Unforgeability of Blind Schnorr in the Limited Concurrency Setting

Franklin Harding[1] and Jiayu Xu[2]

[1] Brown University, USA
[2] Oregon State University, USA

Abstract. Blind signature schemes enable a user to obtain a digital signature on a message from a signer without revealing the message itself. Among the most fundamental examples of such a scheme is *blind Schnorr*, but recent results show that it does not satisfy the standard notion of security against malicious users, *One-More Unforgeability* (OMUF), as it is vulnerable to the ROS attack. However, blind Schnorr does satisfy the weaker notion of *sequential OMUF*, in which only one signing session is open at a time, in the Algebraic Group Model (AGM) + Random Oracle Model (ROM), assuming the hardness of the Discrete Logarithm (DL) problem.

This paper serves as a first step towards characterizing the security of blind Schnorr in the *limited concurrency setting.* Specifically, we show that blind Schnorr satisfies OMUF when at most two signing sessions can be concurrently open (in the AGM+ROM, assuming DL). Our argument suggests that it is plausible that blind Schnorr satisfies OMUF for up to polylogarithmically many concurrent signing sessions. Our security proof involves interesting techniques from linear algebra and combinatorics.

Keywords: Schnorr signatures · blind signatures · algebraic group model · ROS

1 Introduction

Envisioning an untraceable electronic payment system, David Chaum introduced the concept of a *Blind Signature Scheme* (BSS): a two-party protocol which allows a *user* to obtain a digital signature on a message from a *signer* without revealing that message to the signer or knowing the signer's secret key [Cha82]. Chaum explained the idea through the analogy of sealing a document in a carbon-paper envelope before handing it to a signer. Blind signatures have myriad applications including digital cash [DH22], anonymous credentials [BL13], voting protocols [FOO92], and blockchain coin swaps [Nic19].

The typical security notions for a blind signature scheme are *One-More Unforgeability* (OMUF), which is security against a malicious user, and *Blindness*, which is security against a malicious signer. In this work, we focus on OMUF. Roughly speaking, we say that a BSS satisfies ℓ-OMUF if no adversary can produce $\ell + 1$ distinct valid message-signature pairs after conducting up to ℓ signing sessions with the signer.

Introduced by Chaum and Pedersen in 1992, the Schnorr Blind Signature Scheme (SBSS) is simple, efficient, and one of the most well-studied blind signature schemes [CP93]. In his 2001 security analysis of the scheme, Schnorr introduced the "ROS" problem, and proved that if ROS is hard then the SBSS is secure in the Random Oracle Model (ROM) + Generic Group Model (GGM) [Sch01]. He also proved a sort of converse: if ROS is easy, then the SBSS is not secure. On the intractability of ROS itself, Schnorr called it a

E-mail: fharding1@protonmail.com (Franklin Harding), xujiay@oregonstate.edu (Jiayu Xu)

Date of this document: 2024-09-10.

"plausible but novel complexity assumption," and showed that it is statistically hard for small enough parameters.

The GGM is a strong idealized model which is commonly used to justify hardness assumptions; for instance, in the GGM it is possible to prove that any generic DL algorithm runs in time $\Omega(p^{1/2})$ where p is the largest prime divisor of the group order [Sho97]. It is sometimes problematic to apply the GGM on a "scheme level," because there exist real-world schemes which have proofs of security in the GGM but are completely broken when instantiated with concrete groups (as in the real-world) [NS01, SPMS02]. This was the impetus for researchers to investigate whether the SBSS is provably secure without the GGM. Baldimtsi and Lysyanskaya gave a negative answer for ROM-only proofs, showing that "current [as of 2013] techniques for proving security in the random oracle model do not work for the Schnorr blind signature" [BL13]. The open question that remained was whether the SBSS is provably secure in the ROM plus some other assumptions weaker than the GGM.

The Algebraic Group Model (AGM) is a relatively new idealized model which lies between the standard model and the GGM [FKL18]. Consequentially, a proof of SBSS security in the AGM would be a better security guarantee than the existing GGM proof due to Schnorr. In 2020, Fuchsbauer, Plouviez, and Seurin accomplished this: they showed that the SBSS is secure in the ROM+AGM, assuming that ROS and One-More Discrete Logarithm (OMDL) are hard [FPS20].[1]

Until recently, the fastest algorithm for ROS was the subexponential (but still super-polynomial) time algorithm due to Wagner [Wag02]. However, in 2020, Benhamouda et al. unveiled a polynomial-time algorithm for ROS [BLL$^+$22]. This invalidated all existing security proofs for the SBSS. Moreover, in light of Schnorr's result that the SBSS is insecure if ROS is easy, the scheme was rendered completely broken. Indeed, Benhamouda et al. explicitly showed how their algorithm can be used to break OMUF of the SBSS and even provided a Python implementation of their attack. The [BLL$^+$22] algorithm only applies when the dimension of the ROS problem, η, is no smaller than the security parameter λ. In fact, if $\eta = O((\log \lambda)^k)$ for some $k \in \mathbb{N}$, then ROS is statistically hard [FPS20].

While the SBSS is completely broken according to the standard security notion for blind signature schemes, it remains desirable for use in practice due to its efficiency, simplicity, and compatibility with the standardized signature scheme EdDSA.[2] Developers have already had a chance to experiment with the scheme [Nic19], and there are existing implementations. Therefore, it is natural to ask:

Does the SBSS satisfy a less stringent, yet still practical, notion of security?

Motivated by this question, Kastner, Loss, and Xu proved that the SBSS is still secure in the *sequential setting*, where only one signing session can be open at a time [KLX22]. This setting, in contrast to the standard concurrent setting where there is no limitation on the number of signing sessions which can be open at the same time, has also been considered by other authors for various schemes [JLO97, BL13].

The proof of SBSS security in the sequential setting by Kastner, Loss, and Xu essentially relies on the fact that 1-ROS is statistically hard. Since η-ROS is statistically hard when $\eta = \eta(\lambda)$ is eventually bounded by some polylogarithmic function, it seems plausible that the SBSS is secure when the adversary is only allowed to open a small number of signing sessions concurrently. More formally:

Is the SBSS secure when at most η signing sessions can be open concurrently and $\eta = 2$? What about $\eta = O(1)$ or $\eta = O((\log \lambda)^k)$ for some $k \in \mathbb{N}$?

[1] ROM+AGM+ROS+OMDL is still weaker than ROM+GGM+ROS since one can prove that OMDL is hard in the GGM [BFP21].

[2] https://csrc.nist.gov/pubs/fips/186-5/final

Blind Signatures: The State of the Art. Shortly after the ROS attack was published, a number of new blind signature schemes were introduced. A variant of the SBSS called *Clause Blind Schnorr* is concurrently secure under a novel *modified ROS* problem [FPS20]. *Snowblind* still relies upon the AGM (and DL), but is concurrently-secure and basically as efficient as blind Schnorr [CKM+23]. Two new schemes which do not rely on any "non-standard" assumptions (such as the AGM or OMDL) are Chairattana-Apirom et al.'s BS_3 (CDH+ROM) [CATZ24], and Kastner et al.'s round-optimal scheme (sRSA+DDH+ROM) [KNR24]. Lastly, Fuchsbauer and Wolf recently unveiled a method for concurrently secure blind Schnorr that uses NIZK and PKE primitives [FW24].

With these innovative new schemes in mind, one might be tempted to dismiss the SBSS as obsolete. However, it remains important in practice due to the recent standardization of EdDSA by NIST, which is only a slight variation on the Schnorr signature scheme. Of particular importance is compatibility with Ed25519, which is more widely adopted than Ed448. Considering compatibility with existing standards, only two of the aforementioned schemes are at all relevant: Clause Blind Schnorr and Fuchsbauer-Wolf. The former appears incompatible with Ed25519 because it only achieves roughly 70 bits of security when instantiated with a 256-bit curve. The latter requires a pairing-friendly curve (thus incompatible), or otherwise is inefficient.

In summary, of pairing-free blind signatures from prime-order groups, no scheme is all three: efficient, compatible with existing standards, and concurrently secure. The SBSS is the only scheme that satisfies the first two, so it remains relevant for applications where sequential signing is sufficient. Enforcing sequential signing on multiple threads on multiple servers may prove practically difficult, possibly requiring the use of a distributed mutex.

Finally, the SBSS remains one of the most thoroughly studied and conceptually elegant blind signature schemes to date; studying its exact level of security is of theoretical interest in its own right, and could shed some light on security proofs involving other BSS' in the future.

Our Contributions. We prove that in the ROM+AGM, if OMDL is hard then the SBSS is secure when at most 2 signing sessions can be open concurrently (henceforth the 2-concurrent setting). To the best of our knowledge, this is the first result on the security of the SBSS in the "limited concurrency" setting; as such, our work serves as the first step towards a complete characterization of SBSS security, namely closing the gap between the security of the SBSS in the sequential setting [KLX22] and insecurity under the standard definition [BLL+22]. While it is clear from the ROS attack that users targeting 128-bit security should stick with sequential signing, our result has practical relevance: namely, users have some "wiggle room." That is, if some failure in a distributed mutex or some race condition allows two signatures to be signed concurrently, the scheme still attains a reasonable level of security.

Since the proof of SBSS security in the sequential setting essentially relies on the statistical hardness of 1-ROS, one could imagine that a straightforward adaptation of that proof would yield security in the 2-concurrent setting based on the fact that 2-ROS is hard, and perhaps even the η-concurrent setting where η is eventually bounded by some polylogarithmic function. It turns out that this is far from true; even the 2-concurrent setting is exponentially more complex than the sequential setting, and wrangling this complexity requires new techniques that do not appear in the sequential setting proof. To illustrate this point:

- Suppose that there are three signing sessions. In the sequential setting there is only one possibility for the "structure" of how the sessions overlap. In the 2-concurrent setting, there are 11 possibilities.[3] In general, the number of possibilities grows exponentially with the number of signing sessions. As we cannot consider all of them, we must make a combinatorial argument which “reduces” all of them to a few essential cases.

- Further complicating things is that signing sessions are not necessarily closed. In the sequential setting it is clear that the signing session which is never closed has to be the last, which simplifies the proof considerably. In the 2-concurrent setting there might be two unclosed sessions, and one of them might overlap with previous sessions.

The roadmap for our proof of ℓ-OMUF (where ℓ is the number of signing sessions) in the 2-concurrent setting relative to the Schnorr blind signature scheme follows:

1. We prove that any adversary $\mathcal{A}'$ for ℓ-OMUF in the 2-concurrent setting can be turned into an adversary $\mathcal{A}$ for $(\ell+2)$-OMUF in the 2-concurrent setting which *closes all of its signing sessions* and wins at least as often as $\mathcal{A}'$.
2. We prove that $(\ell+2)$-OMDL reduces to $(\ell+2)$-OMUF in the 2-concurrent setting if the adversary closes all of its signing sessions. The reduction constructs $\ell+1$ linear equations $\chi_1, \dots, \chi_{\ell+1}$ and wins if at least one of them is not zero.
3. To upper-bound the probability that the “bad event” $\chi_1 = \cdots = \chi_{i+1} = 0$ occurs, we first identify certain “special queries” of the SBSS random oracle. We then divide the possible configurations of signing sessions and special queries into three cases: (1) there are 2 special queries, both of which are made during the same single signing session (recall that a special query can be made during up to 2 signing sessions); (2) there are 3 special queries made during the same 2 signing sessions; or (3) neither of the prior two cases occur.
4. We show that if case (1) occurs then we can use the adversary to solve 1-ROS, which is essentially the same as [KLX22] except with an explicit reduction rather than a statistical argument. Somewhat of an extension, we show that if case (2) occurs then we can use the adversary to solve 2-ROS. The essentially different case is (3), in which we make a direct argument that the bad event occurs with negligible probability.

The direct argument for case (3) of item 4 is specific to the 2-concurrent setting, although all remaining parts of our approach easily generalize. We suspect that the direct argument can also be made to work for larger values than 2, although that would require more sophisticated reasoning using linear algebra and combinatorics (we explain at the end of this paper why our current technique does not directly apply to the more general setting).

