Two-round trip Schnorr multi-signatures via delinearized witnesses

Handan Kilinc Alper and Jeffrey Burdges

Web 3.0 Foundation

Abstract. We introduce a new *m-entwined ROS* problem that tweaks a random inhomogeneities in an overdetermined solvable system of linear equations (ROS) problem in a scalar field using an associated group. We prove hardness of the 2-entwined ROS-like problem in AGM plus ROM, assuming DLOG hardness in the associated group.
Assuming AGM plus ROM plus KOSK and OMDL, we then prove security for a two-round trip Schnorr multi-signature protocol DWMS that creates its witness aka nonce by delinearizing two pre-witnesses supplied by each signer.
At present, DWMS and MuSig-DN are the only known provably secure two-round Schnorr multi-signatures, or equivalently threshold Schnorr signatures.

All cryptographic schemes have nasty footguns underneath the surface. If a scheme matures properly, these become hidden away behind either the interfaces to cryptographic libraries or preferably by the underlying protocol being made miss-use resistant. Yet, growing up is hard.

Increased operational security demands have driven a growth spurt in multi-signer implementations for signature schemes, including Schnorr. At their core, any multi-signer scheme should protect each individual participating honest signer against forgeries by an adversary who controls all other signers and interacts extensively with our one honest signer.

Yet in [7], Drijvers, et al. broke all previously known multi-signer Schnorr protocols, using the traumatic ROS or k-SUM lesson that nearly killed blind Schnorr signatures. In short, there are forgery attacks against the known two round trip Schnorr signing protocols [1,11,17,12] that work if the adversary engages in enough parallel signing sessions.

In theory, deployments could forbid parallel signing sessions, but footguns abound, and include such horrors as warning users not to place related keys on too many machines. After [7], one required a three round trip signing protocol in which parties first commit to their witness share, second reveal their witness share, and third send their signature share. Although miss-use resistant, an extra round trip brings deployment problems too, so some protocol designers continue making ad hoc arguments that their deployments remain unaffected.

We propose an extremely simple and lightweight two-round multi-signer Schnorr protocol, called delinearized witness multi-signatures (DWMS): first all signers propose two pre-witnesses curve points, and second after obtaining all pre-wtinesses then all signers compute the shared witness by delinearizing these pre-witnesses with a random oracle and produce their signature share using their portion of the combined witness.

We have since January 2020 provided our *delinearized witness* protocol as an option for multi-signatures in the schnorrkel/sr25519 [4] signature scheme used by substrate based blockchains.

In this work, we give a security proof for DWMS in the algebraic group model (AGM) [8], under a knowledge of secret key (KOSK) assumption for the adversary, and assuming hardness of the one more discrete logarithm (OMDL) problem [7, Definition 2]. We deduce that 2-DWMS is secure in

the generic group model (GGM) because OMDL is hard in GGM by [6, Table 2 or §5]. Along the way, we introduce the 2-entwined ROS problem that captures the mathematical problem underlying DWMS, and prove its hardness in AGM.

Aside from DWMS, MuSig-DN [13] is the only other Schnorr multi-signature protocol with a security proof.

MuSig-DN provides deterministic witnesses, a lovely property previously unavailable in a Schnorr multi-signature. It achieves determinism using several beautiful and novel bulletproof optimizations. In MuSig-DN, the first round messages require only 1124 bytes per signer, but their participant only benchmarks show 0.9 second proving times. MuSig-DN requires no additional hardness assumptions, but exploits features bespoke to the secp256k1 curve.

In DWMS, all signers incur a per signer cost of only 64 bytes and only two scalar multiplications. DWMS requires the AGM+OMDL hardness assumptions, but asks no special features of the underlying group. DWMS permits agrement upon the message during the second round.

Also, DWMS is extremely simple, next to the underlying multi-signature implementation. Yet, our second round message agreement comes with the cost that DWMS implementations should prevent witnesses being reused or even saved on disk. MuSig-DN avoids this with determinism.

We break the paper down as follows:

In §1, we introduce DWMS multi-signer protocol and discuss some related concerns. In §2, we introduce the entwined ROS problem and discuss related work on the ROS problem and security proofs for multi-signatures. We refer the reader to [7] for a deeper discussion of past multi-signature protocols. In §3, we recall the algebraic group model (AGM) and introduce doing linear algebra in this model.

After these preliminary sections, we prove hardness of the entwined ROS problem in §4, largely by doing a modified Gaussian elimination in augmented matrices built from non-independent random oracles in AGM. We prove DWMS secure in §5 using a diret reduction to OMDL in AGM. We abuse AGM aggressively in this argument as well, so it remains an open question to reduce DWMS to OMDL, assuming hardness of the entwined ROS problem or similar, but only using more common techniques.

Acknowledgement

The authors warmly thank Raghav Bhaskar for his extensive advise and extremely insightful conversations, including suggestions that pointed to the entwined ROS problem. We also warmly thank Michele Orrù for his helpful conversations, especially around understanding the algebraic and generic group models.

1 Multi-signer Schnorr

We give our two-round *delinearized witness multi-signature* (DWMS) protocol that replaces the witness sharing and combination steps in multi-signer Schnorr protocol with a delinearization phase inspired by the delinearization defense against rogue key attacks in [3].

We recall that Schorr signatures consist of four algorithms for paramater generation, key generation, signing, and verification. Any multi-signer Schnorr protocol leaves paramater generation and verification unchanged, but alter key handling and signing.

We fix the paramater generation results $(\mathbb{G}, p, P)$ throughout this article: We have a group $\mathbb{G}$ of prime order p, in which the discrete logarithm (DLOG) problem is hard. We distinguish one point

P in $\mathbb{G}$ as our default generator. We write $\mathbb{G}$ additively throughout because the multiplicative form obfuscates the underlying linear algebra.

In this work, we defend an individual honest signer against forgeries orchestrated by by cosigners with whom they interacted previously. We focus almost exclusively upon the parallel signing session attacks in [7]. As our method applies broadly, we require a formalism to unify the defenses against rogue key attacks discussed elsewhere like [3].

All Schnorr multi-signature schemes replace the single key generation algorithm with one key generation protocol as well as public and secret key combination algorithms:

- KeyGen returns a public and secret signer key pair $(Y, y) \in \mathbb{G} \times \mathbb{F}_p$ with $Y = yP$, perhaps after interacting with other prospective signers, and perhaps augmenting Y with data beyond only the public $\mathbb{G}$.
- $\mathsf{TweakKey}$ accepts a list K of augmented public keys with one $Y \in K$ distinguished, and returns a public scalar $a_{K,Y} \in \mathbb{F}_p$. We then define both $x_y = a_{K,Y} y$ if $Y = yP$ and $X = \sum_{Y \in K} a_{K,Y} Y$, and write $(X, x_y) = \mathsf{TweakKey}(K, Y, y)$ or $X = \mathsf{TweakKey}(K)$.

At present, most popular multi-signer protocols have the property that their public key records Y prove knowledge of y and the resulting x_y. We could attach a proof-of-possession here, but also most threshold schemes provide an implicit prove-of-knowledge to other signers too, and make $a_{K,Y}$ a Lagrange coefficient. We assume this proof-of-knowledge throughout because it fits with algebraic group model.

In [3], there is an argument that delinearizing the signers keys like $a_{K,Y} = H(K, Y)$ also prevents rogue key attacks, which fits the $\mathsf{TweakKey}$ model but never proves knowledge. In AGM however, we remark that [3] proves $X[Y] \neq 0$ for each $Y \in K$, even if some $Y'' \in K$ satisfies $Y''[Y'] \neq 0$ for a distinct $Y' \in K$. We ignore this variant because it appears rarely in practice and works less cleanly in the algebraic group model.

In DWMS, we instantiate two random oracles, the usual Schnorr challenge $H : \Omega_0 \times \mathbb{G}^2 \to \mathbb{F}_p$ and delinearization $H_1 : \Omega_0 \times \mathbb{G}^{2n+1} \times \mathbb{N}^2 \to \mathbb{F}_p$, for some message space Ω_0 and any $n \geq 1$. We create a typical Schnorr signature (T, s) so our Verfiy routine merely check the usual verification equation $sP = T + H((\omega, X), T)X$.
We define the $\mathsf{DWMS-Sign}$ protocol as follows.

- *Round 1*: The ith signer samples two ephemeral secrets $r_{i,1}, r_{i,2} \in \mathbb{F}_p$ randomly and then broadcasts two corresponding pre-witnesses $T_{i,j} = r_{i,j} P$ for $j = 1, 2$ to all other signers, along with its identity i. It waits for the pre-witnesses of other signers.
- *Round 2*: After agreement upon the message $\omega \in \Omega_0$ and signer set K with $n := |K|$, our ith signer obtains the key pair $(X, x_y) = \mathsf{TweakKey}(K, Y, y)$ and then generates the session identifier $\mathtt{transcript} = ((\omega, X), T_{1,1}, T_{1,2}, \ldots, T_{n,1}, T_{n,2})$. It next computes delinearization scalars $\alpha_{i,j} = H_1(\mathtt{transcript}, i, j)$ for $j = 1, 2$, the delinearized shared witness $S = \sum_{i=1}^{n} \alpha_{i,1} T_{i,1} + \alpha_{i,2} T_{i,2}$, and challenge $c = H((\omega, X), S)$. It finally sends the ith partial signature $s_i = \alpha_{i,1} r_{i,1} + \alpha_{i,2} r_{i,2} + c x_y$. After receiving the partial signatures s_i of other signers, it outputs $\sigma = (S, s)$ as the signature where $s = \sum_{i=1}^{n} s_i$.

We credit [3] with inspiring our approach, but caution that [3] could be missread as implying security for delinearization factors based on a potential signer set larger than the immediate signer set. In fact, you cannot securely base delinearization factors upon some such larger set: An rogue key attack based on [18] requires extreme numbers of signers, but even this sounds plausible if

delinearization were based upon some recent block hash inside a blockchain containing all signer's keys. We expect [2] reduces the required number of adversarial singers to roughly $2 \log p = 512$, making this attack viable.

Aside from multi-signer schemes, there exist other variations on Schnorr signatures:

In [9], there now finally exists a secure Schnorr blind signing protocol, in which a two-round trip **Sign** protocol avoids the message signer obtaining the message. We doubt DWMS integrates naively with [9] however.

"Adaptor" signatures alter the proof-of-knowledge provided by the sP in the verification equation, which implicitly provides a certificate. We expect this too merges seamlessly with DWMS, but again doing so requires analysis, which could prove interesting.

We next introduce the entwined ROS problem and show its hardness. We prove the security of DWMS after that in Section 5 since its security based on the hardness of the entwined ROS problem.

2 Entwined ROS problem

In [15], Peter Schnorr observed a flaw in the existing security argument from [14] for blind Okamoto-Schnorr and blind Schnorr signatures: Any blind signer invariably admits multiple parallel signing sessions, so an adversary could search for relations among challenge oracle results while running several sessions in parallel. Schnorr introduced the *random inhomogeneities in an overdetermined solvable system of linear equations* (ROS) problem, which if hard then yields security despite the weakness. In [18] however, David Wagner demonstrated a sub-exponential time generalized birthday bound attack upon the ROS problem, which he termed a k-sum algorithm.

