Playing Tag with Okamoto-Schnorr: Three-Move Pairing-Free Blind Signatures from DDH *

Rutchathon Chairattana-Apirom[1], Michael Reichle[2], and Stefano Tessaro[1]

[1] Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, US
{rchairat,tessaro}@cs.washington.edu
[2] Department of Computer Science
ETH Zurich, Switzerland
michael.reichle@inf.ethz.ch

Abstract. This paper presents the first blind signature scheme in a pairing-free group with the following properties: (1) the signing protocol consists of only three moves; (2) the proof of one-more unforgeability relies solely on the Decisional Diffie–Hellman (DDH) assumption in the Random Oracle Model (ROM); and (3) the construction makes only black-box use of the underlying group. This resolves a major open problem in the area, as all prior pairing-free blind signatures either additionally relied on the Algebraic Group Model (AGM) or required at least *four* moves. Moreover, a recent lower bound by Dietz *et al.* (ePrint, '26) shows that three moves are optimal for such constructions.

Both the communication complexity and the signature size in our scheme consist of a constant number of group elements. Our construction in fact achieves *strong* one-more unforgeability (which was not known for any of the recent AGM-free constructions requiring four moves), and we also present a partially blind variant. Furthermore, blindness is statistical (in the ROM). Our approach is based on a new construction paradigm that combines a conventional (yet, by itself, not fully secure) blind signature scheme (specifically, the blind Okamoto–Schnorr scheme) with a carefully crafted algebraic MAC.

* ©IACR 2026. This is the full version of an article that will be published in the proceedings of CRYPTO 2026.

Table of Contents

1 Introduction . . . 3
1.1 Technical Overview . . . 4
1.2 Related and Future Work . . . 8
2 Preliminaries . . . 9
3 Three-move Blind Signatures from Algebraic MACs . . . 15
3.1 Building Blocks . . . 15
3.2 Construction . . . 18
3.3 Correctness and Security . . . 20
4 Instantiations . . . 23
4.1 Compact Signatures . . . 23
4.2 Statistical Blindness . . . 24
5 Correctness Proof of $\mathsf{BS}_{\mathrm{alg}}$ . . . 25
6 OMSUF Proof of $\mathsf{BS}_{\mathrm{alg}}$ . . . 27
6.1 Proof of Theorem 3.5 . . . 27
6.2 Proof of Lemma 6.2 (Extracting DLOG from Nonce Collision) . . . 30
6.3 Proof of Lemma 6.3 (Extracting MAC Forgeries from Distinct Nonces) . . . 38
7 Blindness Proof of $\mathsf{BS}_{\mathrm{alg}}$ . . . 48
A Proof of Finished-Session OMUF . . . 55
B Proof of Forking Lemma . . . 55
C Proof of Theorem 3.1 . . . 57

1 Introduction

Blind signatures [Cha82] enable a user to obtain a signature on a chosen message from a signer, without the signer learning either the message being signed or the resulting signature. They play a central role in privacy-preserving authentication. For example, they are widely used in industry proposals [HIP+21, DGS+18, CDVW24] for anonymous authentication tokens, as well as in solutions for fraud-detection in privacy-preserving ad-click measurement [PCM, HIJ+21, Tru]. Other examples are iCloud Private Relay [App] and Google One VPN Service [Goo]. Furthermore, blind signatures underlie several proposals for electronic cash [Cha82, CFN90, OO92] and voting [FOO93].

Pairing-free blind signatures. A very natural question, which has been studied over more than three decades, is whether blind signatures can be built in pairing-free groups, as many classical signatures, such as Schnorr signatures [Sch90], EdDSA [BDL+12], and ECDSA [Ame05] rely on such groups, and it is natural to expect blind versions of these (or related) signatures. This is also a practically relevant question since pairing-free elliptic curves are widely supported by cryptography libraries, in contrast to pairing-friendly curves. The latter, while enabling much simpler blind signatures (such as Blind BLS [Bol03]), are not as widely supported and have less certain security guarantees (as they are subject to asymptotically faster discrete logarithm attacks in the target group). This problem is also theoretically interesting, as pairing-free groups often constitute the simplest algebraic structure enabling efficient public key cryptography.

For such schemes, a major challenge is achieving the standard security notion of *one-more unforgeability (OMUF)* [JLO97], which is expected to hold against malicious users that interact with the signer in multiple *concurrent* sessions. Recently, Dietz *et al.* [DKT26] established that a large class of constructions that can be described in a combination of the generic group model (GGM) [Sho97, Mau05] and random oracle model (ROM) [BR93] require at least three moves between the signer and the user. This class of constructions captures exactly those constructions that make black-box use of the underlying group—these are generally more efficient. But if a blind signature requires more than a single round trip, then concurrent security is not equivalent to sequential security. This often leads to concurrent attacks on natural instantiations.

For example, the blind version of Schnorr signatures [CP93] is easily compromised by concurrent attacks efficiently solving the ROS problem [BLL+21]. This attack extends to a number of other blind signatures based on linear identification, such as blind Okamoto-Schnorr signatures [Oka94]. At the same time, proofs of security for a small number of concurrent sessions (and hence not contradicting the attacks) have been given in the literature [PS00, HKL19, FPS20, KLX22b, KLX22a].

Only recently, a number of efficient three-move blind signatures [FPS20, KLX22b, TZ22, CKM+23] have been given, all with proofs of full OMUF security (i.e., no bounded number of sessions) assuming a combination of the Algebraic Group Model (AGM) [FKL18] and the ROM. Despite this being substantial progress, there is much debate around the assurance we should infer from an AGM proof. Regardless of someone's position in this debate, it is certainly undesirable (and uncommon) if no solution without the AGM exists at all.

Avoiding the AGM. As a testament of the challenge of avoiding reliance on ideal models, Abe [Abe01] proposed a three-move pairing-free blind signature with a proof in the ROM (under DDH) which was later discovered to be flawed and could only be repaired in the GGM+ROM [OA03], and later in the AGM+ROM [KLX22b]. Only very recently, the question of removing reliance on the AGM was fully resolved by a number of pairing-free blind signatures [CATZ24, KRW24, BHKR25, KR25, RR26, KLR25] with security proofs only relying on the ROM and standard assumptions in pairing-free groups. These constructions are summarized in Table 1, along with those proved secure in the AGM+ROM.

It is however noteworthy that without relying on the AGM, to date, no three-move construction is known. This leaves us with the following open question:

> *Can we design a three-move pairing-free blind signature in the ROM for which the security proof does not rely on the AGM?*

This paper fully resolves this question by presenting the first three-move pairing-free blind signature for which OMUF is proved in the ROM under the DDH assumption. Furthermore, blindness is statistical (in

Table 1. Pairing-free blind signature schemes. We present constructions with full OMUF (i.e., there is no limit on the number of concurrent signing sessions) that make black-box use of the group. The schemes below (resp. above) the line rely on the ROM (resp. ROM+AGM). Our schemes from Section 4.1 (resp. Section 4.2) satisfy computational (resp. statistical) blindness. (†): These schemes satisfy a weaker one-more unforgeability notion.

Scheme	Moves	Signature	Communication	Assumptions	OMSUF
Clause Blind Schnorr [FPS20]	3	$1\mathbb{G} + 1\mathbb{Z}_p$	$2\mathbb{G} + 3\mathbb{Z}_p$	OMDL, mROS	✓
Abe [Abe01, KLX22b]	3	$2\mathbb{G} + 6\mathbb{Z}_p$	$\lambda + 3\mathbb{G} + 6\mathbb{Z}_p$	DL	✓
TZ [TZ22]	3	$4\mathbb{Z}_p$	$2\mathbb{G} + 4\mathbb{Z}_p$	DL	✓
Snowblind [CKM+23]	3	$1\mathbb{G} + 2\mathbb{Z}_p$	$2\mathbb{G} + 4\mathbb{Z}_p$	DL	✓
CTZ-1 [CATZ24]†	4	$1\mathbb{G} + 4\mathbb{Z}_p$	$5\mathbb{G} + 5\mathbb{Z}_p$	CT-OMCDH	✗
CTZ-2 [CATZ24]†	5	$1\mathbb{G} + 4\mathbb{Z}_p$	$5\mathbb{G} + 5\mathbb{Z}_p$	CT-OMCDH	✓
CTZ-3 [CATZ24]	4	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$	CDH	✗
KRW [KRW24]	4	$2\mathbb{G} + 5\mathbb{Z}_p$	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$	DDH	✗
BHKR [BHKR25]	4	$10\mathbb{G} + 29\mathbb{Z}_p$	$37\mathbb{G} + 40\mathbb{Z}_p$	DDH	✗
KR-1 [KR25]	4	$1\mathbb{G} + 5\mathbb{Z}_p$	$10\mathbb{G} + 9\mathbb{Z}_p$	DDH	✗
KR-2 [KR25]	5	$1\mathbb{G} + 6\mathbb{Z}_p$	$10\mathbb{G} + 9\mathbb{Z}_p$	DDH	✓
RR [RR26]	4	$2\mathbb{G} + 8\mathbb{Z}_p$	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$	CDH	✗
KLR [KLR25]	4	$6\mathbb{Z}_p$	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$	DL	✗
Our work (cf. Section 4.1)	3	$15\mathbb{G} + 22\mathbb{Z}_p$	$17\mathbb{G} + 14\mathbb{Z}_p$	DDH	✓
Our work (cf. Section 4.2)	3	$21\mathbb{G} + 35\mathbb{Z}_p$	$17\mathbb{G} + 14\mathbb{Z}_p$	DDH	✓

the ROM). As stated above, by [DKT26], the round complexity of our construction is best possible, as it only makes generic use of the underlying group. Both communication and signatures consist of a constant number of group elements.

Additionally, we prove our construction to be one-more *strongly* unforgeable (OMSUF), and this was so far only known to be achievable by *five*-move constructions without the AGM. We also give a version of our construction which is partially blind, i.e., signing can also depend on a public input.

We give further details about our construction paradigm in the technical overview below.

On non-black-box constructions. We stress that one *can* obtain three- (or even two-) move pairing-free constructions fairly generically by using Fischlin's construction [Fis06], in conjunction with non-black-box use of the group (e.g., by using generic NIZKs for statements that depend on the group). Recent work shows that the construction can in fact be instantiated (in the ROM) from a one-way function alone [CMV26]. However, these instantiations are generally much less efficient than what can be achieved by constructions using the group in a black-box way. Recently, Kastner *et al.* [KTZ26] presented a more efficient instantiation of this paradigm, although still relying on generic zero-knowledge. Their security proof also relies on the AGM and other non-standard assumptions.

1.1 Technical Overview

Our starting point is the Okamoto-Schnorr blind signature [PS00], hereafter denoted $\mathsf{BS}_{\mathsf{os}}$, based on the Okamoto-Schnorr identification scheme [Sch90, Oka93]. We give a brief recap. The scheme is parameterized by generators G and H of some group $\mathbb{G}$ of prime-order p. Note that we employ additive notation for $\mathbb{G}$. The public key is $W = w_0 G + w_1 H$ with secret key $\boldsymbol{w} = (w_0, w_1) \leftarrow\!\!\$\, \mathbb{Z}_p^2$. For notational convenience, we denote by ϕ_{os} the linear map such that $\phi_{\mathsf{os}}(\boldsymbol{w}) = W$. To sign a message μ, the user engages in a 3-move protocol with the signer. Roughly, the signer-user interaction issues a Σ-protocol transcript $\tau_{\mathsf{os}} = (A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ for knowledge of $\boldsymbol{w}$ to the user, while the user blinds the transcript τ_{os} and compiles it into a Fiat-Shamir proof $\widetilde{\tau}_{\mathsf{os}} = (\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$, where $\widetilde{c}_{\mathsf{os}} = \mathsf{H}(W, \widetilde{A}_{\mathsf{os}}, \mu)$. In more detail, a signing session for message μ proceeds as follows.

Move 1 ($\mathsf{S} \to \mathsf{U}$)**.** The signer S samples $\boldsymbol{r} \leftarrow\!\!\$\ \mathbb{Z}_p^2$ and computes the nonce $A_{\mathsf{os}} \leftarrow \phi_{\mathsf{os}}(\boldsymbol{r})$. Then, it sends A_{os} to the user U.

Move 2 ($\mathsf{U} \to \mathsf{S}$)**.** The user U blinds A_{os} into $\widetilde{A}_{\mathsf{os}}$, computes $\widetilde{c}_{\mathsf{os}} \leftarrow \mathsf{H}(W, \widetilde{A}_{\mathsf{os}}, \mu)$ and blinds $\widetilde{c}_{\mathsf{os}}$ into c_{os}. That is, the user samples $\boldsymbol{\alpha} \leftarrow\!\!\$\ \mathbb{Z}_p^2$ and $\gamma \leftarrow\!\!\$\ \mathbb{Z}_p$. Next, it sets $\widetilde{A}_{\mathsf{os}} \leftarrow A_{\mathsf{os}} + \phi_{\mathsf{os}}(\boldsymbol{\alpha}) - \gamma W$ and $c_{\mathsf{os}} = \widetilde{c}_{\mathsf{os}} - \gamma$. Then, U sends the blinded challenge c_{os} to the signer S.

Move 3 ($\mathsf{S} \to \mathsf{U}$)**.** The signer S sends the Σ-protocol response $z_{\mathsf{os}} \leftarrow c_{\mathsf{os}}\boldsymbol{w} + \boldsymbol{r}$ to U.

Derivation. The user U blinds z_{os} into $\widetilde{z}_{\mathsf{os}} \leftarrow z_{\mathsf{os}} + \boldsymbol{\alpha}$ and sets $\widetilde{\tau}_{\mathsf{os}} = (\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$. The derived signature is $\sigma \leftarrow \widetilde{\tau}_{\mathsf{os}}$.

Verification. Given $\sigma = (\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ for message μ, check that $\phi_{\mathsf{os}}(\widetilde{z}_{\mathsf{os}}) = \widetilde{A}_{\mathsf{os}} + \widetilde{c}_{\mathsf{os}} \cdot W$ and $\widetilde{c}_{\mathsf{os}} = \mathsf{H}(W, \widetilde{A}_{\mathsf{os}}, \mu)$.

Note that σ verifies by linearity of ϕ_{os}. The scheme is proven one-more unforgeable for $\mathsf{polylog}$-many signing sessions [PS00, HKL19]. This bound is tight due to an efficient attack on the ROS problem by Benhamouda et al. [BLL+21, BLL+22] which undermines the concurrent security of $\mathsf{BS}_{\mathsf{os}}$. We aim to restore security of $\mathsf{BS}_{\mathsf{os}}$ with additional measures. For exposition, it will be instructive to give intuition for the security argument and why it fails to generalize to many concurrent sessions. We sketch the argument for one-more unforgeability (OMUF) and discuss the extension to one-more *strong* unforgeability (OMSUF) later.

OMUF analysis for Okamoto-Schnorr. Recall that in the OMUF game, the adversary obtains the public key W, and access to signing oracles S_1 and S_2 which model the 1-st and 2-nd signer message in the signing protocol, respectively. In particular, the adversary $\mathcal{A}$ can obtain valid transcripts $\tau_{\mathsf{os}} = (A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ for the public key W via oracles S_1 and S_2, where c_{os} is under $\mathcal{A}$'s control. The adversary's goal is to forge $\ell + 1$ signature-message pairs $(\sigma_1, \mu_1), \ldots, (\sigma_{\ell+1}, \mu_{\ell+1})$ for distinct messages, while completing at most ℓ signing sessions. The idea behind $\mathsf{BS}_{\mathsf{os}}$ is that the underlying Σ-protocol is witness-indistinguishable, that is, the transcripts τ_{os} do not reveal any information about the secret key $\boldsymbol{w}$. Indeed, this property is leveraged to show that the scheme is secure for at most $\mathsf{polylog}$-many signing sessions [PS00, HKL19]. Roughly, the reduction embeds a DL challenge H into the parameters and rewinds the adversary $\mathcal{A}$ on a hash query $c_i = \mathsf{H}(W, A_i, \mu_i)$ associated to one of $\mathcal{A}$'s signatures $\sigma_i = (A_i, c_i, z_i)$ for message μ_i. From the second run, the adversary obtains a signature $\sigma_i' = (A_i, c_i', z_i')$ with fresh challenge $c_i' \neq c_i$. In particular, the signatures σ_i and σ_i' form related Σ-protocol transcripts, therefore the reduction can recover some witness $\boldsymbol{w}' = (w_0', w_1')$ such that $\phi_{\mathsf{os}}(\boldsymbol{w}') = W$ by 2-special soundness of the Σ-protocol. If at most $\mathsf{polylog}$-many signing sessions are completed, then [PS00, HKL19] show that $\boldsymbol{w}' \neq \boldsymbol{w}$ with sufficient probability. If so, we have $\phi_{\mathsf{os}}(\boldsymbol{w}' - \boldsymbol{w}) = 0_{\mathbb{G}}$ with $\boldsymbol{w}' - \boldsymbol{w} \neq 0$ by linearity of ϕ_{os}. In that case, the reduction can compute the discrete logarithm of H, breaking DL.

Note that it is tempting to argue that "as the adversary learns nothing about the witness $\boldsymbol{w}$ from the initial run, the extracted witness $\boldsymbol{w}'$ must be fresh in general." However, this argument is flawed as $\boldsymbol{w}'$ may depend on the distribution of the honest transcripts τ_{os} during the initial *and* second run, which are issued during signing via the secret key $\boldsymbol{w}$. Further, the transcripts from initial and second runs together reveal $\boldsymbol{w}$ in some scenarios with concurrent sessions. For example, if a session with nonce A_{os} is opened but not closed before the adversary is rewound, then the reduction may provide two valid transcripts for nonce A_{os}. These related transcripts lead to an extraction of the original secret key $\boldsymbol{w}$. If all of $\mathcal{A}$'s signatures are somehow built from the issued transcripts, then there is no way to argue that the extracted witness $\boldsymbol{w}'$ is different from $\boldsymbol{w}$.

We note that prior attempts [AO00, Abe01, AHJ21] to formalize the above intuition resulted in errors. At this point, there is no known proof technique to argue OMUF via witness-indistinguishability and rewinding beyond $\mathsf{polylog}$-many concurrent sessions. Indeed, there is no hope to formally show such a result for $\mathsf{BS}_{\mathsf{os}}$ due to the attack on ROS [BLL+21, BLL+22].

Our idea. We resolve the apparent issue sketched above by enforcing that the extracted witness is determined in the initial run *statistically* from the adversary's signatures. That is, we base our blind signature on $\mathsf{BS}_{\mathsf{os}}$, but we introduce additional measures such that $\mathcal{A}$'s signatures $(\sigma_1, \mu_1), \ldots, (\sigma_{\ell+1}, \mu_{\ell+1})$ determine the extracted witness $\boldsymbol{w}'$ information-theoretically *before* any rewinding occurs. As the initial run hides the secret key $\boldsymbol{w}$ perfectly, the witness $\boldsymbol{w}'$ extracted via rewinding later on must be distributed independently

from $\boldsymbol{w}$; in particular $\boldsymbol{w}' \neq \boldsymbol{w}$ with sufficient probability as desired. Both $\boldsymbol{w}$ and $\boldsymbol{w}'$ together allow us to compute the discrete logarithm of H as in the original analysis in [PS00], except there is no bound on the number of concurrent sessions.

OUR DESIGN. Let us ignore blindness at this step for exposition. In short, we tag the Okamoto-Schnorr nonce A_{os} with an algebraic MAC σ_{mac} and let the signer issue a Σ-protocol transcript $\tau_{\mathsf{mac}} = (A_{\mathsf{mac}}, c_{\mathsf{mac}}, z_{\mathsf{mac}})$ that certifies that σ_{mac} is indeed a valid tag for A_{os}. The transcript τ_{mac} is compiled into a proof via Fiat-Shamir by the user. The signature then consists of $\sigma = (\tau_{\mathsf{os}}, \tau_{\mathsf{mac}}, \sigma_{\mathsf{mac}})$. Let us inspect the adversary's signatures in the reduction under this modification:

Case 1 (Nonce collision). If there are two distinct signatures with identical nonces $A_{\mathsf{os},i}$ and $A_{\mathsf{os},j}$ amongst $\mathcal{A}$'s output, then the reduction sets $\boldsymbol{w}' = (z_{\mathsf{os},i} - z_{\mathsf{os},j}) / (c_{\mathsf{os},i} - c_{\mathsf{os},j})$ and computes the discrete logarithm of H via $\boldsymbol{w} - \boldsymbol{w}'$. Note that $\boldsymbol{w}$ is the secret key that the reduction samples to simulate the game, chosen such that $\phi_{\mathsf{os}}(\boldsymbol{w}) = W$. To see why this works, observe that by linearity of ϕ_{os}, the verification equations give

$$\phi_{\mathsf{os}}(z_{\mathsf{os},i}) - c_{\mathsf{os},i} \cdot W = \phi_{\mathsf{os}}(z_{\mathsf{os},j}) - c_{\mathsf{os},j} \cdot W \qquad \Longrightarrow \qquad \phi_{\mathsf{os}}(\boldsymbol{w}') = W. \tag{1}$$

Note that $c_{\mathsf{os},i} \neq c_{\mathsf{os},j}$ except for negligible probability as $\mu_i \neq \mu_j$. Also, it must hold that $\boldsymbol{w} \neq \boldsymbol{w}'$ with sufficient probability as the adversary's view $\mathcal{V}_{\mathcal{A}}$ is independent of the secret key $\boldsymbol{w}$ chosen by the reduction, however, the extracted $\boldsymbol{w}'$ is determined entirely by $\mathcal{V}_{\mathcal{A}}$. We formalize this argument with the technique introduced by [PS00]: we exhibit a map Ψ which preserves the adversary's view, and therefore the witness $\boldsymbol{w}'$ determined by $\mathcal{V}_{\mathcal{A}}$, yet the witness $\boldsymbol{w}$ used for simulation is mapped to a different value. In summary, we know that $\phi_{\mathsf{os}}(\boldsymbol{w}) = W = \phi_{\mathsf{os}}(\boldsymbol{w}')$ (as the view is preserved) with $\boldsymbol{w} \neq \boldsymbol{w}'$. Therefore, we can compute the DLOG of H as before.

Case 2 (Distinct nonces). If all $\ell+1$ forgeries σ_i have distinct nonces $A_{\mathsf{os},i}$, we can produce a MAC forgery. That is, observe that the proof $\tau_{\mathsf{mac},i}$ from σ_i ensures that the tags $\sigma_{\mathsf{mac},i}$ from the forgeries are valid. As only ℓ-many tags were issued and the Σ-protocol proof ensures that these tags are valid, this breaks unforgeability of the underlying MAC. We note that to simulate the transcripts τ_{mac} during signing, we require an additional simulation trapdoor for the Σ-protocol. Looking ahead, we will implement with an OR-proof.

Note that at this point, no rewinding is required to extract the ϕ_{os}-preimage $\boldsymbol{w}'$ which simplifies the analysis. However, the protocol is not yet blind and looking ahead, rewinding will become necessary after we ensure blindness with additional measures.

BLINDING THE SIGNATURES. We must ensure that an issued signature $\sigma = (\tau_{\mathsf{os}}, \tau_{\mathsf{mac}}, \sigma_{\mathsf{mac}})$ derived by the user is hard to be linked to its signing session. The first step is that the user blinds the proofs τ_{mac} and τ_{os} into $\widetilde{\tau}_{\mathsf{mac}}$ and $\widetilde{\tau}_{\mathsf{os}}$, respectively. Unfortunately, the tag σ_{mac} remains linkable to the signing session. Even if we somehow randomize σ_{mac}, the tag still verifies for the nonce A_{os} from the signing session, leading to trivial attacks on blindness. Furthermore, if we blind A_{os} into $\widetilde{A}_{\mathsf{os}}$, the tag σ_{mac} tags the original nonce A_{os}, not $\widetilde{A}_{\mathsf{os}}$ from the signature as desired. Our solution is as follows.

Blind the MAC. We blind σ_{mac} to $\widetilde{\sigma}_{\mathsf{mac}}$ with blinding factor r_{rnd} such that $\widetilde{\sigma}_{\mathsf{mac}}$ remains a valid tag for the original nonce A_{os} but cannot be linked back to σ_{mac}. Furthermore, given r_{rnd}, it is possible to recompute σ_{mac}.

Commit to blinding factors. The user commits to the tag blinding factor r_{rnd} and the original nonce A_{os}, together with the nonce's blinding factors $\boldsymbol{\alpha}$ and γ, in a commitment $\mathsf{cmt}_{\mathsf{rnd}}$.

Prove honest blinding. The user proves that the blinded nonce $\widetilde{A}_{\mathsf{os}}$ and tag $\widetilde{\sigma}_{\mathsf{mac}}$ satisfy the following properties with respect to the values committed in $\mathsf{cmt}_{\mathsf{rnd}}$. Note that we assume that $\mathsf{cmt}_{\mathsf{rnd}}$ is perfectly binding to ensure that these relations hold *without* rewinding (cf. discussion below):

1. the blinded tag $\widetilde{\sigma}_{\mathsf{mac}}$ is obtained by blinding σ_{mac} via r_{rnd};
2. the blinded nonce $\widetilde{A}_{\mathsf{os}}$ is obtained by blinding A_{os} via $(\boldsymbol{\alpha}, \gamma)$;
3. the original tag σ_{mac} is valid for the original nonce A_{os}.

The first two points are proven with a Fiat-Shamir compiled proof $\tau_{\mathsf{rnd}} = (\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ computed by the user. Note that the user cannot prove the final point as it cannot verify σ_{mac} itself. To ensure the validity of the original tag σ_{mac}, the user adapts the proof τ_{mac} accordingly, leveraging linear properties of the transcript τ_{mac} and the fact that the blinding σ_{mac} into $\widetilde{\sigma}_{\mathsf{mac}}$ is linear, similar to [CATZ24, OTZZ24, KRW24, KR25].

As alluded to above, we further require a trapdoor for the τ_{mac} proof for simulation, which we implement by a DDH-trapdoor. That is, the public parameters contain a tuple $\boldsymbol{D}$. The proof τ_{mac} is implemented as OR-proof: either it ensures that the MAC is valid or that $\boldsymbol{D}$ is a DDH-tuple, similar to [KRW24]. This gives us the freedom to issue simulated tags in the security analysis, while being able to simulate well-distributed proofs.

We further ensure that $\mathsf{cmt}_{\mathsf{rnd}}$ is extractable (i.e., there is a trapdoor with which the *unique* opening for $\mathsf{cmt}_{\mathsf{rnd}}$ can be recovered). This is important to argue about the distribution of the committed values in $\mathsf{cmt}_{\mathsf{rnd}}$ in the initial run (i.e., before rewinding). As a consequence, we can only commit to group elements directly, therefore we commit to the blinding factors in $\mathbb{Z}_p$ in the exponent.

PUTTING EVERYTHING TOGETHER. In summary, the public parameters contain the generators (G, H) for Okamoto-Schnorr, a tuple $\boldsymbol{D}$ and a commitment key bck for a commitment $\mathsf{C}_{\mathsf{rnd}}$. The public key is $W = \phi_{\mathsf{os}}(\boldsymbol{w})$ for $\boldsymbol{w} \leftarrow\!\!\$\, \mathbb{Z}_p^2$, whereas the secret key is $\boldsymbol{w}$ and the signing key msk for the MAC. The signer-user interaction issues a Σ-protocol transcript $\tau_{\mathsf{os}} = (A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ for knowledge of $\boldsymbol{w}$ to the user, and further a tag σ_{mac} on A_{os} and an OR-proof $\tau_{\mathsf{or}} = (A_{\mathsf{or}}, c_{\mathsf{or}}, z_{\mathsf{or}})$ that either σ_{mac} is valid or $\boldsymbol{D}$ is a DDH-tuple. The user blinds the transcripts τ_{os} into $\widetilde{\tau}_{\mathsf{os}}$, and further blinds τ_{or} together with the tag σ_{mac} into $\widetilde{\tau}_{\mathsf{or}}$ and $\widetilde{\sigma}_{\mathsf{mac}}$, respectively, as described above. At the same time, the user compiles the transcripts $\widetilde{\tau}_{\mathsf{os}}$ and $\widetilde{\tau}_{\mathsf{or}}$ via Fiat-Shamir. Finally, the user commits to the blinding factors in $\mathsf{cmt}_{\mathsf{rnd}}$ and computes a Fiat-Shamir proof $\widetilde{\tau}_{\mathsf{rnd}}$ for honest blinding as described above. The derived signature is $\sigma \leftarrow (\widetilde{\sigma}_{\mathsf{mac}}, \mathsf{cmt}_{\mathsf{rnd}}, \widetilde{\tau}_{\mathsf{os}}, \widetilde{\tau}_{\mathsf{or}}, \widetilde{\tau}_{\mathsf{rnd}})$ For a detailed description of our construction and its building blocks, we refer to Section 3.

Blindness follows from standard arguments. Notably, if $\mathsf{cmt}_{\mathsf{rnd}}$ is statistically hiding, then blindness also holds statistically. We outline the OMUF proof below.

PROVING ONE-MORE UNFORGEABILITY. OMUF still relies on the high-level intuition sketched above, however, the proof becomes much more technical due to the additional blinding factors. In particular, we assume that $\mathsf{cmt}_{\mathsf{rnd}}$ is setup such that the game can extract the messages. We stress that A_{os} can be extracted in clear, however, the blinding factors can only be recovered in the exponent. In the overview below, we assume that $A_{\mathsf{os},i}$ has been extracted from all of $\mathcal{A}$'s signature outputs σ_i for $i \in [\ell + 1]$. Let us consider the two cases outlined above in more detail.

Case 1 (Nonce collision). If there are two distinct signatures with identical *extracted* nonces $A_{\mathsf{os},i} = A_{\mathsf{os},j}$ amongst $\mathcal{A}$'s outputs, then the reduction rewinds the adversary to extract some ϕ_{os}-preimage $\boldsymbol{w}'$ for W from the transcripts. Then, the reduction computes the DLOG of H via $\boldsymbol{w}$ and $\boldsymbol{w}'$, where $\boldsymbol{w}$ is the secret key chosen by the reduction to simulate the game.
That is, observe that the Okamoto-Schnorr blinding factors $(\boldsymbol{\alpha}_i, \gamma_i)$ and $(\boldsymbol{\alpha}_j, \gamma_j)$ are determined by $\mathsf{cmt}_{\mathsf{rnd},i}$ and $\mathsf{cmt}_{\mathsf{rnd},j}$ in the initial run, respectively, in the exponent. Also, $A_{\mathsf{os},i} = A_{\mathsf{os},j}$ is known before any rewinding occurs. Therefore, rewinding is guaranteed to recover exactly these blinding factors. Given the blinding factors and $A_{\mathsf{os},i}$, the reduction computes the desired preimage $\boldsymbol{w}'$ similar to Equation (1), taking into account the additional blinding factors. We stress again that the extracted witness is already determined through the first run, even though there may be a correlation between the initial and rewound transcripts. This allows us to argue that $\boldsymbol{w}' \neq \boldsymbol{w}$, and thus, the reduction can compute the DLOG of H as before.

Case 2 (Distinct nonces). If all extracted Okamoto-Schnorr nonces are distinct, then there must be a fresh nonce $A_{\mathsf{os},i}$ that the game did not tag via the MAC scheme. Note that the MAC $\widetilde{\sigma}_{\mathsf{mac},i}$ is blinded and we must first recover $r_{\mathsf{rnd},i}$ to recompute the tag σ_{mac} on $A_{\mathsf{os},i}$. This is possible via rewinding which allows us to reduce to unforgeability of the MAC scheme. There is an important subtlety in the proof: the reduction may tag the nonce $A_{\mathsf{os},i}$ during rewinding. In this case, we cannot argue that $\sigma_{\mathsf{mac},i}$ constitutes a forgery. However, notice that the reduction exclusively tags the nonces A_{os} during the simulation of

S_1. In particular, these nonces are in control of the reduction and can be sampled *before* any rewinding occurs. This allows us to argue that $A_{\mathsf{os},i}$ is tagged neither in the initial run, nor during rewinding.

An interesting observation is that since the reduction only issues MAC tags on random nonces, it suffices if the MAC is secure if the adversary sees tags on random messages.

INSTANTIATIONS. To instantiate the above template, we require an appropriate MAC and commitment. For the latter, we either choose a multi-recipient variant of ElGamal (for computational blindness and compact signatures) or Groth-Sahai commitments (for statistical blindness but larger signatures). For the former, we note that the MAC must be algebraic and blinding-friendly (i.e., there is a Σ-protocol for the validity of tags that can be blinded together with the tag). We employ the DDH-based MAC $\mathsf{M}_{\mathsf{alg}}$ by Chase et al. [CMZ14]. Note that $\mathsf{M}_{\mathsf{alg}}$ only allows tagging $\mathbb{Z}_p$ elements, but this is sufficient for our purpose due to the following insight. A valid tag on message $m \in \mathbb{Z}_p$ has the form

$$U = uG\ , \qquad V = zU\ ,$$
$$S = u(x_0G + x_1mG)\ , \qquad T = u(y_0G + y_1mG)\ ,$$

where (z, x_0, x_1, y_0, y_1) are part of the secret key and $u \leftarrow\!\!\$\ \mathbb{Z}_p^*$ is the randomness. Instead, we simply compute tags on group elements $M \in \mathbb{G}$ via

$$U = uG\ , \qquad V = zU\ ,$$
$$S = u(x_0G + x_1\lceil\bar{M}\rfloor)\ , \qquad T = u(y_0G + y_1\lceil\bar{M}\rfloor)\ .$$

If there is a collision in the nonces $A_{\mathsf{os},i}$, then we do not require the security of the MAC, in particular, the structure of $\mathsf{M}_{\mathsf{alg}}$ is irrelevant. Otherwise, the reduction can know the discrete logarithm h of H. In particular, the nonces A_{os} can be expressed as $A_{\mathsf{os}} = r_0G + r_1H = (r_0 + hr_1)G$, so the reduction can instead issue a tag for message $m = r_0 + hr_1$. Furthermore, the fresh nonce $A_{\mathsf{os},i}$ extracted from $\mathcal{A}$'s outputs in the reduction can be expressed with known discrete logarithm through knowledge of the blinding factors and the verification equation of Σ_{os}. This allows us to reduce to MAC security with $\mathbb{Z}_p$-messages.

EXTENSIONS. While we sketched the security proof for OMUF, it is possible to prove that the scheme satisfies one-more *strong* unforgeability (OMSUF). This mainly relies on the (computational) unique response property of the Σ-protocols. Furthermore, we can prove partial blindness if the DDH-trapdoor $\boldsymbol{D}$ is sampled with a random oracle H_{pb} evaluated on the common message info. This adds additional complexity in the OMSUF proof, since we must issue additional tags on Okamoto-Schnorr nonces in sessions for distinct common messages. We refer to Section 3.2 for more details.

1.2 Related and Future Work

This paper shows that it is possible to construct three-move blind signatures in pairing-free groups in the ROM alone, resolving a major open problem in the field. We still rely on the DDH assumption for our security analysis; the "dream result" in the field is a three-move blind signature from DL. The recent work [KLR25] gives a complementary result to ours by constructing a four-move blind signature from DL. It remains unclear how to reconcile our techniques and the techniques from [KR25] into a three-move blind signature from DL. We briefly discuss the challenges and leave the problem for future work.

- Our construction relies on an algebraic MAC and a linearly-homomorphic extractable commitment. Both primitives are well-known to exist from DDH. It is unclear how to construct these from DL alone in a black-box manner.
- In the scheme by Klooß et al. [KLR25], signatures consist of a Fiat-Shamir proof for a message-dependent statement. The statement can be punctured for some message m^*. That is, the reduction can produce a proof (and therefore simulate signing) for any statement associated to some $m \neq m^*$, while a proof for statement m^* allows one to solve DL with a rewinding-based argument. However, as the statement for the to-be-issued proof depends on the message, the user must embed the message into the signing session in the first move. The three additional moves then establish the Fiat-Shamir proof. It is unclear how to employ our techniques to compress their scheme down to three moves.

CONCURRENT WORK. A concurrent work by Chen [Che26] achieved the same result as ours by constructing a fully black-box pairing-free three-move blind signature scheme that is secure in the ROM from DDH assumptions. We note that our constructions are more efficient and we also achieve additional properties, such as strong unforgeability and statistical blindness. Their core idea is quite similar to ours: instead of authenticating the Okamoto-Schnorr nonces with an algebraic MAC, they rely on one-time signatures and a new primitive, *encrypted message blind signatures* (EMBS), to authenticate the one-time signature public key.

2 Preliminaries

NOTATION. Let $[n..m] = \{n, n+1, \ldots, m\}$ for any $n, m \in \mathbb{Z}$ where $n \leqslant m$ and $[n] = [1..n]$ for any $n \geqslant 1$. We use λ as the security parameter.

PAIRING-FREE GROUPS. Let GGen be an algorithm that on input 1^λ outputs a group description $\mathbb{G}$ of λ-bit prime-order $p = p(\lambda)$ with generator G. We employ additive notation. Throughout, we assume that algorithms obtain as input $(\mathbb{G}, p, G) \leftarrow\!\!\$\ \mathsf{GGen}(1^\lambda)$ as parameter implicitly. Also, we denote by $\mathsf{dlog}_G X$ the discrete logarithm $x \in \mathbb{Z}_p$ of X with basis G, where $xG = X$. We extend the notation to vectors of group elements $(X_1, \ldots, X_n)$ in the natural manner.

SECURITY ASSUMPTIONS. We recall the DL and DDH assumptions. Let $(\mathbb{G}, p, G)$ as above. We note that the assumptions depend on GGen employed to (implicitly) sample $(\mathbb{G}, p, G)$.

Definition 2.1 (DL Assumption). *The discrete logarithm (*DL*) assumption holds if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{dl}}_{\mathsf{GGen}}(\mathcal{A}, \lambda) := \mathsf{Pr}[x = x' \mid x \leftarrow\!\!\$\ \mathbb{Z}_p, x' \leftarrow \mathcal{A}(G, xG)] = \mathsf{negl}(\lambda)\ .$$

Definition 2.2 (QDDH Assumption). *The Q-fold (for $Q = Q(\lambda)$) decisional Diffie-Hellman (DDH) assumption holds if for any PPT adversary $\mathcal{A}$, it holds that the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q}(\mathcal{A}, \lambda) := \Big|\mathsf{Pr}[a \leftarrow\!\!\$\ \mathbb{Z}_p, \boldsymbol{b} \leftarrow\!\!\$\ \mathbb{Z}_p^Q : \mathcal{A}(G, aG, (b_iG, (ab_i)G)_{i\in[Q]}) = 1] -$$
$$\mathsf{Pr}[a \leftarrow\!\!\$\ \mathbb{Z}_p, \boldsymbol{b}, \boldsymbol{c} \leftarrow\!\!\$\ \mathbb{Z}_p^Q : \mathcal{A}(G, aG, (b_iG, c_iG)_{i\in[Q]}) = 1]\Big|\ .$$

We refer to 1-fold DDH as the DDH assumption.

The DDH assumption tightly implies the QDDH assumption. Concretely, for any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ with roughly the same running time such that $\mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q}(\mathcal{A}, \lambda) \leqslant \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}, \lambda) + \frac{1}{p-1}$ [EHK+13].

BLIND SIGNATURES. Now, we define blind signatures [Cha82] with security definitions from [JLO97]. We consider 3-move schemes with partial blindness [AF96]. A partially blind signature BS is a tuple of PPT algorithms $(\mathsf{BS.Setup}, \mathsf{BS.KeyGen}, (\mathsf{BS.Sign}_1, \mathsf{BS.U}_1, \mathsf{BS.Sign}_2, \mathsf{BS.U}_2), \mathsf{BS.Ver})$ such that

$\mathsf{BS.Setup}(1^\lambda) \to \mathsf{pp}$**:** Given security parameter 1^λ, outputs public parameters pp.
$\mathsf{BS.KeyGen}(\mathsf{pp}) \to (\mathsf{sk}, \mathsf{pk})$**:** Given public parameters pp, outputs a secret key sk and public key pk.
$\mathsf{BS.Sign}_1(\mathsf{pp}, \mathsf{sk}, \mathsf{info}) \to (\mathsf{smsg}_1, \mathsf{st}_\mathsf{S})$**:** Given public parameters pp, a secret key sk and a common message info, outputs a signer message smsg_1 and signer state st_S.
$\mathsf{BS.U}_1(\mathsf{pp}, \mathsf{pk}, \mathsf{info}, \mu, \mathsf{smsg}_1) \to (\mathsf{umsg}, \mathsf{st}_\mathsf{U})$**:** Given public parameters pp, a public key pk, a common message info, a message μ and a signer message smsg_1, outputs a user message umsg and user state st_U.
$\mathsf{BS.Sign}_2(\mathsf{umsg}, \mathsf{st}_\mathsf{S}) \to \mathsf{smsg}_2$**:** Given public parameters pp, a secret key sk, a user message umsg and signer state st_S, outputs a signer message smsg_2.
$\mathsf{BS.U}_2(\mathsf{smsg}_2, \mathsf{st}_\mathsf{U}) \to \sigma$**:** Given signer message smsg_2 and user state st_U, outputs a signature σ.
$\mathsf{BS.Ver}(\mathsf{pp}, \mathsf{pk}, \mathsf{info}, \mu, \sigma) \to b$**:** Given public parameters pp, a public key pk, a common message info, a message μ and a signature σ, deterministically outputs $b \in \{0, 1\}$.

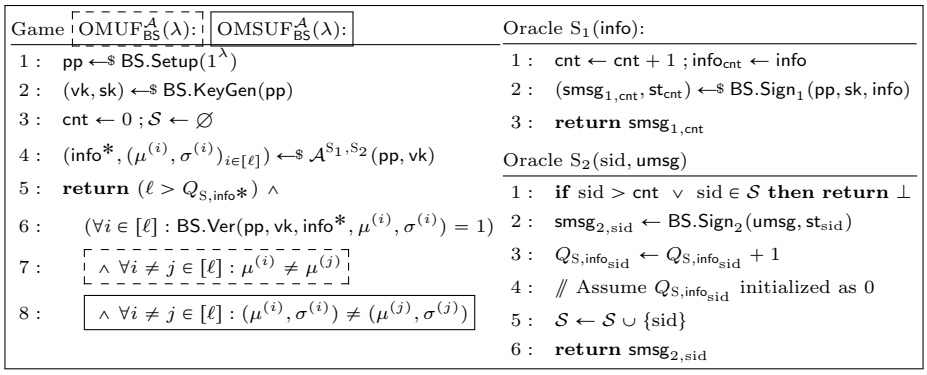

Fig. 1. One-More (Strong) Unforgeability security game of BS. The dashed box indicates the winning condition of the OMUF game, while the solid box indicates the winning condition of the OMSUF game.

If clear by context, we leave the user and signer states implicit for conciseness. We define correctness, partial blindness and one-more strong unforgeability below.

Definition 2.3 (Correctness). *A partially blind signature* BS *is (perfectly) correct if for all public parameters* $\mathsf{pp} \in \mathsf{BS.Setup}(1^\lambda)$, $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{BS.KeyGen}(\mathsf{pp})$, *all messages* μ *and all common messages* info, *it holds that* $\mathsf{BS.Ver}(\mathsf{pp}, \mathsf{pk}, \mathsf{info}, \mu, \sigma) = 1$, *where*

$$\begin{aligned}(\mathsf{smsg}_1, \mathsf{st}_\mathsf{S}) &\in \mathsf{BS.Sign}_1(\mathsf{pp}, \mathsf{sk}, \mathsf{info})\\ (\mathsf{umsg}, \mathsf{st}_\mathsf{U}) &\in \mathsf{BS.U}_1(\mathsf{pp}, \mathsf{pk}, \mathsf{info}, \mu, \mathsf{smsg}_1)\\ \mathsf{smsg}_2 &\in \mathsf{BS.Sign}_2(\mathsf{umsg}, \mathsf{st}_\mathsf{S})\\ \sigma &\in \mathsf{BS.U}_2(\mathsf{smsg}_2, \mathsf{st}_\mathsf{U})\end{aligned}$$

One-more strong unforgeability ensures that it is hard to compute a fresh signature-message pair without interacting with the signer.

Definition 2.4 (One-More (Strong) Unforgeability). *A partially blind signature scheme* BS *is one-more (strong) unforgeable if for any PPT* $\mathcal{A}$ *in the game* OMUF *(or* OMSUF*) as defined in Figure 1, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathrm{omuf/omsuf}}_{\mathsf{BS}}(\mathcal{A}, \lambda) := \mathsf{Pr}[(\mathrm{OMUF/OMSUF})^{\mathcal{A}}_{\mathsf{BS}}(\lambda) = 1] .$$

The following lemma also allows us to assume that any OM(S)UF adversary finishes all signing sessions. Note that similar assumption has been (implicitly) made in prior works on three-move blind signatures [FPS20, TZ22]. We give a proof of the lemma in Appendix A. Moreover, the lemma extends naturally for adversaries in the ROM (with increased random oracle query count).

Lemma 2.5. *Let* $\mathcal{A}$ *be an adversary against OMUF (resp. OMSUF) game of* BS, *running in time* $T_\mathcal{A} = T_\mathcal{A}(\lambda)$ *and making* $Q_{\mathrm{S}_1} = Q_{\mathrm{S}_1}(\lambda), Q_{\mathrm{S}_2} = Q_{\mathrm{S}_2}(\lambda)$ *to* $\mathrm{S}_1, \mathrm{S}_2$, *respectively. Then, there exists an adversary* $\mathcal{B}$ *against OMUF (resp. OMSUF) game of* BS, *running in time* $T_\mathcal{B} \approx T_\mathcal{A}$ *and making* $Q_\mathrm{S} = Q_{\mathrm{S}_1}$ *queries to both* $\mathrm{S}_1, \mathrm{S}_2$, *such that*

$$\mathsf{Adv}^{\mathrm{omuf}}_{\mathsf{BS}}(\mathcal{A}, \lambda) \leqslant \mathsf{Adv}^{\mathrm{omuf}}_{\mathsf{BS}}(\mathcal{B}, \lambda) , \qquad \mathsf{Adv}^{\mathrm{omsuf}}_{\mathsf{BS}}(\mathcal{A}, \lambda) \leqslant \mathsf{Adv}^{\mathrm{omsuf}}_{\mathsf{BS}}(\mathcal{B}, \lambda) .$$

Partial blindness ensures that the signer cannot link signing interactions to the issued signatures, even if the signing key is malicious.

Definition 2.6 (Partial Blindness). *We say that* BS *is partially blind if for any adversary* $\mathcal{A}$ *in the game* Blind *as defined in Figure 2, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{blind}}_{\mathsf{BS}}(\mathcal{A}, \lambda) := \left|\mathsf{Pr}[\mathrm{Blind}^{\mathcal{A}}_{\mathsf{BS}}(\lambda) = 1] - \frac{1}{2}\right| .$$

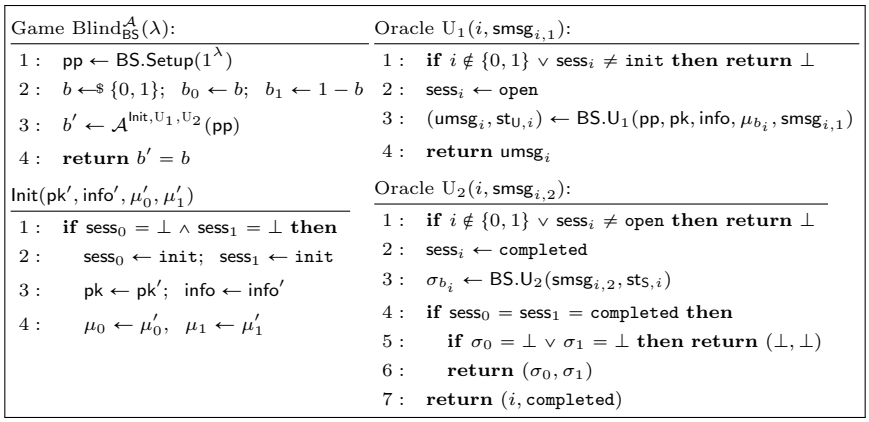

Fig. 2. Blindness security game of BS.