2 Preliminaries

Let λ be the security parameter. For $n \in \mathbb{N}$, we use the notation $[n] := \{1, 2, \dots, n\}$. We write $a \leftarrow\$ S$ to denote sampling a value a uniformly at random from an efficiently samplable set S, and we write $a \leftarrow \mathcal{A}$ to denote assigning a to be the result of some process $\mathcal{A}$. We assume that setting and fetching entries from a hash table takes time $O(1)$. We call $\mathsf{GenGroup}$ a *group generation algorithm* if $(G, p, \mathbf{g}) \leftarrow \mathsf{GenGroup}(1^\lambda)$ is such that $G = \langle \mathbf{g} \rangle$ is a group of prime order p and $p > 2^{\lambda-1}$. When we work with a group G we assume that its elements can be efficiently encoded as bitstrings, and we do not make the

[3] Label the sessions 1, 2, and 3. Two must be non-overlapping; suppose up to relabeling that they are 1 and 2. This induces five periods: before session 1, during session 1, between sessions 1 and 2, during session 2, and after session 2. Session 3 must begin in one of the periods and end in a period no earlier than it began, resulting in $\sum_{i=1}^{5} = 15$ possibilities. Four of them are identical up to relabeling.

```
Game $\mathsf{OMDL}^{\mathcal{A}}_{\mathsf{GenGroup},\ell}(\lambda)$
    $(G, p, \mathbf{g}) \leftarrow \mathsf{GenGroup}(1^\lambda)$
    $q \leftarrow 0$
    $(x_i)_{i=1}^{\ell+1} \leftarrow\!\!\$\ \mathbb{Z}_p^{\ell+1}$
    $(x'_i)_{i=1}^{\ell+1} \leftarrow \mathcal{A}^{\mathsf{DL}}(G, p, \mathbf{g}, \mathbf{g}^{x_1}, \ldots, \mathbf{g}^{x_{\ell+1}})$
    return $(\forall i \in [\ell+1] : x_i = x'_i) \wedge q \leq \ell$

Oracle $\mathsf{DL}(\mathbf{x})$
    $q \leftarrow q + 1$
    return $\log_{\mathbf{g}}(\mathbf{x})$
```

Figure 1: The ℓ-One-More Discrete Logarithm Game

distinction between a group element and its binary encoding. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is *negligible* if for any positive integer c there exists some integer n_c such that for all $n \geq n_c$, we have $|\mathsf{negl}(n)| \leq 1/n^c$. A problem P with parameters par is *hard* if for all probabilistic polynomial-time adversaries $\mathcal{A}$, the *advantage* of $\mathcal{A}$ in problem P relative to parameters par, denoted $\mathsf{Adv}^{P}_{\mathsf{par},\mathcal{A}}(\lambda)$, is negligible.

2.1 The Algebraic Group Model

The Algebraic Group Model (AGM) is an idealized model in which all adversaries are presumed *algebraic* relative to some group [FKL18]. An adversary is called algebraic if, whenever it outputs a group element, it also outputs an explanation of how that element can be computed in terms of the group elements the adversary has received. Formally,

Definition 1 (Algebraic Adversary)**.** Let $(G, p, \mathbf{g})$ be such that G is a group of order p generated by $\mathbf{g}$. Let $\mathcal{A}$ be an adversary who has received group elements $\mathbf{h}_1, \ldots, \mathbf{h}_n \in G$ in that order. We say that $\mathcal{A}$ is algebraic relative to $(G, p, \mathbf{g})$ if whenever $\mathcal{A}$ outputs a group element $\mathbf{x} \in G$, they also output $e_1, \ldots, e_n \in \mathbb{Z}_p$ such that

$$\mathbf{x} = \prod_{i=1}^{n} \mathbf{h}_i^{e_i}.$$

We use the notation $\mathbf{x}_{(e_1,\ldots,e_n)} \leftarrow \mathcal{A}(1^\lambda)$. If $\mathcal{A}$ has access to an oracle, then we consider any group elements output by that oracle to be part of the group elements received by $\mathcal{A}$, and any group elements $\mathcal{A}$ queries to the oracle as part of $\mathcal{A}$'s output.

2.2 One-More Discrete Logarithm

Let ℓ be a positive integer and $\mathsf{GenGroup}$ a group generation algorithm. Given $(G, p, \mathbf{g}) \leftarrow \mathsf{GenGroup}(1^\lambda)$, group elements $\mathbf{h}_1, \ldots, \mathbf{h}_{\ell+1} \in G$ called "challenges," and access to a discrete log oracle[4] that can be queried at most ℓ times, the *ℓ-One-More Discrete Logarithm* (ℓ-OMDL) problem relative to $\mathsf{GenGroup}$ is to compute $x_1, \ldots, x_{\ell+1} \in \mathbb{Z}_p$ such that $\mathbf{h}_i = \mathbf{g}^{x_i}$ for all $i \in [\ell+1]$. We describe a formal security game for ℓ-OMDL in fig. 1.

Definition 2 (ℓ-One-More Discrete Logarithm)**.** Let $\ell \in \mathbb{N}$. We define the ℓ-OMDL advantage of an adversary $\mathcal{A}$ relative to $\mathsf{GenGroup}$ as

$$\mathsf{Adv}^{\mathsf{OMDL}}_{\mathsf{GenGroup},\ell,\mathcal{A}}(\lambda) := \Pr[\mathsf{OMDL}^{\mathcal{A}}_{\mathsf{GenGroup},\ell}(1^\lambda) = 1].$$

[4] Note that the discrete log oracle can be queried on any group elements rather than exclusively challenges, and that this is a needless distinction in the AGM. An algebraic adversary would need to "explain" how it computed the element which it is querying to the discrete log oracle in terms of the group elements it has seen, i.e., the challenges and the generator. Since the discrete logs of these elements are known, it would be easy to compute the discrete log of any group element the adversary could come up with.

```
Game $\mathsf{ROS}^{\mathcal{A}}_{\mathsf{GenGroup},\eta}(1^\lambda)$
$(G, p, \mathbf{g}) \leftarrow \mathsf{GenGroup}(1^\lambda)$
$((\vec{\rho}_i, \mathsf{aux}_i)_{i=1}^{\eta+1}, \vec{c}) \leftarrow \mathcal{A}^{\mathsf{H}_{\mathsf{ROS}}}(G, p, \mathbf{g})$   // note $\vec{\rho}_i, \vec{c} \in \mathbb{Z}_p^\eta$
return $\forall i, j \in [\eta + 1] : i \neq j \implies (\vec{\rho}_i, \mathsf{aux}_i) \neq (\vec{\rho}_j, \mathsf{aux}_j)$
    $\wedge\, \forall i \in [\eta + 1] : \vec{\rho}_i \cdot \vec{c} = \mathsf{H}_{\mathsf{ROS}}(\vec{\rho}_i, \mathsf{aux}_i)$
```

Figure 2: The ROS Game

2.3 ROS

Let η be a positive integer. Given a prime $p > 2^{\lambda-1}$ and random oracle $\mathsf{H}_{\mathsf{ROS}} : \mathbb{Z}_p^\eta \to \mathbb{Z}_p$, the ***R****andom Inhomogeneities in an* ***O****verdetermined* ***S****olvable System of $\eta + 1$ Linear Equations* (η-ROS) problem [Sch01] is to find distinct $\vec{\rho}_1, \ldots, \vec{\rho}_{\eta+1} \in \mathbb{Z}_p^\eta$ and $\vec{c} \in \mathbb{Z}_p^\eta$ such that

$$\vec{\rho}_i \cdot \vec{c} = \mathsf{H}_{\mathsf{ROS}}(\vec{\rho}_i) \text{ for all } i \in [\eta + 1].$$

We define a formal security game in fig. 2. As in [FPS20], the addition of auxillary information aux_i corresponding to each $\vec{\rho}_i$ is for convenience and is equivalent to the standard ROS formulation, as is the use of $\mathsf{GenGroup}$.

Definition 3 (η-ROS)**.** Let $\eta \in \mathbb{N}$. We define the η-ROS advantage of an adversary $\mathcal{A}$ relative to $\mathsf{GenGroup}$ as

$$\mathsf{Adv}^{\mathsf{ROS}}_{\mathsf{GenGroup},\eta,\mathcal{A}}(\lambda) := \Pr[\mathsf{ROS}^{\mathcal{A}}_{\mathsf{GenGroup},\eta}(1^\lambda) = 1].$$

The η-ROS problem is statistically hard for small values of η such as $\eta = O(1)$ or $\eta = O((\log \lambda)^k)$ for some $k \in \mathbb{N}$ [FPS20, Lemma 2]. We recall this result in lemma 1.

Lemma 1 ([FPS20])**.** *Let $\eta \in \mathbb{N}$ and $\mathsf{GenGroup}$ be a group generation algorithm. If $\mathcal{A}$ is an adversary which makes at most q_h queries to $\mathsf{H}_{\mathsf{ROS}}$, then*

$$\mathsf{Adv}^{\mathsf{ROS}}_{\mathsf{GenGroup},\eta,\mathcal{A}}(\lambda) \leq \frac{\binom{q_h}{\eta+1} + 1}{2^{\lambda-1}}.$$

2.4 Blind Signature Schemes

We restrict our definition of a BSS to “three-move” or “sigma” variants, of which the Schnorr blind signature scheme is an example.

Definition 4 (Blind Signature Scheme)**.** A (three-move) blind signature scheme (BSS) is a tuple of algorithms $\mathsf{BS} = (\mathsf{ParGen}, \mathsf{KeyGen}, \mathsf{Sign}_1, \mathsf{Sign}_2, \mathsf{User}_1, \mathsf{User}_2, \mathsf{Verify})$. An honest execution of the signing protocol is illustrated in fig. 3.

We use the convention that after $\mathsf{pp} \leftarrow \mathsf{ParGen}(1^\lambda)$ and $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$ are run, it is implicit that pp is known to all parties.

Definition 5 (Correctness)**.** We say that a blind signature scheme BS satisfies correctness

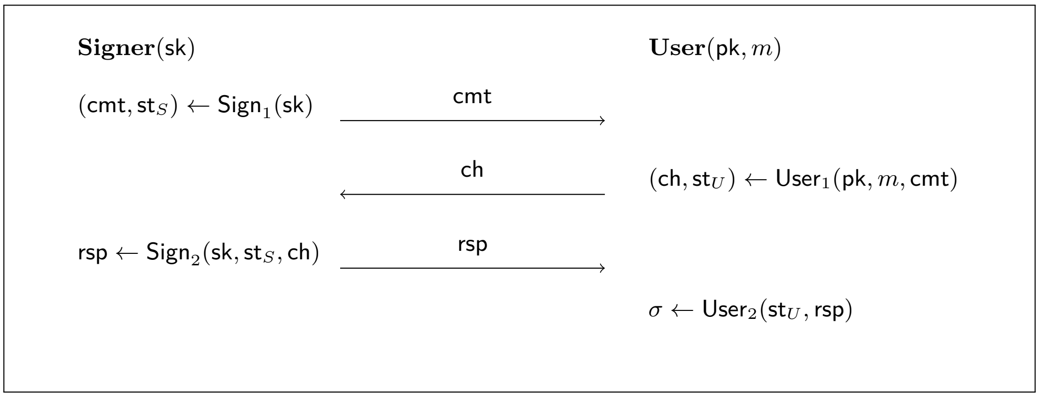
Signer(sk)
User(pk, m)
$(\mathsf{cmt}, \mathsf{st}_S) \leftarrow \mathsf{Sign}_1(\mathsf{sk})$
cmt
ch
$(\mathsf{ch}, \mathsf{st}_U) \leftarrow \mathsf{User}_1(\mathsf{pk}, m, \mathsf{cmt})$
$\mathsf{rsp} \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \mathsf{st}_S, \mathsf{ch})$
rsp
$\sigma \leftarrow \mathsf{User}_2(\mathsf{st}_U, \mathsf{rsp})$

Figure 3: BSS Signing Session

if for all $m \in \{0,1\}^*$,

$$\Pr\left[\mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1 \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{ParGen}(1^\lambda) \\ (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}) \\ (\mathsf{cmt}, \mathsf{st}_S) \leftarrow \mathsf{Sign}_1(\mathsf{sk}) \\ (\mathsf{ch}, \mathsf{st}_U) \leftarrow \mathsf{User}_1(\mathsf{pk}, m, \mathsf{cmt}) \\ \mathsf{rsp} \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \mathsf{st}_S, \mathsf{ch}) \\ \sigma \leftarrow \mathsf{User}_2(\mathsf{pk}, \mathsf{st}_U, \mathsf{rsp}) \end{array}\right] = 1.$$

2.4.1 One-More Unforgeability

The standard notion of security against malicious users for a blind signature scheme is *ℓ-One-More Unforgeability* (ℓ-OMUF). The weakened *ℓ-Sequential One-More Unforgeability* (ℓ-SEQ-OMUF) notion restricts adversaries to opening no more than one signing session at a time [KLX22]. We define a more general security property which parameterizes the maximum number of signing sessions which may be open concurrently, thus encompassing both ℓ-OMUF and ℓ-SEQ-OMUF. We say that a BSS satisfies (ℓ, η)-OMUF if it satisfies ℓ-OMUF and no more than η signing sessions were ever open concurrently. As two special cases, ℓ-OMUF = (ℓ, ∞)-OMUF, and ℓ-SEQ-OMUF = $(\ell, 1)$-OMUF. To accommodate for this change, the OMUF security game in fig. 4 is identical to the definition in [FPS20] except that it additionally checks the max number of signing sessions which were open concurrently at any point in time during the game. The game uses the following variables:

- k_1, the number of signing sessions which have been opened;
- k_2, the number of signing sessions which have been closed and should not surpass ℓ;
- $\mathcal{S}$, the set of indices of currently open signing sessions; and,
- η^*, the maximum value that $|\mathcal{S}|$ ever takes on, i.e., the max number of signing sessions open concurrently at any point in time during the game (so it should not surpass η). Note that when a session is closed, $|\mathcal{S}|$ decrements but η^* remains unchanged.

When the game ends, it outputs a bit b which is 1 if the adversary outputs $\ell + 1$ distinct valid message-signature pairs and never opened more than η signing sessions concurrently.