Definition 1 (ROS [2])**.** Let H_{ros} be a random oracle. Find $\ell+1$ affine functions ρ_i, $\ell+1$ messages ω_i, and a vector $(c_0, \ldots, c_{\ell-1})$ such that $H_{\text{ros}}(\rho_i, \omega_i) = \rho_i(c_i)$ for all $i \in \ell$.

All previous two-round Schnorr multi-signatures and threshold signatures were broken by [7] using this subexponential time k-sum attacks, assuming parallel sessions. Also, [2] discovered a polynomial time attack, using $q_s > \log p$ parallel sessions, which breaks several ad hoc fixes proposed previously.

We introduce an *m-entwined ROS* problem in which an attacker works not only in the scalar field $\mathbb{F}_p$ but also in its associated group $\mathbb{G}$.

We define *pre-witness indices* as a paramater: We let $m \geq 1$ be a positive integer and let I' be an index set with $|I'| = m$, called the *honest pre-witness indices*. We let I'' be another index set disjoint from I', called the *adversarial pre-witness indices*, and define $I := I' \cup I''$.

As discussed above, we protect an honest participant numbered 1 from forgeries by dishonest parties numbered $2, \ldots, n$ with whom they cosigned other messages. We cover some niche situation with the abstracted the pre-witness index sets I' and I''. In practice though, we denoted by m the number of pre-witnesses supplied by each participant, and by n the number of participants So then one takes $I' = \{1\} \oplus [m] = \{(1,1), \ldots, (1,m)\}$ and takes $I'' = \{2, \ldots, n\} \oplus [m]$ to be the set of (i,j) with $2 < i < n$ and $j \in [m]$. Now $|I| = mn$. We use $'$ for honest and $''$ for adversarial throughout.

We fix a message set $\Omega = \Omega_0 \times \mathbb{G}$ that captures actual messages Ω_0 and combined public key used in DWMS, viewing the public key in $\mathbb{G}$ as an identity label. In doing this, we both avoid excess notation for handling public keys, while also emphasizing that public key should always be hashed into the challenge in practice.

We provide access to a Schnorr challenge random oracle $H : \Omega \times \mathbb{G} \to \mathbb{F}_p$ and a new delinearization random oracle $H_1 : \Omega \times \mathbb{G}^I \to \mathbb{F}_p^I$. In this, we view H_1 as a vector valued function or curried function sending messages times pre-witness assignments to maps in $\mathbb{F}_p^I$ from pre-witness indices in I to final $\mathbb{F}_p$ values.

We also fix a parallel session bound $q_s > 1$. We recall the notation $[q_s] = \{1, .., q_s\}$ and that $\mathbb{F}_p^U$ denotes functions from the set U to $\mathbb{F}_p$. We abuse direct sum notation $\oplus$ aggressively to denote Cartesian products as sets, including with singletons.

Definition 2 (**$(I', I''; q_s)$-entwined ROS game in $(\mathbb{F}_p, \mathbb{G})$.**)
We fix a subset $\mathcal{L}_1$ of $\mathbb{G}$ that contains at least the basepoint P.

As input, we supply pre-witness group elements $\mathbf{t}' \in \mathbb{G}^{I' \times [q_s]}$, also written $\mathbf{t}' = (T_{i;l})_{i \in I', l \in [q_s]}$. As output, our adversary returns

1. a message list $\boldsymbol{\omega} = (\omega_l)_{0 \le l \le q_s} \in \Omega^{\{0,\dots,q_s+1\}}$,
2. a scalar field vector $\boldsymbol{\beta} = (\beta_l)_{l \in [q_s] \cup \mathcal{L}_1} \in \mathbb{F}_p^{[q_s] \cup \mathcal{L}_1}$, and
3. their own pre-witness group elements $\mathbf{t}'' = (T_{i;l})_{i \in I'', l \in [q_s]} \in \mathbb{G}^{I'' \times [q_s]}$.

We join $\mathbf{t}'$ and $\mathbf{t}''$ into $\mathbf{t} = \mathbf{t}' \cup \mathbf{t}''$ by union of functions, or equivalently concatenation along their indices if viewed as lists. For $0 < l \le q_s$, we define

- the session information $\texttt{transcript}_l := (\omega_l) \oplus (T_{i;l})_{i \in I}$,
- the delinearization scalars $\alpha_{i;l} = H_1(\texttt{transcript}_l)[i]$ for $i \in I$, and
- the session sum $S_l = \sum_{l=1}^{q_s} \alpha_{i;l} T_{i;l}$

We say the adversary wins if $\boldsymbol{\beta}$, the S_ls, and $\boldsymbol{\omega}$ satisfy the equation

$$\sum_{l=1}^{q_s} \beta_l H\left(\omega_l, S_l\right) = H\left(\omega_0, \sum_{L \in \mathcal{L}_1} \beta_L L + \sum_{l=1}^{q_s} \beta_l S_l\right) \tag{‡}$$

We write (m, q_s)-*entwined ROS* when the context does not require naming the index sets or m-*entwined ROS* when q_s is polynomial in $\log p$ or just *entwined ROS* when $m = 2$.

We abstracted the index sets I' and I'' here because our arguments below care little about the distinction between participants and their pre-witnesses. Interestingly, our m actually becomes the product of the number of honest participants and the number of pre-witnesses they each supply, but m would only be the number of pre-witnesses in practice since participants rarely know whether other honest participants exist.

2.1 Caveats

Importantly, our entwined ROS problem provides no protection unless $m > 1$.

Proposition 3. *The $(1, q_s)$-entwined ROS game has a polynomial time algorithm if $q_s \ge \log p$, and a subexponential time algorithm in general.*

Proof. We take $I' = \{(1, 1)\}$ since $m = 1$. Assume $\mathcal{L}_1 = \{P\}$. An adversary can choose $\beta_l^{-1} = \alpha_{(1,1);l}$ for $l \in [q_s]$ to be the deliniarization scalars, so the β_ls cancel out the $\boldsymbol{\alpha}$ in the $\beta_0 P + \sum_{l=1}^{q_s} \beta_l S_l$ sum on the RHS of (‡), which then holds this term constant. In choosing β_l, they alter only the individual session terms $\beta_l H\left(\omega_l, S_l\right)$ from the LHS which amounts to altering H. We conclude by applying either [2] if $q_s \ge \log p$ or [7] otherwise. □

If $i \in I'$ then $T_{i;l} = r_{i;l}g$ with $r_{i;l}$ is known only by the challenger. At this point, our reader might wonder if $T_{i;l} = r_{i;l}P$ with $r_{i;l}$ known by the adversary should hold for $i \in I''$. We might ask $\mathcal{A}$ provide a proof-of-knowledge for their $T_{i;l}$ either in the entwined ROS game or in DWMS itself, but 2-DWMS and even 3-DWMS are more efficient than such schemes. In fact, an easy replay of pre-witnesses attack exists:

Assuming m denotes participants, we define a related "1.5"-entwined ROS game by taking $\alpha_{(i,1);l} = 1$ and $\alpha_{(i,2);l} = H_1(\texttt{transcript}_l)$[2]. This is the ROS-like game applicable to FROST [10] aka "1.5"-DWMS.

Proposition 4. *The "1.5"-entwined ROS problem has a polynomial time algorithm if $q_s \geq \log p$, and a subexponential time algorithm in general.*

Proof. Our adversary simply removes the honest signers' undelinearized "half" pre-witnesses by choosing $T_{(2,1);l} = -T_{(1,1);l}$ for $l \in [q_s]$. In doing so, all these "half" pre-witnesses disapear from the RHS of (‡) because $\alpha_{(i,1);l} = 1$. We conclude by the argument in Proposition 3 above. □

We remark that FROST never provides a security proof, but instead applies an ad hoc heuristic that resembles a Fiat-Shamir transform, but makes little sense across parallel sessions (see "Extension of Proof to FROST" in [10, §6.2 p. 19]). It's clear that [7] provides a forceful argument against such ad hoc approaches for this problem, which the authors of [7] clarify when they write:

"Schemes without security proofs clearly have no place in modern cryptographic design, especially if efficient provably secure alternatives exist. Apparent resistance against obvious attacks says nothing about the security of a scheme because, as the sub-exponential attacks in this paper have shown, subtler attacks may always be hiding beneath the surface."

It follows from Proposition 4 that our security arguments cannot apply to FROST per se. Yet, we cannot deduce an attack on FROST aka "1.5"-DWMS based upon Proposition 4 alone because its simplistic adversary leaves the honest signature shares $s_{1;l}$ somewhat corrupted.

So the question remains: Is FROST secure?

We think "no" in that FROST appears to compose less well than 2-DWMS. In particular, we expect there exist "adaptor" signature protocols to which 2-DWMS adapts securely but FROST yields an insecure protocol, and for which reasonable applications exists.

2.2 Blind signatures

We briefly remark on the multi-signer blind Schnorr signature problem, meaning adversaries compromise most signers as well as many users who request tokens in parallel. We do not address blind signatures in this work, but blind signatures are intimately related multi-signers, first because the ROS problem impacts them both dramatically, and second because blind signature applications benefit form multiple signers.

We recall both the classic two round blind Schnorr signature broken by Peter Schnorr in [15], as well as the Clause blind Schnorr signature [9, Figure 9, §5]: In Clause, we run two parallel sessions of a classic blind Schnorr signature until the user returns two blinded challenges, at which point the signer randomly aborts one session and returns the other.

We doubt blind Schnorr signatures benefit much from MuSig-DN [13] because the blinding prevents full derandomization and because MuSig-DN appears prohibitively expensive for typical blind signature applications.

We expect DWMS provides one-round witness generation for Clause blind Schnorr signatures of course, but doing this assumes hardness of their MROS problem [9, Figure 10, §5], which competes

favorably with previously known options. Yet, any blind multi-signature based upon MROS requires agreement upon aborting the session, which adds rounds.

We arrive at this question: *Is there a two round multi-signer protocol for blind Schnorr signatures that avoids session aborts, presumably by invoking entwined ROS instead of an MROS hardness assumption?*

Our delinearizing random oracle H_1 necessarily depends upon the combined public key and message $\omega_l \in \Omega$. If not, our adversary could perform a k-sum attack by holding constant both the the LHS witnesses S_l and the RHS witness $\beta_0 P + \sum_{l=1}^{q_s} \beta_l S_l$ while varying the LHS challenge $H(\omega_l, S_l)$ and RHS challenge $H(\omega_0, \cdots)$ by changing the ω_ls.

This complicates addressing blind signatures directly using only the entwined ROS problem. Yet, we conjecture that blind Schnorr signatures can be securely based upon entwined ROS, perhaps with a "homomorphic" challenge hash H.

In [5, Theorem 2.2 §2.1], there is a security proof for Schnorr signatures that replaces ROM with GGM, and admits a somewhat homomorphic challenge oracle H. We need a true random oracle in both §4 and §5 below, as does [9]. In fact, any multi-signar Schnorr demands a collision resistant challenge oracle by [7, §5.3]. Yet, one might adapt arguments from [5] to the multi-signer setting using another more nuanced somewhat homomorphic challenge oracle. If so, the users' blinding factors could hide the challenge first, so that signers compute and incorporate their witness only afterwards.

3 Linear relations in AGM

We employ a simple but more flexible variant of the DLOG assumption in which an adversary discovering any new relationship among group elements interests us.

