Isogeny-based Group Signatures and Accountable Ring Signatures in QROM

Kai-Min Chung[1], Yao-Ching Hsieh[2], Mi-Ying (Miryam) Huang[3], Yu-Hsuan Huang[1,4], Tanja Lange[5], and Bo-Yin Yang[1]

[1] Academia Sinica, Taiwan
[2] University of Washington, United States
[3] University of Southern California, United States
[4] Centrum Wiskunde & Informatica, The Netherlands
[5] Eindhoven University of Technology, The Netherlands
kmchung@iis.sinica.edu.tw, ychsieh@cs.washington.edu, miying.huang@usc.edu, Yu-Hsuan.Huang@cwi.nl, byyang@iis.sinica.edu.tw, tanja@hyperelliptic.org

Abstract. We provide the first isogeny-based group signature (GS) and accountable ring signature (ARS) that are provably secure in the quantum random oracle model (QROM). We do so by building an intermediate primitive called openable sigma protocol and show that every such protocol gives rise to a secure ARS and GS. Additionally, the QROM security is guaranteed if the perfect unique-response property is satisfied. Our design, with the underlying protocol satisfying this essential unique-response property, is sophisticatedly crafted for QROM security. From there, with clever twists to available proving techniques, we obtain the first isogeny-based ARS and GS that are proven QROM-secure.
Concurrently, an efficient construction was proposed by Beullens et al. (Eurocrypt 2022), but is only proven secure in the classical random oracle model (ROM). Our proposal seeks stronger QROM security, although it is less efficient due to the signature size quadratically scaling with the ring/group size.

Keywords: isogeny, group signature, accountable ring signature, quantum random oracle model

1 Introduction

Group signatures and accountable ring signature *Group signatures* (GS), first proposed by Chaum and van Heyst [14], are signature schemes that permit signing by a *group*, a set of players chosen by a prescribed *group manager*. Each

Authors list in alphabetical order; see https://www.ams.org/profession/leaders/culture/CultureStatement04.pdf.

player can generate publicly verifiable signatures on behalf of the group while keeping himself anonymous to everyone except the group manager. The group manager has the authority to *open*, i.e., to reveal the signer's identity from a signature with its *master secret key*. Afterward, there have been numerous works devoted to group signatures. Many of them aimed to give refinements and extensions to the primitive [3, 11]. An important line of research on group signatures studies variants with *dynamic groups*. In contrast to the original formulation where only *static groups* are supported [2, 14], a dynamic group signature allows a group to be updated after the setup stage. The notion of *partially dynamic* group signatures was formulated in [3, 26], where parties can join a group but cannot be removed. However, *Accountable ring signatures* (ARS), first proposed by Xu and Yung [42], provides the "dynamic property for groups" in a different aspect. ARS, while having a "ring signature" [35] within its name, can also be viewed as a variant of group signatures where groups are *fully dynamic* but *not authenticated*. In an ARS scheme, the manager no longer controls the group. Instead, a signer can freely decide which master public key to use and which group to sign for, and the corresponding master secret key can then open its identity. Though seemingly incomparable to a standard group signature, an ARS scheme can, in fact, trivially imply a group signature scheme simply by fixing the group at the setup stage. Later, Bootle, Cerulli, Chaidos, Ghadafi, Groth, and Petit [9] proposed a stringent formulation for ARS and a provable construction based on the DDH assumption. It is further shown in [8] that such a stringent ARS scheme can be generally transformed to a *fully dynamic* group signature scheme.

Signatures with post-quantum assumptions. There has been increasing attention to the importance of post-quantum security for cryptographic primitives. Various attempts emerged to construct group signatures based on cryptographic assumptions that resist quantum attacks. Gordon, Katz, and Vaikuntanathan first gave a group signature construction from lattice-based assumptions [24]. Several constructions of lattice-based group signatures followed this, either for static groups [27, 32], or dynamic groups [29, 30]. There have also been a few attempts at constructing group signatures from other classes of post-quantum assumptions, such as code-based assumptions [21] or hash-based assumptions [1]. However, multiplayer signatures, including ring, accountable ring, and group signatures, are often found in only classical ROM setting. To the best of our knowledge, only a select few lattice-based signatures offer QROM-security [36]. Offering multiplayer signatures with QROM-security from other assumptions, as a result, becomes an intriguing topic.

Signatures from ROM to QROM. Signature construction typically involves a cryptographic hash function, which is then modeled as a random function giving its query access to the considered adversary. This is referred to as the *random oracle model* (ROM), which has been proven successful for security analysis. In light of a full-fledged quantum adversary being able to evaluate the hashes in superposition, one should consider an adversary given quantum query access to the random oracle. Security under this *quantum random oracle model* (QROM) is always considered necessary in post-quantum cryptography. In

terms of isogeny-based cryptography, there have been several primitives proven QROM-secure [6, 22], but there is still a lack of more advanced QROM-secure ones such as ring signatures or group signatures. This leads to the main question of this work:

> *Can we construct QROM-secure group signatures and accountable ring signatures from isogenies?*

1.1 Our results

We construct accountable ring signatures (ARS) from *isogeny-based assumptions* in the quantum random oracle model (QROM). Moreover, since ARS can be easily transformed into group signatures and ring signatures[6] while preserving its QROM security, we also achieve the first provably QROM-secure group signature and ring signature. On top of it, we introduce a primitive for constructing ARS called *openable sigma protocol*, which is simple and fits well with the Fiat-Shamir methodology: Any openable sigma protocol Σ can be securely transformed into an ARS $\mathcal{ARS}^t_\Sigma$ using Fiat-Shamir transformation. Furthermore, we show that a typical requirement for QROM-secure signatures, the *unique-response property*, suffices to provide QROM security to the abovementioned ARS. Note that it is non-trivial to obtain (or get away with) the unique-response property, and we will direct the readers to our technical overview in Section 2 for further details.

Theorem 1. *(Informal) Let Σ be a secure openable sigma protocol. Then $\mathcal{ARS}^t_\Sigma$ is a classically secure ARS. Furthermore, if Σ is perfect unique-response, then $\mathcal{ARS}^t_\Sigma$ is QROM-secure.*

We base our construction on the *decisional CSIDH assumption* (D-CSIDH). From an abstract viewpoint, D-CSIDH is a natural generalization of DDH which is built over the weaker group-action structure. Due to the lack of homomorphic properties in group-action assumptions, it is usually infeasible to transform results obtained from group-based assumptions to those from group-action-based assumptions. Our work demonstrates the possibility of constructing advanced cryptographic primitives with group-action-based assumptions, despite its limited expressiveness.

Concurrent works. Independent and concurrent to our work, [4, 28] also managed to construct isogeny-based group signatures, with the former based on the accountable ring signatures as we do, and the latter based on the so-called collusion-resistant revocable ring signatures. There was coordination in which [4, 28] and we released our preprints simultaneously. At that time they were offering more efficient constructions but only in classical ROM, and we were still working on the QROM analysis. Soon after, our updated version has been able to provide QROM security, whereas neither of [4, 28] does. This is mainly because of

[6] ARS⇒RS is trivial by throwing away the opening functionality.

the following reason. With QROM security borne in mind, our underlying sigma protocol has been designed to be unique-response, which is absent in the construction of [4, 28]. In terms of efficiency, our signature size scales quadratically $O(n^2)$ in the number n of members, which is relatively bigger than their $O(\log n)$ and $O(n \log n)$. In comparing these works, one may regard [4, 28] focusing more on efficiency and ours more on full-fledged quantum security.

1.2 Open problems

Efficiency versus QROM-security. In this work, we construct the first isogeny-based group signatures and accountable ring signatures in QROM. It still falls short in terms of the signature size compared to relevant works. On the other hand, [4, 28] constructed smaller signatures, but with security only proven in classical ROM. It remains open whether there exists a construction that is small and QROM-secure *simultaneously*.

Extra judging functionality. Our setting has premised an honest manager, only to whom the opening result is available. A corrupted manager can thus incriminate any party as the signer of an arbitrary signature. Many previous works on group signatures then provide an extra *judging function* allowing the manager to generate a publicly verifiable proof for its opening results. We offer (in Supplementary B) a weaker version with a simple tweak that will prevent a dishonest manager from incriminating honest non-signers. It remains open to constructing a QROM-secure ARS with a full-fledged judging function.

2 Technical overview

In this overview, we assume some familiarity for sigma protocols and the Fiat-Shamir transformation [23].

2.1 Hurdles to QROM-security

To start off, we explain why one needs to be extra careful when performing security proofs in QROM.

We follow the so-called Fiat-Shamir paradigm. In this paradigm, one often starts with constructing a so-called sigma protocol, where an adversary can make random challenges for the prover to respond. Then, the interaction is removed by substituting the verifier's random challenges with hashes of earlier transcripts.

In proving securities for this kind of construction, two commonly used techniques are *rewinding*, and *reprogramming*. We briefly describe their use-case as follows.

(1) The properties of a sigma protocol Σ can often be lifted to its corresponding Fiat-Shamir signatures, which run almost as Σ, except challenges are computed as the hashes (partly) of to-be-signed messages. An adversary may therefore bias the challenge distribution, by post-selecting the messages depending on previous hash queries. One typically resorts to the reprogramming argument to control such bias.

(2) A common way to argue the underlying sigma protocol indeed proves the knowledge of a certain relation, is by means of running a prover multiple times, and each time rewinding to the point where fresh challenges are about to be made, in order to collect its responses for many different challenges.

The reprogramming argument usually involves reasoning around previous query inputs. For instance, classically one may argue a certain random variable x has not been queried to the random oracle, except with a small probability. However, in the quantum setting where the queries are made in superposition, it is non-trivial to even define the previously queried inputs. It is not possible to copy out the previously queried inputs due to the no-cloning principle, nor to measure them out as it may disturb the running states and be noticed by the adversary. Similar issues occur in the rewinding arguments, where we run the same prover multiple times. Each time, when a response is made, the running state of the prover is disturbed, which then affects later execution as well.

Mitigating the unique-response property?
In [39], Unruh demonstrated how to perform the quantum rewinding argument with an additional so-called collapsing property. The collapsing property is a slight relaxation of the unique-response property, which turns out necessary in the currently available quantum rewinding technique, e.g., [15]. In particular, if the prover can produce two distinct responses for the same challenge, as in [4, 5, 28], then the collapsing property is violated.

A natural question is: for a construction lacking the collapsing property, can we easily "uniquify" their responses? There is a rule of thumb along the line of the so-called Unruh's transformation [18, 39]. The prover commits to one response for each challenge and publishes it in the first message, which the verifier can open and check in the later phase. However, the situation is more complicated, and we need to be more careful when we look into specific constructions and ask if they could be modified and obtain QROM securities.

We note that the previous construction [5] of isogeny-based ring signatures in ROM may also obtain QROM security if we slightly twist the protocol with careful analysis. A twist in our mind is via removing an optimization trick that replaces a list of commitments in the first message to a Merkle-tree root, and after the removal, additionally adopting Unruh's transformation. However, whether or not it works remains to be confirmed, and the authors of [5] did not contemplate such an adaptation in their paper.

As for accountable ring signature and group signature, the following two reasons come as major obstructions for [4, 28] to achieve QROM security:

(1) Within their underlying sigma protocols, a prover is able to compute two distinct responses when the challenge is 0. This violates the collapsing property, which is required in presently available quantum rewinding techniques, e.g., [15,39].

(2) The Unruh's transformation [40] premises a sigma protocol in the plain model and is not directly applicable to [4,28] due to their extra hash commitments involved.

How about a polynomial number of valid responses?
One thing worth mentioning is that there is a variant of Unruh's rewinding tolerating up to polynomially many responses (for the same challenge). However, this variant of rewinding still does not help either because typical constructions involve parallel repetition, which executes the same protocol multiple times independently in order to amplify soundness probability. The number of valid responses will then have an exponential blow-up as the number of repetitions scales, from which the QROM analysis falls apart.

2.2 A technical trailer

Signatures based on isogeny class group action. Stolbunov [38] gave the first attempt toward an isogeny-based signature scheme in his thesis. His scheme applies the Fiat-Shamir transformation [23] to the sigma protocol of Couveignes [17]. While Couveignes' the protocol is structurally similar to the discrete log-based protocol by Chaum and van Heyst [14], its challenge space cannot be extended as in Schnorr's protocol [37]. Parallel repetition is thus necessary for Stolbunov's signature scheme.

Later, following the proposal of an efficient class group action implementation by CSIDH [12], SeaSign [22], and CSI-FiSh [6] separately gave efficient signature constructions based on Stolbunov's approach. One main contribution of their works is that they overcome the lack of canonical representation for elements in the class group $\mathsf{Cl}(\mathcal{O})$. In Stolbunuov's scheme, the signer would reveal rs for $r \xleftarrow{\$} \mathsf{Cl}(\mathcal{O})$ and secret $s \in \mathsf{Cl}(\mathcal{O})$. However, since r and s are represented as element-wise bounded vectors in the CSIDH representation, a naive representation for rs *does not* hide the information of s. To cope with this issue, SeaSign proposed a solution using the Fiat-Shamir with abort technique [31], while CSI-FiSh computes the whole class group structure and its relation lattice for a specific parameter set, CSIDH-512. In this work, we will adopt the latter approach, where we can simply assume canonical representation for elements in $\mathsf{Cl}(\mathcal{O})$.

Recently, Beullens, Katsumata, and Pintore [5] showed how to construct an isogeny-based ring signature with the sigma protocol for an OR-relation. Our work similarly starts with a sigma protocol which additionally supports an opening operation. We want a sigma protocol that takes n statements and a master

public key as inputs, computes a proof for one of the statements and embeds the "identity" of the proved statement into the transcript so that it can be extracted with the master secret key. As the first step, we will discuss how we can embed information for opening the transcript. **Embedding opening information.** In a group signature scheme, the information for the signer's identity must be somehow embedded into the signature so that the master can open it. One natural approach to embed opening information is to encrypt the information with the master public key. Such an approach is proven successful in a few previous works on group signatures [7, 9]. However, since the opening information is now a ciphertext under the master key, a verifier can only check the validity of the ciphertext via homomorphic operations or NIZK. Unfortunately, unlike group-based assumptions, it is not yet known how to achieve such homomorphic property from the weaker *group action structure* given by isogeny-based assumptions. There is also no isogeny-based NIZK construction in the literature. Thus, we will have to develop a structurally more straightforward way to encode our opening information.

In light of this, we construct our opening functionality in a very naive way. For a signature with group/ring size n and a master secret key s_m for opening, we embed the signer identity by one DDH tuple and $n-1$ dummies. Namely, the opening information is in the form

$$\tau = ((r_1E, r_2E, \ldots, r_nE), r_kE_m), \text{ where } r_1, \ldots, r_n \xleftarrow{\$} G \text{ and } E_m = s_mE \ ,$$

which embeds the signer's identity $k \in [n]$ through *position*, and is extractable for the manager holding s_m. Note that such τ keeps all its elements in the form of curves/set elements, hence the verifier can do further group action on τ for consistency checking. This circumvents the previous difficulty, but with the cost of a larger payload.

Openable sigma protocol. To construct a group signature/accountable ring signature scheme through Fiat-Shamir transformation, we first introduce an intermediate primitive called openable sigma protocol. We refer the reader to Section 4 for more details.

The formulation of the openable sigma protocol looks similar to the standard OR sigma protocol. They both take n statements and one witness as input. However, there is a significant difference between them. The OR sigma protocol is a proof of knowledge for the OR-relation. The openable sigma protocol, on the other hand, is a proof of knowledge for the relation of the kth statement, where k is chosen at the proving stage and embedded in the first message com, and can then be extracted by the master secret key s_m.

For our openable sigma protocol, the special soundness would thus require an extractor that extracts the kth witness, which matches the opening result. Such a stronger extractor is crucial for proving unforgeability for group signatures, in which we transform a forger for party k into the extractor for the kth witness. Extractors for standard OR sigma protocols cannot provide such reduction.

Also, unlike an OR sigma protocol, an openable sigma protocol cannot get anonymity directly from the HVZK property, as the proving statement is now embedded in com. To achieve anonymity, we need an extra property *computational witness indistinguishability (CWI)*, which states that for an honest master key pair $(\mathsf{mpk}, \mathsf{msk})$, the proof for the ith the statement is indistinguishable from the proof for the jth statement. This promises that when transformed into signatures, the signer would be anonymous as long as the manager has not colluded.