Definition 6 ((ℓ, η)-One-More Unforgeability)**.** Let $\ell, \eta \in \mathbb{N}$. We define the (ℓ, η)-OMUF advantage of an adversary $\mathcal{A}$ relative to BS as

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BS},\ell,\eta,\mathcal{A}}(\lambda) := \Pr[\mathsf{OMUF}^{\mathcal{A}}_{\mathsf{BS},\ell,\eta}(1^\lambda) = 1].$$

```
Game $\mathsf{OMUF}^{\mathcal{A}}_{\mathsf{BS},\ell,\eta}(1^\lambda)$
$\mathsf{pp} \leftarrow \mathsf{ParGen}(1^\lambda)$
$(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
$k_1 \leftarrow 0; k_2 \leftarrow 0; \mathcal{S} \leftarrow \emptyset$
$\eta^* \leftarrow 0$
$(m_i^*,\sigma_i^*)_{i=1}^{\ell+1} \leftarrow \mathcal{A}^{S_1,S_2}(\mathsf{pk})$
return $k_2 \leq \ell \wedge \eta^* \leq \eta \wedge$
    $\wedge\, \forall i,j \in [\ell+1] : i \neq j \implies (m_i^*,\sigma_i^*) \neq (m_j^*,\sigma_j^*)$
    $\wedge\, \forall i \in [\ell+1] : \mathsf{Verify}(\mathsf{pk},m_i^*,\sigma_i^*) = 1$
```

```
Oracle $S_1()$
$k_1 \leftarrow k_1 + 1$
$(\mathsf{cmt},\mathsf{st}_{k_1}) \leftarrow \mathsf{Sign}_1(\mathsf{sk})$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{k_1\}$
$\eta^* \leftarrow \max(\eta^*,|\mathcal{S}|)$
return $(k_1,\mathsf{cmt})$
```

```
Oracle $S_2(j,\mathsf{ch})$
if $j \notin S$ then return $\perp$
$\mathsf{rsp} \leftarrow \mathsf{Sign}_2(\mathsf{sk},\mathsf{st}_j,\mathsf{ch})$
$\mathcal{S} \leftarrow \mathcal{S} \setminus \{j\}; k_2 \leftarrow k_2 + 1$
return $\mathsf{rsp}$
```

Figure 4: The One-More-Unforgeability Game

An existing result [FPS20, Lemma 3] says that (translated into our notation) any adversary for (ℓ,∞)-OMUF opening at most q_s signing sessions can be turned into an adversary for (q_s,∞)-OMUF *which closes all of its signing sessions.*[5] In (ℓ,η)-OMUF, a winning adversary closes at most ℓ signing sessions and never opens more than η signing sessions concurrently. Consequently, a winning (ℓ,η)-OMUF adversary never opens more than $\ell+\eta$ signing sessions. From this observation, it follows from the proof of [FPS20, Lemma 3] that a winning adversary for (ℓ,η)-OMUF can be turned into a winning adversary for $(\ell+\eta,\ell+\eta)$-OMUF which closes all of its signing sessions. We now introduce a lemma which is slightly stronger than this result: any winning adversary for (ℓ,η)-OMUF can be turned into a winning adversary for $(\ell+\eta,\eta)$-OMUF which closes all of its signing sessions.

Lemma 2. *Let $\ell,\eta \in \mathbb{N}$ and* BS *be a blind signature scheme satisfying correctness. Let $\mathcal{A}$ be an adversary for (ℓ,η)-*OMUF *running in time τ and outputting messages of length at most* mlen*. Then there exists an adversary $\mathcal{B}$ for $(\ell+\eta,\eta)$-*OMUF *making exactly $\ell+\eta$ queries to S_1 and exactly $\ell+\eta$ valid queries (i.e., whose output is not $\perp$) to S_2 such that*

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BS},\ell+\eta,\eta,\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BS},\ell,\eta,\mathcal{A}}(\lambda),$$

and $\mathcal{B}$ runs in time $\tau + O(\mathsf{mlen}^2)$.

Proof. Let $\mathcal{B}$ be the adversary described in fig. 5. In fig. 5, diag is standard bitwise diagonalization after padding.

Claim. *$\mathcal{B}$ makes exactly $\ell+\eta$ queries to S_1 and exactly $\ell+\eta$ valid queries to S_2.*

When $\mathcal{A}$ halts, $k_1' \leq \ell+\eta$ is the number of queries that have been made to S_1 and $k_2' \leq \ell$ is the number of valid queries that have been made to S_2. Note that $|\mathcal{S}| = k_1' - k_2'$. Considering the total number of queries to S_1, we count

$$k_1' + (\eta - |\mathcal{S}|) + (\ell - k_1' + |\mathcal{S}|) = \ell + \eta. \tag{1}$$

If $j \in \mathcal{S}$ after $\mathcal{A}$ halts then $\mathcal{A}$ never made a valid $S_2(j,\cdot)$ query. Hence, $\mathcal{B}$'s S_2 queries after $\mathcal{A}$ halts are guaranteed to be valid by correctness of BS. Therefore, the total number of valid S_2 queries is

$$k_2' + |\mathcal{S}| + (\eta - |\mathcal{S}|) + (\ell - k_1' + |\mathcal{S}|) = \ell + \eta - (k_1' + k_2') + |\mathcal{S}| = \ell + \eta. \tag{2}$$

[5] Their lemma is written for the SBSS, but its proof essentially works for any correct three-move BSS.

```
$\mathcal{B}^{S_1,S_2}(\mathsf{pk})$
$\mathcal{S} \leftarrow \emptyset$;    $\mathcal{S} \leftarrow (\,)$
$k_1' \leftarrow 0$;    $k_2' \leftarrow 0$;    $\eta' \leftarrow 0$
$(m_i^*, \sigma_i^*)_{i=1}^{\ell+1} \leftarrow \mathcal{A}^{S_1',S_2'}(\mathsf{pk})$
$\vec{s} \leftarrow (m_i^*, \sigma_i^*)_{i=1}^{\ell+1}$
$m \leftarrow \mathrm{diag}(m_1^*, \ldots, m_{\ell+1}^*)$
// use $\mathcal{A}$'s unfinished sessions to create more valid signatures
for $j$ in $\mathcal{S}$    // $|\mathcal{S}| = k_1' - k_2'$
    $(\mathsf{ch}, \mathsf{st}_U) \leftarrow \mathsf{User}_1(\mathsf{pk}, m\|0\|0^j, \mathcal{S}(j))$
    $\mathsf{rsp} \leftarrow S_2(j, \mathsf{ch})$
    $\sigma \leftarrow \mathsf{User}_2(\mathsf{pk}, \mathsf{st}_U, \mathsf{rsp})$
    $\vec{s} \leftarrow \vec{s}\|(m\|0\|0^j, \sigma)$
// to guarantee $(\ell+1) + |\mathcal{S}| + (\eta - |\mathcal{S}|) = \ell + \eta + 1$ signatures
for $i$ in $[\eta - |\mathcal{S}|]$
    $(j, \mathsf{cmt}) \leftarrow S_1()$
    $(\mathsf{ch}, \mathsf{st}_U) \leftarrow \mathsf{User}_1(\mathsf{pk}, m\|1\|0^i, \mathsf{cmt})$
    $\mathsf{rsp} \leftarrow S_2(j, \mathsf{ch})$
    $\sigma \leftarrow \mathsf{User}_2(\mathsf{pk}, \mathsf{st}_U, \mathsf{rsp})$
    $\vec{s} \leftarrow \vec{s}\|(m\|1\|0^i, \sigma)$
// to guarantee exactly $\ell + \eta$ $S_1$ and $S_2$ queries
for $i$ in $[\ell - k_1' + |\mathcal{S}|]$
    $(j, \mathsf{cmt}) \leftarrow S_1()$
    $(\mathsf{ch}, \mathsf{st}_U) \leftarrow \mathsf{User}_1(\mathsf{pk}, \epsilon, \mathsf{cmt})$
    $\mathsf{rsp} \leftarrow S_2(j, \mathsf{ch})$
return $\vec{s}$
```

```
Oracle $S_1'()$
if $k_1' \geq \ell + \eta$ then
    return $\perp$
$(j, \mathsf{cmt}) \leftarrow S_1()$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{j\}$
$\eta' \leftarrow \max(\eta', |\mathcal{S}|)$
$\mathcal{S}(j) \leftarrow \mathsf{cmt}$
$k_1' \leftarrow k_1' + 1$
return $(j, \mathsf{cmt})$
```

```
Oracle $S_2'(j, \mathsf{ch})$
if $k_2' \geq \ell \wedge j \in \mathcal{S}$ then
    return $\perp$
$\mathsf{rsp} \leftarrow S_2(j, \mathsf{ch})$
if $\mathsf{rsp} \neq \perp$ then
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{j\}$
    $k_2' \leftarrow k_2' + 1$
return $\mathsf{rsp}$
```

Figure 5: Adversary $\mathcal{B}$ for $(\ell + \eta, \eta)$-OMUF which closes every signing session

Claim. *If $\eta' \leq \eta$ when $\mathcal{A}$ halts, then $\mathcal{B}$ never opens more than η signing sessions concurrently.*

Throughout the entire game $|\mathcal{S}| \leq \eta'$, so if $\eta' \leq \eta$ when $\mathcal{A}$ halts then up to that point in the game $|\mathcal{S}|$ has never exceeded η. Therefore, $\mathcal{B}$ has not opened more than η signing sessions concurrently. After $\mathcal{A}$ halts $\mathcal{B}$ closes all open signing sessions and then only opens and closes signing sessions sequentially, so the number of maximum concurrently open signing sessions is not affected.

Claim. *Considering $\mathcal{A}$'s output $(m_i^*, \sigma_i^*)_{i=1}^{\ell+1}$ and η' after $\mathcal{A}$ halts,*

$$\Pr\left[\begin{array}{c} \forall i \in [\ell+1] : \mathsf{Verify}(\mathsf{pk}, m_i^*, \sigma_i^*) = 1 \\ \forall i, j \in [\ell+1] : i \neq j \implies (m_i^*, \sigma_i^*) \neq (m_j^*, \sigma_j^*) \\ \eta' \leq \eta \end{array}\right] \geq \mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BS},\ell,\eta,\mathcal{A}}(\lambda).$$

The left-hand side is exactly $\mathcal{A}$'s winning condition ($k_2' \leq \ell$ is implied), so it suffices to show that $\mathcal{B}$ perfectly simulates the (ℓ, η)-OMUF game to $\mathcal{A}$ assuming $\mathcal{A}$ wins. While simulating (ℓ, η)-OMUF, adversary $\mathcal{B}$ does nothing more than forward $\mathcal{A}$'s queries to S_1' and S_2' to its own (respective) S_1 and S_2 oracles — except that $\mathcal{B}$ might return $\perp$ if $k_1' \geq \ell + \eta$ on a query to S_1' or if $k_2' \geq \ell$ on a query to S_2'. But a winning adversary cannot open more than $\ell + \eta$ sessions or close more than ℓ sessions, so $\mathcal{B}$'s behavior in these cases does not matter.

Claim.

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{BS},\ell+\eta,\eta,\mathcal{B}}(\lambda) \geq \Pr\left[\begin{array}{c} \forall i \in [\ell+1] : \mathsf{Verify}(\mathsf{pk}, m_i^*, \sigma_i^*) = 1 \\ \forall i, j \in [\ell+1] : i \neq j \implies (m_i^*, \sigma_i^*) \neq (m_j^*, \sigma_j^*) \\ \eta' \leq \eta \end{array}\right]$$

Assuming the conditions on the right-hand side, we show that $\mathcal{B}$ wins by checking its winning conditions one by one:

- *Adversary $\mathcal{B}$'s output message-signature pairs are valid:* After $\mathcal{A}$ halts, each $j \in \mathcal{S}$ is such that $(j, \mathcal{S}(j))$ was the output of some S_1 query but no valid $S_2(j, \mathsf{ch})$ query was ever made for some ch. Hence, by correctness of BS, each $(m\|0\|0^j, \sigma)$ pair that $\mathcal{B}$ adds to $\vec{s}$ is such that $\mathsf{Verify}(\mathsf{pk}, m\|0\|0^j, \sigma) = 1$, as are the $(m\|1\|0^i, \sigma)$ pairs.

- *Adversary $\mathcal{B}$'s output message-signature pairs are distinct:* As m is distinct from $m_1^*, \ldots, m_{\ell+1}^*$, the $(\ell+2)$-th through $(\ell+\eta+1)$-th elements of $\vec{s}$, which are of the form

$$(m\|0\|0^1, \cdot), \cdots, (m\|0\|0^{|\mathcal{S}|}, \cdot), (m\|1\|0^1, \cdot), \cdots, (m\|1\|0^{\eta-|\mathcal{S}|}, \cdot),$$

  are all distinct from each other and distinct from the first $(\ell+1)$-th elements of $\vec{s}$. The first ℓ elements of $\vec{s}$ are distinct by assumption.

- *Adversary $\mathcal{B}$ never opens more than η signing sessions concurrently:* $\eta' \leq \eta$ implies by a previous claim that $\mathcal{B}$ never opens more than η signing sessions concurrently.

- *Adversary $\mathcal{B}$ does not make more than $\ell + \eta$ valid queries to S_2:* previously shown.

