, the bits β and b are independent of $\mathcal{A}$'s view due to the above modifications. Importantly, the advantage of $\mathcal{A}$ is independent of the bits β and b. As in Game 13.k.3, we can show that the abort probability in Game R.15is exactly $1/2$. Thus, by removing the abort condition, the advantage of $\mathcal{A}$ doubles. That is, we have

$$\epsilon_{\mathrm{R.15}} = 2\epsilon_{\mathrm{R.14}} \ .$$

Game R.16 ($\hat{=}$13.k.1; Set $z_{2,j} \coloneqq 0$): In this game, the challenger sets $z_{2,j} = 0$ in the j-th signing session. As from Game 13.k.1 to Game 13.k.2, this step is justified by the IND-CPA security of RE such that

$$|\epsilon_{\mathrm{R.16}} - \epsilon_{\mathrm{R.15}}| \leq \mathsf{AdvINDCPA}^{\mathsf{RE}}_{\mathcal{R}^4_{\mathsf{indcpa}}}(\lambda) \ .$$

Game R.17 ($\hat{=}$13.k.0; Forget β and set $x_2 \coloneqq 0$): In this game, the challenger does not sample the bit β anymore. Also, it sets $x_2 \coloneqq 0$. As $\mathsf{pp}_2^{\mathsf{X}}$ is setup in hiding mode and the values x_2, β are not used elsewhere, we have

$$\epsilon_{\mathrm{R.17}} = \epsilon_{\mathrm{R.16}} \ .$$

A simple inspection yields that Game R.17 is identical to Game R.0.

□