COMMITMENT. A commitment scheme C with message space $\mathsf{C}.\mathcal{M}$, key space $\mathsf{C}.\mathcal{K}$, commitment space $\mathsf{C}.\mathcal{C}$ and randomness space $\mathsf{C}.\mathcal{R}$ is a tuple of PPT algorithms $(\mathsf{C.Setup}, \mathsf{C.Commit})$ such that

$\mathsf{C.Setup}(1^\lambda) \to \mathsf{ck}$**:** Given security parameter 1^λ, outputs commitment key ck.
$\mathsf{C.Commit}(\mathsf{ck}, m; r) \to \mathrm{cmt}$**:** Given the commitment key ck, message $m \in \mathsf{C}.\mathcal{M}$ and explicit randomness $r \in \mathsf{C}.\mathcal{R}$, it outputs a commitment c to m.

We do not define an explicit verification algorithm, instead we verify whether a commitment cmt opens to message m with randomness r via $\mathrm{cmt} = \mathsf{C.Commit}(\mathsf{ck}, m; r)$. We say that C is linearly-homomorphic if $\mathsf{C}.\mathcal{M}$, $\mathsf{C}.\mathcal{C}$ and $\mathsf{C}.\mathcal{R}$ are $\mathbb{Z}_p$-vector spaces and if for each commitment key $\mathsf{ck} \in \mathsf{C}.\mathcal{K}$, the commitment function $\mathsf{C.C}(\mathsf{ck}, \cdot\,; \cdot) : (m, r) \mapsto \mathrm{cmt}$ is $\mathbb{Z}_p$-linear. Below, we recall desirable security properties of commitment schemes.

Definition 2.7 (Hiding). *A commitment scheme* C *is computationally hiding if for any PPT adversary* $\mathcal{A}$*, we have that the following advantage is negligible*

$$\mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}}(\mathcal{A}, \lambda) = \left| \mathsf{Pr}\left[b = b' \,\middle|\, \begin{array}{l} \mathsf{ck} \leftarrow\!\!\$\ \mathsf{C.Setup}(1^\lambda),\ \ (m_0, m_1, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}(\mathsf{pp}), \\ b \leftarrow\!\!\$\ \{0,1\},\ \ r \leftarrow\!\!\$\ \mathsf{C}.\mathcal{R},\ \ b' \leftarrow \mathcal{A}(c, \mathsf{st}_{\mathcal{A}}) \end{array} \right] - \frac{1}{2} \right| ,$$

where $c \leftarrow \mathsf{C.Commit}(\mathsf{ck}, m_b; r)$*. We say that* C *is statistically hiding if the above holds for* **all** *adversaries* $\mathcal{A}$*.*

Definition 2.8 ((Strong) Binding). *A commitment scheme* C *is computationally binding if for any PPT adversary* $\mathcal{A}$*, we have*

$$\mathsf{Adv}^{\mathsf{bind}}_{\mathsf{C}}(\mathcal{A}, \lambda) := \mathsf{Pr}\left[m_0 \neq m_1 \wedge c_0 = c_1 \,\middle|\, \begin{array}{l} \mathsf{ck} \leftarrow \mathsf{C.Setup}(1^\lambda), \\ (m_0, m_1, r_0, r_1) \leftarrow \mathcal{A}(\mathsf{ck}) \end{array} \right] = \mathsf{negl}(\lambda),$$

where $c_i = \mathsf{C.Commit}(\mathsf{ck}, m_i; r_i)$ *for* $i \in \{0,1\}$*. We say that* C *is computationally "strongly" binding if the winning condition above is* $(m_0 \neq m_1 \vee r_0 \neq r_1) \wedge c_0 = c_1$*, and we denote the advantage with* $\mathsf{Adv}^{\mathsf{sbind}}_{\mathsf{C}}(\mathcal{A}, \lambda)$ *instead. We say the scheme is perfectly (strongly) binding if the advantage term above is* 0 *for all potentially inefficient adversaries* $\mathcal{A}$*.*

Definition 2.9 (Extractability). *A commitment scheme* C *is extractable if there exist PPT algorithms* $\mathsf{Ext} = (\mathsf{C.ExtSetup}, \mathsf{C.Extract})$ *defined as follows*

$\mathsf{C.ExtSetup}(1^\lambda) \to (\mathsf{ck}, \mathsf{td})$ *: Generates a commitment key* ck *and a trapdoor* td*, and outputs* $(\mathsf{ck}, \mathsf{td})$*.*
$\mathsf{C.Extract}(\mathsf{td}, c) \to m$ *: On input of a trapdoor* td *and a commitment* $c \in \mathsf{C}.\mathcal{C}$*, deterministically outputs the committed message* $m \in \mathsf{C}.M$*.*

Game $\text{UF-RMA}^{\mathcal{A}}_{\mathsf{M}}(\lambda)$:	Oracle Tag():
1 : $\mathcal{Q} \leftarrow \varnothing$	1 : $m \leftarrow\!\!\$\ \mathsf{M}.\mathcal{M}$
2 : $\mathsf{mpp} \leftarrow\!\!\$\ \mathsf{M.Setup}(1^\lambda)$; $(\mathsf{msk}, \mathsf{ipp}) \leftarrow\!\!\$\ \mathsf{M.KeyGen}(\mathsf{mpp})$	2 : $\sigma_{\mathsf{alg}} \leftarrow\!\!\$\ \mathsf{M.Tag}(\mathsf{mpp}, \mathsf{msk}, m)$
3 : $(m^*, \sigma^*_{\mathsf{alg}}) \leftarrow\!\!\$\ \mathcal{A}^{\text{Tag}}(\mathsf{mpp}, \mathsf{ipp})$	3 : $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
4 : **return** $m^* \notin \mathcal{Q} \ \wedge\ \mathsf{M.Ver}(\mathsf{mpp}, \mathsf{ipp}, \mathsf{msk}, m^*, \sigma^*_{\mathsf{alg}}) = 1$	4 : **return** $(m, \sigma_{\mathsf{alg}})$

Fig. 3. UF-RMA security game of M.

We require that:

(a) For any PPT adversary $\mathcal{A}$, the advantage

$$\mathsf{Adv}^{\mathsf{ext\text{-}dist}}_{\mathsf{C},\mathsf{Ext}}(\mathcal{A}, \lambda) := |\mathsf{Pr}[\mathcal{A}(\mathsf{ck}_{\mathsf{real}}) = 1] - \mathsf{Pr}[\mathcal{A}(\mathsf{ck}_{\mathsf{sim}}) = 1]| \ ,$$

is negligible, where $\mathsf{ck}_{\mathsf{real}} \leftarrow\!\!\$\ \mathsf{C.Setup}(1^\lambda)$ and $(\mathsf{ck}_{\mathsf{sim}}, \mathsf{td}) \leftarrow\!\!\$\ \mathsf{C.ExtSetup}(1^\lambda)$.

(b) For all $(\mathsf{ck}, \mathsf{td}) \leftarrow\!\!\$\ \mathsf{C.ExtSetup}(1^\lambda), m \in \mathsf{C}.\mathcal{M}, r \in \mathsf{C}.\mathcal{R}$ and $c \leftarrow \mathsf{C.Commit}(\mathsf{ck}, m; r) \in \mathsf{C}.\mathcal{C}$, it holds that $m = \mathsf{C.Extract}(\mathsf{ck}, \mathsf{td}, c)$.

MESSAGE AUTHENTICATION CODES. We recall syntax of a Message Authentication Code (MAC) and recall the unforgeability against random message attack security (UF-RMA). A MAC M consists of (probabilistic) algorithms $(\mathsf{M.Setup}, \mathsf{M.KeyGen}, \mathsf{M.Tag}, \mathsf{M.Ver})$ which have the following syntax:

$\mathsf{M.Setup}(1^\lambda) \to \mathsf{mpp}$**:** Generates the public parameters mpp which implicitly defines the message space $\mathsf{M}.\mathcal{M}$.
$\mathsf{M.KeyGen}(\mathsf{mpp}) \to (\mathsf{msk}, \mathsf{ipp})$**:** Generates the issuer's secret key msk and its corresponding issuer's public parameters ipp.
$\mathsf{M.Tag}(\mathsf{mpp}, \mathsf{msk}, m \in \mathsf{M}.\mathcal{M}) \to \sigma_{\mathsf{alg}}$**:** Outputs the tag σ_{alg}.
$\mathsf{M.Ver}(\mathsf{mpp}, \mathsf{ipp}, \mathsf{msk}, m, \sigma_{\mathsf{alg}}) \to \{0, 1\}$**:** Verify whether the message-tag pair $(m, \sigma_{\mathsf{alg}})$ is valid with respect to msk.

Definition 2.10 (UF-RMA of M). *A MAC M is unforgeable against random message attack (UF-RMA) if for any PPT adversary $\mathcal{A}$, the following advantage is negligible:*

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

Σ-PROTOCOLS. We establish Σ-protocol notation and security. Let $\mathfrak{R} = \{(\mathbb{x}, \mathbb{w})\}$ be an NP-relation and denote by $\mathfrak{L}_{\mathfrak{R}} = \{\mathbb{x} \mid \exists \mathbb{w} \text{ s.t. } (\mathbb{x}, \mathbb{w}) \in \mathfrak{R}\}$ the induced language. A parameterized Σ-protocol Σ for $\mathfrak{R}$ with challenge space $\Sigma.\mathcal{C}$ is a tuple of PPT algorithms $(\Sigma.\mathsf{Init}, \Sigma.\mathsf{Resp}, \Sigma.\mathsf{Ver})$, where $\Sigma.\mathsf{Resp}$ and $\Sigma.\mathsf{Ver}$ are deterministic, such that

$\Sigma.\mathsf{Init}(\mathbb{x}, \mathbb{w}) \to (A, \mathsf{st})$**:** Given a statement-witness pair $(\mathbb{x}, \mathbb{w}) \in \mathfrak{R}$, outputs a first flow message (i.e., nonce) A and a state st.
$\Sigma.\mathsf{Resp}(\mathsf{st}, c) \to z$**:** Given a state st and a challenge $c \in \Sigma.\mathcal{C}$, outputs a third flow message (i.e., response) z.
$\Sigma.\mathsf{Ver}(\mathbb{x}, A, c, z) \to b$**:** Given a statement $\mathbb{x}$, a nonce A, challenge $c \in \Sigma.\mathcal{C}$, and a response z, outputs a bit $b \in \{0, 1\}$.

We refer to the tuple $\tau = (A, c, z)$ as *transcript*. We say that τ is valid for $\mathbb{x}$ if $\mathsf{Ver}(\mathbb{x}, A, c, z)$ outputs 1. We may omit $\mathbb{x}$ if clear by context. We recall desired security notions for Σ-protocols below, following [KR25].

Definition 2.11 (Correctness). *Let $\mathfrak{R}$ be an NP-relation and Σ be a Σ-protocol for $\mathfrak{R}$ with challenge space $\Sigma.\mathcal{C}$. We say Σ is correct, if for all $(\mathbb{x}, \mathbb{w}) \in \mathfrak{R}$, the following probability is 1:*

$$\mathsf{Pr}\left[\Sigma.\mathsf{Ver}(\mathbb{x}, A, c, z) = 1 \mid (A, \mathsf{st}) \leftarrow\!\!\$\ \Sigma.\mathsf{Init}(\mathbb{x}, \mathbb{w});\ c \leftarrow\!\!\$\ \Sigma.\mathcal{C};\ z \leftarrow \Sigma.\mathsf{Resp}(\mathsf{st}, c)\right] .$$

Definition 2.12 (Special Soundness). *Let $\mathfrak{R}$ be an NP-relation and Σ be an Σ-protocol for $\mathfrak{R}$. We call Σ (2-)special sound, if there exists a deterministic polynomial-time extractor $\Sigma.\mathsf{Ext}$ that given statement $\mathbb{x}$ and two valid transcripts $\{(A, c_b, z_b)\}_{b\in\{0,1\}}$ with $c_0 \neq c_1$, outputs a witness $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in \mathfrak{R}$.*

Definition 2.13 (SHVZK). *Let $\mathfrak{R}$ be an NP-relation and Σ be an Σ-protocol for $\mathfrak{R}$. We say that Σ is perfect special honest-verifier zero-knowledge (*SHVZK*), if there exists a PPT zero-knowledge simulator $\Sigma.\mathsf{Sim}$ such that for any $(\mathbb{x}, \mathbb{w}) \in \mathfrak{R}$ and $c \in \Sigma.\mathcal{C}$ that $\mathsf{D}_{\mathsf{Real}} = \mathsf{D}_{\mathsf{SHVZK}}$ for*

$$\begin{aligned}\mathsf{D}_{\mathsf{Real}} &:= \{(\mathbb{x}, \mathbb{w}, (A, c, z)) \mid A \leftarrow\!\!\$\ \Sigma.\mathsf{Init}(\mathbb{x}, \mathbb{w});\ z \leftarrow \Sigma.\mathsf{Resp}(\mathsf{st}, c)\},\\ \mathsf{D}_{\mathsf{SHVZK}} &:= \{(\mathbb{x}, \mathbb{w}, (A, c, z)) \mid (A, z) \leftarrow \Sigma.\mathsf{Sim}(\mathbb{x}, c)\}.\end{aligned}$$

Definition 2.14 ((Relaxed) Unique Responses). *Let Σ be a Σ-protocol for relation $\mathfrak{R}$. We say that Σ has unique responses if for any statement $\mathbb{x}$ and (A, c) there exists at most one z such that (A, c, z) is valid for $\mathbb{x}$. We say that Σ has relaxed unique responses for relation $\widetilde{\mathfrak{R}}$ if there exists a deterministic polynomial-time extractor $\Sigma.\mathsf{UExt}$ that given statement $\mathbb{x}$ and two valid transcripts $\{(A, c, z_b)\}_{b\in\{0,1\}}$ with $z_0 \neq z_1$, outputs a witness $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in \widetilde{\mathfrak{R}}$.*

We further recall the notions of Rand-randomizable transcripts (defined for a specific efficient randomized algorithm Rand) and a useful corollary from [KR25] which facilitates proving blindness. We note that the strong randomizable transcripts property is well-defined since the simulator is well-defined for all statements $\mathbb{x}$ (but SHVZK only says that $\Sigma.\mathsf{Sim}$ is identically distributed to a usual transcript for valid statements $\mathbb{x} \in \mathfrak{L}_{\mathfrak{R}}$).

Definition 2.15 (Rand-Randomizable Transcripts). *Let Σ be a Σ-protocol for relation $\mathfrak{R}$ with challenge space $\Sigma.\mathcal{C}$, and suppose Σ is* SHVZK*. The algorithm $\mathsf{Rand}(\mathbb{x}, (A, c, z))$, given a valid transcript (A, c, z) for $\mathbb{x}$, outputs a new valid transcript for $\mathbb{x}$.*

We say Σ has (perfect) Rand-randomizable transcripts (resp. strongly Rand-randomizable transcripts) for all $\mathbb{x} \in \mathfrak{L}_{\mathfrak{R}}$ (resp. all $\mathbb{x}$) and all accepting $\pi^* = (A^*, c^*, z^*)$, the following distributions are identical:

$$\begin{aligned}\mathsf{D}_{\mathsf{SHVZK}} &:= \{(\mathbb{x}, (A, c, z)) \mid c \leftarrow\!\!\$\ \Sigma.\mathcal{C};\ (A, z) \leftarrow\!\!\$\ \Sigma.\mathsf{Sim}(\mathbb{x}, c)\}\ ,\\ \mathsf{D}_{\mathsf{Rand}} &:= \{(\mathbb{x}, (A, c, z)) \mid (A, c, z) \leftarrow\!\!\$\ \mathsf{Rand}(\mathbb{x}, \pi^*)\}\ .\end{aligned}$$

Corollary 2.16. *Suppose Σ is a Σ-protocol for relation $\mathfrak{R}$ which is SHVZK and has Rand-randomizable transcripts. Then for all $(\mathbb{x}, \mathbb{w}) \in \mathfrak{R}$ and all accepting $\pi^* = (A^*, c^*, z^*)$, the following distributions are identical:*

$$\begin{aligned}\mathsf{D}_{\mathsf{Real}} &:= \{(\mathbb{x}, (A, c, z)) \mid A \leftarrow \mathsf{Init}(\mathbb{x}, \mathbb{w});\ c \leftarrow\!\!\$\ \Sigma.\mathcal{C};\ z \leftarrow \mathsf{Resp}(\mathsf{st}, c)\}\\ \mathsf{D}_{\mathsf{SHVZK}} &:= \{(\mathbb{x}, (A, c, z)) \mid c \leftarrow\!\!\$\ \Sigma.\mathcal{C};\ (A, z) \leftarrow \Sigma.\mathsf{Sim}(\mathbb{x}, c)\}\\ \mathsf{D}_{\mathsf{Rand}} &:= \{(\mathbb{x}, (A, c, z)) \mid (A, c, z) \leftarrow \mathsf{Rand}(\mathbb{x}, \pi^*)\}\end{aligned}$$

Canonical Σ-Protocols Let $\phi\colon \mathcal{W} \to \mathcal{X}$ be a $\mathbb{Z}_p$-linear map. We will instantiate our Σ-protocols with canonical Σ-protocols for linear maps [Mau15, KR25]. The canonical Σ-protocol Σ_ϕ for the relation $\mathfrak{R}_\phi := \{(\mathbb{x} = (\phi, T), \mathbb{w}) \mid \phi(\mathbb{w}) = T\}$ with challenge space $\mathcal{C} = \mathbb{Z}_p$ is as follows.

- $\Sigma_\phi.\mathsf{Init}(\mathbb{x}, \mathbb{w}) \to (A, \mathsf{st})$: Sample $r \leftarrow\!\!\$\ \mathcal{W}$. Output (st, A) where $\mathsf{st} \leftarrow (\mathbb{w}, r)$ and $A = \phi(r)$.
- $\Sigma_\phi.\mathsf{Resp}(\mathsf{st}, c) \to z$: Output $z \leftarrow c\mathbb{w} + r$
- $\Sigma_\phi.\mathsf{Ver}(\mathbb{x}, A, c, z) \to b$: Check that elements are in their respective spaces of definition. [3] Return 1 if $\phi(z) = A + cT$, else 0.

This concept naturally generalizes to families of maps ϕ_{pp} indexed by parameters pp. The following lemma summarizes well-known properties of Σ_ϕ (cf. [KR25, Lemma 2.9]) with some additions.

[3] We will leave this step implicit in the following.

Algorithm $\mathsf{Fork}^{\mathcal{A}}(\mathsf{inp})$:	Algorithm $\mathsf{Fork}_3^{\mathcal{A}}(\mathsf{inp})$:
1 : Pick random coins $\rho_{\mathcal{A}}$; $\boldsymbol{h} \leftarrow\!\!\$\ \mathcal{C}^Q$	1 : Pick random coins $\rho_{\mathcal{A}}$; $\boldsymbol{h} \leftarrow\!\!\$\ \mathcal{C}^Q$
2 : $(I, \mathsf{aux}) \leftarrow \mathcal{A}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})$	2 : $((I_0, I_1), (\mathsf{aux}_{I_k})_{k\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})$
3 : **if** $I = \perp$ **then return** $\perp$	3 : **if** $(I_0, I_1) = \perp$ **then return** $\perp$
4 : $\boldsymbol{h}'_{[I:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I+1}$; $\boldsymbol{h}' = \boldsymbol{h}_{[1:I-1]} \| \boldsymbol{h}'_{[I:Q]}$	4 : $\boldsymbol{h}'_{[I_0:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_0+1}$; $\boldsymbol{h}' = \boldsymbol{h}_{[1:I_0-1]} \| \boldsymbol{h}'_{[I_0:Q]}$
5 : $(I', \mathsf{aux}') \leftarrow \mathcal{A}(\mathsf{inp}, \boldsymbol{h}'; \rho_{\mathcal{A}})$	5 : $((I'_0, I'_1), (\mathsf{aux}'_{I'_k})_{k\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{inp}, \boldsymbol{h}'; \rho_{\mathcal{A}})$
6 : **if** $I \neq I' \ \vee\ h_I = h'_I$ **then return** $\perp$	6 : $\boldsymbol{h}''_{[I_1:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_1+1}$; $\boldsymbol{h}'' = \boldsymbol{h}_{[1:I_1-1]} \| \boldsymbol{h}''_{[I_1:Q]}$
7 : **return** $(1, \mathsf{aux}, \mathsf{aux}')$	7 : $((I''_0, I''_1), (\mathsf{aux}''_{I''_k})_{k\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{inp}, \boldsymbol{h}''; \rho_{\mathcal{A}})$
	8 : **if** $I_0 \notin \{I'_0, I'_1\} \ \vee\ I_1 \notin \{I''_0, I''_1\} \ \vee$
	9 : $h_{I_0} = h'_{I_0} \ \vee\ h_{I_1} = h''_{I_1}$ **then return** $\perp$
	10 : **return** $(1, (\mathsf{aux}_{I_k})_{k\in\{0,1\}}, (\mathsf{aux}'_{I_0}, \mathsf{aux}''_{I_1}))$

Fig. 4. Forking algorithms

Lemma 2.17 (Properties of Σ_ϕ). *Let Σ_ϕ denote the above canonical Σ-protocol for relation $\mathfrak{R}_\phi$. Then Σ_ϕ is correct, 2-special sound,* SHVZK*, and has strongly* Rand*-Randomizable transcripts. More concretely,* Sim *and* Rand_ϕ *are defined as follows:*

- $\Sigma_\phi.\mathsf{Sim}(\mathbb{x}, c)$ *samples* $z \leftarrow\!\!\$\ \mathcal{W}$*, sets* $A \leftarrow c \cdot T - \phi(z)$ *and outputs* (A, z)*.*
- $\mathsf{Rand}_\phi(\mathbb{x}, (A, c, z))$ *samples* $\gamma \leftarrow\!\!\$\ \mathbb{Z}_p$ *and* $\alpha \leftarrow\!\!\$\ \mathcal{W}$ *and outputs*

$$(\widetilde{A}, \widetilde{c}, \widetilde{z}) = (A\ -\gamma T\ +\phi(\alpha)\ , c\ +\gamma\ , z\ +\alpha\ ) \ ..$$

Also, Σ_ϕ satisfies relaxed unique response for the relaxed relation $\widetilde{\mathfrak{R}}_\phi := \{(\mathbb{x} = (\phi, T), \mathbb{w} \neq 0) \mid \phi(\mathbb{w}) = 0 \in \mathcal{X}\}$. If ϕ is injective, then Σ_ϕ has unique responses.

Proof. The statement is shown in [KR25, Lemma 2.9] except the claim concerning relaxed unique response. The latter follows immediately from inspecting the verification algorithm. □

FORKING LEMMA. In our OMSUF proof, we require two variants of forking lemmas given below: the first is the version introduced by Bellare and Neven [BN06], while the second is an alternate version where the forking occurs at two indices with the proof given in Appendix B.

Lemma 2.18 (Forking Lemma [BN06]). *Fix an integer $Q \geqslant 1$ and a set $\mathcal{C}$ of size $|\mathcal{C}| \geqslant 2$. Let $\mathcal{A}$ be a randomized algorithm that on input $\mathsf{inp}, \boldsymbol{h} \in \mathcal{C}^Q$ returns a pair $(I \in [Q] \cup \{\perp\}, \mathsf{aux})$. Let* IG *be a randomized algorithm for sampling* inp*. Let* $\mathsf{acc} := \mathsf{Pr}[I \neq \perp | \mathsf{inp} \leftarrow\!\!\$\ \mathsf{IG};\ \boldsymbol{h} \leftarrow\!\!\$\ \mathcal{C}^Q;\ (I, \mathsf{aux}) \leftarrow\!\!\$\ \mathcal{A}(x, h_1, \ldots, h_q)]$*. Let* $\mathsf{frk} = \mathsf{Pr}[b = 1 : \mathsf{inp} \leftarrow\!\!\$\ \mathsf{IG}; (b, \mathsf{aux}, \mathsf{aux}') \leftarrow\!\!\$\ \mathsf{Fork}^{\mathcal{A}}(\mathsf{inp})]$ *where* $\mathsf{Fork}^{\mathcal{A}}$ *is defined in Figure 4. Then,*

$$\mathsf{frk} \geqslant \mathsf{acc}\left(\frac{\mathsf{acc}}{Q} - \frac{1}{|\mathcal{C}|}\right) \text{ , or alternatively, } \mathsf{acc} \leqslant \sqrt{Q \cdot \mathsf{frk}} + \frac{Q}{|\mathcal{C}|} \ .$$

Lemma 2.19 (Two-Index Forking). *Fix $Q \geqslant 1$, $\mathcal{C}$, and* IG *be as in the prior lemma. Let $\mathcal{A}$ be a randomized algorithm that on input $\mathsf{inp}, \boldsymbol{h} \in \mathcal{C}^Q$ returns a pair $(I = (I_0, I_1) \in [Q]^2 \cup \{\perp\}, (\mathsf{aux}_{I_k})_{k\in\{0,1\}})$. Let* acc *still be defined as the probability that $\mathcal{A}$ outputs $I \neq \perp$. Let* $\mathsf{frk}_3 = \mathsf{Pr}[b = 1 : \mathsf{inp} \leftarrow\!\!\$\ \mathsf{IG}; (b, \mathsf{aux}, \mathsf{aux}') \leftarrow\!\!\$\ \mathsf{Fork}_3^{\mathcal{A}}(\mathsf{inp})]$ *where* $\mathsf{Fork}_3^{\mathcal{A}}$ *is defined in Figure 4. Then,*

$$\mathsf{frk}_3 \geqslant \frac{\mathsf{acc}}{2}\left(\frac{\mathsf{acc}}{4Q} - \frac{1}{|\mathcal{C}|}\right)^2 \text{ , or alternatively, } \mathsf{acc} \leqslant \sqrt[3]{32Q^2 \cdot \mathsf{frk}} + \frac{4Q}{|\mathcal{C}|} \ .$$

3 Three-move Blind Signatures from Algebraic MACs

In this section, we give our three-move blind signature construction based on pairing-free groups. We introduce our building blocks in Section 3.1, present our construction in Section 3.2, and discuss its properties in Section 3.3.

3.1 Building Blocks

LINEARLY-HOMOMORPHIC COMMITMENTS. We employ two linearly-homomorphic commitment schemes $\mathsf{C}_{\mathsf{mac}}$ and $\mathsf{C}_{\mathsf{rnd}}$ with message spaces $\mathbb{Z}_p^2$ and $\mathbb{G}^7$, respectively. Looking ahead, we employ $\mathsf{C}_{\mathsf{mac}}$ to fix parts of a MAC secret key and $\mathsf{C}_{\mathsf{rnd}}$ to fix blinding randomness in the protocol. For our instantiation, we may choose packed ElGamal [BBKS07] (for compactness) or Groth-Sahai commitments [GS08] (for statistical blindness) for $\mathsf{C}_{\mathsf{rnd}}$. We discuss more on this in Section 4.

Algebraic MAC. We describe the algebraic MAC we employ to tag the Okamoto-Schnorr nonces within our blind signature.

DDH-BASED ALGEBRAIC MAC. We rely on the DDH-based algebraic MAC scheme by Chase et al. [CMZ14], denoted $\mathsf{M}_{\mathsf{alg}}$. We give a brief description of the scheme below. Note in contrast to [CMZ14], we commit to the partial secret key (x_0, y_0) in the issuer's public parameters via $\mathsf{C}_{\mathsf{mac}}$.

$\mathsf{M}_{\mathsf{alg}}.\mathsf{Setup}(1^\lambda) \to \mathsf{mpp}$**:** The public parameters mpp (implicitly containing the group description of $\mathbb{G}$ with generator G) is $\mathsf{mpp} = (F, \mathsf{mck})$, where $F \leftarrow\!\!\$\ \mathbb{G}$ and $\mathsf{mck} \leftarrow\!\!\$\ \mathsf{C}_{\mathsf{mac}}.\mathsf{Setup}(1^\lambda)$.

$\mathsf{M}_{\mathsf{alg}}.\mathsf{KeyGen}(\mathsf{mpp}) \to (\mathsf{msk}, \mathsf{ipp})$**:** The secret key contains random scalars $x_0, x_1, y_0, y_1, z \leftarrow\!\!\$\ \mathbb{Z}_p$. The issuer's public parameters, containing a commitment of (x_0, y_0) and the embeddings of x_1, y_1, z in $\mathbb{G}$, will be used in the blind signature scheme along with a proof that the tags are valid. More precisely, we let $X_1 = x_1F, Y_1 = y_1F, Z = zF$, $\mathrm{cmt}_0 = \mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (x_0, y_0); \rho_0)$, and

$$\mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho_0)\ , \qquad \mathsf{ipp} = (\mathrm{cmt}_0, X_1, Y_1, Z)\ .$$

$\mathsf{M}_{\mathsf{alg}}.\mathsf{Tag}(\mathsf{mpp}, \mathsf{msk}, m) \to \sigma_{\mathsf{alg}}$**:** For a message $m \in \mathbb{Z}_p$, samples $u \leftarrow\!\!\$\ \mathbb{Z}_p^*$ and outputs $\sigma_{\mathsf{alg}} = (U, V, S, T)$ with

$$U = uG\ , \qquad V = zU\ ,$$
$$S = u(x_0 + x_1 m)G\ , \qquad T = u(y_0 + y_1 m)G\ .$$

We note that this is the version of the tagging algorithm that will be used in the OMUF security proof of our blind signature scheme.

$\mathsf{M}_{\mathsf{alg}}.\mathsf{Ver}(\mathsf{mpp}, \mathsf{ipp}, \mathsf{msk}, m, \sigma_{\mathsf{alg}}) \to \{0, 1\}$**:** To verify the tag $\sigma_{\mathsf{alg}} = (U, V, S, T)$ of a message $m \in \mathbb{Z}_p$, we use the secret key sk to efficiently check that

$$(U \neq 0_{\mathbb{G}}) \ \wedge\ (V = zU) \ \wedge\ (S = (x_0 + x_1 m)U) \ \wedge\ (T = (y_0 + y_1 m)U)\ .$$

We recall the following theorem adapted from the UF-CMA security of a variant of $\mathsf{M}_{\mathsf{alg}}$ without cmt_0 in ipp stated in [CAHLT25, Theorem 6.1.]. The additional hiding advantage is due to switching the cmt_0 to commit to $(0, 0)$ instead. We refer to Appendix Cfor the formal proof.

Theorem 3.1 (UF-RMA Security of $\mathsf{M}_{\mathsf{alg}}$). *Let* GGen *be a group parameters generator outputting groups of prime-order* $p = p(\lambda)$*,* $\mathsf{C}_{\mathsf{mac}}$ *be a commitment scheme,* $\mathsf{M}_{\mathsf{alg}} = \mathsf{M}_{\mathsf{alg}}[\mathsf{GGen}, \mathsf{C}_{\mathsf{mac}}]$ *be the algebraic MAC scheme described above. For any* $\mathcal{A}$ *running in time* $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$ *and making at most* $Q = Q(\lambda)$ *queries to the* Tag *oracle, there exist adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *running in time* $\approx T_{\mathcal{A}}$ *such that*

$$\mathsf{Adv}^{\mathsf{ufrma}}_{\mathsf{M}_{\mathsf{alg}}}(\mathcal{A}, \lambda) \leqslant \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_1, \lambda) + Q \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}_2, \lambda) + \frac{3}{p}\ .$$

TAGGING ON $\mathbb{G}$-MESSAGES. We describe an alternative mode to tag group elements. This mode will be used explicitly in our construction. In short, we replace mG with $M \in \mathbb{G}$ in $\mathsf{M}_{\mathsf{alg}}$ above. That is, for a message $M \in \mathbb{G}$, the tagging algorithm outputs a tag $\sigma = (U, V, S, T)$ such that for $u \leftarrow\!\!\$\ \mathbb{Z}_p^*$ we have

$$U = uG\ , \qquad V = zU\ ,$$
$$S = u(x_0 G + x_1 M)\ , \qquad T = u(y_0 G + y_1 M)\ .$$

We remark that the tag cannot be efficiently verified using the secret key as u is not known to the verifier. Nevertheless, there is an underlying structure expressible in terms of a relation induced by a linear map. In particular, we have the following statement and witness

$$\mathbb{x}_{\mathsf{mac}} = (\mathsf{ipp}, M, U, V, S_0, T_0, S_1 = 0_{\mathbb{G}}, T_1 = 0_{\mathbb{G}})\ ,$$
$$\mathbb{w}_{\mathsf{mac}} = (u' = u^{-1}, z, x_0, x_1, y_0, y_1, \rho_0)\ .$$

The corresponding map $\phi^{\mathsf{mac}}_{\mathsf{mpp},\mathbb{x}_{\mathsf{mac}}}$ is given below. Note that $\phi^{\mathsf{mac}}_{\mathsf{mpp},\mathbb{x}_{\mathsf{mac}}}$ is linear as $\mathsf{C}_{\mathsf{mac}}$ is linearly-homomorphic.

$$\begin{pmatrix} \mathrm{cmt}_0 \\ X_1 \\ Y_1 \\ Z \\ G \\ V \\ S_1 \\ T_1 \end{pmatrix} = \phi^{\mathsf{mac}}_{\mathsf{mpp},\mathbb{x}_{\mathsf{mac}}}(\mathbb{w}_{\mathsf{mac}}) := \begin{pmatrix} \mathsf{C}_{\mathsf{mac}}(\mathsf{mck}, (x_0, y_0); \rho_0) \\ x_1 F \\ y_1 F \\ zF \\ u'U \\ zU \\ x_0 G + x_1 M - u' S_0 \\ y_0 G + y_1 M - u' T_0 \end{pmatrix}\ . \tag{2}$$

RERANDOMIZING THE MAC TAGS. For blindness, we need to rerandomize the MAC tags and also hide the underlying group element message. We take a similar approach as done in [CMZ14] but with modifications to accommodate the group element message. In particular, the user will compute the following group elements

$$\widetilde{U} \leftarrow \beta U\ , \qquad \widetilde{V} \leftarrow \beta V\ , \qquad \widetilde{M} \leftarrow M + r_M F\ ,$$
$$\widetilde{S}_0 \leftarrow \beta S - r_x \widetilde{U}\ , \qquad \widetilde{S}_1 \leftarrow r_x G + r_M X_1\ ,$$
$$\widetilde{T}_0 \leftarrow \beta T - r_y \widetilde{U}\ , \qquad \widetilde{T}_1 \leftarrow r_y G + r_M Y_1\ .$$

where $\beta \leftarrow\!\!\$\ \mathbb{Z}_p^*, r_M, r_x, r_y \leftarrow\!\!\$\ \mathbb{Z}_p$. Note that the following statement and witness is still in the relation induced by the map $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}$ such that

$$\mathbb{x}_{\mathsf{mac}} = (\mathsf{ipp}, \widetilde{M}, \widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{T}_0, \widetilde{S}_1, \widetilde{T}_1)\ , \mathbb{w}_{\mathsf{mac}} = (u' = u^{-1}\beta^{-1}, x_0, x_1, y_0, y_1, z, \rho_0)\ .$$

Σ-protocols. We will use multiple Σ-protocols in our construction. We assume that all these Σ-protocols share the following public parameters (generated by the setup of our blind signature scheme) which will be a part of the statement

$$\mathsf{pp} = (\mathbb{G}, p, G, H, \mathsf{mpp}, \mathsf{bck}, D_1)\ ,$$

where $\mathsf{mpp} = (F, \mathsf{mck})$ is as above in Section 3.1, and $H \leftarrow\!\!\$\ \mathbb{G}$, $\mathsf{bck} \leftarrow \mathsf{C}_{\mathsf{rnd}}.\mathsf{Setup}(1^\lambda)$ and $D_1 \leftarrow\!\!\$\ \mathbb{G}$. (By slight abuse of notation, we will pass these as input to the Σ-protocol algorithms for concreteness.) The commitment key bck is later employed to commit to the blinding factors for the rerandomization of $\mathsf{M}_{\mathsf{alg}}$.

Σ-PROTOCOL FOR OKAMOTO-SCHNORR. We denote by Σ_{os} the Σ-protocol underlying the Okamoto-Schnorr identification scheme [Sch90, Oka93]. The statement and witness are defined as follows

$$\mathbb{x}_{\mathsf{os}} = (W \in \mathbb{G})\ , \qquad \mathbb{w}_{\mathsf{os}} = (w_0, w_1) \in \mathbb{Z}_p^2\ .$$

The underlying relation defined by the linear map $\phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}$ is

$$W = \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\mathbb{w}_{\mathsf{os}}) := w_0 G + w_1 H \ . \tag{3}$$

We note that Σ_{os} satisfies relaxed unique-response for the relation $\widetilde{\mathfrak{R}}_{\mathsf{os}}$ corresponding to $\mathbb{w} \neq 0$ such that $\phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\mathbb{w}) = 0_{\mathbb{G}}$.

Σ-PROTOCOL FOR MAC VALIDITY. We denote Σ_{mac} to be the Σ-protocol for the relation induced by $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}$ given in Equation (2). We note that Σ_{mac} satisfies relaxed unique-response for the relation $\widetilde{\mathfrak{R}}_{\mathsf{mac}}$ corresponding to witnesses $\mathbb{w} \neq 0$ such that $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}(\mathbb{w}) = \mathbf{0}_{\mathbb{G}}$.

Σ-PROTOCOL FOR EQUALITY OF DISCRETE LOGARITHMS. We denote Σ_{dleq} as a simple Σ-protocol for proving membership of the discrete-logarithm equality language. The statement and witness are defined as follows

$$\mathbb{x}_{\mathsf{dleq}} = (D_1, D_2, D_3) \ , \qquad\qquad \mathbb{w}_{\mathsf{dleq}} = d \in \mathbb{Z}_p \ .$$

The underlying relation $\mathfrak{R}_{\mathsf{dleq}}$ is induced by the linear map $\phi^{\mathsf{dleq}}_{\mathsf{pp},\mathbb{x}_{\mathsf{dleq}}}$ defined as

$$\begin{pmatrix} D_2 \\ D_3 \end{pmatrix} = \phi^{\mathsf{dleq}}_{\mathsf{pp},D_1}(\mathbb{w}_{\mathsf{dleq}}) := \begin{pmatrix} dG \\ dD_1 \end{pmatrix} \ . \tag{4}$$

We note that Σ_{dleq} satisfies unique responses.

Σ-PROTOCOL FOR MAC BLINDING FACTORS. For our blind signature construction, we need the user to commit to the original nonce and their blinding factors along with proving knowledge of these blinding factors and the unblinded MAC tag. For this, the commitment $\mathrm{cmt}_{\mathsf{rnd}}$ will commit to

- the original Okamoto-Schnorr nonce A_{os},
- the blinding factors $\alpha_0, \alpha_1, \gamma$ for Okamoto-Schnorr nonce, and
- the blinding factors r_x, r_y for the MACs, and
- the blinding factor r_M for the tagged message.

Formally, we define the Σ-protocol Σ_{rnd} for the relation $\mathfrak{R}_{\mathsf{rnd}}$ with the following statement and witness

$$\begin{aligned} \mathbb{x}_{\mathsf{rnd}} &= (W, \widetilde{A}_{\mathsf{os}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{M}, \widetilde{U}, \widetilde{S}_1, \widetilde{T}_1) \ , \\ \mathbb{w}_{\mathsf{rnd}} &= (\rho_{\mathsf{rnd}}, \alpha_0, \alpha_1, \gamma, r_x, r_y, r_M) \ , \end{aligned}$$

which satisfy the identities induced by the map $\phi^{\mathsf{rnd}}_{\mathsf{pp},\mathbb{x}_{\mathsf{rnd}}}$ defined as follows. Let $\boldsymbol{B} = (\alpha_0\widetilde{U}, \alpha_1\widetilde{U}, \gamma\widetilde{U}, r_x\widetilde{U}, r_y\widetilde{U}, r_M\widetilde{U}, \gamma W - \alpha_0 G - \alpha_1 H)$ be a placeholder for the committed values (and is a linear function over the witness) and $\mathrm{cmt}' = \mathsf{C}_{\mathsf{rnd}}.\mathsf{Commit}(\mathsf{ck}, (0_{\mathbb{G}}, \ldots, 0_{\mathbb{G}}, \widetilde{A}_{\mathsf{os}}); 0)$, and let

$$\begin{pmatrix} \mathrm{cmt}_{\mathsf{rnd}} - \mathrm{cmt}' \\ \widetilde{M} - \widetilde{A}_{\mathsf{os}} \\ \widetilde{S}_1 \\ \widetilde{T}_1 \end{pmatrix} = \phi^{\mathsf{rnd}}_{\mathsf{pp},\mathbb{x}_{\mathsf{rnd}}}(\mathbb{w}_{\mathsf{rnd}}) := \begin{pmatrix} \mathsf{C}_{\mathsf{rnd}}.\mathsf{Commit}(\mathsf{ck}, \boldsymbol{B}; \rho_{\mathsf{rnd}}) \\ r_M F - \alpha_0 G - \alpha_1 H + \gamma W \\ r_x G + r_M X_1 \\ r_y G + r_M Y_1 \end{pmatrix}. \tag{5}$$

The equality on the first entry shows that $\mathrm{cmt}_{\mathsf{rnd}}$ commits to the blinding factors and $A_{\mathsf{os}} = \widetilde{A}_{\mathsf{os}} - \alpha_0 G - \alpha_1 H + \gamma W$. The second one shows that $\widetilde{M} = \widetilde{A}_{\mathsf{os}} - \alpha_0 G - \alpha_1 H + \gamma W + r_M F = A_{\mathsf{os}} + r_M F$. For the third and fourth entries, the equations show that $\widetilde{S}_1, \widetilde{T}_1$ are computed with respect to the blinding factors r_x, r_y, r_M. We also note that Σ_{rnd} satisfies relaxed unique response for the following relaxed relation $\widetilde{\mathfrak{R}}_{\mathsf{rnd}}$ containing witnesses $\mathbb{w} \neq 0 \in \mathsf{C}_{\mathsf{rnd}}.\mathcal{R} \times \mathbb{Z}_p^6$ such that $\phi^{\mathsf{rnd}}_{\mathsf{pp},\mathbb{x}_{\mathsf{rnd}}}(\mathbb{w}) = 0$.

Remark 3.2 (Interplay of Σ_{mac} and Σ_{rnd}). We note that given a blinded $\mathsf{M}_{\mathsf{alg}}$ tag $\widetilde{\sigma}_{\mathsf{mac}} = (\widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{T}_0, \widetilde{S}_1, \widetilde{T}_1)$ and $\widetilde{M}$ such that (1) along with ipp, the tuple $(\mathsf{ipp}, \widetilde{M}, \widetilde{\sigma})$ is in the language induced by $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}$, and (2) when additionally given the witness $\mathbb{w}_{\mathsf{rnd}} = (\rho_{\mathsf{rnd}}, \alpha_0, \alpha_1, \gamma, r_x, r_y, r_M)$ of the relation induced by $\phi^{\mathsf{rnd}}_{\mathsf{pp},\mathbb{x}_{\mathsf{rnd}}}$, one can recover a valid MAC tag $\sigma^*_{\mathsf{mac}} = (\widetilde{U}, \widetilde{V}, \widetilde{S}_0 + r_x\widetilde{U}, \widetilde{T}_0 + r_y\widetilde{U})$ for $m^* = \mathsf{dlog}_G A_{\mathsf{os}} = \mathsf{dlog}_G \widetilde{A}_{\mathsf{os}} - \alpha_0 - \alpha_1 \mathsf{dlog}_G H + \gamma_{\mathsf{os}} \mathsf{dlog}_G W$. In our security analysis we will leverage MAC security in one step (cf. **Case 2** in Section 3.3). In that case, the elements H and W will be generated with known discrete logarithms, so the reduction can recover the discrete logarithm of $\widetilde{A}_{\mathsf{os}}$ from the Σ_{os} transcript. This allows us to employ MAC security for $\mathbb{Z}_p$ elements.

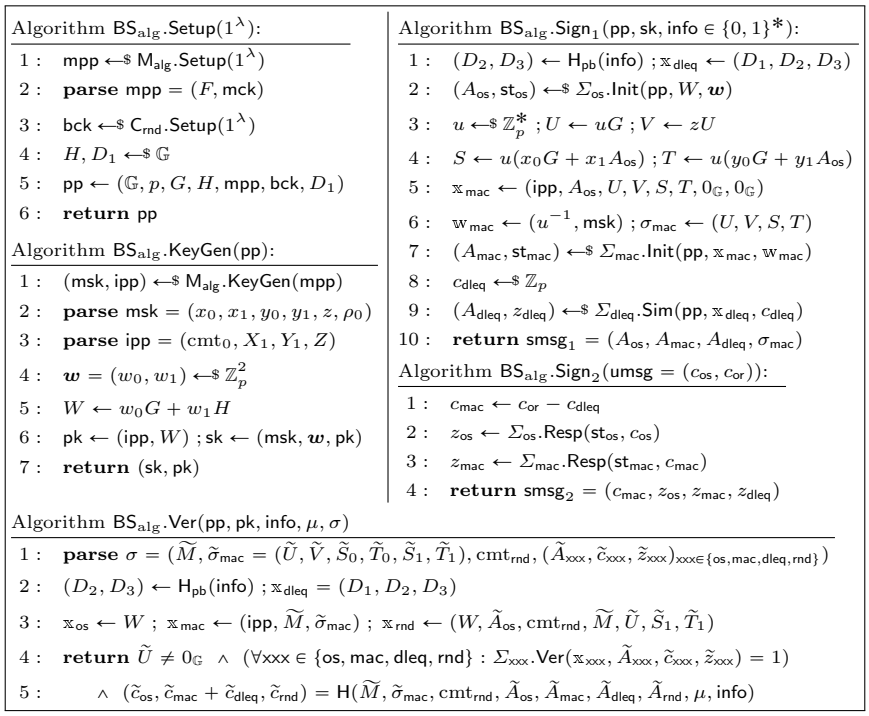

Fig. 5. Setup, key generation, verification, and signer-side algorithms of the three-move blind signature scheme $\mathsf{BS}_{\mathsf{alg}} = \mathsf{BS}_{\mathsf{alg}}[\mathsf{GGen}, \mathsf{C}_{\mathsf{mac}}, \mathsf{C}_{\mathsf{rnd}}, \mathsf{H}, \mathsf{H}_{\mathsf{pb}}]$ and $\mathsf{M}_{\mathsf{alg}} = \mathsf{M}_{\mathsf{alg}}[\mathsf{GGen}, \mathsf{C}_{\mathsf{mac}}]$. Here, $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_p^3$ and $\mathsf{H}_{\mathsf{pb}} : \{0,1\}^* \to \mathbb{G}^2$. Recall that $(\mathbb{G}, p, G) \leftarrow\!\!\$\ \mathsf{GGen}(1^\lambda)$ is an implicit input for all algorithms. We omitted the state for $\mathsf{BS}_{\mathrm{alg}}.\mathsf{Sign}_2$, assuming that this algorithm can access values from the prior round.

3.2 Construction

We now describe our blind signature construction in Figure 5 (the signer-side algorithms) and Figure 6 (the user-side algorithms). The secret key sk consists of the $\mathsf{M}_{\mathsf{alg}}$ secret key msk and the Okamoto-Schnorr witness $\boldsymbol{w} \in \mathbb{Z}_p^2$ with the public key containing the corresponding issuer parameters ipp and $W = \phi^{\mathsf{os}}_{\mathsf{pp},W}(\boldsymbol{w})$.