□

```
$\mathsf{ParGen}(1^\lambda)$
    $(G, p, \mathbf{g}) \leftarrow \mathsf{GenGroup}(1^\lambda)$
    return $(G, p, \mathbf{g})$

$\mathsf{Sign}_1(x)$
    $r \leftarrow\!\!\$\ \mathbb{Z}_p$
    $\mathbf{r} \leftarrow \mathbf{g}^r$
    return $(\mathbf{r}, r)$

$\mathsf{Sign}_2(x, r, c)$
    $s \leftarrow r + cx$
    return $s$

$\mathsf{Verify}(\mathbf{x}, m, \sigma = (\mathbf{r}', s'))$
    $c' \leftarrow \mathsf{H}(\mathbf{r}', m)$
    return $\mathbf{g}^{s'} = \mathbf{r}' \cdot \mathbf{x}^{c'}$

$\mathsf{KeyGen}(\mathsf{pp})$
    $x \leftarrow\!\!\$\ \mathbb{Z}_p$
    $\mathbf{x} \leftarrow \mathbf{g}^x$
    return $(x, \mathbf{x})$

$\mathsf{User}_1(\mathbf{x}, m, \mathbf{r})$
    $\alpha, \beta \leftarrow\!\!\$\ \mathbb{Z}_p$
    $\mathbf{r}' \leftarrow \mathbf{r} \cdot \mathbf{g}^\alpha \cdot \mathbf{x}^\beta$
    $c' \leftarrow \mathsf{H}(\mathbf{r}', m)$
    $c \leftarrow c' + \beta$
    $\mathsf{st}_U \leftarrow (\mathbf{r}, \alpha, \beta, c)$
    return $(c, \mathsf{st}_U)$

$\mathsf{User}_2(\mathbf{x}, \mathsf{st}_U, s)$
    $(\mathbf{r}, \alpha, \beta, c) \leftarrow \mathsf{st}_U$
    if $\mathbf{g}^s \neq \mathbf{r} \cdot \mathbf{x}^c$ then
        return $\perp$
    $\mathbf{r}' \leftarrow \mathbf{r} \cdot \mathbf{g}^\alpha \cdot \mathbf{x}^\beta$
    $s' \leftarrow s + \alpha$
    $\sigma \leftarrow (\mathbf{r}', s')$
    return $\sigma$
```

$$\mathsf{SBS}_{\mathsf{GenGroup}} = (\mathsf{ParGen}, \mathsf{KeyGen}, \mathsf{Sign}_1, \mathsf{Sign}_2, \mathsf{User}_1, \mathsf{User}_2, \mathsf{Verify})$$

Figure 6: The Schnorr Blind Signature Scheme [CP93]

2.5 Schnorr Blind Signature Scheme

Let $\mathsf{GenGroup}$ be a group generation algorithm and $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_p$ a random oracle. The Schnorr Blind Signature Scheme (SBSS) is presented in fig. 6.

Theorem 1 ([CP93]). $\mathsf{SBS}_{\mathsf{GenGroup}}$ *satisfies correctness.*

Corollary 1. *Lemma 2 holds if* $\mathsf{BS} = \mathsf{SBS}_{\mathsf{GenGroup}}$ *and* $\mathcal{A}$ *is algebraic relative to* $\mathsf{GenGroup}$*. Furthermore,* $\mathcal{B}$ *can be constructed such that it is algebraic relative to* $\mathsf{GenGroup}$.[6]

Proof. The SBSS satisfies correctness per theorem 1, so lemma 2 applies. To make $\mathcal{B}$ algebraic:

- For the first $\ell + 1$ elements of $\vec{s}$, output the algebraic representation output by $\mathcal{A}$.
- The $(\ell + 2)$-th through $(\ell + \eta + 1)$-th elements of $\vec{s}$, as well as any queries to H, result from honest execution of the $\mathsf{SBS}_{\mathsf{GenGroup}}$ algorithms User_1 and User_2. These algorithms are algebraic, so it is possible to output algebraic representations accordingly. □

3 Unforgeability in the 2-Concurrent Setting

In this section, we show that the SBSS is $(\ell', 2)$-OMUF in the ROM+AGM, assuming the hardness of ℓ-OMDL where $\ell := \ell' + 2$. We first explain the high-level idea behind the proof.

Reduction From OMDL. Per corollary 1, we can assume that $\mathcal{A}$ plays the $(\ell, 2)$-OMUF game and closes all its signing sessions. Let $\mathbf{h}_1, \ldots, \mathbf{h}_{\ell+1}$ with discrete log oracle DL be an ℓ-OMDL instance. Our goal is to compute $x_1, \ldots, x_{\ell+1}$ such that $\mathbf{g}^{x_i} = \mathbf{h}_i$ for all $i \in [\ell + 1]$, and we can only query DL at most ℓ times. To accomplish this, we'll exploit $\mathcal{A}$'s ability to generate $\ell + 1$ valid signatures after conducting only ℓ signing sessions.

We simulate the OMUF game to $\mathcal{A}$ using $\mathbf{x} = \mathbf{h}_{\ell+1}$ as the public key and $\mathbf{h}_1, \ldots, \mathbf{h}_\ell$ as answers to S_1 queries. To simulate valid $S_2(j, c_j)$ queries, we return

$$s_j := \mathsf{DL}(\mathbf{h}_j\, \mathbf{x}^{c_j}). \tag{3}$$

Since $\mathcal{A}$ makes ℓ valid queries to S_2, we queried DL only ℓ times. Assuming $\mathcal{A}$ wins, their output $(m_i^*, \sigma_i^* = (\mathbf{r}_i^*, s_i^*))_{i=1}^{\ell+1}$ is such that every message-signature pair is valid, meaning

$$\mathbf{g}^{s_i^*} = \mathbf{r}_i^*\, \mathbf{x}^{c_i^*} \tag{4}$$

where $c_i^* = H(\mathbf{h}_i^*, m_i^*)$. Since $\mathcal{A}$ is algebraic it provides an algebraic representation of each $\mathbf{r}_i^*$ in terms of the group elements it received, i.e., $(\gamma_i^*, \xi_i^*, \rho_{i,1}^*, \ldots, \rho_{i,\ell}^*)$ such that

$$\mathbf{r}_i^* = \mathbf{g}^{\gamma_i^*}\, \mathbf{x}^{\xi_i^*} \prod_{j=1}^{\ell} \mathbf{h}_j^{\rho_{i,j}^*}. \tag{5}$$

With some calculations, eqs. (3) to (5) yield

$$\mathbf{g}^{s_i^* - \gamma_i^* - \sum_{j=1}^{\ell} \rho_{i,j}^* s_j} = \mathbf{x}^{c_i^* + \xi_i^* - \sum_{j=1}^{\ell} \rho_{i,j}^* c_j}.$$

If $\chi_i := c_i^* + \xi_i^* - \sum_{j=1}^{\ell} \rho_{i,j}^* c_j$ is not zero for some $i \in [\ell + 1]$, we can compute the discrete log $x \in \mathbb{Z}_p$ of $\mathbf{x} = \mathbf{h}_{\ell+1}$. Also, we can "recover" the discrete log of $\mathbf{h}_i$ for all $i \in [\ell]$ since eq. (3) implies $\mathbf{h}_i = \mathbf{g}^{s_i - c_i x}$. Altogether, we obtain a solution to the ℓ-OMDL instance. Up to this point, the argument has been completely analogous to [FPS20].

[6] Adversary $\mathcal{B}$ is not explicitly constructed so that it is algebraic in [FPS20, Lemma 3].

Bad Event: No Invertible χ_i. The elephant in the room is the assumption that $\chi_1 = \cdots = \chi_{\ell+1} = 0$ doesn't happen. Showing that this "bad event" happens with negligible probability is the crux of the entire proof and where our argument differs from [FPS20] and [KLX22]. In [FPS20], an upper bound is accomplished via a reduction from ℓ-ROS. However, as shown in [BLL+22], ℓ-ROS is actually easy if $\ell \geq \lambda$, meaning that this argument only works if the adversary opens fewer than λ signing sessions. [KLX22] gives a *statistical* upper-bound for the probability of this "bad event," which they denote E, in the sequential setting. Their argument, in broad strokes, is by the following steps:

1. If $\mathcal{A}$ does not query $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ for some $i \in [\ell+1]$ then $\chi_i = 0$ occurs with probability $1/p$. Now assume that $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ for all $i \in [\ell+1]$; call this the i-th *special query* and assume without loss of generality that $\mathcal{A}$ never repeats an H query.

2. If $\mathcal{A}$ makes a special query when no signing sessions are open, event E occurs with probability at most $1/p$.[7]

3. Assuming that neither of the first two cases happen, every special query is made when a signing session is open. As there are $\ell+1$ of these queries and only ℓ signing sessions, there is some signing session in which two of these queries, say the i-th and $(i+1)$-th, are made. If $\chi_i = \chi_{i+1} = 0$ then they effectively constitute a 1-ROS solution. Indeed, one can explicitly reduce from 1-ROS to E occurring in this case.

The first two cases also appear in the limited concurrency setting and their respective arguments can essentially be reused. However, it is surprisingly non-trivial to make an argument analogous to the third case in even just the 2-concurrent setting, as signing sessions can overlap and be interleaved in a manner decided by the adversary, generating a large number of possibilities that we must consider. For intuition, we discuss two representative examples of how an adversary might behave in this setting.

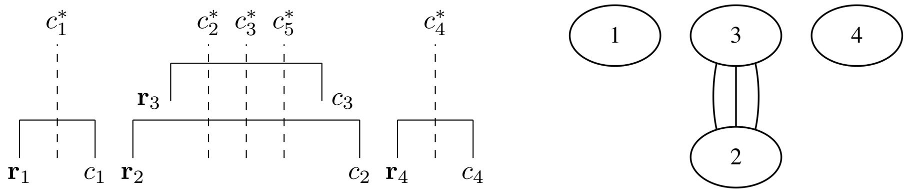

Figure 7: Example $(4,2)$-OMUF game. Left: visualized as a timeline. Right: visualized as a graph where distinct signing sessions are connected by special queries.

First Example. Our first example is illustrated in fig. 7. Each "staple" represents a signing session; each $\mathbf{r}_j$, which is the output of an S_1 query, marks the start of a signing session; each c_j, which is the input of an S_2 query, marks the end of a signing session; and each dashed line, marked by c_i^*, represents the i-th special query by the adversary, of which c_i^* is the output. The bad event occurs if $\chi_i = 0$ for all $i \in [5]$, which corresponds to eq. (6).

$$\begin{cases} c_1^* + \xi_1^* - \rho_{1,1}^* c_1 & = 0 \\ c_2^* + \xi_2^* - \rho_{2,1}^* c_1 - \rho_{2,2}^* c_2 - \rho_{2,3}^* c_3 & = 0 \\ c_3^* + \xi_3^* - \rho_{3,1}^* c_1 - \rho_{3,2}^* c_2 - \rho_{3,3}^* c_3 & = 0 \\ c_5^* + \xi_5^* - \rho_{5,1}^* c_1 - \rho_{5,2}^* c_2 - \rho_{5,3}^* c_3 & = 0 \\ c_4^* + \xi_4^* - \rho_{4,1}^* c_1 - \rho_{4,2}^* c_2 - \rho_{4,3}^* c_3 - \rho_{4,4}^* c_4 & = 0 \end{cases} \tag{6}$$

[7] This is called "C_1" in [KLX22] and the loose upper-bound there is q_h/p where q_h is the number of H queries made by the adversary.