The construction of our openable sigma protocol is built on top of the previous identity embedding component. For statements $E_1, \ldots, E_n$ along with the kth witness s_k s.t. $E_k = s_k E$ and the master key pair $(s_m, E_m = s_m E)$, the opening information in our protocol is set to

$$\tau = (E^{\beta}, E^{\mathsf{Open}}) = ((r_1 E_1, r_2 E_2, \ldots, r_n E_n), r_k s_k E_m), r_1, \ldots, r_n \xleftarrow{\$} G$$

As argued earlier, the manager can extract k from τ with s_m. To complete a proof of knowledge protocol, we use two challenges ($\mathsf{ch} = 1, 2$) to extract the knowledge of each r_i, and use another two challenges ($\mathsf{ch} = 3, 4$) to extract "some $d = r_k s_k$ s.t. $dE \in E^{\beta}$ and $dE_m = E^{\mathsf{Open}}$." This gives us a four-challenge openable sigma protocol with a corresponding special soundness property.

We detail the full construction and security proofs in Section 4.

Parallel repetitions and Fiat-Shamir transformation. From our 4-challenge sigma protocol with opening property, we immediately obtain an identification scheme with a soundness error $\frac{3}{4}$. In order to amplify away the soundness error, the designed openable sigma protocol is executed in parallel repetitions. The parallel repeated proof can be opened by taking the majority of the opening results from each of its sessions.

It may be tempting to claim that we can achieve soundness $(\frac{3}{4})^{\lambda}$ through a λ repetition. Unfortunately, this is not the case because each parallel session can be independently generated with a different witness, and some of the witnesses might be validly owned by the adversary. As a concrete example, in a λ-parallel protocol, an adversary that owns 3 keys can generate $\lambda/4 - 1$ honest parallel sessions on behalf of each key, and then cheat on only $\lambda/4 + 3$ sessions to achieve a successful forgery. Looking ahead, for an adversary owning n_A keys, we would need $n_A \cdot \mathsf{poly}(\lambda)$ repetitions to keep the error negligibly bounded, where the bound holds with its polynomial degree depending on the adversary being classical or quantum.

With an identification scheme under a negligible soundness error, we can now apply the Fiat-Shamir transformation and obtain a signature scheme. This is done by substituting the verifier's random message with the hash of earlier transcripts. Classically, a μ-special-sound protocol with some constant μ, under parallel repetitions, preserves its proof of knowledge (PoK) property after being Fiat-Shamir transformed. This is done by adopting the so-called improved forking lemma, which rewinds a soundness adversary under the hood multiple times in order to collect multiple outputs and extract a secret from it. However, in the

quantum setting, this does not work trivially. Each time measuring an output of a quantum adversary potentially corrupts its internal state and would require a different set of techniques for analysis.

A recently developed technique, measure-and-reprogram [18,19], gives a non-trivial reduction from non-interactive PoK to the interactive one. Then, by means of generalized Unruh's rewinding, one can again rewind the interactive adversary and use it to extract a secret, assuming the *collapsing property* of the underlying protocol. Keeping the goal to construct a signature in mind, at some point, we need to reduce security against *chosen-message attacks* to *no-message attacks*. The crucial part of such reduction lies in the simulation of the signing oracle. One typically needs to reprogram the random oracle and argue that such reprogramming is not noticeable by the adversary except with some small probability. Assuming the reprogrammed places are of high min-entropy, a recent technique *adaptive reprogramming* [25] helps one show that such reprogramming is not noticeable, even if the distribution of reprogrammed places is chosen by the quantum adversary on-the-fly.

Now, it becomes retrospectively clear why an openable sigma protocol should be defined in such a way that the index k of the proven statement is embedded in its first message com, and not in the responses: only then, obtaining μ accepted responses to distinct challenges at the same position $i \in [t]$ implies extraction because they share the same first message and therefore the same opening result. In order to obtain these μ responses, the abovementioned QROM tools are used in a non-blackbox manner. In particular, we are taking advantage of the fact that the extracted responses are subject to uniformly and independently chosen challenges in each repetition and rewinding. We can then fix the sessions that open to the desired k, and argue about the probability where μ distinct challenges occur at the same position.

3 Preliminary

3.1 Isogeny and class group action

At the bottom level of our construction is the so-called *isogeny class group action*, which considers a commutative class group $\mathsf{Cl}(\mathcal{O})$ acting on the set of supersingular elliptic curves $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ up to $\mathbb{F}_p$ isomorphisms. The group action is free and transitive: for every $E_1, E_2 \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$, there is exactly one $\mathfrak{a} \in \mathsf{Cl}(\mathcal{O})$ such that $E_2 \cong_{\mathbb{F}_p} \mathfrak{a}E_1$. For the use of cryptography, we note that computing the action is efficient while extracting $\mathfrak{a}$ from the end-point curves is considered intractable. This introduces a hard-to-compute relation while regarding the curves as public keys and the group element $\mathfrak{a}$ as secret. Note that validating the public key is efficient because it is efficient to validate the supersingularity of a curve. We refer readers to Supplementary C for a guided walk-through.

Hardness assumptions. Hardness for the *group action inverse problem* (GAIP) in Definition 1 is commonly assumed for the above-mentioned group action,

which has been shown useful on constructing signature schemes such as CSI-FiSh [6] and SeaSign [22].

Definition 1 (Group Action Inverse Problem (GAIP)). *On inputs* $E_1, E_2 \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$*, find* $\mathfrak{a} \in \mathsf{Cl}(\mathcal{O})$ *such that* $E_2 \cong_{\mathbb{F}_p} \mathfrak{a} \cdot E_1$*.*

In this work, we need to assume hardness for a weaker problem, the *decisional CSIDH problem* (abbreviated as D-CSIDH[7]) in Definition 2, which was considered already in [17,38], and is the natural generalization of the decisional Diffie-Hellman problem for group actions.

Definition 2 (Decisional CSIDH (D-CSIDH) / DDHAP). *For* $E \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$*, distinguish the two distributions*

- $(E, \mathfrak{a}E, \mathfrak{b}E, \mathfrak{c}E)$*, where* $\mathfrak{a}, \mathfrak{b}, \mathfrak{c} \xleftarrow{\$} \mathsf{Cl}(\mathcal{O})$*,*
- $(E, \mathfrak{a}E, \mathfrak{b}E, \mathfrak{a}\mathfrak{b}E)$*, where* $\mathfrak{a}, \mathfrak{b} \xleftarrow{\$} \mathsf{Cl}(\mathcal{O})$*.*

We note that for typical cryptographic constructions such as CSIDH, additional heuristic assumptions are required to sample a random element from the class group (as in Definition 2). This is because the "CSIDH-way" for doing this is by sampling exponents $(e_1, \ldots, e_n)$ satisfying $\forall i : |e_i| \leq b_i$, and the resulting distribution for ideals $\mathfrak{l}_1^{e_1} \ldots \mathfrak{l}_n^{e_n}$ is generally non-uniform within $\mathsf{Cl}(\mathcal{O})$. To get rid of such heuristics, one could instead work with specific parameters, where a bijective (yet efficient) representation of ideals is known. For instance, in [6], the structure of $\mathsf{Cl}(\mathcal{O})$ is computed, including a full generating set of ideals $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ and the entire lattice $\Lambda := \{(e_1, \ldots, e_n) | \mathfrak{l}_1^{e_1} \ldots \mathfrak{l}_n^{e_n} = \mathsf{id}\}$. Evaluating the group action is just a matter of approximating a *closest vector* and then evaluating the residue as in CSIDH. In this work, we will be working with such a "perfect" representation of ideals, unless otherwise specified.

As a remark, we note that the D-CSIDH problem for characteristic $p = 1 \mod 4$ is known to be broken [13]. Nevertheless, the attack is not applicable to the standard CSIDH setting where $p = 3 \mod 4$.

3.2 Group action DDH

In this section, we give an abstract version of the CSIDH group action. Such formulation will simplify our further construction and security proof.

A commutative group action $\mathcal{GA}_\lambda = (G_\lambda, \mathcal{E}_\lambda)$ with security parameter λ (we will omit the subscripts for simplicity) is called a DDH-secure group action if the following holds:

[7] This problem is called the decisional Diffie-Hellman group action problem (DDHAP) in [38].

– G acts freely and transitively on $\mathcal{E}$.
– DDHAP is hard on $\mathcal{GA}_\lambda$. i.e., for any efficient adversary A and $E \in \mathcal{E}$, the advantage for A distinguishing the following two distributions is $\mathsf{negl}(\lambda)$.
  • (E, aE, bE, cE), $a, b, c \xleftarrow{\$} G$
  • (E, aE, bE, abE), $a, b \xleftarrow{\$} G$

As a side remark, the GAIP problem is also hard on a DDH-secure group action.

For a DDH-secure group action, we can also have a natural parallel extension for DDHAP. Such extension is also discussed in [20].

Definition 3 (Parallelized-DDHAP (P-DDHAP)). *Given $E \in \mathcal{E}$, distinguish the two distributions*

– $(aE, \{b_iE\}_{i\in[m]}, \{c_iE\}_{i\in[m]})$, *where* $a, \{b_i\}_{i\in[m]}, \{c_i\}_{i\in[m]} \xleftarrow{\$} G$,
– $(aE, \{b_iE\}_{i\in[m]}, \{ab_iE\}_{i\in[m]})$, *where* $a, \{b_i\}_{i\in[m]} \xleftarrow{\$} G$.

By a simple hybrid argument, we can easily see that if $DDHAP$ is ϵ-hard, then P-DDHAP is $m\epsilon$ hard. To see this, note that a single DDHAP can be turned into a P-DDHAP as $(aE, \{r_ibE\}_{i\in[m]}, \{r_icE\}_{i\in[m]})$ for $\{r_i\}_{i\in[m]} \xleftarrow{\$} G$.

In the following we will use this in the form $(aE, \{b_iE\}_{i\in[m]}, \{c_iE\}_{i\in[m]}) \approx_c (aE, \{c_ia^{-1}E\}_{i\in[m]}, \{c_iE\}_{i\in[m]})$.

3.3 Sigma protocol

A sigma protocol is a three-message public coin proof of knowledge protocol. For a relation $R \subseteq X \times W$, where X is the space of statements and W is the space of witnesses, a sigma protocol for R consists of two proving algorithm P_1, P_2 and a verifying algorithm V. $P_1(x, w) \to (\mathsf{com}, st)$ outputs the first prover message com, named commitment, and a state st for P_2. The second message $\mathsf{ch} \xleftarrow{\$} \mathcal{C}$ from verifier, named challenge, honestly samples a challenge from the challenge space $\mathcal{C}$. Finally, $P_2(st, \mathsf{ch}) \to \mathsf{resp}$ outputs the third message resp, named response. The verifying algorithm $V(x, \mathsf{com}, \mathsf{ch}, \mathsf{resp}) \to 0(\text{reject})/1(\text{accept})$ outputs whether the verifier accepts the transcript. A sigma protocol should satisfy several properties. We refer readers to Supplementary D for further details.

3.4 The forking lemma

A sigma-protocol-based signature naturally allows witness extraction from the special soundness property. By extracting the witness from signature forgeries, one can reduce the unforgeability property to the hardness of computing the witness. However, the main gap between special soundness and unforgeability is that special soundness needs multiple related transcripts to extract the witness,

while a signature forging adversary only provides one. The forking lemma [33] is thus proposed to close this gap. For our particular application, as elaborated in Supplementary E.1, a generalized variant is adopted for the classical analysis.

3.5 Group signature

A group signature scheme consists of one manager and n parties. The manager can set up a group and provide secret keys to each party. Every party is allowed to generate signatures on behalf of the whole group. Such signatures are publicly verifiable without revealing the corresponding signers, except the manager can open signers' identities with his master's secret key. We refer readers to Supplementary G for group signature syntax and formal definitions.

3.6 Accountable ring signature

Accountable ring signatures (ARS) are a natural generalization for both group and ring signatures. Compared to a group signature, ARS gives the power of group decision to the signer. On signing, the signer can sign for an arbitrary group (or ring, to fit the original naming) and can decide on a master independent of the choice of the group. The master can open the signer's identity among the group without needing to participate in the key generation of parties in the ring. Note that accountable ring signatures directly imply group signatures simply by fixing the group and the master party at the key generation step. Thus, ARS can be viewed as a more flexible form of group signature.

Syntax. An accountable ring signature scheme $\mathcal{ARS}$ is associated with the following sets $\mathcal{M}$, $\mathcal{K}_m$, $\mathcal{K}$, $\mathcal{KP}_m$, $\mathcal{KP}$ and (efficient) algorithms **MKeygen**, **Keygen**, **Sign**, **Verify**, **Open**, as elaborated below.

- **MKeygen**$(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$ generates a master public-secret key pair.
- **Keygen**$(1^\lambda) \to (\mathsf{pk}, \mathsf{sk}) \in \mathcal{KP}$ generates a public key-secret key pair for a ring member.
- **Sign**$(\mathsf{mpk}, S, m, \mathsf{sk}) \to \sigma$, for a message $m \in \mathcal{M}$, a finite set of public keys $S \subset_{\mathsf{fin}} \mathcal{K}$ with the existence of $\mathsf{pk} \in S$ such that $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{KP}$, generates a signature σ.
- **Verify**$(\mathsf{mpk}, S, m, \sigma) \to \mathsf{acc} \in \{0, 1\}$ verifies whether the signature is valid.
- **Open**$(\mathsf{msk}, S, m, \sigma) \to \mathsf{pk} \in S \cup \{\perp\}$ reveals an identity pk, which presumably should be the public key of the signer of σ. It outputs $\mathsf{pk} = \perp$ when the opening fails, (e.g. when σ is malformed).

We refer to $\mathcal{M}$ as the message space, $\mathcal{K}_m$ as the master public key space and $\mathcal{K}$ as the public key space. We also define $\mathcal{KP}_m$ to be the set of all master key pairs $(\mathsf{mpk}, \mathsf{msk})$, and $\mathcal{KP}$ to be the set of all public-private key pairs $(\mathsf{pk}, \mathsf{sk})$. For simplicity, we keep the parameter λ implicit for the before-mentioned key

spaces, and additionally require public keys to be all distinct for a set S of size $|S| \leq \mathsf{poly}(\lambda)$.

An accountable ring signature scheme should satisfy the following security properties.

Correctness. An ARS is said to be correct if every honest signature can be correctly verified and opened.

Definition 4. *An accountable ring signature scheme $\mathcal{ARS}$ is correct if for any master key pair $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$, any key pair $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{KP}$, and any set of public keys S such that $\mathsf{pk} \in S$,*

$$\Pr\left[\mathsf{acc}=1 \wedge \mathsf{out}=\mathsf{pk} \;\middle|\; \begin{array}{c} \sigma \leftarrow \mathbf{Sign}(\mathsf{mpk}, S, m, \mathsf{sk}), \\ \mathsf{acc} \leftarrow \mathbf{Verify}(\mathsf{mpk}, S, m, \sigma), \\ \mathsf{out} \leftarrow \mathbf{Open}(\mathsf{msk}, S, m, \sigma) \end{array}\right] > 1 - \mathsf{negl}(\lambda).$$

Anonymity. An ARS is said to be anonymous if no adversary can determine the signer's identity within the set of signers of a signature without using the master secret key.

Definition 5. *An accountable ring signature scheme $\mathcal{ARS}$ is anonymous if for any PPT adversary A and any two key pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \in \mathcal{KP}$,*

$$\left|\Pr\left[1 \leftarrow A^{\mathbf{Sign}^*(\mathsf{mpk}_\bullet, \bullet, \bullet, \mathsf{sk}_0), \mathsf{mpk}_\bullet}(x)\right] - \Pr\left[1 \leftarrow A^{\mathbf{Sign}^*(\mathsf{mpk}_\bullet, \bullet, \bullet, \mathsf{sk}_1), \mathsf{mpk}_\bullet}(x)\right]\right| \leq \mathsf{negl}(\lambda)\ ,$$

with each query $\mathbf{Sign}^*(\mathsf{mpk}_\nu, S, m, \mathsf{sk}_b)$ returning an honest signature only when both $\mathsf{pk}_0, \mathsf{pk}_1 \in S$ and otherwise abort, where each master key pairs $(\mathsf{mpk}_\nu, \mathsf{msk}_\nu) \leftarrow \mathbf{MKeygen}(1^\lambda)$ are sampled honestly.

Remark 1. As we do not forbid x to contain information about the secret keys, adversaries in Definition 5 are referred to as being under the full key exposure.