Definition 5 (discrete log relation (DLR) problem)**.** If given $X_i \in \mathbb{G}$ for $i = 1, \ldots, k$ with $k > 1$ then find $y_i \in \mathbb{F}_p$ not all zero such that $\sum_{i=1}^{k} y_i X_i = 0$.

We know discrete logarithm relations (DLR) become trivial in groups for which solvers exist for the discrete logarithm/division problem (DLOG), but DLOG also reduces to DLR.

Proposition 6. *DLOG reduces to DLR.*

Proof. Let X denote our DLOG challenge over some base point P. We choose Pedersen commitments $X_i = a_i X + b_i P$ for random $a_i, b_i \in \mathbb{F}_p$. We may assume $\sum_i y_i a_i \neq 0$ except with probability $\frac{1}{p}$ because these Pedersen commitments X_i are perfectly hiding. It follows that $X = \frac{\sum_i y_i b_i}{\sum_i y_i a_i} P$, as desired. □

We recall $\mathbb{F}_p^{\mathcal{L}}$ denotes the vector space of dimension $|\mathcal{L}|$ over $\mathbb{F}_p$ given by maps $\mathcal{L} \to \mathbb{F}_p$. Assuming $\mathcal{L} \subset \mathbb{G}$, anytime we have a vector $X \in \mathbb{F}_p^{\mathcal{L}}$ then we define a distinguished homomorphism to $\mathbb{G}$ by $\sum_{\mathbb{G}} X = \sum_{V \in \mathcal{L}} X[V] \cdot V$.

We always take $\mathcal{L}$ to be a DLR challenges below, so intuitively either our adversary $\mathcal{A}$ could solve the DLR problem $\mathcal{L}$ if it wished, or else $\mathcal{A}$ perceives the map $\sum_{\mathbb{G}} : \mathbb{F}_p^{\mathcal{L}} \to \mathbb{G}$ as injective. We need non-blackbox access to our adversaries' computations in $\mathbb{G}$ to formalize this intuition however.

We could realize this intuition well by working in the generic group model (GGM) [16]. In fact, one could tweak the generic reduction from the algebraic group model (AGM) to GGM in [8], which already builds elements of $\mathbb{F}_p^{\mathcal{L}}$ from group oracle invokations.

Instead, we work solely in the algebraic group model (AGM) for $\mathbb{G}$ [8] both for proving hardness of the m-entwined ROS-like problem and for reducing it to our two-round trip Schnorr multi-signature protocol.

AGM entails two game alterations: First, our games track the list $\mathcal{L}$ of all group elements provided to the adversary $\mathcal{A}$. Second, our games demand $\mathcal{A}$ outputs elements of $\mathbb{F}_p^{\mathcal{L}}$ wherever we describe the adversary as outputting an element of $\mathbb{G}$, so both in final outputs and in other oracle invokations.

In GGM, a reduction observes each equality tests in $\mathbb{G}$ performed by $\mathcal{A}$, and hence it knows everything $\mathcal{A}$ knows about $\mathbb{G}$. We seemingly loose this power in AGM since $\mathcal{A}$ might know more about $\mathbb{G}$ than we do: $\mathcal{A}$ could perform equality tests in $\mathbb{G}$ without informing us, or even possess a DLOG solver to which $\mathcal{A}$ denies us access.

We make up this shortfall because anytime $\mathcal{A}$ invokes an oracle on $\mathbb{G}$ the representation it supplies acts like a commitment to its view. We give an easy case for the importance of this sequencing in the remainder of this section.

Analogously with AGM, we define the *weak AGM* to consist of two game alterations: First, our games track the list $\mathcal{L}_n$ of all group elements provided to the adversary $\mathcal{A}$ by its nth oracle invokation, like in AGM. Second. our games demand that $\mathcal{A}$'s final output reveal elements of $\mathbb{F}_p^{\mathcal{L}_n}$ that correspond to any elements of $\mathbb{G}$ that $\mathcal{A}$ provided during its nth oracle invokations.

We also define *weak AGM plus DL* to be the weak AGM together with a discret logarithm oracle that only responds to the adversary, not the challenger, and neither reports its queries to the challenger nor imposes the weak AGM restriction

Proposition 7. *There is a weak AGM plus DL adversary that breaks m-entwined ROS for any m.*

Proof. Our adversary runs a k-sum attack against the usual ROS problem by replacing the sum $\beta_0 g + \sum_{l=1}^{q_s} \beta_l T_l$ on the RHS of (‡) with any group elements they like, which yields LHS elements T_l for $l = 1, \ldots, q_s$ and an RHS element T, with which the adversary answers.

Our adversary finally answers the delinearization oracle H_1 debts by these T_l: It next invokes DL to obtain β_0 such that $\beta_0 g + \sum_{l=1}^{q_s} \beta_l T_l$, which it returns. It finally finds unobjectionable answers for all other queries. □

In other words, a weak AGM adversary who solves discrete logarithms can solve m-entwined ROS for any m without revealing that they broke the discrete logarithm assumption. It follows that m-entwined ROS cannot easily be proven secure in the weak AGM or any lesser security model, including any black box model.

As usual in AGM, we support adversaries who sample random group elements similarly to [8, pp. 5]. There is a designated generator $P \in \mathcal{L}$ with which we simulate a random oracle $H_{\mathbb{G}}$ that maps into $\mathbb{G}$ using a secret random oracle H' that maps into $\mathbb{F}_p$, so $H_{\mathbb{G}}(m)$ returns $H'(m)P$ and adds this value to $\mathcal{L}$. An adversary who detects this replacement would have violated the random oracle assumption on H'. We may ignore such oracles presence because they appear only in the g term.

4 Hardness of 2-Entwined ROS

We consider an AGM adversary $\mathcal{A}$ that solves the m, q_s-entwined ROS problem, while invoking H_1 and H together fewer than q_h times.

Algorithm 8 (DLR reduction to Entwined ROS)**.** We fix index sets I', I'' with $|I'| = m$. Also fix our additional points $\mathcal{L}_1 \subset \mathbb{G}$ such that $P \in \mathcal{L}_1$. We construct an adversary $\mathcal{B}$ for the mq_s-DLR problem, which simulates the $(I', I''; q_s)$-entwined ROS problem against $\mathcal{A}$:

First, $\mathcal{B}$ accepts a DLR challenge $\mathcal{L}_0 : I' \times [q_s] \to \mathbb{G}$ consisting of mq_s group elements in $\mathbb{G}$, which we regard as a map. We set $\mathcal{L} = \mathcal{L}_0 \cup \mathcal{L}_1$ extending the $\mathcal{L}_0$ map by our additional symbols in $\mathcal{L}_1$, including our basepoint.

Next, $\mathcal{B}$ randomly chooses $\zeta_{i;l} \in \mathbb{F}_p$ for $i \in I'$ and $l \in [q_s]$: We define $T_{i;l} = \zeta_{i;l} P + \mathcal{L}_0[i;l]$ to be Pedersen commitments to the elements of $\mathcal{L}_0$. We also define $\mathbf{t}' = (T_{i;l})_{\substack{i \in I' \\ l \in [q_s]}} \in \mathbb{G}^{I' \times [q_s]}$.

Next, $\mathcal{B}$ invokes $\mathcal{A}$ on $\mathbf{t}'$. $\mathcal{B}$ simulates and observes invokations of the random oracles H and H_1. As $\mathcal{A}$ lives in AGM, anytime $\mathcal{A}$ queries our oracles H or H_1, which includes its final answer, then $\mathcal{A}$ represents any required group elements as elements of $\mathbb{F}_p^{\mathbf{t}' \cup \mathcal{L}_1}$. $\mathcal{B}$ translates vectors $V \in \mathbb{F}_p^{\mathbf{t}' \cup \mathcal{L}_1}$ that $\mathcal{A}$ supplies to oracles into its preferred $\mathbb{F}_p^{\mathcal{L}}$ vector.

$$\begin{aligned} V \mapsto V &+ \Big(\sum\nolimits_{\substack{i \in I' \\ l \in [q_s]}} \zeta_{i;l} V[i;l] \Big) P \\ &= \Big(V[L] \Big)_{L \in \mathcal{L}_1 \setminus \{P\}} \oplus \Big(V[P] + \sum\nolimits_{\substack{i \in I' \\ l \in [q_s]}} \zeta_{i;l} V[i;l] \Big)_{P} \oplus \Big(V[i;l] \mathcal{L}_0[i;l] \Big)_{\substack{i \in I' \\ l \in [q_s]}} \end{aligned}$$

$\mathcal{B}$ stores these translated oracle invocations from $\mathcal{A}$, inside a database $\mathbb{A}_{\mathcal{L}} \subset \mathbb{F}_p^{\mathcal{L}}$, which we elaborate upon below.

Finally, if $\mathcal{A}$ ever gives any two distinct $x, y \in \mathbb{A}_{\mathcal{L}}$ with $\sum_{\mathbb{G}} x = \sum_{\mathbb{G}} y$ then $\mathcal{B}$ returns these as mq_s-DLR answers.

As a convenience, we identify elements of $\mathbf{t}'$ with their corresponding vector in $\mathbb{F}_p^{\mathcal{L}}$, meaning we regard $T_{i;l}$ as an element of $\mathbb{G}$ or $\mathbb{F}_p^{\mathcal{L}}$ in context. $\mathcal{B}$ avoids introducing any further group elements itself of course. We create any random group elements sampled by $\mathcal{A}$, as discussed in §3.

We give a two step probabilistic argument that this algorithm is a reduction:

We can assume $\mathcal{A}$ invokes the oracles H_1 and H for its final answer because otherwise it guesses blindly. First, we show these final oracle invocations act somewhat like "commitments" to $\mathcal{A}$'s choices of representatives in $\mathbb{F}_p^{\mathcal{L}}$, by exploring the injectivity-like properties of two maps who composition yields the RHS group element in (‡): We introduce the $\phi_{\mathbb{A}}$ map below and analyse it in Lemma 18, but the $\sum_{\mathbb{G}}$ map is immediate:

Lemma 9. *If $\mathcal{B}$ fails then $\sum_{\mathbb{G}}$ is injective on $\mathbb{A}_{\mathcal{L}}$.*

Second, we show in Theorem 19 below that if $\mathcal{A}$ acts consistently with its oracle query "commitments" then $\mathcal{A}$ actually explores too few (‡) solutions, and thus cannot solve the entwined ROS often enough.

It shall then follow that any $\mathcal{A}$ that solves entwined ROS in AGM yields a $\mathcal{B}$ that solves DLR via Definition 8.

4.1 Oracle records

Any *candidate* lth session consists of an oracle record index ℓ that uniquely indicates l and its data $(\omega_\ell, \mathbf{t}_\ell)$, which consists of both the output pre-witnesses $T_{i;\ell} \in \mathbb{F}_p^{\mathcal{L}}$ for $i \in I''$ and message $\omega_\ell \in \Omega$.

$\mathcal{A}$ considers numerous candidate lth sessions $(\omega_\ell, \mathbf{t}_\ell)$ by querying the delinearization H_1 and Schnorr H oracles, which $\mathcal{B}$ observes via these oracles. We can compute oracles from $(\ell, \omega_\ell, \mathbf{t}_\ell)$ since the session index l uniquely determines the data $(T_{i;l})_{i\in I'} = \mathbf{t}'[\![I', l]\!]^t$ derived from our challenge input.