Obtaining a 4-ROS solution is straightforward: when $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$, return $c_i^* := \mathsf{H}_{\mathsf{ROS}}(\rho_i^*, (\gamma_i^*, \xi_i^*, m_i^*)) - \xi_i^*$ (this is what is done in [FPS20]). However, this is not good enough: if $\ell \geq \lambda$, then we can only reduce from ℓ-ROS, which is easy. In our setting, where $\mathcal{A}$ can only open at most 2 concurrent signing sessions, we show how to obtain a 2-ROS solution. When $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}_1^*, m_1^*)$, no sessions have been closed, and we answer with $\mathsf{H}_{\mathsf{ROS}}((\rho_{1,1}^*\ 0), (\gamma_1^*, \xi_1^*, \rho_1^*, m_1^*)) - \xi_1^*$ — similar to how we solve 4-ROS above. However, when $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}_2^*, m_2^*)$, session 1 has been closed and c_1 (together with its algebraic coefficient $\rho_{1,1}^*$) has been determined, so we can incorporate that item into the query answer. That is, we answer the query with $\mathsf{H}_{\mathsf{ROS}}((\rho_{2,1}^*\ 0), (\gamma_2^*, \xi_2^*, \rho_2^*, m_2^*)) - \xi_2^* + \rho_{2,1}^* c_1$. Oracle queries $\mathsf{H}(\mathbf{r}_3^*, m_3^*)$ and $\mathsf{H}(\mathbf{r}_5^*, m_5^*)$ are answered similarly, and when the $\mathsf{H}(\mathbf{r}_4^*, m_4^*)$ query is made, sessions 1, 2 and 3 are all closed, so the answer is $\mathsf{H}_{\mathsf{ROS}}((\rho_{4,4}^*\ 0), (\gamma_4^*, \xi_4^*, \rho_4^*, m_4^*)) - \xi_4^* + \rho_{4,1}^* c_1 + \rho_{4,2}^* c_2 + \rho_{4,3}^* c_3$. In this way we eliminate all c_j where session j has been closed when the special query is made, making sure that every equation has at most 2 outstanding c_j items. Answering H queries in this manner, eq. (6) becomes

$$\begin{cases} \mathsf{H}_{\mathsf{ROS}}((\rho_{1,1}^*\ 0), (\gamma_1^*, \xi_1^*, \rho_1^*, m_1^*)) - \rho_{1,1}^* c_1 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{2,2}^*\ \rho_{2,3}^*), (\gamma_2^*, \xi_2^*, \rho_2^*, m_2^*)) - \rho_{2,2}^* c_2 - \rho_{2,3}^* c_3 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{3,2}^*\ \rho_{3,3}^*), (\gamma_3^*, \xi_3^*, \rho_3^*, m_3^*)) - \rho_{3,2}^* c_2 - \rho_{3,3}^* c_3 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{5,2}^*\ \rho_{5,3}^*), (\gamma_5^*, \xi_5^*, \rho_5^*, m_5^*)) - \rho_{5,2}^* c_2 - \rho_{5,3}^* c_3 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{4,4}^*\ 0), (\gamma_4^*, \xi_4^*, \rho_4^*, m_4^*)) - \rho_{4,4}^* c_4 & = 0 \end{cases} \tag{7}$$

We obtain a 2-ROS solution as $(\rho_{i,2}^*\ \rho_{i,3}^*) \cdot (c_2\ c_3) = \mathsf{H}_{\mathsf{ROS}}((\rho_{i,2}^*, \rho_{i,3}^*), (\gamma_i^*, \xi_i^*, \rho_i^*, m_i^*))$ for all $i \in \{2, 3, 5\}$. It could have also been the case that c_5^* was queried in the first signing session, in which case we could try to obtain a 1-ROS solution with some minor changes to how we answer H queries. In either case there is some "nice" structure to the adversary's behavior which we are exploiting. However, this structure is not inherent.

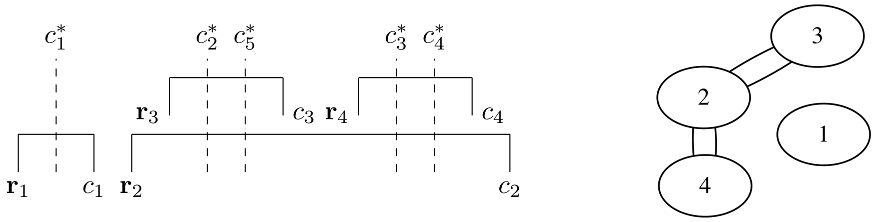

Figure 8: "Problematic" (4, 2)-OMUF game. Left: visualized as a timeline. Right: visualized as a graph where distinct signing sessions are connected by special queries.

Second Example. Considering fig. 8, even if we answer H queries as previously described, $\chi_i = 0$ for all $i \in [5]$ only boils down to eq. (8).

$$\begin{cases} \mathsf{H}_{\mathsf{ROS}}((\rho_{1,1}^*\ 0), (\gamma_1^*, \xi_1^*, \rho_1^*, m_1^*)) - \rho_{1,1}^* c_1 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{2,2}^*\ \rho_{2,3}^*), (\gamma_2^*, \xi_2^*, \rho_2^*, m_2^*)) - \rho_{2,2}^* c_2 - \rho_{2,3}^* c_3 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{5,2}^*\ \rho_{5,3}^*), (\gamma_5^*, \xi_5^*, \rho_5^*, m_5^*)) - \rho_{5,2}^* c_2 - \rho_{5,3}^* c_3 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{3,2}^*\ \rho_{3,4}^*), (\gamma_3^*, \xi_3^*, \rho_3^*, m_3^*)) - \rho_{3,2}^* c_2 - \rho_{3,4}^* c_4 & = 0 \\ \mathsf{H}_{\mathsf{ROS}}((\rho_{4,2}^*\ \rho_{4,4}^*), (\gamma_4^*, \xi_4^*, \rho_4^*, m_4^*)) - \rho_{4,2}^* c_2 - \rho_{4,4}^* c_4 & = 0 \end{cases} \tag{8}$$

No 1-ROS or 2-ROS solution is to be found, but we can make a more direct argument. The key point is to isolate $\mathsf{H}(\mathbf{r}_2^*, m_2^*)$ and $\mathsf{H}(\mathbf{r}_5^*, m_5^*)$ queries, both of which are made during sessions 2 and 3; and $\mathsf{H}(\mathbf{r}_3^*, m_3^*)$ and $\mathsf{H}(\mathbf{r}_4^*, m_4^*)$ queries, both of which are made during sessions 2 and 4. Looking at eq. (8), from $\chi_2 = \chi_5 = 0$ we obtain

$$\underbrace{\begin{pmatrix} \rho^*_{2,2} & \rho^*_{2,3} \\ \rho^*_{5,2} & \rho^*_{5,3} \end{pmatrix}}_{L} \begin{pmatrix} c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} \mathsf{H}_{\mathsf{ROS}}((\rho^*_{2,2}\ \rho^*_{2,3}), (\gamma^*_2, \xi^*_2, \rho^*_2, m^*_2)) \\ \mathsf{H}_{\mathsf{ROS}}((\rho^*_{5,2}\ \rho^*_{5,3}), (\gamma^*_5, \xi^*_5, \rho^*_5, m^*_5)) \end{pmatrix} \tag{9}$$

and from $\chi_3 = \chi_4 = 0$ we obtain

$$\underbrace{\begin{pmatrix} \rho^*_{3,2} & \rho^*_{3,4} \\ \rho^*_{4,2} & \rho^*_{4,4} \end{pmatrix}}_{R} \begin{pmatrix} c_2 \\ c_4 \end{pmatrix} = \begin{pmatrix} \mathsf{H}_{\mathsf{ROS}}((\rho^*_{3,2}\ \rho^*_{3,4}), (\gamma^*_3, \xi^*_3, \rho^*_3, m^*_3)) \\ \mathsf{H}_{\mathsf{ROS}}((\rho^*_{4,2}\ \rho^*_{4,4}), (\gamma^*_4, \xi^*_4, \rho^*_4, m^*_4)) \end{pmatrix} \tag{10}$$

Assuming that both L and R are full-rank, eq. (9) has a unique solution (c_2, c_3) uniform in $\mathbb{Z}_p^2$, and eq. (10) has a unique solution (c'_2, c_4) uniform in $\mathbb{Z}_p^2$. As $c_2 = c'_2$ occurs with probability $1/p$, the bad event that $\chi_i = 0$ for all $i \in [5]$ occurs with probability at most $1/p$, conditioned on c^*_3 and c^*_4 being the specific H queries picked by $\mathcal{A}$ as special queries. Adversary $\mathcal{A}$ could make many (not exceeding q_h) queries when sessions 2 and 4 are open and for each pair check whether the solution in terms of those special queries satisfies $c_2 = c'_2$. By the union bound, $\chi_i = 0$ for all $i \in [5]$ occurs with probability $\binom{q_h}{2}/p$.

In the formal proof we will also argue that if L (resp. R) is not full-rank, then $\chi_2 = \chi_5 = 0$ (resp. $\chi_2 = \chi_4 = 0$) occurs with probability at most $1/p$.

Summary: Handling the Bad Event $\chi_1 = \cdots = \chi_{i+1} = 0$. Summarizing the two examples above,

1. In fig. 7, there are two signing sessions (2 and 3) and three special queries (whose results are c^*_2, c^*_3, c^*_5) such that all three queries are made during the two sessions. In general, this case can be reduced from 2-ROS.[8] Similarly, if there is one signing session and two special queries such that both queries are made during the session (and no other sessions), then it can be reduced from 1-ROS (this argument has already been made in [KLX22]).

2. The case in fig. 8 is less intuitive to generalize, but the point is as follows: signing sessions 2, 3 and 4 form a grouping that is disjoint from the other signing session, and there are four special queries $(c^*_2, c^*_3, c^*_4, c^*_5)$ that occur within this group. These four special queries can be partitioned into two disjoint sets: c^*_2 and c^*_5 involve signing sessions 2 and 3, and c^*_3 and c^*_4 involve signing sessions 2 and 4. Each set corresponds to a linear system with two unknowns and two equations. The key is that there is a single session (session 2) that "bridges" these two sets of special queries, which means that there is a single unknown (c_2) that appears in both linear systems, and the probability that there is a value for c_2 that satisfies both linear systems is $1/p$, conditioned on the choice of c^*_3 and c^*_4.
   In general we first isolate a signing session grouping G' where there are more special queries than signing sessions, and then partition all special queries that occur within G' into two sets, U and V, such that (1) all signing sessions in U come before all signing sessions in V, (2) each set involves at least as many special queries as signing sessions, and (3) there is a single signing session that is involved in both sets of special queries. In this case we can make a direct argument that $\chi_i = 0$ for all $i \in [\ell + 1]$ occurs with probability at most $\binom{q_h}{2}/p$.

The rest of the proof is to show that essentially one of the two cases above must occur for a $(\ell, 2)$-OMUF adversary. Roughly, this is because of the following: for any two concurrently open signing sessions in the same grouping G', there is a special query "connecting" them;

[8] In fact it suffices even if some of the three queries are made during *one of* the two sessions.

this "uses up" $|G'| - 1$ special queries, but more than $|G'|$ special queries are made during G', so there are at least two remaining special queries; let them be the j-th and then the k-th. If queries j and k are made during the same pair of signing sessions then case 1 happens. Otherwise let U be all special queries before k, and V be all special queries afterwards; then there must be a signing session that "bridges" U and V, so case 2 happens.

Theorem 2. *Let* GenGroup *be a group generation algorithm and* $\ell' \in \mathbb{N}$. *Let* $\mathcal{A}'$ *be an algebraic adversary for the* $(\ell', 2)$-OMUF *game relative to* $\mathsf{SBS}_{\mathsf{GenGroup}}$ *making at most* q_h *queries to the random oracle* H *and outputting messages of length at most* m. *Define* $\ell := \ell' + 2$. *Then there exist adversaries* $\mathcal{B}_{\mathsf{OMDL}}$, $\mathcal{B}_1$, *and* $\mathcal{B}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{SBS},\ell',2,\mathcal{A}'}(\lambda) \leq \mathsf{Adv}^{\mathsf{OMDL}}_{\mathsf{GenGroup},\ell,\mathcal{B}_{\mathsf{OMDL}}}(\lambda) + \sum_{\eta'=1}^{2} \mathsf{Adv}^{\mathsf{ROS}}_{\mathsf{GenGroup},\eta',\mathcal{B}_{\eta'}}(\lambda) + \frac{q_h^2 + q_h + 10}{2p}.$$

Moreover, $\mathcal{B}_{\mathsf{OMDL}}$ *runs in time* $\tau + O(m^2 + q_h)$ *and* $\mathcal{B}_1, \mathcal{B}_2$ *make at most* q_h *queries to their respective* $\mathsf{H}_{\mathsf{ROS}}$ *oracles.*

The following corollary immediately follows from theorem 2 and lemma 1.

Corollary 2. *Let* GenGroup *be a group generation algorithm and* $\ell \in \mathbb{N}$. *If* $(\ell + 2)$*-OMDL is hard relative to* GenGroup*, then* $\mathsf{SBS}_{\mathsf{GenGroup}}$ *satisfies* $(\ell, 2)$-OMUF *in the AGM+ROM.*

Proof of theorem 2. By corollary 1, there exists an algebraic adversary $\mathcal{A}$ for $(\ell, 2)$-OMUF such that $\mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{SBS},\ell,2,\mathcal{A}}(\lambda) \geq \mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{SBS},\ell',2,\mathcal{A}'}(\lambda)$, and $\mathcal{A}$ makes exactly ℓ queries to S_1 and ℓ valid queries to S_2. We also assume without loss of generality that $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}, m)$ at most once for each pair $(\mathbf{r}, m)$.

Consider the sequence of games illustrated in fig. 9. As $\mathcal{A}$ makes exactly ℓ valid queries to S_2, we omit $k_2 \leq \ell$ as part of $\mathcal{A}$'s winning condition. We check if there exist distinct indices $i, j \in [\ell + 1]$ such that $(\mathbf{r}_i^*, m_i^*) = (\mathbf{r}_j^*, m_i^*)$ rather than $(m_i^*, \sigma_i^* = (\mathbf{r}_i^*, s_i^*)) = (m_j^*, \sigma_j^* = (\mathbf{r}_j^*, s_j^*))$, which is equivalent as s_i^* is completely determined by $\mathbf{r}_i^*$ and m_i^* for a valid signature. Additionally, we halt and output 0 immediately if $\eta^* > \eta$ after $\mathcal{A}$ halts.

We now outline the high-level idea behind each game hop:

Game_1. We keep track of the algebraic representations of each $\mathbf{r}$ in an $\mathsf{H}(\mathbf{r}, m)$ query. Once the adversary is finished, in addition to checking the previous winning conditions, we let the adversary lose if $\chi_1 = \cdots = \chi_{i+1} = 0$ and $S_i = \emptyset$ for some $i \in [\ell + 1]$. This rules out the "bad event" discussed in the outline of the proof, in the case that there was a special query made when *no* signing sessions were open; this corresponds to step 2 of the [KLX22] argument (see the summary of [KLX22] above).