Unforgeability. An ARS is said to be unforgeable if no adversary can forge a valid signature that fails to open or opens to some non-corrupted party, even if the manager has also colluded.

We model this property with the unforgeability game $G^{\mathsf{UF}}_{n_h}$. Among n_h honest keys pairs, the adversary A can call the signing oracle to obtain honest signatures, or call the corruption oracle to obtain the secret keys of the honest parties. The adversary wins if it outputs a valid signature that opens to a non-corrupted party or fails to open. We abuse the notation $\mathsf{sk}_i \in \mathsf{Hon}$ if $\mathsf{pk}_i \in \mathsf{Hon}$.

$G_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})$: Unforgeability game

1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathbf{Keygen}(1^\lambda)$
2: $(S^*, m^*, \sigma^*) \leftarrow A^{\mathbf{Sign}(\bullet,\bullet,\bullet,\mathsf{sk}),H}(\mathsf{pk})$
3: **check** σ^* is not produced by querying $\mathbf{Sign}(\mathsf{mpk}, S^*, m^*, \mathsf{sk})$
4: **check** $1 \leftarrow \mathbf{Verify}(\mathsf{mpk}, S^*, m^*, \sigma^*)$
5: **check** pk or $\perp \leftarrow \mathbf{Open}(\mathsf{msk}, S^*, m^*, \sigma^*)$
6: A wins if all check pass

Definition 6. *An accountable ring signature scheme $\mathcal{ARS}$ is unforgeable if for any PPT adversary A, any valid master key pair $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$*

$$\Pr[A \text{ wins } G_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})] < \mathsf{negl}(\lambda).$$

Transforming ARS to GS. As mentioned earlier, an accountable ring signature can be viewed as a generalization of a group signature. We give here the general transformation from an ARS scheme $\mathcal{ARS}$ to a group signature scheme $\mathcal{GS}^{\mathcal{ARS}}$.

The algorithms of the group signature scheme $\mathcal{GS}^{\mathcal{ARS}}$ are detailed as follows:

- $\mathbf{GKeygen}(1^\lambda, 1^n)$:
  1: $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathcal{ARS}.\mathbf{MKeygen}(1^\lambda)$
  2: $\forall i \in [n], (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathcal{ARS}.\mathbf{Keygen}(1^\lambda)$ and $S = \{\mathsf{pk}_i\}_{i\in[n]}$
  3: **return** $(\mathsf{gpk} = (\mathsf{mpk}, \mathsf{S}), \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk})$
- $\mathbf{GSign}(\mathsf{gpk} = (\mathsf{mpk}, S), m, \mathsf{sk}_k)$
  1: **return** $\sigma \leftarrow \mathcal{ARS}.\mathbf{Sign}(\mathsf{mpk}, S, m, \mathsf{sk}_k)$
- $\mathbf{GVerify}(\mathsf{gpk} = (\mathsf{mpk}, S), m, \sigma)$:
  1: **return** $\sigma \leftarrow \mathcal{ARS}.\mathbf{Verify}(\mathsf{mpk}, S, m, \sigma)$
- $\mathbf{GOpen}(\mathsf{gpk} = (\mathsf{mpk}, S), \mathsf{msk}, m, \sigma)$:
  1: $\mathsf{pk} \leftarrow \mathcal{ARS}.\mathbf{Open}(\mathsf{msk}, S, m, \sigma)$
  2: **return** k s.t. $\mathsf{pk} = \mathsf{pk}_k \in S$ or $\perp$ otherwise

Note that the transformation only changes the formulation of the setup stage. Thus, the security properties from $\mathcal{ARS}$ transfer directly to the induced group signature scheme $\mathcal{GS}^{\mathcal{ARS}}$.

4 Openable sigma protocol

In this section, we will introduce the openable sigma protocol, which is an intermediate primitive toward group signatures and accountable ring signatures. We will first give some intuition on how we formulate this primitive, and then give a formal definition and construction from DDH-hard group actions.

4.1 Intuition

Typical construction of a Fiat-Shamir-based signature starts from a sigma protocol. As introduced in Section 3.3, the three message protocol $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ only requires special soundness, which is, informally speaking, weaker than the unforgeability property in the sense that multiple transcripts are required in order to break the underlying hardness. The forking lemma closes this gap with the power of rewinding and random oracle programming. As stated in Section 3.4, the lemma takes a forger that outputs a single forgery and gives an algorithm that outputs multiple instances of valid $(\mathsf{com}, \mathsf{ch}_j, \mathsf{resp}_j)$'s. This gives a transformation from a signature breaker to a witness extractor, bridging the two security notions.

For our accountable ring signature, we thus plan to follow the previous roadmap. We design a sigma protocol that supports an extra "opening" property. Our openable sigma protocol takes n statements as input and additionally requires the prover to take a master public key mpk as input on generating the first message com. The function **Open**, with the master secret key msk, can then extract the actual statement to which the proving witness corresponds. For a com generated from statement $(x_1, \ldots, x_n)$ and witness w_i with $(x_i, w_i) \in R$, we have $x_i = \mathbf{Open}(\mathsf{com}, \mathsf{msk})$. As our target is a signature scheme, (x_i, w_i) would be set to public key/secret key pairs, and thus the open function outputs the signer's identity.

To achieve the stronger security property of ARS after the Fiat-Shamir transformation, our openable sigma protocol needs to have modified security properties correspondingly. For *special soundness*, we would not be satisfied with extracting only "one of the witnesses"; instead, we need to build an extractor that extracts a witness which matches the opening result. Such a stronger extractor will allow us to extract secret keys from adversaries that can impersonate other players. For *honest verifier zero knowledge* (HVZK), we require the transcript to be ZK even when given the master secret key msk. This is crucial for proving that the impersonating attack cannot succeed even with a corrupted manager. Note that when given msk, one cannot hope to hide the signer's identity, so we only require ZK against the signer's witness. The formulation for the HVZK simulator thus takes the signer identity as input. Finally, we need an extra property to provide anonymity for the signer, which we named *computational witness indistinguishability* (CWI). CWI requires that, given honest master key pairs, the transcript generated from two different witnesses/identities should be indistinguishable. This property is formulated as the indistinguishability of two signing oracles.

4.2 Definition

An openable sigma protocol Σ is defined with respect to two relations. A base relation $R \subset X \times W$ and an opening relation R_m. Each $\mathcal{R} \in \{R, R_m\}$ of the

both relations is efficiently samplable with respect to some distribution, but for a fresh sample $(x, s) \leftarrow \mathcal{R}(1^\lambda)$, it is hard to derive the witness s from the statement x as the security parameter λ scales. We will keep λ implicit for convenience if the context is clear. Additionally, we define the OR-relation for R, i.e. $(\{x_i\}_{i\in[n]}, s) \in R_n$ if and only if all x_i are distinct and $\exists i \in [n]$ s.t. $(x_i, s) \in R$. The openable sigma protocol Σ contains the following four algorithms.

- $\mathbf{Commit}(x_m, \{x_i\}_{i\in[n]}, s) \to (\mathsf{com}, st)$ generates a commitment com based on $(\{x_i\}_{i\in[n]}, s) \in R_n$. **Commit** also generates a state st which is shared with **Resp** and will be kept implicit for convenience.
- $\mathbf{Resp}(x_m, \{x_i\}_{i\in[n]}, s, \mathsf{com}, \mathsf{ch}, st) \to \mathsf{resp}$ computes a response resp relative to a challenge $\mathsf{ch} \xleftarrow{\$} \mathcal{C}$.
- $\mathbf{Verify}(x_m, \{x_i\}_{i\in[n]}, \mathsf{com}, \mathsf{ch}, \mathsf{resp}) \to 1/0$ verifies whether a tuple $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ is valid. **Verify** outputs 1 if the verification passes and 0 otherwise.
- $\mathbf{Open}(s_m, \{x_i\}_{i\in[n]}, \mathsf{com}) \to x \in \{x_i\}_{i\in[n]} \cup \{\perp\}$ reveals some $(x, s) \in R$, where s is the witness used to generate the commitment com. It outputs $x = \perp$ when the opening fails. (i.e. when com is malformed)

An openable sigma protocol is secure if it is high min-entropy, computational unique-response, correct, μ-special sound for some constant μ and statistical honest-verifier zero-knowledge, as defined below.

Definition 7 (High min-entropy). *An openable sigma protocol Σ is of high min-entropy if the for any possible commitment com_0*

$$\Pr[\mathbf{Commit}(x) \to \mathsf{com} = \mathsf{com}_0] \leq \mathsf{negl}(\lambda) .$$

Definition 8 (Unique-response property). *An openable sigma protocol Σ is computational unique-response if for every $(x_m, s_m) \in R_m$ and every efficient algorithm A*

$$\Pr_{(x,s)\leftarrow R(1^\lambda)} \left[\begin{array}{c} (S,\mathsf{com},\mathsf{ch},\mathsf{resp}_1,\mathsf{resp}_2))\leftarrow A(x) \\ \forall i\in[2]{:}1\leftarrow \mathbf{Verify}(x_m,S,\mathsf{com},\mathsf{ch}_i,\mathsf{resp}_i) \\ \mathsf{pk} \text{ or } \perp\leftarrow \mathbf{Open}(s_m,S,\mathsf{com}) \\ \mathsf{resp}_1\neq\mathsf{resp}_2 \end{array} \right] \leq \mathsf{negl}(\lambda) .$$

Furthermore, Σ is called perfect unique-response if for every $x_m, S, \mathsf{com}, \mathsf{ch}$ there is at most one resp such that $1 \leftarrow \mathbf{Verify}(x_m, S, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$.

Definition 9 (Correctness). *An openable sigma protocol Σ is correct if for all $n = poly(\lambda)$, $(x_m, s_m) \in R_m$, $(\{x_i\}_{i\in[n]}, s) \in R_n$, $\mathsf{ch} \in \mathcal{C}$, and $x \in \{x_i\}_{i\in[n]}$ such that $(x, s) \in R$,*

$$\Pr \left[\mathsf{acc} = 1 \wedge id = x \left| \begin{array}{c} \mathsf{com}\leftarrow\mathbf{Commit}(x_m,\{x_i\}_{i\in[n]},s), \\ \mathsf{resp}\leftarrow\mathbf{Resp}(x_m,\{x_i\}_{i\in[n]},s,\mathsf{com},\mathsf{ch}), \\ \mathsf{acc}\leftarrow\mathbf{Verify}(\{x_m,\{x_i\}_{i\in[n]},\mathsf{com},\mathsf{ch},\mathsf{resp}), \\ id\leftarrow\mathbf{Open}(s_m,\{x_i\}_{i\in[n]},\mathsf{com}) \end{array} \right. \right] \geq 1 - \mathsf{negl}(\lambda).$$

Definition 10 (μ-Special Soundness). *An openable sigma protocol Σ is μ-special sound if for all $n = poly(\lambda)$ there exists an efficient extractor* **Ext** *such*

that, for all $(x_m, s_m) \in R_m$ *and any* $(\{x_i\}_{i\in[n]}, \mathsf{com}, \{\mathsf{ch}_j\}_{j\in[\mu]}, \{\mathsf{resp}_j\}_{j\in[\mu]})$ *such that each* $\mathsf{ch}_j \in \mathcal{C}$ *are distinct, then*

$$\Pr\left[\begin{matrix}(\forall j\in[\mu],\ \mathsf{acc}_j=1)\wedge \\ (x=\perp\vee(x,s)\notin R)\end{matrix}\middle|\begin{matrix}\forall j\in\mathcal{C},\ \mathsf{acc}_j\leftarrow\mathbf{Ver}(x_m,\{x_i\}_{i\in[n]},\mathsf{com},\mathsf{ch}_j,\mathsf{resp}_j), \\ x\leftarrow\mathbf{Open}(s_m,\{x_i\}_{i\in[n]},\mathsf{com}), \\ s\leftarrow\mathbf{Ext}(\{x_i\}_{i\in[n]},\mathsf{com},\{\mathsf{ch}_j\}_{j\in[\mu]},\{\mathsf{resp}_j\}_{j\in[\mu]})\end{matrix}\right] = 0. \tag{1}$$

Definition 11 (Statistical honest-verifier zero-knowledge / sHVZK). *An openable sigma protocol* Σ *is statistical HVZK if there exists an efficient simulator* **Sim** *such that, for any* $x_m \in X_m$*, any* $(\{x_i\}_{i\in[n]}, s) \in R_n$*, and* $x \in \{x_i\}_{i\in[n]}$ *such that* $(x, s) \in R$*,*

$$\mathbf{Trans}(x_m, \{x_i\}_{i\in[n]}, s) \approx_s \mathbf{Sim}(x_m, \{x_i\}_{i\in[n]}, x)$$

where **Trans** *outputs honest transcript* $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ *generated honestly by* **Commit** *and* **Resp** *with honestly sampled* $\mathsf{ch} \xleftarrow{\$} \mathcal{C}$*.*

Definition 12 (Computational witness indistinguishability / CWI). *An openable sigma protocol* Σ *is computational witness indistinguishable, if for any two* $(x_i, s_i), (x_j, s_j) \in R$ *and any efficient adversary* A*, with* $x_m^\bullet(\nu)$ *returning* x_m^ν *where* $(x_m^\nu, s_m^\nu) \leftarrow R_m$ *is freshly sampled for each* ν*, we have*

$$\left|\Pr\left[1 \leftarrow A^{\mathbf{Trans}^*(x_m^\bullet,\bullet,s_i),x_m^\bullet}(x)\right] - \Pr\left[1 \leftarrow A^{\mathbf{Trans}^*(x_m^\bullet,\bullet,s_j),x_m^\bullet}(x)\right]\right| \leq \mathsf{negl}(\lambda)$$

where $\mathbf{Trans}^*(\mathsf{mpk}_\nu, S, s_k)$ *for whichever* $k \in \{i, j\}$ *returns an honest transcript* $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ *tuple from* Σ *if both* $x_i, x_j \in S$ *and aborts otherwise.*

4.3 Construction

Here, we give our construction to an openable sigma protocol Σ_{GA} for relations from our DDH-secure group action $\mathcal{GA} = (G, \mathcal{E})$. We let $E \in \mathcal{E}$ be some fixed element in $\mathcal{E}$. When implemented with CSIDH, we can choose the curve $E_0 : y^2 = x^3 + x$ for simplicity. Let the relation $R_E = \{(aE, a) | a \in G\} \subset \mathcal{E} \times G$.

For our Σ_{GA}, we set its opening and base relations $R_m = R = R_E$, with the natural instance generator that samples $a \xleftarrow{\$} G$ and outputs (aE, a). For inputs $E_m \in \mathcal{E}$ and $(\{E_i\}_{i\in[n]}, s) \in R_n$ with any $n = poly(\lambda)$, the algorithms for Σ_{GA} are constructed as follow.