In the signing protocol, the signer computes the Σ_{os} nonce A_{os}, uses msk to compute its tag $\sigma_{\mathsf{mac}} = (U, V, S, T)$, and computes the nonces for the OR-proof between (a) Σ_{mac}, which proves that σ_{mac} satisfies certain structure in Equation (2) with respect to ipp and A_{os}, and (b) Σ_{dleq}, which proves that (D_2, D_3) (computed by hashing the common message info) has the same discrete logarithm with respect to the bases (G, D_1) in pp (this side is simulated). The user then blinds the $\mathsf{M}_{\mathsf{alg}}$ tag along with the nonces while proving knowledge (via Σ_{rnd}) of the blinding factors $(\alpha_{\mathsf{os}}, \gamma_{\mathsf{os}}, r_x, r_y, r_M)$ (committed to via $\mathsf{C}_{\mathsf{rnd}}$ [4]) for A_{os} and σ_{mac}. The user then hashes the blinded tag and nonces to obtain the challenges $(\widetilde{c}_{\mathsf{os}}, \widetilde{c}_{\mathsf{or}}, \widetilde{c}_{\mathsf{rnd}})$, respectively for Σ_{os}, the OR-proof between Σ_{mac} and Σ_{dleq}, and Σ_{rnd}, which are blinded and sent to the signer. The signer finalizes its side by computing the corresponding responses. The user finally computes the signature by blinding the responses.

Remark 3.3 (Desirable Properties and Optimizations). Our setup is transparent (the setups of $\mathsf{C}_{\mathsf{mac}}$ and $\mathsf{C}_{\mathsf{rnd}}$ are transparent in our instantiations), and can be generated by simply hashing a fixed string as with some prior blind signatures [KRS23, KR25]. Also, as with many existing signatures from the Fiat-Shamir paradigm and canonical Σ-protocols (e.g., Schnorr signatures [Sch90]), the nonces $\widetilde{A}_{\mathsf{xxx}}$ in our signature can be efficiently reconstructed from the statement $\mathbb{x}_{\mathsf{xxx}}$, the challenge $\widetilde{c}_{\mathsf{xxx}}$, and the response $\widetilde{z}_{\mathsf{xxx}}$ (see the

[4] Notably, we need to embed these blinding factors with base $\widetilde{U}$ due to a technical step in the OMSUF proof—we further expand on this in Section 3.3

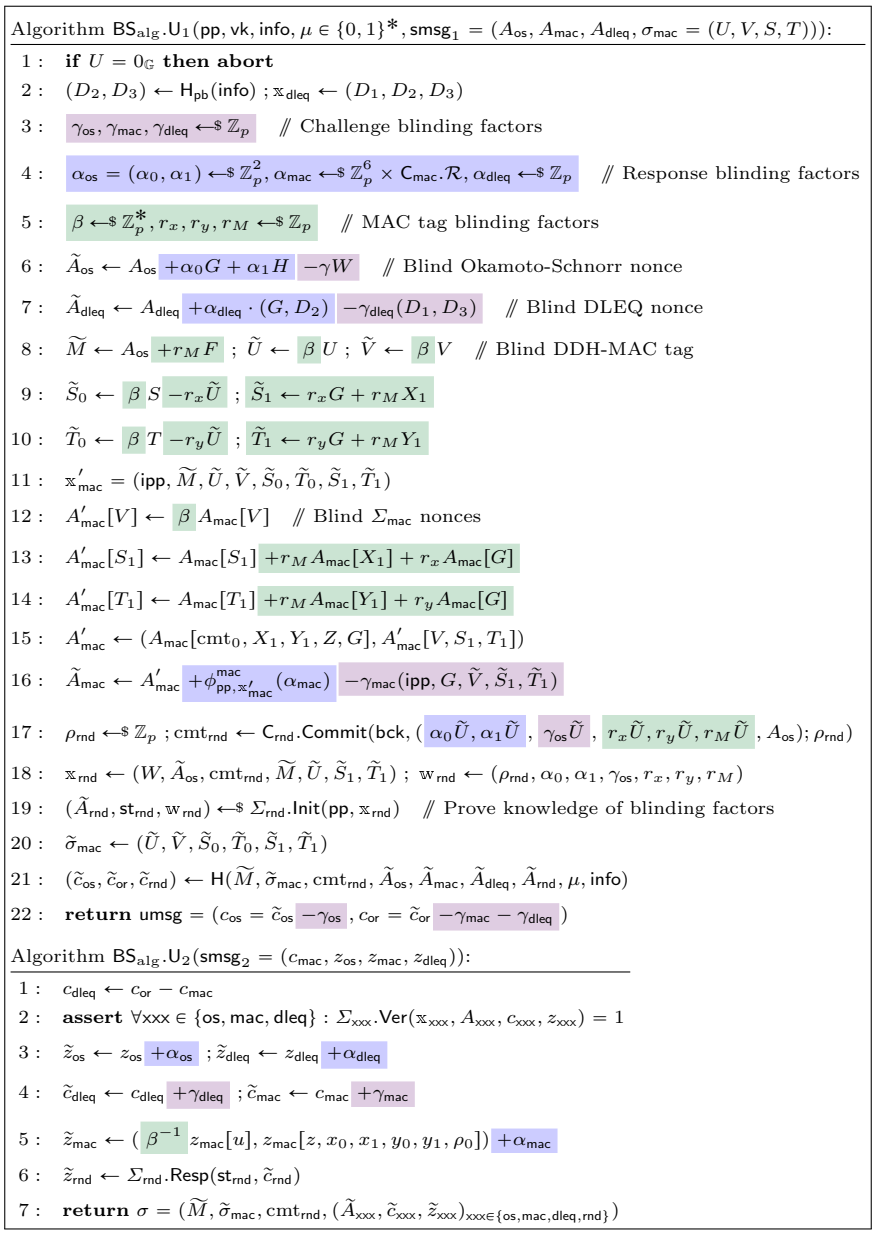

Fig. 6. User-side algorithms of the blind signature scheme $\mathsf{BS}_{\mathrm{alg}}$ with the state for $\mathsf{BS}_{\mathrm{alg}}.\mathsf{U}_2$ omitted, assuming that this algorithm can access values from the prior round. We denote by $A_{\mathsf{mac}}[\mathsf{xxx}]$ the component of A_{mac} that corresponds to value xxx. That is, $A_{\mathsf{mac}}[\mathrm{cmt}_0] = (A_{\mathsf{mac},1}, \ldots, A_{\mathsf{mac},\ell_{\mathcal{C}}})$, $A_{\mathsf{mac}}[X_1] = A_{\mathsf{mac},\ell_{\mathcal{C}}+1}$, $A_{\mathsf{mac}}[Y_1] = A_{\mathsf{mac},\ell_{\mathcal{C}}+2}$, and so on, where $\ell_{\mathcal{C}}$ is the dimension of $\mathsf{C}_{\mathsf{mac}}.\mathcal{C}$. We follow the same convention for $\widetilde{z}_{\mathsf{mac}}$. For readability, we highlight blinding of challenge, responses and MAC-related terms by , , and , respectively.

notations in $\mathsf{BS.Ver}$ in Figure 5). This allows us to reduce the signature size significantly by omitting the nonces in the derived signature. Additionally, if the scheme is to be used as a (non-partially) blind signature scheme, the signer's first move can be preprocessed.

3.3 Correctness and Security

In this section, we establish correctness and concrete security guarantees for one-more strong unforgeability and partial blindness of $\mathsf{BS}_{\mathrm{alg}}$ in the following theorems. The correctness proof is given in Section 5, the OMSUF proof is given in Section 6, while the partial blindness proof is given in Section 7.

Theorem 3.4 (Correctness). *Let* GGen *be a group parameters generator outputting groups of prime-order* $p = p(\lambda)$, $\mathsf{C}_{\mathsf{mac}}, \mathsf{C}_{\mathsf{rnd}}$ *be commitment schemes,* $\mathsf{M}_{\mathsf{alg}} = \mathsf{M}_{\mathsf{alg}}[\mathsf{GGen}, \mathsf{C}_{\mathsf{mac}}]$ *be the algebraic MAC scheme described in Section 3.1, and* $\mathsf{H}, \mathsf{H}_{\mathsf{pb}}$ *be random oracles. Let* $\mathsf{BS}_{\mathrm{alg}} = \mathsf{BS}_{\mathrm{alg}}[\mathsf{GGen}, \mathsf{C}_{\mathsf{mac}}, \mathsf{C}_{\mathsf{rnd}}, \mathsf{H}, \mathsf{H}_{\mathsf{pb}}]$. *The blind signature* $\mathsf{BS}_{\mathrm{alg}}$ *is perfectly correct.*

Proof (Sketch). This follows from straightforward calculation. In particular, $\widetilde{U} \neq 0$ and the hash verification query are immediate. The transcript $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ is computed from a witness $\mathbb{w}_{\mathsf{rnd}}$ such that $(\mathbb{x}_{\mathsf{rnd}}, \mathbb{w}_{\mathsf{rnd}}) \in \mathfrak{R}_{\mathsf{rnd}}$ and is therefore valid. Note that the signer-issued Σ_{os} transcript $(A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ is computed with witness $\mathbb{w}_{\mathsf{os}}$ such that $(\mathbb{x}_{\mathsf{os}}, \mathbb{w}_{\mathsf{os}}) \in \mathfrak{R}_{\mathsf{os}}$ and the signer-issued Σ_{dleq} transcript $(A_{\mathsf{dleq}}, c_{\mathsf{dleq}}, z_{\mathsf{dleq}})$ is simulated for $\mathbb{x}_{\mathsf{dleq}}$. Further, observe that the blinded Σ-protocols transcripts $(\widetilde{A}_{\mathsf{xxx}}, \widetilde{c}_{\mathsf{xxx}}, \widetilde{z}_{\mathsf{xxx}})$ for $\mathsf{xxx} \in \{\mathsf{os}, \mathsf{dleq}\}$ are obtained by the user running Rand specified in Lemma 2.17 on the signer-issued transcripts $(A_{\mathsf{xxx}}, c_{\mathsf{xxx}}, z_{\mathsf{xxx}})$. Thus, these transcripts remain valid by Definition 2.15. The Σ_{mac} transcript requires more calculation and we refer to the full proof for details. Roughly, we mimic the blinding of σ_{mac} on the Σ-protocol transcript to adapt $\tau_{\mathsf{mac}} = (A_{\mathsf{mac}}, c_{\mathsf{mac}}, z_{\mathsf{mac}})$ into a valid transcript for $\widetilde{\sigma}_{\mathsf{mac}}$, leveraging the linear structure of Σ_{mac}. These steps are highlighted with in Figure 6. The adapted transcript is valid for the blinded MAC $\widetilde{\sigma}_{\mathsf{mac}}$ and is randomized via Rand as the other transcripts. □

Theorem 3.5 (One-More Strong Unforgeability). *Let* $(\mathsf{GGen}, p, \mathsf{C}_{\mathsf{mac}}, \mathsf{C}_{\mathsf{rnd}}, \mathsf{BS}_{\mathrm{alg}}, \mathsf{M}_{\mathsf{alg}})$ *be as in Theorem 3.4, and* $\mathsf{H}, \mathsf{H}_{\mathsf{pb}}$ *be random oracles. Assume also that* $\mathsf{C}_{\mathsf{mac}}$ *is perfectly strong binding and* $\mathsf{C}_{\mathsf{rnd}}$ *satisfies extractability.*

For any adversary $\mathcal{A}$ *running in time* $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$, *making at most* $Q_{\mathrm{S}} = Q_{\mathrm{S}}(\lambda), Q_{\mathsf{H}} = Q_{\mathsf{H}}(\lambda)$, *and* $Q_{\mathsf{pb}} = Q_{\mathsf{pb}}(\lambda)$ *queries to oracles* S_1, H, *and* H_{pb}, *respectively, there exist adversaries* $\mathcal{B}_{\mathsf{ur},\mathsf{xxx}}$ *(for* $\mathsf{xxx} \in \{\mathsf{os}, \mathsf{mac}, \mathsf{rnd}\}$*),* $\mathcal{B}_{\mathsf{dist}}$, $\mathcal{B}_{\mathsf{ddh},k}$ *(for* $k \in [4]$*),* $\mathcal{B}_{\mathsf{hide}}$, $\mathcal{B}_{\mathsf{dl}}$ *and* $\mathcal{B}_{\mathrm{alg}}$ *such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{omsuf}}_{\mathsf{BS}_{\mathrm{alg}}}(\mathcal{A}, \lambda) \leqslant\ & \frac{1 + Q_{\mathsf{pb}}(Q_{\mathsf{H}}^2 + 12Q_{\mathsf{H}} + 3Q_{\mathrm{S}} + 4)}{p} + \mathsf{Adv}^{\mathsf{dl}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ur},\mathsf{os}}, \lambda) + \mathsf{Adv}^{\mathsf{sbind}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_{\mathsf{ur},\mathsf{mac}}, \lambda) \\
& + \mathsf{Adv}^{\mathsf{sbind}}_{\mathsf{C}_{\mathsf{rnd}}}(\mathcal{B}_{\mathsf{ur},\mathsf{rnd}}, \lambda) + Q_{\mathsf{pb}} \cdot \Bigg(\mathsf{Adv}^{\mathsf{ext\text{-}dist}}_{\mathsf{C}_{\mathsf{rnd}},\mathsf{Ext}}(\mathcal{B}_{\mathsf{dist}}, \lambda) + 2\sqrt[3]{32Q_{\mathsf{H}}^2 \cdot \mathsf{Adv}^{\mathsf{dl}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{dl}}, \lambda)} \\
& + \mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathsf{pb}}}(\mathcal{B}_{\mathsf{ddh},1}, \lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathrm{S}}}(\mathcal{B}_{\mathsf{ddh},2}, \lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_{\mathsf{hide}}, \lambda) \\
& + \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ddh},3}, \lambda) + \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ddh},4}, \lambda) + \sqrt{Q_{\mathsf{H}} \cdot \mathsf{Adv}^{\mathsf{ufrma}}_{\mathsf{M}_{\mathsf{alg}}}(\mathcal{B}_{\mathrm{alg}}, \lambda)} \Bigg) .
\end{aligned}$$

Additionally, all adversaries run in time roughly $T_{\mathcal{A}}$, *except for* $\mathcal{B}_{\mathsf{dl}}$ *and* $\mathcal{B}_{\mathrm{alg}}$ *with running time roughly* $3T_{\mathcal{A}}$ *and* $2T_{\mathcal{A}}$, *respectively. Moreover,* $\mathcal{B}_{\mathrm{alg}}$ *makes* Q_{S} *queries to its* Tag *oracle.*

Proof (Detailed sketch). We assume w.l.o.g. that any OMSUF adversary finishes all signing sessions, as shown in Lemma 2.5. Our proof first moves to a game that guesses the common message info^* that $\mathcal{A}$ will forge on (via a guessing argument and hence the Q_{pb}-factor loss) and extracts (via extractability of $\mathsf{C}_{\mathsf{rnd}}$) the group elements committed to via $\mathrm{cmt}_{\mathsf{rnd}}$ of all forgeries, including the elements $(\overline{A}_0, \overline{A}_1, \overline{\Gamma}_{\mathsf{os}}, \overline{R}_x, \overline{R}_y, \overline{R}_M)$, which embed the blinding factors in their exponents to basis $\widetilde{U}$, and the underlying Okamoto-Schnorr nonce $\overline{A}_{\mathsf{os}}$. This induces the two following cases on collision of the extracted $\overline{A}_{\mathsf{os}}$ as discussed in Section 1.1.

Case 1: A pair of extracted $\overline{A}_{\mathsf{os}}$ collide. We consider two sub-cases.

- *The two corresponding forgeries share the same RO query in their verification.* We rely on (relaxed) unique-response properties of all the underlying Σ-protocols to extract hard to compute witnesses of $\widetilde{\mathfrak{R}}_{\mathsf{os}}, \widetilde{\mathfrak{R}}_{\mathsf{mac}}$, and $\widetilde{\mathfrak{R}}_{\mathsf{rnd}}$, and ultimately reduce to DL and strong binding of $\mathsf{C}_{\mathsf{mac}}$ and $\mathsf{C}_{\mathsf{rnd}}$.[5] We remark that this step is crucial for *strong unforgeability*. For OMUF, the messages which are inputs to H are distinct, so this case never occurs.
- *The two corresponding forgeries do not share the same RO query.* Intuitively, the blinding factors $\boldsymbol{\alpha}_{\mathsf{os}} = (\alpha_0, \alpha_1), \gamma_{\mathsf{os}}$ embedded in the extracted $(\overline{A}_0, \overline{A}_1, \overline{\Gamma}_{\mathsf{os}})$ of the two forgeries will allow us to obtain two (unblinded) Σ_{os} transcripts with the same nonce $\overline{A}_{\mathsf{os}}$. These determine a witness $\boldsymbol{w}'$ that can be efficiently computed upon rewinding. *Before rewinding* and extracting these blinding factors, we make two key steps to ensure that (a) the unblinded challenges for the two transcripts are distinct which allows us to apply the special-soundness extractor, and that (b) we extract a different witness $\boldsymbol{w}' \neq \boldsymbol{w}$ of W with sufficiently high probability. To see this more clearly, let $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ and $(\widetilde{A}'_{\mathsf{os}}, \widetilde{c}'_{\mathsf{os}}, \widetilde{z}'_{\mathsf{os}})$ be the blinded transcripts from these two signatures and $(\boldsymbol{\alpha}_{\mathsf{os}}, \gamma_{\mathsf{os}})$ and $(\boldsymbol{\alpha}'_{\mathsf{os}}, \gamma'_{\mathsf{os}})$ be the corresponding blinding factors.

  For (a), observe that the unblinded nonces being identical means

  $$\widetilde{c}_{\mathsf{os}} - \gamma_{\mathsf{os}} = \widetilde{c}'_{\mathsf{os}} - \gamma'_{\mathsf{os}} \, .$$

  With $\widetilde{c}_{\mathsf{os}}, \widetilde{c}'_{\mathsf{os}}$ being outputs of the random oracle H, this correlation only occurs with negligible probability (since γ_{os} and γ'_{os} are fixed when the queries are made). For (b), we employ similar statistical argument as in [PS00, TZ23, HKL19]. Essentially, we show that the view of $\mathcal{A}$ is identical between the game using $\boldsymbol{w} = (w_0, w_1)$ or $\boldsymbol{w}' = (w_0 - \mathsf{dlog}_G H, w_1 + 1)$ to simulate the Σ_{os} part of the protocol. This allows us to argue that for any (even unbounded) adversary $\mathcal{A}$, the witness $\boldsymbol{w}'$ to W that it finds will be distinct from the one the game uses at least half of the time.

  The above arguments allow us to argue that upon rewinding, the extracted witness $\boldsymbol{w}'$ is distinct from the witness $\boldsymbol{w}$ employed to simulate the signing oracles. Therefore, we can recompute the discrete logarithm of H via $\boldsymbol{w}$ and $\boldsymbol{w}'$ which breaks DL. (Note that $\mathsf{dlog}_G(H)$ is not required for simulation.)

Case 2: Extracted $\overline{A}_{\mathsf{os}}$ are distinct. Our goal in this case is to reduce to the UF-RMA security of $\mathsf{M}_{\mathsf{alg}}$. The proof is more complex than sketched in Section 1.1 due to partial blindness. We elaborate; the proof goes through the following steps.

- Set up H while knowing $h = \mathsf{dlog}_G H$ and compute Okamoto-Schnorr nonce while knowing $\mathsf{dlog}_G A_{\mathsf{os}} = r_{\mathsf{os},0} + r_{\mathsf{os},1} h$.
- Simulate the MAC tags for sessions that do not sign the guessed common message info^* (identified with the initially guessed H_{pb}-query index). In particular, compute the tag as $\sigma_{\mathsf{mac}} = (U = uF, V = uZ, S \leftarrow\!\!\$\, \mathbb{G}, T \leftarrow\!\!\$\, \mathbb{G})$. This change is due to an observation that $(U, x_0 U, y_0 U)$ (where $x_0 U, y_0 U$ are computed in the computation of S, T, respectively) is a DDH tuple with respect to the base $(G, x_0 G, y_0 G)$. To give the reduction to DDH (where x_0, y_0 are not known), we have to (1) generate $(D_2, D_3) = \mathsf{H}(\mathsf{info})$ with $(D_2, D_3) = a(G, D_1)$ for some $a \in \mathbb{Z}_p$, and (2) simulate Σ_{mac} and honestly compute Σ_{dleq} for the signing using the witness a. (Note that this is done to all sessions.)
- To enforce the structure check on $\widetilde{\sigma}_{\mathsf{mac}}$ in the following bullet point, we have to switch $\mathsf{H}(\mathsf{info}^*)$ back to uniformly random, and accordingly honestly compute Σ_{mac} while simulating Σ_{dleq} during the signing of info^*. (Note that the simulation of signing $\mathsf{info} \neq \mathsf{info}^*$ is still identical to the previous step.)
- The game then enforces a structure check based on the linear map $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{X}_{\mathsf{mac}}}$ (Eq. 2) on the MAC tags $\widetilde{\sigma}_{\mathsf{mac}} = (\widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{T}_0, \widetilde{S}_1, \widetilde{T}_1)$ and $\widetilde{M}$ from $\mathcal{A}$'s signature outputs using msk. This is possible through soundness of the OR-proof $\Sigma_{\mathsf{mac}} \vee \Sigma_{\mathsf{dleq}}$, since $(D_2^*, D_3^*) = \mathsf{H}(\mathsf{info}^*)$ is still uniformly random so $(D_1, D_2^*, D_3^*) \notin \mathfrak{L}_{\mathsf{dleq}}$. In particular, if $(\mathsf{ipp}, \widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}) \notin \mathfrak{L}_{\mathsf{mac}}$, the adversary can only output a valid $\Sigma_{\mathsf{mac}} \vee \Sigma_{\mathsf{dleq}}$ proof with negligible probability.
- The proof now (1) switches $\mathsf{H}(\mathsf{info}^*)$ to $(D_2^*, D_3^*) = a^*(G, D_1)$ for $a \in \mathbb{Z}_p$ and (2) simulates *all signing sessions* by honestly running Σ_{dleq} with the witness and simulating Σ_{mac}. Note that we have to maintain

[5] In the formal proof in Section 6, we introduce this step much earlier (before the extraction), but note that it is only required in this case.

the structure checks introduced in the previous steps, and more importantly they have to be efficient due to a DDH reduction to switch $\mathsf{H}(\mathsf{info}^*)$. This step is the reason why we need to commit to the blinding factors with base $\widetilde{U}$ as naively checking (Eq. 2) requires knowing $\mathsf{dlog}_G \widetilde{U}$. The efficient check using msk is:

$$\widetilde{V} = z\widetilde{U} \ , \qquad \widetilde{S}_0 + \overline{R}_x = x_0\widetilde{U} + x_1 C_M \ , \qquad \widetilde{T}_0 + \overline{R}_y = y_0\widetilde{U} + y_1 C_M \ ,$$

where $C_M = (\mathsf{dlog}_G \widetilde{U}) \cdot \overline{A}_{\mathsf{os}}$ is efficiently computable from the transcript $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}}), h, \boldsymbol{w} = (w_0, w_1)$ and the extracted $\overline{A}_0, \overline{A}_1, \overline{\Gamma}_{\mathsf{os}}$. Intuitively, this check ensures that $(\widetilde{U}, \widetilde{V}, \widetilde{S}_0 + \overline{R}_x, \widetilde{T}_0 + \overline{R}_y)$ is a valid tag for $\overline{A}_{\mathsf{os}}$. For more detail, we refer to the proof in Section 6.3.

- Finally, we prepare for the rewinding by generating all the A_{os} nonces and their corresponding tags in advance so as to avoid tagging additional nonces during rewinding (as alluded to in Section 1.1). Then, we use the UF-RMA oracle to generate these prepared nonces and tags. At the end, we rewind $\mathcal{A}$ to extract the blinding factor to obtain $m^* = \mathsf{dlog}_G \overline{A}_{\mathsf{os}}$. We also recompute the unblinded tag σ^*_{mac} for m^* from the blinding factors and $\widetilde{\sigma}_{\mathsf{mac}}$. Together, m^* and σ^*_{mac} constitute a UF-RMA forgery (cf. Remark 3.2). □

We show that $\mathsf{BS}_{\mathrm{alg}}$ is partially blind in the random oracle model if $\mathsf{C}_{\mathsf{rnd}}$ is hiding. Note that if $\mathsf{C}_{\mathsf{rnd}}$ is perfectly hiding, then $\mathsf{BS}_{\mathrm{alg}}$ satisfies *statistical* partial blindness.

Theorem 3.6 (Partial Blindness). *Let* $(\mathsf{GGen}, p, \mathsf{C}_{\mathsf{mac}}, \mathsf{C}_{\mathsf{rnd}}, \mathsf{BS}_{\mathrm{alg}}, \mathsf{M}_{\mathsf{alg}})$ *be as in Theorem 3.4, and* $\mathsf{H}, \mathsf{H}_{\mathsf{pb}}$ *be random oracles. Assume that* $\mathsf{C}_{\mathsf{mac}}$ *is perfectly binding. For any adversary* $\mathcal{A}$ *running in time* $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$*, making at most* $Q_{\mathsf{H}} = Q_{\mathsf{H}}(\lambda)$ *and* $Q_{\mathsf{pb}} = Q_{\mathsf{pb}}(\lambda)$ *queries to oracles* H *and* H_{pb}*, respectively, there exists an adversary* $\mathcal{B}$ *running in time roughly* $T_{\mathcal{A}}$*, such that*

$$\mathsf{Adv}^{\mathsf{rnd}}_{\mathsf{BS}_{\mathrm{alg}}}(\mathcal{A}, \lambda) \leqslant \frac{2Q_{\mathsf{H}}}{p^3} + \frac{Q_{\mathsf{pb}} + 2}{p} + 2 \cdot \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{rnd}}}(\mathcal{B}, \lambda).$$

Proof (Sketch). We employ the framework of randomizable transcripts for canonical Σ-protocols from [KR25]. Roughly, we go through the following hybrids, starting from the original blindness game.

- We simulate the Σ_{rnd} transcript $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ as usual: the game samples a random challenge $\widetilde{c}_{\mathsf{rnd}}$, uses the SHVZK simulator to simulate the nonce-response pair $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ for $\widetilde{c}_{\mathsf{rnd}}$, and programs the random oracle H in the third slot with challenge $\widetilde{c}_{\mathsf{rnd}}$ accordingly.
- We simulate the transcript $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ and program H in the first slot with $\widetilde{c}_{\mathsf{os}}$, as above. Prior, the transcript was obtained by randomizing the (potentially maliciously-generated) transcript $(A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ via the randomization algorithm from Lemma 2.17. As the transcript $(A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ must verify due to the user-side check, the randomization algorithm and Corollary 2.16 ensure that the distribution of $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ remains identical.
- The transcripts of *both* branches of the OR-proof (i.e., $(\widetilde{A}_{\mathsf{mac}}, \widetilde{c}_{\mathsf{mac}}, \widetilde{z}_{\mathsf{mac}})$ and $(\widetilde{A}_{\mathsf{dleq}}, \widetilde{c}_{\mathsf{dleq}}, \widetilde{z}_{\mathsf{dleq}})$) are simulated and $\widetilde{c}_{\mathsf{or}} = \widetilde{c}_{\mathsf{mac}} + \widetilde{c}_{\mathsf{dleq}}$ is programmed into the second slot of H. The distribution does not change with a similar argument to the above. Here, it is important that the modifications to the Σ_{mac} transcript $(A_{\mathsf{mac}}, c_{\mathsf{mac}}, z_{\mathsf{mac}})$, due to the blinding of $\widetilde{\sigma}_{\mathsf{mac}}$, retain validity of the transcripts. (The latter is shown in the formal proof of Theorem 3.4.)
- We commit to the zero vector over $\mathbb{G}^7$ in $\mathrm{cmt}_{\mathsf{rnd}}$. As the Σ_{rnd} transcript is simulated, we can reduce to hiding of the commitment scheme $\mathsf{C}_{\mathsf{rnd}}$.

At this point, we do not require any computational assumptions anymore; the following modifications are inefficient but statistically indistinguishable.

- The game aborts if the adversarially-issued tag σ_{mac} does not follow the structure induced by $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}$. To perform the (inefficient) check, the game brute-forces the secret key $\mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho_0)$ for $\mathsf{M}_{\mathsf{alg}}$ (which is unique by perfect binding of $\mathsf{C}_{\mathsf{mac}}$) and verifies that the MAC has the desired structure. For this step, it is important to note that the transcript $(A_{\mathsf{mac}}, c_{\mathsf{mac}}, z_{\mathsf{mac}})$ functions as interactive proof that σ_{mac} has the desired structure. Then, the claim follows from 2-special soundness of both Σ_{mac} and Σ_{dleq}, as $(D_1, \mathsf{H}(\mathsf{info}))$ is not a DDH-tuple except with negligible probability.

- Now, we compute the blinded MAC $\widetilde{\sigma}_{\mathsf{mac}}$ and $\widetilde{M}$ as follows

$$\begin{aligned} u &\leftarrow\!\!\$\ \mathbb{Z}_p^* \ , & \widetilde{M} &\leftarrow\!\!\$\ \mathbb{G} \ , \\ \widetilde{U} &\leftarrow uG \ , & \widetilde{V} &\leftarrow uzG \ , \\ \widetilde{S}_1 &\leftarrow\!\!\$\ \mathbb{G} \ , & \widetilde{S}_0 &\leftarrow u(x_0G + x_1\widetilde{M} - \widetilde{S}_1) \ , \\ \widetilde{T}_1 &\leftarrow\!\!\$\ \mathbb{G} \ , & \widetilde{T}_0 &\leftarrow u(y_0G + y_1\widetilde{M} - \widetilde{T}_1) \ . \end{aligned}$$

  Due to the above abort condition, the distribution of the blinded MAC remains identical.

Note that at this point, the view of the adversary is independent from the secret bit. This concludes the proof. □

Remark 3.7 (ROM-free blindness). We still program the random oracle H in our blindness proof. Indeed, for an arbitrary hash function H, the scheme is not statistically blind. For instance, an inefficient adversary may choose info such that $(D_1, \mathsf{H}(\mathsf{info}))$ is a DDH-tuple and then issue a malformed tag σ_{mac} in one session, and a valid tag in the other. Indeed, the blinded outputs will retain the structure of the tags, leading to a trivial (but inefficient) attack.

This can be remedied as follows. We let the signer issue an additional interactive proof that σ_{mac} is well-formed with an info-independent OR-branch, e.g., a fixed (non-)DDH-tuple $\boldsymbol{D} \leftarrow\!\!\$\ \mathbb{G}^3$. This allows us to simulate the interactive proof in the OMUF analysis, plus it is statistically hard to issue malformed MACs. Also, for the Σ_{rnd} proof, we can employ witness-indistinguishability if the commitment scheme is statistically hiding, which avoids programming the oracle H. We note that there are still some steps that rely on programming H, but these seem to be proof artifacts. Indeed, it is likely that a direct analysis of the distributions of blinded transcripts gives statistical blindness outside the ROM, akin to the analysis in [PS00, TZ22, CATZ24].

4 Instantiations

In this section, we present our two instantiations for compact signatures and statistical blindness (in the ROM).

4.1 Compact Signatures

Commitment schemes for compact signatures. For our compact instantiation that achieves computational blindness, we instantiate $\mathsf{C}_{\mathsf{rnd}}$ with a multi-input ElGamal commitment, denoted $\mathsf{C}_{\mathsf{elg}}$, with message space $\mathbb{G}^7$. Similarly, we instantiate $\mathsf{C}_{\mathsf{mac}}$ with $\mathsf{C}_{\mathsf{elg}}$ for $\mathbb{Z}_p^2$ messages as well (the commitment commits to $(a, b) \in \mathbb{Z}_p^2$ by committing to $(aG, bG) \in \mathbb{G}^2$). We present in Figure 7 the scheme with general message space $\mathbb{G}^\ell$. Note that the properties of ElGamal encryption implies additive homomorphism, perfectly strong binding, extractability, and hiding (with the concrete bound implied from [BBKS07]) of $\mathsf{C}_{\mathsf{elg}}$.

Lemma 4.1. *Let* GGen *be a group parameter generator for groups of prime-order* $p = p(\lambda)$*. The commitment scheme* $\mathsf{C}_{\mathsf{elg}} = \mathsf{C}_{\mathsf{elg}}[\mathsf{GGen}]$ *as given in Figure 7 satisfies:*

1. *Linear-homomorphism.*
2. *Perfect strong binding.*
3. *Extraction with the extraction key identically distributed to* $\mathsf{C}_{\mathsf{elg}}.\mathsf{Setup}$.
4. *Computational hiding from DDH: For any adversary* $\mathcal{A}$ *running in time* $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$*, there exists an adversary* $\mathcal{B}$ *running in time* $\approx T_{\mathcal{A}}$ *such that* $\mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{elg}}}(\mathcal{A}, \lambda) \leqslant 2 \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}, \lambda) + \frac{1}{2^{\lambda-2}}$.

Security. The following corollary follows from Theorems 3.1 and 3.4 to 3.6 and lemma 4.1 along with DDH random self-reducibility (e.g., [EHK+13]).

Corollary 4.2 (Security of $\mathsf{BS}_{\mathrm{alg}}$ with $\mathsf{C}_{\mathsf{elg}}$.). *Let* $\mathsf{BS}_{\mathsf{elg}} = \mathsf{BS}_{\mathrm{alg}}[\mathsf{GGen}, \mathsf{C}_{\mathsf{elg}}, \mathsf{C}_{\mathsf{elg}}, \mathsf{H}, \mathsf{H}_{\mathsf{pb}}]$*. Then,* $\mathsf{BS}_{\mathsf{elg}}$ *is perfectly correct, one-more strongly unforgeable, and computationally partially blind, under the DDH assumption.*

Algorithm $\mathsf{C}_{\mathsf{elg}}.\mathsf{Setup}(1^\lambda)$:	Algorithm $\mathsf{C}_{\mathsf{elg}}.\mathsf{ExtSetup}(1^\lambda)$:
$(\mathbb{G}, p, G) \leftarrow\!\$\ \mathsf{GGen}(1^\lambda)$	$(\mathbb{G}, p, G) \leftarrow\!\$\ \mathsf{GGen}(1^\lambda)$
$\mathsf{pk} \leftarrow\!\$\ \mathbb{G}^\ell$	$\mathsf{sk} \leftarrow\!\$\ \mathbb{Z}_p^\ell$; $\mathsf{pk} \leftarrow \mathsf{sk} \cdot G$
$\mathsf{ck} \leftarrow (\mathbb{G}, p, G, \mathsf{pk})$	$\mathsf{ck} \leftarrow (\mathbb{G}, p, G, \mathsf{pk})$
return ck	**return** $(\mathsf{ck}, \mathsf{td} = \mathsf{sk})$
Algorithm $\mathsf{C}_{\mathsf{elg}}.\mathsf{Commit}(\mathsf{ck}, \boldsymbol{M} \in \mathbb{G}^\ell; \rho \in \mathbb{Z}_p)$:	Algorithm $\mathsf{C}_{\mathsf{elg}}.\mathsf{Extract}(\mathsf{td}, \mathrm{cmt} \in \mathbb{G}^{\ell+1})$:
return $(\rho G, \boldsymbol{M} + \rho \cdot \mathsf{pk})$	**return** $\mathrm{cmt}_{[1:\ell]} - \mathsf{sk} \cdot \mathrm{cmt}_0$

Fig. 7. Multi-input ElGamal commitment for $\mathsf{C}_{\mathsf{elg}}.\mathcal{M} = \mathbb{G}^\ell$ along with its extraction algorithms $\mathsf{Ext} = (\mathsf{C}_{\mathsf{elg}}.\mathsf{ExtSetup}, \mathsf{C}_{\mathsf{elg}}.\mathsf{Extract})$

Algorithm $\mathsf{C}_{\mathsf{gs}}.\mathsf{Setup}(1^\lambda)$:	Algorithm $\mathsf{C}_{\mathsf{gs}}.\mathsf{ExtSetup}(1^\lambda)$:
$(\mathbb{G}, p, G) \leftarrow\!\$\ \mathsf{GGen}(1^\lambda)$; $(A, B, C) \leftarrow\!\$\ \mathbb{G}^3$	$(\mathbb{G}, p, G) \leftarrow\!\$\ \mathsf{GGen}(1^\lambda)$; $A \leftarrow\!\$\ \mathbb{G}$; $s \leftarrow\!\$\ \mathbb{Z}_p$
$\mathbf{D} \leftarrow \begin{pmatrix} G & A \\ B & C \end{pmatrix}$	$\mathbf{D} \leftarrow \begin{pmatrix} G & A \\ sG & sA \end{pmatrix}$
$\mathsf{ck} \leftarrow (\mathbb{G}, p, G, \mathbf{D})$	$\mathsf{ck} \leftarrow (\mathbb{G}, p, G, \mathbf{D})$
return ck	**return** $(\mathsf{ck}, \mathsf{td} = s)$
Algorithm $\mathsf{C}_{\mathsf{gs}}.\mathsf{Commit}(\mathsf{ck}, M \in \mathbb{G}^\ell; (\rho_i \in \mathbb{Z}_p^2)_{i\in[\ell]})$:	Algorithm $\mathsf{C}_{\mathsf{gs}}.\mathsf{Extract}(\mathsf{td}, (\mathrm{cmt}_i \in \mathbb{G}^2)_{i\in[\ell]})$:
return $((0, M_i) + \mathbf{D} \cdot \rho_i)_{i\in[\ell]}$	**return** $(\mathrm{cmt}_{i,1} - s \cdot \mathrm{cmt}_{i,0})_{i\in[\ell]}$

Fig. 8. Groth-Sahai commitment for message space $\mathsf{C}_{\mathsf{gs}}.\mathcal{M} = \mathbb{G}^\ell$ along with its extraction algorithms $\mathsf{Ext} = (\mathsf{C}_{\mathsf{gs}}.\mathsf{ExtSetup}, \mathsf{C}_{\mathsf{gs}}.\mathsf{Extract})$.

EFFICIENCY. The first move of the signing protocol incurs $1\mathbb{G}$ for A_{os}, $10\mathbb{G}$ for A_{mac}, $2\mathbb{G}$ for A_{dleq}, and $4\mathbb{G}$ for σ_{mac}. The second move only requires $2\mathbb{Z}_p$ elements. The third move requires $2\mathbb{Z}_p$ for z_{os}, $8\mathbb{Z}_p$ for $c_{\mathsf{mac}}, z_{\mathsf{mac}}$, and $2\mathbb{Z}_p$ for $c_{\mathsf{dleq}}, z_{\mathsf{dleq}}$. This counts to $17\mathbb{G} + 14\mathbb{Z}_p$ of communication. The signature (ignoring the noncesas they can be reconstructed) contains $7\mathbb{G}$ elements for $(\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}})$, $8\mathbb{G}$ for $\mathrm{cmt}_{\mathsf{rnd}}$, $14\mathbb{Z}_p$ for $(\widetilde{c}_{\mathsf{xxx}}, \widetilde{z}_{\mathsf{xxx}})_{\mathsf{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq}\}}$, and $8\mathbb{Z}_p$ for $(\widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$. In total, our signature size is $15\mathbb{G} + 22\mathbb{Z}_p$.

4.2 Statistical Blindness

COMMITMENT SCHEMES FOR STATISTICAL BLINDNESS. We use Groth-Sahai commitments [GS08], denoted C_{gs}, to individually commit to each group element as described in Figure 8. For $\mathsf{C}_{\mathsf{mac}}$, we use $\mathsf{C}_{\mathsf{elg}}$ for $\mathbb{Z}_p^2$ messages as in the prior construction, since the hiding property of $\mathsf{C}_{\mathsf{elg}}$ does not affect blindness. The following lemma establishes the properties for C_{gs}.

Lemma 4.3. *Let* GGen *be a group parameter generator for groups of prime-order* $p = p(\lambda)$*. The commitment scheme* $\mathsf{C}_{\mathsf{gs}} = \mathsf{C}_{\mathsf{gs}}[\mathsf{GGen}]$ *as given in Figure 8 satisfies:*

1. *Linear-homomorphism.*
2. *Computational strong binding from DL: For any adversary* $\mathcal{A}$ *running in time* $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$*, there exists an adversary* $\mathcal{B}$ *running in time* $\approx T_{\mathcal{A}}$ *such that* $\mathsf{Adv}^{\mathsf{sbind}}_{\mathsf{C}_{\mathsf{gs}}}(\mathcal{A}, \lambda) \leqslant \mathsf{Adv}^{\mathsf{dlog}}_{\mathsf{GGen}}(\mathcal{B}, \lambda)$.
3. *Extraction with extraction key computationally indistinguishability: For any adversary* $\mathcal{A}$ *running in time* $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$*, there exists an adversary* $\mathcal{B}$ *running in time* $\approx T_{\mathcal{A}}$ *such that* $\mathsf{Adv}^{\mathsf{ext\text{-}dist}}_{\mathsf{C}_{\mathsf{gs}},\mathsf{Ext}}(\mathcal{A}, \lambda) \leqslant \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}, \lambda)$.
4. *Statistical hiding: For any* $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{gs}}}(\mathcal{A}, \lambda) \leqslant \frac{1}{p}$.

Proof. We show the lemma for messages of length $\ell = 1$, since the scheme for messages of length $\ell > 1$ simply commits to each element independently. Note that additive homomorphism, extraction, and statistical hiding is already implied by [GS08] adapted to pairing-free settings. Note that due to $\mathsf{C}_{\mathsf{gs}}.\mathsf{Setup}$ generating ck uniformly, there is a $1/p$ probability that $(B, C) = b(G, A)$ for some $b \in \mathbb{Z}_p$, which makes ck not perfectly hiding constituting to the $1/p$ advantage.

Strong binding follows from observing that if two openings $(M, \rho = (\rho_0, \rho_1)) \neq (M', \rho' = (\rho'_0, \rho'_1))$ are valid for some $\mathsf{cmt} \in \mathbb{G}^2$. Then, we have that

$$0^2_{\mathbb{G}} = \mathsf{C_{gs}.Commit}(\mathsf{ck}, M - M'; \rho - \rho') = \begin{pmatrix} 0 \\ M - M' \end{pmatrix} + \begin{pmatrix} G & A \\ B & C \end{pmatrix} \begin{pmatrix} \rho_0 - \rho'_0 \\ \rho_1 - \rho'_1 \end{pmatrix} .$$

Observe that there are two cases: (1) $\rho \neq \rho'$, meaning $(\rho_0 - \rho'_0)G + (\rho_1 - \rho'_1)A = 0$ so $\mathsf{dlog}_G A = (\rho_0 - \rho'_0)(\rho_1 - \rho'_1)^{-1}$ can be computed. (2) $\rho = \rho'$ and $M \neq M'$, which cannot occur, since it implies $M = M'$. Thus, a simple reduction can be computed. □

SECURITY. The following corollary follows from Theorems 3.1 and 3.4 to 3.6 and lemma 4.3 along with DDH random self-reducibility (e.g., [EHK+13]).

Corollary 4.4 (Security of $\mathsf{BS_{alg}}$ with $\mathsf{C_{gs}}$.). *Let* $\mathsf{BS_{stat}} = \mathsf{BS_{alg}}[\mathsf{GGen}, \mathsf{C_{mac}} = \mathsf{C_{elg}}, \mathsf{C_{rnd}} = \mathsf{C_{gs}}, \mathsf{H}, \mathsf{H_{pb}}]$. *Then,* $\mathsf{BS_{stat}}$ *is perfectly correct, one-more strongly unforgeable under the DDH assumption, and statistically partially blind.*

EFFICIENCY. The signing protocol is identical to the compact instantiation with $17\mathbb{G} + 14\mathbb{Z}_p$ communication, since $\mathsf{C_{rnd}}$ only affects the signature. The signature contains $7\mathbb{G}$ elements for $(\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}})$, $14\mathbb{Z}_p$ for $(\widetilde{c}_{\mathsf{xxx}}, \widetilde{z}_{\mathsf{xxx}})_{\mathsf{xxx} \in \{\mathsf{os},\mathsf{mac},\mathsf{dleq}\}}$ (similar to before), and $14\mathbb{G}$ for $\mathsf{cmt_{rnd}}$ (committing to $7\mathbb{G}$ elements) and $21\mathbb{Z}_p$ for $(\widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$. In total, our signature size is $21\mathbb{G} + 35\mathbb{Z}_p$.

5 Correctness Proof of $\mathsf{BS_{alg}}$

We show Theorem 3.4, i.e., that $\mathsf{BS_{alg}}$ is correct. Recall that $\mathsf{mpp} = (F, \mathsf{mck})$, $\mathsf{pp} = (\mathbb{G}, p, G, H, \mathsf{mpp}, \mathsf{bck}, \mathbb{x}_{\mathsf{dleq}})$, $\mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho_0)$ and $\mathsf{ipp} = (\mathrm{cmt}_0, X_1, Y_1, Z)$. Denote by

$$\sigma = (\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, (\widetilde{A}_{\mathsf{xxx}}, \widetilde{c}_{\mathsf{xxx}}, \widetilde{z}_{\mathsf{xxx}})_{\mathsf{xxx} \in \{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$$

the output of a honest signing session. We need to show that

- $\widetilde{U} \neq 0_{\mathbb{G}}$,
- $(\widetilde{c}_{\mathsf{os}}, \widetilde{c}_{\mathsf{mac}} + \widetilde{c}_{\mathsf{dleq}}, \widetilde{c}_{\mathsf{rnd}}) = \mathsf{H}(\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{A}_{\mathsf{os}}, \widetilde{A}_{\mathsf{mac}}, \widetilde{A}_{\mathsf{dleq}}, \widetilde{A}_{\mathsf{rnd}}, \mu, \mathsf{info})$, and
- the Fiat-Shamir proofs verify.

We have $u, \beta \leftarrow\!\!\$\ \mathbb{Z}_p^*$ and $\widetilde{U} = \beta U = (\beta u)G$. Thus $\widetilde{U} \neq 0_{\mathbb{G}}$. Also, $c_{\mathsf{or}} = c_{\mathsf{dleq}} + c_{\mathsf{mac}}$ as chosen in $\mathsf{BS_{alg}.Sign_2}$, therefore we have as desired that

$$\widetilde{c}_{\mathsf{mac}} + \widetilde{c}_{\mathsf{dleq}} = c_{\mathsf{mac}} + \gamma_{\mathsf{mac}} + c_{\mathsf{dleq}} + \gamma_{\mathsf{dleq}} = c_{\mathsf{or}} + \gamma_{\mathsf{mac}} + \gamma_{\mathsf{dleq}} = \widetilde{c}_{\mathsf{or}}.$$

Below, we focus on the Fiat-Shamir proofs $\widetilde{\tau}_{\mathsf{xxx}} := (\widetilde{A}_{\mathsf{xxx}}, \widetilde{c}_{\mathsf{xxx}}, \widetilde{z}_{\mathsf{xxx}})$ for statements $\mathbb{x}_{\mathsf{xxx}}$, where $\mathsf{xxx} \in \{\mathsf{os}, \mathsf{mac}, \mathsf{dleq}, \mathsf{rnd}\}$. The transcripts $\widetilde{\tau}_{\mathsf{xxx}}$ are computed based on Σ-protocol transcripts $\tau_{\mathsf{xxx}} = (A_{\mathsf{xxx}}, c_{\mathsf{xxx}}, z_{\mathsf{xxx}})$ issued by the signer as follows:

OKAMOTO-SCHNORR. The signer sets $\mathbb{x}_{\mathsf{os}} = W$ and $\mathbb{w}_{\mathsf{os}} = \boldsymbol{w}$ in $\mathsf{BS_{alg}.KeyGen}$. Note that $\phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\mathbb{w}_{\mathsf{os}}) = W$ by design. The signer sets $(A_{\mathsf{os}}, \mathsf{st}_{\mathsf{os}}) \leftarrow\!\!\$\ \Sigma_{\mathsf{os}}.\mathsf{Init}(\mathsf{pp}, W, \boldsymbol{w})$ in $\mathsf{BS_{alg}.Sign_1}$. Next, in $\mathsf{BS_{alg}.Sign_2}$, the signer sets $z_{\mathsf{os}} \leftarrow \Sigma_{\mathsf{os}}.\mathsf{Resp}(\mathsf{st}_{\mathsf{os}}, c_{\mathsf{os}})$ with $c_{\mathsf{os}} \in \mathbb{Z}_p$ supplied by the user. By correctness of Σ_{os}, we have that $(A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ is valid for $\mathbb{x}_{\mathsf{os}}$. The user derives the transcript $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ from $(A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ via Rand as specified in Lemma 2.17. Therefore, also $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ is valid for $\mathbb{x}_{\mathsf{os}}$ by Definition 2.15.