Game_2. This game changes corresponds to the case where we can upper-bound the probability of the bad event $\chi_1 = \cdots = \chi_{i+1} = 0$ via reduction from 1-ROS ($|I| = 2$), or 2-ROS ($|I| = 3$). The former event corresponds to step 3 of the [KLX22] argument, while Figure 7 is a representative example of the latter event.

Game_3. This final game change corresponds to the only remaining case that the bad event $\chi_1 = \cdots = \chi_{i+1} = 0$ can occur: every special query was made during a signing session, but there is no way to reduce from 1-ROS or 2-ROS. Figure 8 is a representative example of this case.

Finally, we upper-bound the adversary's advantage in Game_3 via a reduction from the hardness of ℓ-OMDL.

Claim.

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) \geq \mathsf{Adv}^{\mathsf{OMUF}}_{\mathsf{SBS},\ell,2,\mathcal{A}}(\lambda) - \frac{1}{p}.$$

```
$\mathsf{OMUF}^{\mathcal{A}}_{\mathsf{SBS},\ell}(1^\lambda)$  $\mathsf{Game}_1$  $\mathsf{Game}_2$  $\mathsf{Game}_3$
$(G, p, \mathbf{g}) \leftarrow \mathsf{GenGroup}(1^\lambda)$
$(x, \mathbf{x}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
$k_1 \leftarrow 0; k_2 \leftarrow 0; \mathcal{S} \leftarrow \emptyset$
$\eta^* \leftarrow 0$
$S \leftarrow (\ ); T \leftarrow (\ ); U \leftarrow (\ )$
$(m_i^*, \sigma_i^* = (\mathbf{r}_i^*, s_i^*))_{i=1}^{\ell+1} \leftarrow \mathcal{A}^{S_1,S_2}(\mathbf{x})$
if $\eta^* > \eta$ then
  return $(0, \eta^*)$
if $\exists i, j \in [\ell+1] : i \neq j$
        $\wedge\, (\mathbf{r}_i^*, m_i^*) = (\mathbf{r}_j^*, m_j^*)$ then
  return $0$
$(S_i)_{i=1}^{\ell+1} \leftarrow (S(\mathbf{r}_i^*, m_i^*))_{i=1}^{\ell+1}$
$(c_i^*)_{i=1}^{\ell+1} \leftarrow (\mathsf{H}(\mathbf{r}_i^*, m_i^*))_{i=1}^{\ell+1}$
$(\gamma_i^*, \xi_i^*, \rho_i^*)_{i=1}^{\ell+1} \leftarrow (U(\mathbf{r}_i^*, m_i^*))_{i=1}^{\ell+1}$
$(\chi_i)_{i=1}^{\ell+1} = \left(c_i^* + \xi_i^* - \sum_{j=1}^{\ell} \rho_{i,j}^* c_j\right)_{i=1}^{\ell+1}$
if $\forall i \in [\ell+1] : \chi_i = 0$ then
  if $\exists i \in [\ell+1] : S_i = \emptyset$ then
    return $0$
  if $\exists I \subseteq [\ell+1] : |I| = 2$
        $\wedge\, |\cup_{i \in I} S_i| = 1$ then
    return $0$
  if $\exists I \subseteq [\ell+1] : |I| = 3$
        $\wedge\, |\cup_{i \in I} S_i| = 2$ then
    return $0$
  return $0$
return $(\forall i \in [\ell+1] : \mathsf{Verify}(\mathbf{x}, m_i^*, \sigma_i^*) = 1, \eta^*)$
```

```
Oracle $S_1()$
$k_1 \leftarrow k_1 + 1$
$(\mathbf{r}_{k_1}, r_{k_1}) \leftarrow \mathsf{Sign}_1(x)$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{k_1\}$
$\eta^* \leftarrow \max(\eta^*, |\mathcal{S}|)$
return $(k_1, \mathbf{r})$
```

```
Oracle $S_2(j, c)$
if $j \notin S$ then return $\perp$
$c_j \leftarrow c$
$s \leftarrow \mathsf{Sign}_2(x, r_j, c)$
$\mathcal{S} \leftarrow \mathcal{S} \setminus \{j\}; k_2 \leftarrow k_2 + 1$
return $s$
```

```
Oracle $\mathsf{H}(\mathbf{r}_{(\gamma,\xi,\rho)}, m)$
if $T(\mathbf{r}, m) = \perp$ then
  $S(\mathbf{r}, m) \leftarrow \mathcal{S}$
  $U(\mathbf{r}, m) \leftarrow (\gamma, \xi, \rho)$
  $T(\mathbf{r}, m) \leftarrow\!\!\$\ \mathbb{Z}_p$   // $(\diamond)$
return $T(\mathbf{r}, m)$
```

Figure 9: Sequence of games for $(\ell, 2)$-OMUF. In each successive game more code is added to the previous game, starting with $\mathsf{OMUF}^{\mathcal{A}}_{\mathsf{SBS},\ell}$. For example, Game_2 is all of the unboxed code plus all of the dashed box code plus all of the solid box white background code

In Game_1 we simulate queries to H via lazy sampling, which makes no external change. The other change is that Game_1 outputs 0 if $\chi_i = 0$ for all $i \in [\ell+1]$ and there is some $i \in [\ell+1]$ such that $S_i = S(\mathbf{r}_i^*, m_i^*) = \emptyset$. S_i is set to $\mathcal{S}$ when $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ is queried, at which point $\mathcal{S}$ is the set of indices of signing sessions that are open. Note that if $\mathcal{A}$ does not query $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$, then the challenger makes this query after $\mathcal{A}$ halts. Therefore, $S_i = \emptyset$ can occur as a result of exactly one of the following two scenarios:

1. The challenger queried $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ after $\mathcal{A}$ halted; or,

2. Adversary $\mathcal{A}$ queried $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ when no signing sessions were open.

The first case is easy: c_i^* is sampled uniformly at random from $\mathbb{Z}_p$ after all other terms in χ_i are fixed, so $\chi_i = 0$ occurs with probability $1/p$. For the second case, consider the moment when $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$. At this point, $\mathcal{A}$ has received group elements $\mathbf{x}$ and $\mathbf{r}_1, \ldots, \mathbf{r}_{k_1}$. As all signing sessions are closed, $c_1, \ldots, c_{k_1}$ are fixed. As $\mathcal{A}$ has not yet received group elements $\mathbf{r}_{k_1+1}, \ldots, \mathbf{r}_\ell$, we have $\rho_{i,j}^* = 0$ for all $j \in \{k_1+1, \ldots, \ell\}$. Putting this together,

$$\chi_i = c_i^* + \xi_i^* - \sum_{j=1}^{\ell} \rho_{i,j}^* c_j = c_i^* + \xi_i^* - \sum_{j \in [k_1]} \rho_{i,j}^* c_j. \tag{11}$$

Additionally, $\mathcal{A}$ provides ξ_i^* and $\rho_{i,1}^*, \ldots, \rho_{i,k_1}^*$ when they query $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$, at which point they are fixed.[9] As everything in eq. (11) is fixed before c_i^* is sampled uniformly at random from $\mathbb{Z}_p$, we conclude that $\chi_i = 0$ occurs with probability $1/p$.

Claim. *There exist adversaries $\mathcal{B}_1, \mathcal{B}_2$ for 1-ROS and 2-ROS respectively, making at most q_h queries to their respective $\mathsf{H}_{\mathsf{ROS}}$ oracles, such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(\lambda) \geq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(\lambda) - \sum_{\eta'=1}^{2} \mathsf{Adv}_{\mathsf{GenGroup},\eta',\mathcal{B}_{\eta'}}^{\mathsf{ROS}}(\lambda).$$

Adversary $\mathcal{A}$ has identical advantage in Game_2 unless $\chi_i = 0$ for all $i \in [\ell+1]$; there is no $i \in [\ell+1]$ such that $S_i = \emptyset$; and for some $\eta' \in \{1,2\}$ there exists $I \subseteq [\ell+1]$ of size $\eta'+1$ such that $\left|\bigcup_{i \in I} S_i\right| = \eta'$. If there is no $i \in [\ell+1]$ such that $S_i = \emptyset$ then for all $i \in [\ell+1]$ adversary $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$, and when they do so there is some signing session which is open. We call the query to $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ the i-th *special query*. With this terminology, we can restate the previous claim more intuitively: $\mathcal{A}$'s advantage is the same unless $\chi_i = 0$ for all $i \in [\ell+1]$; every special query occurs during a signing session; and either there are two special queries which occur during the same signing session, or there are three special queries and two signing sessions such that all three queries are made during one or two of these sessions (see fig. 7 for an example). We call the former event R_1, and the latter event R_2. We upper-bound $\Pr[R_1]$ via reduction from 1-ROS, and $\Pr[R_2]$ via reduction from 2-ROS. Let $\mathcal{B}_{\eta'}^{\mathsf{H}_{\mathsf{ROS}}}$ be the η'-ROS adversary which, on input $(G, p, \mathbf{g})$ where $G = \langle \mathbf{g} \rangle$ is a group of prime order $p > 2^\lambda$ and oracle $\mathsf{H}_{\mathsf{ROS}}$, does the following:

1. Simulate Game_2 to $\mathcal{A}$ with $(G, p, \mathbf{g})$ and instead of $T(\mathbf{r}, m) \leftarrow\!\!\$\ \mathbb{Z}_p$ on the line with comment $(\diamondsuit)$ in H, do the following:
   (a) Let $\mathcal{S} = \{j_1, \ldots, j_{k_1-k_2}\}$ where $j_1 < \cdots < j_{k_1-k_2}$. Also, define $\mathcal{G} := [k_1] \setminus \mathcal{S}$, that is, the set of indices of signing sessions that have been closed.

[9] This argument is also made in [FPS20, KLX22]. Adversary $\mathcal{A}$ provides algebraic representations of $\mathbf{r}_i^*$ when it queries $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ and also when it finally outputs $\mathbf{r}_i^*$. These representations might differ, so it is important that ξ_i^* and ρ_i^* are fixed on the H query in Game_1. This is done in [FPS20], but [KLX22] incorrectly takes the representation from when $\mathcal{A}$ outputs $\mathbf{r}_i^*$.

(b) Set $T(\mathbf{r}, m)$ to be

$$\mathsf{H}_{\mathsf{ROS}}((\rho_{j_1} \ \cdots \ \rho_{j_{k_1-k_2}})\|0^{k-(k_1-k_2)}, (\gamma, \xi, \rho, m)) - \xi + \sum_{j \in \mathcal{G}} \rho_j c_j.$$

2. Trying all combinations,[10] find $I \subseteq [\ell+1]$ of size $\eta'+1$ such that $\left|\bigcup_{i \in I} S_i\right| = \eta'$. If no such indices exist, then halt. Otherwise, let $\bigcup_{i \in I} S_i = \{j_1, \ldots, j_{\eta'}\}$ where $j_1 < j_2 < \cdots < j_{\eta'}$.

3. For each $i \in [\ell+1]$, set $\mathsf{aux}_i \leftarrow (\gamma_i^*, \xi_i^*, \rho_i^*, m_i^*)$.

4. Output $(((\rho^*_{i_q,j_1} \ \cdots \ \rho^*_{i_q,j_{\eta'}}), \mathsf{aux}_q)_{q=1}^{\eta'+1}, (c_{j_1} \ \cdots \ c_{j_{\eta'}}))$.