Recall, we define $\texttt{transcript}_\ell$, $\alpha_{i;\ell}$, and S_ℓ as in the entwined ROS game, meaning $S_\ell = \sum_{l=1}^{q_s} \alpha_{i;\ell} T_{i;\ell}$ where the $\alpha_{i;\ell} = H_1(\omega, (T_{i;\ell})_{i\in I})[i]$ are random oracles dependent upon $(T_{i;\ell})_{i\in I''}$ and ω_ℓ. We write $S_l = (\alpha_{i;\ell})_{i\in I} \cdot (T_{i;\ell})_{i\in I} = \sum_{i\in I} \alpha_{i;\ell} T_{i;\ell}$ using vector notation.

Although $\mathcal{A}$ only returns one final answer to $\mathcal{B}$, we speak about potential full game $(\boldsymbol{\beta}, \boldsymbol{\omega}, \mathbf{t})$ which consist q_s candidate sessions and $\boldsymbol{\beta} \in \mathbb{F}_p^{[q_s]\cup\mathcal{L}_0}$, but might not make $\mathcal{A}$ win by satisfying (‡). Implicitly, potential full games determine a maps $l \mapsto \ell$ that selects an oracle record ℓ for each session. We therefore abuse notation by writing l or (l, k) for ℓ whenever the context makes full game clear or determines it by k.

We define the map $\phi : \mathbb{F}_p^{[q_s]\cup\mathcal{L}_0} \times \left(\Omega \times (\mathbb{F}_p^{\mathcal{L}})^{I''}\right)^{q_s} \to \mathbb{F}_p^{\mathcal{L}}$ from potential full games $(\boldsymbol{\beta}, \boldsymbol{\omega}, \mathbf{t})$ to representatives in $\mathbb{F}_p^{\mathcal{L}}$ for RHS group elements, given by

$$\phi(\boldsymbol{\beta}, \boldsymbol{\omega}, \mathbf{t}) = \sum_{L\in\mathcal{L}_0} \beta_L L + \sum_{l=1}^{q_s} \beta_l S_l.$$

In this, we attach session inputs together by rearranging the lth session message ω_l for $l > 0$ to accompany the lth session pre-witnesses $(T_{j;l})_{j\in I''} \in \mathbb{F}_p^{\mathcal{L}}$ for $i \in I''$, and omit ω_0 from $\boldsymbol{\omega}$ since it plays no role in the RHS group elements.

We next distinguish candidate sessions as being those oracle invokations that chain H and H_1 together correctly for (‡). We distinguish two tables among the oracle invocation records $\mathbb{A}_{\mathcal{L}}$: the Schnorr oracle calls $\mathbb{A}_H \subset \mathbb{F}_p^{\mathcal{L}} \times \Omega$, and the curried delinearization oracle calls $\mathbb{A}_{H_1} \subset \Omega \times (\mathbb{F}_p^{\mathcal{L}})^n$. We let $\mathbb{A}'_{H_1} \subset \mathbb{A}_{H_1}$ denote those delinearization oracle calls later used by some Schnorr oracles call, so

$$\mathbb{A}'_{H_1} = \{\, (\omega_\ell) \oplus (T_{i;\ell})_{i\in I} \in \mathbb{A}_{H_1} \mid (\omega_\ell, T_\ell) \in \mathbb{A}_H \,\}$$

We define $\phi_{\mathbb{A}}$ to be ϕ restricted to $\mathbb{F}_p^{\{0,\ldots,q_s+1\}} \times (\mathbb{A}'_{H_1})^{q_s}$, meaning ϕ restricted to those collections of (queried) candidate sessions $(\boldsymbol{\omega}, \mathbf{t})$ on which $\mathcal{A}$ evaluates both H_1 and H.

Assuming hardness of the mq_s-DLR problem, we will prove hardness of the m, q_s-entwined ROS problem by proving roughly speaking that $\phi_{\mathbb{A}}$ has enough injectivity to act like a commitment to group element decompositions in AGM, and such commitments obstruct efficient Entwined ROS algorithms.

4.2 Polite injectivity

We observe that $\phi_{\mathbb{A}}$ cannot be injective on its lth session $(T_{i;l})_{i\in I}$ among those potential full games $(\boldsymbol{\beta}, \boldsymbol{\omega}, \mathbf{t})$ for which $\beta_l = 0$ for some $l \in [q_s]$. In our candidate session notation, an injectivity failure of $\phi_{\mathbb{A}}$ consists of a $(\boldsymbol{\omega}_k, \mathbf{t}''_k) \in \mathbb{A}'_{H_1}$ and $\boldsymbol{\beta}_k = (\beta_{l,k})_{l\in[q_s]\cup\mathcal{L}_1} \in \mathbb{F}_p^{[q_s]\cup\mathcal{L}_1}$ for $k = 1, 2$, that satisfy the $\mathbb{F}_p^{\mathcal{L}}$ vector equality $\phi(\boldsymbol{\beta}_1, \boldsymbol{\omega}_1, \mathbf{t}_1) = \phi(\boldsymbol{\beta}_2, \boldsymbol{\omega}_2, \mathbf{t}_2)$ aka

$$\sum_{L\in\mathcal{L}_0} \beta_{L,1} L + \sum_{l=1}^{q_s} \beta_{l,1} S_{l,1} = \sum_{L\in\mathcal{L}_0} \beta_{L,2} L + \sum_{l=1}^{q_s} \beta_{l,2} S_{l,2}. \tag{†}$$

Definition 9. We say $\phi_{\mathbb{A}}$ is *politely injective* if $\phi(\boldsymbol{\beta}_1, \boldsymbol{\omega}_1, \mathbf{t}_1) = \phi(\boldsymbol{\beta}_2, \boldsymbol{\omega}_2, \mathbf{t}_2)$ implies that, for $l \in [q_s]$, either $\beta_{l,1} = 0 = \beta_{l,2}$, or else $(\boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \mathbf{t}_k)$ have the same lth session for both $k = 1, 2$. So $\phi_{\mathbb{A}}$ is injective on $\mathbb{F}_p \times (\mathbb{F}_p^\times)^{q_s} \times \mathbb{A}'_{H_1}$ in particular.

We spend the next subsection proving that $\phi_{\mathbb{A}}$ is politely injective. All our arguments aries by viewing the vector equation (†) as a system of "row" equations corresponding to the coefficients of the $T_{i;l}$ for $i \in I'$ and $l \le q_s$. We first discuss in this section a couple useless straw-men that help explain our final argument.

As our first straw-man, we consider an internal proof-of-knowledge (IPoK) assumption that $\mathcal{A}$ honestly constructs all their $\mathbf{t}''$ as $T_{j;l} = r_{j;l} P$ from some $r_{i;l}$ with $j \in I''$ and $l \le q_s$.

Straw-man Lemma 10. *If $m \ge 2$ and IPoK holds in $\mathbb{A}_{\mathcal{L}}$, then $\phi_{\mathbb{A}}$ is politely injective, except with probability $\le \frac{2 q_s q_h}{p}$.*

Proof. Assume $\phi_{\mathbb{A}}$ is not politely injective, so (†) holds. According to IPoK, the system of equations (†) become merely $\beta_{l,1} \alpha_{i;(l,1)} = \beta_{l,2} \alpha_{i;(l,2)}$, along with some equation in coefficients of P. We may assume $(\boldsymbol{\omega}_1, \mathbf{t''}_1) \neq (\boldsymbol{\omega}_2, \mathbf{t''}_2)$ because otherwise $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$ as well.

As H_1 is a random oracle, it follows that $\alpha_{i;(l,k)} \neq 0$ with probability $\frac{1}{p}$ and that $\alpha_{i;(l,1)} / \alpha_{i;(l,2)}$ is random. If $\beta_{l,1} \neq 0$ then we reach the contraction $\alpha_{1;(l,1)} / \alpha_{1;(l,2)} = \beta_{l,2} / \beta_{l,1} = \alpha_{2;(l,1)} / \alpha_{2;(l,2)}$. It follows that $\beta_{l,k} = 0$ for $l \le q_s$ and $k = 1, 2$ except with probability $\le \frac{2}{p}$, so our result follows. □

Interestingly, we could prove FROST secure under this IPoK assumption, which implies an interesting if delicate strategy by which to attempt to prove vanilla FROST secure: Invent some weak aggregate proof-of-knowledge property that suffices to prove hardness of "1.5"-entwined ROS, and then also strengthen Theorem 23 below to exploit this new entwined ROS property.

We cannot expect real adversaries obey IPoK of course.[1] Instead we shall analyse $\phi_{\mathbb{A}}$ using linear algebra: If we fix $\boldsymbol{\omega}$ and $\mathbf{t}''$ then we obtain a linear transformation $\Phi_{\boldsymbol{\omega},\mathbf{t}''} : \mathbb{F}_p^{[q_s] \cup \mathcal{L}_1} \to \mathbb{F}_p^{\mathcal{L}}$ given by $\boldsymbol{\beta} \mapsto \phi_{\mathbb{A}}(\boldsymbol{\beta}, \boldsymbol{\omega}, \mathbf{t})$. As a matrix, $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$ has q_s columns numbered by session indices $l \in [q_s]$, each of which multiplies its β_l, as well as $m q_s + |\mathcal{L}_1|$ rows that each multiply a distinct element of $\mathcal{L}$ in the $\sum_{\mathbb{G}}$ map. In other words, its rows are numbered first by $(i; l)$ for inputs $i \in I'$ and session indices $l \in [q_s]$, each of which multiplies $\mathcal{L}_0[i, l]$ in the $\sum_{\mathbb{G}}$ map, as well as second by the elements of $L \in \mathcal{L}_1$ including P.

At this point, equation (†) amounts to $\Phi_{\boldsymbol{\omega}_1,\mathbf{t''}_1}(\boldsymbol{\beta}_1) = \Phi_{\boldsymbol{\omega}_2,\mathbf{t''}_2}(\boldsymbol{\beta}_2)$, We saw in Straw-man Lemma 10 that (IPoK) gave $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$ a jagged diagonal form with random oracles on the diagonal, which motivates our next trick: We decompose $\Phi_{\boldsymbol{\omega},\mathbf{t}''} = \Phi'_{\boldsymbol{\omega},\mathbf{t}''} + \Phi''_{\boldsymbol{\omega},\mathbf{t}''}$ with $\Phi'_{\boldsymbol{\omega},\mathbf{t}''}$ and $\Phi''_{\boldsymbol{\omega},\mathbf{t}''}$ containing all random oracles $\alpha_{i;l}$ for $i \in I'$ and $i \in I''$, respectively. We obtain

$$\begin{aligned} \Phi'_{\boldsymbol{\omega},\mathbf{t}''}[(i;l), l] &= \alpha_{i;l} \quad \text{and zero elsewhere, while} \\ \Phi''_{\boldsymbol{\omega},\mathbf{t}''}[L, y] &= \sum_{j \in I''} \alpha_{j;y} T_{j;y}[L] \quad \text{for } L \in (I' \times [s_q]) \cup \mathcal{L}_1. \end{aligned} \tag{1}$$

In particular, $\Phi'_{\boldsymbol{\omega},\mathbf{t}''}$ has a jagged diagonal form, while $\Phi''_{\boldsymbol{\omega},\mathbf{t}''}$ captures our deviation from (IPoK).