- **Commit**$(E_m, \{E_i\}_{i\in[n]}, s)$
  1: set $k \in [n]$ s.t. $(E_k, s) \in R$.
  2: $\{\Delta_i\}_{i\in[n]}, \{\Delta'_i\}_{i\in[n]}, b \xleftarrow{\$} G$
  3: $\tau \xleftarrow{\$} sym(n)$ {τ is a random permutation}
  4: $\forall i \in [n] : E_i^\alpha := \Delta_i E_i$
  5: $\forall i \in [n] : E_i^\beta := \Delta'_i E_i^\alpha = \Delta_i \Delta'_i E_i$

6: $\forall i \in [n] : E_i^\gamma := bE_i^\beta = \Delta_i \Delta_i' bE_i$
7: $E^{\mathsf{Open}} := \Delta_k \Delta_k' sE_m$
8: $E^{\mathsf{Check}} := \Delta_k \Delta_k' bsE_m = bE^{\mathsf{Open}}$
9: $\mathsf{st} = (\{\Delta_i\}_{i\in[n]}, \{\Delta_i'\}_{i\in[n]}, b, l = \Delta_k \Delta_k' bs)$
10: **return** $(\mathsf{com}, \mathsf{st}) = ((\{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]}, \tau(\{E_i^\gamma\}_{i\in[n]}), E^{\mathsf{Open}}, E^{\mathsf{Check}}), \mathsf{st})$
{We use $\tau(\bullet)$ as a lazy convention of sending a permuted list}

– **Resp**$(E_m, \{E_i\}_{i\in[n]}, s, \mathsf{com}, \mathsf{ch}, st)$:

1: **if** $\mathsf{ch} = 1$ **then**
2: **return** $\mathsf{resp} := \{\Delta_i\}_{i\in[n]}$
3: **if** $\mathsf{ch} = 2$ **then**
4: **return** $\mathsf{resp} := \{\Delta_i'\}_{i\in[n]}$
5: **if** $\mathsf{ch} = 3$ **then**
6: **return** $\mathsf{resp} := b$
7: **if** $\mathsf{ch} = 4$ **then**
8: **return** $\mathsf{resp} := l = \Delta_k \Delta_k' bs$

– **Verify**$(E_m, \{E_i\}_{i\in[n]}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$:

1: **return** 0 if $\{E_i\}_{i\in[n]}$ or $\{E_i^\beta\}_{i\in[n]}$ are not all distinct
2: **if** $\mathsf{ch} = 1$ **then**
3: **check** $\forall i \in [n] : E_i^\alpha = \Delta_i E_i$
4: **if** $\mathsf{ch} = 2$ **then**
5: **check** $\forall i \in [n] : \Delta_i' E_i^\alpha = E_i^\beta$
6: **if** $\mathsf{ch} = 3$ **then**
7: **check** $\exists \tau' \in sym(n)$ s.t. $\tau'(\{bE_i^\beta\}_{i\in[n]}) = \tau(\{E_i^\gamma\}_{i\in[n]})$
8: **check** $E^{\mathsf{Check}} = bE^{\mathsf{Open}}$
9: **if** $\mathsf{ch} = 4$ **then**
10: **check** $E^{\mathsf{Check}} = lE_m$
11: **check** $\exists E^\gamma \in \tau(\{E_i^\gamma\}_{i\in[n]})$ *s.t.* $E^\gamma = lE$
12: **return** 1 **if all checks pass**

– **Open**$(s_m := \mathsf{msk}, \{E_i\}_{i\in[n]} := \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{com})$:

1: **for** $i \in [n]$ **do**
2: **if** $s_m E_i^\beta = E^{\mathsf{Open}}$ **then**
3: **return** E_i
4: **return** $\perp$

The construction of our openable sigma protocol looks complicated, but the intuition is simple. The core section of the message com is $(E^\beta, E^{\mathsf{Open}})$, which allows opening. The other parts of com are to ensure that the opening section is honestly generated. E^α along with the challenge/response pair on $\mathsf{ch} = 1, 2$ allows extraction for $\Delta_i \Delta_i'$'s, ensuring that E^β is honestly generated. $(E^\gamma, E^{\mathsf{Check}})$ along with the challenge/response pair on $\mathsf{ch} = 3, 4$ verifies the relation between E^β and E^{Open}. By using a permuted E^γ, the CWI property is preserved through such a verification process. Combined together, we complete the proof of knowledge protocol.

Theorem 2. *Σ_{GA} is an openable sigma protocol with R_E being both the opening relation and the base relation*

4.4 Security

The proof for Theorem 2 is broken down into proving each of the required properties. First, by construction one immediately get Σ_{GA} being perfect unique-response, and high min-entropy. It is also easy to show that Σ_{GA} is correct and statistical HVZK (see Supplementary A for full proof).

Lemma 1. *Σ_{GA} is **correct** and **statistical honest-verifier zero-knowledge**.*

Lemma 2. *Σ_{GA} is **4-special sound**.*

Proof. For any $E_m \in \mathcal{E}$ and any $(\{E_i\}_{i\in[n]}, \mathsf{com}, \{\mathsf{resp}_j\}_{j\in\mathcal{C}})$ where $\mathsf{com} = (\{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]}, \sigma(\{E_i^\gamma\}_{i\in[n]})$, and $\{\mathsf{resp}_j\}_{j\in[4]} = (\{\Delta_i\}_{i\in[n]}, \{\Delta_i'\}_{i\in[n]}, b, l)$. Suppose that $\forall j \in [4], 1 \leftarrow \mathbf{Ver}(E_m, \{E_i\}_{i\in[n]}, \mathsf{com}, j, \mathsf{resp}_j)$, then by the definition of **Verify**, we can get the following equations:

$$\begin{cases} \{E_i\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]} \text{ are both pairwise distinct sets} \\ \forall i \in [n] : E_i^\alpha = \Delta_i E_i, E_i^\beta = \Delta_i' E_i^\alpha \\ \exists \tau' \in sym(n)\ s.t.\ \tau'(\{bE_i^\beta\}_{i\in[n]}) = \tau(\{E_i^\gamma\}_{i\in[n]}) \\ \exists E^\gamma \in \tau(\{E_i^\gamma\}_{i\in[n]})\ s.t.\ E^\gamma = lE \\ E^{\mathsf{Check}} = lE_m = bE^{\mathsf{Open}} \end{cases}$$

Thus, there exists a unique $k \in [n]$ such that $lE = bE_k^\beta = \Delta_k' bE_k^\alpha = \Delta_k\Delta_k' bE_k$, which means $l(\Delta_k\Delta_k' b)^{-1}E = E_k$. This implies that $(E_k, l(\Delta_k\Delta_k' b)^{-1}) \in R_E$. Furthermore, we also have $E^{\mathsf{Open}} = b^{-1}lE_m = s_m b^{-1} lE = s_m E_k^\beta$. This implies that $E_k \leftarrow \mathbf{Open}(s_m, \{E_i\}_{i\in[n]}, \mathsf{com})$. Thus **Open** does not output $\perp$. From these observations, we can easily construct the extractor $\mathbf{Ext}(\mathsf{com}, \{\mathsf{resp}_j\}_{j\in\mathcal{C}})$, which simply searches through $k \in [n]$ for k satisfying $lE = bE_k^\beta$, then output $s = l(\Delta_k\Delta_k' b)^{-1}$. This concludes the proof that Σ_{GA} is **4-special sound**. □

Lemma 3. *Σ_{GA} is **computational witness indistinguishable** (assuming DDHAP is hard for $\mathcal{GA}$).*

Here we will finally use the fact that $\mathcal{GA}$ is DDH-hard. We will prove this theorem through two hybrids. We highlight the changes between **Trans** and $\mathbf{Hyb}_1$ and between $\mathbf{Hyb}_1$ and $\mathbf{Hyb}_2$ with different colors for easier comparison.

Proof. For any $E_m \in \mathcal{E}$ and any $(\{E_i\}_{i\in[n]}, \mathsf{com}, \{\mathsf{resp}_j\}_{j\in\mathcal{C}})$ where $\mathsf{com} = (\{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]}, \sigma(\{E_i^\gamma\}_{i\in[n]})$, and $\{\mathsf{resp}_j\}_{j\in[4]} = (\{\Delta_i\}_{i\in[n]}, \{\Delta_i'\}_{i\in[n]}, b, l)$.

Suppose that $\forall j \in [4], 1 \leftarrow \mathbf{Ver}(E_m, \{E_i\}_{i\in[n]}, \mathsf{com}, j, \mathsf{resp}_j)$, then by the definition of **Verify**, we can get the following equations:

$$\begin{cases} \{E_i\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]} \text{ are both pairwise distinct sets} \\ \forall i \in [n] : E_i^\alpha = \Delta_i E_i, E_i^\beta = \Delta_i' E_i^\alpha \\ \exists \tau' \in sym(n) \ s.t. \ \tau'(\{bE_i^\beta\}_{i\in[n]}) = \tau(\{E_i^\gamma\}_{i\in[n]}) \\ \exists E^\gamma \in \tau(\{E_i^\gamma\}_{i\in[n]}) \ s.t. \ E^\gamma = lE \\ E^{\mathsf{Check}} = lE_m = bE^{\mathsf{Open}} \end{cases}$$

Thus, there exists a unique $k \in [n]$ such that $lE = bE_k^\beta = \Delta_k' bE_k^\alpha = \Delta_k \Delta_k' bE_k$, which means $l(\Delta_k\Delta_k' b)^{-1}E = E_k$. This implies that $(E_k, l(\Delta_k\Delta_k' b)^{-1}) \in R_E$. Furthermore, we also have $E^{\mathsf{Open}} = b^{-1}lE_m = s_m b^{-1} lE = s_m E_k^\beta$. This implies that $E_k \leftarrow \mathbf{Open}(s_m, \{E_i\}_{i\in[n]}, \mathsf{com})$. Thus **Open** does not output $\perp$. From these observations, we can easily construct the extractor $\mathbf{Ext}(\mathsf{com}, \{\mathsf{resp}_j\}_{j\in\mathcal{C}})$, which simply searches through $k \in [n]$ for k satisfying $lE = bE_k^\beta$, then output $s = l(\Delta_k\Delta_k' b)^{-1}$. This concludes the proof that Σ_{GA} is **4-special sound**. □

Lemma 4. *For any $s \in G$, any efficient adversary A with $E_m^\bullet(\nu)$ generating E_m^ν from $(E_m^\nu, s_m^\nu) \leftarrow R_E$ for each ν, we have*

$$\left| \Pr\left[1 \leftarrow A^{\mathbf{Trans}(E_m^\bullet, \bullet, s), E_m^\bullet}(x)\right] - \Pr\left[1 \leftarrow A^{\mathbf{Hyb}_1(E_m^\bullet, \bullet, s), E_m^\bullet}(x)\right] \right| \leq \mathsf{negl}(\lambda) \ ,$$

where $\mathbf{Hyb}_1$ is as specified below.

$\mathbf{Hyb}_1(E_m, \{E_i\}_{i\in[n]}, s)$

1: $\mathsf{ch} \xleftarrow{\$} \{1,2,3,4\}$
2: set $k \in [n]$ s.t. $(E_k, s) \in R$.
3: $\{\Delta_i\}_{i\in[n]}, \{\Delta_i'\}_{i\in[n]}, b \xleftarrow{\$} G$
4: $\tau \xleftarrow{\$} sym(n)$
5: $\forall i \in [n] : E_i^\alpha = \Delta_i E_i, \ E_i^\beta = \Delta_i' E_i^\alpha$
6: $\forall i \in [n] : E_i^\gamma = bE_i^\beta$
7: $r \xleftarrow{\$} G, \ E^{\mathsf{Open}} = rE$
8: **if** $\mathsf{ch} = 1,2,3$ **then**
9: $\quad E^{\mathsf{Check}} = bE^{\mathsf{Open}}$
10: **else if** $\mathsf{ch} = 4$ **then**
11: $\quad l = \Delta_k\Delta_k' bs, \ E^{\mathsf{Check}} = lE_m$
12: set resp honestly w.r.t ch
13: **return** $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$

Proof. Each query input of **Trans** and $\mathbf{Hyb}_1$ is of form $(E_m, \{E_i\}_{i\in[n]}, s)$ where $(\{E_i\}_{i\in[n]}, s) \in R_n$ and E_m is the curve correspoinding to the random master public key. We first note that the difference between honest transcript **Trans** and $\mathbf{Hyb}_1$ is that $\mathbf{Hyb}_1$ replaces honest E^{Open} with rE for a random $r \in G$. For $\mathsf{ch} \neq 4$, E^{Check} is also replaced accordingly to E^{Open}.

We will prove the indistinguishability of $(\mathsf{com}, \mathsf{ch}, \mathsf{resp}) \leftarrow \mathbf{Trans}$ and $(\mathsf{com}', \mathsf{ch}', \mathsf{resp}') \leftarrow \mathbf{Hyb}_1$ for each different challenge $\mathsf{ch} \in \mathcal{C}$ separately. In the following proof, we set k s.t. $(E_k, s) \in R$, as in both **Trans** and $\mathbf{Hyb}_1$

For $\mathsf{ch}' = 1$, we have $\mathsf{resp}' = \{\Delta_i\}_{i\in[n]}$, which is honestly generated and thus identical to **Trans**. We thus focus on the com' part.

By the hardness of P-DDHAP, for random $\Delta'_k, r \xleftarrow{\$} G$, we have

$$(E_m, \Delta'_k E, \Delta'_k E_m) \approx_c (E_m, \Delta'_k E, rE)$$

Hence, for random $\Delta_k, \Delta'_k, b, r \xleftarrow{\$} G$ and honestly generated $(E_m, E_k^\beta, E_k^\gamma, E^{\mathsf{Open}}, E^{\mathsf{Check}})$, we have

$$\begin{aligned}
&(E_m, E_k^\beta, E_k^\gamma, E^{\mathsf{Open}}, E^{\mathsf{Check}}) \\
=&(E_m, \Delta_k s(\Delta'_k E), \Delta_k bs(\Delta'_k E), \Delta_k s(\Delta'_k E_m), \Delta_k bs(\Delta'_k E_m)) \\
\approx_c&(E_m, \Delta_k s(\Delta'_k E), \Delta_k bs(\Delta'_k E), \Delta_k s(rE), \Delta_k bs(rE)) \\
=&(E_m, E_k^\beta, E_k^\gamma, r'E, br'E)
\end{aligned}$$

Where the left-hand side is the output com from **Trans**, restricted to the variables dependent on s_m or Δ'_k. The right-hand side is the corresponding partial output from $\mathbf{Hyb}_1$. As the remaining parts of **Trans** and $\mathbf{Hyb}_1$ are equivalent, this equation shows that the output distributions of **Trans** and $\mathbf{Hyb}_1$ are indistinguishable for $\mathsf{ch} = 1$.

For the case $\mathsf{ch} = 2, 3$, the indistinguishability can be proved in a similar fashion. Notice again that for random $\Delta_k, r \xleftarrow{\$} G$, $(E_m, \Delta_k E, \Delta_k E_m) \approx_c (E_m, \Delta_k E, rE)$. Thus for random $\Delta_k, \Delta'_k, b, r \xleftarrow{\$} G$

$$\begin{aligned}
&(E_m, E_k^\alpha, E_k^\beta, E_k^\gamma, E^{\mathsf{Open}}, E^{\mathsf{Check}}) \\
=&(E_m, s(\Delta_k E), \Delta'_k s(\Delta_k E), \Delta'_k bs(\Delta_k E), \Delta'_k s(\Delta_k E_m), \Delta'_k bs(\Delta_k E_m)) \\
\approx_c&(E_m, s(\Delta_k E), \Delta'_k s(\Delta_k E), \Delta'_k bs(\Delta_k E), \Delta'_k s(rE), \Delta'_k bs(rE)) \\
=&(E_m, E_k^\alpha, E_k^\beta, E_k^\gamma, r'E, br'E)
\end{aligned}$$

For the case $\mathsf{ch} = 4$, we would need a slight change. First we recall the fact that, since $\mathcal{GA}$ is free and transitive, for every E_i there exists a unique $s_i \in G$ s.t. $s_i E = E_i$. Thus, sampling $\{D_i\}_{i\in[n]}, b \xleftarrow{\$} G$ and letting $\Delta'_i = (bs_i)^{-1} D_i$ gives us a uniformly distributed $\{\Delta'_i\}_{i\in[n]}$.

Now, again from P-DDHAP, for random $b, r \xleftarrow{\$} G$,

$$(E_m, b^{-1}E, b^{-1}E_m) \approx_c (E_m, b^{-1}E, rE)$$

Thus, for random $\{\Delta_i\}_{i\in[n]}, \{D_i\}_{i\in[n]}, b, r \xleftarrow{\$} G$ where $D_i = \Delta_i' b s_i$, we have

$$\begin{aligned}
&(E_m, \{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]}, \{E_i^\gamma\}_{i\in[n]}, E^{\mathsf{Open}}, E^{\mathsf{Check}}, l)\\
=&(E_m, \{\Delta_i E_i\}_{i\in[n]}, \{\Delta_i\Delta_i' E_i\}_{i\in[n]}, \{\Delta_i\Delta_i' b E_i\}_{i\in[n]}, \Delta_k\Delta_k' s_k E_m, \Delta_k\Delta_k' b s_k E_m, \Delta_k\Delta_k' b s_k)\\
=&(E_m, \{\Delta_i E_i\}_{i\in[n]}, \{\Delta_i D_i (b^{-1}E)\}_{i\in[n]}, \{\Delta_i D_i E\}_{i\in[n]}, \Delta_k D_k (b^{-1}E_m), \Delta_k D_k E_m, \Delta_k D_k)\\
\approx_c&(E_m, \{\Delta_i E_i\}_{i\in[n]}, \{\Delta_i D_i (b^{-1}E)\}_{i\in[n]}, \{\Delta_i D_i E\}_{i\in[n]}, \Delta_k D_k (rE), \Delta_k D_k E_m, \Delta_k D_k)\\
=&(E_m, \{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]}, \{E_i^\gamma\}_{i\in[n]}, r'E, E^{\mathsf{Check}}, l)
\end{aligned}$$

Finally, since both ch and ch' are sampled randomly in $\{1,2,3,4\}$, we can conclude that **Trans** and $\mathbf{Hyb}_1$ are computationally indistinguishable. □

Lemma 5. *For any $s \in G$, any efficient adversary A with $E_m^\bullet(\nu)$ generating E_m^ν from $(E_m^\nu, s_m^\nu) \leftarrow R_E$ for each ν, we have*

$$\left|\Pr\left[1 \leftarrow A^{\mathbf{Hyb_1}(E_m^\bullet, \bullet, s), E_m^\bullet}(x)\right] - \Pr\left[1 \leftarrow A^{\mathbf{Hyb_2}(E_m^\bullet, \bullet, s), E_m^\bullet}(x)\right]\right| \leq \mathsf{negl}(\lambda)\ ,$$

where $\mathbf{Hyb}_2$ is as defined below.