As $\mathbf{r}$ is completely determined by $\mathcal{A}$'s choice of its algebraic representation (γ, ξ, ρ), and $\mathsf{H}_{\mathsf{ROS}}$ outputs uniformly random elements in $\mathbb{Z}_p$, adversary $\mathcal{A}$'s view is identical in Game_2 and in $\mathcal{B}_{\eta'}$'s simulation. Hence, $R_{\eta'}$ occurs with the same probability in Game_2 and in $\mathcal{B}_{\eta'}$'s simulation of Game_2 to $\mathcal{A}$. If $R_{\eta'}$ occurs in $\mathcal{B}_{\eta'}$'s simulation, then $\mathcal{B}_{\eta'}$ does not halt on item 2. Note that when $\mathcal{A}$ queries $\mathsf{H}(\mathbf{r}_i^*, m_i^*)$ for any $i \in [\ell+1]$, $\mathcal{B}$ sets $\mathcal{G} \leftarrow [k_1] \setminus S_i$. As $\mathcal{A}$ has only received group elements $\mathbf{x}$ and $\mathbf{r}_1, \ldots, \mathbf{r}_{k_1}$ by this point, we have $\rho^*_{i,k_1+1} = \ldots = \rho^*_{i,\ell} = 0$. Therefore, at the end of the game once c_i is defined for all $i \in [\ell]$,

$$\sum_{j \in \mathcal{G}} \rho^*_{i,j} c_j = \sum_{j \in [k_1] \setminus S_i} \rho^*_{i,j} c_j + \sum_{j=k_1+1}^{\ell} 0 \cdot c_j = \sum_{j \in [\ell] \setminus S_i} \rho^*_{i,j} c_j.$$

We have $\chi_i = 0$ for all $i \in [\ell+1]$, which implies that for all $i \in I$,

$$\begin{aligned}
\chi_i &= c_i^* + \xi_i^* - \sum_{j=1}^{\ell} \rho^*_{i,j} c_j \\
&= \left(\mathsf{H}_{\mathsf{ROS}}((\rho_{i,j_1} \ \cdots \ \rho_{i,j_{\eta'}}), \mathsf{aux}_i) - \xi_i^* + \sum_{j \in [\ell] \setminus S_i} \rho^*_{i,j} c_j \right) + \xi_i^* - \sum_{j=1}^{\ell} \rho^*_{i,j} c_j \\
&= \mathsf{H}_{\mathsf{ROS}}((\rho_{i,j_1} \ \cdots \ \rho_{i,j_{\eta'}}), \mathsf{aux}_i) - \sum_{j \in S_i} \rho^*_{i,j} c_j \\
&= \mathsf{H}_{\mathsf{ROS}}((\rho_{i,j_1} \ \cdots \ \rho_{i,j_{\eta'}}), \mathsf{aux}_i) - \sum_{k \in [\eta']} \rho^*_{i,j_k} c_{j_k} = 0.
\end{aligned}$$

If $R_{\eta'}$ occurs then $(\mathbf{r}_1^*, m_i^*), \ldots, (\mathbf{r}_{\ell+1}^*, m_{\ell+1}^*)$ are all distinct (otherwise $\mathcal{A}$ would have halted before $R_{\eta'}$ could occur), so $\mathsf{aux}_1, \ldots, \mathsf{aux}_{\eta'+1}$ are all distinct. Therefore, $\mathcal{B}_{\eta'}$ wins the η'-ROS game. Moreover, $\mathcal{B}_{\eta'}^{\mathsf{H}_{\mathsf{ROS}}}$ makes q_h queries to $\mathsf{H}_{\mathsf{ROS}}$ as $\mathcal{A}$ makes q_h queries to H.

Claim.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_3}(\lambda) \geq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(\lambda) - \frac{q_h^2 + q_h + 8}{2p}.$$

Adversary $\mathcal{A}$ has identical advantage in Game_3 unless $\chi_i = 0$ for all $i \in [\ell+1]$; there is no $i \in [\ell+1]$ such that $S_i = \emptyset$; and for all $\eta' \in \{1, 2\}$, if $I \subseteq [\ell+1]$ is of size $\eta'+1$ then

$$\left| \bigcup_{i \in I} S_i \right| > \eta'.$$

[10] This takes time $O(\ell^{\eta'})$, but we don't actually care about the running time of $\mathcal{B}_{\eta'}$.

See fig. 8 for an example. For a set V of indices of signing sessions, define

$$Q_V := \{i \in [\ell+1] : S_i \cap V \neq \emptyset\}.$$

Recall that S_i is the set of signing sessions which are open when $\mathcal{A}$ makes the i-th special query. Set Q_V is basically the special queries which occur when at least one signing session in V is open. Consider the following graph:

$$G := ([\ell], \{\{a,b\} : a, b \in S_i \text{ for some } i \in [\ell+1] \wedge a \neq b\}).$$

G is visualized for concrete OMUF games in figs. 7 and 8. Vertices correspond to signing sessions and distinct vertices a, b are connected by an edge if there was a special query which was made when signing sessions a and b were both open. We refer to each connected component of G as a *signing session grouping.* As each special query, of which there are $\ell+1$, occurs during a signing session, of which there are ℓ, there exists a signing session grouping with vertex set G' such that

$$|Q_{G'}| \geq |G'| + 1, \tag{12}$$

that is, in which more special queries occur than there are signing sessions. Additionally, we have the requirements

$$\forall \text{ distinct } i, j \in [\ell+1] : |S_i \cup S_j| > 1 \tag{$\star$}$$
$$\forall \text{ distinct } i, j, k \in [\ell+1] : |S_i \cup S_j \cup S_k| > 2 \tag{$\star\star$}$$

We may assume that $|G'| \geq 3$, otherwise eqs. ($\star$) and ($\star\star$) cannot hold. Since G' is the set of vertices of a connected component of G, and each special query connects at most two vertices, there is a set $B \subseteq [\ell+1]$ of size $|G'| - 1$ such that edges $\{\{a,b\}, a, b \in S_i \text{ for some } i \in B\}$ span G'. In other words, if T is a tree that covers all vertices of G', then B is the set of special queries corresponding to the edges of T. As $B \subseteq Q_{G'}$, there are at least two remaining special queries in $Q_{G'} \setminus B$ per eq. (12); let them be j and k, and assume w.l.o.g. that the j-th special query occurs before the k-th.

Claim. *There exists some $v \in G'$ such that $v \in S_k$ and $v \notin S_j$. That is, there is some signing session v such that the k-th special query is made during signing session v but the j-th special query is not.*

As $j \in Q_{G'}$ and at most two signing sessions can be open when the j-th special query is made, we have $|S_j| = 1$ or $|S_j| = 2$.

Suppose $S_j = \{t, u\}$ for some $t, u \in G'$. Then signing sessions t and u are connected and at some point concurrently open. As they are at some point concurrently open and at most two signing sessions can ever be concurrently, the only way that B spans G' is if there is some $i \in B$ such that $S_i = \{t, u\}$ also. If $S_k \subseteq S_j$ then i, j, k are such that $|S_i \cup S_j \cup S_k| = 2$, which contradicts eq. ($\star\star$)

Next, suppose $S_j = \{t\}$ for some $t \in G'$. If $S_k = \{t\}$ then i, j contradict eq. ($\star$). If $S_k = \{t, u\}$ for some $u \in G'$, then there is some $i \in B$ such that $S_k = \{t, u\}$ also, and then i, j, k contradict eq. ($\star\star$).

Claim. *There exist $U, V \subseteq [\ell+1]$ which partition $Q_{G'}$ such that*

$$\left|\left(\bigcup_{i \in U} S_i\right) \cap \left(\bigcup_{i \in V} S_i\right)\right| = 1, \quad \left|\bigcup_{i \in U} S_i\right| \leq |U|, \quad \left|\bigcup_{i \in V} S_i\right| \leq |V|,$$

and all the special queries in U occur before all the special queries in V.

There is a lot of notation to unpack here,[11] so we restate the claim in plain English: we can partition all of the special queries which occur in the signing session grouping G' into two sets, U and V, such that:

- the signing sessions open when the special queries in U are made and the signing sessions open when the special queries in V are made have only one in common;
- the number of signing sessions open when the special queries in U are made is at most the number of special queries in U, and likewise for V; and,
- all the special queries in U occur before all the special queries in V.

As at least one and at most two signing sessions are open when the k-th special query is made, $|S_k| \in \{1, 2\}$.

Suppose that $S_k = \{v\}$ for some $v \in G'$. The idea is to let U start from j and V start from k (recall that j and k are the two special queries that are not in B, and j comes first); then we extend U to include all special queries in $Q_{G'}$ that occur before the k-th, and extend V to include all special queries in $Q_{G'}$ that occur afterwards. Formally, let

$$V = \{k\} \cup \{i' \in Q_{G'} : \text{ the } i'\text{-th special query occurs after the } k\text{-th}\},$$

and $U = Q_{G'} \setminus V$. Clearly U and V partition $Q_{G'}$ and are such that all of the special queries in U occur before all of the special queries in V. We now show that $\left(\bigcup_{i\in U} S_i\right) \cap \left(\bigcup_{i\in V} S_i\right) = \{v\}$. Since $S_k = \{v\}$, any special query after k is made during v or some signing session which is opened after k is queried, so it is not in $\bigcup_{i\in U} S_i$; similarly, any special query before k is made during v or some signing session which is closed before k is queried, so it is not in $\bigcup_{i\in V} S_i$. Essentially, query k acts as a "barrier" for signing sessions. Therefore, $\left(\bigcup_{i\in U} S_i\right) \cap \left(\bigcup_{i\in V} S_i\right) \subseteq \{v\}$. As $v \in S_k$ and $k \in V$ we know that $\bigcup_{i\in V} S_i$ contains v. Additionally, by the previous claim $v \notin S_j$, however G' is connected which means that at least one of the signing sessions which is open when j is queried is connected to signing session v. Therefore, there is some special query in $Q_{G'}$ which occurs before k and is made when v is open, so $\bigcup_{i\in U} S_i$ contains v. Thus $\left(\bigcup_{i\in U} S_i\right) \cap \left(\bigcup_{i\in V} S_i\right) = \{v\}$. Now considering $\bigcup_{i\in V} S_i$, we already know that this union contains v. For each additional signing session in G' which is opened after the k-th special query is made, there is an additional special query in V which connects it to the rest of the signing sessions in $\bigcup_{i\in V} S_i$. We conclude $|V| \geq |\bigcup_{i\in V} S_i|$. The argument for $|U| \geq |\bigcup_{i\in U} S_i|$ is analogous.

Now suppose that $S_k = \{u, v\}$ for some distinct $u, v \in G'$. If there exists $i \in S_{G'}$ such that $S_i = \{u, v\}$ also and the i-th special query occurs before the k-th then let $q = i$; otherwise set $q = k$. Let

$$V = \{q\} \cup \{i' \in Q_{G'} : \text{ the } i'\text{-th special query occurs after the } q\text{-th}\},$$

and $U = Q_{G'} \setminus V$. Clearly U and V partition $Q_{G'}$ and are such that all of the special queries in U occur before all of the special queries in V. In this case one can argue analogously that either $\left(\bigcup_{i\in U} S_i\right) \cap \left(\bigcup_{i\in V} S_i\right) = \{u\}$ or $\left(\bigcup_{i\in U} S_i\right) \cap \left(\bigcup_{i\in V} S_i\right) = \{v\}$ depending on which of u or v is opened first. Also, similarly to the previous argument, for each signing session in G' opened after q is made, there is an additional special query which connects it to the rest of the sessions in $\bigcup_{i\in V} S_i$.

Claim. $\Pr[\chi_i = 0 \text{ for all } i \in G'] \leq \frac{\binom{q_h}{2}+q_h+4}{p} = \frac{q_h^2+q_h+8}{2p}$.

[11] As an example, in fig. 8, $U = \{2, 5\}$ and $V = \{3, 4\}$: when the second and fifth special queries are made, signing sessions 2 and 3 are open; when the third and fourth special queries are made, signing sessions 2 and 4 are open; and there is one signing session in common, namely session 2.

Let $|U| = q$, and sort all special queries in U from first to last as $U = \{i_1, \dots, i_q\}$. Similarly, let $|V| = r$ and sort all special queries in V from first to last as $V = \{j_1, \dots, j_r\}$. Consider the algebraic coefficients $\rho^*_{i_1}, \dots, \rho^*_{i_q}, \rho^*_{j_1}, \dots, \rho^*_{j_r}$; we identify three cases which follow.

Suppose that $\rho^*_{i_1} = \vec{0}$. We have $\chi_{i_1} = c^*_{i_1} + \xi^*_{i_1}$ and $\xi^*_{i_1}$ is fixed before $c^*_{i_1}$ is sampled uniformly at random from $\mathbb{Z}_p$, so $\Pr[\chi_{i_1} = 0] = 1/p$. Similarly, if $\rho^*_{j_1} = \vec{0}$, then $\Pr[\chi_{j_1} = 0] = 1/p$.

Next, suppose that $\rho^*_{i_1} \neq \vec{0}$ and $\{\rho^*_{i_1}, \dots, \rho^*_{i_q}\}$ is linearly dependent. It follows that there is some $j \geq 2$ such that $\rho^*_{i_j}$ is a linear combination of $\rho^*_{i_1}, \dots, \rho^*_{i_{j-1}}$. Let $\lambda_1, \dots, \lambda_{j-1} \in \mathbb{Z}_p$ such that $\rho^*_{i_j} = \sum_{k=1}^{j-1} \lambda_k \rho^*_{i_k}$. Now we upper-bound $\Pr[\chi_{i_j} = 0 \mid \chi_{i_1} = \cdots = \chi_{i_{j-1}} = 0]$. Consider the moment when $\mathcal{A}$ makes the (i_j)-th query to H. At this point, $\rho^*_{i_j}$ is fixed and $\chi_{i_j} = 0$ holds if

$$\begin{aligned}
\chi_{i_j} &= c^*_{i_j} + \xi^*_{i_j} - \rho^*_{i_j} \cdot (c_1 \ \cdots \ c_\ell) \\
&= c^*_{i_j} + \xi^*_{i_j} - \left(\sum_{k=1}^{j-1} \lambda_k \rho^*_{i_k} \right) \cdot (c_1 \ \dots \ c_\ell) \\
&= c^*_{i_j} + \xi^*_{i_j} - \sum_{k=1}^{j-1} \lambda_k (c^*_{i_k} - \xi^*_{i_k}) = 0,
\end{aligned}$$

where the last equation is because $\chi_{i_k} = 0$. All terms in the expression are fixed before $c^*_{i_j}$ is sampled uniformly at random from $\mathbb{Z}_p$, hence this occurs with probability at most $1/p$. By an analogous argument, if $\{\rho^*_i\}_{i \in V}$ is linearly dependent then $\chi_i = 0$ for all $i \in [\ell]$ occurs with probability $1/p$.