Straw-man Lemma 11. *If $m \ge 1$, then $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$ is injective, except with negligible probability.*

[1] Incorporating "real" VRF outputs into $\texttt{transcript}_l$ yields interesting protocols however.

As an intuition for this second straw-man, there is a unique random oracle on each row of $\Phi'_{\boldsymbol{\omega},\mathbf{t}''}$ that appears independent from all other elements of its column in $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$. Yet, one cannot formalize this intuition directly because any statement about the row rank of $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$ boils down to Gaussian elimination, which introduces relationships among the columns. As columns are not independent from one another, elementary operations conceivably break independence within some columns.

Vague proof sketch. We perform Gaussian elimination with row operations on $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$. Inside the elimination procedure, we construct an inductive argument that employs our random oracle assumptions to prove $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$ has row rank at least q_s, except with negligible probability. All this resembles but in simpler than Lemmas 12, 15, and 16 below. We deduce that $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$ is injective because row rank equals column rank. □

We know this straw-man cannot suffice because we saw a polynomial time algorithm for the 1-entwined ROS in Proposition 3. We recall $\phi_{\mathbb{A}}$ being politely injective means $\Phi_{\boldsymbol{\omega}_1,\mathbf{t}''_1}(\boldsymbol{\beta}_1) = \Phi_{\boldsymbol{\omega}_2,\mathbf{t}''_2}(\boldsymbol{\beta}_2)$ occurs only if, for session indices $l \in [q_s]$, either $\beta_{l,1} = 0 = \beta_{l,2}$, or else $(\boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \mathbf{t}''_k)$ have the same lth session for both $k = 1, 2$. We cannot express this condition with any statement about $\Phi_{\boldsymbol{\omega},\mathbf{t}''}$ alone. We therefore trash Straw-man Lemma 11 and instead perform similar algebra on the augmented matrix for this full equality (†).

4.3 Augmented Gaussian elimination

We consider $2q_s$ session oracle invokations $(\omega_{l,k}) \oplus (T_{j;(l,k)})_{j\in I''}$ in $\mathbb{A}'_{H_1} \subset \Omega \times (\mathbb{F}_p^{\mathcal{L}})^{I''}$ for $l \in [q_s]$ and $k = 1, 2$. Assemble these into $\boldsymbol{\omega}_k = (\omega_{l,k})_{l\in[q_s]}$ and $\mathbf{t}''_k = (T_{i;(l,k)} : i \in I'', l \in [q_s])$ for $k = 1, 2$.

If $\phi_{\mathbb{A}}$ were not politely injective, then two $(\boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \mathbf{t}''_k)_{k=1,2}$ exist such that (†) holds $\Phi_{\boldsymbol{\omega}_1,\mathbf{t}''_1}(\boldsymbol{\beta}_1) = \Phi_{\boldsymbol{\omega}_2,\mathbf{t}''_2}(\boldsymbol{\beta}_2)$, and for some $l \in [q_s]$ we have $\beta_{l,k} \neq 0$ for $k = 1, 2$ and either $\omega_{l,1} \neq \omega_{l,2}$ or $(T_{j;(l,1)})_{j\in I''} \neq (T_{j;(l,2)})_{j\in I''}$. We avert this scenario by analyzing the system of equations (†).

We view (†) as the augmented matrix $\mathbf{A}_0 = \Phi_{\boldsymbol{\omega}_1,\mathbf{t}''_1} \oplus \Phi_{\boldsymbol{\omega}_2,\mathbf{t}''_2}$. This augmented matrix form amounts to $(\Phi_{\boldsymbol{\omega}_1,\mathbf{t}''_1} \oplus \Phi_{\boldsymbol{\omega}_2,\mathbf{t}''_2})(\boldsymbol{\beta}_1 \oplus -\boldsymbol{\beta}_2)^t = 0$ of course. Again, we have mq_s rows with indexes from $(i; l) \in I' \times [q_s]$ that represent coefficients of $\mathcal{L}_0[i; l]$, along with $|\mathcal{L}_1|$ rows with indexes $L \in \mathcal{L}_1$ that represent coefficients of L, including P. Among this second group, we depict the row indexed by P as a "zeroth row". In the augmented matrix, we now have $2q_s$ columns that we index like $(l', k) \in [q_s] \times [2]$ to express that they represent coefficients of $\beta_{l',k}$. In other words, $\mathbf{A}_0[(i; l), (l', k)]$ aka $\mathbf{A}_0[ml + i, q_s(k - 1) + l']$ is the coefficient of $\beta_{l',k}\mathcal{L}_0[i; l]$ in (†) for $i \in I'$, $l, l' \leq q_s$, and $k = 1, 2$.

We apply the decomposition $\Phi_{\boldsymbol{\omega},\mathbf{t}''} = \Phi'_{\boldsymbol{\omega},\mathbf{t}''} + \Phi''_{\boldsymbol{\omega},\mathbf{t}''}$, discussed above to our augmented matrix, so $\mathbf{A}_0 = \mathbf{A}' + \mathbf{A}''$ where $\mathbf{A}' = \Phi'_{\boldsymbol{\omega}_1,\mathbf{t}''_1} \oplus \Phi'_{\boldsymbol{\omega}_2,\mathbf{t}''_2}$ and $\mathbf{A}'' = \Phi''_{\boldsymbol{\omega}_1,\mathbf{t}''_1} \oplus \Phi''_{\boldsymbol{\omega}_2,\mathbf{t}''_2}$. Above, we observed the jagged double diagonal augmented matrix $\mathbf{A}'$ inside the proof of Straw-man Lemma 10 and discussed that $\mathbf{A}'' \neq 0$ iff $\mathcal{A}$ violates (IPoK). We define $\zeta'_{(l,k)} = \sum_{i\in I'} \zeta_{i;l}\alpha_{i;(l,k)}$ as a notational convenience.

$$\mathbf{A}' = \left(\begin{array}{ccc|ccc} & \beta_{l,1} & & & \beta_{l,2} & \\ \cdots & \zeta'_{(l,1)} & \cdots & \cdots & \zeta'_{(l,2)} & \cdots \\ \ddots & & & \ddots & & \\ & \alpha_{1;(l,1)} & & & \alpha_{i;(l,2)} & \\ & \vdots & & & \vdots & \\ & \alpha_{i;(l,1)} & & & \alpha_{i;(l,2)} & \\ & \vdots & & & \vdots & \\ & \alpha_{m;(l,1)} & & & \alpha_{m;(l,2)} & \\ & & \ddots & & & \ddots \end{array}\right) \begin{array}{c} P \\ \\ \mathcal{L}_0[1;l] \\ \vdots \\ \mathcal{L}_0[i;l] \\ \vdots \\ \mathcal{L}_0[m;l] \end{array}$$

Fig. 1. $\mathbf{A}'$: Arrangement of victim's random oracles

Now $\mathbf{A}''$ (resp. $\mathbf{A}'$) contain all output (resp. input) delinearizing random oracles $\alpha_{i;(l,k)}$ with $i \in I''$ (resp. $i \in I'$), so

$$\begin{aligned} \mathbf{A}'[P,(l,k)] &= \zeta'_{(l,k)} = \sum_{i \in I'} \zeta_{i;l} \alpha_{i;(l,k)}, \\ \mathbf{A}'[(i;l),(l,k)] &= \alpha_{i;(l,k)}, \quad \text{and} \\ \mathbf{A}'[(i;x),(y,k)] &= 0 \quad \text{if } x \neq y \\ &\text{while} \\ \mathbf{A}''[L,(y,k)] &= \sum_{j \in I''} \alpha_{j;y} T_{j;y}[L] \quad \text{for } L \in \mathcal{L}_1 \cup ([m] \times I'). \end{aligned}$$

So $\mathcal{A}$ places non-zero entries anywhere they like in $\mathbf{A}''$. Yet, if $\mathcal{A}$ touches anything in a column (l,k) then they rerandomize the random oracles $\alpha_{i;(l,k)}$ for $i \in I$ in the same column, including both the output ones with $i \in I''$ in $\mathbf{A}''$ and the input ones with $i \in I'$ in the jagged diagonal matrix $\mathbf{A}'$. We caution however that our adversary chooses the degrees of freedom for their rerandomization, from between 1 and m, even if their rerandomization touches every row. In particular, our adversary might introduce dependencies between random oracles in different columns.

We employ elementary row operations to transform $\mathbf{A}_0$ into a matrix $\mathbf{A}_{q_s}$ that resembles $\mathbf{A}'$, in that all non-zero elements live on two jagged diagonals consisting of unique random values, independent from others in their own column. In doing so, we never impact the solutions $\boldsymbol{\beta}_1 \oplus \boldsymbol{\beta}_2$ because elementary row operations act by left multiplication by (invertible) elementary matrices.

As a handy picture, consider, for distinct $x, y \in [q_s]$ and $i \in I'$, both the two columns (y,k) for $k = 1, 2$ of $\mathbf{A}_0$, associated to the yth sessions, as well as the row of $\mathbf{A}_0$ associated to the input $T_{i;x}$. We define $\zeta''_{(x,k)} = \sum_{j \in I''} \alpha_{j;(y,k)} T_{j;(y,k)}[P]$ as an $\mathbf{A}''$ analog of $\zeta'_{(x,k)}$ too.

We recall that Gaussian elimination operates by zeroing whole columns (resp. rows) using row (resp. column) operations before moving on to other columns (resp. rows) so that zeroed columns (resp. row) do not interfear with row (resp. column) operations processed later. As a convenient notation, if M is a matrix then we define $M[\![\text{rows}, \text{columns}]\!]$ to be the rows-by-columns submatrix

$$\mathbf{A}_0 = \left(\begin{array}{cccc|cccc|c} & (x,1) & & (y,1) & & (x,2) & & (y,2) & \\ & \zeta'_{(x,1)} + \zeta''_{(x,1)} & \cdots & \zeta'_{(y,1)} + \zeta''_{(y,1)} & & \zeta'_{(x,1)} + \zeta''_{(y,1)} & \cdots & \zeta'_{(y,2)} + \zeta''_{(y,2)} & P \\ \ddots & & & & \ddots & & & & \\ & \alpha_{i;(x,1)} + \mathbf{A}''[\cdot\cdot] & & 0 + \mathbf{A}''[\cdot\cdot] & & \alpha_{i;(x,2)} + \mathbf{A}''[\cdot\cdot] & & 0 + \mathbf{A}''[\cdot\cdot] & (i,x) \\ & & \ddots & & & & \ddots & & \\ & 0 + \mathbf{A}''[\cdot\cdot] & & \alpha_{1;(y,1)} + \mathbf{A}''[\cdot\cdot] & & 0 + \mathbf{A}''[\cdot\cdot] & & \alpha_{1;(y,2)} + \mathbf{A}''[\cdot\cdot] & (1,y) \\ & 0 + \mathbf{A}''[\cdot\cdot] & & \alpha_{2;(y,1)} + \mathbf{A}''[\cdot\cdot] & & 0 + \mathbf{A}''[\cdot\cdot] & & \alpha_{2;(y,2)} + \mathbf{A}''[\cdot\cdot] & (2,y) \\ & & & \ddots & & & & \ddots & \end{array}\right)$$

Fig. 2. $\mathbf{A}_0$: Initial arrangement of random oracles

consisting of the intersections of the specified rows and columns sets. If one views the matrix M as a map from index pairs to $\mathbb{F}_p$ then this describes the restriction to the map rows $\times$ columns.