$\mathbf{Hyb}_2(E_m, \{E_i\}_{i\in[n]}, s)$

1: $\mathsf{ch} \xleftarrow{\$} \{1,2,3,4\}$
2: set $k \in [n]$ s.t. $(E_k, s) \in R$.
3: $\{\Delta_i\}_{i\in[n]}, \{\Delta_i'\}_{i\in[n]}, b \xleftarrow{\$} G$
4: $\tau \xleftarrow{\$} sym(n)$
5: $\forall i \in [n] : E_i^\alpha = \Delta_i E_i,\ E_i^\beta = \Delta_i' E_i^\alpha$
6: $r \xleftarrow{\$} G,\ E^{\mathsf{Open}} = rE$
7: **if** $\mathsf{ch} = 1,2,3$ **then**
8: $\quad E^{\mathsf{Check}} = bE^{\mathsf{Open}}$
9: $\quad \forall i \in [n] : E_i^\gamma = bE_i^\beta$
10: **else if** $\mathsf{ch} = 4$ **then**
11: $\quad \forall i \in [n] : r_i \xleftarrow{\$} G, E_i^\gamma = r_i E$
12: $\quad l = r_k,\ E^{\mathsf{Check}} = lE_m$
13: set resp honestly w.r.t ch
14: **return** $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$

Proof. The hybrids $\mathbf{Hyb}_1$ and $\mathbf{Hyb}_2$ differ only in the case $\mathsf{ch} = 4$, in which we replace the whole E^γ with random curves, E^{Check} and l are also changed correspondingly. As in the previous proof, we use the fact that sampling $\{D_i\}_{i\in[n]}, b \xleftarrow{\$} G$ and letting $\Delta_i' = (bs_i)^{-1} D_i$ gives us uniformly random $(\{\Delta_i'\}_{i\in[n]}, b)$

By P-DDHAP, for random $b, \{D_i\}_{i\in[n]\setminus\{k\}}, \{r_i\}_{i\in[n]\setminus\{k\}}$,

$$\begin{aligned}
&(b^{-1}E, \{D_i E\}_{i\in[n]\setminus\{k\}}, \{D_i b^{-1} E\}_{i\in[n]\setminus\{k\}})\\
\approx_c&(b^{-1}E, \{r_i E\}_{i\in[n]\setminus\{k\}}, \{D_i b^{-1} E\}_{i\in[n]\setminus\{k\}})
\end{aligned}$$

For simplicity, we let $S = [n]\setminus\{k\}$. Now, for random $\{\Delta_i\}_{i\in[n]}, \{D_i\}_{i\in[n]}, b, \{r_i\}_{i\in S}$ where $D_i = \Delta_i' b s_i$, and $(E_m, \{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in S}, \{E_i^\gamma\}_{i\in S}, E_k^\beta, E_k^\gamma, E^{\mathsf{Check}}, l)$ are the elements output from $\mathbf{Hyb}_1$, we have

$$
\begin{aligned}
&(E_m, \{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in S}, \{E_i^\gamma\}_{i\in S}, E_k^\beta, E_k^\gamma, E^{\mathsf{Check}}, l)\\
=&(E_m, \{\Delta_i E_i\}_{i\in[n]}, \{\Delta_i(D_i b^{-1}E)\}_{i\in S}, \{\Delta_i(D_i E)\}_{i\in S}, \Delta_k D_k(b^{-1}E),\\
&\quad \Delta_k D_k E, \Delta_k D_k E_m, \Delta_k D_k)\\
\approx_c&(E_m, \{\Delta_i E_i\}_{i\in[n]}, \{\Delta_i(D_i b^{-1}E)\}_{i\in S}, \{\Delta_i(r_i E)\}_{i\in S}, \Delta_k D_k(b^{-1}E),\\
&\quad \Delta_k D_k E, \Delta_k D_k E_m, \Delta_k D_k)\\
=&(E_m, \{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in S}, \{r'_i E_i\}_{i\in S}, E_k^\beta, E_k^\gamma, E^{\mathsf{Check}}, l)
\end{aligned}
$$

Finally we let $r'_k = \Delta_k D_k$, which is obviously independent from all other r'_i, then $(E_k^\beta, E_k^\gamma, E^{\mathsf{Check}}, l) = (r'_k b^{-1}E, r'_k E, r'_k E_m, r'_k)$. Note that $r'_k b^{-1}$ gives fresh randomness since b is now independent from all other elements in the right-hand side. Thus the right-hand side perfectly fits the distribution for $\mathbf{Hyb}_2$. This concludes that $\mathbf{Hyb}_1$ and $\mathbf{Hyb}_2$ are computationally indistinguishable.

□

Lemma 6. *For any $E_m \in X_m$, $\{E_i\}_{i\in[n]} \in X^n$, and s_{k_0}, s_{k_1} s.t. both $(\{E_i\}_{i\in[n]}, s_{k_0})$, $(\{E_i\}_{i\in[n]}, s_{k_1}) \in R_n$ then*

$$\mathbf{Hyb}_2(E_m, \{E_i\}_{i\in[n]}, s_{k_0}) = \mathbf{Hyb}_2(E_m, \{E_i\}_{i\in[n]}, s_{k_1}) \ ,$$

where "$=$" is understood as the output distribution being identical.

Proof. We always have $\mathbf{Hyb}_2(E_m, \{E_i\}_{i\in[n]}, s_{k_0}) = \mathbf{Hyb}_2(E_m, \{E_i\}_{i\in[n]}, s_{k_1})$ for $\mathsf{ch} = 1, 2, 3$, as every elements in the output is generated independently from k. For $\mathsf{ch} = 4$, we can give a deeper look on elements in $(\mathsf{com}, \mathsf{resp}) = (E^\alpha, E^\beta, E^\gamma, E^{\mathsf{Open}}, E^{\mathsf{Check}}, l)$. The part $(E^\alpha, E^\beta, E^{\mathsf{Open}})$ is generated independent from k, and the part $(E^\gamma, E^{\mathsf{Check}}, l)$ is of the form $(\tau(\{r_i E\}_{i\in[n]}), r_k E_m, r_k)$. Since τ is a random permutation and r_i's are independent randomness, the two distributions $(\tau(\{r_i E\}_{i\in[n]}), r_{k_0} E_m, r_{k_0})$ and $(\tau(\{r_i E\}_{i\in[n]}), r_{k_1} E_m, r_{k_1})$ are obviously identical. Hence $\mathbf{Hyb}_2(E_m, \{E_i\}_{i\in[n]}, s_{k_0}) = \mathbf{Hyb}_2(E_m, \{E_i\}_{i\in[n]}, s_{k_1})$.

□

Finally, by combining Lemma 4, Lemma 5, and Lemma 6, we conclude that for any efficient adversary A with $\mathsf{E}_m^\bullet$ and $\mathbf{Trans}^*$ defined as usual, and any $s_i, s_j \in G$, we have

$$\left|\Pr\left[1 \leftarrow A^{\mathbf{Trans}^*(E_m^\bullet, \bullet, s_i), E_m^\bullet}(x)\right] - \Pr\left[1 \leftarrow A^{\mathbf{Trans}^*(E_m^\bullet, \bullet, s_j), E_m^\bullet}(x)\right]\right| \leq \mathsf{negl}(\lambda) \ ,$$

by restricting the query inputs $(E_m, \{E_i\}_{i\in[n]}, s_k)$ to those $(\{E_i\}_{i\in[n]}, s_i), (\{E_i\}_{i\in[n]}, s_j) \in R_n$ for whichever $k \in \{i, j\}$. This concludes the proof of Lemma 3, and thus Σ_{GA} is indeed an openable sigma protocol.

5 Constructing accountable ring signatures

In this section, we will show how to obtain an accountable ring signature scheme from our openable sigma protocol. The construction can be decomposed into two parts. We first take multiple parallel repetitions to the protocol for soundness amplification; then, we apply the Fiat-Shamir transformation on the parallelized protocol to obtain the full construction. One subtle issue is that since every sigma protocol in the parallel repetition is generated independently, each parallel session of the transcript may open to a different party. Hence, we need an opening function for the parallelized protocol, which returns the majority output over the opening results of the parallel sessions.

5.1 Construction

More generally, we are going to construct our ARS scheme $\mathcal{ARS}^t_\Sigma$ by performing Fiat-Shamir transformation to the parallel repeated protocol $\Sigma^{\otimes t}$ where the number of repetitions $t(\lambda, n)$ depends on the security parameter λ and the number of members n. The construction of $\mathcal{ARS}^t_\Sigma$ is detailed as follows.

Remark 2. This can later be instantiated with $\mathcal{ARS}_{GA} := \mathcal{ARS}^t_{\Sigma_{GA}}$ by choosing $\Sigma := \Sigma_{GA}$ to be our previously constructed protocol over the group action and $t(\lambda, n) := 2\lambda n$.

- **MKeygen**(1^λ):
  1: **return** $(\mathsf{mpk}, \mathsf{msk}) \leftarrow R_m(1^\lambda)$
- **Keygen**(1^λ):
  1: **return** $(\mathsf{pk}, \mathsf{sk}) \leftarrow R(1^\lambda)$
- **Sign**$(\mathsf{mpk}, S, m, \mathsf{sk})$
  1: $t := t(\lambda, |S|)$
  2: $\forall j \in [t], (\mathsf{com}_j, st_j) \leftarrow \Sigma_{GA}.\mathbf{Commit}(\mathsf{mpk}, S, \mathsf{sk})$
  3: $(\mathsf{ch}_1, \ldots, \mathsf{ch}_t) \leftarrow H(m, \mathsf{com}_1, \ldots, \mathsf{com}_t, S)$
  4: $\forall j \in [t], \mathsf{resp}_j \leftarrow \Sigma.\mathbf{Resp}(\mathsf{mpk}, S, \mathsf{sk}, \mathsf{com}_j, \mathsf{ch}_j, st_j)$
  5: **return** $\sigma = (\mathsf{com}, \mathsf{resp}) := ((\mathsf{com}_1, \ldots, \mathsf{com}_t), (\mathsf{resp}_1, \ldots, \mathsf{resp}_t))$
- **Verify**$(\mathsf{mpk}, S, m, \sigma)$:
  1: $t := t(\lambda, |S|)$
  2: **parse** $\sigma = (\mathsf{com}, \mathsf{resp})$
  3: $\mathsf{ch} := H(m, \mathsf{com}, S)$
  4: **check** $\forall j \in [t] : 1 \leftarrow \Sigma.\mathbf{Verify}(\mathsf{mpk}, \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{com}_j, \mathsf{ch}_j, \mathsf{resp}_j)$
  5: **return** 1 **if all checks pass**
- **Open**$(\mathsf{msk}, S, m, \sigma)$:
  1: $t := t(\lambda, |S|)$
  2: **parse** $\sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$
  3: $\forall j \in [t], \mathsf{out}_j \leftarrow \Sigma.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}_i)$

4: $\mathsf{pk} = \mathbf{Maj}(\{\mathsf{out}_j\}_{j\in[t]})$ {**Maj** outputs the majority element from its input list. In case of ties, it outputs a random choice of the majority elements.}

5: **return** pk

Theorem 3. *Let Σ be a secure openable sigma protocol. Then $\mathcal{ARS}^t_\Sigma$ is secure for every $t(\lambda, n) = n \cdot \mathsf{poly}(\lambda)$. If Σ is furthermore perfect-unique-response, then $\mathcal{ARS}^t_\Sigma$ is QROM-secure.*

Proof. See Section 5.2, 5.3 for the proof. This is concluded jointly from Lemma 7, 8, 9, 10, 15. □

From Section 4.4 we know that Σ_{GA} is a secure openable sigma protocol being 4-special sound, and by applying the transformation from Section 3.6, we immediately get the following corollaries.

Corollary 1. *$\mathcal{ARS}_{GA}$ is a QROM-secure ARS scheme, if DDHAP is hard.*

Corollary 2. *$\mathcal{GS}^{\mathcal{ARS}_{GA}}$ is a QROM-secure GS scheme, if DDHAP is hard.*

This completes our construction of *both* an accountable ring signature scheme and a group signature scheme.

Remark 3. One additional benefit of using class group action as the key relation is that honest public keys can be efficiently verified. As discussed in Section 3.1, any $E_i \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ is a valid public key since the group action is transitive, and furthermore, any $E_i \notin \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ can be efficiently detected. This prevents the possibility of a malformed master key or malformed public keys, which is a potential attacking interface of an ARS scheme.

5.2 Classical Security

In this section, we are going to provide classical security proof of the $\mathcal{ARS}^t_\Sigma$ described earlier. Starting from classical ROM, and then lifting those to QROM.

For the proof of Theorem 3 we again break down the theorem into proving each security property, i.e. correctness, anonymity and unforgeability. For correctness, there is no difference between classical and quantum settings, but since the proof does not exploit "quantum-ness" of an adversary, we put it in this section as well.

Lemma 7. *Let Σ be a secure openable sigma protocol, then $\mathcal{ARS}^t_\Sigma$ is **correct**.*

Proof. For any master key pair $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$, any key pair $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{KP}$, and any set of public keys S such that $\mathsf{pk} \in S$, we directly have $(\mathsf{mpk}, \mathsf{msk}) \in R_m$

and $(S, \mathsf{sk}) \in R_n$ where $n = |S|$. Let $\sigma \leftarrow \mathbf{Sign}(\mathsf{mpk}, S, m, \mathsf{sk})$ be an honest signature on message m and ring S. Notice that in an honest execution of **Sign**, each com_j and resp_j is honestly generated according to Σ. Thus by the correctness of Σ, we know for $\mathsf{ch} := H(\mathsf{com}, m)$ and every $j \in [t]$ with probability $1 - \mathsf{negl}(\lambda)$, that $1 \leftarrow \Sigma.\mathbf{Verify}(\mathsf{mpk}, S, \mathsf{com}_j, \mathsf{ch}_j, \mathsf{resp}_j)$ and $\mathsf{pk} \leftarrow \Sigma.\mathbf{Open}(\mathsf{mpk}, S, \mathsf{com}_j)$. Hence we directly obtain that, with probability $1 - t \cdot \mathsf{negl}(\lambda) = 1 - \mathsf{negl}(\lambda)$, we have that $1 \leftarrow \mathbf{Verify}(\mathsf{mpk}, S, m, \sigma)$ and $\mathsf{pk} \leftarrow \mathbf{Open}(\mathsf{msk}, S, m, \sigma)$. This concludes the proof that $\mathcal{ARS}_\Sigma$ is correct. □

Lemma 8. *Let Σ be a secure openable sigma protocol, then $\mathcal{ARS}^t_\Sigma$ is **anonymous** for every $t(\lambda, n) \leq \mathsf{poly}(\lambda, n)$ in classical ROM.*

Proof. The anonymity of $\mathcal{ARS}_\Sigma$ follows immediately from the CWI property of Σ. For any efficient adversary A with at most q queries to the random oracle, it can have at most $q \cdot \mathsf{negl}(\lambda) \leq \mathsf{negl}(\lambda)$ advantage on distinguishing $\mathbf{Sign}^*$ and $(\mathbf{Trans}^*)^t$. And by CWI from Σ, we have $\mathbf{Trans}^*(\mathsf{mpk}, S, \mathsf{sk}_{id_0}) \approx_c \mathbf{Trans}^*(\mathsf{mpk}, S, \mathsf{sk}_{id_1})$. Hence we can directly conclude that $\mathbf{Sign}^*(\mathsf{mpk}, S, \mathsf{sk}_{id_0}) \approx_c \mathbf{Sign}^*(\mathsf{mpk}, S, \mathsf{sk}_{id_1})$, which proves that $\mathcal{ARS}^t_\Sigma$ is anonymous. □

Lemma 9. *Let Σ be a secure openable sigma protocol. Then $\mathcal{ARS}^t_\Sigma$ is **unforgeable** for every $t(\lambda, n) = n \cdot \mathsf{poly}(\lambda)$ in the classical ROM.*

We refer readers to E.2 for the proof.