Finally, suppose that both $\{\rho^*_i\}_{i \in U}$ and $\{\rho^*_i\}_{i \in V}$ are linearly independent. By the previous claim $|\bigcup_{i \in U} S_i| \leq |U|$ and if the inequality is strict then $\{\rho^*_i\}_{i \in U}$ cannot be linearly independent. Hence $|\bigcup_{i \in U} S_i| = |U| = q$ and by the same argument $|\bigcup_{i \in V} S_i| = |V| = r$. Also recall that all the special queries in U come before all the special queries in V.

Let $\bigcup_{i \in U} S_i = \{a_1, \dots, a_q\}$ and $\bigcup_{i \in V} = \{b_1, \dots, b_r\}$. The system $\{\chi_i = 0\}_{i \in U}$ corresponds to the matrix equation

$$\begin{pmatrix} \rho^*_{i_1} \\ \vdots \\ \rho^*_{i_q} \end{pmatrix} \begin{pmatrix} c_{a_1} \\ \vdots \\ c_{a_q} \end{pmatrix} = \begin{pmatrix} c^*_{i_1} + \xi^*_{i_1} \\ \vdots \\ c^*_{i_q} + \xi^*_{i_q} \end{pmatrix},$$

which has a unique solution $(c_{a_1}, \dots, c_{a_q})$ which is uniform in $\mathbb{Z}_p^q$. Now we consider the system $\{\chi_i = 0\}_{i \in V}$. Recall from the previous claim that our specific construction of V is such that the first special query in V occurs when signing session v is open, i.e., $v \in S_{j_1}$; where $\left(\bigcup_{i \in U} S_i\right) \cap \left(\bigcup_{i \in V} S_i\right) = \{v\}$. Consider the set S_{j_1}:

- If $S_{j_1} = \{v\}$, then $\chi_{j_1} = c^*_{j_1} + \xi^*_{j_1} - \rho^*_{j_1,v} c_v = 0$ implies $c_v = c^*_{j_1} (\rho^*_{j_1,v})^{-1}$. On the other hand, $v \in \bigcup_{i \in U} S_i$, so (as argued above) the system $\{\chi_i = 0\}_{i \in V}$ has a unique solution for c_v which is uniform in $\mathbb{Z}_p$. The probability that the same c_v works for both systems is $1/p$ conditioned on the specific H query that $\mathcal{A}$ picks for j_1, of which there are fewer than q_h possibilities.[12] By the union bound, $\Pr[\chi_{j_1} = 0] \leq q_h/p$.

- If $S_{j_1} = \{u, v\}$ for some $u \in G'$ then our construction was such that $S_{j_2} = \{u, v\}$. Similarly to before, $\chi_{j_1} = \chi_{j_2} = 0$ has a unique solution for c_v which is uniform in $\mathbb{Z}_p$. Considering the number of ways to choose special queries j_1 and j_2 from the total number of H queries, we obtain $\Pr[\chi_{j_1} = \chi_{j_2} = 0] \leq \binom{q_h}{2}/p$.

[12] Note that $\mathcal{A}$ can make a number of H queries, and after seeing the results check if any of the corresponding c_v works for both systems.

Claim. *There exists an adversary $\mathcal{B}_{\mathsf{OMDL}}$ such that*

$$\mathsf{Adv}^{\mathsf{OMDL}}_{\mathsf{GenGroup},\ell,\mathcal{B}_{\mathsf{OMDL}}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(\lambda)$$

and $\mathcal{B}_{\mathsf{OMDL}}$ runs in time $\tau + O(m^2 + q_h)$.

Let $\mathcal{B}_{\mathsf{OMDL}}$ be the adversary which, on OMDL instance $(G, p, \mathbf{g}, \mathbf{h}_1, \ldots, \mathbf{h}_{\ell+1})$ and discrete log oracle DL, simulates Game_3 to $\mathcal{A}$ with public parameters $(G, p, \mathbf{g})$, $\mathbf{x} = \mathbf{h}_{\ell+1}$, and the following changes:

- While answering $\mathcal{A}$'s S_1 queries, instead of doing $(\mathbf{r}_{k_1}, r_{k_1}) \leftarrow \mathsf{Sign}_1(x)$, which cannot be simulated without knowledge of the discrete logarithm of $\mathbf{x}$, set $\mathbf{r}_{k_1} \leftarrow \mathbf{h}_{k_1}$.
- While answering $\mathcal{A}$'s S_2 queries, instead of doing $s \leftarrow \mathsf{Sign}_2(x, r_j, c)$, which cannot be simulated without knowledge of r_j, the discrete logarithm of $\mathbf{r}_{k_1} = \mathbf{h}_{k_1}$, query
$$s_j \leftarrow \mathsf{DL}(\mathbf{h}_j\ \mathbf{x}^c) \tag{13}$$
and set $s \leftarrow s_j$.

As $\mathbf{h}_1, \ldots, \mathbf{h}_{\ell+1}$ are uniform in G, adversary $\mathcal{B}$'s simulation of S_1 is perfect. Additionally, s_j is uniform in $\mathbb{Z}_p$ and such that $\mathbf{g}^{s_j} = \mathbf{h}_j \mathbf{x}^{c_j}$ for all $j \in [\ell]$, so $\mathcal{B}$'s simulation of S_2 is also perfect.

After $\mathcal{A}$ halts, $s_1, \ldots, s_\ell$ are defined since $\mathcal{A}$ makes exactly ℓ valid queries to S_2. Then:

1. Find some $k \in [\ell+1]$ such that $\chi_k \neq 0$. If none exists, then halt.
2. Compute $x_{\ell+1} \leftarrow \chi_k^{-1}(s_k^* - \gamma_k^* - \sum_{j=1}^{\ell} \rho_{k,j}^* s_j)$.
3. Compute $x_i \leftarrow s_i - c_i x_{\ell+1}$ for all $i \in [\ell]$.
4. Output $(x_1, \ldots, x_{\ell+1})$.

As $\mathcal{B}$'s simulation of Game_3 is perfect,

$$\Pr\left[\begin{array}{c} \exists k \in [\ell+1] : \chi_k \neq 0 \wedge \\ \forall i \in [\ell+1] : \mathsf{Verify}(\mathbf{x}, m_i^*, \sigma_i^* = (\mathbf{r}_i^*, s_i^*)) = 1 \end{array}\right] = \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(\lambda). \tag{14}$$

If the first event occurs then $\mathcal{B}$ does not halt on item 1, and if the second occurs then for all $i \in [\ell+1]$,

$$\mathbf{g}^{s_i^*} = \mathbf{r}_i^* \mathbf{x}^{c_i^*}. \tag{15}$$

Since $\mathcal{A}$ is algebraic, it outputs $(\gamma_k^*, \xi_k^*, \rho_k^*)$ such that

$$\mathbf{r}_k^* = \mathbf{g}^{\gamma_k^*} \mathbf{x}^{\xi_k^*} \prod_{j=1}^{\ell} \mathbf{h}_j^{\rho_{k,j}^*}. \tag{16}$$

Using that $\mathbf{g}^{s_k^*} \mathbf{x}^{-c_k^*} = \mathbf{r}_k^*$ from eq. (15), we rewrite eq. (16) as

$$\mathbf{g}^{s_k^*} \mathbf{x}^{-c_k^*} = \mathbf{g}^{\gamma_k^*} \mathbf{x}^{\xi_k^*} \prod_{j=1}^{\ell} \mathbf{h}_j^{\rho_{k,j}^*}.$$

Plugging in that $\mathbf{g}^{s_j} \mathbf{x}^{-c_j} = \mathbf{h}_j$ for all $j \in [\ell]$ from eq. (13),

$$\mathbf{g}^{s_k^*} \mathbf{x}^{-c_k^*} = \mathbf{g}^{\gamma_k^*} \mathbf{x}^{\xi_k^*} \prod_{j=1}^{\ell} (\mathbf{g}^{s_j} \mathbf{x}^{-c_j})^{\rho_{k,j}^*}.$$

Finally, rewriting the equation, we get

$$\mathbf{g}^{s_k^* - \gamma_k^* - \sum_{j=1}^{\ell} \rho_{k,j}^* s_j} = \mathbf{x}^{c_k^* + \xi_k^* - \sum_{j=1}^{\ell} \rho_{k,j}^* c_j}.$$

Therefore $\mathbf{g}^{x_{\ell+1}} = \mathbf{g}^{\chi_k^{-1}(s_k^* - \gamma_k^* - \sum_{j=1}^{\ell} \rho_{k,j}^* s_j)} = \mathbf{x}$. Using eq. (13), we have $\mathbf{g}^{s_i}\mathbf{x}^{-c_i} = \mathbf{h}_i$ for all $i \in [\ell]$, hence $\mathbf{g}^{x_i} = \mathbf{g}^{s_i - c_i x_{\ell+1}} = \mathbf{g}^{s_i}\mathbf{x}^{-c_i} = \mathbf{h}_i$ for all $i \in [\ell]$. As $\mathcal{A}$ queries S_2 exactly ℓ times, $\mathcal{B}$ made exactly ℓ queries to DL, so $\mathcal{B}$ wins. We have that

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{Game}_3}(\lambda) = \Pr\left[\begin{array}{c} \exists k \in [\ell+1] : \chi_k \neq 0 \wedge \\ \forall i \in [\ell+1] : \mathsf{Verify}(\mathbf{x}, m_i^*, \sigma_i^* = (\mathbf{r}_i^*, s_i^*)) = 1 \end{array}\right]. \tag{17}$$

Combining eq. (14) and eq. (17) yields the claim. □

Difficulties Generalizing to Polylog-Concurrency. One might ask whether our security proof in the 2-concurrent setting can be generalized to the η-concurrent setting where $\eta = O((\log \lambda)^k)$. As discussed in the introduction, all of our argument can essentially be reused in the more general setting, except for the case where the probability of the "bad event" cannot be upper-bounded via a reduction from ROS—i.e., Game_3 in the formal proof. To see the difficulties here, recall that our proof relies on analyzing signing session groupings represented by a graph G, where vertices correspond to signing sessions and two vertices are connected if a special query is made while both signing sessions are open. Even in the 3-concurrent setting, it is unclear what G would become, since it is possible that a special query is made during three sessions. In fact, the number of "essentially different cases" of how signing sessions may interleave appears to grow rapidly while η increases, so we might need more powerful tools from combinatorics and/or linear algebra to make an argument for the more general η-concurrent setting.

References

[BFP21] Balthazar Bauer, Georg Fuchsbauer, and Antoine Plouviez. The one-more discrete logarithm assumption in the generic group model. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 587–617. Springer, Cham, December 2021. doi:10.1007/978-3-030-92068-5_20.

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 1087–1098. ACM Press, November 2013. doi:10.1145/2508859.2516687.

[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. doi:10.1007/s00145-022-09436-0.

[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. doi:10.1007/978-3-031-68376-3_6.

[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. doi:10.1007/978-1-4757-0602-4_18.

[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. doi:10.1007/978-3-031-38557-5_23.

[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. doi:10.1007/3-540-48071-4_7.

[DH22] Gian Demarmels and Lucien Heuzeveldt, 2022. Accessed: 2024-04-11. URL: https://taler.net/papers/cs-thesis.pdf.

[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. doi:10.1007/978-3-319-96881-0_2.

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology - AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1992. doi:10.1007/3-540-57220-1_66.

[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. doi:10.1007/978-3-030-45724-2_3.

[FW24] Georg Fuchsbauer and Mathias Wolf. Concurrently secure blind schnorr signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 124–160. Springer, Cham, May 2024. doi:10.1007/978-3-031-58723-8_5.

[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. doi:10.1007/BFb0052233.

[KLX22] 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. doi:10.1007/978-3-030-97131-1_16.

[KNR24] Julia Kastner, Ky Nguyen, and Michael Reichle. Pairing-free blind signatures from standard assumptions in the ROM. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 210–245. Springer, Cham, August 2024. doi:10.1007/978-3-031-68376-3_7.

[Nic19] Jonas Nick. Blind signatures in scriptless scripts, 2019. Accessed: 2024-04-11. URL: https://jonasnick.github.io/blog/2018/07/31/blind-signatures-in-scriptless-scripts/.

[NS01] Phong Q. Nguyen and Igor Shparlinski. On the insecurity of a server-aided RSA protocol. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 21–35. Springer, Berlin, Heidelberg, December 2001. doi:10.1007/3-540-45682-1_2.

[Sch01] Claus-Peter Schnorr. Security of blind discrete log signatures against interactive attacks. In Sihan Qing, Tatsuaki Okamoto, and Jianying Zhou, editors, *ICICS 01*, volume 2229 of *LNCS*, pages 1–12. Springer, Berlin, Heidelberg, November 2001. doi:10.1007/3-540-45600-7_1.

[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. doi:10.1007/3-540-69053-0_18.

[SPMS02] Jacques Stern, David Pointcheval, John Malone-Lee, and Nigel P. Smart. Flaws in applying proof methodologies to signature schemes. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 93–110. Springer, Berlin, Heidelberg, August 2002. doi:10.1007/3-540-45708-9_7.

[Wag02] David Wagner. A generalized birthday problem. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, Berlin, Heidelberg, August 2002. doi:10.1007/3-540-45708-9_19.