Definition 11. We say a session index $y \in [q_s]$ is *eliminable* in $\mathbf{A}_x$ if our $m+1$-by-2 submatrix

$$\mathbf{A}_x \left[\!\left[\{0\} \cup \{(i;y)\}_{i \in I'}, (y) \oplus [2] \right]\!\right] = \Big(\mathbf{A}_x[j,(y,k)] \Big)_{\substack{j \in [m] \\ k=1,2}}$$

has all pairwise linearly independent rows. In other words, we say y is eliminable in $\mathbf{A}_x$ if any 2-by-2 submatrix under session index y that should be entirely non-zero in $\mathbf{A}'$ still has non-zero determinant in $\mathbf{A}_x$.

Lemma 12. *Assuming $m \geq 2$, then any session index y is eliminable in $\mathbf{A}_0$, except with probability $\leq m(m+1)q_s q_y/\sqrt{p}$ where $q_y \leq q_h$ denotes the number of queries $\mathcal{A}$ made in column y.*

Proof. If $i \in I'$ then we consider the 1-by-2 row $X_i := \mathbf{A}_0 \left[\!\left[(i;y)\,,\, y \oplus [2] \right]\!\right]$, which takes the form $X_i = [\alpha_{i;(y,1)}, \alpha_{i;(y,2)}] + \mathbf{A}''[\cdot\cdot]$. For $k = 1,2$, these $\alpha_{i;(y,k)}$ are defined by random oracles that make them statistically independent from their accompanying $\mathbf{A}''[\cdot\cdot]$ terms, and anything else in its session (y,k). It follows that if $j \in I'$ satisfies $j \neq i$ then all pairs of 1-dimensional subspaces $\mathbb{F}_p X_i$ and $\mathbb{F}_p X_j$ are equally likely, even if one session is fixed. $\mathcal{A}$ selects among q_y such session choices however. It follows that any two such rows are linearly independent from one another except with probability $\leq \frac{q_y}{\sqrt{p}}$.

Also, our zeroth aka Pth row $X_0 := \mathbf{A}_0 \left[0, (y) \oplus [2]\right]$ has the form $X_0 = [\zeta'_{(y,1)}, \zeta'_{(y,2)}] + \mathbf{A}''[\cdot\cdot]$. As $m \geq 2$, again if we select $i \neq j$ then X_0 has a summand $\zeta_{j;y}\alpha_{j;(y,k)}$ that is statistically independent from the $\alpha_{i;(y,k)}$ and $\alpha_{i'';(y,k)}$ with $i'' \in I''$ summands of X_i. Again $\mathcal{A}$ selects among q_y such choices. It again follows that X_0 is linearly independent from X_i, except with probability $\leq \frac{q_y}{\sqrt{p}}$.

We deduce the result because $m(m+1)$ such pairwise experiments occur in each of q_s jagged bands of m rows determined by x. □

We use the eliminable property inductively to simultaneously jagged diagonalize both sides of the augmented matrix $\mathbf{A}_0$.

Definition 12. We define *jagged augmented Gaussian elimination*: We begin with $\mathbf{A}_0$ defined as above. At the yth stage, we define $\mathbf{A}_y$ from $\mathbf{A}_{y-1}$ as follows. We apply row operations to zero the $m(q_s-1)+1$ positions in the two columns (y,k) for $k=1,2$ that lie off the jagged diagonal. So, for each $x \in [q_s]$ with $x \neq y$ and each $i \in I'$, we subtract from row $(i;x)$ appropriate multiples of the m rows $(i';y)$ for any $i' \in I'$, and of the zeroth aka P row.

$$\mathbf{A}_y[(i;x),\cdot] = \mathbf{A}_{y-1}[(i;x),\cdot] - u_{y,(i;x),P}\mathbf{A}_{y-1}[P,\cdot] - \sum_{i' \in I'} u_{y,(i;x),i'}\mathbf{A}_{y-1}[(i';y),\cdot]$$

After this, we conclude the yth stage by zeroing our zeroth aka Pth row in these two columns too, which we achieve by subtracting appropriate multiples of only the m rows $(i';y)$ for any $i' \in I'$.

$$\mathbf{A}_y[P,\cdot] = \mathbf{A}_{y-1}[P,\cdot] - \sum_{i' \in I',} v_{y,i'}\mathbf{A}_{y-1}[(i';y),\cdot]$$

We safely ignore the rows $L \in \mathcal{L}_1$ with $L \neq P$ here.

We first observe that jagged augmented Gaussian elimination works, assuming eliminablity.

Lemma 13. *If session index y is eliminable in $\mathbf{A}_{y-1}$, then for each row (i,x) with $x \neq y$, there is at least an $m-1$ dimensional subspace $U_{y,(i,x)}$ of choices of $(u_{y,(i;x),i'})_{i' \in \{P\} \cup I'}$s that complete the yth stage of jagged augmented Gaussian elimination, meaning $\mathbf{A}_y[(i;x),(y,k)] = 0$.*

Proof. Assume $m \geq 1$ since otherwise the statment is vacuous. For each $k=1,2$, we have homogeneous linear constraints $\mathbf{A}_y[(i;x),(y,k)] = 0$ on the $u_{y,(i;x),i'}$ with $i' \in \{P\} \cup I'$. As their solutions have codimension one, and they lie in general position, their intersection has codimension two, which gives the desired vector space of $(u_{y,(i;x),i'})_{i' \in \{P\} \cup I'}$s choices. □

An adversary could apply Wagner's k-sum algorithm [18] or [2] to the vector space $\mathbb{F}_p^d$ of course. They randomly sample q_h' ordered bases for $\mathbb{F}_p^d$ and then find q_s' such that the sum of the ith basis vectors yields a dependent set. We observe $q_s' = O(pd)$ yields a polynomial time algorithm using [2], but stronger results sound likely.

We next show that jagged augmented Gaussian elimination actually preserves eliminablity. We first address dependencies only among the $(i;(x,k))$ rows using the extra degree of freedom provided by the zeroth aka Pth row in Lemma 13.

Hypothesis 14. Assume $m \geq 2$. Also assume the session indices x and y are eliminable in $\mathbf{A}_{y-1}$ and $x > y$.

Lemma 15. *Assuming Hypothesis 14, there is a generic subvariety $U_{x,y}$ of $\bigoplus_{i \in I'} U_{y,(i,x)}$ that leaves x potentially eliminable in $\mathbf{A}_y$ in that the m-by-2 submatrix $\mathbf{A}_y[\![(i;l)_{i \in I'}, (l) \oplus [2]]\!]$ has all pairwise linearly independent rows.*

Proof. We express our constraint that x remain potentially eliminable in $\mathbf{A}_y$ by the $\binom{m}{2}$ quadratic inequalities that the pairs of rows yield 2-by-2 matrices with non-zero determinants.

We know little about $\mathbf{A}_{y-1}$ terms off the jagged diagonal, but in row $(i;x)$ of $\mathbf{A}_y$ they only appear multiplied by some $u_{y,(i;x),i'}$ with $i' \in \{P\} \cup I'$.

For each $i \in I'$, we obtained from Lemma 13 the $m-1$ dimensional subspace $U_{y,(i,x)}$ of choices for $(u_{y,(i;x),i'})_{i' \in \{P\} \cup I'}$s that complete the yth stage, meaning $\mathbf{A}_y[(i;x),(y,k)] = 0$.

We never repeat a session within a full game, so for $k = 1, 2$ there exist linear maps $\delta_{i,k} : U_{y,(i,x)} \to \mathbb{F}_p$ given by $(u_{y,(i;x),i'})_{i' \in \{P\} \cup I'} \mapsto (\mathbf{A}_y - \mathbf{A}_{y-1})[(i;x),(x,k)]$, not just the affine maps we'd face if we could repeat sessions. So either $\delta_{i,k} \equiv 0$ or else $\delta_{i,k}$ has range $\mathbb{F}_p$.

We next have the linear map $\delta_i = \delta_{i,1} \oplus_{\text{row}} \delta_{i,2}$ to 1-by-2 rows whose range has dimension 0 or 1 in one of four "shapes". For $i, j \in I'$ with $i \neq j$, we have the linear map $\delta_i \oplus_{\text{col}} \delta_j$ that equals the 2-by-2 matrix $(\mathbf{A}_y - \mathbf{A}_{y-1}) [\![\{(i;x),(j;x)\}, (x) \oplus [2]]\!]$. We have several cases for the ranges of δ_i, but in all cases we deduce from x being eliminable in $\mathbf{A}_{y-1}$ the independence of the two rows of $\mathbf{A}_y [\![\{(i;x),(j;x)\}, (x) \oplus [2]]\!]$ for almost all $U_{y,(i,x)}$, as desired. □

If $m > 2$ then we could actually finish here by taking $\zeta_{i;l} = 0$ and ignoring the zeroth aka Pth row above. We instead used the zeroth aka Pth row above for the $m = 2$ case, so we must now zero zeroth aka Pth row in session index y too.

We loose the extra degree of freedom from Lemma 13 however, so perhaps $|V_x| = 1$ if $m = 2$.

Lemma 16. *Assuming Hypothesis 14, there is a non-empty but only $m-2$ dimensional subspace V_x of choices for $(v_{y,i'})_{i' \in I'}$ such that $\mathbf{A}_y [\![P, (y) \oplus [2]]\!] = 0$.*

Proof. Immediate since y is eliminable in $\mathbf{A}_{y-1}$, □

We analyze this case using our "Pedersen perturbation" of the zeroth aka Pth row, so as to address this $|V_x| = 1$ case.

Lemma 17. *Assuming Hypothesis 14 then x is eliminable in $\mathbf{A}_y$, except with probability $\leq \frac{m+1}{p}$.*

Proof. After applying Lemma 15, we only need the two rows of $\mathbf{A}_y [\![\{P\} \cup \{(i,x)\}, (x) \oplus [2]]\!]$ to be linearly independent for each $i \in I'$. We encounter trouble if $m = 2$ since only one unique linear combination of rows (i,x) with $i \in I'$ works in Lemma 16.

We could explore $U_{x,y}$ further in the cases where $\delta_i \not\equiv 0$ in Lemma 15, but maybe $\delta_i \equiv 0$ here. If some $z \leq y$ had its $\delta_i : U_{z,(i,x)} \to \mathbb{F}_p^2$ associated to $U_{x,z}$ satisfy $\delta_i \not\equiv 0$ then at stage z we selected $(u_{z,(i;x),i'})_{i' \in \{P\} \cup I'}$ randomly from $U_{x,z}$, which randomized our line $\mathbf{A}_y [\![(i,x), (x) \oplus [2]]\!]$. It follows that linear dependence occurs with odds $\leq \frac{1}{p}$, as desired.