5.3 QROM security

In this section, we are going to show QROM security.

To start, we show the anonymity first, where an adversary is asked to distinguish the signing oracles $\mathbf{Sign}^*(\mathsf{mpk}_\bullet, \bullet, \bullet, \mathsf{sk}_k)$ for $k \in \{i, j\}$. Recall that $\mathbf{Sign}^*$ is defined with respect to two fixed public-secret key pairs $(\mathsf{pk}_k, \mathsf{sk}_k) \in \mathcal{KP}$ for $k \in \{i, j\}$, a query $\mathbf{Sign}^*(\mathsf{mpk}_\nu, S, m, \mathsf{sk}_k)$ must be such that $\{\mathsf{pk}_i, \mathsf{pk}_j\} \subseteq S$.

The idea is that $\mathbf{Sign}^*$ behaves *almost* as if running t repetitions of the openable sigma protocol $\mathbf{Trans}^*(\mathsf{mpk}_\bullet, \bullet, \mathsf{sk}_k)$, which one cannot distinguish between $k \in \{i, j\}$. There is one exception: namely, $\mathbf{Sign}^*$ computes the challenges by hash evaluation $\mathsf{ch} := H(m, \mathsf{com}, S)$, but then since m and S are chosen by the adversary, this may cause bias to the challenge distribution.

Such bias is handled by reprogramming techniques. Note that the first message com is freshly sampled with high min-entropy in each query to $\mathbf{Sign}^*$. Therefore, it is unlikely that $H(m, \mathsf{com}, S)$ has been queried, and thus ch is almost unbiased. In the quantum setting, one cannot simply identify previous queries to H, but the adaptive reprogramming technique [25] can still be used to mimic this line of reasoning.

For convenience, we will use the prefix "$\Sigma^{\otimes t}$." to specify that the scope of the object lies in the t-time repetitions of Σ, with $\Sigma^{\otimes t}.\mathbf{Verify}$ outputting 1 if all

repetitions are accepted, and $\Sigma^{\otimes t}.\mathbf{Open}$ outputting the majority of the opening results.

Lemma 10. *Let Σ be an openable sigma protocol that is high min-entropy. Then $\mathcal{ARS}^t_\Sigma$ is* **anonymous** *for every $t(\lambda, n) \leq \mathsf{poly}(n, \lambda)$ in QROM.*

Proof. For the purpose of analysis, define the following $\mathbf{Sign}^*_2(\mathsf{mpk}_\bullet, \bullet, \bullet, \mathsf{sk}_k)$ oracle for $k \in \{i, j\}$.

- $\mathbf{Sign}^*_2(\mathsf{mpk}_\nu, S, m, \mathsf{sk}_k)$:
  1: **abort** if $\{\mathsf{pk}_i, \mathsf{pk}_j\} \not\subseteq S$
  2: $(\mathsf{com}, \mathsf{st}) \leftarrow \Sigma^{\otimes t}.\mathbf{Commit}(\mathsf{mpk}, S, \mathsf{sk}_k)$
  3: **program** $H(m, \mathsf{com}, S) := \mathsf{ch} \leftarrow \Sigma^{\otimes t}.\mathcal{C}$
  4: $\mathsf{resp} \leftarrow \Sigma^{\otimes t}.\mathbf{Resp}(\mathsf{mpk}, S, \mathsf{sk}_k, \mathsf{com}, \mathsf{ch}, \mathsf{st})$
  5: **return** $(\mathsf{com}, \mathsf{resp})$

Note that $\mathbf{Sign}^*_2$ only replaces the computation of challenge in $\mathbf{Sign}^*$ from $\mathsf{ch} := H(m, \mathsf{com}, S)$ using the random oracle H, to freshly sampling ch and reprogramming to $H(m, \mathsf{com}, S) := \mathsf{ch}$. As described earlier, since com is high-min-entropy, by [25, Theorem 1] we obtain $A^{\mathbf{Sign}^*(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_k),\mathsf{mpk}_\bullet,H}(\mathsf{pk}_k) \approx A^{\mathbf{Sign}^*_2(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_k),\mathsf{mpk}_\bullet,H}(\mathsf{pk}_k)$ being indistinguisable. Now, via Zhandry's compressed oracle technique, or alternatively as described in [16, Appendix A], there is an efficient quantum algorithm $B^{\mathbf{Trans}^*(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_k),\mathsf{mpk}_\bullet}(\mathsf{pk}_k)$ that run as if $A^{\mathbf{Sign}^*_2(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_k),\mathsf{mpk}_\bullet,H}(\mathsf{pk}_k)$ but emulating the random oracle and reprogramming by itself. Since Σ is computational witness-indistinguishable, B cannot distinguish between $k \in \{i, j\}$. Putting things together, we obtain the following chain of indistinguishability,

$$\begin{aligned}
&A^{\mathbf{Sign}^*(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_i),\mathsf{mpk}_\bullet,H}(\mathsf{pk}_i) \approx A^{\mathbf{Sign}^*_2(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_i),\mathsf{mpk}_\bullet,H}(\mathsf{pk}_i)\\
&\approx B^{\mathbf{Trans}^*(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_i),\mathsf{mpk}_\bullet}(\mathsf{pk}_i) \approx B^{\mathbf{Trans}^*(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_j),\mathsf{mpk}_\bullet}(\mathsf{pk}_j)\\
&\approx A^{\mathbf{Sign}^*_2(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_j),\mathsf{mpk}_\bullet,H}(\mathsf{pk}_j) \approx A^{\mathbf{Sign}^*(\mathsf{mpk}_\bullet,\bullet,\bullet,\mathsf{sk}_j),\mathsf{mpk}_\bullet,H}(\mathsf{pk}_j)\ .
\end{aligned}$$

This concludes the proof.

□

For the rest of this section, we show unforgeability in QROM. The key to lifting Lemma 9 into QROM is a quantum extraction technique. The classical forking lemma, which measures out part of the transcript before rewinding, may ruin the internal quantum state of the adversary and therefore does not trivially apply to the quantum setting.

First, we give a CMA-to-NMA reduction, i.e. transforming an adversary A against G^{UF}_A into an adversary against $\widetilde{G}^{\mathsf{UF}}_A$ as defined below.

$\widetilde{G}_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})$: NMA-Unforgeability game

1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathbf{Keygen}(1^\lambda)$
2: $(S^*, m^*, \sigma^*) \leftarrow A^H(\mathsf{pk})$
3: **check** $1 \leftarrow \mathbf{Verify}(\mathsf{mpk}, S^*, m^*, \sigma^*)$
4: **check** pk or $\perp \leftarrow \mathbf{Open}(\mathsf{msk}, S^*, m^*, \sigma^*)$
5: A wins if all check pass

This is by means of simulating the signing queries **Sign** via a simulator **Sim** as follows.

Sign$(\mathsf{mpk}, S, m, \mathsf{sk})$:
1: $t := t(\lambda, |S|)$
2: $\mathsf{com} \leftarrow \Sigma^{\otimes t}.\mathbf{Commit}$
3: $\mathsf{ch} := H(m, \mathsf{com}, S)$
4: $\mathsf{resp} \leftarrow \Sigma^{\otimes t}.\mathbf{Resp}$
5: **return** $(\mathsf{com}, \mathsf{resp})$

Sim(mpk, S, m):
1: $t := t(\lambda, |S|)$
2: $(\mathsf{com}, \mathsf{ch}, \mathsf{resp}) \leftarrow \Sigma^{\otimes t}.\mathbf{Sim}(\mathsf{mpk}, S, \mathsf{pk})$
3: **program** $H(a, m) := \mathsf{ch}$
4: **return** $(\mathsf{com}, \mathsf{resp})$

Lemma 11. *Let Σ be a statistical HVZK, high min-entropy openable sigma protocol, the number of repetitions be $t(\lambda, n) \leq \mathsf{poly}(\lambda, n)$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathbf{Keygen}(1^\lambda)$ be freshly sampled. Then for every efficient quantum algorithm A, we have*

$$\left|\Pr\left[1 \leftarrow A^{\mathbf{Sign}(\bullet,\bullet,\bullet,\mathsf{sk}),H}(\mathsf{pk})\right] - \Pr\left[1 \leftarrow A^{\mathbf{Sim},H}(\mathsf{pk})\right]\right| \leq \mathsf{negl}(\lambda) .$$

Proof. Define an intermediate oracle $\mathbf{Sign}_2$ as follows.

– $\mathbf{Sign}_2(\mathsf{mpk}, S, m)$:
1: $t := t(\lambda, |S|)$
2: $\mathsf{com} \leftarrow \Sigma^{\otimes t}.\mathbf{Commit}$
3: **program** $H(m, \mathsf{com}, S) := \mathsf{ch} \xleftarrow{\$} \mathcal{C}$
4: $\mathsf{resp} \leftarrow \Sigma^{\otimes t}.\mathbf{Resp}$
5: **return** $(\mathsf{com}, \mathsf{resp})$

Note that **Sign** and $\mathbf{Sign}_2$ only differs at one place, where the former computes the challenge $\mathsf{ch} := H(m, \mathsf{com}, S)$ using the random oracle H, but the latter samples a fresh challenge ch and then reprogrammed the corresponding entry $H(m, \mathsf{com}, S) := \mathsf{ch}$. Since com is of high-min-entropy, a direct application of [25, Theorem 1] implies $A^{\mathbf{Sign}(\bullet,\bullet,\bullet,\mathsf{sk}),H}(\mathsf{pk}) \approx A^{\mathbf{Sign}_2,H}(\mathsf{pk})$ being indistinguishable. Furthermore, $\mathbf{Sign}_2$ and **Sim** only differ in how the transcript is respectively generated, with the former produced via an honest execution $\Sigma^{\otimes t}.\mathbf{Trans}$, and the latter via the corresponding simulator $\Sigma^{\otimes t}.\mathbf{Sim}$. It follows directly from the HVZK property that $A^{\mathbf{Sign}_2,H}(\mathsf{pk}) \approx A^{\mathbf{Sim},H}(\mathsf{pk})$ is indistinguishable. This concludes the proof. □

Now we are ready to prove the CMA-to-NMA reduction.

Lemma 12. *Let Σ be a statistical HVZK, high min-entropy, computationally unique-response openable sigma protocol and the number of repetitions $t(\lambda, n) \leq \mathsf{poly}(\lambda, n)$. For every valid master key pair $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$ efficient (CMA) quantum adversary A against $G_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})$, there is an efficient (NMA) quantum adversary B against $\widetilde{G}_B^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})$ such that*

$$\left|\Pr\left[A \text{ wins } G_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})\right] - \Pr\left[B \text{ wins } \widetilde{G}_B^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})\right]\right| \leq \mathsf{negl}(\lambda) \ .$$

Proof. Let $B^H(\mathsf{pk})$ run $(S^*, m^*, \sigma^*) \leftarrow A^{\mathbf{Sim}, H}(\mathsf{pk})$ but emulating the reprogramming of H by itself. Already from Lemma 11 we may conclude the following

$$\left|\Pr\left[\begin{smallmatrix}(S^*,m^*,\sigma^*)\leftarrow A^{\mathbf{Sign}(\bullet,\bullet,\bullet,\mathsf{sk}),H}(\mathsf{pk})\\ 1\leftarrow\mathbf{Verify}(\mathsf{mpk},\mathsf{S}^*,\mathsf{m}^*,\sigma^*)\\ \mathsf{pk}\text{ or }\bot\leftarrow\mathbf{Open}(\mathsf{msk},S^*,m^*,\sigma^*)\end{smallmatrix}\right] - \Pr\left[\begin{smallmatrix}(S^*,m^*,\sigma^*)\leftarrow A^{\mathbf{Sim},H}(\mathsf{pk})\\ 1\leftarrow\mathbf{Verify}^H(\mathsf{mpk},\mathsf{S}^*,\mathsf{m}^*,\sigma^*)\\ \mathsf{pk}\text{ or }\bot\leftarrow\mathbf{Open}(\mathsf{msk},S^*,m^*,\sigma^*)\end{smallmatrix}\right]\right| \leq \mathsf{negl}(\lambda) \ ,$$

where $\mathbf{Verify}^H$ is understood as the verification with respect to the possibly reprogrammed random oracle H.

Without loss of generality we may assume $A^{\mathbf{Sim},H}$ never outputs σ^* produced by querying $\mathbf{Sim}(\mathsf{mpk}, S^*, m^*)$ for the message m^*. If the produced $(S^*, m^*, \sigma^*) \leftarrow A^{\mathbf{Sim},H}(\mathsf{pk})$ satisfies $1 \leftarrow \mathbf{Verify}^H(\mathsf{mpk}, \mathsf{S}^*, \mathsf{m}^*, \sigma^*)$ and $\mathsf{pk} \leftarrow \mathbf{Open}(\mathsf{msk}, S^*, m^*, \sigma^*)$. It may be (1) there has been a query of form $(\mathsf{com}^*, \mathsf{resp}) \leftarrow \mathbf{Sim}(\mathsf{mpk}, S^*, m^*, \mathsf{pk})$ for some resp so that there has been the reprogramming of form $H(m^*, \mathsf{com}^*, S^*) := \mathsf{ch}^*$, in which case $\mathsf{resp} \neq \mathsf{resp}^*$ so $(\mathsf{com}^*, \mathsf{ch}^*, \mathsf{resp}^*)$ and $(\mathsf{com}^*, \mathsf{ch}^*, \mathsf{resp})$ are distinct valid transcripts of $\Sigma^{\otimes t}$, which is hard to find due to the computational unique-response property, or (2) there has not been such a query, in which case $H(m^*, \mathsf{com}^*, S^*)$ would not have been reprogrammed (except with negligible probability), and so the verification $\mathbf{Verify}(\mathsf{msk}, S^*, m^*, \sigma^*)$ with respect to the un-reprogrammed H will pass. This concludes the proof.

□

Next, for every valid master key pair $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$, define the interactive unforgeability game $G_A^{\mathsf{int}}(\mathsf{mpk}, \mathsf{msk})$ as follows.

$G_A^{\mathsf{int}}(\mathsf{mpk}, \mathsf{msk})$: Interactive unforgeability game

---

1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathbf{Keygen}(1^\lambda)$
2: $(S^*, m^*, \mathsf{com}^*, \mathsf{st}) \leftarrow A(\mathsf{pk})$ and $t := t(\lambda, |S|)$
3: $\mathsf{ch} \xleftarrow{\$} \Sigma^{\otimes t}.\mathcal{C}$
4: $\mathsf{resp}^* \leftarrow A(\mathsf{st}, \mathsf{ch})$
5: A wins if the following holds: $1 \leftarrow \Sigma^{\otimes t}.\mathbf{Verify}(\mathsf{mpk}, S^*, \mathsf{com}^*, \mathsf{ch}^*, \mathsf{resp}^*)$ and pk or $\bot \leftarrow \Sigma^{\otimes t}.\mathbf{Open}(\mathsf{msk}, S^*, \mathsf{com}^*)$

We are going to reduce an NMA adversary to the another interactive adversary against the openable sigma protocol, with freedom to choose which set S of instances to break on its choice, so long as the secret key sk is included in S.

Lemma 13. *Let Σ be an openable sigma protocol. For every $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$ and every efficient (NMA) quantum adversary A against $\widetilde{G}^{\mathsf{UF}}_A(\mathsf{mpk}, \mathsf{msk})$ making at most q queries to the random oracle H, there is an efficient (interactive) quantum adversary B against $\widetilde{G}^{\mathsf{int}}_B$ such that the following holds*

$$\frac{\Pr\left[A \textit{ wins } \widetilde{G}^{\mathsf{UF}}_A(\mathsf{mpk}, \mathsf{msk})\right]}{(2q+1)^2} \leq \Pr\left[B \textit{ wins } G^{\mathsf{int}}_A(\mathsf{mpk}, \mathsf{msk})\right] .$$

Proof. This is via direct application of the measure-and-reprogram technique. For every fixed choice of $\mathsf{pk}^\circ \in \mathcal{KP}$, let V_{pk° be the predicate as described below.