DLOG EQUALITY. The signer simulates the transcript $(A_{\mathsf{dleq}}, c_{\mathsf{dleq}}, z_{\mathsf{dleq}})$ for statement $\mathbb{x}_{\mathsf{dleq}} = (D_1, D_2, D_3)$, where c_{dleq} is provided by the user. Thus, $(A_{\mathsf{dleq}}, c_{\mathsf{dleq}}, z_{\mathsf{dleq}})$ is valid for statement $\mathbb{x}_{\mathsf{dleq}}$. As above, the user derives the transcript $(\widetilde{A}_{\mathsf{dleq}}, \widetilde{c}_{\mathsf{dleq}}, \widetilde{z}_{\mathsf{dleq}})$ from $(A_{\mathsf{dleq}}, c_{\mathsf{dleq}}, z_{\mathsf{dleq}})$ via Rand from Lemma 2.17. In particular, the transcript $(\widetilde{A}_{\mathsf{dleq}}, \widetilde{c}_{\mathsf{dleq}}, \widetilde{z}_{\mathsf{dleq}})$ remains valid for $\mathbb{x}_{\mathsf{dleq}}$ as Rand satisfies Definition 2.15.

MAC VALIDITY. The signer sets $\mathbb{x}_{\mathsf{mac}} \leftarrow (\mathsf{ipp}, A_{\mathsf{os}}, U, V, S, T, 0_{\mathbb{G}}, 0_{\mathbb{G}})$ and $\mathbb{w}_{\mathsf{mac}} \leftarrow (u^{-1}, \mathsf{msk})$, where $\mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho_0)$. The transcript τ_{mac} is computed via $(A_{\mathsf{mac}}, \mathsf{st}_{\mathsf{mac}}) \leftarrow\!\!\$\ \Sigma_{\mathsf{mac}}.\mathsf{Init}(\mathsf{pp}, \mathbb{x}_{\mathsf{mac}}, \mathbb{w}_{\mathsf{mac}})$ and $z_{\mathsf{mac}} \leftarrow \Sigma_{\mathsf{mac}}.\mathsf{Resp}(\mathsf{st}_{\mathsf{mac}}, c_{\mathsf{mac}})$, where $c_{\mathsf{mac}} \in \mathbb{Z}_p$. Observe that $u^{-1} \cdot U = G$ and as $S = u(x_0 G + x_1 M)$, it holds that $x_0 G + x_1 M = u^{-1} S$ (similarly for $y_0 G + y_1 M = u^{-1} T$). In addition, $(\mathsf{msk}, \mathsf{ipp})$ is setup honestly by the signer in $\mathsf{BS}_{\mathrm{alg}}.\mathsf{KeyGen}$, therefore we have $\phi^{\mathsf{mac}}_{\mathsf{mpp},\mathbb{x}_{\mathsf{mac}}}(\mathbb{w}_{\mathsf{mac}}) = (\mathrm{cmt}_0, X_1, Y_1, Z, G, V, 0_{\mathbb{G}}, 0_{\mathbb{G}})^{\mathsf{T}}$ (cf. Equation (2)). By correctness of Σ_{mac}, we obtain

$$\phi^{\mathsf{mac}}_{\mathsf{mpp},\mathbb{x}_{\mathsf{mac}}}(z_{\mathsf{mac}}) = A_{\mathsf{mac}} + c_{\mathsf{mac}} \cdot (\mathrm{cmt}_0, X_1, Y_1, Z, G, V, 0_{\mathbb{G}}, 0_{\mathbb{G}})^{\mathsf{T}}.$$

Note that $\beta \leftarrow\!\!\$\ \mathbb{Z}_p^*, r_x, r_y, r_M \leftarrow\!\!\$\ \mathbb{Z}_p$ and the MAC randomized as described in Section 3.1 by the user. That is, we have $\widetilde{U} = \beta U$, $\widetilde{V} = \beta V$, $\widetilde{M} = A_{\mathsf{os}} + r_M F$, and

$$\widetilde{S}_0 \leftarrow \beta S - r_x \widetilde{U}\ , \widetilde{S}_1 \leftarrow r_x G + r_M X_1\ ,$$
$$\widetilde{T}_0 \leftarrow \beta T - r_y \widetilde{U}\ , \widetilde{T}_1 \leftarrow r_y G + r_M Y_1\ .$$

Following the notation in Figure 6, we next show that $(A'_{\mathsf{mac}}, c, \beta^{-1} z_{\mathsf{mac}}[u], z_{\mathsf{mac}}[z, x_0, x_1, y_0, y_1, \rho_0])$ is a valid transcript for statement $\mathbb{x}'_{\mathsf{mac}} = (\mathsf{ipp}, \widetilde{M}, \widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{T}_0, \widetilde{S}_1, \widetilde{T}_1)$ below, where $A'_{\mathsf{mac}} = (A_{\mathsf{mac}}[\mathrm{cmt}_0, X_1, Y_1, Z, G], \beta A_{\mathsf{mac}}[V], A_{\mathsf{mac}}[S_1] + r_M A_{\mathsf{mac}}[X_1] + r_x A_{\mathsf{mac}}[G], A_{\mathsf{mac}}[T_1] + r_M A_{\mathsf{mac}}[Y_1] + r_y A_{\mathsf{mac}}[G])$.

$$\begin{aligned}
&\phi^{\mathsf{mac}}_{\mathsf{mpp},\mathbb{x}'_{\mathsf{mac}}}(\beta^{-1} z_{\mathsf{mac}}[u], z_{\mathsf{mac}}[z, x_0, x_1, y_0, y_1, \rho_0]) \\
&= \begin{pmatrix} \mathsf{C}_{\mathsf{mac}}(\mathsf{mck}, (z_{\mathsf{mac}}[x_0], z_{\mathsf{mac}}[y_0]); z_{\mathsf{mac}}[\rho_0]) \\ z_{\mathsf{mac}}[x_1]F \\ z_{\mathsf{mac}}[y_1]F \\ z_{\mathsf{mac}}[z]F \\ \beta^{-1} z_{\mathsf{mac}}[u]\widetilde{U} \\ z_{\mathsf{mac}}[z]\widetilde{U} \\ z_{\mathsf{mac}}[x_0]G + z_{\mathsf{mac}}[x_1]\widetilde{M} - \beta^{-1} z_{\mathsf{mac}}[u]\widetilde{S}_0 \\ z_{\mathsf{mac}}[y_0]G + z_{\mathsf{mac}}[y_1]\widetilde{M} - \beta^{-1} z_{\mathsf{mac}}[u]\widetilde{T}_0 \end{pmatrix} \\
&= \begin{pmatrix} A_{\mathsf{mac}}[\mathrm{cmt}_0] + c_{\mathsf{mac}}\mathrm{cmt}_0 \\ A_{\mathsf{mac}}[X_1] + c_{\mathsf{mac}} X_1 \\ A_{\mathsf{mac}}[Y_1] + c_{\mathsf{mac}} Y_1 \\ A_{\mathsf{mac}}[Z] + c_{\mathsf{mac}} Z \\ \beta^{-1} z_{\mathsf{mac}}[u](\beta U) \\ z_{\mathsf{mac}}[z](\beta U) \\ z_{\mathsf{mac}}[x_0]G + z_{\mathsf{mac}}[x_1](A_{\mathsf{os}} + r_M F) - \beta^{-1} z_{\mathsf{mac}}[u](\beta S - r_x \widetilde{U}) \\ z_{\mathsf{mac}}[y_0]G + z_{\mathsf{mac}}[y_1](A_{\mathsf{os}} + r_M F) - \beta^{-1} z_{\mathsf{mac}}[u](\beta T - r_y \widetilde{U}) \end{pmatrix} \\
&= \begin{pmatrix} A_{\mathsf{mac}}[\mathrm{cmt}_0, X_1, Y_1, Z] + c_{\mathsf{mac}} \cdot (\mathrm{cmt}_0, X_1, Y_1, Z)^{\mathsf{T}} \\ z_{\mathsf{mac}}[u]U \\ \beta\,(z_{\mathsf{mac}}[z]U) \\ z_{\mathsf{mac}}[x_0]G + z_{\mathsf{mac}}[x_1]A_{\mathsf{os}} - z_{\mathsf{mac}}[u]S + r_M z_{\mathsf{mac}}[x_1]F + \beta^{-1} z_{\mathsf{mac}}[u] r_x \widetilde{U} \\ z_{\mathsf{mac}}[y_0]G + z_{\mathsf{mac}}[y_1]A_{\mathsf{os}} - z_{\mathsf{mac}}[u]T + r_M z_{\mathsf{mac}}[y_1]F + \beta^{-1} z_{\mathsf{mac}}[u] r_y \widetilde{U} \end{pmatrix} \\
&= \begin{pmatrix} A_{\mathsf{mac}}[\mathrm{cmt}_0, X_1, Y_1, Z] + c_{\mathsf{mac}} \cdot (\mathrm{cmt}_0, X_1, Y_1, Z)^{\mathsf{T}} \\ A_{\mathsf{mac}}[G] + c_{\mathsf{mac}} G \\ \beta\,(A_{\mathsf{mac}}[V] + c_{\mathsf{mac}} V) \\ A_{\mathsf{mac}}[S_1] + c_{\mathsf{mac}} 0_{\mathbb{G}} + r_M (A_{\mathsf{mac}}[X_1] + c_{\mathsf{mac}} X_1) + \beta^{-1} z_{\mathsf{mac}}[u] r_x \beta U \\ A_{\mathsf{mac}}[T_1] + c_{\mathsf{mac}} 0_{\mathbb{G}} + r_M (A_{\mathsf{mac}}[Y_1] + c_{\mathsf{mac}} Y_1) + \beta^{-1} z_{\mathsf{mac}}[u] r_y \beta U \end{pmatrix}
\end{aligned}$$

$$
\begin{aligned}
&= \begin{pmatrix} A'_{\mathsf{mac}}[\mathrm{cmt}_0, X_1, Y_1, Z, G] + c_{\mathsf{mac}} \cdot (\mathrm{cmt}_0, X_1, Y_1, Z, G)^{\mathsf{T}} \\ \beta\, A_{\mathsf{mac}}[V] + c_{\mathsf{mac}}\, \beta\, V \\ A_{\mathsf{mac}}[S_1] + r_M A_{\mathsf{mac}}[X_1] + c_{\mathsf{mac}} r_M X_1 + r_x(A_{\mathsf{mac}}[G] + c_{\mathsf{mac}} G) \\ A_{\mathsf{mac}}[T_1] + r_M A_{\mathsf{mac}}[Y_1] + c_{\mathsf{mac}} r_M Y_1 + r_y(A_{\mathsf{mac}}[G] + c_{\mathsf{mac}} G) \end{pmatrix} \\
&= \begin{pmatrix} A'_{\mathsf{mac}}[\mathrm{cmt}_0, X_1, Y_1, Z, G] + c_{\mathsf{mac}} \cdot (\mathrm{cmt}_0, X_1, Y_1, Z, G)^{\mathsf{T}} \\ A'_{\mathsf{mac}}[V] + c_{\mathsf{mac}} \widetilde{V} \\ A_{\mathsf{mac}}[S_1] + r_M A_{\mathsf{mac}}[X_1] + r_x(A_{\mathsf{mac}}[G]) + c_{\mathsf{mac}}(r_M X_1 + r_x G) \\ A_{\mathsf{mac}}[T_1] + r_M A_{\mathsf{mac}}[X_1] + r_y(A_{\mathsf{mac}}[G]) + c_{\mathsf{mac}}(r_M Y_1 + r_y G) \end{pmatrix} \\
&= \begin{pmatrix} A'_{\mathsf{mac}}[\mathrm{cmt}_0, X_1, Y_1, Z, G, V] + c_{\mathsf{mac}} \cdot (\mathrm{cmt}_0, X_1, Y_1, Z, G, \widetilde{V})^{\mathsf{T}} \\ A'_{\mathsf{mac}}[S_1] + c_{\mathsf{mac}} \widetilde{S}_1 \\ A'_{\mathsf{mac}}[T_1] + c_{\mathsf{mac}} \widetilde{T}_1 \end{pmatrix} \\
&= A'_{\mathsf{mac}} + c_{\mathsf{mac}} \cdot (\mathrm{cmt}_0, X_1, Y_1, Z, G, \widetilde{V}, \widetilde{S}_1, \widetilde{T}_1)^{\mathsf{T}}
\end{aligned}
$$

Note that we only used that $(A_{\mathsf{mac}}, c_{\mathsf{mac}}, z_{\mathsf{mac}})$ is a valid transcript which is useful for blindness. To conclude, observe that the user derives the transcript $(\widetilde{A}_{\mathsf{mac}}, \widetilde{c}_{\mathsf{mac}}, \widetilde{z}_{\mathsf{mac}})$ from $(A'_{\mathsf{mac}}, c_{\mathsf{mac}}, \beta^{-1} z_{\mathsf{mac}}[u], z_{\mathsf{mac}}[z, x_0, x_1, y_0, y_1, \rho_0])$ via Rand from Lemma 2.17. Therefore, the transcript $(\widetilde{A}_{\mathsf{mac}}, \widetilde{c}_{\mathsf{mac}}, \widetilde{z}_{\mathsf{mac}})$ remains valid for $\mathbb{x}_{\mathsf{dleq}}$ as Rand satisfies Definition 2.15.

MAC Blinding Factors The transcript $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ is computed by the user via $\mathbb{w}_{\mathsf{rnd}} = (\rho_{\mathsf{rnd}}, \alpha_0, \alpha_1, \gamma_{\mathsf{os}}, r_x, r_y, r_M, A_{\mathsf{os}})$ for $\mathbb{x}_{\mathsf{rnd}} \leftarrow (W, \widetilde{A}_{\mathsf{os}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{M}, \widetilde{U}, \widetilde{S}_1, \widetilde{T}_1)$. That is, the user sets $(\widetilde{A}_{\mathsf{rnd}}, \mathsf{st}_{\mathsf{rnd}}) \leftarrow\!\!\$\ \Sigma_{\mathsf{rnd}}.\mathsf{Init}(\mathsf{pp}, \mathbb{x}_{\mathsf{rnd}})$, derives $\mathsf{BlindChal}_{\mathsf{rnd}}$ from H and sets $\widetilde{z}_{\mathsf{rnd}} \leftarrow \Sigma_{\mathsf{rnd}}.\mathsf{Resp}(\mathsf{st}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}})$. It is straightforward to verify that $(\mathbb{x}_{\mathsf{rnd}}, \mathbb{w}_{\mathsf{rnd}}) \in \mathfrak{R}_{\mathsf{rnd}}$. Thus, $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ is a valid transcript for $\mathbb{x}_{\mathsf{rnd}}$ by correctness of Σ_{rnd}.

6 OMSUF Proof of $\mathsf{BS}_{\mathbf{alg}}$

We show that $\mathsf{BS}_{\mathrm{alg}}$ satisfies one-more strong unforgeability.

6.1 Proof of Theorem 3.5

Let $\mathcal{A}$ be an adversary against the OMSUF game of $\mathsf{BS}_{\mathrm{alg}}$, running in time $T_{\mathcal{A}}$, making at most $Q_{\mathrm{S}} = Q_{\mathrm{S}}(\lambda), Q_{\mathsf{H}} = Q_{\mathsf{H}}(\lambda)$, and $Q_{\mathsf{pb}} = Q_{\mathsf{pb}}(\lambda)$ queries to oracles S_1, H, and H_{pb}, respectively. Due to Lemma 2.5, we assume that it makes Q_{S} to oracle S_2 as well, i.e., completing all started sessions. Also, denote $Q_{\mathrm{S},\mathsf{info}}$ as the number of S_1 queries made with common message info ($Q_{\mathrm{S}} = \sum_{\mathsf{info}} Q_{\mathrm{S},\mathsf{info}}$). At the end of the game, $\mathcal{A}$ outputs its forgeries, namely $(\mathsf{info}^*, (\mu^{(i)}, \sigma^{(i)})_{i \in [Q_{\mathrm{S},\mathsf{info}^*}+1]})$. Note that we assume that $\mathcal{A}$ gives exactly $\ell = Q_{\mathrm{S},\mathsf{info}^*} + 1$ forgeries instead of $\ell > Q_{\mathrm{S},\mathsf{info}^*}$ for simplicity, and will use ℓ throughout the proof to refer to this value. For simplicity of the notations, we assume that the query count to the random oracles H and H_{pb} includes the query made by the game in the signing oracle and forgery verification. (This means that Q_{H} and Q_{pb} is at most $Q_{\mathrm{S}} + 1$ more than the number of queries actually made by $\mathcal{A}$.) For simplicity, we also assume that $\mathcal{A}$ does not make the same random oracle (RO) query twice.

Now, we establish the notations for messages sent throughout the game. For the j-th signing session (determined by the j-th query to S_1 with $j \in [Q_{\mathrm{S}}]$), we use subscript j to denote the messages being sent in this signing session (i.e., for any value x sent in the j-th session, we write x_j). Also, we parse the i-th ($i \in [\ell]$) forgery of $\mathcal{A}$ along with the corresponding statements as

$$
\begin{aligned}
\sigma^{(i)} &= (\widetilde{M}^{(i)}, \tilde{\sigma}^{(i)}_{\mathsf{mac}} = (\widetilde{U}^{(i)}, \widetilde{V}^{(i)}, \widetilde{S}^{(i)}_0, \widetilde{T}^{(i)}_0, \widetilde{S}^{(i)}_1, \widetilde{T}^{(i)}_1), \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, (\widetilde{A}^{(i)}_{\mathsf{xxx}}, \widetilde{c}^{(i)}_{\mathsf{xxx}}, \widetilde{z}^{(i)}_{\mathsf{xxx}})_{\mathsf{xxx} \in \{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}}) \ , \\
\mathbb{x}^{(i)}_{\mathsf{mac}} &= (\mathsf{ipp}, \widetilde{M}^{(i)}, \tilde{\sigma}^{(i)}_{\mathsf{mac}}) \ , \qquad \mathbb{x}^{(i)}_{\mathsf{rnd}} = (W, \widetilde{A}^{(i)}_{\mathsf{os}}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, \widetilde{M}^{(i)}, \widetilde{U}^{(i)}, \widetilde{S}^{(i)}_1, \widetilde{T}^{(i)}_1) \ ,
\end{aligned}
$$

with $\mathbb{x}_{\mathsf{os}} = W, \mathbb{x}_{\mathsf{dleq}} = (D_1, \mathsf{H}(\mathsf{info}^*))$. Then, we consider the following sequence of games, where we denote $\Pr[\mathbf{G}^{\mathcal{A}} \Rightarrow 1]$ (omitting the parameter λ) as the probability of $\mathcal{A}$ winning in game $\mathbf{G}$.

Game $\mathbf{G}_0$ (Original OMSUF Game): This game samples the secret key sk as in the scheme and runs the signing oracles honestly. The adversary $\mathcal{A}$ wins if all forgeries $(\mathsf{info}^*, (\mu^{(i)}, \sigma^{(i)})_{i\in[\ell]})$ verify and $(\mu^{(i)}, \sigma^{(i)})$ are all distinct. Note that

$$\mathsf{Adv}^{\mathrm{omsuf}}_{\mathsf{BS}_{\mathrm{alg}}}(\mathcal{A}, \lambda) = \mathsf{Pr}[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] \,.$$

Game $\mathbf{G}_1$ (No forgery with the same H query): In this game, after $\mathcal{A}$ outputs the forgeries, the game aborts if there exist $i_0 \neq i_1$ with $i_0, i_1 \in [\ell]$ such that for the corresponding hash queries

$$\mathsf{HQry}^{(i_0)} := (\widetilde{M}^{(i_0)}, \widetilde{\sigma}^{(i_0)}_{\mathsf{mac}}, \mathrm{cmt}^{(i_0)}_{\mathsf{rnd}}, \widetilde{A}^{(i_0)}_{\mathsf{os}}, \widetilde{A}^{(i_0)}_{\mathsf{mac}}, \widetilde{A}^{(i_0)}_{\mathsf{dleq}}, \widetilde{A}^{(i_0)}_{\mathsf{rnd}}, \mu^{(i_0)}, \mathsf{info}^*) \,,$$
$$\mathsf{HQry}^{(i_1)} := (\widetilde{M}^{(i_1)}, \widetilde{\sigma}^{(i_1)}_{\mathsf{mac}}, \mathrm{cmt}^{(i_1)}_{\mathsf{rnd}}, \widetilde{A}^{(i_1)}_{\mathsf{os}}, \widetilde{A}^{(i_1)}_{\mathsf{mac}}, \widetilde{A}^{(i_1)}_{\mathsf{dleq}}, \widetilde{A}^{(i_1)}_{\mathsf{rnd}}, \mu^{(i_1)}, \mathsf{info}^*) \,,$$

we have that $\mathsf{HQry}^{(i_0)} = \mathsf{HQry}^{(i_1)}$. Denote the event $\mathsf{SameQuery}$ as $\mathcal{A}$ winning in $\mathbf{G}_0$ but not in $\mathbf{G}_1$. We can see that

$$\mathsf{Pr}[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] + \mathsf{Pr}[\mathsf{SameQuery}] \,.$$

Then, the following lemma gives a bound on the probability of $\mathsf{SameQuery}$ by relying on relaxed unique responses of the Σ-protocols.

Lemma 6.1. *There exist adversaries $\mathcal{B}_{\mathsf{ur,os}}, \mathcal{B}_{\mathsf{ur,mac}}$ and $\mathcal{B}_{\mathsf{ur,rnd}}$ such that*

$$\mathsf{Pr}[\mathsf{SameQuery}] \leqslant \mathsf{Adv}^{\mathsf{sbind}}_{\mathsf{C}_{\mathsf{rnd}}}(\mathcal{B}_{\mathsf{ur,rnd}}, \lambda) + \mathsf{Adv}^{\mathsf{dl}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ur,os}}, \lambda) + \mathsf{Adv}^{\mathsf{sbind}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_{\mathsf{ur,mac}}, \lambda) + \frac{Q_{\mathsf{pb}} + 1}{p} \,.$$

Moreover, all these adversaries runs in time $T_{\mathsf{ur,os}}, T_{\mathsf{ur,mac}}, T_{\mathsf{ur,rnd}} \approx T_{\mathcal{A}}$.

Proof (of Lemma 6.1). By definition of $\mathsf{SameQuery}$, we have that all forgeries are valid, meaning the Σ_{xxx} transcripts $(\widetilde{A}^{(i)}_{\mathsf{xxx}}, \widetilde{c}^{(i)}_{\mathsf{xxx}}, \widetilde{z}^{(i)}_{\mathsf{xxx}})$ for $\mathsf{xxx} \in \{\mathsf{os}, \mathsf{mac}, \mathsf{dleq}, \mathsf{rnd}\}$ are valid with respect to their corresponding statement. We now consider the two forgeries indexed i_0, i_1 in the event. Recall that the two forgeries correspond to the same H query $\mathsf{HQry}^{(i_0)} = \mathsf{HQry}^{(i_1)}$ which contains $\widetilde{M}^{(i_0)} = \widetilde{M}^{(i_1)}$, the blinded tag $\widetilde{\sigma}^{(i_0)}_{\mathsf{mac}} = \widetilde{\sigma}^{(i_1)}_{\mathsf{mac}}$, the commitment $\mathrm{cmt}^{(i_0)}_{\mathsf{rnd}} = \mathrm{cmt}^{(i_1)}_{\mathsf{rnd}}$, identical nonces $\widetilde{A}^{(i_0)}_{\mathsf{xxx}} = \widetilde{A}^{(i_1)}_{\mathsf{xxx}}$ for all xxx and message $\mu^{(i_0)} = \mu^{(i_1)}$. However, the two forgeries are distinct by the winning condition. Therefore, the differences can only come from the challenges $\widetilde{c}_{\mathsf{xxx}}$ or responses $\widetilde{z}_{\mathsf{xxx}}$ parts of the forgery, which are the only non-identical parts.

We now consider the following cases on how the two forgeries can differ:

Distinct challenges: Note that because $\widetilde{c}_{\mathsf{os}}$ and $\widetilde{c}_{\mathsf{rnd}}$ are outputs of H and the queries are the same, they cannot differ. Thus, it could only be the case that $\widetilde{c}^{(i_0)}_{\mathsf{mac}} \neq \widetilde{c}^{(i_1)}_{\mathsf{mac}}$ or $\widetilde{c}^{(i_0)}_{\mathsf{dleq}} \neq \widetilde{c}^{(i_1)}_{\mathsf{dleq}}$. However, these two cases are identical because by verification of $\mathsf{BS}_{\mathrm{alg}}.\mathsf{Ver}$, $\widetilde{c}^{(i_0)}_{\mathsf{mac}} + \widetilde{c}^{(i_0)}_{\mathsf{dleq}} = \widetilde{c}^{(i_1)}_{\mathsf{mac}} + \widetilde{c}^{(i_1)}_{\mathsf{dleq}}$ since they are the output of the same H query. Hence, we only focus on the case that $\widetilde{c}^{(i_0)}_{\mathsf{dleq}} \neq \widetilde{c}^{(i_1)}_{\mathsf{dleq}}$.

In this case, since two Σ_{dleq} transcripts $(\widetilde{A}^{(i)}_{\mathsf{dleq}}, \widetilde{c}^{(i)}_{\mathsf{dleq}}, \widetilde{z}^{(i)}_{\mathsf{dleq}})_{i\in\{i_0,i_1\}}$ are valid for the same statement defined by $(D_2^*, D_3^*) = \mathsf{H}_{\mathsf{pb}}(\mathsf{info}^*)$, by 2-special-soundness and the fact that the nonces are identical, we have that $\mathsf{dlog}_G D_2^* = \mathsf{dlog}_{D_1} D_3^*$. However, recall that outputs to H_{pb} are uniformly random in $\mathbb{G}^2$, so this case only occurs with probability at most Q_{pb}/p.

Identical challenges, but distinct responses: Here, we consider different cases based on which Σ_{xxx} whose responses $\widetilde{z}^{(i_0)}_{\mathsf{xxx}} \neq \widetilde{z}^{(i_1)}_{\mathsf{xxx}}$ are distinct. In particular, we will invoke the relaxed unique-response extractor $\Sigma_{\mathsf{xxx}}.\mathsf{UREXt}$ of these Σ-protocols. For simplicity, we will refer to the identical queries as $(\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{A}_{\mathsf{os}}, \widetilde{A}_{\mathsf{mac}}, \widetilde{A}_{\mathsf{dleq}}, \widetilde{A}_{\mathsf{rnd}}, \mu, \mathsf{info}^*)$, the challenges as $\widetilde{c}_{\mathsf{xxx}}$, and the corresponding statements for each of these as $\mathbb{x}_{\mathsf{xxx}}$.

1. Σ_{os}: In this case, computing $\mathbb{w} = \boldsymbol{\delta}_{\mathsf{os}} \leftarrow \Sigma_{\mathsf{os}}.\mathsf{UREXt}(\mathbb{x}_{\mathsf{os}}, (A_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}^{(i)}_{\mathsf{os}})_{i\in\{i_0,i_1\}})$ gives us $\delta_{\mathsf{os},0}G + \delta_{\mathsf{os},1}H = 0_{\mathbb{G}}$ and $\boldsymbol{\delta}_{\mathsf{os}} \neq \mathbf{0}$, by definition of $\widetilde{\mathfrak{R}}_{\mathsf{os}}$ that $\phi^{\mathsf{os}}(\mathbb{w}) = 0_{\mathbb{G}}$. Thus, $\mathsf{dlog}_G H = -\delta_{\mathsf{os},0} \cdot \delta^{-1}_{\mathsf{os},1}$ can be efficiently computed; so we can construct a reduction $\mathcal{B}_{\mathsf{ur,os}}$ simulating the game to $\mathcal{A}$ (as $\mathsf{dlog}_G H$ is not needed in $\mathbf{G}_0$) and upper bounding the probability of this case occurring.

2. Σ_{mac}: We can extract $\mathbb{w} \leftarrow \Sigma_{\mathsf{mac}}.\mathsf{UREXt}(\mathbb{x}_{\mathsf{mac}}, (A_{\mathsf{mac}}, \widetilde{c}_{\mathsf{mac}}, \widetilde{z}^{(i)}_{\mathsf{mac}})_{i\in\{i_0,i_1\}})$ where $((\mathsf{pp}, \mathbb{x}_{\mathsf{mac}}), \mathbb{w}') \in \widetilde{\mathfrak{R}}_{\mathsf{mac}}$. Notably $\mathbb{w} = (\delta_u, \delta_{x,0}, \delta_{x,1}, \delta_{y,0}, \delta_{y,1}, \delta_z, \delta_\rho) \neq (\mathbf{0} \in \mathbb{Z}_p^6, 0 \in \mathsf{C}_{\mathsf{mac}}.\mathcal{R})$ such that $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}(\mathbb{w}) = 0$, which includes the following equations

$$\delta_u \widetilde{U} = 0_{\mathbb{G}} \ , \qquad (\delta_{x,1}, \delta_{y,1}, \delta_z)F = (0_{\mathbb{G}}, 0_{\mathbb{G}}, 0_{\mathbb{G}}) \ ,$$
$$\mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (\delta_{x,0}, \delta_{y,0}); \delta_\rho) = \mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (0,0); 0) \ .$$

We know $\widetilde{U} \neq 0_{\mathbb{G}}$, so $\delta_u = 0$. Unless $F = 0_{\mathbb{G}}$ (occurs with probability $1/p$), we also have $(\delta_{x,1}, \delta_{y,1}, \delta_z) = \mathbf{0}$. This means that $(\delta_{x,0}, \delta_{y,0}) \neq 0$ or $\delta_\rho \neq 0$. This however violates the strong binding property of $\mathsf{C}_{\mathsf{mac}}$: we obtain a zero and non-zero opening for the same commitment. Thus, we can construct a reduction $\mathcal{B}_{\mathsf{ur},\mathsf{mac}}$ simulating the game to $\mathcal{A}$ (recall $\mathbf{G}_0$ samples pp with $\mathsf{BS}_{\mathrm{alg}}.\mathsf{Setup}$ which coincides with how mck is sampled) upper bounding the probability of this case occurring.

3. Σ_{dleq}: Since Σ_{dleq} satisfies unique responses, this case cannot occur.
4. Σ_{rnd}: We can extract $\mathbb{w} \leftarrow \Sigma_{\mathsf{rnd}}.\mathsf{UREXt}(\mathbb{x}_{\mathsf{rnd}}, (\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}^{(i)}_{\mathsf{rnd}})_{i\in\{i_0,i_1\}})$ where $((\mathsf{pp}, \mathbb{x}_{\mathsf{rnd}}), \mathbb{w}') \in \widetilde{\mathfrak{R}}_{\mathsf{rnd}}$. Notably, $\mathbb{w} = (\delta_\rho, \boldsymbol{\delta} \in \mathbb{Z}_p^6) \neq (0 \in \mathsf{C}_{\mathsf{rnd}}.\mathcal{R}, \mathbf{0})$ and $\phi^{\mathsf{rnd}}_{\mathsf{pp},\mathbb{x}_{\mathsf{rnd}}}(\mathbb{w}) = 0$, which implies $\mathsf{C}_{\mathsf{rnd}}.\mathsf{Commit}(\mathsf{bck}, \mathbf{0}_{\mathbb{G}}; 0) = \mathsf{C}_{\mathsf{rnd}}.\mathsf{Commit}(\mathsf{bck}, \boldsymbol{\Delta}; \delta_\rho)$ where $\boldsymbol{\Delta} = (\boldsymbol{\delta} \cdot \widetilde{U}, \delta_2 W - \delta_0 G - \delta_1 H)$. Since $\widetilde{U} \neq 0_{\mathbb{G}}$ via the verification, if $\boldsymbol{\delta} \neq \mathbf{0}$ then $\boldsymbol{\Delta} \neq \mathbf{0}_{\mathbb{G}}$. Otherwise, $\delta_\rho \neq 0$; either of which violate strong binding of $\mathsf{C}_{\mathsf{rnd}}$. Therefore, we can similarly construct a reduction $\mathcal{B}_{\mathsf{ur},\mathsf{rnd}}$ upper bounding the probability of this case occurring. □

Game $\mathbf{G}_2$ (Guess hash index of info^*): In this game, a uniformly random index $i^*_{\mathsf{pb}} \leftarrow\!\!\$\ [Q_{\mathsf{pb}}]$ is sampled at the beginning of the game, and when $\mathcal{A}$ outputs the forgeries, the game aborts if $\mathsf{info}^* \neq \mathsf{info}_{i^*_{\mathsf{pb}}}$. From this point on, we will also interchangeably call signing sessions for signing $\mathsf{info}_{i^*_{\mathsf{pb}}}$ as signing info^*.

Since Q_{pb} includes the queries to H_{pb} made by the game, there exists $i_{\mathsf{pb}} \in [Q_{\mathsf{pb}}]$ such that $\mathsf{info}^* = \mathsf{info}_{i_{\mathsf{pb}}}$. Since the oracle simulation are independent of the uniformly random $\mathsf{info}_{i^*_{\mathsf{pb}}}$, we have that

$$\mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \leqslant Q_{\mathsf{pb}} \cdot \mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] \ .$$

Game $\mathbf{G}_3$ (Generate bck with extraction trapdoor): In this game, the commitment key bck is generated along with a trapdoor td. That is, the game sets $(\mathsf{bck}, \mathsf{td}) \leftarrow\!\!\$\ \mathsf{C}_{\mathsf{rnd}}.\mathsf{ExtSetup}(1^\lambda)$ for the extractor $\mathsf{Ext} = (\mathsf{C}_{\mathsf{rnd}}.\mathsf{ExtSetup}, \mathsf{C}_{\mathsf{rnd}}.\mathsf{Extract})$ of $\mathsf{C}_{\mathsf{rnd}}$ (via extraction property of $\mathsf{C}_{\mathsf{rnd}}$).

By extractability of $\mathsf{C}_{\mathsf{rnd}}$, we can then construct an adversary $\mathcal{B}_{\mathsf{dist}}$ running in time $T_{\mathsf{dist}} \approx T_{\mathcal{A}}$ such that

$$\left|\mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1]\right| \leqslant \mathsf{Adv}^{\mathsf{ext\text{-}dist}}_{\mathsf{C}_{\mathsf{rnd}},\mathsf{Ext}}(\mathcal{B}_{\mathsf{dist}}, \lambda) \ .$$

Game $\mathbf{G}_4$ (Extract from $\mathrm{cmt}^{(i)}_{\mathsf{rnd}}$ in the forgeries): When $\mathcal{A}$ returns the forgeries containing info^* and $(\mu^{(i)}, \sigma^{(i)})_{i\in[Q_{\mathrm{S},\mathsf{info}^*}]}$), this game efficiently extract for each $i \in [\ell]$ the following group elements

$$(\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}_{\mathsf{os}}, \overline{R}^{(i)}_x, \overline{R}^{(i)}_y, \overline{R}^{(i)}_M, \overline{A}^{(i)}_{\mathsf{os}}) \leftarrow \mathsf{C}_{\mathsf{rnd}}.\mathsf{Extract}(\mathsf{td}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}) \ .$$

Since these extracted elements are not used to determine the winning condition, we have that

$$\mathsf{Pr}[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] \ .$$

We note that by extractability of $\mathsf{C}_{\mathsf{rnd}}$ when bck is generated with an extraction trapdoor, these extracted elements are the only possible elements with valid openings for $\mathrm{cmt}^{(i)}_{\mathsf{rnd}}$. Now, we consider the two following events with respect to the extracted $(\overline{A}^{(i)}_{\mathsf{os}})_{i\in[\ell]}$.

- $\mathsf{NonceColl}$: The adversary $\mathcal{A}$ wins in $\mathbf{G}_4$, and there exist $i_0 \neq i_1$ with $i_0, i_1 \in [\ell]$ such that

$$\overline{A}^{(i_0)}_{\mathsf{os}} = \overline{A}^{(i_1)}_{\mathsf{os}} .$$

- $\mathsf{NonceDist}$: The adversary $\mathcal{A}$ wins in $\mathbf{G}_4$, and for all $i_0, i_1 \in [\ell]$ with $i_0 \neq i_1$, we have

$$\overline{A}^{(i_0)}_{\mathsf{os}} \neq \overline{A}^{(i_1)}_{\mathsf{os}} .$$

Observe that these two events are efficiently checkable, and this will be relevant later on in some game hops in Section 6.3. It is easy to see that the two events are disjoint, and $\mathsf{NonceColl} \vee \mathsf{NonceDist}$ is equivalent to $\mathcal{A}$ winning in the game $\mathbf{G}_4^{\mathcal{A}}$, we have

$$\mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}_{\mathbf{G}_4^{\mathcal{A}}}[\mathsf{NonceColl}] + \mathsf{Pr}_{\mathbf{G}_4^{\mathcal{A}}}[\mathsf{NonceDist}] \ .$$

Thus, to conclude the proof, we bound the two probabilities as established in the following lemmas.

The case NonceColl is analyzed via a reduction to DL with the core technical idea being that an Okamoto-Schnorr witness (different from the one used to simulate the game) is likely to be extracted from the two forgeries with colliding $\overline{A}_{\mathsf{os}}^{(i_0)} = \overline{A}_{\mathsf{os}}^{(i_1)}$.

The case NonceDist follows from a reduction to UF-RMA security of $\mathsf{M}_{\mathsf{alg}}$. Note that the nonces A_{os} from singing sessions of info^* are *exactly* the values for which a valid tag is issued, and due to Lemma 2.5 (all signing sessions are completed), there must be an extracted nonce $\overline{A}_{\mathsf{os}}^{(i)}$ that has not yet been tagged during signing of info^*. For signing on $\mathsf{info} \neq \mathsf{info}^*$, we will later move to a game where the MAC tags are not generated honestly. To avoid issues with rewinding, we will sample all nonces A_{os} in advance to avoid tagging new nonces during the second run. Finally, the bound in the theorem statement follows via inspection. □

Lemma 6.2. *There exists an adversary $\mathcal{B}_{\mathsf{dl}}$ running in time $T_{\mathsf{dlog}} \approx 3 \cdot T_{\mathcal{A}}$ such that*

$$\mathsf{Pr}_{\mathbf{G}_4^{\mathcal{A}}}[\mathsf{NonceColl}] \leqslant \frac{Q_{\mathsf{H}}^2 + 9Q_{\mathsf{H}}}{p} + 2\sqrt[3]{32Q_{\mathrm{H}}^2 \cdot \mathsf{Adv}^{\mathsf{dl}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{dl}}, \lambda)} \ .$$

Lemma 6.3. *There exist adversaries $\mathcal{B}_{\mathrm{alg}}$ running in time $T_{\mathrm{alg}} \approx 2 \cdot T_{\mathcal{A}}$, $\mathcal{B}_{\mathsf{hide}}$ running in time $T_{\mathsf{hide}} \approx T_{\mathcal{A}}$, and $\mathcal{B}_{\mathsf{ddh},k}$ (for $k \in [4]$) running in time $T_{\mathsf{ddh},k} \approx T_{\mathcal{A}}$ such that*

$$\begin{aligned}\mathsf{Pr}_{\mathbf{G}_4^{\mathcal{A}}}[\mathsf{NonceDist}] \leqslant\ & \mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathsf{pb}}}(\mathcal{B}_{\mathsf{ddh},1}, \lambda) + 2\mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathrm{S}}}(\mathcal{B}_{\mathsf{ddh},2}, \lambda) \\ & + 2 \cdot \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_{\mathsf{hide}}, \lambda) + \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ddh},3}, \lambda) + \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ddh},4}, \lambda) \\ & + \sqrt{Q_{\mathsf{H}} \cdot \mathsf{Adv}^{\mathsf{ufrma}}_{\mathsf{M}_{\mathsf{alg}}}(\mathcal{B}_{\mathrm{alg}}, \lambda)} + \frac{3Q_{\mathsf{H}} + 3Q_{\mathrm{S}} + 3}{p} \ .\end{aligned}$$

Moreover, $\mathcal{B}_{\mathrm{alg}}$ makes at most Q_{S} queries to its $\mathrm{Tag}(\cdot)$ oracle.

6.2 Proof of Lemma 6.2 (Extracting DLOG from Nonce Collision)

We refer the readers to the proof sketch of this case in Section 3.3. In this proof, we consider the same adversary $\mathcal{A}$ as in Section 6.1 within the game $\mathbf{G}_4$ with the winning condition changed to NonceColl occurring. We let indices $i_0, i_1 \in [\ell]$ with $i_0 < i_1$ denote the lexicographically smallest pair (i_0, i_1) that satisfy conditions for NonceColl. We stick to the same notations as in Section 6.1 and consider the following sequence of games $\mathbf{G}_{\mathrm{A}.k}$ for $k \in [0, 4]$.

Game $\mathbf{G}_{\mathrm{A}.0}$ ($\mathbf{G}_4$ with NonceColl as winning condition): This game is identical to $\mathbf{G}_4$ except that the adversary now only wins if NonceColl occurs. We can see that

$$\mathsf{Pr}_{\mathbf{G}_4^{\mathcal{A}}}[\mathsf{NonceColl}] = \mathsf{Pr}[\mathbf{G}_{\mathrm{A}.0}^{\mathcal{A}} \Rightarrow 1] \ .$$

Game $\mathbf{G}_{\mathrm{A}.1}$ (Extract from $\mathrm{cmt}_{\mathsf{rnd}}$ in all H queries): When $\mathcal{A}$ makes H queries of the form

$$(\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{A}_{\mathsf{os}}, \widetilde{A}_{\mathsf{mac}}, \widetilde{A}_{\mathsf{dleq}}, \widetilde{A}_{\mathsf{rnd}}, \mu, \mathsf{info}) \ ,$$

the game also extracts from $\mathrm{cmt}_{\mathsf{rnd}}$ the group elements

$$(\overline{A}_0, \overline{A}_1, \overline{\Gamma}_{\mathsf{os}}, \overline{R}_x, \overline{R}_y, \overline{R}_M, \overline{A}_{\mathsf{os}}) \leftarrow \mathsf{C}_{\mathsf{rnd}}.\mathsf{Extract}(\mathsf{td}, \mathrm{cmt}_{\mathsf{rnd}}) \ ,$$

as done with the forgeries. Since this change affects neither the simulation nor the winning condition, we have

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.0} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.1} \Rightarrow 1] \ .$$

The changes from this point on will be inefficient checks on the forgeries. Note that the final reduction will still be efficient as it does not explicitly implement these checks. However, the goal for introducing them is to bound the probability that the forgeries fix the underlying nonces and blinding factors which can be used to break DL; and consequently, this probability will be relevant in the analysis of the reduction.

Game $\mathbf{G}_{\mathrm{A}.2}$ (Abort if "extracted unblinded challenges" are equal): At the end of the game, this game introduces an additional abort: for each $i \in [\ell]$, let $\gamma^{(i)}_{\mathsf{os}} = \mathsf{dlog}_{\widetilde{U}^{(i)}} \overline{\Gamma}^{(i)}_{\mathsf{os}}$ and $\boldsymbol{\alpha}^{(i)}_{\mathsf{os}} = \mathsf{dlog}_{\widetilde{U}^{(i)}}(\overline{A}^{(i)}_0, \overline{A}^{(i)}_1)$, then it aborts if

$$\gamma^{(i_0)}_{\mathsf{os}} - \widetilde{c}^{(i_0)}_{\mathsf{os}} = \gamma^{(i_1)}_{\mathsf{os}} - \widetilde{c}^{(i_1)}_{\mathsf{os}} \ . \tag{6}$$

Intuitively, this bad event implies that unblinded challenges of $\widetilde{c}^{(i_0)}_{\mathsf{os}}$ and $\widetilde{c}^{(i_1)}_{\mathsf{os}}$ are identical, which would not allow our final reduction to extract the underlying witness of the Okamoto-Schnorr public key W.

We now argue that the introduced above occurs with small probability. We consider each pair of queries HQry and HQry' to H, with HQry' being the later query and will later apply the union bound over the $\binom{Q_{\mathsf{H}}}{2} \leqslant Q^2_{\mathsf{H}}$ queries. For these two queries, denote

$$\mathsf{HQry} := (\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}} = (\widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{T}_0, \widetilde{S}_1, \widetilde{T}_1), \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{A}_{\mathsf{os}}, \widetilde{A}_{\mathsf{mac}}, \widetilde{A}_{\mathsf{dleq}}, \widetilde{A}_{\mathsf{rnd}}, \mu, \mathsf{info}) \ , \tag{7}$$

$$\mathsf{HQry}' := (\widetilde{M}', \widetilde{\sigma}'_{\mathsf{mac}} = (\widetilde{U}', \widetilde{V}', \widetilde{S}'_0, \widetilde{T}'_0, \widetilde{S}'_1, \widetilde{T}'_1), \mathrm{cmt}'_{\mathsf{rnd}}, \widetilde{A}'_{\mathsf{os}}, \widetilde{A}'_{\mathsf{mac}}, \widetilde{A}'_{\mathsf{dleq}}, \widetilde{A}'_{\mathsf{rnd}}, \mu', \mathsf{info}') \ , \tag{8}$$

the corresponding bad event which leads to the abort is such that

$$\mathsf{dlog}_{\widetilde{U}} \overline{\Gamma}_{\mathsf{os}} - \widetilde{c}_{\mathsf{os}} = \mathsf{dlog}_{\widetilde{U}'} \overline{\Gamma}'_{\mathsf{os}} - \widetilde{c}'_{\mathsf{os}} \ ,$$

where $\widetilde{c}_{\mathsf{os}}$ and $\widetilde{c}'_{\mathsf{os}}$ are parts of the output of $\mathsf{H}(\mathsf{HQry})$ and $\mathsf{H}(\mathsf{HQry}')$, respectively. At the time that $\mathcal{A}$ made the query HQry', HQry and $\widetilde{c}_{\mathsf{os}}$ are already fixed; and hence, $\overline{\Gamma}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \overline{\Gamma}'_{\mathsf{os}}$ as well. Thus, the probability that Equation (6) occurs is at most $1/p$ due to $\widetilde{c}'_{\mathsf{os}}$ being uniformly random in $\mathbb{Z}_p$. Accordingly, we have that

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.1} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.2} \Rightarrow 1] + \frac{Q^2_{\mathsf{H}}}{p} \ .$$

Game $\mathbf{G}_{\mathrm{A}.3}$ (Abort if the extracted $\overline{A}^{(i)}_{\mathsf{os}}$ is not unblinded $\widetilde{A}^{(i)}_{\mathsf{os}}$): At the end of the game, the game introduces the following abort: if there exists an $i \in [\ell]$ such that for $\boldsymbol{\alpha}^{(i)}_{\mathsf{os}}, \gamma^{(i)}_{\mathsf{os}}$ as defined in the previous game

$$\overline{A}^{(i)}_{\mathsf{os}} \neq \widetilde{A}^{(i)}_{\mathsf{os}} - \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\boldsymbol{\alpha}^{(i)}_{\mathsf{os}}) + \gamma^{(i)}_{\mathsf{os}} \cdot W \ .$$

Observe here that the commitment key bck is generated with extraction trapdoor and via extractability of $\mathsf{C}_{\mathsf{rnd}}$. Therefore, if such an event occurs, the corresponding statement $\mathbb{x}^{(i)}_{\mathsf{rnd}} = (W, X_1, Y_1, \widetilde{A}^{(i)}_{\mathsf{os}}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, \widetilde{M}^{(i)}, \widetilde{U}^{(i)}, \widetilde{S}^{(i)}_1, \widetilde{T}_1)^{(i)}$ is not in the language induced by the map $\phi^{\mathsf{rnd}}_{\mathsf{pp},\mathbb{x}^{(i)}_{\mathsf{rnd}}}$.