Consider the remaining case: For each $z \leq y$ the $\delta_i : U_{z,(i,x)} \to \mathbb{F}_p^2$ associated to $U_{x,z}$ satisfies $\delta_i \equiv 0$. In this case, we have not yet performed row operations on $(i;x)$ and $\mathbf{A}_z[(i;x),(z,k)] = 0$ for $z \leq y$ and $k = 1, 2$. We have performed row operations on the zeroth aka Pth row however, so Lemma 12 does not suffice. We know from $\mathbf{A}_z[(i;x),(z,k)] = 0$ however that $\zeta_{i;x}$s never appear in $\mathbf{A}_0[0,(z,k)]$. Also our $\zeta'_{(x,k)} = \sum_{j \in I'} \zeta_{i;x} \alpha_{i;(x,k)}$ with $k = 1, 2$ are perfectly hidden from $\mathcal{A}$. We conclude that linear dependence occurs with odds $\leq \frac{1}{p}$, as desired. □

Proposition 18. *If $m \geq 2$ then $\phi_{\mathbb{A}}$ is politely injective, except with probability $< (m+1)^2 q_s^2 q_h / \sqrt{p}$.*

Proof. As discussed above, we consider two $(\boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \mathbf{t}''_k)_{k=1,2}$ such that $\Phi_{\boldsymbol{\omega}_1, \mathbf{t}''_1}(\boldsymbol{\beta}_1) = \Phi_{\boldsymbol{\omega}_2, \mathbf{t}''_2}(\boldsymbol{\beta}_2)$ (†) holds. We view (†) as an equation in the $\boldsymbol{\beta}_k$, which we write as an augmented matrix $\mathbf{A}_0$.

We take all sessions eliminable in $\mathbf{A}_0$, which occurs by Lemma 12, except with probability $< m(m+1) q_s^2 q_h / \sqrt{p}$. We now run jagged augmented Gaussian elimination. At the yth stage, we assume inductively for $x \geq y$ that the xth session index is eliminable in $\mathbf{A}_{y-1}$, so all remaining sessions stay eliminable in $\mathbf{A}_y$ by Lemma 17, except with probability $\leq \frac{m+1}{p}$.

We conclude that $\mathbf{A}_{q_s}$ has our desired jagged diagonal form, except with odds $< (m+1)^2 q_s^2 q_h / \sqrt{p}$. We kept pairwise independence among the rows of the m-by-2 matrices $\mathbf{A}_{q_s} [\![[m] \oplus (x), (x) \oplus [2]]\!]$

because we never touched their rows after the xth stage. As $m \geq 2$, we conclude that for session index $l \in [q_s]$ either $\beta_{l,1} = 0 = \beta_{l,2}$, or else $(\boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \mathbf{t}''_k)$ have the same lth session for both $k = 1, 2$, as desired. □

4.4 Polite injectivity as commitment

We now prove that $\mathcal{B}$ given in Definition 8 acts as a reduction with high probability.

Theorem 19. *In AGM plus ROM, if $m \geq 2$ then m, q_s-entwined ROS reduces to mq_s-DLR with its success probability loss of only $(m+1)^2 q_s^2 q_h^2/\sqrt{p}$, and hence it reduces to DLOG.*

Proof. There is a probability space $\mathcal{P}$ of runs of our algorithm $\mathcal{B}$ from Definition 8 given by running $\mathcal{B}$ on uniformly distributed DLR challenges.

We consider the event where $\mathcal{B}$ fails: We let $\hat{\boldsymbol{\omega}}$, $\hat{\boldsymbol{\beta}}$, and $\hat{\mathbf{t}} = (\hat{\mathbf{t}'}, \hat{\mathbf{t}''})$ denote the final responce of $\mathcal{A}$, so our final RHS group representative in $\mathbb{F}_p^{\mathcal{L}}$ becomes $Y = \phi(\hat{\boldsymbol{\beta}}, \hat{\mathbf{t}}) = \sum_{L \in \mathcal{L}_1} \hat{\beta}_L L +_{\mathbb{G}} \sum_{l=1}^{q_s} \beta_l \hat{T}_l$. We know from Lemmas 9 that $\mathcal{A}$ never queried $\sum_{\mathbb{G}} Y$ with another representative besides Y.

We assume that $\phi_{\mathbb{A}}$ is politely injective because according to Lemma 18 this holds except with negligible probability $< (m+1)^2 q_s^2 q_h/p$. So $Z := \phi_{\mathbb{A}}^{-1}(Y)$ contains only one meaningful answer: If Z has two distinct lth sessions then $\beta_l = 0$ for all lth sessions, so they play no role in either the RHS or LHS of (‡). As H is a random oracle, $\mathcal{A}$ might still satisfy (‡) using this Y, but only with odds $1/p$.

It follows that, among runs where $\mathcal{B}$ fails, $\mathcal{A}$ has odds of success $\leq (m+1)^2 q_s^2 q_h/p$ per H query.

As $\mathcal{A}$ makes strictly fewer than q_h queries to H, we deduce that $\mathcal{A}$ has odds of success $< (m+1)^2 q_s^2 q_h^2/p$. [2] If $\mathcal{A}$ fails with odds ϵ then we deduce that $\mathcal{B}$ provides a reduction that fails with odds $< \epsilon + (m+1)^2 q_s^2 q_h^2/p$, as desired. □

5 Security of DWMS

We shall adapt the multi-signature security game to our 2-DWMS protocol, apply the entwined ROS result, and then prove its equivalence to an OMDL challenger form below.

Game 1 (2-DWMS Security)**.** An adversary $\mathcal{A} = \mathcal{A}^{H,H_2,Q_1,Q_2}$ for 2-DWMS is an algorithm that takes a public key Y, some random coins ρ, and four oracles based on §1, the Schnorr challenge H and delinearization H_1 random oracles, as well as oracles Q_1 and Q_2 for two DWMS-Sign rounds.

We say $\mathcal{A}$ breaks 2-DWMS if they win the game: Sample $(Y, y) \longleftarrow \mathsf{KeyGen}$. Initialize $\mathtt{db} = \emptyset$. Run $(K_*, \omega_*, \sigma_*) \longleftarrow \mathcal{A}^{H,H_1,Q_1,Q_2}(Y; \rho)$. Return fail unless $Y \in K_*$ and $|\mathtt{db}_H| \leq q_h$ and $|\mathtt{db}_{Q_1}| \leq q_s$. Return that $\mathcal{A}$ wins if $\mathsf{Verify}((\omega_*, X_*), \sigma_*)$ passes where $X_* = \mathsf{TweakKey}(K_*)$.

In this, we implement H and H_1 as random oracles that return a fresh random value whenever queried with a fresh input, but memoize their results by recording a partial mapping in $\mathtt{db}_H$ and $\mathtt{db}_{H_1}$, respectively.

We implement $Q_1(l)$ naively with DWMS-Sign round 1: Sample $r_{(1,1);l}, r_{(1,2);l} \longleftarrow \mathbb{F}_p$. Store $l \mapsto (r_{(1,1);l}, r_{(1,2);l}$ in $\mathtt{db}_{Q_1}$. Compute $T_{(1,j);l} = r_{(1,j);l}$ for $j = 1, 2$. Return $(T_{(1;1);l}, T_{(1;2);l})$.

We implement $Q_2(l, K_l, \omega_l, (T_{(i,1);l}, T_{(i,2);l})_{i \neq 1})$ naively with DWMS-Sign round 2: Abort if $Y \notin K_l$ or $(l, \cdot) \notin \mathtt{db}_{Q_1}$. Compute $(X_l, x_1) = \mathsf{TweakKey}(K_l, Y, y)$. Build $\mathtt{transcript}_l$ using K_l, ω_l, $(T_{(i;1);l}, T_{(i,2);l})_{i \neq 1}$ and $T_{(1,1);l}, T_{(1,2);l}$, recomputed from $\mathtt{db}_{Q_1}[l]$. Compute the delinearization scalars

[2] We expect $\mathcal{A}$ queries H sessions evenly, so $q_y \approx q_h/q_s$, and our final bound resembles $< (m+1)^2 q_h^2/p$.

$\alpha_{(i,j);l} = H_1(\texttt{transcript}_l)[i,j]$ for $j = 1, 2$, the delinearized shared witness $S_l = \sum_{i=1}^{n} \alpha_{(i,1);l} T_{(i,1);l} + \alpha_{(i,2);l} T_{(i,2);l}$, and challenge $c_l = H((\omega_l, X_l), S_l)$. Return our partial signature $s_1 = \alpha_{i,1} r_{i,1} + \alpha_{i,2} r_{i,2} + cx$.

We immediately exclude the $2q_s$-entwined ROS story to obtain the disentwined 2-DWMS security game.

Definition 19. We retain the notation from the 2-DWMS security game. Assume AGM so instead of $\mathbb{G}$ we work in $\mathbb{F}_p^{\mathcal{L}}$ for some $\mathcal{L} = \mathcal{L}_0 \cup \mathcal{L}_1$ Suppose $\mathcal{L}_1 = \{P, Y\}$ where Y is the victim's public key.

We define a tweaked Schnorr random oracle

$$H_0((\omega, X), S) = a_{K,Y} H((\omega, X), S) - S[Y],$$

using the $a_{K,Y}$ from $\mathsf{TweakKey}(K, Y)$ and the Y component $S[Y]$ of $XS \in \mathbb{F}_p^{\mathcal{L}}$, and assuming $X = \mathsf{TweakKey}(K)$ for some K that contains Y.

If $|\mathcal{L}_0| \leq q_s$ then there is an algorithm that finds linear dependencies among the $S_\ell[\![\mathcal{L}_0]\!]$ for $\ell \in [q_s] \cup \{*\}$ and if successful computes β_l for $l \in \mathcal{L}_1 \cup [q_s]$ such that

$$S_* = \beta_P P + \beta_Y Y + \sum_{l=1}^{q_s} \beta_l S_l \tag{2}$$

We say a 2-DWMS security game (Game 1) run is $Y, \mathcal{L}$-*disentwined* if we abort whenever

$$\sum_{l=1}^{q_s} \beta_l H_0(\omega_l, S_l) = H_0\left(\omega_0, \beta_P P + \beta_Y Y + \sum_{l=1}^{q_s} \beta_l S_l\right) \tag{3}$$

Proposition 20. *Assuming AGM plus ROM in Game 1 then any given run is $Y, \mathcal{L}$-disentwined except with probability $\leq (m+1)^2 q_s^3 q_h^2 / \sqrt{p}$. So $\mathcal{A}$ cannot distinguish whether we $Y, \mathcal{L}$-disentwine Game 1.*

Proof. We first observe that H_0 is a random oracle because any changes that alter either $a_{K,Y}$ or $S[Y]$ also alters the actual random oracle $H((\omega, X), S)$. It follows from Theorem 19 that $\mathcal{A}$ cannot distinguish whether we $Y, \mathcal{L}$-disentwined Game 1, In this, we caution that Theorem 19 could internally work with a different AGM basis vector set $\mathcal{L}$. □

We next alter the disentwined 2-DWMS security game to transform $\mathcal{A}$ into an algorithm usable as a challenger for the *one-more discrete logarithm* (OMDL) problem.

Definition 21 (n-OMDL Problem [7, Definition 2])**.** Let DL be a discrete logarithm oracle taking as input a point $Q \in \mathbb{G}$ and returning $a \in \mathbb{F}_p$ such that $aP = Q$. An algorithm $\mathcal{A}$ is said to (n, t, ϵ)-solve the n-OMDL problem in $\mathbb{G}$ if on input of $n+1$ random points $R_0, \ldots, R_n \in \mathbb{G}$, it runs in time at most t, makes at most n queries to DL, and returns $a_0, \ldots, a_n \in \mathbb{F}_p$ such that $R_i = a_i P$ for all $i \in n+1$ with probability at least ϵ, where the probability is taken over the random draw of $R_0, \ldots, R_n$ and the random coins of $\mathcal{A}$.