- $V_{\mathsf{pk}^\circ}(x = (m, \mathsf{com}, S), \mathsf{ch}, \mathsf{resp})$:
  1: $t := t(\lambda, |S|)$
  2: **check** $1 \leftarrow \Sigma^{\otimes t}.\mathbf{Verify}(\mathsf{mpk}, S, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$
  3: **check** pk° or $\bot \leftarrow \Sigma^{\otimes t}.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com})$
  4: **return** 1 iff all check pass

By construction, for $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathbf{Keygen}(1^\lambda)$ we have

$$\Pr\left[A \text{ wins } \widetilde{G}^{\mathsf{UF}}_A\right] = \sum_{(\mathsf{pk}^\circ, \mathsf{sk}^\circ) \in \mathcal{KP}} \Pr\left[\begin{smallmatrix}\mathsf{pk}=\mathsf{pk}^\circ \\ \mathsf{sk}=\mathsf{sk}^\circ\end{smallmatrix}\right] \cdot \Pr_H\left[\begin{smallmatrix}(S^*, m^*, (\mathsf{com}^*, \mathsf{resp}^*)) \leftarrow A^H(\mathsf{pk}^\circ) \\ 1 \leftarrow V_{\mathsf{pk}^\circ}(x, H(x), \mathsf{resp}^*)\end{smallmatrix}\right]$$

$$\Pr\left[B \text{ wins } \widetilde{G}^{\mathsf{int}}_A\right] = \sum_{(\mathsf{pk}^\circ, \mathsf{sk}^\circ) \in \mathcal{KP}} \Pr\left[\begin{smallmatrix}\mathsf{pk}=\mathsf{pk}^\circ \\ \mathsf{sk}=\mathsf{sk}^\circ\end{smallmatrix}\right] \cdot \Pr_{\mathsf{ch} \leftarrow \Sigma^{\otimes t}.\mathcal{C}}\left[\begin{smallmatrix}(S^*, m^*, \mathsf{com}^*, \mathsf{st}) \leftarrow B(\mathsf{pk}^\circ) \\ \mathsf{resp}^* \leftarrow B(\mathsf{st}, \mathsf{ch}) \\ 1 \leftarrow V_{\mathsf{pk}^\circ}(x, H(x), \mathsf{resp}^*)\end{smallmatrix}\right] ,$$

for every interactive algorithm B, where x denotes $(m^*, \mathsf{com}^*, S^*)$. Summing over $x_\circ$ in [19, Theorem 2], we obtain the existence of an efficient B such that the following holds for all $\mathsf{pk}^\circ, \mathsf{sk}^\circ$

$$\Pr_{\mathsf{ch} \leftarrow \Sigma^{\otimes t}.\mathcal{C}}\left[\begin{smallmatrix}(S^*, m^*, \mathsf{com}^*, \mathsf{st}) \leftarrow B(\mathsf{pk}^\circ) \\ \mathsf{resp}^* \leftarrow B(\mathsf{st}, \mathsf{ch}) \\ 1 \leftarrow V_{\mathsf{pk}^\circ}(x, \mathsf{ch}^*, \mathsf{resp}^*)\end{smallmatrix}\right] \geq \Pr_H\left[\begin{smallmatrix}(S^*, m^*, (\mathsf{com}^*, \mathsf{resp}^*)) \leftarrow A^H(\mathsf{pk}^\circ) \\ 1 \leftarrow V_{\mathsf{pk}^\circ}(x, H(x), \mathsf{resp}^*)\end{smallmatrix}\right] \Big/ (2q+1)^2 .$$

Finally, summing over all choice of $(\mathsf{pk}^\circ, \mathsf{sk}^\circ)$ with suitable probability, the proof is concluded. □

Finally, we reduce an interactive adversary against G^{int}_A into another adversary that extract the secret key sk from the public key pk. Note that the key generation samples a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow R(1^\lambda)$ with respect to a hard relation R, and thus the secret key sk should be hard to extract.

The idea of extraction goes as follows. Let $t(\lambda, n) = (n+1)\kappa$ be the number of repetitions, where κ is to be decided later. If A wins G^{int}_A, i.e. producing a

valid transcript that is opened to pk or $\perp$, then by the pigeonhole principle, there must be at least κ repetitions opened to pk or at least κ opened to $\perp$. We then perform rewinding in order to collect sufficient number of accepted responses for these repetitions. Once there are μ accepted responses in the same repetition being produced with non-zero probability, (1) immediately falsify them being opened to $\perp$, and so we can always extract a secret key sk using the extractor $\Sigma.\mathbf{Ext}$ provided by the μ-special-sound property.

Note that it is not just a black-box evocation of (generalized) Unruh's rewinding because it only provides guarantee toward the number of collected valid transcripts, but not toward the content of those transcripts. When analyzing a parallel-repetition multi-special-sound protocol, one needs to open up the rewinding argument and see what's inside. On a very high-level, thanks to the fact that the opening result is determined once the first message com is produced, one can still argue that conditioned on any fixed choice of the opening result, the collected transcripts are with challenges being uniformly random. The analysis is more involved, and we refer interested readers to Supplementary F.1.

Lemma 14. *Let Σ be a μ-special-sound openable sigma protocol, the number of repetitions be $t(\lambda, n) = (n+1) \cdot \kappa(\lambda, n)$. For every $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$ and every efficient quantum adversary A against G_A^{int}, there exists an efficient quantum adversary B such that*

$$\Pr\left[A \text{ wins } G_A^{\mathsf{int}}(\mathsf{mpk}, \mathsf{msk})\right]^{2\mu-1} \leq \Pr\left[\begin{smallmatrix}(\mathsf{pk},\mathsf{sk})\leftarrow R(1^\lambda)\\ \mathsf{sk}\leftarrow B(\mathsf{pk})\end{smallmatrix}\right] + \exp\left(\frac{-\kappa}{\mu^\mu}\right) .$$

Putting everything together, we conclude unforgeability in QROM. For completion, see Supplementary F.2 for a rather formal wrapping up.

Lemma 15. *Let μ be a constant and Σ be an openable sigma protocol being correct, μ-special-sound, statistical HVZK, perfect-unique-response, and high min-entropy. Then $\mathcal{ARS}_\Sigma^t$ is* **unforgeable** *in QROM for every $t(\lambda, n) = n \cdot \mathsf{poly}(\lambda)$.*

Acknowledgments

Authors were supported by Taiwan Ministry of Science and Technology Grant 109-2221-E-001-009-MY3, Sinica Investigator Award (AS-IA-109-M01), Executive Yuan Data Safety and Talent Cultivation Project (AS-KPQ-109-DSTCP), and Young Scholar Fellowship (Einstein Program) of the Ministry of Science and Technology (MOST) in Taiwan, under grant number MOST 110-2636-E-002-012, and by the Netherlands Organisation for Scientific Research (NWO) under grants 628.001.028 (FASOR) and 613.009.144 (Quantum Cryptanalysis of Post-Quantum Cryptography), and by the NWO funded project HAPKIDO (Hybrid Approach for quantum-safe Public Key Infrastructure Development for Organisations). Mi-Ying (Miryam) Huang is additionally supported by the NSF CAREER award 2141536, the United State. This work was carried out while the fifth author was visiting Academia Sinica, she is grateful for the hospitality.

References

1. Bansarkhani, R.E., Misoczki, R.: G-Merkle: A Hash-Based Group Signature Scheme from Standard Assumptions. In: PQCrypto. Lecture Notes in Computer Science, vol. 10786, pp. 441–463. Springer (2018)
2. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 2656, pp. 614–629. Springer (2003)
3. Bellare, M., Shi, H., Zhang, C.: Foundations of Group Signatures: The Case of Dynamic Groups. In: CT-RSA. Lecture Notes in Computer Science, vol. 3376, pp. 136–153. Springer (2005)
4. Beullens, W., Dobson, S., Katsumata, S., Lai, Y.F., Pintore, F.: Group signatures and more from isogenies and lattices: Generic, simple, and efficient. Cryptology ePrint Archive (2021)
5. Beullens, W., Katsumata, S., Pintore, F.: Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 12492, pp. 464–492. Springer (2020)
6. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 11921, pp. 227–247. Springer (2019)
7. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: CRYPTO. Lecture Notes in Computer Science, vol. 3152, pp. 41–55. Springer (2004)
8. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of Fully Dynamic Group Signatures. In: ACNS. Lecture Notes in Computer Science, vol. 9696, pp. 117–136. Springer (2016)
9. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J., Petit, C.: Short Accountable Ring Signatures Based on DDH. In: ESORICS (1). Lecture Notes in Computer Science, vol. 9326, pp. 243–265. Springer (2015)
10. Brickell, E.F., Pointcheval, D., Vaudenay, S., Yung, M.: Design Validations for Discrete Logarithm Based Signature Schemes. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 1751, pp. 276–292. Springer (2000)
11. Camenisch, J., Michels, M.: A Group Signature Scheme with Improved Efficiency. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 1514, pp. 160–174. Springer (1998)
12. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: An Efficient Post-Quantum Commutative Group Action. In: ASIACRYPT (3). Lecture Notes in Computer Science, vol. 11274, pp. 395–427. Springer (2018)
13. Castryck, W., Sotáková, J., Vercauteren, F.: Breaking the Decisional Diffie-Hellman Problem for Class Group Actions Using Genus Theory. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 12171, pp. 92–120. Springer (2020)
14. Chaum, D., van Heyst, E.: Group Signatures. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 547, pp. 257–265. Springer (1991)
15. Chiesa, A., Ma, F., Spooner, N., Zhandry, M.: Post-quantum succinct arguments: breaking the quantum rewinding barrier. arXiv preprint arXiv:2103.08140 (2021)
16. Chung, K.M., Fehr, S., Huang, Y.H., Liao, T.N.: On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In: EUROCRYPT. pp. 598–629. Lecture Notes in Computer Science, Springer (2021)
17. Couveignes, J.: Hard Homogeneous Spaces. Cryptology ePrint Archive, Report 2006/291 (2006)

18. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 11693, pp. 356–383. Springer (2019)
19. Don, J., Fehr, S., Majenz, C.: The measure-and-reprogram technique 2.0: multi-round fiat-shamir and more. In: CRYPTO. pp. 602–631. Lecture Notes in Computer Science, Springer (2020)
20. El Kaafarani, A., Katsumata, S., Pintore, F.: Lossy CSI-FiSh: Efficient Signature Scheme with Tight Reduction to Decisional CSIDH-512. In: Public Key Cryptography (2). Lecture Notes in Computer Science, vol. 12111, pp. 157–186. Springer (2020)
21. Ezerman, M.F., Lee, H.T., Ling, S., Nguyen, K., Wang, H.: A Provably Secure Group Signature Scheme from Code-Based Assumptions. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 9452, pp. 260–285. Springer (2015)
22. Feo, L.D., Galbraith, S.D.: SeaSign: Compact Isogeny Signatures from Class Group Actions. In: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 11478, pp. 759–789. Springer (2019)
23. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: CRYPTO. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer (1986)
24. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A Group Signature Scheme from Lattice Assumptions. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 6477, pp. 395–412. Springer (2010)
25. Grilo, A.B., Hövelmanns, K., Hülsing, A., Majenz, C.: Tight adaptive reprogramming in the qrom. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 637–667. Springer (2021)
26. Kiayias, A., Yung, M.: Secure scalable group signature with dynamic joins and separable authorities. Int. J. Secur. Networks **1**(1/2), 24–45 (2006)
27. Laguillaumie, F., Langlois, A., Libert, B., Stehlé, D.: Lattice-Based Group Signatures with Logarithmic Signature Size. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 8270, pp. 41–61. Springer (2013)
28. Lai, Y.F., Dobson, S.: Collusion resistant revocable ring signatures and group signatures from hard homogeneous spaces. Cryptology ePrint Archive (2021)
29. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions. In: ASIACRYPT (2). Lecture Notes in Computer Science, vol. 10032, pp. 373–403 (2016)
30. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Lattice-Based Group Signatures: Achieving Full Dynamicity with Ease. In: ACNS. Lecture Notes in Computer Science, vol. 10355, pp. 293–312. Springer (2017)
31. Lyubashevsky, V.: Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 5912, pp. 598–616. Springer (2009)
32. Nguyen, P.Q., Zhang, J., Zhang, Z.: Simpler Efficient Group Signatures from Lattices. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 9020, pp. 401–426. Springer (2015)
33. Pointcheval, D., Stern, J.: Security Proofs for Signature Schemes. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 1070, pp. 387–398. Springer (1996)
34. Pointcheval, D., Stern, J.: Security Arguments for Digital Signatures and Blind Signatures. J. Cryptol. **13**(3), 361–396 (2000)

35. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: ASIACRYPT. Lecture Notes in Computer Science, vol. 2248, pp. 552–565. Springer (2001)
36. Şahin, M.S., Akleylek, S.: A constant-size lattice-based partially-dynamic group signature scheme in quantum random oracle model. Journal of King Saud University-Computer and Information Sciences (2022)
37. Schnorr, C.: Efficient Signature Generation by Smart Cards. J. Cryptol. **4**(3), 161–174 (1991)
38. Stolbunov, A.: Cryptographic Schemes Based on Isogenies. Ph.D. thesis (01 2012). https://doi.org/10.13140/RG.2.2.20826.44488
39. Unruh, D.: Quantum Proofs of Knowledge. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 7237, pp. 135–152. Springer (2012)
40. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 755–784. Springer (2015)
41. Vélu, J.: Isogénies entre courbes elliptiques. CR Acad. Sci. Paris, Séries A **273**, 305–347 (1971)
42. Xu, S., Yung, M.: Accountable Ring Signatures: A Smart Card Approach. In: CARDIS. IFIP, vol. 153, pp. 271–286. Kluwer/Springer (2004)

Supplementary Material

A Security proofs for Σ_{GA}

A.1 Proof of Lemma 1

Proof of correctness. By the definition of **Commit** and **Verify**, any honestly generated $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ based on $(\{E_i\}_{i\in[n]}, s) \in R_n$ will be accepted as long as the set $\{E_i^\beta\}_{i\in[n]}$ is pairwise distinct. Since $\mathcal{GA}$ is free and transitive, there is a unique $g \in G$ s.t. $gE_i = E_j$. Thus, $E_i^\beta = E_j^\beta$ if and only if $(\Delta_j\Delta'_j)^{-1}\Delta_i\Delta'_i = g$, which happens with negligible probability since all Δ's are honestly sampled. Hence with probability $1 - n \cdot \mathsf{negl}(\lambda)$, the set $\{E_i^\beta\}_{i\in[n]}$ are all distinct, and hence **Verify** accepts.

For the function **Open**, note that if $(E_m, s_m) \in R_m$ and $(E_k, s) \in R$, then $E^{\mathsf{Open}} = \Delta_k\Delta'_k s E_m = \Delta_k\Delta'_k s s_m E$, hence $s_m E_k^\beta = E^{\mathsf{Open}}$. As argued previously, $\{E_i^\beta\}_{i\in[n]}$ are all distinct with probability $1 - \mathsf{negl}(\lambda)$, and k would be unique if this is the case. Thus the probability that **Open** outputs E_k is overwhelming, concluding the proof that Σ_{GA} is correct.

□

Proof of statistical HVZK. The construction of **Sim** is given in the following algorithm. We will show that **Sim** is in fact a perfect simulator for **Trans**.

Sim$(E_m, \{E_i\}_{i\in[n]}, E_k)$

$\mathsf{ch} \xleftarrow{\$} \{1,2,3,4\}$
$b \xleftarrow{\$} G$, $\tau \xleftarrow{\$} sym(n)$
if $\mathsf{ch} = 1$ **then**
  $\{\Delta_i\}_{i\in[n]}, \{D_i\}_{i\in[n]} \xleftarrow{\$} G$
  $\forall i \in [n] : E_i^{\alpha} := \Delta_i E_i$
  $\forall i \in [n] : E_i^{\beta} := \Delta_i D_i E$
  $E^{\mathsf{Open}} := \Delta_k D_k E_m$
else if $\mathsf{ch} = 2,3,4$ **then**
  $\{D_i\}_{i\in[n]}, \{\Delta'_i\}_{i\in[n]} \xleftarrow{\$} G$
  $\forall i \in [n] : E_i^{\alpha} := D_i E$
  $\forall i \in [n] : E_i^{\beta} := \Delta'_i E_i^{\alpha}$
  $E^{\mathsf{Open}} := D_k \Delta'_k E_m$
$\forall i \in [n] : E_i^{\gamma} := b E_i^{\beta}$
if $\mathsf{ch} = 1,2,3$ **then**
  $E^{\mathsf{Check}} := b E^{\mathsf{Open}}$
else if $\mathsf{ch} = 4$ **then**
  $l := \Delta_k D_k b$
  $E_{km}^{\mathsf{Check}} := l E_m$
return $(\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ with the same definition as honest **Commit** and **Resp**

Since $\mathcal{GA}$ is free and transitive, for every $E_i \in \mathcal{E}$, there exists a unique $s_i \in G$ s.t. $s_i E = E_i$. In **Sim**, we can thus set $\Delta'_i = D_i s_i^{-1}$ in case $\mathsf{ch} = 1$ and $\Delta_i = D_i s_i^{-1}$ in case $\mathsf{ch} = 2,3,4$. Since the distribution of $D_i s_i^{-1}$ is uniformly random, **Sim** generates identical distributions for Δ's as **Trans**. Thus the output distribution of **Sim** should also be identical to the real transcript. Checking that verification passes for all cases shows that **Sim** is a perfect simulator. □

B Judging the opening

Due to the majority voting that we have adopted in our opening design, we do not know yet how to construct a proof for the exact opening output. However, as a natural byproduct of our construction, we could also empower the manager to generate a proof π additionally from **Open** that could be publicly verified showing for multiple sessions $s_m E_k^{\beta} = E^{\mathsf{Open}}$ (as in Section 4.3), which is done with a slight twist to Couveignes' sigma protocol, as defined in **Judge** below.