We now bound the probability that such abort can occur via special-soundness of Σ_{rnd}. We consider each RO query HQry to H parse as in Equation (7). Via 2-special-soundness of Σ_{rnd}, we have that if the corresponding statement $\mathbb{x}_{\mathsf{rnd}} = (W, X_1, Y_1, \widetilde{A}_{\mathsf{os}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{M}, \widetilde{U}, \widetilde{S}_1, \widetilde{T}_1)$ parsed from HQry is not in the language, there is at most one challenge $\widetilde{c}_{\mathsf{rnd}}$ which allows the adversary to output a valid response $\widetilde{z}_{\mathsf{rnd}}$ to the nonce $\widetilde{A}_{\mathsf{rnd}}$. Hence, there is at most $1/p$ probability of such uniformly random $\widetilde{c}_{\mathsf{rnd}}$ being sampled; thus, by the union bound over H queries, we have

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.2} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.3} \Rightarrow 1] + \frac{Q_{\mathsf{H}}}{p} \ .$$

We note that this game ensures that for all $i \in [\ell]$, we have

$$\begin{aligned}\overline{A}_{\mathsf{os}}^{(i)} &= \widetilde{A}_{\mathsf{os}}^{(i)} - \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\boldsymbol{\alpha}_{\mathsf{os}}^{(i)}) + \gamma_{\mathsf{os}}^{(i)} \cdot W \\ &= \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\widetilde{z}_{\mathsf{os}}^{(i)} - \boldsymbol{\alpha}_{\mathsf{os}}^{(i)}) - (\widetilde{c}_{\mathsf{os}}^{(i)} - \gamma_{\mathsf{os}}^{(i)}) \cdot W \ .\end{aligned}$$

Game $\mathbf{G}_{\mathrm{A}.4}$ (Abort if the to-be-extracted witness of W is not new): At the end of the game, the game introduces an abort with respect to the i_0-th and i_1-th forgeries as defined in the event $\mathsf{NonceColl}$ (i.e., $\overline{A}_{\mathsf{os}}^{(i_0)} = \overline{A}_{\mathsf{os}}^{(i_1)}$). In particular, the game aborts if

$$(\gamma_{\mathsf{os}}^{(i_0)} - \gamma_{\mathsf{os}}^{(i_1)} + \widetilde{c}_{\mathsf{os}}^{(i_1)} - \widetilde{c}_{\mathsf{os}}^{(i_0)}) \cdot \boldsymbol{w} \neq \widetilde{z}_{\mathsf{os}}^{(i_0)} - \boldsymbol{\alpha}_{\mathsf{os}}^{(i_0)} - \widetilde{z}_{\mathsf{os}}^{(i_1)} + \boldsymbol{\alpha}_{\mathsf{os}}^{(i_1)} \ .$$

We remark again that these checks are not efficient; our reduction will not implement these checks. However, we will define a good event later on which will happen with prob of adv winning in $\mathbf{G}_4$. Then, we will relate this particular event occurring to the event that the reduction successfully breaks DL. The following claim then lower bounds $\mathcal{A}$'s winning probability in $\mathbf{G}_{\mathrm{A}.4}$.

Lemma 6.4. $\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.3} \Rightarrow 1] \leqslant 2 \cdot \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.4} \Rightarrow 1]$.

Proof. We prove this claim in two steps:

1. First, we show that for any (not necessarily efficient) adversary $\mathcal{A}$ given group element W (generated by sampling $\boldsymbol{w} \leftarrow\!\!\$\ \mathbb{Z}_p^2$ and setting $W = \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\boldsymbol{w})$) and oracle access to the Okamoto-Schnorr signing protocol (as in Figure 9), its probability of finding *any preimage* $\boldsymbol{w}'$ of W with respect to the map $\phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}$ is at most twice the probability of finding a different preimage $\boldsymbol{w}' \neq \boldsymbol{w}$. The former corresponds to the game PR and the latter to CR.
2. Second, we relate the above cases to games $\mathbf{G}_{\mathrm{A}.3}$ and $\mathbf{G}_{\mathrm{A}.4}$. In particular, we will show that the winning event in game $\mathbf{G}_{\mathrm{A}.3}$ corresponds to finding a preimage of W, and the winning event in game $\mathbf{G}_{\mathrm{A}.4}$ corresponds to finding a *different* preimage of W.

Lemma 6.5 (Preimage to collision-finding). *Consider the games in Figure 9. For any adversary* (possibly unbounded) $\mathcal{A}$, $\mathsf{Pr}[\mathrm{PR}^{\mathcal{A}}_{\mathsf{GGen}}(\lambda) \Rightarrow 1] \leqslant 2 \cdot \mathsf{Pr}[\mathrm{CR}^{\mathcal{A}}_{\mathsf{GGen}}(\lambda) \Rightarrow 1]$.

Proof (of Lemma 6.5). The proof is mostly identical that of [TZ23, Theorem 1.], but we include it since their theorem statement does not directly imply our lemma statement. In particular, their statement shows an existence of another adversary $\mathcal{B}$ breaking collision resistance using $\mathcal{A}$, but in our case, we need to bound probability that $\mathcal{A}$ wins in PR with the probability that $\mathcal{A}$ wins in CR *of the same adversary* $\mathcal{A}$.

Observe that the execution of this game depends on $\mathsf{pp}_{\mathsf{os}} = (\mathbb{G}, p, G, H)$, the random coins $\rho_{\mathcal{A}}$ of $\mathcal{A}$, and the random values $\boldsymbol{w}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{Q_{\mathrm{S}}} \in \mathbb{Z}_p^2$. We denote the two following sets $\mathrm{WIN}_{\mathrm{PR}}$ and $\mathrm{WIN}_{\mathrm{CR}}$ as the sets of $\mathsf{inp} = (\mathsf{pp} = (\mathsf{pp}_{\mathsf{os}}, \rho_{\mathcal{A}}), \boldsymbol{w}, \mathbf{r} = (\boldsymbol{r}_i)_{i \in [Q_{\mathrm{S}}]})$ that makes $\mathcal{A}$ wins the games $\mathrm{PR}^{\mathcal{A}}_{\mathsf{GGen}}(\lambda)$ (finding a preimage of W) and $\mathrm{CR}^{\mathcal{A}}_{\mathsf{GGen}}(\lambda)$ (finding a preimage $\boldsymbol{w}' \neq \boldsymbol{w}$ of W), respectively.

We now consider a map $\Psi(\mathsf{pp}, \boldsymbol{w}, \mathbf{r})$ which preserves the view of $\mathcal{A}$, similar to the proofs of [TZ23, Theorem 1] and [HKL19, Lemma 7.3]. First, we let $\boldsymbol{c}(\mathsf{pp}, \boldsymbol{w}, \mathbf{r}) = (c_i)_{i \in [Q_{\mathrm{S}}]}$ denote the values c_{sid} that $\mathcal{A}(\mathsf{pp}_{\mathsf{os}}, W = w_0 G + w_1 H); \rho_{\mathcal{A}})$ queries S_2 with. Then, let $\Psi : (\mathsf{pp}, \boldsymbol{w}, \mathbf{r}) \mapsto (\mathsf{pp}, \boldsymbol{w} + \boldsymbol{\Delta}, \mathbf{r} - \boldsymbol{c}(\mathsf{pp}, \boldsymbol{w}, \mathbf{r})\boldsymbol{\Delta})$ where $\boldsymbol{\Delta} = (-\mathsf{dlog}_G H, 1)$. We claim that this map preserves the view of $\mathcal{A}$.

- pp remains the same.
- W is unchanged. This is because $\phi^{\mathsf{os}}_{\mathsf{pp}_{\mathsf{os}}}(\boldsymbol{\Delta}) = 0_{\mathbb{G}}$, so by linearity $\phi^{\mathsf{os}}_{\mathsf{pp}_{\mathsf{os}}}(\boldsymbol{w}) = \phi^{\mathsf{os}}_{\mathsf{pp}_{\mathsf{os}}}(\boldsymbol{w} + \boldsymbol{\Delta})$.[6]
- The outputs of S_1 remains unchanged with the similar linearity argument above.
- Finally, we can inductively argue that the output of S_2 is unchanged as follows: note that for any $\boldsymbol{r}$ and c, $\boldsymbol{r} + c\boldsymbol{w} = \boldsymbol{r} - c\boldsymbol{\Delta} + c \cdot (\boldsymbol{w} + \boldsymbol{\Delta})$; thus, it is only left to argue that $\boldsymbol{c}(\mathsf{pp}, \boldsymbol{w}, \mathbf{r}) = \boldsymbol{c}(\Psi(\mathsf{pp}, \boldsymbol{w}, \mathbf{r}))$. Assume that the queries made to S_2 are identical up to the i-th ($i \in [Q_{\mathrm{S}}]$) query between the two runs (with $(\mathsf{pp}, \boldsymbol{w}, \mathbf{r})$ and $\Psi(\mathsf{pp}, \boldsymbol{w}, \mathbf{r})$), then by the above observation, the output of this query is also identical. Therefore, the view of $\mathcal{A}$ remains the same.

[6] We removed $\mathbb{x}_{\mathsf{os}}$ from the map definition to make it clear that the map only depends on $\mathsf{pp}_{\mathsf{os}}$.

Game $\mathrm{PR}^{\mathcal{A}}_{\mathsf{GGen}}(\lambda)$: $\boxed{\mathrm{CR}^{\mathcal{A}}_{\mathsf{GGen}}(\lambda)}$:	Oracle $\mathrm{S}_1()$:
1 : $\mathcal{S} \leftarrow \varnothing$	1 : $Q \leftarrow Q + 1$
2 : $(\mathbb{G}, p, G) \leftarrow\!\$\ \mathsf{GGen}(1^\lambda)$	2 : $\boldsymbol{r}_Q \leftarrow\!\$\ \mathbb{Z}_p^2$
3 : $H \leftarrow\!\$\ \mathbb{G}\ ; \boldsymbol{w} \leftarrow\!\$\ \mathbb{Z}_p^2\ ; Q \leftarrow 0$	3 : **return** $(Q, \phi^{\mathsf{os}}_{\mathsf{pp}_{\mathsf{os}}}(\boldsymbol{r}_Q))$
4 : $\mathsf{pp}_{\mathsf{os}} \leftarrow (\mathbb{G}, p, G, H)$	Oracle $\mathrm{S}_2(\mathrm{sid}, c_{\mathrm{sid}})$:
5 : $W \leftarrow w_0 G + w_1 H$	1 : **if** $\mathrm{sid} \in \mathcal{S}$ **then return** $\perp$
6 : $\boldsymbol{w}' \leftarrow\!\$\ \mathcal{A}^{\mathrm{S}_1, \mathrm{S}_2}(\mathsf{pp}_{\mathsf{os}}, W)$	2 : $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathrm{sid}\}$
7 : **return** $W = \phi^{\mathsf{os}}_{\mathsf{pp}_{\mathsf{os}}}(\boldsymbol{w}')$ $\boxed{\wedge\ \boldsymbol{w} \neq \boldsymbol{w}'}$	3 : **return** $c_{\mathrm{sid}} \cdot \boldsymbol{w} + \boldsymbol{r}_{\mathrm{sid}}$

Fig. 9. Preimage and collision resistance games for $\phi^{\mathsf{os}}_{\mathsf{pp}_{\mathsf{os}}}$. We left out the $\mathbb{x}_{\mathsf{os}}$ since the map only depends on G and H.

Since this map preserves the view of $\mathcal{A}$, we can see that it is a bijection as the inverse $\Psi' : (\mathsf{pp}, \boldsymbol{w}, \mathbf{r}) \mapsto (\mathsf{pp}, \boldsymbol{w} - \boldsymbol{\Delta}, \mathbf{r} + \boldsymbol{c}(\mathsf{pp}, \boldsymbol{w}, \mathbf{r})\boldsymbol{\Delta})$ is such that $\Psi'(\Psi(\mathsf{pp}, \boldsymbol{w}, \mathbf{r})) = (\mathsf{pp}, \boldsymbol{w}, \mathbf{r})$ since $\boldsymbol{c}(\mathsf{pp}, \boldsymbol{w}, \mathbf{r}) = \boldsymbol{c}(\Psi(\mathsf{pp}, \boldsymbol{w}, \mathbf{r}))$.

Then, we can infer that if $\mathsf{inp} \in \mathrm{WIN}_{\mathrm{PR}}$ (i.e., $\mathcal{A}$ find a preimage of W), then either the output $\boldsymbol{w}'$ of $\mathcal{A}$ is $\boldsymbol{w}' = \boldsymbol{w}$ or $\boldsymbol{w}' \neq \boldsymbol{w}$. If the former case $\boldsymbol{w}' = \boldsymbol{w}$ occurs, this implies that $\boldsymbol{w}' \neq \boldsymbol{w} + \boldsymbol{\Delta}$; thus, $\Psi(\mathsf{inp}) \in \mathrm{WIN}_{\mathrm{CR}}$ because Ψ preserve the view of $\mathcal{A}$, so the output of $\mathcal{A}$ is preserved. If the latter case occurs $\boldsymbol{w}' \neq \boldsymbol{w}$, so $\mathsf{inp} \in \mathrm{WIN}_{\mathrm{CR}}$.

Therefore, we have that

$$\begin{aligned}
\mathsf{Pr}[\mathsf{inp} \in \mathrm{WIN}_{\mathrm{CR}}] &= \frac{1}{2}(\mathsf{Pr}[\mathsf{inp} \in \mathrm{WIN}_{\mathrm{CR}}] + \mathsf{Pr}[\mathsf{inp} \in \mathrm{WIN}_{\mathrm{CR}}]) \\
&= \frac{1}{2}(\mathsf{Pr}[\mathsf{inp} \in \mathrm{WIN}_{\mathrm{CR}}] + \mathsf{Pr}[\Psi(\mathsf{inp}) \in \mathrm{WIN}_{\mathrm{CR}}]) \\
&\geqslant \frac{1}{2}\mathsf{Pr}[\mathsf{inp} \in \mathrm{WIN}_{\mathrm{CR}} \ \vee\ \Psi(\mathsf{inp}) \in \mathrm{WIN}_{\mathrm{CR}}] \geqslant \frac{1}{2}\mathsf{Pr}[\mathsf{inp} \in \mathrm{WIN}_{\mathrm{PR}}]\,.
\end{aligned}$$

The first line is trivial. The second line follows from Ψ being a bijection, so $\Psi(\mathsf{inp})$ is identically distributed to uniformly sampled inp. The second to last inequality follows from the union bound. The last inequality follows from our observation above that $\mathsf{inp} \in \mathrm{WIN}_{\mathrm{PR}}$ implies $\mathsf{inp} \in \mathrm{WIN}_{\mathrm{CR}}$ or $\Psi(\mathsf{inp}) \in \mathrm{WIN}_{\mathrm{CR}}$. This then concludes the proof of the lemma. □

Relating the lemma to $\mathbf{G}_{\mathrm{A.3}}$ and $\mathbf{G}_{\mathrm{A.4}}$. We recall the winning conditions of the two games. In $\mathbf{G}_{\mathrm{A.3}}$, the winning condition is that (1) $\mathsf{NonceColl}$ occurs, (2) for the two corresponding forgeries i_0, i_1, $\gamma^{(i_0)}_{\mathsf{os}} - \tilde{c}^{(i_0)}_{\mathsf{os}} = \gamma^{(i_1)}_{\mathsf{os}} - \tilde{c}^{(i_1)}_{\mathsf{os}}$ (Equation (6)), and (3) for all forgeries $i \in [\ell]$, $\overline{A}^{(i)}_{\mathsf{os}} = \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\tilde{z}^{(i)}_{\mathsf{os}} - \boldsymbol{\alpha}^{(i)}_{\mathsf{os}}) - (\tilde{c}^{(i)}_{\mathsf{os}} - \gamma^{(i)}_{\mathsf{os}}) \cdot W$. Thus, by how i_0, i_1 are defined, we have that

$$\begin{aligned}
0_{\mathbb{G}} =& \overline{A}^{(i_1)}_{\mathsf{os}} - \overline{A}^{(i_0)}_{\mathsf{os}} \\
0_{\mathbb{G}} =& \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\tilde{z}^{(i_1)}_{\mathsf{os}} - \boldsymbol{\alpha}^{(i_1)}_{\mathsf{os}}) - (\tilde{c}^{(i_1)}_{\mathsf{os}} - \gamma^{(i_1)}_{\mathsf{os}}) \cdot W - (\phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\tilde{z}^{(i_0)}_{\mathsf{os}} - \boldsymbol{\alpha}^{(i_0)}_{\mathsf{os}}) - \\
& (\tilde{c}^{(i_0)}_{\mathsf{os}} - \gamma^{(i_0)}_{\mathsf{os}}) \cdot W) \\
W =& \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}((\tilde{c}^{(i_1)}_{\mathsf{os}} - \gamma^{(i_1)}_{\mathsf{os}} + \gamma^{(i_0)}_{\mathsf{os}} - \tilde{c}^{(i_0)}_{\mathsf{os}})^{-1} \cdot (\tilde{z}^{(i_1)}_{\mathsf{os}} - \boldsymbol{\alpha}^{(i_1)}_{\mathsf{os}} - \tilde{z}^{(i_0)}_{\mathsf{os}} + \boldsymbol{\alpha}^{(i_0)}_{\mathsf{os}}))\,.
\end{aligned}$$

The last equality follows from the observation that the inverted $\mathbb{Z}_p$-element is non-zero. In this case, we see that the event implies the existence of a preimage of W, namely $\boldsymbol{w}' = (\gamma^{(i_0)}_{\mathsf{os}} - \gamma^{(i_1)}_{\mathsf{os}} + \tilde{c}^{(i_1)}_{\mathsf{os}} - \tilde{c}^{(i_0)}_{\mathsf{os}})^{-1} \cdot (\tilde{z}^{(i_0)}_{\mathsf{os}} - \boldsymbol{\alpha}^{(i_0)}_{\mathsf{os}} - \tilde{z}^{(i_1)}_{\mathsf{os}} + \boldsymbol{\alpha}^{(i_1)}_{\mathsf{os}})$. Similarly, we have that the winning condition in game $\mathbf{G}_{\mathrm{A.4}}$ is exactly when $\boldsymbol{w} \neq \boldsymbol{w}'$.

Therefore, we can construct an adversary $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ (given in Figures 10 and 11) that plays either the PR game or the CR game, running $\mathcal{A}$, and using the given oracles to simulate the signing sessions, such that $\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A.3}} \Rightarrow 1] = \mathsf{Pr}[\mathrm{PR}^{\mathcal{A}_{\mathsf{pr\text{-}cr}}}_{\mathsf{GGen}}(\lambda) \Rightarrow 1]$ and $\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A.4}} \Rightarrow 1] = \mathsf{Pr}[\mathrm{CR}^{\mathcal{A}_{\mathsf{pr\text{-}cr}}}_{\mathsf{GGen}}(\lambda) \Rightarrow 1]$ to conclude the proof. The adversary $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ simulates the game $\mathbf{G}_{\mathrm{A.3}}$ (where the oracle simulation is the same as in $\mathbf{G}_{\mathrm{A.4}}$) to $\mathcal{A}$ except

Algorithm $\mathcal{A}_{\mathsf{pr\text{-}cr}}^{\bar{\mathrm{S}}_1,\bar{\mathrm{S}}_2}(\mathsf{pp}_{\mathsf{os}}, W)$:

$\mathbf{parse}\ \mathsf{pp}_{\mathsf{os}} = (\mathbb{G}, p, G, H)$
$\mathsf{mpp} = (F, \mathsf{mck}) \leftarrow\!\$\ \mathsf{M}_{\mathsf{alg}}.\mathsf{Setup}(1^\lambda)$
$\mathsf{bck} \leftarrow\!\$\ \mathsf{C}_{\mathsf{rnd}}.\mathsf{Setup}(1^\lambda)\ ; D_1 \leftarrow\!\$\ \mathbb{G}$
$\mathsf{pp} \leftarrow (\mathbb{G}, p, G, H, \mathsf{mpp}, \mathsf{bck}, D_1)$
$(\mathsf{msk}, \mathsf{ipp}) \leftarrow\!\$\ \mathsf{M}_{\mathsf{alg}}.\mathsf{KeyGen}(\mathsf{mpp})$
$\mathbf{parse}\ \mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho_0)\ , \mathsf{ipp} = (\mathrm{cmt}_0, X_1, Y_1, Z)$
$i^*_{\mathsf{pb}} \leftarrow\!\$\ [Q_{\mathsf{pb}}]$
$i_{\mathsf{pb}}, i_{\mathsf{H}}, J, J^* \leftarrow 0$ // Initialize counters
$(\mathsf{info}^*, (\mu^{(i)}, \sigma^{(i)})_{i\in[J^*+1]}) \leftarrow\!\$\ \mathcal{A}^{\mathrm{S}_1,\mathrm{S}_2,\mathsf{H},\mathsf{H}_{\mathsf{pb}}}(\mathsf{pp}, \mathsf{vk} = (\mathsf{ipp}, W))$
$\mathbf{for}\ i \in [J^*+1]:$
  $\mathbf{parse}\ \sigma^{(i)} = (\widetilde{M}^{(i)}, \tilde{\sigma}^{(i)}_{\mathsf{mac}}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, (\tilde{A}^{(i)}_{\mathsf{xxx}}, \tilde{c}^{(i)}_{\mathsf{xxx}}, \tilde{z}^{(i)}_{\mathsf{xxx}})_{\mathsf{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$
  $\mathbf{parse}\ \tilde{\sigma}^{(i)}_{\mathsf{mac}} = (\widetilde{U}^{(i)}, \widetilde{V}^{(i)}, \widetilde{S}^{(i)}_0, \widetilde{T}^{(i)}_0, \widetilde{S}^{(i)}_1, \widetilde{T}^{(i)}_1)$
  $\mathsf{HQry}^{(i)} \leftarrow (\widetilde{M}^{(i)}, \tilde{\sigma}^{(i)}_{\mathsf{mac}}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, \tilde{A}^{(i)}_{\mathsf{os}}, \tilde{A}^{(i)}_{\mathsf{mac}}, \tilde{A}^{(i)}_{\mathsf{dleq}}, \tilde{A}^{(i)}_{\mathsf{rnd}}, \mu^{(i)}, \mathsf{info}^*)$
  $(\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}_{\mathsf{os}}, \overline{R}^{(i)}_x, \overline{R}^{(i)}_y, \overline{R}^{(i)}_M, \overline{A}^{(i)}_{\mathsf{os}}) \leftarrow \mathsf{C}_{\mathsf{rnd}}.\mathsf{Extract}(\mathsf{td}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}})$
  $(\boldsymbol{\alpha}^{(i)}_{\mathsf{os}} = (\alpha^{(i)}_{\mathsf{os},0}, \alpha^{(i)}_{\mathsf{os},1}), \gamma^{(i)}_{\mathsf{os}}) \leftarrow \mathsf{dlog}_{\widetilde{U}^{(i)}}(\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}_{\mathsf{os}})$
$\mathbf{if}\ \mathsf{info}^* \neq \mathsf{info}[i^*_{\mathsf{pb}}]\ \vee\ \exists i \in [J^*+1] : \mathsf{BS}_{\mathrm{alg}}.\mathsf{Ver}(\mathsf{vk}, \mathsf{info}^*, \mu^{(i)}, \sigma^{(i)}) = 0$
  $\vee\ \exists i_0 \neq i_1 \in [J^*+1] : \mathsf{HQry}^{(i_0)} = \mathsf{HQry}^{(i_1)}$
  $\vee\ \forall i_0 \neq i_1 \in [J^*+1] : \overline{A}^{(i_0)}_{\mathsf{os}} \neq \overline{A}^{(i_1)}_{\mathsf{os}}$
  $\vee\ \exists i \in [J^*+1] : \overline{A}^{(i)}_{\mathsf{os}} \neq \phi^{\mathsf{os}}_{\mathsf{pp},\mathbb{x}_{\mathsf{os}}}(\tilde{z}^{(i)}_{\mathsf{os}} - \boldsymbol{\alpha}^{(i)}_{\mathsf{os}}) - (\tilde{c}^{(i)}_{\mathsf{os}} - \gamma^{(i)}_{\mathsf{os}}) \cdot W$
    $\mathbf{then\ return}\ \bot$
$(i_0, i_1) \leftarrow \min\{(i_0, i_1) : i_0 \neq i_1\ \wedge\ \overline{A}^{(i_0)}_{\mathsf{os}} = \overline{A}^{(i_1)}_{\mathsf{os}}\}$
$\mathbf{if}\ \gamma^{(i_0)}_{\mathsf{os}} - \tilde{c}^{(i_0)}_{\mathsf{os}} = \gamma^{(i_1)}_{\mathsf{os}} - \tilde{c}^{(i_1)}_{\mathsf{os}}\ \mathbf{then\ return}\ \bot$
$\boldsymbol{w}' \leftarrow (\gamma^{(i_0)}_{\mathsf{os}} - \gamma^{(i_1)}_{\mathsf{os}} + \tilde{c}^{(i_1)}_{\mathsf{os}} - \tilde{c}^{(i_0)}_{\mathsf{os}})^{-1} \cdot (\tilde{z}^{(i_1)}_{\mathsf{os}} - \boldsymbol{\alpha}^{(i_1)}_{\mathsf{os}} - \tilde{z}^{(i_0)}_{\mathsf{os}} + \boldsymbol{\alpha}^{(i_0)}_{\mathsf{os}})$
$\mathbf{return}\ \boldsymbol{w}'$

Fig. 10. Adversary $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ for the proof of Lemma 6.4. We simplify the description of S_2 be removing the sid tracking and assume that the second time an sid is queried to S_2, $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ simply return $\bot$. The simulation of oracles is described in Figure 11.

Oracle $\mathrm{S}_1(\mathsf{info})$:

$J \leftarrow J + 1$
$(D_2, D_3) \leftarrow \mathsf{H}(\mathsf{info})\ ; i_{\mathsf{info}} = \arg_{i\in[i_{\mathsf{pb}}]}\{\mathsf{info} = \mathsf{info}[i]\}$
$\mathbb{x}_{\mathsf{dleq}} = (D_1, D_2, D_3)\ ; c_{\mathsf{dleq},J} \leftarrow\!\$\ \mathbb{Z}_p$
$(A_{\mathsf{dleq},J}, z_{\mathsf{dleq},J}) \leftarrow\!\$\ \Sigma_{\mathsf{dleq}}.\mathsf{Sim}(\mathsf{pp}, \mathbb{x}_{\mathsf{dleq}}, c_{\mathsf{dleq},J})$
$A_{\mathsf{os},J} \leftarrow\!\$\ \bar{\mathrm{S}}_1()$
$u_J \leftarrow\!\$\ \mathbb{Z}^*_p\ ; U_J \leftarrow u_J G\ ; V_J \leftarrow zU_J$
$S_J \leftarrow u_J(x_0 G + x_1 A_{\mathsf{os},J})\ ; T_J \leftarrow u_J(y_0 G + y_1 A_{\mathsf{os},J})$
$\sigma_{\mathsf{mac},J} \leftarrow (U_J, V_J, S_J, T_J)$
$\mathbb{x}_{\mathsf{mac}} = (\mathsf{ipp}, A_{\mathsf{os},J}, \sigma_{\mathsf{mac},J}, 0_{\mathbb{G}}, 0_{\mathbb{G}})$
$\mathbb{w}_{\mathsf{mac}} = (u_J^{-1}, \mathsf{msk})$
$(A_{\mathsf{mac},J}, \mathsf{st}_{\mathsf{mac},J}) \leftarrow \Sigma_{\mathsf{mac}}.\mathsf{Init}(\mathsf{pp}, \mathbb{x}_{\mathsf{mac}}, \mathbb{w}_{\mathsf{mac}})$
$\mathbf{if}\ i_{\mathsf{info}} = i^*_{\mathsf{pb}}\ \mathbf{then}\ J^* \leftarrow J^* + 1$
$\mathbf{return}\ (\sigma_{\mathsf{mac},J}, A_{\mathsf{os},J}, A_{\mathsf{mac},J}, A_{\mathsf{dleq},J})$

Oracle $\mathrm{S}_2(\mathrm{sid}, c_{\mathrm{sid}})$:

$\mathbf{parse}\ c_{\mathrm{sid}} = (c_{\mathrm{os,sid}}, c_{\mathrm{or,sid}})$
$z_{\mathrm{os,sid}} \leftarrow \bar{\mathrm{S}}_2(\mathrm{sid}, c_{\mathrm{os,sid}})$
$c_{\mathrm{mac,sid}} \leftarrow c_{\mathrm{or,sid}} - c_{\mathsf{dleq},\mathrm{sid}}$
$z_{\mathrm{mac,sid}} \leftarrow \Sigma_{\mathsf{mac}}.\mathsf{Resp}(\mathsf{st}_{\mathrm{mac,sid}}, c_{\mathrm{mac,sid}})$
$\mathbf{return}\ (c_{\mathrm{mac,sid}}, z_{\mathrm{os,sid}}, z_{\mathrm{mac,sid}}, z_{\mathsf{dleq},\mathrm{sid}})$

Oracle $\mathsf{H}(\mathrm{str})$:

$\mathbf{if}\ \mathsf{H}[\mathrm{str}] = \bot\ \mathbf{then}\ \mathsf{H}[\mathrm{str}] \leftarrow\!\$\ \mathbb{Z}^3_p$
$\mathbf{return}\ \mathsf{H}[\mathrm{str}]$

Oracle $\mathsf{H}_{\mathsf{pb}}(\mathsf{info})$:

$\mathbf{if}\ \mathsf{H}_{\mathsf{pb}}[\mathsf{info}] = \bot\ \mathbf{then}$
  $i_{\mathsf{pb}} \leftarrow i_{\mathsf{pb}} + 1$
  $\mathsf{info}[i_{\mathsf{pb}}] \leftarrow \mathsf{info}$
  $\mathsf{H}_{\mathsf{pb}}[\mathsf{info}] \leftarrow\!\$\ \mathbb{G}^2$
$\mathbf{return}\ \mathsf{H}_{\mathsf{pb}}[\mathsf{info}]$

Fig. 11. Continuation of the description of adversary $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ in Figure 10 for the proof of Lemma 6.4. We simplify the description of S_2 be removing the sid tracking and assume that the second time an sid is queried to S_2, $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ simply return $\bot$.

Alg. $\mathcal{A}_{\mathsf{NonceColl}}(\mathsf{inp}, \boldsymbol{h} \in \mathbb{Z}_p^{Q_\mathsf{H}}; \rho_\mathcal{A})$:

1 : **parse** $\mathsf{inp} = (\mathsf{pp}, \mathsf{ipp}, \mathsf{msk}, \mathsf{td}, \boldsymbol{w}, (\rho_{\mathrm{S},j})_{j\in[Q_\mathrm{S}]}, (D_{2,i}, D_{3,i})_{i\in[Q_\mathsf{pb}]}, (\widetilde{c}_i)_{i\in[Q_\mathrm{H}]}, i^*_\mathsf{pb})$

2 : **parse** $\mathsf{pp} = (H, \mathsf{mpp} = (F, \mathsf{mck}), \mathsf{bck}, D_1)$, $\mathsf{ipp} = (\mathrm{cmt}_0, X_1, Y_1, Z)$

3 : **for** $j \in [Q_\mathrm{S}]$: **parse** $\rho_{\mathrm{S},j} = (u_j, \rho_{\mathsf{os},j}, \rho_{\mathsf{mac},j}, \rho_{\mathsf{dleq},j}, c_{\mathsf{dleq},j})$

4 : **for** $i \in [Q_\mathrm{H}]$: **parse** $\widetilde{c}_i = (\widetilde{c}_{\mathsf{os},i}, \widetilde{c}_{\mathsf{or},i})$; $\widetilde{c}'_i \leftarrow (\widetilde{c}_{\mathsf{os},i}, \widetilde{c}_{\mathsf{or},i}, h_i)$

5 : $W \leftarrow w_0 G + w_1 G$

6 : $i_\mathsf{pb}, i_\mathrm{H}, J, J^* \leftarrow 0$ // Initialize counters

7 : $(\mathsf{info}^*, (\mu^{(i)}, \sigma^{(i)})_{i\in[J^*+1]}) \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{S}_1,\mathrm{S}_2,\mathsf{H},\mathsf{H}_\mathsf{pb}}(\mathsf{pp}, \mathsf{vk} = (\mathsf{ipp}, W); \rho_\mathcal{A})$ // Run $\mathcal{A}$

8 : **for** $i \in [J^* + 1]$:

9 : **parse** $\sigma^{(i)} = (\widetilde{M}^{(i)}, \widetilde{\sigma}^{(i)}_\mathsf{mac}, \mathrm{cmt}^{(i)}_\mathsf{rnd}, (\widetilde{A}^{(i)}_\mathsf{xxx}, \widetilde{c}^{(i)}_\mathsf{xxx}, \widetilde{z}^{(i)}_\mathsf{xxx})_{\mathsf{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$

10 : $\mathsf{HQry}^{(i)} \leftarrow (\widetilde{M}^{(i)}, \widetilde{\sigma}^{(i)}_\mathsf{mac}, \mathrm{cmt}^{(i)}_\mathsf{rnd}, \widetilde{A}^{(i)}_\mathsf{os}, \widetilde{A}^{(i)}_\mathsf{mac}, \widetilde{A}^{(i)}_\mathsf{dleq}, \widetilde{A}^{(i)}_\mathsf{rnd}, \mu^{(i)}, \mathsf{info}^*)$

11 : $\mathsf{mes}^{(i)}_\mathsf{rnd} = (\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}_\mathsf{os}, \overline{R}^{(i)}_x, \overline{R}^{(i)}_y, \overline{R}^{(i)}_M, \overline{A}^{(i)}_\mathsf{os}) \leftarrow \mathsf{C}_\mathsf{rnd}.\mathsf{Extract}(\mathsf{td}, \mathrm{cmt}^{(i)}_\mathsf{rnd})$

12 : **if** $\mathsf{info}^* \neq \mathsf{info}[i^*_\mathsf{pb}] \ \vee\ \exists i \in [J^* + 1] : \mathsf{BS}_\mathrm{alg}.\mathsf{Ver}(\mathsf{vk}, \mathsf{info}^*, \mu^{(i)}, \sigma^{(i)}) = 0$ // Abort if $\neg\mathsf{NonceColl}$

13 : $\vee\ \exists i_0 \neq i_1 \in [J^* + 1] : \mathsf{HQry}^{(i_0)} = \mathsf{HQry}^{(i_1)}$

14 : $\vee\ \forall i_0 \neq i_1 \in [J^* + 1] : \overline{A}^{(i_0)}_\mathsf{os} \neq \overline{A}^{(i_1)}_\mathsf{os}$ **then return** $\bot$

15 : $(i_0, i_1) \leftarrow \min\{(i_0, i_1) : i_0 \neq i_1 \ \wedge\ \overline{A}^{(i_0)}_\mathsf{os} = \overline{A}^{(i_1)}_\mathsf{os}\}$

16 : **if** $(i_0, i_1) = \bot$ **then return** $\bot$

17 : **for** $b \in \{0, 1\} : I_b \leftarrow \arg_i\{\mathrm{str}[i] = \mathsf{HQry}^{(i_b)}\}$

18 : **return** $(I = (I_0, I_1), (\sigma^{(i_b)})_{b\in\{0,1\}})$

Fig. 12. Wrapper algorithm $\mathcal{A}_{\mathsf{NonceColl}}$ for the proof of Lemma 6.2. The oracles simulation are given in Figure 13. We denote internal states of $\mathcal{A}_{\mathsf{NonceDist}}$ $\mathsf{info}[\cdot], \mathrm{str}[\cdot]$ for mappings from indices $i \in [Q_\mathsf{pb}]$ and $i \in [Q_\mathsf{H}]$ to the i-th query to H_pb and H, respectively; additionally, $\mathsf{H}_\mathsf{pb}[\cdot], \mathsf{H}[\cdot]$ are mappings from query to output of the corresponding RO. We assume that the tuple $(\mathbb{G}, p, G)$ is implicit in pp.

that the transcript of Σ_os is simulated with its oracle access in the game PR or CR. At the end of the game, $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ extracts $\boldsymbol{w}'$ by (inefficiently) computing the discrete logarithms of $\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}$ for $i \in \{i_0, i_1\}$. This particular step does not require rewinding as we are only analyzing specific probabilities, and so we do not run into the issues with Okamoto-Schnorr OMUF as discussed in Section 1.1.

Now we argue the claimed equalities on success probabilities. Notice that $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ only outputs $\boldsymbol{w}'$ if $\mathcal{A}$ wins in game $\mathbf{G}^\mathcal{A}_{\mathrm{A}.3}$, and that if $\mathcal{A}$ wins, $\phi^\mathsf{os}_\mathsf{pp}(\boldsymbol{w}') = W$ (i.e., $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ wins in PR game). Thus, the first equality is satisfied. Similarly, the additional winning condition of $\mathbf{G}^\mathcal{A}_{\mathrm{A}.4}$ is that $\boldsymbol{w}' \neq \boldsymbol{w}$, so the second equality follows, proving the claim. □

Finally, we bound $\mathsf{Pr}[\mathbf{G}^\mathcal{A}_{\mathrm{A}.4} \Rightarrow 1]$ as in the following lemma statement. We will prove the lemma in three steps: (1) defining a wrapper algorithm to apply the forking lemma, (2) defining the reduction to DL, and (3) analyzing the reduction using the forking lemma. In particular, we will require using the forking algorithm Fork_3 from Lemma 2.19 to extract the blinding factors $\boldsymbol{\alpha}^{(i)}_\mathsf{os}, \gamma^{(i)}_\mathsf{os}$ for $i \in \{i_0, i_1\}$ from the i_0-th and i_1-th forgeries. Thus, this requires us to run $\mathcal{A}$ for 3 times and fork once at the H queries corresponding to each of the forgery.

Lemma 6.6 (Forking for NonceColl). *There exists an adversary $\mathcal{B}_\mathsf{dl}$ running in time $T_\mathrm{alg} \approx 3T_\mathcal{A}$ such that*

$$\mathsf{Pr}[\mathbf{G}^\mathcal{A}_{\mathrm{A}.4} \Rightarrow 1] \leqslant \sqrt[3]{32Q_\mathrm{H}^2 \cdot \mathsf{Adv}^\mathsf{dl}_{\mathsf{GGen}}(\mathcal{B}_\mathsf{dl}, \lambda)} + \frac{4Q_\mathrm{H}}{p} .$$

We particularly note that the theorem holds, even if $\mathcal{A}$ is computational. This is because our reduction does not perform the game's inefficient checks.

Proof (of Lemma 6.6). We first define the wrapper algorithm on which we will apply the forking lemma.

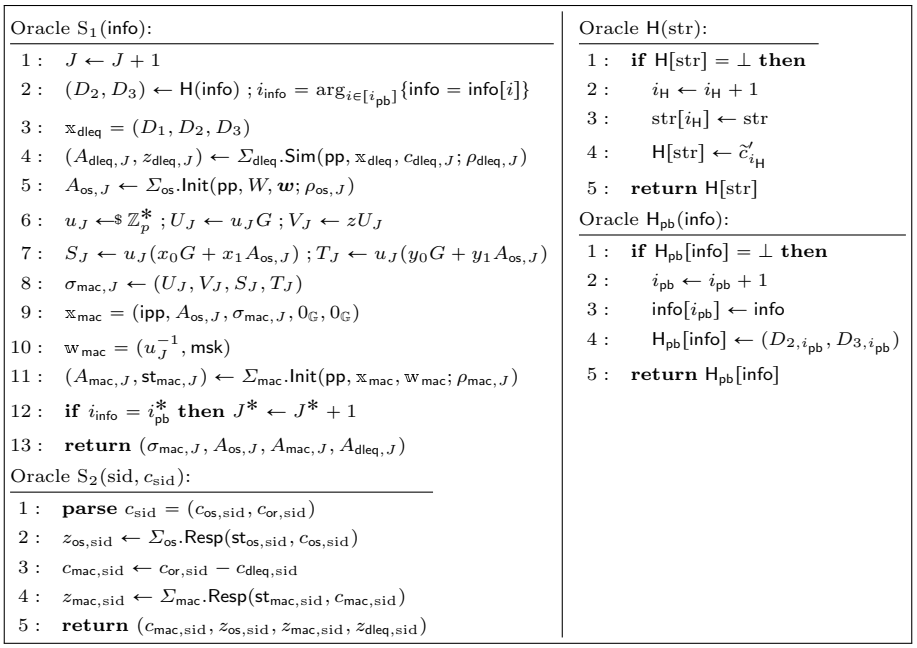

Fig. 13. Oracle simulation by $\mathcal{A}_{\mathsf{NonceColl}}$ extended from Figure 12. We simplify the description of S_2 be removing the sid tracking and assume that the second time an sid is queried to S_2, $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ simply return $\perp$.

WRAPPER $\mathcal{A}_{\mathsf{NonceColl}}$. The wrapper $\mathcal{A}_{\mathsf{NonceColl}}$ (defined in Figures 12 and 13) simulates the game $\mathbf{G}_{\mathrm{A}.4}$ to $\mathcal{A}$. In particular, it takes as input the public parameters pp as generated in $\mathbf{G}_{\mathrm{A}.4}$, ipp generated along with msk via $\mathsf{M}_{\mathsf{alg}}.\mathsf{KeyGen}$, extraction trapdoor td, witness $\boldsymbol{w} \in \mathbb{Z}_p^2$ for Σ_{os}, and prepared randomness $((\rho_{\mathrm{S},j})_{j\in[Q_{\mathrm{S}}]}, (D_{2,i}, D_{3,i})_{i\in[Q_{\mathsf{pb}}]}, (\widetilde{c}_i)_{i\in[Q_{\mathsf{H}}]}, i^*_{\mathsf{pb}})$ to be used for oracle simulation along with $\boldsymbol{h}$. It then runs $\mathcal{A}$ and simulate the game as follows:

Input to $\mathcal{A}$: The wrapper forwards pp and $\mathsf{vk} = (\mathsf{ipp}, W = w_0 G + w_1 H)$ to $\mathcal{A}$.
Oracle simulation: We list the corresponding simulation below:

- **Random oracles $\mathsf{H}, \mathsf{H}_{\mathsf{pb}}$:** $(\widetilde{c}_i \in \mathbb{Z}_p^2)_{i\in[Q_{\mathsf{H}}]}$ and $\boldsymbol{h}$ are used to simulate H where the i-th query is replied with $(\widetilde{c}_i, h_i) \in \mathbb{Z}_p^3$, while $(D_{2,i}, D_{3,i})$ is used to simulate the i-th query to the random oracle H_{pb}.
- **Signing oracles $\mathbf{S}_1, \mathbf{S}_2$:** The wrapper uses $\rho_{\mathrm{S},j} = (u_j, \rho_{\mathsf{os},j}, \rho_{\mathsf{mac},j}, \rho_{\mathsf{dleq},j}, c_{\mathsf{dleq},j})$ are the randomness used to compute the transcript for the j-th signing session. In particular, u_j is the randomness for computing σ_{mac} along with msk, $\rho_{\mathsf{os},j}$ is the randomness for Σ_{os} transcript computed using $\boldsymbol{w}$, $\rho_{\mathsf{mac},j}$ is the randomness for Σ_{mac} transcript, and $\rho_{\mathsf{dleq},j}, c_{\mathsf{dleq},j}$ are used to simulate the Σ_{dleq} transcript. Note that these corresponds to how the oracles are simulated in $\mathbf{G}_{\mathrm{A}.4}$.

Output of $\mathcal{A}_{\mathsf{NonceColl}}$: The wrapper checks where the event $\mathsf{NonceColl}$ occurs (it uses i^*_{pb} from the input as the guessed index of info^*). If so, it outputs the corresponding indices $I_0, I_1 \in [Q_{\mathsf{H}}]$ to the H query in the verification of the i_0-th and i_1-th forgeries along with $\sigma^{(i_0)}$, $\sigma^{(i_1)}$, respectively. Note that the two forgeries i_0, i_1 exist due to how $\mathsf{NonceColl}$ is defined If $\mathsf{NonceColl}$ does not occur, it returns $\perp$.

We note that the wrapper *does not* check the additional winning condition introduced in $\mathbf{G}_{\mathrm{A}.2}$ to $\mathbf{G}_{\mathrm{A}.4}$. This is due to these checks requiring inefficient discrete logarithm computation. However, we remark that if $\mathcal{A}$ wins in game $\mathbf{G}_{\mathrm{A}.4}$, then the output $(I = (I_0, I_1), \sigma = (\sigma_0, \sigma_1)) \leftarrow \mathcal{A}_{\mathsf{NonceColl}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})$ also satisfies the checks introduced in those games.