In our new game, we accept an OMDL challenge $(Y_0, \ldots, Y_{q_s})$, use Y_0 as the victim's public key Y, use the remaining Y_i to define two $T_{(1;j);l}$ values for $j = 1, 2$, and answer signature requests in Q_2 with the DL oracle.

Game 2 (OMDL challenger from 2-DWMS)**.** We alter the $Y_0, \mathcal{L}$-disentwined 2-DWMS security game, meaning Game 1 plus the $Y_0, \mathcal{L}$-disentwined check, played by $\mathcal{A}$ to become an algorithm $\mathcal{A}_{\mathsf{OMDL}}(Y_0, \ldots, Y_{2q_s}; c_1, \ldots, c_{q_h}; \rho)$ that processes OMDL challenges:

We accept a q_s-OMDL challenge $(Y_0, \ldots, Y_{q_s})$ and the new game defines $Y = Y_0$ instead of sampling a key pair with KeyGen.

Our first round multi-signer oracle $Q_2(l)$ samples $\gamma_{l,1}, \gamma_{l,2} \longleftarrow \mathbb{F}_p$ to computes return values $T_{(1;j);l} = \gamma_{l,j} Y_l$ for $j = 1, 2$, where $l = |\mathtt{db}_{Q_1}| \leq q_s$. It also stores $l \mapsto (\gamma_{l,1}, \gamma_{l,2})$ in $\mathtt{db}_{Q_1}$ since it never learns the challengers' secret scalars $(r_{(1,1);l}, r_{(1,2);l})$.

Our second round multi-signer oracle $Q_2(l, K_l, \omega_l, (T_{(i,1)}, T_{(i,2)})_{i \neq 1})$ defines $s_1 \longleftarrow \mathsf{DL}(\alpha_{1,1} T_{1,1} + \alpha_{1,2} T_{1,2} + c a_{K_l, Y_0} Y_0)$ using the a_{K_l, Y_0} in $\mathsf{TweakKey}(K_l, Y_0)$, after fetching $\gamma_{l,j}$ from $\mathtt{db}_{Q_1}[l]$ and recomputing $T_{(1;j);l}$ for $j = 1, 2$. We make Q_2 store its invokations $l \mapsto (K_l, \omega_l, (T_{(i,1)}, T_{(i,2)})_{i \neq 1})$ into a separate $\mathtt{db}_{Q_2}$. Also Q_2 now memoizes its DL queries, meaning it stores $(l, \alpha_{1,1}, \alpha_{1,2}, c) \mapsto s_1$ in $\mathtt{db}_{\mathsf{DL}}$ and checks for this memoized record to prevent wasting DL invokations.

Lemma 22. *$\mathcal{A}$ cannot distinguish between the $Y_0, \mathcal{L}$-disentwined 2-DWMS security game, and Game 2, provided ρ and $\mathcal{C}$ are random.*

Proof. It's always safe to expose our random coins as arguments required to be random. $\mathcal{A}$ cannot distinguish if some valid discrete logarithm comes from $\mathtt{db}_{Q_2}$ or from a fresh DL query of course.

We know the $T_{(1;j);l} = \gamma_{l,j} Y_l$ are uniformly distributed because our $\gamma_{l,j}$ are uniformly distributed. It follows that $\mathcal{A}$ cannot distinguish the pre-witness distributions.

We deduce further that s_1 are uniformly distributed because our pre-witnesses are uniformly distributed and the DL oracle behaves correctly. We avoid invoking DL if we find some valid signature in $\mathtt{db}_{Q_2}$ already, but this does not change the result. It follows $\mathcal{A}$ cannot distinguish the signature distributions either. □

Theorem 23. *In AGM plus ROM, if $\mathcal{A}$ provides its public keys before learning any pre-witnesses, ala KOSK, then OMDL reduces to 2-DWMS.*

Proof. Set $\mathcal{L}_0 = \{Y_1, \ldots, Y_{q_s}\}$ and $\mathcal{L} = \mathcal{L}_0 \cup \{P, Y_0\}$. We apply Lemma 22 to the $Y_0, \mathcal{L}$-disentwined 2-DWMS security game from Proposition 20, yielding a potential OMDL challenger.

We compute from $\mathtt{db}_{Q_2}$ the equations $S_l = \sum_{i \in [n], j=1,2} \alpha_{(i,j);l} T_{(i,j);l}$. We discover from $\mathtt{db}_{\mathsf{DL}}$ exactly q_s equations

$$s_{1;l} P = c_l a_{K_l, Y_0} Y_0 + \sum_{j=1,2} \alpha_{(1,j);l} T_{(1,j);l} \quad \text{for } l \in [q_s], \tag{4}$$

where $c_l = H((\omega_l, X_l), S_l)$. We complete the shared witness on the RHS of (4) like

$$s_{1;l} P + \sum_{1 < i \leq n} \sum_{j=1,2} \alpha_{(i,j);l} T_{(i,j);l} = c_l a_{K_l, Y_0} Y_0 + S_l. \tag{5}$$

We also learn the forgery equation $s_* P = c_* X_* + S_*$, where $c_* = H((\omega_*, X_*), S_*)$ and $X_* = \sum_{Z \in K_*} a_{K_*, Z} Z$ and $Y_0 \in K_*$. So

$$s_* P - c_* \sum_{Z \in K_* \setminus \{Y_0\}} a_{K_*, Z} Z = c_* a_{K_*, Y_0} Y_0 + S_* \tag{6}$$

We know β_l for $l \in \mathcal{L}_1 \cup [q_s]$ such that (2) holds by our $Y_0, \mathcal{L}$-disentwined check. We subtract the lth equation in (5) multiplied by β_l from (6) for $l \in [q_s]$. As (3) fails by our $Y_0, \mathcal{L}$-disentwined check, we have a nonzero value

$$\lambda = c_* a_{K_*,Y_0} - S_*[Y_0] - \sum_{l \in [q_s]} \beta_l \left(c_l a_{K_l,Y_0} - S_l[Y_0] \right) \tag{7}$$

such that

$$\begin{aligned} \lambda Y_0 = \left(s_* - \sum_{l \in [q_s]} \beta_l s_l \right) P - c_* \sum_{Z \in K_* \setminus \{Y_0\}} a_{K_*,Z} Z \\ - \sum_{l \in [q_s]} \beta_l \sum_{1 < i \le n} \sum_{j=1,2} \alpha_{(i,j);l} T_{(i,j);l} [\![\{P\} \cup \mathcal{L}_0]\!]. \end{aligned} \tag{8}$$

In (8), we have a non-zero multiple of Y_0 on the LHS while our RHS consists of multiples of P and the Y_l for $l \in [q_s]$, assuming by KOSK that $Z[P]\, P = Z$ for $Z \in K_* \setminus \{Y_0\}$. We now substitute this Y_0 back into the system (4).

We know the forgery equation is linearly independent from the (4) except with negligible probability because otherwise it would not be a forgery. We might clear some Y_l terms during this substitution, but as (4) has rank q_s we obtain a solvable system of equations regardless. We now solve for the Y_l and finally Y_0 to obtain the OMDL result. □

Interestingly, the OMDL problem is actually hard in the generic group model (GGM), according to [6, Table 2 or §5], so our results below yield hardness of 2-DWMS in GGM.

6 Conclusion

We introduced in §1 and proved security for (2-)DWMS in Theorem 23, which provides a simple and lightweight two-round multi-signer Schnorr protocol.

In so doing, we introduced the entwined ROS problem in Definition 2 and proved its hardness under AGM plus ROM plus OMDL in Theorem 19. We believe entwined ROS provides an effective tool with which to address composition of multi-signer protocols and Schnorr variants, especially adaptor signatures, implicit certificates, and perhaps help with blind signatures.

We showed 1-DWMS to be insecure. We also showed efficient algorithms for the "1.5"-entwined ROS problem associated to "1.5"-DWMS aka FROST in Proposition 4, which excludes FROST form our security arguments. We further conjecture that FROST itself is insecure, at least when composed with Schnorr variants, like some adaptor signatures or implicit certificates, and maybe future blind signature schemes.

As discussed in §2.2, we think multi-signer Schnorr blind signatures emerge from this work as an extremely interesting open problem.

References

1. Ali Bagherzandi and Stanisław Jarecki. Multisignatures using proofs of secret key possession, as secure as the diffie-hellman problem. In *International Conference on Security and Cryptography for Networks*, pages 218–235. Springer, 2008.

2. Fabrice Benhamouda, Tancrède Lepoint, Michele Orrù, and Mariana Raykova. On the (in)security of ros. Cryptology ePrint Archive, Report 2020/945, 2020. https://eprint.iacr.org/2020/945.
3. Dan Boneh, Manu Drijvers, and Gregory Neven. *Compact Multi-signatures for Smaller Blockchains: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part II*, pages 435–464. 01 2018.
4. Jeffrey Burdges. schnorrkel. https://github.com/w3f/schnorrkel/commit/fa6c35f832, January 2020.
5. Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach. Does fiat-shamir require a cryptographic hash function? Cryptology ePrint Archive, Report 2020/915, 2020. https://eprint.iacr.org/2020/915.
6. Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. *Non-Uniform Bounds in the Random-Permutation, Ideal-Cipher, and Generic-Group Models: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part I*, pages 693–721. Lecture Notes in Computer Science. Springer, 01 2018. https://eprint.iacr.org/2018/226.
7. Manu Drijvers, Kasra Edalatnejad, Bryan Ford, Eike Kiltz, Julian Loss, Gregory Neven, and Igors Stepanovs. On the security of two-round multi-signatures. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1084–1101. IEEE, 2019.
8. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In *Advances in Cryptology – CRYPTO 2018*, volume 10992 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2018.
9. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind schnorr signatures and signed elgamal encryption in the algebraic group model. In *39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings*, volume 12105 of *Lecture Notes in Computer Science*. Springer, 2020.
10. Chelsea Komlo and Ian Goldberg. Frost: Flexible round-optimized schnorr threshold signatures. 2020.
11. Changshe Ma, Jian Weng, Yingjiu Li, and Robert Deng. Efficient discrete logarithm based multi-signature scheme in the plain public key model. *Designs, Codes and Cryptography*, 54(2):121–133, 2010.
12. Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple schnorr multi-signatures with applications to bitcoin. *Designs, Codes and Cryptography*, 87(9):2139–2164, 2019.
13. Jonas Nick, Tim Ruffing, Yannick Seurin, and Pieter Wuille. Musig-dn: Schnorr multi-signatures with verifiably deterministic nonces. Cryptology ePrint Archive, Report 2020/1057, 2020. https://eprint.iacr.org/2020/1057.
14. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. In *In Journal of Cryptology 13*, pages 361–396, 2000.
15. Claus Peter Schnorr. Security of blind discrete log signatures against interactive attacks. In *ICICS 2001, LNCS 2229*, pages 1–12. Springer-Verlag, 2001.
16. Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology – EUROCRYPT 97*, pages 256–266, 1997.
17. Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. Keeping authorities" honest or bust" with decentralized witness cosigning. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 526–545. Ieee, 2016.
18. David A. Wagner. A generalized birthday problem. In *CRYPTO*, 2002.