In particular, we consider the event Good defined over the output $\sigma = (\sigma_0, \sigma_1)$ of $\mathcal{A}_{\mathsf{NonceColl}}$. We use the same notations as in $\mathbf{G}_{\mathrm{A}.4}$ and denote $(\cdot)^{(i_b)}$ to values corresponding to σ_b. We define Good to be the event that $(\gamma_{\mathsf{os}}^{(i_0)} - \gamma_{\mathsf{os}}^{(i_1)} + \widetilde{c}_{\mathsf{os}}^{(i_1)} - \widetilde{c}_{\mathsf{os}}^{(i_0)}) \cdot \boldsymbol{w} \neq \widetilde{z}_{\mathsf{os}}^{(i_0)} - \boldsymbol{\alpha}_{\mathsf{os}}^{(i_0)} - \widetilde{z}_{\mathsf{os}}^{(i_1)} + \boldsymbol{\alpha}_{\mathsf{os}}^{(i_1)}$ where $\boldsymbol{\alpha}_{\mathsf{os}}^{(i)}, gamma_{\mathsf{os}}^{(i)}$ are in the exponents. Note that the occurrence of Good cannot be verified efficiently, and Good is implied by $\mathcal{A}$ winning

Alg. $\mathcal{B}_{\mathsf{dl}}(\mathsf{pp}_{\mathsf{os}} = (\mathbb{G}, p, G, H))$:

1 : $\mathsf{mpp} \leftarrow\!\!\$\ \mathsf{M}_{\mathsf{alg}}.\mathsf{Setup}(\mathbb{G}, p, G)$

2 : $(\mathsf{bck}, \mathsf{td}) \leftarrow\!\!\$\ \mathsf{C}_{\mathsf{rnd}}.\mathsf{ExtSetup}(\mathbb{G}, p, G)\ ; \boldsymbol{w} \leftarrow\!\!\$\ \mathbb{Z}_p^2$

3 : $\mathsf{pp} \leftarrow (H, \mathsf{mpp}, \mathsf{bck})$

4 : $(\mathsf{msk}, \mathsf{ipp}) \leftarrow\!\!\$\ \mathsf{M}_{\mathsf{alg}}.\mathsf{KeyGen}(\mathsf{mpp})$

5 : **for** $j \in [Q_{\mathrm{S}}]$:

6 : $\quad u_j \leftarrow\!\!\$\ \mathbb{Z}_p^*\ ; \rho_{\mathsf{os},j} \leftarrow\!\!\$\ \mathbb{Z}_p^2$

7 : $\quad \rho_{\mathsf{mac},j} \leftarrow\!\!\$\ \mathbb{Z}_p^6 \times \mathsf{C}_{\mathsf{mac}}.\mathcal{R}\ ; \rho_{\mathsf{dleq},j}, c_{\mathsf{dleq},j} \leftarrow\!\!\$\ \mathbb{Z}_p$

8 : $\quad \rho_{\mathrm{S},j} \leftarrow (u_j, \rho_{\mathsf{mac},j}, \rho_{\mathsf{dleq},j}, c_{\mathsf{mac},j})$

9 : $D_{2,1}, D_{3,1}, \ldots, D_{2,Q_{\mathsf{pb}}}, D_{3,Q_{\mathsf{pb}}} \leftarrow\!\!\$\ \mathbb{G}\ ; \tilde{c}_1, \ldots, \tilde{c}_{Q_{\mathsf{H}}} \leftarrow\!\!\$\ \mathbb{Z}_p^2\ ; i_{\mathsf{pb}}^* \leftarrow\!\!\$\ [Q_{\mathsf{pb}}]$

10 : $\mathsf{inp} \leftarrow (\mathsf{pp}, \mathsf{ipp}, \mathsf{msk}, \mathsf{td}, \boldsymbol{w}, (\rho_{\mathrm{S},j})_{j\in[Q_{\mathrm{S}}]}, (D_{2,i}, D_{3,i})_{i\in[Q_{\mathsf{pb}}]}, (\tilde{c}_i)_{i\in[Q_{\mathsf{H}}]}, i_{\mathsf{pb}}^*)$

11 : $(\mathsf{frk}, \sigma = (\sigma_0, \sigma_1), \sigma' = (\sigma_0', \sigma_1')) \leftarrow\!\!\$\ \mathsf{Fork}_3^{\mathcal{A}_{\mathsf{NonceColl}}}(\mathsf{inp})$

12 : **if** $\mathsf{frk} = \bot$ **then return** $\bot$

13 : **for** $b \in \{0, 1\}$:

14 : $\quad$ **parse** $\sigma_b = (\widetilde{M}, \tilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, (\tilde{A}_{\mathsf{xxx}}, \tilde{c}_{\mathsf{xxx}}, \tilde{z}_{\mathsf{xxx}})_{\mathsf{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$

15 : $\quad$ **parse** $\sigma_b' = (\widetilde{M}, \tilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, (\tilde{A}_{\mathsf{xxx}}, \tilde{c}'_{\mathsf{xxx}}, \tilde{z}'_{\mathsf{xxx}})_{\mathsf{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$

16 : $\quad$ **parse** $\tilde{\sigma}_{\mathsf{mac}} = (\widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{S}_1, \widetilde{T}_0, \widetilde{T}_1)$

17 : $\quad \mathbb{x}_{\mathsf{rnd}} = (W = w_0 G + w_1 H, X_1, Y_1, \tilde{A}_{\mathsf{os}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{M}, \widetilde{U}, \widetilde{S}_1, \widetilde{T}_1)$

18 : $\quad (\alpha_0, \alpha_1, \gamma_{\mathsf{os}}, r_x, r_y, r_M, \rho_{\mathsf{rnd}}) \leftarrow \Sigma_{\mathsf{rnd}}.\mathsf{Ext}(\mathbb{x}_{\mathsf{rnd}}, \tilde{A}_{\mathsf{rnd}}, \tilde{c}_{\mathsf{rnd}}, \tilde{z}_{\mathsf{rnd}}, \tilde{c}'_{\mathsf{rnd}}, \tilde{z}'_{\mathsf{rnd}})$

19 : $\quad z_{\mathsf{os}}^{(b)} \leftarrow \tilde{z}_{\mathsf{os}} - (\alpha_0, \alpha_1)\ ; \quad c_{\mathsf{os}}^{(b)} \leftarrow \tilde{c}_{\mathsf{os}} - \gamma_{\mathsf{os}}$

20 : **if** $c_{\mathsf{os}}^{(0)} = c_{\mathsf{os}}^{(1)}$ **then return** $\bot$

21 : $\boldsymbol{w}' \leftarrow (z_{\mathsf{os}}^{(1)} - z_{\mathsf{os}}^{(0)}) \cdot (c_{\mathsf{os}}^{(1)} - c_{\mathsf{os}}^{(0)})^{-1}$

22 : **if** $\boldsymbol{w} = \boldsymbol{w}'$ **then return** $\bot$

23 : **return** $(w_0' - w_0) \cdot (w_1 - w_1')^{-1}$

Fig. 14. Reduction $\mathcal{B}_{\mathsf{dl}}$.

in game $\mathbf{G}_{\mathrm{A}.4}$. Furthermore, for our reduction's analysis, we define an inefficient wrapper $\mathcal{A}'_{\mathsf{NonceColl}}$ which runs $\mathcal{A}_{\mathsf{NonceColl}}$ but also checks for Good on the output, and returns $\bot$ if Good does not occur. Hence, we obtain

$$\begin{aligned}
&\mathsf{Pr}_{\mathsf{inp},\boldsymbol{h},\rho_{\mathcal{A}}}[\bot \neq (I,\sigma) \leftarrow \mathcal{A}'_{\mathsf{NonceColl}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})] \\
=&\mathsf{Pr}_{\mathsf{inp},\boldsymbol{h},\rho_{\mathcal{A}}}[\bot \neq (I,\sigma) \leftarrow \mathcal{A}_{\mathsf{NonceColl}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}}) \ \wedge\ \mathsf{Good}] \\
\geqslant&\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.4} \Rightarrow 1].
\end{aligned}$$

The first equality follows from the definition of $\mathcal{A}'_{\mathsf{NonceColl}}$ and the second follows from definition of $\mathbf{G}_{\mathrm{A}.4}$.

Rewinding reduction $\mathcal{B}_{\mathsf{dl}}$. We now consider the reduction $\mathcal{B}_{\mathsf{dl}}$ (given in Figure 14) which runs $\mathcal{A}_{\mathsf{NonceColl}}$ for three times via the forking algorithm Fork_3 with the input to $\mathcal{A}_{\mathsf{NonceColl}}$ set up as in $\mathbf{G}_{\mathrm{A}.4}$ along with the required randomness for oracle simulation.

Remark 6.7. Recall that Fork_3 runs the wrapper once (on uniformly random $\boldsymbol{h} \in \mathbb{Z}_p^{Q_{\mathsf{H}}}$), obtains two indices (I_0, I_1) along with the corresponding outputs $(\mathsf{aux}_{I_0}, \mathsf{aux}_{I_1})$ (which are individual signatures in our cases), and runs the wrapper again twice on $\boldsymbol{h}'$ and $\boldsymbol{h}''$. Here, $\boldsymbol{h}'$ is uniformly sampled conditioning on the first $I_0 - 1$ values agreeing with $\boldsymbol{h}$, and $\boldsymbol{h}''$ is uniformly sampled conditioning on the first $I_1 - 1$ values agreeing with $\boldsymbol{h}$. Then, Fork_3 succeeds if (1) $h_{I_0} \neq h'_{I_0}$, (2) $h_{I_1} \neq h''_{I_1}$, (3) the run using $\boldsymbol{h}'$ outputs I_0 as one of the index, and similarly the run using $\boldsymbol{h}''$ outputs I_1. Also, Fork_3 outputs $\mathsf{aux} = (\mathsf{aux}_{I_0}, \mathsf{aux}_{I_1})$ and $\mathsf{aux}' = (\mathsf{aux}'_{I_0}, \mathsf{aux}''_{I_1})$, where the latter is parsed from the output of the second and third runs of the wrapper, respectively.

Also, we need to rewind the adversary twice in order to extract the blinding factors from two signatures. Rewinding once does not suffice as it will only allow the reduction to extract from only one of the forgery, and we need the blinding factors from both to extract the discrete logarithm.

Within $\mathcal{B}_{\mathsf{dl}}$, if $\mathsf{Fork}_3^{\mathcal{A}_{\mathsf{NonceColl}}}$ does not succeed, it aborts. Otherwise, Fork_3 succeeds and returns $\sigma = (\sigma_0, \sigma_1)$ and $\sigma' = (\sigma'_0, \sigma'_1)$. Here, σ contains the i_0-th and i_1-th forgeries from the first run of $\mathcal{A}_{\mathsf{NonceColl}}$. On the other hand, σ'_0 comes from the second run of $\mathcal{A}_{\mathsf{NonceColl}}$ within Fork_3 and corresponds to the same RO query used for verifying σ_0. Similarly, σ'_1 comes from the third run of $\mathcal{A}_{\mathsf{NonceColl}}$ and corresponds to the same RO query used for verifying σ_1. Then, $\mathcal{B}_{\mathsf{dl}}$ does the following:

1. From both pairs (σ_b, σ'_b) for $b \in \{0,1\}$, extract the underlying witnesses using the special-soundness extractor $\Sigma_{\mathsf{rnd}}.\mathsf{Ext}$. Note that because Fork_3 succeeds, the two challenges for Σ_{rnd} transcripts are distinct.
2. Then, for each $b \in \{0,1\}$, use the extracted witness $\mathbb{w}_{\mathsf{rnd}} = (\alpha_0, \alpha_1, \gamma_{\mathsf{os}}, r_x, r_y, r_M, \rho_{\mathsf{rnd}})$ (omitting the superscripts (b) for readability) to compute the original Okamoto-Schnorr response and challenge
$$z_{\mathsf{os}}^{(b)} = \widetilde{z}_{\mathsf{os}} - (\alpha_0, \alpha_1)\ , c_{\mathsf{os}}^{(b)} = \widetilde{c}_{\mathsf{os}} - \gamma_{\mathsf{os}}\ ,$$
where $\widetilde{z}_{\mathsf{os}}$ and $\widetilde{c}_{\mathsf{os}}$ are parsed from σ_b.
3. Abort if $c_{\mathsf{os}}^{(0)} = c_{\mathsf{os}}^{(1)}$. Otherwise, compute $\boldsymbol{w}' = (z_{\mathsf{os}}^{(1)} - z_{\mathsf{os}}^{(0)}) \cdot (c_{\mathsf{os}}^{(1)} - c_{\mathsf{os}}^{(0)})^{-1}$, i.e., extracting the witness for Σ_{os} via special soundness.
4. Abort if $\boldsymbol{w} = \boldsymbol{w}'$. Otherwise, compute $h = (w'_0 - w_0) \cdot (w_1 - w'_1)^{-1}$. Note that with $W = w_0 G + w_1 H = w'_0 G + w'_1 H$, we can see that $h = \mathsf{dlog}_G H$.

Analysis of $\mathcal{B}_{\mathsf{dl}}$. By how $\mathcal{B}$ is constructed, we can see that if $\mathcal{B}$ does not abort, it wins in the DL game. Accordingly, the event that $\mathcal{B}$ does not abort corresponds to the event that in the first run of $\mathcal{A}_{\mathsf{NonceColl}}$ within Fork_3 outputs $\sigma = (\sigma_0, \sigma_1)$ satisfying the event Good defined above; equivalently, the inefficient wrapper $\mathcal{A}'_{\mathsf{NonceColl}}$ does not abort. Therefore,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{dl}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{dl}}, \lambda) &\geqslant \mathsf{Pr}_{\mathsf{inp}}[\mathsf{Fork}_3^{\mathcal{A}'_{\mathsf{NonceColl}}}(\mathsf{inp}) \neq \bot] \\ &\geqslant \frac{\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.4} \Rightarrow 1]}{2} \left(\frac{\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.4} \Rightarrow 1]}{4Q_{\mathsf{H}}} - \frac{1}{p} \right)^2 ,\end{aligned}$$

where the last inequality follows from Lemma 2.19 and that $\mathsf{Pr}_{\mathsf{inp},\boldsymbol{h},\rho_{\mathcal{A}}}[\bot \neq (I, \sigma) \leftarrow \mathcal{A}'_{\mathsf{NonceColl}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})] \geqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{A}.4} \Rightarrow 1]$. Note that $\mathcal{B}_{\mathsf{dl}}$ runs in time roughly $3T_{\mathcal{A}}$, concluding the proof. □

6.3 Proof of Lemma 6.3 (Extracting MAC Forgeries from Distinct Nonces)

We will use the notations as in the proof of Theorem 3.5 in Section 6.1. Moreover, we number the signing sessions with indices $j \in [Q_{\mathrm{S}}]$ corresponding to the session where $\mathcal{A}$ makes the j-th call to the oracle S_1; accordingly, we refer to these indices as the session ID (sid in the OMSUF game)with j as well. We consider the following sequence of games $\mathbf{G}_{\mathrm{B}.k}$ for $k \in [0, 14]$. To help the readers navigate the proof, we explain the high-level detail of the games in this sequence:

- Game $\mathbf{G}_{\mathrm{B}.1}$ sets up the group element H while knowing the discrete logarithm. This will help later on when we give a reduction from UF-RMA security of $\mathsf{M}_{\mathsf{alg}}$ where the tagging oracle knows the discrete logarithm of the message A_{os}, which is the Okamoto-Schnorr nonce.
- Games $\mathbf{G}_{\mathrm{B}.2}$ to $\mathbf{G}_{\mathrm{B}.7}$: These games are used to deal with the "partially blind" part of the protocol. In particular, it simulates the signing queries on $\mathsf{info} \neq \mathsf{info}^*$ by computing $\mathsf{M}_{\mathsf{alg}}$ tags without using the secret key msk for these sessions, along with simulating Σ_{mac} side of the signing for these sessions.
- Games $\mathbf{G}_{\mathrm{B}.8}$ to $\mathbf{G}_{\mathrm{B}.10}$: Enforce certain structures on the blinded $\mathsf{M}_{\mathsf{alg}}$ tags $\widetilde{\sigma}^{(i)}_{\mathsf{mac}}$ in the forgery to ensure that the final reduction can recover a fresh and valid $\mathsf{M}_{\mathsf{alg}}$ tag.
- Games $\mathbf{G}_{\mathrm{B}.11}$ to $\mathbf{G}_{\mathrm{B}.12}$: Simulate the proof of validity of the MAC (Σ_{mac}), preparing for the reduction from UF-CMA security which does not know the secret key (aka witness used in Σ_{mac} protocol).
- Game $\mathbf{G}_{\mathrm{B}.13}$ to $\mathbf{G}_{\mathrm{B}.14}$: Syntactic changes to prepare for the rewinding argument, namely sampling the randomness at the beginning of the game and preparing $\mathsf{M}_{\mathsf{alg}}$ tags used in signing sessions with common message info^* in advance.

- In the final game, by definition of $\mathsf{NonceDist}$, there exists some index $i^* \in [\ell]$ (w.l.o.g. we assume i^* is the minimal such index) such that the extracted $\overline{A}_{\mathsf{os}}^{(i^*)}$ is one of the nonces not used in the signing sessions on info^*. Thus, the blinded MAC tag will correspond to a fresh MAC forgery; accordingly, we can rewind and reduce to UF-RMA security of the MAC. Note that the reduction needs rewinding to extract (1) the blinding factor of the blinded MAC in order to recover the unblinded MAC forgery, and (2) the discrete logarithm of $\overline{A}_{\mathsf{os}}^{(i^*)}$ which corresponds to our message for the MAC forgery. We note that due to the change in the previous game, our reduction will prepare the nonces and their MAC tags in advance, so as to avoid making extra MAC tag queries during the rewinding.

Game $\mathbf{G}_{\mathrm{B}.0}$ (Game $\mathbf{G}_4$ with winning condition $\mathsf{NonceDist}$): This game is identical to game $\mathbf{G}_4$ except that the winning condition is now $\mathsf{NonceDist}$. We can see that

$$\mathsf{Pr}_{\mathbf{G}_4^{\mathcal{A}}}[\mathsf{NonceDist}] = \mathsf{Pr}[\mathbf{G}_{\mathrm{B}.0}^{\mathcal{A}} \Rightarrow 1] \ .$$

Game $\mathbf{G}_{\mathrm{B}.1}$ (Sample H while knowing $h = \mathsf{dlog}_G H$, and compute $w' = \mathsf{dlog}_G W, r'_{\mathsf{os},j} = \mathsf{dlog}_G A_{\mathsf{os},j}$): This game sample $h \leftarrow_{\$} \mathbb{Z}_p$ and set $H \leftarrow hG$. Also, set $w' = w_0 + w_1 h = \mathsf{dlog}_G W$ along with $r'_{\mathsf{os},j} = r_{\mathsf{os},j,0} + r_{\mathsf{os},j,1} h$, where $r_{\mathsf{os},j,0}, r_{\mathsf{os},j,1}$ are randomness used for computing $A_{\mathsf{os},j}$ in the j-th signing session. It is easy to see that this change does not affect the adversary's advantage.

$$\mathsf{Pr}[\mathbf{G}_{\mathrm{B}.0}^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_{\mathrm{B}.1}^{\mathcal{A}} \Rightarrow 1] \ .$$

Game $\mathbf{G}_{\mathrm{B}.2}$ (Switch $\mathsf{H}(\mathsf{info})$ to be DLEQ tuples): Recall the guessed index i^*_{pb} introduced in $\mathbf{G}_2$. In this game, for the i-th query info_i to H_{pb} , the game samples $a_i \leftarrow_{\$} \mathbb{Z}_p$ and set $\mathsf{H}_{\mathsf{pb}}(\mathsf{info}_i) \leftarrow (a_i G, a_i D_1)$. Note that since the winning condition (i.e., event $\mathsf{NonceDist}$) is efficiently checkable, we can give a straightforward reduction $\mathcal{B}_{\mathsf{ddh},1}$ from the Q_{pb}-DDH problem with running time $T_{\mathsf{ddh},1} \approx T_{\mathcal{A}}$ such that

$$|\mathsf{Pr}[\mathbf{G}_{\mathrm{B}.1}^{\mathcal{A}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}_{\mathrm{B}.2}^{\mathcal{A}} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathsf{pb}}}(\mathcal{B}_{\mathsf{ddh},1}, \lambda) \ .$$

Game $\mathbf{G}_{\mathrm{B}.3}$ (Compute Σ_{dleq} honestly in signing oracles): In this game, for queries to S_1 and S_2 (with session ID $j \in [Q_{\mathrm{S}}]$) with tag info that corresponds to the i-th H_{pb} query with $(D_{2,i}, D_{3,i}) \leftarrow \mathsf{H}_{\mathsf{pb}}(\mathsf{info})$, the game computes $(A_{\mathsf{dleq},j}, c_{\mathsf{dleq},j}, z_{\mathsf{dleq},j})$ as follows:

(1) Compute $(A_{\mathsf{dleq},j}, \mathsf{st}_{\mathsf{dleq},j}) \leftarrow_{\$} \Sigma_{\mathsf{dleq}}.\mathsf{Init}(\mathsf{pp}, \mathbb{x}_{\mathsf{dleq}} = (D_1, D_{2,i}, D_{3,i}), \mathbb{w}_{\mathsf{dleq}} = a_i)$,
(2) Sample $c_{\mathsf{dleq},j} \leftarrow_{\$} \mathbb{Z}_p$
(3) Compute $z_{\mathsf{dleq},j} \leftarrow \Sigma_{\mathsf{dleq}}.\mathsf{Resp}(\mathsf{st}_{\mathsf{dleq},j}, c_{\mathsf{dleq},j})$

Due to perfect SHVZK of Σ_{dleq}, we have

$$\mathsf{Pr}[\mathbf{G}_{\mathrm{B}.2}^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_{\mathrm{B}.3}^{\mathcal{A}} \Rightarrow 1] \ .$$

Note also that in this game for session j's where the changes were made, the joint distribution of $(c_{\mathsf{dleq},j}, c_{\mathsf{mac},j})$ is identical to sampling $c_{\mathsf{mac},j} \leftarrow_{\$} \mathbb{Z}_p$ first and computing $c_{\mathsf{dleq},j} \leftarrow c_{\mathsf{or},j} - c_{\mathsf{mac},j}$.

Game $\mathbf{G}_{\mathrm{B}.4}$ (Simulate Σ_{mac} in signing oracles): In this game, for queries to S_1 and S_2 (with session ID $j \in [Q_{\mathrm{S}}]$) with tag info that corresponds to the i-th H_{pb} query, the game computes $(A_{\mathsf{mac},j}, c_{\mathsf{mac},j}, z_{\mathsf{mac},j})$ by: (1) sampling $c_{\mathsf{mac},j} \leftarrow_{\$} \mathbb{Z}_p$, (2) $(A_{\mathsf{mac},j}, z_{\mathsf{mac},j}) \leftarrow_{\$} \Sigma_{\mathsf{mac}}.\mathsf{Sim}(\mathsf{pp}, \mathbb{x}_{\mathsf{mac},j})$ with $\mathbb{x}_{\mathsf{mac},j} = (\mathsf{ipp}, A_{\mathsf{os},j}, U_j, V_j, S_j, T_j, 0_{\mathbb{G}}, 0_{\mathbb{G}})$, and (3) compute $c_{\mathsf{dleq},j} \leftarrow c_{\mathsf{or},j} - c_{\mathsf{mac},j}$ in S_2. Due to perfect SHVZK of Σ_{mac}, we have that

$$\mathsf{Pr}[\mathbf{G}_{\mathrm{B}.3}^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_{\mathrm{B}.4}^{\mathcal{A}} \Rightarrow 1] \ .$$

Game $\mathbf{G}_{\mathrm{B}.5}$ (Computed $\mathsf{M}_{\mathsf{alg}}$ tags without using msk when signing $\mathsf{info} \neq \mathsf{info}^*$): In this game, for queries to S_1 (with session ID $j \in [Q_{\mathrm{S}}]$) with tag info that does not correspond to the i^*_{pb}-th H_{pb} query, the game computes the $\mathsf{M}_{\mathsf{alg}}$ tag $\sigma_{\mathsf{mac},j}$ as

$$u_j \leftarrow_{\$} \mathbb{Z}_p^* \ , U_j \leftarrow u_j F \ , V_j \leftarrow u_j Z \ , S_j, T_j \leftarrow_{\$} \mathbb{G} \ .$$

The advantage change follows from the following claim. The key idea of the proof is that we can view the U_j, x_0U_j, y_0U_j portions of the MAC tags as a DDH instance with respect to the bases (G, x_0G, y_0G). Note that we do not require V_j to be uniformly random, since it can now be computed while only knowing ipp which contains Z.

The changes in this game is to prepare for the final reduction, which will only output honestly computed tags for signing sessions on info^*. Otherwise, notice that the MAC tags are, however, not bound to the common message info. If the tags given during signing of $\mathsf{info} \neq \mathsf{info}^*$ were valid, the adversary could potentially use these tags in the one-more forgery. Thus, making the reduction not successful.

Remark 6.8. We only make these changes for signing queries with $\mathsf{info} \neq \mathsf{info}^*$, because later on, we still need to switch $\mathsf{H}(\mathsf{info}^*)$ to be uniformly random to argue certain structures on the blinded MAC tags in the forgery via soundness of Σ_{mac} and Σ_{dleq}. For these steps, we still need the MAC tags in signing queries on info^* to be honestly generated to compute Σ_{mac} transcript honestly and simulate Σ_{dleq} during signing of info^*.

Lemma 6.9 (Uniformly random $\mathsf{M}_{\mathsf{alg}}$ tags). *There exists an adversary $\mathcal{B}_{\mathsf{ddh},2}$ playing the* QDDH *game running in time $T_{\mathsf{ddh},2} \approx T_{\mathcal{A}}$ and $\mathcal{B}_{\mathsf{hide}}$ also running in time $T_{\mathsf{hide}} \approx T_{\mathcal{A}}$ such that*

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.4} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.5} \Rightarrow 1] + 2 \cdot \mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathrm{S}}}(\mathcal{B}_{\mathsf{ddh},2}, \lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_{\mathsf{hide}}, \lambda) + \frac{2Q_{\mathrm{S}} + 1}{p} \ .$$

Proof (of Lemma 6.9). We consider the following sequence of games:

- **Game $\mathbf{G}_{\mathrm{B}.5.0}$ (Game $\mathbf{G}_{\mathrm{B}.4}$):** This game is identical to $\mathbf{G}_{\mathrm{B}.4}$.
- **Game $\mathbf{G}_{\mathrm{B}.5.1}$ (Sample $u_j \leftarrow\!\!\$\ \mathbb{Z}_p$ instead of $\mathbb{Z}_p^*$):** In this game, the signing oracle S_1, on its j-th query, samples $u_j \leftarrow\!\!\$\ \mathbb{Z}_p$ instead. Since uniform distributions over $\mathbb{Z}_p$ and $\mathbb{Z}_p^*$ have statistical distance of $1/p$, over Q_{S} queries, we have $|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.5.0} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.5.1} \Rightarrow 1]| \leqslant \frac{Q_{\mathrm{S}}}{p}$.
- **Game $\mathbf{G}_{\mathrm{B}.5.2}$ (Compute cmt_0 as a commitment of $(0,0)$):** In this game, the key generation generates $\mathrm{cmt}_0 = \mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (0,0); \rho)$ for $\rho \leftarrow\!\!\$\ \mathsf{C}_{\mathsf{mac}}.\mathcal{R}$ instead.

  We argue the advantage change via a reduction $\mathcal{B}_{\mathsf{hide}}$ to the hiding property of $\mathsf{C}_{\mathsf{mac}}$. In particular, notice that due to the change in games $\mathbf{G}_{\mathrm{B}.2}$ to $\mathbf{G}_{\mathrm{B}.4}$, the signing oracles simulate the Σ_{mac} side of the OR-proof, so they do not require ipp to be in the language induced by the linear map $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}$. Therefore, the reduction can simply simulate the game as in $\mathbf{G}_{\mathrm{B}.5.1}$. Hence, $|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.5.1} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.5.2} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_{\mathsf{hide}}, \lambda)$.
- **Game $\mathbf{G}_{\mathrm{B}.5.3}$ (Sample $S_j \leftarrow\!\!\$\ \mathbb{G}$ when signing $\mathsf{info} \neq \mathsf{info}^*$):** In this game, on the j-th query to the signing oracle S_1, if the common message info does not correspond to the input of the i^*_{pb}-th H_{pb} query (this means that either the i^*_{pb}-th query has not been made, or the i^*_{pb}-th query's input is not info), the game samples $S_j \leftarrow\!\!\$\ \mathbb{G}$ instead of computing it using msk.

  We now show that there exists a reduction $\mathcal{B}_{\mathsf{ddh},2}$ such that $|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.5.2} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.5.3} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathrm{S}}}(\mathcal{B}_{\mathsf{ddh},2}, \lambda)$. The reduction running $\mathcal{A}$ is defined as follows taking as input the QDDH instance $(G, A, (B_j, C_j)_{j \in [Q_{\mathrm{S}}]})$: (The dashed boxes denote transition from (B.5.2) to (B.5.3), while highlighted boxes denote transition from (B.5.3) to (B.5.4).)

  Setup pp and vk: The public parameters pp and portions of vk except ipp are computed as in $\mathbf{G}_{\mathrm{B}.5.2}$. We compute ipp by sampling $y_0 \leftarrow\!\!\$\ \mathbb{Z}_p$ $x_0 \leftarrow\!\!\$\ \mathbb{Z}_p$, $x_1, y_1, z, \rho_0 \leftarrow\!\!\$\ \mathbb{Z}_p$ and setting $\mathsf{ipp} = (\mathsf{C}_{\mathsf{mac}}.\mathsf{C}(\mathsf{mck}, (0,0); \rho_0), X_1 \leftarrow x_1F, Y_1 \leftarrow y_1F, Z = zH)$

  Signing oracles $\mathbf{S}_1, \mathbf{S}_2$: On the j-th ($j \in [Q_{\mathrm{S}}]$) signing session with tag info, The $\Sigma_{\mathsf{os}}, \Sigma_{\mathsf{mac}}, \Sigma_{\mathsf{dleq}}$ transcripts are computed as in $\mathbf{G}_{\mathrm{B}.5.2}$, i.e., honestly generated, simulated, and honestly generated, respectively. However, the $\mathsf{M}_{\mathsf{alg}}$ tag $\sigma_{\mathsf{mac},j}$ is computed as follows:
  - If info corresponds to the i^*_{pb}-th H_{pb} query, sample $u_j \leftarrow\!\!\$\ \mathbb{Z}_p$ and set $U_j \leftarrow u_jG, V_j \leftarrow zU_j$, $S_j \leftarrow u_jA + u_jx_1A_{\mathsf{os},j}, T_j \leftarrow u_jy_0G + u_jy_1A_{\mathsf{os},j}$ $S_j \leftarrow u_jx_0G + u_jx_1A_{\mathsf{os},j}, T_j \leftarrow u_jA + u_jy_1A_{\mathsf{os},j}$.

– Otherwise, recall that $A_{\mathsf{os},j}$ is computed honestly as $r_{\mathsf{os},j,0}G + r_{\mathsf{os},j,1}H$ and the reduction knows $h = \mathsf{dlog}_G H$, so it knows $r'_{\mathsf{os},j} = r_{\mathsf{os},j,0} + r_{\mathsf{os},j,1}h$. It sets $U_j \leftarrow B_j, V_j \leftarrow zU_j$, $\boxed{S_j \leftarrow C_j + x_1 r'_{\mathsf{os},j} U_j, T_j \leftarrow y_0 U_j + y_1 r'_{\mathsf{os},j} U_j}$ $\boxed{S_j \leftarrow\!\!\$\, \mathbb{G}, T_j \leftarrow C_j + y_1 r'_{\mathsf{os},j} U_j}$

Other oracles: They are simulated as in $\mathbf{G}_{\mathrm{B.5.2}}$.
Output bit: Here, $\mathcal{B}_{\mathsf{ddh},2}$ outputs 1 if $\mathcal{A}$ pass the (efficiently checkable) winning condition in $\mathbf{G}_{\mathrm{B.5.2}}$.
We can easily see that if $C_j = (\mathsf{dlog}_G A) \cdot B_j$, the view of $\mathcal{A}$ within $\mathcal{B}_{\mathsf{ddh},2}$ is identical to its view in $\mathbf{G}_{\mathrm{B.5.2}}$. Similarly, if $C_j \leftarrow\!\!\$\, \mathbb{G}$, S_j is uniformly random for j-th S_1 query with info that is not info^*.

- **Game $\mathbf{G}_{\mathrm{B.5.4}}$ (Sample $T_j \leftarrow\!\!\$\, \mathbb{G}$ when signing $\mathsf{info} \neq \mathsf{info}^*$):** In this game, the signing oracle S_1, on its j-th query, if the common message info does not correspond to the i^*_{pb}-th H_{pb}, the game samples $T_j \leftarrow\!\!\$\, \mathbb{G}$ instead of computing it using msk. The advantage change from (B.5.3) to (B.5.4) are justified with the reduction above in highlighted boxes. Thus, we have that $|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.3}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.4}} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{qddh}}_{\mathsf{GGen},Q_{\mathrm{S}}}(\mathcal{B}_{\mathsf{ddh},2}, \lambda)$.
- **Game $\mathbf{G}_{\mathrm{B.5.5}}$ (Compute cmt_0 as a commitment of (x_0, y_0)):** In this game, the key generation generates $\mathrm{cmt}_0 = \mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (x_0, y_0); \rho)$ for $\rho \leftarrow\!\!\$\, \mathsf{C}_{\mathsf{mac}}.\mathcal{R}$ instead. This simply undoes the change made in $\mathbf{G}_{\mathrm{B.5.2}}$.

  We argue the advantage change via a reduction $\mathcal{B}_{\mathsf{hide}}$ to the hiding property of $\mathsf{C}_{\mathsf{mac}}$. In particular, notice that due to the change in games $\mathbf{G}_{\mathrm{B.2}}$ to $\mathbf{G}_{\mathrm{B.4}}$, the signing oracles simulate the Σ_{mac} side of the OR-proof, so they do not require ipp to be in the language induced by the linear map $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}$. Therefore, the reduction can simply simulate the game as in $\mathbf{G}_{\mathrm{B.5.4}}$. Hence, $|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.4}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.5}} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_{\mathsf{hide}}, \lambda)$.
- **Game $\mathbf{G}_{\mathrm{B.5.6}}$ (Sample $u_j \leftarrow\!\!\$\, \mathbb{Z}_p^*$ instead of $\mathbb{Z}_p$):** In this game, the signing oracle S_1, on its j-th query, samples $u_j \leftarrow\!\!\$\, \mathbb{Z}_p^*$ instead. Similar to the transition from $\mathbf{G}_{\mathrm{B.5.0}}$ to $\mathbf{G}_{\mathrm{B.5.1}}$, we have $|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.5}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.6}} \Rightarrow 1]| \leqslant \frac{Q_{\mathrm{S}}}{p}$.
- **Game $\mathbf{G}_{\mathrm{B.5.7}}$ (Compute $U_j \leftarrow u_j F$ instead of $u_j G$ when signing $\mathsf{info} \neq \mathsf{info}^*$):** In this game, on its j-th query the signing oracle S_1 with the input info, if info does not correspond to the i^*_{pb}-th H_{pb} query's input, U_j and V_j are now computed as: sample $u_j \leftarrow \mathbb{Z}_p^*$, set $U_j \leftarrow u_j F, V_j \leftarrow u_j Z$. Note that except for when $F = 0_{\mathbb{G}}$, the view of $\mathcal{A}$ is still identical because $Z = zF$ so $V_j = u_j zF = zU_j$, so $|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.6}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5.7}} \Rightarrow 1]| \leqslant \frac{1}{p}$.

Note that $\mathbf{G}_{\mathrm{B.5.7}}$ is identical to $\mathbf{G}_{\mathrm{B.5}}$. □

Game $\mathbf{G}_{\mathrm{B.6}}$ (Compute Σ_{mac} honestly and simulate Σ_{dleq} for signing on tags info^*): For queries to S_1 and S_2 (with session ID $j \in [Q_{\mathrm{S}}]$) with tag info that corresponds to the i^*_{pb}-th H_{pb} query, this game computes Σ_{dleq} and Σ_{mac} transcripts as follows:

- $c_{\mathsf{dleq},j} \leftarrow\!\!\$\, \mathbb{Z}_p, (A_{\mathsf{dleq},j}, z_{\mathsf{dleq},j}) \leftarrow\!\!\$\, \Sigma_{\mathsf{dleq}}.\mathsf{Sim}(\mathsf{pp}, \mathbb{x}_{\mathsf{dleq},j}, c_{\mathsf{dleq},j})$ where $\mathbb{x}_{\mathsf{dleq},j} = (D_1, D_{2,i^*_{\mathsf{pb}}}, D_{3,i^*_{\mathsf{pb}}})$.
- $(A_{\mathsf{mac},j}, \mathsf{st}_{\mathsf{mac},j}) \leftarrow\!\!\$\, \Sigma_{\mathsf{mac}}.\mathsf{Init}(\mathsf{pp}, \mathbb{x}_{\mathsf{mac},j}, \mathbb{w}_{\mathsf{mac},j})$ where $\mathbb{x}_{\mathsf{mac},j} = (\mathsf{ipp}, A_{\mathsf{os},j}, U_j, V_j, S_j, T_j, 0_{\mathbb{G}}, 0_{\mathbb{G}})$ and $\mathbb{w}_{\mathsf{mac},j} = (u_j^{-1}, \mathsf{msk})$.
- In S_2, compute $c_{\mathsf{mac},j} \leftarrow c_{\mathsf{or},j} - c_{\mathsf{dleq}}, z_{\mathsf{mac}} \leftarrow \Sigma_{\mathsf{mac}}.\mathsf{Resp}(\mathsf{st}_{\mathsf{mac},j}, c_{\mathsf{mac},j})$.

This game simply reverts the changes made in $\mathbf{G}_{\mathrm{B.3}}$ and $\mathbf{G}_{\mathrm{B.4}}$ but only for signing sessions with tag info^*. Similar to the changes from game $\mathbf{G}_{\mathrm{B.2}}$ to $\mathbf{G}_{\mathrm{B.4}}$, we have that

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.5}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.6}} \Rightarrow 1] \,.$$

Game $\mathbf{G}_{\mathrm{B.7}}$ (Switch $\mathsf{H}_{\mathsf{pb}}(\mathsf{info}^*)$ to random): In this game, the output of the i^*_{pb}-th query to H_{pb} is now switched back to uniformly random, i.e., $(D_{2,i^*_{\mathsf{pb}}}, D_{3,i^*_{\mathsf{pb}}}) \leftarrow\!\!\$\, \mathbb{G}^2$. Note that since the Σ_{dleq} transcript for signing on a common message info corresponding to this query is simulated, we do not need the underlying discrete log to compute the transcript. Therefore, one can construct a reduction $\mathcal{B}_{\mathsf{ddh},3}$ from the DDH problem running in time roughly $T_{\mathcal{A}}$ such that

$$|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.6}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.7}} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ddh},3}, \lambda) \,.$$

From this point, we will enforce structure checks on the blinded MAC tags in the forgeries. We will start by introducing *inefficient checks* that require computing the discrete logarithm of $\widetilde{U}^{(i)}$. In particular, we want to check that the blinded MAC tags actually satisfy the relation defined by $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathbb{x}_{\mathsf{mac}}}$ (Equation (2)). Then, later on, relax this check to an *efficient check* that only uses the MAC secret key msk. This is possible due to the extracted group elements $(\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}_{\mathsf{os}}, \overline{R}^{(i)}_x, \overline{R}^{(i)}_y, \overline{R}^{(i)}_M)$ which embed the blinding factors with the base $\widetilde{U}^{(i)}$.

Game $\mathbf{G}_{\mathrm{B}.8}$ (Enforce (inefficient) check on $\widetilde{\sigma}_{\mathsf{mac}}$ structure): This game introduces a winning condition such that for all forgeries $i \in [Q_{\mathrm{S},\mathsf{info}*}+1]$, we require that $\widetilde{\sigma}^{(i)}_{\mathsf{mac}} = (\widetilde{U}^{(i)} \neq 0_{\mathbb{G}}, \widetilde{V}^{(i)}, \widetilde{S}^{(i)}_0, \widetilde{T}^{(i)}_0, \widetilde{S}^{(i)}_1, \widetilde{T}^{(i)}_1)$ satisfy the relation defined by ϕ^{mac}, i.e., if for $u^{(i)} = \mathsf{dlog}_G \widetilde{U}^{(i)}$

$$\begin{aligned}
\widetilde{V}^{(i)} &= z\widetilde{U}^{(i)} \,, \\
(u^{(i)})^{-1}\widetilde{S}^{(i)}_0 + \widetilde{S}^{(i)}_1 &= x_0 G + x_1 \widetilde{M}^{(i)} \,, \\
(u^{(i)})^{-1}\widetilde{T}^{(i)}_0 + \widetilde{T}^{(i)}_1 &= y_0 G + y_1 \widetilde{M}^{(i)} \,.
\end{aligned}$$

To bound the advantage change, we will upper bound the probability that $\mathcal{A}$ wins in game $\mathbf{G}_{\mathrm{B}.7}$ but not in this game by invoking soundness of Σ_{mac}. Hence, we make the following observations.

- With probability $1/p$, $(G, D_1, D_{2,i^*_{\mathsf{pb}}}, D_{3,i^*_{\mathsf{pb}}})$ will be a DLEQ tuple, otherwise, it is not a valid statement for Σ_{dleq}. Then, if it is not a valid statement, by special-soundness of Σ_{dleq}, there is only one $\widetilde{c}_{\mathsf{dleq}}$ that allows $\mathcal{A}$ to produce a valid response.
- Moreover, if $F \neq 0_{\mathbb{G}}$ (except with probability $1/p$) and due to perfectly strong binding of $\mathsf{C}_{\mathsf{mac}}$, ipp and $\widetilde{U}^{(i)}$ statistically fixes the underlying witness $((u^{(i)})^{-1}, x_0, x_1, y_0, y_1, z, \rho_0)$ of the statement $\mathbb{x}^{(i)}_{\mathsf{mac}} = (\mathsf{ipp}, \widetilde{U}^{(i)} \neq 0_{\mathbb{G}}, \widetilde{V}^{(i)}, \widetilde{S}^{(i)}_0, \widetilde{T}^{(i)}_0, \widetilde{S}^{(i)}_1, \widetilde{T}^{(i)}_1)$. Note that if $\widetilde{\sigma}^{(i)}_{\mathsf{mac}}$ is not in the language, no such witness would exist. Therefore, if $\widetilde{\sigma}^{(i)}_{\mathsf{mac}}$ does not satisfy the introduced checks, $\mathbb{x}^{(i)}_{\mathsf{mac}}$ is not in the language induced by the map $\phi^{\mathsf{mac}}_{\mathsf{pp},\mathsf{st}^{(i)}_{\mathsf{mac}}}$. Again, via 2-special-soundness of Σ_{mac}, there is only one $\widetilde{c}_{\mathsf{dleq}}$ that allows $\mathcal{A}$ to produce a valid response.

Therefore, except for the bad cases, there is at most $1/p$ probability of a bad challenge $\widetilde{c}_{\mathsf{or}}$ being sampled per each RO query; hence, via the union bound over the bad cases ($F = 0_{\mathbb{G}}$, $(G, D_1, D_{2,i^*_{\mathsf{pb}}}, D_{3,i^*_{\mathsf{pb}}})$ is a DLEQ tuple, and bad challenge), we have

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.7} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.8} \Rightarrow 1] + \frac{Q_{\mathsf{H}}+2}{p} \,.$$

Game $\mathbf{G}_{\mathrm{B}.9}$ (Enforce (inefficient) check on $\mathrm{cmt}_{\mathsf{rnd}}$ structure): This game introduces a winning condition such that for all forgeries $i \in [Q_{\mathrm{S},\mathsf{info}*}+1]$, the committed values of $\mathrm{cmt}^{(i)}_{\mathsf{rnd}}$ satisfies certain relations. In particular, for each i, we let

$$(\boldsymbol{\alpha}^{(i)}_{\mathsf{os}} = (\alpha^{(i)}_0, \alpha^{(i)}_1), \gamma^{(i)}_{\mathsf{os}}, r^{(i)}_x, r^{(i)}_y, r^{(i)}_M) \leftarrow \mathsf{dlog}_{\widetilde{U}^{(i)}}(\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}_{\mathsf{os}}, \overline{R}^{(i)}_x, \overline{R}^{(i)}_y, \overline{R}^{(i)}_M) \,,$$

and check the following equations

$$\begin{aligned}
\overline{A}^{(i)}_{\mathsf{os}} &= \widetilde{A}^{(i)}_{\mathsf{os}} - \phi^{\mathsf{os}}_{\mathsf{pp},W}(\boldsymbol{\alpha}^{(i)}_{\mathsf{os}}) + \gamma^{(i)}_{\mathsf{os}} \cdot W \,, \\
\widetilde{M}^{(i)} &= \overline{A}^{(i)}_{\mathsf{os}} + r^{(i)}_M F \,, \\
\widetilde{S}^{(i)}_1 &= r^{(i)}_x G + r^{(i)}_M X_1 \,, \quad \widetilde{T}^{(i)}_1 = r^{(i)}_y G + r^{(i)}_M Y_1 \,.
\end{aligned}$$

We now argue the change in advantage from the previous game. We note that due to extractability of of $\mathsf{C}_{\mathsf{rnd}}$, if $\mathrm{cmt}^{(i)}_{\mathsf{rnd}}$ does not satisfy the above check, then $\mathbb{x}^{(i)}_{\mathsf{rnd}} = (W, X_1, Y_1, \widetilde{A}^{(i)}_{\mathsf{os}}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, \widetilde{M}^{(i)}, \widetilde{U}^{(i)}, \widetilde{S}^{(i)}_1, \widetilde{T}^{(i)}_1)$ is

not a valid statement for Σ_{rnd}. Therefore, via special-soundness, there is only one bad challenge $\widetilde{c}_{\mathsf{rnd}}$ possible per each H query; then, with the union bound over all queries, we have that

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.8} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.9} \Rightarrow 1] + \frac{Q_{\mathsf{H}}}{p} \ .$$

Game $\mathbf{G}_{\mathrm{B}.10}$ (Relax the checks from $\mathbf{G}_{\mathrm{B}.8}, \mathbf{G}_{\mathrm{B}.9}$ to be efficient): For each $i \in [Q_{\mathrm{S},\mathsf{info}*} + 1]$, the game computes

$$C^{(i)}_M \leftarrow (\widetilde{z}^{(i)}_{\mathsf{os},0} + \widetilde{z}^{(i)}_{\mathsf{os},1} \cdot h - \widetilde{c}^{(i)}_{\mathsf{os}} \cdot w')\widetilde{U}^{(i)} - \overline{A}^{(i)}_0 - h \cdot \overline{A}^{(i)}_1 + w'\overline{T}^{(i)}_{\mathsf{os}} \ . \tag{9}$$

and checks

$$\widetilde{V}^{(i)} = z\widetilde{U}^{(i)} \ , \tag{10}$$

$$\widetilde{S}_0 + \overline{R}^{(i)}_x = x_0\widetilde{U}^{(i)} + x_1 C^{(i)}_M \ , \tag{11}$$

$$\widetilde{T}_0 + \overline{R}^{(i)}_y = y_0\widetilde{U}^{(i)} + y_1 C^{(i)}_M \ . \tag{12}$$

It also stops making the inefficient checks introduced in $\mathbf{G}_{\mathrm{B}.8}$ and $\mathbf{G}_{\mathrm{B}.9}$. Intuitively, we make these changes to allow the game hops (via DDH) to still efficiently check specific structure of the blinded MAC tags in the signatures while not having to compute the discrete log of $\widetilde{U}^{(i)}$. This particular step is also why we need to commit to group elements in $\mathsf{cmt}_{\mathsf{rnd}}$ with base generator $\widetilde{U}$ in our scheme instead of G.

Due to the constraints from games $\mathbf{G}_{\mathrm{B}.8}$ and $\mathbf{G}_{\mathrm{B}.9}$, this is only relaxing the winning condition, i.e., if the prior checks on $\widetilde{\sigma}^{(i)}_{\mathsf{mac}}$ and $\mathsf{cmt}^{(i)}_{\mathsf{rnd}}$ pass, the check in this game also passes. More formally, from the checks in $\mathbf{G}_{\mathrm{B}.8}$, (10) follows trivially, and we can see that

$$\begin{aligned} C^{(i)}_M &= (\widetilde{z}^{(i)}_{\mathsf{os},0} + \widetilde{z}^{(i)}_{\mathsf{os},1} \cdot h - \widetilde{c}^{(i)}_{\mathsf{os}} \cdot w')\widetilde{U}^{(i)} - \overline{A}^{(i)}_0 - h \cdot \overline{A}^{(i)}_1 + w'\overline{T}^{(i)}_{\mathsf{os}} \ , \\ &= \mathsf{dlog}_G(\widetilde{A}^{(i)}_{\mathsf{os}})\widetilde{U}^{(i)} - \mathsf{dlog}_G(\phi^{\mathsf{os}}_{\mathsf{pp},W}(\boldsymbol{\alpha}^{(i)}_{\mathsf{os}}) + \gamma^{(i)}_{\mathsf{os}} \cdot W)\widetilde{U}^{(i)} \\ &= u^{(i)}(\widetilde{A}^{(i)}_{\mathsf{os}} + \phi^{\mathsf{os}}_{\mathsf{pp},W}(\boldsymbol{\alpha}^{(i)}_{\mathsf{os}}) + \gamma^{(i)}_{\mathsf{os}} \cdot W) = u^{(i)}\overline{A}^{(i)}_{\mathsf{os}} = u^{(i)}\widetilde{M}^{(i)} - u^{(i)}r^{(i)}_M F \ . \end{aligned}$$

The second equation follows from validity of Σ_{os} transcript in $\sigma^{(i)}$, and the checks in $\mathbf{G}_{\mathrm{B}.8}$. The third follows from $u^{(i)}G = \widetilde{U}^{(i)}$, and the last follows from the check on $\widetilde{M}^{(i)}$ in $\mathbf{G}_{\mathrm{B}.8}$. Thus,

$$\begin{aligned} (u^{(i)})^{-1}\widetilde{S}^{(i)}_0 + \widetilde{S}^{(i)}_1 &= x_0 G + x_1\widetilde{M}^{(i)} \\ \widetilde{S}^{(i)}_0 + u^{(i)}\widetilde{S}^{(i)}_1 &= x_0\widetilde{U}^{(i)} + x_1 u^{(i)}\widetilde{M}^{(i)} \\ \widetilde{S}^{(i)}_0 + u^{(i)}(r^{(i)}_x G + r^{(i)}_M X_1) &= x_0\widetilde{U}^{(i)} + x_1 u^{(i)}\widetilde{M}^{(i)} \\ \widetilde{S}^{(i)}_0 + \overline{R}^{(i)}_x + x_1 r^{(i)}_M u^{(i)} F &= x_0\widetilde{U}^{(i)} + x_1 C^{(i)}_M + x_1 r^{(i)}_M u^{(i)} F \\ \widetilde{S}^{(i)}_0 + \overline{R}^{(i)}_x &= x_0\widetilde{U}^{(i)} + x_1 C^{(i)}_M \ . \end{aligned}$$

The third line follows from the checks introduced in $\mathbf{G}_{\mathrm{B}.9}$, the fourth follows from the above equation on $C^{(i)}_M$, and the fifth follows from cancelling the terms from both sides. Similar equalities apply to $\widetilde{T}^{(i)}_0$ and $\widetilde{T}^{(i)}_1$. Hence, we obtain

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.9} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.10} \Rightarrow 1] \ .$$

Game $\mathbf{G}_{\mathrm{B}.11}$ (Switch $\mathsf{H}(\mathsf{info}^*)$ to be DLEQ instance): In this game, the output of the i^*_{pb}-th query to H_{pb} is now switched to sampling $a_{i^*_{\mathsf{pb}}} \leftarrow \mathbb{Z}_p$ and set $(D_{2,i^*_{\mathsf{pb}}}, D_{3,i^*_{\mathsf{pb}}}) \leftarrow (a_{i^*_{\mathsf{pb}}}G, a_{i^*_{\mathsf{pb}}}D_1)$. Note that since the Σ_{dleq} transcript for signing on a tag info^* is now simulated, we do not need the underlying discrete log to compute the transcript. Also, the winning condition is efficiently checkable using msk. Therefore, one can construct a straightforward reduction $\mathcal{B}_{\mathsf{ddh},4}$ from the DDH problem running in time roughly $T_{\mathcal{A}}$ such that

$$|\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.10} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.11} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}_{\mathsf{ddh},4}, \lambda) \ .$$

Game $\mathbf{G}_{\mathrm{B.12}}$ (Compute Σ_{dleq} honestly and simulate Σ_{mac} for signing on all tags): In this game, we revert the changes made in $\mathbf{G}_{\mathrm{B.6}}$, such that all signing oracle queries now generate the Σ_{dleq} and Σ_{mac} transcripts by honestly computing using the corresponding a_i witness of $\mathsf{H}_{\mathsf{pb}}(\mathsf{info}_i)$ and simulating, respectively. With similar arguments from $\mathbf{G}_{\mathrm{B.5}}$ to $\mathbf{G}_{\mathrm{B.6}}$, we have that

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.11}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.12}} \Rightarrow 1] \ .$$

Game $\mathbf{G}_{\mathrm{B.13}}$ (Change how A_{os} are sampled when signing info^*): For signing session $j \in [Q_{\mathrm{S}}]$ where info is the i^*_{pb}-th H_{pb} query, we sample $r'_{\mathsf{os},j}, r_{\mathsf{os},j,1} \leftarrow\!\!\$\ \mathbb{Z}_p$ and set $A_{\mathsf{os},j} \leftarrow r'_{\mathsf{os},j}G$, while we set $r_{\mathsf{os},j,0} \leftarrow r'_{\mathsf{os},j} - r_{\mathsf{os},j,1}h$ and $\rho^*_{\mathsf{os},j} \leftarrow (r_{\mathsf{os},j,0}, r_{\mathsf{os},j,1})$. Note that the view of the adversary remains the same (since we only change the sampling order which is initially sampling $r_{\mathsf{os},j} \leftarrow\!\!\$\ \mathbb{Z}_p^2$ and setting $r'_{\mathsf{os},j} = r_{\mathsf{os},j,0} + hr_{\mathsf{os},j,1}$). Thus, we have

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.12}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.13}} \Rightarrow 1] \ .$$

Game $\mathbf{G}_{\mathrm{B.14}}$ (Sample all A_{os} in advance): This game samples randomness that will be used in the simulation in advance. In particular, it prepares two sets of randomness to be used in computing the MAC tag σ_{mac} and the Σ_{os} nonce A_{os} in signing sessions corresponding and not corresponding to i^*_{pb}-th H_{pb} query, respectively:

$$(\rho^*_{\mathrm{S},j} = (u^*_j \in \mathbb{Z}^*_p, r'_{\mathsf{os},j} \in \mathbb{Z}_p, r_{\mathsf{os},j,1} \in \mathbb{Z}_p))_{j\in[Q_{\mathrm{S}}]} \ ,$$
$$(\rho_{\mathrm{S},j} = (u_j \in \mathbb{Z}_p, S_j \in \mathbb{G}, T_j \in \mathbb{G}, \rho_{\mathsf{os},j}))_{j\in[Q_{\mathrm{S}}]} \ .$$

The game also keeps track of counters J and J^* (initially 0) which will be incremented when S_1 queries with $\mathsf{info} \neq \mathsf{info}^*$ and info^*, respectively, are made. In particular, each case the game does the following to compute σ_{mac} and A_{os} on the query info to S_1:

- info **does not correspond to i^*_{pb}-th H_{pb} query:** Increment J. Set $\sigma_{\mathsf{mac}} = (U \leftarrow u_J F, V \leftarrow u_J Z, S_J, T_J)$ and $A_{\mathsf{os}} \leftarrow \Sigma_{\mathsf{os}}.\mathsf{Init}(W, \boldsymbol{w}; \rho_{\mathsf{os},J})$.
- info **corresponds to i_{pb}-th H_{pb} query:** Increment J^*. Set $\sigma_{\mathsf{mac}} = (U \leftarrow u^*_J G, V \leftarrow zU, S \leftarrow u^*_J(x_0 + x_1 r'_{\mathsf{os},J^*})G, T \leftarrow u^*_J(y_0 + y_1 r'_{\mathsf{os},J^*})G)$ and $A_{\mathsf{os}} \leftarrow \Sigma_{\mathsf{os}}.\mathsf{Init}(W, \boldsymbol{w}; \rho^*_{\mathsf{os},J^*})$ where $\rho^*_{\mathsf{os},J} = (r'_{\mathsf{os},J^*} - r_{\mathsf{os},J^*,1}h, r_{\mathsf{os},J,1})$.

Then, $\rho_{\mathrm{S},J}$ and ρ^*_{S,J^*} will be used to compute σ_{mac} and A_{os} according to the cases. Finally, this game introduces an abort at the end of the game if $\overline{A}^{(i^*)}_{\mathsf{os}} \in \{r'_{\mathsf{os},j}G\}_{j\in[Q_{\mathrm{S}}]}$.

The purpose generating the nonces and tags beforehand is to prepare for the rewinding reduction. In particular, we note the first run of $\mathcal{A}$ will (statistically) fix a MAC forgery for a fresh nonce $\overline{A}^{(i^*)}_{\mathsf{os}}$. Then, consider the situation when the reduction makes extra MAC tag queries in the second run of $\mathcal{A}$, it is possible that a tag was given for this nonce; and thus, invalidating the freshness of the MAC forgery. The abort is introduced to avoid a situation where $\mathcal{A}$ does not make all of its Q_{S} signing queries on info^*. Indeed, the prepared nonces and tags will not be given to $\mathcal{A}$, but we still want to ensure that we can find a fresh MAC tag and nonce from $\mathcal{A}$'s forgeries—which should not correspond to any of the prepared but unused tags and nonces.

Despite the change, the view of $\mathcal{A}$ remains the same, because we only move the sampling of the values to the start. The only change in advantage is due to the introduced abort. Since by definition of i^* [7], $\overline{A}^{(i^*)}_{\mathsf{os}} \neq r'_{\mathsf{os},j}G$ for $1 \leqslant j < \ell$, we only have to consider when $\overline{A}^{(i^*)}_{\mathsf{os}} = r'_{\mathsf{os},j}G$ for $\ell + 1 \leqslant j \leqslant Q_{\mathrm{S}}$. However, $r'_{\mathsf{os},j}$ for $\ell \leqslant j \leqslant Q_{\mathrm{S}}$ are uniformly random and hidden from the adversary, since these are prepared random values that were not given to $\mathcal{A}$ as it only makes $\ell - 1$ queries on info^*. Therefore, the probability of such abort occurring is at most Q_{S}/p via the union bound over $\ell \leqslant j \leqslant Q_{\mathrm{S}}$. Therefore,

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.13}} \Rightarrow 1] \leqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.14}} \Rightarrow 1] + \frac{Q_{\mathrm{S}}}{p} \ .$$

[7] Recall that the index $i^* \in [\ell]$ corresponds to a forgery where the extracted $\overline{A}^{(i^*)}_{\mathsf{os}}$ is one of the nonces not used in the signing sessions on info^*. This exists due to the extracted nonces being distinct and the pigeonhole principle.

Alg. $\mathcal{A}_{\mathsf{NonceDist}}(\mathsf{inp}, \boldsymbol{h} \in \mathbb{Z}_p^{Q_\mathsf{H}}; \rho_{\mathcal{A}})$:

1 : **parse** $\mathsf{inp} = (\mathsf{pp}, \mathsf{ipp}, \mathsf{td}, \boldsymbol{w}, h, (\rho^*_{\mathrm{S},j}, \rho_{\mathrm{S},j}, \rho_{\Sigma,j})_{j\in[Q_\mathrm{S}]}, (a_i)_{i\in[Q_{\mathsf{pb}}]}, (\widetilde{c}_i)_{i\in[Q_\mathsf{H}]}, i^*_{\mathsf{pb}})$

2 : **parse** $\mathsf{pp} = (H, \mathsf{mpp} = (F, \mathsf{mck}), \mathsf{bck}, D_1)$, $\mathsf{ipp} = (\mathrm{cmt}_0, X_1, Y_1, Z)$

3 : **for** $j \in [Q_\mathrm{S}]$:

4 : **parse** $\rho^*_{\mathrm{S},j} = (\sigma_{\mathsf{mac},j}, r'_{\mathsf{os},j}, r_{\mathsf{os},j,0})$, $\rho_{\mathrm{S},j} = (u_j, S_j, T_j, \rho_{\mathsf{os},j})$

5 : **parse** $\rho_{\Sigma,j} = (\rho_{\mathsf{mac},j}, \rho_{\mathsf{dleq},j}, c_{\mathsf{mac},j})$

6 : **for** $i \in [Q_\mathsf{H}]$:

7 : **parse** $\widetilde{c}_i = (\widetilde{c}_{\mathsf{os},i}, \widetilde{c}_{\mathsf{or},i})$

8 : $\widetilde{c}'_i \leftarrow (\widetilde{c}_{\mathsf{os},i}, \widetilde{c}_{\mathsf{or},i}, h_i)$

9 : **for** $i \in [Q_{\mathsf{pb}}] : (D_{2,i}, D_{3,i}) \leftarrow a_i(G, D_1)$

10 : $w' \leftarrow w_0 + hw_1$; $W \leftarrow w'G$

11 : $i_{\mathsf{pb}}, i_\mathsf{H}, J, J^*, \mathsf{cnt}_\mathrm{S} \leftarrow 0$ // Initialize counters

12 : $(\mathsf{info}^*, (\mu^{(i)}, \sigma^{(i)})_{i\in[J^*+1]}) \leftarrow\!\$\ \mathcal{A}^{\mathrm{S}_1,\mathrm{S}_2,\mathsf{H},\mathsf{H}_{\mathsf{pb}}}(\mathsf{pp}, \mathsf{vk} = (\mathsf{ipp}, W); \rho_{\mathcal{A}})$ // Run $\mathcal{A}$

13 : **for** $i \in [J^* + 1]$:

14 : **parse** $\sigma^{(i)} = (\widetilde{M}^{(i)}, \widetilde{\sigma}^{(i)}_{\mathsf{mac}}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, (\widetilde{A}^{(i)}_{\mathrm{xxx}}, \widetilde{c}^{(i)}_{\mathrm{xxx}}, \widetilde{z}^{(i)}_{\mathrm{xxx}})_{\mathrm{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$

15 : $\mathsf{HQry}^{(i)} \leftarrow (\widetilde{M}^{(i)}, \widetilde{\sigma}^{(i)}_{\mathsf{mac}}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}}, \widetilde{A}^{(i)}_{\mathsf{os}}, \widetilde{A}^{(i)}_{\mathsf{mac}}, \widetilde{A}^{(i)}_{\mathsf{dleq}}, \widetilde{A}^{(i)}_{\mathsf{rnd}}, \mu^{(i)}, \mathsf{info}^*)$

16 : $\mathsf{mes}^{(i)}_{\mathsf{rnd}} = (\overline{A}^{(i)}_0, \overline{A}^{(i)}_1, \overline{\Gamma}^{(i)}_{\mathsf{os}}, \overline{R}^{(i)}_x, \overline{R}^{(i)}_y, \overline{R}^{(i)}_M, \overline{A}^{(i)}_{\mathsf{os}}) \leftarrow \mathsf{C}_{\mathsf{rnd}}.\mathsf{Extract}(\mathsf{td}, \mathrm{cmt}^{(i)}_{\mathsf{rnd}})$

17 : **if** $\mathsf{info}^* \neq \mathsf{info}[i^*_{\mathsf{pb}}] \ \vee\ \exists i \in [J^* + 1] : \mathsf{BS}_{\mathrm{alg}}.\mathsf{Ver}(\mathsf{vk}, \mathsf{info}^*, \mu^{(i)}, \sigma^{(i)}) = 0$ // Abort if $\neg\mathsf{NonceDist}$

18 : $\quad \vee\ \exists i_0 \neq i_1 \in [J^* + 1] : \mathsf{HQry}^{(i_0)} = \mathsf{HQry}^{(i_1)} \ \vee\ \overline{A}^{(i_0)}_{\mathsf{os}} = \overline{A}^{(i_1)}_{\mathsf{os}}$ **then return** $\perp$

19 : $i^* \leftarrow \min\{i : \overline{A}^{(i)}_{\mathsf{os}} \notin \{r'_{\mathsf{os},j}G\}_{j\in[Q_\mathrm{S}]}\}$

20 : **if** $i^* = \perp$ **then return** $\perp$

21 : $I \leftarrow \arg_i\{\mathrm{str}[i] = \mathsf{HQry}^{(i^*)}\}$

22 : **return** $(I, \sigma^{(i^*)})$

Fig. 15. Wrapper algorithm $\mathcal{A}_{\mathsf{NonceDist}}$ for the proof of Lemma 6.3. The oracles simulation are given in Figure 16. We denote internal states of $\mathcal{A}_{\mathsf{NonceDist}}$ $\mathsf{info}[\cdot], \mathrm{str}[\cdot]$ for mappings from indices $i \in [Q_{\mathsf{pb}}]$ and $i \in [Q_\mathsf{H}]$ to the i-th query to H_{pb} and H, respectively; additionally, $\mathsf{H}_{\mathsf{pb}}[\cdot], \mathsf{H}[\cdot]$ are mappings from query to output of the corresponding RO. We assume that the tuple $(\mathbb{G}, p, G)$ is implicit in pp.

To bound the advantage of $\mathcal{A}$ in $\mathbf{G}_{\mathrm{B}.14}$, we will rewind the adversary to extract (via special-soundness of Σ_{rnd}) the underlying randomization on $\widetilde{M}^{(i^*)}$ and $\widetilde{\sigma}^{(i^*)}_{\mathsf{mac}}$ and then construct a reduction to UF-RMA of $\mathsf{M}_{\mathsf{alg}}$. We establish this in the following lemma. □

Lemma 6.10 (Forking for NonceDist). *There exists an adversary $\mathcal{B}_{\mathrm{alg}}$ in game* UF-RMA *running in time $T_{\mathrm{alg}} \approx 2T_{\mathcal{A}}$ making Q_S queries to the* Tag *such that*

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.14} \Rightarrow 1] \leqslant \sqrt{Q_\mathrm{H} \cdot \mathsf{Adv}^{\mathsf{ufrma}}_{\mathsf{M}_{\mathsf{alg}}}(\mathcal{B}_{\mathrm{alg}}, \lambda)} + \frac{Q_\mathrm{H}}{p} .$$

Proof (of Lemma 6.10). The proof follows the following steps: (1) Define a wrapper algorithm $\mathcal{A}_{\mathsf{NonceDist}}$, (2) Construct a reduction $\mathcal{B}_{\mathrm{alg}}$ against UF-RMA game of $\mathsf{M}_{\mathsf{alg}}$ that rewinds $\mathcal{A}_{\mathsf{NonceDist}}$, (3) Analyze $\mathcal{B}_{\mathrm{alg}}$ via the forking lemma.

WRAPPER $\mathcal{A}_{\mathsf{NonceDist}}$. The wrapper $\mathcal{A}_{\mathsf{NonceDist}}$ (defined in Figures 15 and 16) simulates the game $\mathbf{G}_{\mathrm{B}.14}$ to $\mathcal{A}$. In particular, it takes as input the public parameters pp as generated in $\mathbf{G}_{\mathrm{A}.14}$, ipp generated via $\mathsf{M}_{\mathsf{alg}}.\mathsf{KeyGen}$ (but without access to msk), extraction trapdoor td, witness $\boldsymbol{w} \in \mathbb{Z}_p^2$ for Σ_{os}, discrete logarithm $h \in \mathbb{Z}_p$, and prepared randomness $((\rho^*_{\mathrm{S},j}, \rho_{\mathrm{S},j}, \rho_{\Sigma,j})_{j\in[Q_\mathrm{S}]}, (a_i)_{i\in[Q_{\mathsf{pb}}]}, (\widetilde{c}_i)_{i\in[Q_\mathsf{H}]}, i^*_{\mathsf{pb}})$ to be used for oracle simulation along with challenges $\boldsymbol{h} \in \mathbb{Z}_p^{Q_\mathsf{H}}$. It then runs $\mathcal{A}$ and simulates the game as follows:

Input to $\mathcal{A}$: The wrapper forwards pp and $\mathsf{vk} = (\mathsf{ipp}, W = w_0G + w_1H)$ to $\mathcal{A}$.

Random oracles $\mathsf{H}, \mathsf{H}_{\mathsf{pb}}$: $(\widetilde{c}_i \in \mathbb{Z}_p^2)_{i\in[Q_\mathsf{H}]}$ and $\boldsymbol{h}$ are used to simulate H where the i-th query is replied with $(\widetilde{c}_i, h_i) \in \mathbb{Z}_p^3$, while a_i is used to simulate H_{pb} where the i-th query is replied with $a_i(G, D_1)$.

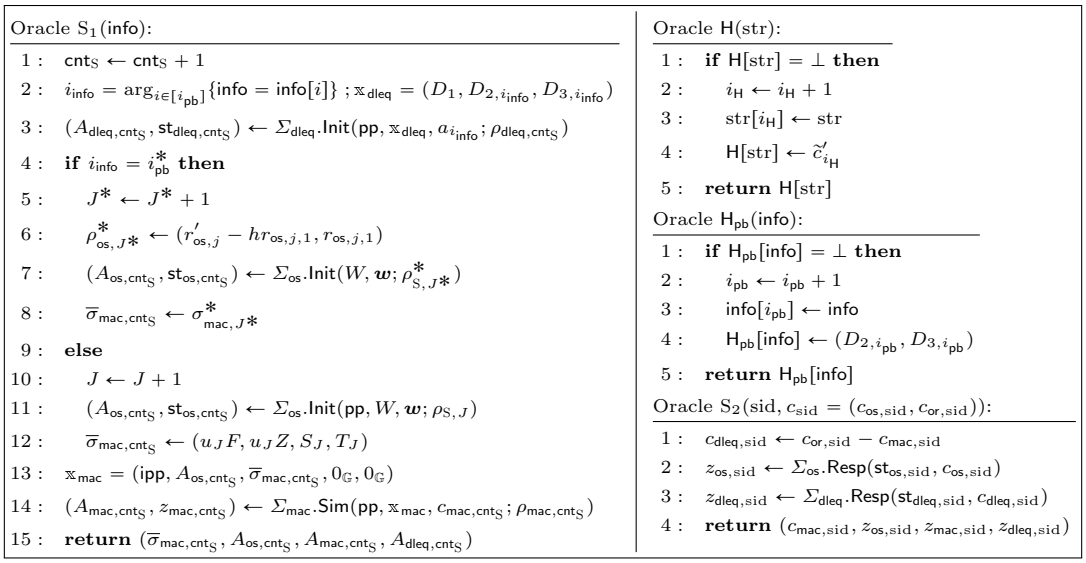

Fig. 16. Oracle simulation by $\mathcal{A}_{\mathsf{NonceDist}}$ extended from Figure 15. We simplify the description of S_2 be removing the sid tracking and assume that the second time an sid is queried to S_2, $\mathcal{A}_{\mathsf{pr\text{-}cr}}$ simply return $\perp$.

Signing oracles $\mathbf{S}_1, \mathbf{S}_2$: Throughout the simulation, i^*_{pb} corresponds to the guessed index of info^*. Counters J, J^* are incremented as in $\mathbf{G}_{\mathrm{A.14}}$.

- $\rho^*_{\mathrm{S},j} = (\sigma_{\mathsf{mac},j}, r'_{\mathsf{os},j}, r_{\mathsf{os},j,0})$ is used to compute A_{os} in j-th signing session where info^* is queried. However, we note that $\rho^*_{\mathrm{S},j}$ already contains precomputed $\mathsf{M}_{\mathsf{alg}}$ tag $\sigma_{\mathsf{mac},j}$, which will come from the Tag oracle in our reduction.
- $\rho_{\mathrm{S},j} = (u_j, S_j, T_j, \rho_{\mathsf{os},j})$ is used to compute the tag $(u_J F, u_J Z, S_J, T_J)$ and A_{os} for the j-th signing session where $\mathsf{info} \neq \mathsf{info}^*$ as in $\mathbf{G}_{\mathrm{B.14}}$.
- $\rho_{\Sigma,j}$ will be used in the j-th signing session (including both signing info^* and $\mathsf{info} \neq \mathsf{info}^*$) to compute Σ_{mac} and Σ_{dleq} transcripts.

Output of $\mathcal{A}_{\mathsf{NonceDist}}$: The wrapper checks that the event $\mathsf{NonceDist}$ actually occurs and that there is an index i^* where $\overline{A}^{(i^*)}_{\mathsf{os}}$ does not correspond to the prepared values $r'_{\mathsf{os},j}G$ as checked in $\mathbf{G}_{\mathrm{B.14}}$. Then, it outputs the corresponding index $I \in [Q_{\mathsf{H}}]$ to the H query in the verification of the i^*-th forgery along with $\sigma^{(i^*)}$ If $\mathsf{NonceDist}$ does not occur, it returns $\perp$.

We note that the wrapper does not check the additional winning condition introduced in $\mathbf{G}_{\mathrm{B.10}}$. This is due to the reduction not having access to the MAC secret key msk. However, if $\mathcal{A}$ wins in game $\mathbf{G}_{\mathrm{B.14}}$, then the output $(I, \sigma) \leftarrow \mathcal{A}_{\mathsf{NonceDist}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})$ also satisfy the checks in Equations (10) to (12). In particular, if we let

$$\sigma = (\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}} = (\widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{S}_1, \widetilde{T}_0, \widetilde{T}_1), \mathrm{cmt}_{\mathsf{rnd}}, (\widetilde{A}_{\mathsf{xxx}}, \widetilde{c}_{\mathsf{xxx}}, \widetilde{z}_{\mathsf{xxx}})_{\mathsf{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}}) \ ,$$
$$(\overline{A}_0, \overline{A}_1, \overline{\Gamma}_{\mathsf{os}}, \overline{R}_x, \overline{R}_y, \overline{R}_M, \overline{A}_{\mathsf{os}}) \leftarrow \mathsf{C}_{\mathsf{rnd}}.\mathsf{Extract}(\mathsf{td}, \mathrm{cmt}_{\mathsf{rnd}}) \ ,$$

we have that

$$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B.14}} \Rightarrow 1] \leqslant \mathsf{Pr}_{\mathsf{inp},\boldsymbol{h},\rho_{\mathcal{A}}}\left[\begin{array}{l|l} I \neq \perp \ \wedge \ \widetilde{V} = z\widetilde{U} \ \wedge & \\ \widetilde{S}_0 + \overline{R}_x = x_0\widetilde{U} + x_1 C_M \ \wedge & (I, \sigma) \leftarrow \mathcal{A}_{\mathsf{NonceDist}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}}) \\ \widetilde{T}_0 + \overline{R}_y = y_0\widetilde{U} + y_1 C_M & \end{array}\right]$$

where

$$C_M \leftarrow (\widetilde{z}_{\mathsf{os},0} + \widetilde{z}_{\mathsf{os},1} \cdot h - \widetilde{c}_{\mathsf{os}} \cdot w')\widetilde{U} - \overline{A}_0 - h \cdot \overline{A}_1 + w'\overline{\Gamma}_{\mathsf{os}} \ , \tag{13}$$

as in Equation (9). Due to perfect strong binding of $\mathsf{C}_{\mathsf{mac}}$, the underlying secret keys x_0, y_0, x_1, y_1, z are fixed in the ipp. Denote such event on the output σ as Good. For our reduction's analysis, we define an additional

Alg. $\mathcal{B}_{\text{alg}}^{\text{Tag}}(\mathsf{mpp}, \mathsf{ipp})$:

$h \leftarrow\!\!\$\ \mathbb{Z}_p\ ;\ H \leftarrow hG\ ;\ (\mathsf{bck}, \mathsf{td}) \leftarrow\!\!\$\ \mathsf{C}_{\mathsf{rnd}}.\mathsf{ExtSetup}(\mathbb{G}, p, G)\ ;\ \boldsymbol{w} \leftarrow\!\!\$\ \mathbb{Z}_p^2$

$\mathsf{pp} \leftarrow (H, \mathsf{mpp} = (F, \mathsf{mck}), \mathsf{bck})$

for $j \in [Q_{\text{S}}]$:

  $(r'_{\mathsf{os},j}, \sigma_{\mathsf{mac},j}) \leftarrow \text{Tag}()$ // Set up MAC tags

  $u_j \leftarrow\!\!\$\ \mathbb{Z}_p^*\ ;\ S_j, T_j \leftarrow\!\!\$\ \mathbb{G}$

  $r_{\mathsf{os},j,1} \leftarrow\!\!\$\ \mathbb{Z}_p\ ;\ \rho_{\mathsf{os},j} \leftarrow\!\!\$\ \mathbb{Z}_p^2$

  $\rho_{\mathsf{mac},j} \leftarrow\!\!\$\ \mathbb{Z}_p^6 \times \mathsf{C}_{\mathsf{mac}}.\mathcal{R}\ ;\ \rho_{\mathsf{dleq},j}, c_{\mathsf{mac},j} \leftarrow\!\!\$\ \mathbb{Z}_p$

  $\rho^*_{\text{S},j} \leftarrow (\sigma_{\mathsf{mac},j}, r'_{\mathsf{os},j}, r_{\mathsf{os},j,1})\ ;\ \rho_{\text{S},j} \leftarrow (u_j, S_j, T_j, \rho_{\mathsf{os},j})$

  $\rho_{\Sigma,j} \leftarrow (\rho_{\mathsf{mac},j}, \rho_{\mathsf{dleq},j}, c_{\mathsf{mac},j})$

$a_1, \ldots, a_{Q_{\mathsf{pb}}} \leftarrow\!\!\$\ \mathbb{Z}_p\ ;\ \widetilde{c}_1, \ldots, \widetilde{c}_{Q_{\mathsf{H}}} \leftarrow\!\!\$\ \mathbb{Z}_p^2\ ;\ i^*_{\mathsf{pb}} \leftarrow\!\!\$\ [Q_{\mathsf{pb}}]$

$\mathsf{inp} \leftarrow (\mathsf{pp}, \mathsf{ipp}, \mathsf{td}, \boldsymbol{w}, h, (\rho^*_{\text{S},j}, \rho_{\text{S},j}, \rho_{\Sigma,j})_{j\in[Q_{\text{S}}]}, (a_i)_{i\in[Q_{\mathsf{pb}}]}, (\widetilde{c}_i)_{i\in[Q_{\mathsf{H}}]}, i^*_{\mathsf{pb}})$

$(b, \sigma, \sigma') \leftarrow\!\!\$\ \mathsf{Fork}^{\mathcal{A}_{\mathsf{NonceDist}}}(\mathsf{inp})$

if $b = 0$ **then return** $\perp$

parse $\sigma = (\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \text{cmt}_{\mathsf{rnd}}, (\widetilde{A}_{\text{xxx}}, \widetilde{c}_{\text{xxx}}, \widetilde{z}_{\text{xxx}})_{\text{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$

parse $\sigma' = (\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \text{cmt}_{\mathsf{rnd}}, (\widetilde{A}_{\text{xxx}}, \widetilde{c}'_{\text{xxx}}, \widetilde{z}'_{\text{xxx}})_{\text{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})$

parse $\widetilde{\sigma}_{\mathsf{mac}} = (\widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{T}_0, \widetilde{S}_1, \widetilde{T}_1)$

$\mathbb{x}_{\mathsf{rnd}} = (W = w_0 G + w_1 H, X_1, Y_1, \widetilde{A}_{\mathsf{os}}, \text{cmt}_{\mathsf{rnd}}, \widetilde{M}, \widetilde{U}, \widetilde{S}_1, \widetilde{T}_1)$

$(\alpha_0, \alpha_1, \gamma_{\mathsf{os}}, r_x, r_y, r_M, \rho_{\mathsf{rnd}}) \leftarrow \Sigma_{\mathsf{rnd}}.\mathsf{Ext}(\mathbb{x}_{\mathsf{rnd}}, \widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}}, \widetilde{c}'_{\mathsf{rnd}}, \widetilde{z}'_{\mathsf{rnd}})$

$w' \leftarrow w_0 + w_1 h$

$m^* \leftarrow \widetilde{z}_{\mathsf{os},0} + \widetilde{z}_{\mathsf{os},1} h - \widetilde{c}_{\mathsf{os}} w' - \alpha_0 - \alpha_1 h + \gamma_{\mathsf{os}} w'$

$\sigma^*_{\text{alg}} \leftarrow (\widetilde{U}, \widetilde{V}, \widetilde{S}_0 + r_x \widetilde{U}, \widetilde{T}_0 + r_y \widetilde{U})$

return $(m^*, \sigma^*_{\text{alg}})$

Fig. 17. Reduction $\mathcal{B}_{\text{alg}}$. We assume that the tuple $(\mathbb{G}, p, G)$ is implicit in mpp.

inefficient wrapper $\mathcal{A}'_{\mathsf{NonceDist}}$ which runs $\mathcal{A}_{\mathsf{NonceDist}}$ but also checks for Good on the output, and return $\perp$ if it does not occur; hence,

$$\begin{aligned}&\mathsf{Pr}_{\mathsf{inp},\boldsymbol{h},\rho_{\mathcal{A}}}[\perp \neq (I,\sigma) \leftarrow \mathcal{A}'_{\mathsf{NonceDist}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})]\\ &:= \mathsf{Pr}_{\mathsf{inp},\boldsymbol{h},\rho_{\mathcal{A}}}[\perp \neq (I,\sigma) \leftarrow \mathcal{A}_{\mathsf{NonceDist}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}}) \ \wedge\ \mathsf{Good}] \geqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\text{B}.14} \Rightarrow 1]\ .\end{aligned}$$

REDUCTION $\mathcal{B}_{\text{alg}}$. Our reduction $\mathcal{B}_{\text{alg}}$ (given in Figure 17) takes as input mpp and ipp from the UF-RMA game and runs $\mathcal{A}_{\mathsf{NonceDist}}$ twice (via the forking algorithm $\mathsf{Fork}^{\mathcal{A}_{\mathsf{NonceDist}}}$), preparing its input inp as follows:

- Sample $H \in \mathbb{G}$ while knowing $\mathsf{dlog}_G H$, bck along with an extraction trapdoor td from $\mathsf{C}_{\mathsf{rnd}}.\mathsf{ExtSetup}$, the Okamoto-Schnorr witness $\boldsymbol{w} \leftarrow\!\!\$\ \mathbb{Z}_p^2$ and setup $W \leftarrow w_0 G + w_1 H$, and $i^*_{\mathsf{pb}} \leftarrow\!\!\$\ [Q_{\mathsf{pb}}]$.
- **Randomness for signing info^*:** Call the Tag oracle to setup the Q_{S} randomness $r'_{\mathsf{os},j}$ and the corresponding $\mathsf{M}_{\mathsf{alg}}$ tag $\sigma_{\mathsf{mac},j}$. Also, sample $r_{\mathsf{os},j,1} \leftarrow\!\!\$\ \mathbb{Z}_p$.
- **Randomness for signing $\mathsf{info} \neq \mathsf{info}^*$:** Sample $u_j \leftarrow\!\!\$\ \mathbb{Z}_p^*, S_j, T_j \leftarrow\!\!\$\ \mathbb{G}$ and $\rho_{\mathsf{os},j} \leftarrow\!\!\$\ \mathbb{Z}_p^2$.
- **Randomness for $\Sigma_{\mathsf{mac}}, \Sigma_{\mathsf{dleq}}$:** Sample $c_{\mathsf{mac},j}, \rho_{\mathsf{mac},j}$ and $\rho_{\mathsf{dleq},j}$ to be used for computing the $\Sigma_{\mathsf{mac}}, \Sigma_{\mathsf{dleq}}$ transcripts in signing.
- **RO outputs:** Sample $a_1, \ldots, a_{Q_{\mathsf{pb}}} \leftarrow\!\!\$\ \mathbb{Z}_p, \widetilde{c}_1, \ldots, \widetilde{c}_{Q_{\mathsf{H}}} \leftarrow\!\!\$\ \mathbb{Z}_p^2$ and $\boldsymbol{h} \leftarrow\!\!\$\ \mathbb{Z}_p^{Q_{\mathsf{H}}}$.

It runs $\mathsf{Fork}^{\mathcal{A}_{\mathsf{NonceDist}}}(\mathsf{inp})$, where inp is sampled as above, and aborts if the forking algorithm is unsuccessful. Otherwise, it parses the outputs of Fork as

$$\begin{aligned}\sigma &= (\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \text{cmt}_{\mathsf{rnd}}, (\widetilde{A}_{\text{xxx}}, \widetilde{c}_{\text{xxx}}, \widetilde{z}_{\text{xxx}})_{\text{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})\ ,\\ \sigma' &= (\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \text{cmt}_{\mathsf{rnd}}, (\widetilde{A}_{\text{xxx}}, \widetilde{c}'_{\text{xxx}}, \widetilde{z}'_{\text{xxx}})_{\text{xxx}\in\{\mathsf{os},\mathsf{mac},\mathsf{dleq},\mathsf{rnd}\}})\ ,\\ \widetilde{\sigma}_{\mathsf{mac}} &= (\widetilde{U}, \widetilde{V}, \widetilde{S}_0, \widetilde{S}_1, \widetilde{T}_0, \widetilde{T}_1)\end{aligned}$$

Note that $\widetilde{M}, \widetilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{A}_{\mathsf{xxx}}$ are shared since they are contained in the same query to H within the two runs due to the success condition of Fork. Also, with $h_I \neq h'_I$, $\widetilde{c}_{\mathsf{rnd}} \neq \widetilde{c}'_{\mathsf{rnd}}$. Then, it uses the extractor of Σ_{rnd} to extract the underlying witness denoted $(\alpha_0, \alpha_1, \gamma_{\mathsf{os}}, r_x, r_y, r_M)$ from the two valid transcripts $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ and $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}'_{\mathsf{rnd}}, \widetilde{z}'_{\mathsf{rnd}})$. Finally, the reduction returns the following values as $\mathsf{M}_{\mathsf{alg}}$ forgery:

$$\begin{aligned} m^* &= \widetilde{z}_{\mathsf{os},0} + \widetilde{z}_{\mathsf{os},1} h - \widetilde{c}_{\mathsf{os}} w' - \alpha_0 - \alpha_1 h + \gamma_{\mathsf{os}} w' \ , \\ \sigma^*_{\mathrm{alg}} &= (\widetilde{U}, \widetilde{V}, \widetilde{S}_0 + r_x \widetilde{U}, \widetilde{T}_0 + r_y \widetilde{U}) \ . \end{aligned}$$

Analysis of $\mathcal{B}_{\mathrm{alg}}$. First, recall that by the definition of $\mathcal{A}_{\mathsf{NonceDist}}$, the underlying $\overline{A}_{\mathsf{os}}$ extracted from $\mathrm{cmt}_{\mathsf{rnd}}$ does not correspond to any $\{r'_{\mathsf{os},j} G\}_{j \in [Q_{\mathrm{S}}]}$. Hence, by definition of the mapping $\phi^{\mathsf{rnd}}_{\mathsf{pp}, \mathbb{x}_{\mathsf{rnd}}}$ and the extracted witnesses, we have

$$\begin{aligned} \overline{A}_{\mathsf{os}} &= \widetilde{A}_{\mathsf{os}} - \alpha_0 G - \alpha_1 H + \gamma_{\mathsf{os}} W \\ &= \phi^{\mathsf{os}}_{\mathsf{pp}, \mathbb{x}_{\mathsf{os}}}(\widetilde{z}_{\mathsf{os}}) - \widetilde{c}_{\mathsf{os}} W - \alpha_0 G - \alpha_1 H + \gamma_{\mathsf{os}} W \\ &= \widetilde{z}_{\mathsf{os},0} G + \widetilde{z}_{\mathsf{os},1} H - \widetilde{c}_{\mathsf{os}} W - \alpha_0 G - \alpha_1 H + \gamma_{\mathsf{os}} W \\ &= (\widetilde{z}_{\mathsf{os},0} + \widetilde{z}_{\mathsf{os},1} h - \widetilde{c}_{\mathsf{os}} w' - \alpha_0 - \alpha_1 h + \gamma_{\mathsf{os}} w') G = m^* G \ . \end{aligned}$$

The second equality follows from the verification equation of Σ_{os}. Thus, m^* is a fresh message not output from Tag before.

Therefore, we only need to identify an event where $\mathcal{B}_{\mathrm{alg}}$ outputs a valid σ^*_{alg} for m^*. Recall the checks in the event Good, i.e., we have

$$\widetilde{V} = z \widetilde{U} \ , \qquad \widetilde{S}_0 + \overline{R}_x = x_0 \widetilde{U} + x_1 C_M \ , \qquad \widetilde{T}_0 + \overline{R}_y = y_0 \widetilde{U} + y_1 C_M \ .$$

Here, by special soundness of Σ_{rnd}, the extracted witness satisfies $(\alpha_0, \alpha_1, \gamma_{\mathsf{os}}, r_x, r_y, r_M)\widetilde{U} = (\overline{A}_0, \overline{A}_1, \overline{\Gamma}_{\mathsf{os}}, \overline{R}_x, \overline{R}_y, \overline{R}_M)$. Thus, via Equation (13), we have that

$$\begin{aligned} C_M &= (\widetilde{z}_{\mathsf{os},0} + \widetilde{z}_{\mathsf{os},1} \cdot h - \widetilde{c}_{\mathsf{os}} \cdot w') \widetilde{U} - \overline{A}_0 - h \cdot \overline{A}_1 + w' \overline{\Gamma}_{\mathsf{os}} \\ &= (\widetilde{z}_{\mathsf{os},0} + \widetilde{z}_{\mathsf{os},1} \cdot h - \widetilde{c}_{\mathsf{os}} \cdot w' - \alpha_0 - \alpha_1 h + \gamma_{\mathsf{os}} w') \widetilde{U} = m^* \widetilde{U} \ . \end{aligned}$$

Therefore, the reduction's output satisfies

$$\begin{aligned} \widetilde{S}_0 + \overline{R}_x &= x_0 \widetilde{U} + x_1 C_M \\ \widetilde{S}_0 + r_x \widetilde{U} &= (x_0 + x_1 m^*) \widetilde{U} \ . \end{aligned}$$

Similarly, $\widetilde{T}_0 + r_y \widetilde{U} = (y_0 + y_1 m^*) \widetilde{U}$, meaning that σ^*_{alg} is valid for m^* with respect to the underlying msk. Thus, if in both runs of $\mathcal{A}_{\mathsf{NonceDist}}$ within Fork, the output indices $I = I'$ and the event Good occurs (i.e., the additional wrapper $\mathcal{A}'_{\mathsf{NonceDist}}$ outputs $I = I' \neq \bot$), $\mathcal{B}_{\mathrm{alg}}$ wins in the UF-RMA game. Therefore,

$$\mathsf{Adv}^{\mathsf{ufrma}}_{\mathsf{M}_{\mathsf{alg}}}(\mathcal{B}_{\mathrm{alg}}, \lambda) \geqslant \mathsf{Pr}_{\mathsf{inp}}[\mathsf{Fork}^{\mathcal{A}'_{\mathsf{NonceDist}}}(\mathsf{inp}) \neq \bot] \geqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.14} \Rightarrow 1] \left(\frac{\mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.14} \Rightarrow 1]}{Q_{\mathsf{H}}} - \frac{1}{p} \right) \ ,$$

where the last inequality follows from Lemma 2.18 and that $\mathsf{Pr}_{\mathsf{inp}, \boldsymbol{h}, \rho_{\mathcal{A}}}[\bot \neq (I, \sigma) \leftarrow \mathcal{A}'_{\mathsf{NonceDist}}(\mathsf{inp}, \boldsymbol{h}; \rho_{\mathcal{A}})] \geqslant \mathsf{Pr}[\mathbf{G}^{\mathcal{A}}_{\mathrm{B}.14} \Rightarrow 1]$. Note that $\mathcal{B}_{\mathrm{alg}}$ runs in time roughly $2T_{\mathcal{A}}$, concluding the proof. □

7 Blindness Proof of $\mathsf{BS}_{\mathsf{alg}}$

In this section, we show Theorem 3.6. Recall that Q_{H} (resp. Q_{pb}) denotes the number of hash queries to H (resp. H_{pb}), including the queries induced by the blindness game. Let $\mathcal{A}$ be an adversary against the blindness game of $\mathsf{BS}_{\mathrm{alg}}$, running in time $T_{\mathcal{A}}$. Let us consider the following sequence of games, where we denote

$\mathsf{Pr}[\mathbf{G}^{\mathcal{A}} \Rightarrow 1]$ (omitting the parameter λ) as the probability of the game $\mathbf{G}$ outputting 1 after interacting with adversary $\mathcal{A}$.

Game $\mathbf{G}_0$ (Original blindness game): This game is the original blindness game, i.e., the game runs $\mathsf{pp} \leftarrow\!\!\$\ \mathsf{BS}_{\mathrm{alg}}.\mathsf{Setup}(1^\lambda)$, samples $b \leftarrow\!\!\$\ \{0,1\}$ and interacts with $\mathcal{A}$ on input pp simulating the oracles $\mathsf{Init}, \mathrm{U}_1$ and U_2 honestly. When $\mathcal{A}$ outputs b', the game outputs 1 iff $b' = b$. By definition, we have

$$\mathsf{Adv}^{\mathsf{blind}}_{\mathsf{BS}_{\mathrm{alg}}}(\mathcal{A}, \lambda) = \left| \mathsf{Pr}[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2} \right|.$$

In the following, we assume that $U \neq 0_{\mathbb{G}}$ and that the Σ-protocol transcripts issued by the adversary verify. This is without loss of generality, as these checks are performed by $\mathsf{BS}_{\mathrm{alg}}.\mathsf{U}_2$; if a check fails, then the game outputs $(\bot, \bot)$ for both signing sessions.

Game $\mathbf{G}_1$ (Abort if H was queried before $\mathcal{A}$ recevied (σ_0, σ_1)): When the game queries $\mathsf{in}_{\mathsf{H}} := (\widetilde{M}, \tilde{\sigma}_{\mathsf{mac}}, \mathrm{cmt}_{\mathsf{rnd}}, \widetilde{A}_{\mathsf{os}}, \widetilde{A}_{\mathsf{mac}}, \widetilde{A}_{\mathsf{dleq}}, \widetilde{A}_{\mathsf{rnd}}, \mu, \mathsf{info})$ to H in oracle U_1, the game aborts if in_{H} was already queried to H by $\mathcal{A}$. Further, the game also aborts in oracle H if in_{H} is later queried by $\mathcal{A}$ before it obtains (σ_0, σ_1).

Observe that $\widetilde{U} = \beta U \neq 0_{\mathbb{G}}$ and that the values $r_M, r_x, r_y \leftarrow\!\!\$\ \mathbb{Z}_p$ are information-theoretically hidden from $\mathcal{A}$ before obtaining (σ_0, σ_1). As the values $\widetilde{M} = A_{\mathsf{os}} + r_M F, \widetilde{S}_0 = \beta S - r_x \widetilde{U}, \widetilde{T}_0 = \beta T - r_y \widetilde{U}$ are uniform over $\mathbb{G}$, the probability that the game aborts is at most Q_{H}/p^3. As there are at most two non-aborting calls to U_1, we have

$$\left| \mathsf{Pr}[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \right| \leqslant \frac{2Q_{\mathsf{H}}}{p^3}.$$

Game $\mathbf{G}_2$ (Simulate $(\widetilde{A}_{\mathsf{rnd}}, \tilde{c}_{\mathsf{rnd}}, \tilde{z}_{\mathsf{rnd}})$ in U_2): The game simulates the proof $(\widetilde{A}_{\mathsf{rnd}}, \tilde{c}_{\mathsf{rnd}}, \tilde{z}_{\mathsf{rnd}})$ instead of computing it honestly in U_2. That is, the game samples $\tilde{c}_{\mathsf{rnd}} \leftarrow\!\!\$\ \mathbb{Z}_p$ and sets $(\widetilde{A}_{\mathsf{rnd}}, \tilde{z}_{\mathsf{rnd}}) \leftarrow \Sigma_{\mathsf{rnd}}.\mathsf{Sim}(\mathbb{x}_{\mathsf{rnd}}, \tilde{c}_{\mathsf{rnd}})$. Further, it programs H such that $\mathsf{H}(\mathsf{in}_{\mathsf{H}}) = (\cdot, \cdot, \tilde{c}_{\mathsf{rnd}})$ in the last component, where in_{H} is defined as in $\mathbf{G}_1$ above. Else, it proceeds as in $\mathbf{G}_1$.

Note that the output of H on input in_{H} changes between U_1 and U_2. However, due to the abort condition introduced in $\mathbf{G}_1$, this change is not noticeable by the adversary and the H outputs are identically distributed in the view of $\mathcal{A}$. Further, note that $(\mathbb{x}_{\mathsf{rnd}}, \mathbb{w}_{\mathsf{rnd}}) \in \mathfrak{R}_{\mathsf{rnd}}$ by construction. Thus, by SHVZK of Σ_{rnd}, the Σ-protocol transcripts are identically distributed. We have

$$\mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_3$ (Simulate the OR-proof): Instead of randomizing the transcripts $(A_{\mathsf{dleq}}, c_{\mathsf{dleq}}, z_{\mathsf{dleq}})$ and $(A'_{\mathsf{mac}}, c_{\mathsf{mac}}, z'_{\mathsf{mac}})$ in both U_1 and U_2, the game simulates the OR-proof for $\mathfrak{L}_{\mathsf{mac}} \cup \mathfrak{L}_{\mathsf{dleq}}$ and programs H appropriately. Here, the transcript $(A'_{\mathsf{mac}}, c_{\mathsf{mac}}, z'_{\mathsf{mac}})$ denotes the adversarially-issued Σ_{mac} transcript after applying the modifications marked with in Figure 6, but before the other blinding operations are applied.

In more detail, the game sends a random challenge $c_{\mathsf{or}} \leftarrow\!\!\$\ \mathbb{Z}_p$ instead of $\tilde{c}_{\mathsf{or}} - \gamma_{\mathsf{mac}} - \gamma_{\mathsf{dleq}}$ in U_1. Further, it simulates the proofs $(\widetilde{A}_{\mathsf{dleq}}, \tilde{c}_{\mathsf{dleq}}, \tilde{z}_{\mathsf{dleq}})$ and $(\widetilde{A}_{\mathsf{mac}}, \tilde{c}_{\mathsf{mac}}, \tilde{z}_{\mathsf{mac}})$ in U_2. That is, in U_2, the game samples $\tilde{c}_{\mathsf{dleq}}, \tilde{c}_{\mathsf{or}} \leftarrow\!\!\$\ \mathbb{Z}_p$ and sets $\tilde{c}_{\mathsf{mac}} \leftarrow \tilde{c}_{\mathsf{or}} - \tilde{c}_{\mathsf{dleq}}$. Then, it computes $(\widetilde{A}_{\mathsf{dleq}}, \tilde{z}_{\mathsf{dleq}}) \leftarrow \Sigma_{\mathsf{dleq}}.\mathsf{Sim}(\mathbb{x}_{\mathsf{dleq}}, \tilde{c}_{\mathsf{dleq}})$ and $(\widetilde{A}_{\mathsf{mac}}, \tilde{z}_{\mathsf{mac}}) \leftarrow \Sigma_{\mathsf{mac}}.\mathsf{Sim}(\mathbb{x}'_{\mathsf{mac}}, \tilde{c}_{\mathsf{mac}})$. Finally, it programs H such that $\mathsf{H}(\mathsf{in}_{\mathsf{H}}) = (\cdot, \tilde{c}_{\mathsf{or}}, \cdot)$ in the second component, where in_{H} is defined as in $\mathbf{G}_1$.

Observe that in $\mathbf{G}_2$, the transcripts $(\widetilde{A}_{\mathsf{dleq}}, \tilde{c}_{\mathsf{dleq}}, \tilde{z}_{\mathsf{dleq}})$ and $(\widetilde{A}_{\mathsf{mac}}, \tilde{c}_{\mathsf{mac}}, \tilde{z}_{\mathsf{mac}})$ are obtained from $(A_{\mathsf{dleq}}, c_{\mathsf{dleq}}, z_{\mathsf{dleq}})$ and $(A'_{\mathsf{mac}}, c_{\mathsf{mac}}, z'_{\mathsf{mac}})$ by the randomization algorithm Rand_ϕ from Lemma 2.17 (for the corresponding ϕ linear maps), respectively. Also, note that the modified Σ_{mac} transcript $(A'_{\mathsf{mac}}, c_{\mathsf{mac}}, z'_{\mathsf{mac}})$ verifies if $(A_{\mathsf{mac}}, c_{\mathsf{mac}}, z_{\mathsf{mac}})$ verifies, as shown in section 5. As Σ_{dleq} and Σ_{mac} have strongly Rand_ϕ-randomizable transcripts (for the corresponding ϕ linear maps), the simulated transcripts are distributed identically to the randomized transcripts (cf. Corollary 2.16). It remains to argue that the oracle H is identically distributed in the view of $\mathcal{A}$. This follows by the abort condition introduced in $\mathbf{G}_1$ and as $\tilde{c}_{\mathsf{or}}$ is identically distributed in both games.

$$\mathsf{Pr}[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_4$ (Simulate $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ in U_2): Instead of randomizing the transcripts $(A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ in both U_1 and U_2, the game simulates the proof $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ and programs H appropriately. That is, the game sends a random challenge $c \leftarrow\!\!\$\ \mathbb{Z}_p$ instead of $\widetilde{c}_{\mathsf{os}} - \gamma_{\mathsf{os}}$ in U_1. Further, it simulates $(\widetilde{A}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}}) \leftarrow \Sigma_{\mathsf{os}}.\mathsf{Sim}(\mathbb{x}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}})$ in U_2 for $\widetilde{c}_{\mathsf{os}} \leftarrow\!\!\$\ \mathbb{Z}_p$ and programs H such that $\mathsf{H}(\mathsf{in}_{\mathsf{H}}) = (\widetilde{c}_{\mathsf{os}}, \cdot, \cdot)$ in the first component, where in_{H} is defined as in $\mathbf{G}_1$.

Observe that in $\mathbf{G}_3$, the transcript $(\widetilde{A}_{\mathsf{os}}, \widetilde{c}_{\mathsf{os}}, \widetilde{z}_{\mathsf{os}})$ is obtained from $(A_{\mathsf{os}}, c_{\mathsf{os}}, z_{\mathsf{os}})$ by the randomization algorithm $\mathsf{Rand}_{\phi_{\mathsf{os}}}$ from Lemma 2.17. As Σ_{os} has strongly $\mathsf{Rand}_{\phi_{\mathsf{os}}}$-randomizable transcripts, the simulated transcript is distributed identical to the randomized transcript (cf. Corollary 2.16). Furthermore, the oracle H is identically distributed in the view of $\mathcal{A}$ by the abort condition introduced in $\mathbf{G}_1$.

$$\mathsf{Pr}[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_5$ (Commit to zero in $\mathrm{cmt}_{\mathsf{rnd}}$): The game now commits to the zero vector over $\mathbb{G}^7$ in $\mathrm{cmt}_{\mathsf{rnd}}$. As the opening is not required anymore as witness in the Σ_{rnd} proof, it is straightforward to show that there exists an adversary $\mathcal{B}$ such that

$$\left|\mathsf{Pr}[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1]\right| = 2 \cdot \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{rnd}}}(\mathcal{B}, \lambda).$$

Note that all following game hops are statistical. We note that $\mathbf{G}_6$ and $\mathbf{G}_7$ are inefficient.

Game $\mathbf{G}_6$ (Abort if σ_{mac} is dishonest): The game (inefficiently) brute-forces the mac key $\mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho_0)$ and verifies that the obtained σ_{mac} in U_1 is honestly computed. That is, the game parses the issuer parameters ipp from pk as $\mathsf{ipp} = (\mathrm{cmt}_0, X_1, Y_1, Z)$ and computes x_0, x_1, y_0, y_1, z and ρ_0 such that $\mathrm{cmt}_0 = \mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (x_0, y_0); \rho_0), X_1 = x_1 F, Y_1 = y_1 F$ and $Z = zF$. In U_1, the game parses $\sigma_{\mathsf{mac}} = (U, V, S, T)$, brute-forces $u \in \mathbb{Z}_p^*$ such that $U = uG$ and checks that $V = zU, S = u(x_0 G + x_1 A_{\mathsf{os}})$ and $T = u(y_0 G + y_1 A_{\mathsf{os}})$. If a check fails, the game aborts, else the game proceeds as before.

Note that the game aborts iff $\mathbb{x}_{\mathsf{mac}} = (\mathsf{ipp}, M, U, V, S, T, 0, 0) \notin \mathfrak{L}_{\mathsf{mac}}$ as $\phi^{\mathsf{mac}}_{\mathsf{pp}, \mathbb{x}_{\mathsf{mac}}}$ is injective (i.e., the brute-forced witness msk is unique) by perfect strong binding of $\mathsf{C}_{\mathsf{mac}}$. Also, observe that as there are at most Q_{pb} queries to H_{pb}, the tuple $\mathbb{x}_{\mathsf{dleq}} = (D_1, D_2, D_3)$ is *not* a DDH-tuple (i.e., $\mathbb{x}_{\mathsf{dleq}} \notin \mathfrak{L}_{\mathsf{dleq}}$) except with probability Q_{pb}/p. In total, if an abort occurs, then $\mathbb{x}_{\mathsf{mac}} \notin \mathfrak{L}_{\mathsf{mac}}$ and $\mathbb{x}_{\mathsf{dleq}} \notin \mathfrak{L}_{\mathsf{dleq}}$ except with probability Q_{pb}/p. Further, the adversary provides A_{dleq} and A_{mac} in U_1. By special soundness of Σ_{dleq} and Σ_{rnd}, there is at most a single challenge pair $(c_{\mathsf{dleq}}, c_{\mathsf{mac}})$ such that there exists responses $(z_{\mathsf{dleq}}, z_{\mathsf{mac}})$ that complete the Σ-protocol transcripts (such that both transcripts verify). The probability that $c_{\mathsf{or}} = c_{\mathsf{dleq}} + c_{\mathsf{mac}}$ is at most $1/p$ and since there are at most two non-aborting U_1 calls in total, we obtain

$$\left|\mathsf{Pr}[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1]\right| \leqslant \frac{Q_{\mathsf{pb}} + 2}{p}.$$

Game $\mathbf{G}_7$ (Compute $\widetilde{\sigma}_{\mathsf{mac}}$ independently from σ_{mac}): The game computes the blinded MAC $\widetilde{\sigma}_{\mathsf{mac}}$ and the blinded message $\widetilde{M}$ as follows

$$\begin{aligned}
u &\leftarrow\!\!\$\ \mathbb{Z}_p^* , & \widetilde{M} &\leftarrow\!\!\$\ \mathbb{G}^* , \\
\widetilde{U} &\leftarrow uG , & \widetilde{V} &\leftarrow uzG , \\
\widetilde{S}_1 &\leftarrow\!\!\$\ \mathbb{G} , & \widetilde{S}_0 &\leftarrow u(x_0 G + x_1 \widetilde{M} - \widetilde{S}_1) , \\
\widetilde{T}_1 &\leftarrow\!\!\$\ \mathbb{G} , & \widetilde{T}_0 &\leftarrow u(y_0 G + y_1 \widetilde{M} - \widetilde{T}_1) ,
\end{aligned}$$

where $\mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho_0)$ is as in $\mathbf{G}_6$. Note that this is possible because the transcripts $(\widetilde{A}_{\mathsf{mac}}, \widetilde{c}_{\mathsf{mac}}, \widetilde{z}_{\mathsf{mac}})$ and $(\widetilde{A}_{\mathsf{rnd}}, \widetilde{c}_{\mathsf{rnd}}, \widetilde{z}_{\mathsf{rnd}})$ are simulated and the commitment $\mathrm{cmt}_{\mathsf{rnd}}$ commits to the zero vector.

It is easy to see that $(\widetilde{\sigma}_{\mathsf{mac}}, \widetilde{M})$ is identically distributed in $\mathbf{G}_6$ and $\mathbf{G}_7$ due to the abort condition introduced in $\mathbf{G}_6$. That is, in $\mathbf{G}_6$, we have $u \in \mathbb{Z}_p^*$ such that $U = uG$ and $V = zU, S = u(x_0 G + x_1 A_{\mathsf{os}})$ and

$T = u(y_0 G + y_1 A_{\mathsf{os}})$. Further, $(\widetilde{\sigma}_{\mathsf{mac}}, \widetilde{M})$ is computed from (U, V, S, T) and A_{os} as follows

$$\begin{aligned}
&\beta \leftarrow_{\$} \mathbb{Z}_p^*, \quad r_x, r_y, r_M \leftarrow_{\$} \mathbb{Z}_p \ , && \widetilde{M} \leftarrow A_{\mathsf{os}} + r_M F \ , \\
&\widetilde{U} \leftarrow \beta U \ , && \widetilde{V} \leftarrow \beta V \ , \\
&\widetilde{S}_1 \leftarrow r_x G + r_M X_1 \ , && \widetilde{S}_0 \leftarrow \beta S - r_x \widetilde{U} \ , \\
&\widetilde{T}_1 \leftarrow r_y G + r_M Y_1 \ , && \widetilde{T}_0 \leftarrow \beta T - r_y \widetilde{U} \ .
\end{aligned}$$

Recall that $x_1 F = X_1$ and $y_1 F = Y_1$. The claim follows since the values $\widetilde{M}, \widetilde{U} = (\beta u)G, \widetilde{V} = (\beta u z)G$ in $\mathbf{G}_6$ trivially follow the distribution from $\mathbf{G}_7$ and further, in $\mathbf{G}_6$ we have

$$\begin{aligned}
\beta u(x_0 G + x_1 \widetilde{M} - \widetilde{S}_1) &= \beta u(x_0 G + x_1(A_{\mathsf{os}} + r_M F) - r_x G - r_M(x_1 F)) \\
&= \beta u(x_0 G + x_1 A_{\mathsf{os}} - r_x G) \\
&= \beta S - r_x \widetilde{U} \\
&= \widetilde{S}_0
\end{aligned}$$

so also $(\widetilde{S}_0, \widetilde{S}_1)$ follow the distribution from $\mathbf{G}_7$, and similarly for $(\widetilde{T}_0, \widetilde{T}_1)$. In total, we obtain

$$\mathsf{Pr}[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1] = \mathsf{Pr}[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1].$$

Advantage of $\mathcal{A}$ in $\mathbf{G}_7$. Observe that the view of $\mathcal{A}$ in $\mathbf{G}_7$ is independent from the bit b sampled by the game. Thus, we have $b = b'$ with probability $1/2$, therefore

$$\mathsf{Pr}[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1] = \frac{1}{2}.$$

The claim follows by collecting all bounds. □

Acknowledgements

Chairattana-Apirom and Tessaro's research was partially supported by NSF grants CNS-2026774, CNS-2154174, CNS-2426905, a gift from Microsoft, and a Stellar Development Foundation Academic Research Award.

References

Abe01. Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 136–151. Springer, Berlin, Heidelberg, May 2001.

AF96. Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 244–251. Springer, Berlin, Heidelberg, November 1996.

AHJ21. Nabil Alkeilani Alkadri, Patrick Harasser, and Christian Janson. BlindOR: an efficient lattice-based blind signature scheme from OR-proofs. In Mauro Conti, Marc Stevens, and Stephan Krenn, editors, *CANS 21*, volume 13099 of *LNCS*, pages 95–115. Springer, Cham, December 2021.

Ame05. American National Standards Institute, Inc. ANSI X9.62 public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ECDSA), November 16, 2005.

AO00. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286. Springer, Berlin, Heidelberg, August 2000.

App. iCloud private relay overview. https://www.apple.com/privacy/docs/iCloud_Private_Relay_Overview_Dec2021.PDF.

BBKS07. Mihir Bellare, Alexandra Boldyreva, Kaoru Kurosawa, and Jessica Staddon. Multi-recipient encryption schemes: Efficient constructions and their security. *IEEE Transactions on Information Theory*, 53(11):3927–3943, 2007.

BCJ08. Ali Bagherzandi, Jung Hee Cheon, and Stanislaw Jarecki. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008*, pages 449–458. ACM Press, October 2008.

BDL+12. Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, September 2012.

BHKR25. Nicholas Brandt, Dennis Hofheinz, Michael Klooß, and Michael Reichle. Tightly-secure blind signatures in pairing-free groups. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VI*, volume 16250 of *LNCS*, pages 337–369. Springer, Singapore, December 2025.

BLL+21. Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On the (in)security of ROS. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 33–53. Springer, Cham, October 2021.

BLL+22. Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On the (in)security of ROS. *Journal of Cryptology*, 35(4):25, October 2022.

BN06. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006.

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

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

CAHLT25. Rutchathon Chairattana-Apirom, Franklin Harding, Anna Lysyanskaya, and Stefano Tessaro. Server-aided anonymous credentials. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 291–324. Springer, Cham, August 2025.

CATZ24. Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. Pairing-free blind signatures from CDH assumptions. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 174–209. Springer, Cham, August 2024.

CDVW24. Sofia Celi, Alex Davidson, Steven Valdez, and Christopher A. Wood. Privacy Pass Issuance Protocols. RFC 9578, June 2024.

CFN90. David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 319–327. Springer, New York, August 1990.

Cha82. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

Che26. Yanbo Chen. Three-move blind signatures in pairing-free groups. To appear at CRYPTO 2026, 2026.

CKM+23. Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Snowblind: A threshold blind signature in pairing-free groups. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 710–742. Springer, Cham, August 2023.

CMV26. Michele Ciampi, Pierpaolo Della Monica, and Ivan Visconti. Round-optimal GUC-secure blind signatures from minimal computational and setup assumptions. Cryptology ePrint Archive, Paper 2026/145, 2026.

CMZ14. Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic MACs and keyed-verification anonymous credentials. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014*, pages 1205–1216. ACM Press, November 2014.

CP93. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 89–105. Springer, Berlin, Heidelberg, August 1993.

DGS+18. Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *PoPETs*, 2018(3):164–180, July 2018.

DKT26. Marian Dietz, Julia Kastner, and Stefano Tessaro. On the impossibility of round-optimal pairing-free blind signatures in the ROM. To appear at CRYPTO 2026, 2026.

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

Fis06. Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, Berlin, Heidelberg, August 2006.

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

FOO93. Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Yuliang Zheng, editors, *AUSCRYPT'92*, volume 718 of *LNCS*, pages 244–251. Springer, Berlin, Heidelberg, December 1993.

FPS20. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind Schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Cham, May 2020.

Goo. VPN by Google One, explained. https://one.google.com/about/vpn/howitworks.

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

HIJ+21. Sharon Huang, Subodh Iyengar, Sundar Jeyaraman, Shiv Kushwah, Chen-Kuei Lee, Zutian Luo, Payman Mohassel, Ananth Raghunathan, Shaahid Shaikh, Yen-Chieh Sung, and Albert Zhang. DIT: De-Identified Authenticated Telemetry at Scale. Technical report, April 2021.

HIP+21. Scott Hendrickson, Jana Iyengar, Tommy Pauly, Steven Valdez, and Christopher A. Wood. Private Access Tokens. Internet-Draft draft-private-access-tokens-01, Internet Engineering Task Force, October 2021. Work in Progress.

HKL19. Eduard Hauck, Eike Kiltz, and Julian Loss. A modular treatment of blind signatures from identification schemes. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 345–375. Springer, Cham, May 2019.

JLO97. Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski, Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 150–164. Springer, Berlin, Heidelberg, August 1997.

KLR25. Michael Klooß, Russell W. F. Lai, and Michael Reichle. Blind signatures from arguments of inequality. Cryptology ePrint Archive, Report 2025/1886, 2025.

KLX22a. Julia Kastner, Julian Loss, and Jiayu Xu. The Abe-Okamoto partially blind signature scheme revisited. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 279–309. Springer, Cham, December 2022.

KLX22b. Julia Kastner, Julian Loss, and Jiayu Xu. On pairing-free blind signature schemes in the algebraic group model. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 468–497. Springer, Cham, March 2022.

KR25. Michael Klooß and Michael Reichle. Blind signatures from proofs of inequality. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 157–189. Springer, Cham, August 2025.

KRS23. Shuichi Katsumata, Michael Reichle, and Yusuke Sakai. Practical round-optimal blind signatures in the ROM from standard assumptions. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part II*, volume 14439 of *LNCS*, pages 383–417. Springer, Singapore, December 2023.

KRW24. Michael Klooß, Michael Reichle, and Benedikt Wagner. Practical blind signatures in pairing-free groups. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part I*, volume 15484 of *LNCS*, pages 363–395. Springer, Singapore, December 2024.

KTZ26. Julia Kastner, Stefano Tessaro, and Greg Zaverucha. Round-optimal pairing-free blind signatures. Cryptology ePrint Archive, Paper 2026/091, 2026.

Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Berlin, Heidelberg, December 2005.

Mau15. Ueli Maurer. Zero-knowledge proofs of knowledge for group homomorphisms. *DCC*, 77(2-3):663–676, 2015.

OA03. Miyako Ohkubo and Masayuki Abe. Security of some three-move blind signature schemes reconsidered. In *The 2003 Symposium on Cryptography and Information Security*, 2003.

Oka93. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Berlin, Heidelberg, August 1993.

Oka94. Tatsuaki Okamoto. Designated confirmer signatures and public-key encryption are equivalent. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 61–74. Springer, Berlin, Heidelberg, August 1994.

OO92. Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 324–337. Springer, Berlin, Heidelberg, August 1992.

OTZZ24. Michele Orrù, Stefano Tessaro, Greg Zaverucha, and Chenzhi Zhu. Oblivious issuance of proofs. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IX*, volume 14928 of *LNCS*, pages 254–287. Springer, Cham, August 2024.

PCM. PCM: Click fraud prevention and attribution sent to advertiser. https://webkit.org/blog/11940/pcm-click-fraud-prevention-and-attribution-sent-to-advertiser/. Accessed: 2021-09-30.

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

RR26. Michael Reichle and Zoé Reinke. Threshold blind signatures from cdh. In Shi Bai and Edoardo Persichetti, editors, *Public-Key Cryptography – PKC 2026*, pages 269–301, Cham, 2026. Springer Nature Switzerland.

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, New York, August 1990.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Berlin, Heidelberg, May 1997.

Tru. Trust tokens. https://developer.chrome.com/docs/privacy-sandbox/trust-tokens/.

TZ22. Stefano Tessaro and Chenzhi Zhu. Short pairing-free blind signatures with exponential security. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 782–811. Springer, Cham, May / June 2022.

TZ23. Stefano Tessaro and Chenzhi Zhu. Threshold and multi-signature schemes from linear hash functions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 628–658. Springer, Cham, April 2023.

A Proof of Lemma 2.5

Let $\mathcal{A}$ be the adversary running in time $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$ and making $Q_{\mathrm{S}_1} = Q_{\mathrm{S}_1}(\lambda), Q_{\mathrm{S}_2} = Q_{\mathrm{S}_2}(\lambda)$ to $\mathrm{S}_1, \mathrm{S}_2$, respectively. We consider the following adversary $\mathcal{B}$ simulating the OMUF/OMSUF game to $\mathcal{A}$ while playing the OMUF/OMSUF game itself.

Parameters and key. The adversary $\mathcal{B}$ takes as input $(\mathsf{pp}, \mathsf{vk})$ from the game and forwards them to $\mathcal{A}$.

Signing oracle S_1. When $\mathcal{A}$ makes a query info to S_1, $\mathcal{B}$ forwards the query to its own S_1 oracle, while keeping track of the counter cnt of the number of sessions opened and set $\mathsf{info}_{\mathsf{cnt}} \leftarrow \mathsf{info}$. Then, forward the output $\mathsf{smsg}_{1,\mathsf{cnt}}$ from the oracle to $\mathcal{A}$.

Signing oracle S_2. When $\mathcal{A}$ makes a query $(\mathrm{sid}, \mathsf{umsg})$ to S_2, $\mathcal{B}$ forwards the query to S_2. Then, forwards the output $\mathsf{smsg}_{2,\mathrm{sid}}$ back to $\mathcal{A}$.

Finishing opened sessions. When $\mathcal{A}$ returns the forgeries $(\mathsf{info}^*, (\mu^{(i)}, \sigma^{(i)})_{i\in[\ell]})$, $\mathcal{B}$ finishes the unclosed sessions as follows. For each sid such that $\mathcal{A}$ does not make S_2 query on,

(a) Pick μ_{sid} such that $\mu_{\mathrm{sid}} \notin \{\mu^{(i)}\}_{i\in[\ell]}$ and $\mathcal{B}$ has not used μ_{sid} to finish a session before,
(b) Compute $(\mathsf{umsg}_{\mathrm{sid}}, \mathsf{st}^u_{\mathrm{sid}}) \leftarrow\!\!\$\ \mathsf{BS}_{\mathrm{alg}}.\mathsf{U}_1(\mathsf{pp}, \mathsf{vk}, \mathsf{info}_{\mathrm{sid}}, \mu_{\mathrm{sid}}, \mathsf{smsg}_{1,\mathrm{sid}})$,
(c) Query $\mathsf{smsg}_{2,\mathrm{sid}} \leftarrow \mathrm{S}_2(\mathrm{sid}, \mathsf{umsg}_{\mathrm{sid}})$, and
(d) Compute $\sigma^*_{\mathrm{sid}} \leftarrow \mathsf{BS}_{\mathrm{alg}}.\mathsf{U}_2(\mathsf{smsg}_{2,\mathrm{sid}}, \mathsf{st}^u_{\mathrm{sid}})$.

Forgery. Let S be the set of sid with $\mathsf{info}^* = \mathsf{info}_{\mathrm{sid}}$ and $\mathcal{B}$ closes the signing sessions in the previous step. Then, $\mathcal{B}$ returns $(\mathsf{info}^*, (\mu^{(i)}, \sigma^{(i)})_{i\in[\ell]} \cup (\mu_{\mathrm{sid}}, \sigma^*_{\mathrm{sid}})_{\mathrm{sid}\in S})$

Note that by correctness of $\mathsf{BS}_{\mathrm{alg}}$ the pairs $(\mu_{\mathrm{sid}}, \sigma^*_{\mathrm{sid}})_{\mathrm{sid}\in S}$ are valid signatures with respect to pp, vk and info^*. Thus, if $\mathcal{A}$ wins in OMUF/OMSUF game, $\mathcal{B}$ wins as well, implying the advantage bound. Accordingly, the query count for $\mathcal{B}$ is exactly Q_{S_1} for both oracles and the running time of $\mathcal{B}$ is roughly that of $\mathcal{A}$. □

B Proof of Lemma 2.19

We first recall the splitting lemma in the version used in [BCJ08]. Our proof resembles that of [BCJ08] except that we do not rewind for multiple times per index, but only once.

Lemma B.1 (Splitting Lemma [BCJ08]). *Let $A, B, S \subseteq \mathcal{X} \times \mathcal{Y}$, $A \subseteq S$ with $\mathsf{Pr}[S] > 0$, and $B := \{(x, y) : \mathsf{Pr}_{y' \leftarrow\!\!\$\ \mathcal{Y}}[(x, y') \in A] \geqslant \delta\}$ for $\delta \in [0, 1]$. Then,*

1. $\mathsf{Pr}[A \cap B] \geqslant \mathsf{Pr}[A] - \delta$
2. $\mathsf{Pr}[A \cap B|S] \geqslant \mathsf{Pr}[A|S] - \delta/\mathsf{Pr}[S]$

We first consider the following sets of inputs $(\mathsf{inp}, \rho, \boldsymbol{h})$ of $\mathcal{A}$ where $\mathsf{inp} \in [\mathsf{IG}()]$, ρ is the random coins of $\mathcal{A}$, and $\boldsymbol{h} \in \mathcal{C}^Q$.

- $\mathcal{S} := \{(\mathsf{inp}, \rho, \boldsymbol{h}) : \mathcal{A}(\mathsf{inp}, \boldsymbol{h}; \rho) \neq \bot\}$ be the set of inputs where $\mathcal{A}$ succeeds.
- For each $i \in [Q]$, let $\mathcal{W}_i := \{(\mathsf{inp}, \rho, \boldsymbol{h}) : (I, \mathsf{aux}) \leftarrow \mathcal{A}(\mathsf{inp}, \boldsymbol{h}; \rho), i \in I\} = \bigcup_{j\in\{0,1\}} W^j_i$ be the set of inputs where the output of $\mathcal{A}$ contains index i.
- For each $j \in \{0, 1\}, i \in [Q]$, let $\mathcal{W}^j_i := \{(\mathsf{inp}, \rho, \boldsymbol{h}) : (I, \mathsf{aux}) \leftarrow \mathcal{A}(\mathsf{inp}, \boldsymbol{h}; \rho); i = I_j\} \subseteq \mathcal{S}$ be the set of inputs where the j-th position of the output I of $\mathcal{A}$ is i.
- For each $i \in [Q]$, let $\mathsf{Heavy}_i := \{(\mathsf{inp}, \rho, \boldsymbol{h}) : \mathsf{Pr}_{\boldsymbol{h}'_{[i:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-i+1}}[(\mathsf{inp}, \rho, \boldsymbol{h}_{1:i-1} \| \boldsymbol{h}'_{[i:Q]}) \in \mathcal{W}_i] \geqslant \delta\}$. Here, Heavy_i defines the set of inputs where after forking at index i, the forked run outputs index i.
- Accordingly, for each $j \in \{0, 1\}, i \in [Q]$, let

$$\mathsf{Heavy}^j_i := \{(\mathsf{inp}, \rho, \boldsymbol{h}) : \mathsf{Pr}_{\boldsymbol{h}'_{[i:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-i+1}}[(\mathsf{inp}, \rho, \boldsymbol{h}_{1:i-1} \| \boldsymbol{h}'_{[i:Q]}) \in \mathcal{W}^j_i] \geqslant \delta\} .$$

  Note that since $\mathcal{W}^j_i \subseteq \mathcal{W}_i$, we can see that $\mathsf{Heavy}^j_i \subseteq \mathsf{Heavy}_i$.
- For each $j \in \{0, 1\}$, let $E_j = \bigcup_{i\in[Q]}(\mathcal{W}^j_i \cap \mathsf{Heavy}_i) \subseteq \mathcal{S}$ be the set of inputs that for the j-th position I_j of the output of $\mathcal{A}$, the input is in Heavy_{I_j}.

The general proof strategy is that if the first run of $\mathcal{A}$ within Fork_3 leads to the input that is in $E = E_0 \cap E_1$, the latter runs are much likelier to result in the forking. In particular, we have that

$$
\begin{aligned}
\mathsf{frk} &= \mathsf{Pr}_{\mathsf{inp},\rho}\left[\begin{array}{l|l} (\mathsf{inp},\rho,\boldsymbol{h}) \in \mathcal{S} \ \wedge & \boldsymbol{h} \leftarrow\!\!\$\ \mathcal{C}^Q \\ (\mathsf{inp},\rho,\boldsymbol{h}') \in \mathcal{W}_{I_0} \ \wedge & ((I_0,I_1),(\mathsf{aux}_{I_k})_{k\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{inp},\boldsymbol{h};\rho_{\mathcal{A}}) \\ (\mathsf{inp},\rho,\boldsymbol{h}'') \in \mathcal{W}_{I_1} \ \wedge & \boldsymbol{h}'_{[I_0:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_0+1}\ ; \boldsymbol{h}' = \boldsymbol{h}_{[1:I_0-1]} \| \boldsymbol{h}'_{[I_0:Q]} \\ h'_{I_0} \neq h_{I_0} \ \wedge \ h'_{I_1} \neq h_{I_1} & \boldsymbol{h}''_{[I_1:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_1+1}\ ; \boldsymbol{h}'' = \boldsymbol{h}_{[1:I_1-1]} \| \boldsymbol{h}''_{[I_1:Q]} \end{array}\right] \\
&\geqslant \mathsf{Pr}_{\mathsf{inp},\rho}\left[\begin{array}{l|l} (\mathsf{inp},\rho,\boldsymbol{h}) \in E \ \wedge & \boldsymbol{h} \leftarrow\!\!\$\ \mathcal{C}^Q \\ (\mathsf{inp},\rho,\boldsymbol{h}') \in \mathcal{W}_{I_0} \ \wedge & ((I_0,I_1),(\mathsf{aux}_{I_k})_{k\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{inp},\boldsymbol{h};\rho_{\mathcal{A}}) \\ (\mathsf{inp},\rho,\boldsymbol{h}'') \in \mathcal{W}_{I_1} \ \wedge & \boldsymbol{h}'_{[I_0:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_0+1}\ ; \boldsymbol{h}' = \boldsymbol{h}_{[1:I_0-1]} \| \boldsymbol{h}'_{[I_0:Q]} \\ h'_{I_0} \neq h_{I_0} \ \wedge \ h'_{I_1} \neq h_{I_1} & \boldsymbol{h}''_{[I_1:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_1+1}\ ; \boldsymbol{h}'' = \boldsymbol{h}_{[1:I_1-1]} \| \boldsymbol{h}''_{[I_1:Q]} \end{array}\right] \\
&= \mathsf{Pr}_{\mathsf{inp},\rho,\boldsymbol{h}}[(\mathsf{inp},\rho,\boldsymbol{h}) \in E] \\
&\qquad \cdot \mathsf{Pr}_{\mathsf{inp},\rho}\left[\begin{array}{l|l} & \boldsymbol{h} \leftarrow\!\!\$\ \mathcal{C}^Q \\ (\mathsf{inp},\rho,\boldsymbol{h}') \in \mathcal{W}_{I_0} \ \wedge & (\mathsf{inp},\rho,\boldsymbol{h}) \in E \\ (\mathsf{inp},\rho,\boldsymbol{h}'') \in \mathcal{W}_{I_1} \ \wedge & ((I_0,I_1),(\mathsf{aux}_{I_k})_{k\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{inp},\boldsymbol{h};\rho_{\mathcal{A}}) \\ h'_{I_0} \neq h_{I_0} \ \wedge \ h'_{I_1} \neq h_{I_1} & \boldsymbol{h}'_{[I_0:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_0+1}\ ; \boldsymbol{h}' = \boldsymbol{h}_{[1:I_0-1]} \| \boldsymbol{h}'_{[I_0:Q]} \\ & \boldsymbol{h}''_{[I_1:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_1+1}\ ; \boldsymbol{h}'' = \boldsymbol{h}_{[1:I_1-1]} \| \boldsymbol{h}''_{[I_1:Q]} \end{array}\right] \\
&= \mathsf{Pr}[E] \cdot \\
&\qquad \sum_{i_0,i_1\in[Q]} \mathsf{Pr}_{\mathsf{inp},\rho}\left[\begin{array}{l|l} I_0 = i_0 \ \wedge \ I_1 = i_1 \ \wedge & \boldsymbol{h} \leftarrow\!\!\$\ \mathcal{C}^Q \\ (\mathsf{inp},\rho,\boldsymbol{h}') \in \mathcal{W}_{I_0} \ \wedge & (\mathsf{inp},\rho,\boldsymbol{h}) \in E \\ (\mathsf{inp},\rho,\boldsymbol{h}'') \in \mathcal{W}_{I_1} \ \wedge & ((I_0,I_1),(\mathsf{aux}_{I_k})_{k\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{inp},\boldsymbol{h};\rho_{\mathcal{A}}) \\ h'_{I_0} \neq h_{I_0} \ \wedge \ h'_{I_1} \neq h_{I_1} & \boldsymbol{h}'_{[I_0:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_0+1}\ ; \boldsymbol{h}' = \boldsymbol{h}_{[1:I_0-1]} \| \boldsymbol{h}'_{[I_0:Q]} \\ & \boldsymbol{h}''_{[I_1:Q]} \leftarrow\!\!\$\ \mathcal{C}^{Q-I_1+1}\ ; \boldsymbol{h}'' = \boldsymbol{h}_{[1:I_1-1]} \| \boldsymbol{h}''_{[I_1:Q]} \end{array}\right] \\
&= \mathsf{Pr}[E] \sum_{i_0,i_1\in[Q]} \mathsf{Pr}_{\mathsf{inp},\rho,\boldsymbol{h},\boldsymbol{h}',\boldsymbol{h}''}\left[\begin{array}{l|l} (\mathsf{inp},\rho,\boldsymbol{h}) \in W^0_{i_0} \cap W^1_{i_1} \ \wedge & (\mathsf{inp},\rho,\boldsymbol{h}) \in E \\ (\mathsf{inp},\rho,\boldsymbol{h}') \in \mathcal{W}_{i_0} \ \wedge & \boldsymbol{h}'_{[1:i_0-1]} = \boldsymbol{h}_{[1:i_0-1]} \\ (\mathsf{inp},\rho,\boldsymbol{h}'') \in \mathcal{W}_{i_1} \ \wedge & \boldsymbol{h}''_{[1:i_1-1]} = \boldsymbol{h}_{[1:i_1-1]} \\ h'_{i_0} \neq h_{i_0} \ \wedge \ h'_{i_1} \neq h_{i_1} & \end{array}\right] \\
&= \mathsf{Pr}[E] \sum_{i_0,i_1\in[Q]} \mathsf{Pr}_{\mathsf{inp},\rho,\boldsymbol{h}}[(\mathsf{inp},\rho,\boldsymbol{h}) \in W^0_{i_0} \cap W^1_{i_1} | E] \\
&\qquad \cdot \mathsf{Pr}_{\mathsf{inp},\rho,\boldsymbol{h},\boldsymbol{h}',\boldsymbol{h}''}\left[\begin{array}{l|l} (\mathsf{inp},\rho,\boldsymbol{h}') \in \mathcal{W}_{i_0} \ \wedge & (\mathsf{inp},\rho,\boldsymbol{h}) \in E \cap W^0_{i_0} \cap W^1_{i_1} \\ (\mathsf{inp},\rho,\boldsymbol{h}'') \in \mathcal{W}_{i_1} \ \wedge & \boldsymbol{h}'_{[1:i_0-1]} = \boldsymbol{h}_{[1:i_0-1]} \\ h'_{i_0} \neq h_{i_0} \ \wedge \ h'_{i_1} \neq h_{i_1} & \boldsymbol{h}''_{[1:i_1-1]} = \boldsymbol{h}_{[1:i_1-1]} \end{array}\right] .
\end{aligned}
$$

The first line is by definition of frk and Fork_3. The second line is by $E \subseteq \mathcal{S}$. The third line is by conditional probability. The fourth line is by law of total probability over all the possible values of I_0, I_1 (here, we also use shorthand $\mathsf{Pr}[E]$ to denote $(\mathsf{inp},\rho,\boldsymbol{h}) \in E$). The fifth line is only rewriting the probability notation and that $I_0 = i_0, I_1 = i_1$. The last equality is due to conditional probability.

Note that for $(\mathsf{inp},\rho,\boldsymbol{h}) \in E \cap W^0_{i_0} \cap W^1_{i_1}$, we have that $(\mathsf{inp},\rho,\boldsymbol{h}) \in \mathsf{Heavy}_{i_0} \cap \mathsf{Heavy}_{i_1}$ by definition of E. Therefore, by definition of Heavy_i,

$$
\mathsf{Pr}_{\mathsf{inp},\rho,\boldsymbol{h},\boldsymbol{h}',\boldsymbol{h}''}\left[\begin{array}{l|l} (\mathsf{inp},\rho,\boldsymbol{h}') \in \mathcal{W}_{i_0} \ \wedge & (\mathsf{inp},\rho,\boldsymbol{h}) \in E \cap W^0_{i_0} \cap W^1_{i_1} \\ (\mathsf{inp},\rho,\boldsymbol{h}'') \in \mathcal{W}_{i_1} \ \wedge & \boldsymbol{h}'_{[1:i_0-1]} = \boldsymbol{h}_{[1:i_0-1]} \\ h'_{I_0} \neq h_{I_0} \ \wedge \ h'_{I_1} \neq h_{I_1} & \boldsymbol{h}''_{[1:i_1-1]} = \boldsymbol{h}_{[1:i_1-1]} \end{array}\right] \geqslant \left(\delta - \frac{1}{|\mathcal{C}|}\right)^2 .
$$

Therefore, we can lower bound the sum as $(\delta - 1/|\mathcal{C}|)^2$.

Finally, we lower bound $\mathsf{Pr}[E]$.

$$
\mathsf{Pr}[E] = \mathsf{Pr}[\mathcal{S}]\mathsf{Pr}[E|\mathcal{S}] = \mathsf{acc} \cdot \mathsf{Pr}[E_0 \cap E_1 | \mathcal{S}]
$$

$$\geqslant \mathsf{acc} \cdot (1 - \sum_{j \in \{0,1\}} \mathsf{Pr}[\neg E_j | \mathcal{S}])$$

Thus, we now upper bound $\mathsf{Pr}[\neg E_j|\mathcal{S}]$, or equivalently, lower bound $\mathsf{Pr}[E_j|\mathcal{S}]$. In particular, we claim that $\mathsf{Pr}[E_j|\mathcal{S}] \geqslant 1 - Q\delta \cdot \mathsf{acc}^{-1}$. Here, observe that $\mathcal{W}_i^j$ are disjoint for $i \in [Q]$, meaning

$$\begin{aligned}
\mathsf{Pr}[E_j|\mathcal{S}] &= \sum_{i \in [Q]} \mathsf{Pr}[\mathcal{W}_i^j \cap \mathsf{Heavy}_i | \mathcal{S}] \\
&\geqslant \sum_{i \in [Q]} \mathsf{Pr}[\mathcal{W}_i^j \cap \mathsf{Heavy}_i^j | \mathcal{S}] \\
&\geqslant \sum_{i \in [Q]} \mathsf{Pr}[\mathcal{W}_i^j | \mathcal{S}] - \delta \mathsf{Pr}[\mathcal{S}]^{-1} \\
&1 - Q\delta \cdot \mathsf{acc}^{-1} \ .
\end{aligned}$$

The second line follows from $\mathsf{Heavy}_i^j \subseteq \mathsf{Heavy}_i$. The third line follows from the splitting lemma. The final line follows from $\mathcal{S} = \bigcup_{i \in [Q]} \mathcal{W}_i^j$ since at index j, I_j needs to take on exactly one value from $[Q]$.

Finally, setting $\delta = \frac{\mathsf{acc}}{4Q}$, we have that

$$\begin{aligned}
\mathsf{frk} &\geqslant \mathsf{Pr}[E] \left(\delta - \frac{1}{|\mathcal{C}|}\right)^2 \\
&\geqslant \mathsf{acc}(1 - \sum_{j \in \{0,1\}} \mathsf{Pr}[\neg E_j | \mathcal{S}]) \left(\frac{\mathsf{acc}}{4Q} - \frac{1}{|\mathcal{C}|}\right)^2 \\
&\geqslant \mathsf{acc}(1 - 2Q\delta \cdot \mathsf{acc}^{-1}) \left(\frac{\mathsf{acc}}{4Q} - \frac{1}{|\mathcal{C}|}\right)^2 \\
&\geqslant \frac{\mathsf{acc}}{2} \left(\frac{\mathsf{acc}}{4Q} - \frac{1}{|\mathcal{C}|}\right)^2 \ .
\end{aligned}$$

Therefore, we have that $\mathsf{frk} \geqslant \frac{1}{32Q^2} \left(\mathsf{acc} - \frac{4Q}{|\mathcal{C}|}\right)^3$, which implies that $\mathsf{acc} \leqslant \sqrt[3]{32Q^2 \cdot \mathsf{frk}} + \frac{4Q}{|\mathcal{C}|}$. □

Remark B.2. We only state the lemma for 2-index forking. However, we can extend the proof to the case where $I \in [Q]^\ell$ for $\ell \leqslant Q$ instead of $\ell = 2$ with similar techniques. In particular, the bound becomes

$$\mathsf{frk} \geqslant \frac{\mathsf{acc}}{2} \left(\frac{\mathsf{acc}}{2\ell Q} - \frac{1}{|\mathcal{C}|}\right)^\ell \ .$$

C Proof of Theorem 3.1

We give the simple proof for the algebraic MAC for the self-containment of the paper. Let $\mathcal{A}$ be an adversary making at most Q Tag-oracle queries. At the end of the game, it outputs $(m^*, \sigma^*_{\mathsf{alg}})$ and wins if $\mathsf{M}_{\mathsf{alg}}.\mathsf{Ver}(\mathsf{mpp}, \mathsf{ipp}, \mathsf{msk}, m^*, \sigma^*_{\mathsf{alg}}) = 1$. Then, we consider the following sequence of games $\mathbf{G}_0, \mathbf{G}_1$.

Game $\mathbf{G}_0$ (Original game.): This is the original UF-RMA game for $\mathsf{M}_{\mathsf{alg}}$ such that $\mathsf{mpp} = (F, \mathsf{mck})$, where $F \leftarrow\!\!\$\ \mathbb{G}$ and $\mathsf{mck} \leftarrow\!\!\$\ \mathsf{C}_{\mathsf{mac}}.\mathsf{Setup}(1^\lambda)$, the issuer's public parameters $\mathsf{ipp} = (\mathrm{cmt}_0, X_1, Y_1, Z)$ computed from the secret key $\mathsf{msk} = (x_0, x_1, y_0, y_1, z, \rho)$ as

$$\mathrm{cmt}_0 = \mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (x_0, y_0); \rho), X_1 = x_1 F, Y_1 = y_1 F, Z = zF$$

Game $\mathbf{G}_1$ (cmt_0 commits to $(0,0)$.): In this game, cmt_0 is now $\mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (0,0); \rho)$ for $\rho \leftarrow\!\!\$\ \mathsf{C}_{\mathsf{mac}}.\mathcal{R}$ instead. We can argue that

$$|\mathsf{Pr}[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \mathsf{Pr}[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1]| \leqslant \mathsf{Adv}^{\mathsf{hide}}_{\mathsf{C}_{\mathsf{mac}}}(\mathcal{B}_1, \lambda) \ ,$$

by constructing $\mathcal{B}_1$ which picks two messages to be committed $(x_0, y_0) \leftarrow\!\!\$\ \mathbb{Z}_p^2$ and $(0,0)$. When $\mathcal{B}_1$ receives cmt_0, it simulates Tag as in the original game using (x_0, x_1, y_0, y_1, z). At the end of the game, output 1 if $\mathcal{A}$ returns a valid forgery; otherwise output 0. The reduction does not need ρ to simulate the Tag oracle nor the verification at the end. This is by how $\mathsf{M}_{\mathsf{alg}}.\mathsf{Tag}$ and $\mathsf{M}_{\mathsf{alg}}.\mathsf{Ver}$ are defined. Thus, we have shown the advantage change. Note that $\mathcal{B}_1$ runs in time roughly $T_{\mathcal{A}}$.

Finally, we refer to [CAHLT25, Theorem 6.1] adapted to UF-RMA security. The original theorem proves the stronger UF-CMVA security of $\mathsf{M}'_{\mathsf{alg}}$, which is $\mathsf{M}_{\mathsf{alg}}$ without the commitment cmt_0.

Theorem C.1 (Theorem 6.1 of [CAHLT25] adapted). *Let* GGen *be a group parameters generator outputting groups of prime-order* $p = p(\lambda)$*,* $\mathsf{M}'_{\mathsf{alg}} = \mathsf{M}'_{\mathsf{alg}}[\mathsf{GGen}]$ *be the algebraic MAC scheme identical to* $\mathsf{M}_{\mathsf{alg}}$ *except that* mpp *does not contain* mck *and* ipp *does not contain* cmt_0*. For any* $\mathcal{A}$ *running in time* $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$ *and making at most* $Q = Q(\lambda)$ *queries to the* Tag *oracle, there exist adversaries* $\mathcal{B}$ *running in time* $\approx T_{\mathcal{A}}$ *such that*

$$\mathsf{Adv}^{\mathsf{ufrma}}_{\mathsf{M}'_{\mathsf{alg}}}(\mathcal{A}, \lambda) \leqslant Q \cdot \mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{GGen}}(\mathcal{B}, \lambda) + \frac{3}{p}\,.$$

Our final reduction $\mathcal{B}_2$ can simulate the game $\mathbf{G}_1$ to $\mathcal{A}$ by computing $\mathrm{cmt}_0 \leftarrow \mathsf{C}_{\mathsf{mac}}.\mathsf{Commit}(\mathsf{mck}, (0,0); \rho)$, forwarding other elements in mpp and ipp of $\mathsf{M}'_{\mathsf{alg}}$ to $\mathcal{A}$, Tag is simulated by delegating to its own Tag oracle, and $\mathcal{A}$'s forgery is $\mathcal{B}_2$'s forgery to its UFRMA game. Therefore, we have the bound as in Theorem 3.1. □