- **Open**$(\mathsf{msk}, S = \{\mathsf{pk}_i\}_{i\in[n]}, m, \sigma) \to (\mathsf{pk}, \pi) \in (S \cup \{\perp\}) \times \{0,1\}^*$: The redefined open algorithm not only reveals signer identity pk but also produces a publicly verifiable proof π for it.
- **Judge**$(\mathsf{mpk}, S = \{\mathsf{pk}_i\}_{i\in[n]}, \sigma, \mathsf{pk}, \pi) \to \mathsf{acc} \in \{0,1\}$: The judge algorithm accepts if the manager opened correctly,

Note that in Section 4.3, the opening within the sigma protocol is done by picking the index k such that $s_m E_k^{\beta} = E^{\mathsf{Open}}$. A manager could therefore prove this equality in a Schnorr-like manner, re-starting from the sigma protocol Σ_{GA} with three additional algorithms **JCommit**, **JResp**, **JVerify**.

- **JCommit**$(s_m := \mathsf{msk}, \{E_i\}_{i\in[n]} := \{\mathsf{pk}_i\}_{i\in[n]}, \mathsf{com})$:
  1: $b' \xleftarrow{\$} G$

2: **parse** $\mathsf{com} = (\{\mathsf{E}_\mathsf{i}^\alpha\}_{\mathsf{i}\in[\mathsf{n}]}, \{\mathsf{E}_\mathsf{i}^\beta\}_{\mathsf{i}\in[\mathsf{n}]}, \tau(\{\mathsf{E}_\mathsf{i}^\gamma\}_{\mathsf{i}\in[\mathsf{n}]}), \mathsf{E}^{\mathsf{Open}}, \mathsf{E}^{\mathsf{Check}})$ {We use $\tau(\bullet)$ as a lazy convention of sending a permuted list}
3: $E^{\mathsf{Judge}} := b' E^{\mathsf{Open}}$
4: $E_m^{b'} := b' s_m E$
5: **return** $(\mathsf{jcom}, \mathsf{jst}) = \Big((E^{\mathsf{Judge}}, E_m^{b'}), (b', s_m)\Big)$

– **JResp**$(E_m, \{E_i\}_{i\in[n]}, \mathsf{jcom}, \mathsf{jch}, \mathsf{jst})$:
1: **parse** $\mathsf{jst} = (b', s_m)$
2: **if** $\mathsf{jch} = 0$ **then**
3: **return** $\mathsf{jresp} := b'$
4: **if** $\mathsf{jch} = 1$ **then**
5: **return** $\mathsf{jresp} := l' = b' s_m$

– **JVerify**$(E_m := \mathsf{mpk}, \{E_i\}_{i\in[n]} := \{\mathsf{pk}_i\}_{i\in[n]}, E_k := \mathsf{pk}, \mathsf{com}, \mathsf{jcom}, \mathsf{jch}, \mathsf{jresp})$:
1: **parse** $\mathsf{com} = (\{E_i^\alpha\}_{i\in[n]}, \{E_i^\beta\}_{i\in[n]}, \tau(\{E_i^\gamma\}_{i\in[n]}), E^{\mathsf{Open}}, E^{\mathsf{Check}})$
2: **parse** $\mathsf{jcom} = (E^{\mathsf{Judge}}, E_m^{b'})$
3: **if** $\mathsf{jch} = 0$ **then**
4: **check** $E^{\mathsf{Judge}} = b' E^{\mathsf{Open}}$
5: **check** $E_m^{b'} = b' E_m$
6: **if** $\mathsf{jch} = 1$ **then**
7: **check** $E^{\mathsf{Judge}} = l' E_k^\beta$
8: **check** $E_m^{b'} = l' E$
9: **return** 1 **if all check pass**

For each run of **Commit** $\to (\mathsf{com}, \mathsf{st})$, we have to do additionally ι repetitions of **JCommit** (and thus ιt repetitions in total) to confirm that it is opened to the k-th signer with $\mathsf{negl}(\iota)$ probability. Similar as before, the Fiat-Shamir transform is applied for non-interactivity as follows.

– **Open**$(\mathsf{msk}, S = \{\mathsf{pk}_i\}_{i\in[n]}, m, \sigma)$
1: $t = t(\lambda, |S|)$; $\iota = \lambda$
2: **parse** $\sigma = (\mathsf{com}, \mathsf{resp})$
3: $\forall j \in [t], \mathsf{out}_j \leftarrow \Sigma_{GA}.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}_i)$
4: $\forall (i,j) \in [\iota] \times [t], \mathsf{jcom}_{i,j} \leftarrow \Sigma_{GA}.\mathbf{JCommit}(\mathsf{msk}, \{E_i\}_{i\in[n]}, \mathsf{com}_j)$
5: $\mathsf{jch} := \{\mathsf{jch}_{i,j}\}_{(i,j)\in[\iota]\times[t]} \leftarrow H(\sigma, \{\mathsf{jcom}_{i,j}\}_{(i,j)\in[\iota]\times[t]})$
6: $\forall (i,j) \in [\iota]\times[t], (\mathsf{jresp}_{i,j}, \mathsf{jst}_{i,j}) \leftarrow \Sigma_{GA}.\mathbf{JResp}(E_m, \{E_i\}_{i\in[n]}, \mathsf{jcom}_{i,j}, \mathsf{jch}_{i,j}, \mathsf{jst}_{i,j})$
7: $\mathsf{pk} = \mathbf{Maj}(\{\mathsf{out}_j\}_{j\in[t]})$ {**Maj** outputs the majority element of a set. In case of ties, **Maj** outputs a random choice of the marjority elements.}
8: $\pi := \{\mathsf{jcom}_{i,j}, \mathsf{jresp}_{i,j}\}_{(i,j)\in[\iota]\times[t]}$
9: **return** (pk, π)

– **Judge**$(\mathsf{mpk}, S = \{\mathsf{pk}_i\}_{i\in[n]}, \sigma, \mathsf{pk}, \pi)$:
1: **return** 0 **if** $\mathsf{pk} = \perp$
2: $t = 2\lambda|S|$; $\iota = \lambda$
3: **parse** $\sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$
4: **parse** $\pi = \{\mathsf{jcom}_{i,j}, \mathsf{jresp}_{i,j}\}_{(i,j)\in[\iota]\times[t]}$
5: $\mathsf{jch} := \{\mathsf{jch}_{i,j}\}_{(i,j)\in[\iota]\times[t]} \leftarrow H(\sigma, \{\mathsf{jcom}_{i,j}\}_{(i,j)\in[\iota]\times[t]})$

6: $\forall j \in [t], \mathsf{jout}_j \leftarrow \bigwedge_{i\in[\iota]} \Sigma_{GA}.\mathbf{JVerify}(\mathsf{mpk}, \{\mathsf{E}_i\}_{i\in[n]}, \mathsf{pk}, \mathsf{com}_j, \mathsf{jcom}_{i,j}, \mathsf{jch}_{i,j}, \mathsf{jresp}_{i,j})$
7: **return** 1 **if** $\sum_{j\in[t]} \mathsf{jout}_j \geq \lambda$

Here, a corrupted manager gets to selectively generate a partial proof $\{E_{\mathsf{out}_j}, \mathsf{jcom}_{i,j}, \mathsf{jch}_{i,j}, \mathsf{jresp}_{i,j}\}_{(i,j)\in[\iota]\times\mathcal{J}}$ where $\mathcal{J} \subseteq [t]$ is adaptively chosen. So long as we have $\sum_{j\in\mathcal{J}} \mathsf{jout}_j \geq \lambda$, the judged proof is accepted. This does not prevent the manager from generating accepted proofs that open to different members when $\#\{E_{\mathsf{out}_j}\} > 1$, which could happen if the corresponding signature is generated by multiple colluding signers. Otherwise, incriminating an honest non-signer would require to make up at least λ valid sessions of **Commit**, which will succeed with only negligible probability, i.e. for any PPT adversary A, any $t(\lambda, n) = n \cdot \mathsf{poly}(\lambda)$ and valid master key pair $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$,

$$\Pr[A \text{ wins } G_A^{\mathsf{JUF}}(\mathsf{mpk}, \mathsf{msk})] \leq \mathsf{negl}(\lambda),$$

where the *judging unforgeability game* G_A^{JUF} is as specified below.

G_A^{JUF}: Judging unforgeability game

1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathbf{Keygen}(1^\lambda)$
2: $(S, m^*, \sigma^*) \leftarrow A^{\mathbf{Sign}(\bullet,\bullet,\bullet,\mathsf{sk}),H}(\mathsf{pk})$
3: A wins if σ^* is not produced by querying $\mathbf{Sign}(\mathsf{mpk}, S^*, m^*, \mathsf{sk})$, $1 \leftarrow \mathbf{Verify}(\mathsf{mpk}, S, m^*, \sigma^*)$, $(\mathsf{out}, \pi) \leftarrow \mathbf{Open}(\mathsf{msk}, S, m, \sigma^*)$ satisfies $\mathsf{out} \in \{\mathsf{pk}, \perp\}$ and $1 \leftarrow \mathbf{Judge}(\mathsf{mpk}, S, \sigma, \mathsf{out}, \pi)$.

C Isogeny class group action

Here we briefly cover the basics for *elliptic curve isogenies*. For simplicity, we consider a working (finite) field $\mathbb{F}_q$ with characteristic $p > 3$. An isogeny ϕ between elliptic curves $E_1 \to E_2$ defined over an algebraic closure $\bar{\mathbb{F}}_q$ is a surjective homomorphism between the groups of rational points $E_1(\bar{\mathbb{F}}_q) \to E_2(\bar{\mathbb{F}}_q)$ with a finite kernel. If, additionally, ϕ is assumed *separable*, i.e. the induced extension of function fields $\phi^* : \bar{\mathbb{F}}_q(E_2) \hookrightarrow \bar{\mathbb{F}}_q(E_1)$ by $\bar{\mathbb{F}}_p(E_2) \ni f \mapsto f \circ \phi \in \bar{\mathbb{F}}_p(E_1)$ is separable, then for any finite subgroup $H \leq E_1(\bar{\mathbb{F}}_p)$, there is an isogeny $\phi : E_1 \to E_2$ having H as its kernel, and the co-domain curve is furthermore uniquely determined up to isomorphisms (in $\bar{\mathbb{F}}_q$). We refer to the co-domain curve as the *quotient curve*, denoted E_1/H. A corresponding isogeny could be computed using Velu's formula specified in [41], which works by expanding the coordinates

of $Q = \phi(P)$ as follows,

$$x(Q) = x(P) + \sum_{R \in H \setminus \{0\}} \left(x(P+R) - x(R)\right),$$
$$y(Q) = y(P) + \sum_{R \in H \setminus \{0\}} \left(y(P+R) - y(R)\right).$$

The separable degree $\deg_{\mathsf{sep}} \phi$ is defined as the separable degree for ϕ^*, which coincides with the size of its kernel $\#\ker\phi$, and since any isogeny could be acquired by precomposing Frobenius maps to a separable isogeny, i.e. of form $\phi \circ \pi_p^k$ where ϕ is separable, we can (equivalently) define the (full) degree $\deg\left(\phi \circ \pi_{p^k}\right) = \deg_{\mathsf{sep}}(\phi) p^k$. From now on, we will assume separability of isogenies unless otherwise specified, and therefore $\deg\phi = \deg_{\mathsf{sep}}\phi$ in this case.

For large degree ϕ, when both domain E_1 and co-domain E_2 (supersingular) curves are prescribed, it could be hard to determine the kernel (and thus ϕ). The current best-known (generic) quantum algorithm is *claw finding*, which takes $\tilde{O}\left(\deg(\phi)^{1/3}\right)$ operations.

One important structure for isogenies is the so-called *isogeny class group action*, which was first used for cryptographic constructions by [17, 38], and was viewed as a weaker alternative for discrete logarithm. However, although theoretically feasible, the instantiated group action used to rely heavily on techniques regarding the so-called *modular polynomials*, which is computationally expensive in practice. Later on, improvements in the *Commutative SIDH* (CSIDH) [12] scheme got rid of these techniques. Concretely, the space X is instantiated as a set $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p) = \{E/\mathbb{F}_p \text{ supersingular elliptic curves}\}/\cong_{\mathbb{F}_p}$ acted by their ideal class group $\mathsf{Cl}(\mathcal{O})$ of the $\mathbb{F}_p$-rational endomorphism ring $\mathcal{O} = \mathsf{End}_p(E)$ where $E \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ but $\mathcal{O} \otimes \mathbb{Q}$ tensored as a $\mathbb{Z}$-module is identical regardless of the choice of $E \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ thus so is $\mathsf{Cl}(\mathcal{O})$. The additional parameter π_p denotes the p-power Frobenius $\pi_p : (x, y) \mapsto (x^p, y^p)$. Elements of $\mathsf{Cl}(\mathcal{O})$ are equivalence classes $\mathfrak{a}$ of ideals of the (partial) endomorphism ring $\mathcal{J} \lhd \mathsf{End}_p(\mathcal{O})$. Any such ideal class $\mathfrak{a} \in \mathsf{Cl}(\mathcal{O})$ therefore acts on the curves by sending $E \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ to the quotient curve $\mathfrak{a} \cdot E := E/E[\mathcal{J}]$ where $\mathcal{J} \in \mathfrak{a}$ is a representative of the equivalence class $\mathfrak{a}$ and $E[\mathcal{J}] = \bigcap_{f \in \mathcal{J}} \ker f$ is the simultaneous kernel of $\mathcal{J}$.

The working base field $\mathbb{F}_p$ for CSIDH is carefully selected such that $p = 4\ell_1 \cdots \ell_n - 1$ where each $\ell_i > 2$ is a small prime generally referred to as an *Elkies prime*. This allows one to generate a heuristically large enough sub-covering $\{\mathfrak{l}_1^{e_1} \ldots \mathfrak{l}_n^{e_n} | \forall i : |e_i| \leq b_i\}$ of $\mathsf{Cl}(\mathcal{O})$ where each prescribed b_i is small[8] and each $\mathfrak{l}_i^{\pm 1}$ is the class of ideal $\langle \pi_p \mp 1, \ell_i \rangle$. The indices $(e_1, \ldots, e_n)$ thus represent the ideal class $\mathfrak{l}_i^{e_1} \cdots \mathfrak{l}_n^{e_n}$, making it easier to compute the co-domain curve. In particular, for a curve $E \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ and any choice of ℓ_i, the curve $\mathfrak{l}_i \cdot E := E/E[\langle \pi_p -$

[8] For CSIDH-512 [12] proposes $b_1 = \cdots = b_n = 5$.

$1, \ell_i\rangle]$ is computed by sampling a generator of the kernel,

$$E[\langle \pi_p - 1, \ell_i \rangle] = E(\mathbb{F}_p)[\ell_i] = \{P \in E(\mathbb{F}_p) | \ell_i P = 0\},$$

which is a one dimensional $\mathbb{Z}/\ell$-linear eigen-subspace of π_p within the ℓ_i-torsion $E[\ell_i]$. For the opposite direction, one can compute $\mathfrak{l}_i^{-1} \cdot E = (\mathfrak{l}_i \cdot E^t)^t$ where the superscript ^{t} is referred to as the quadratic twist of the specified curve, by taking the convention that the curve is fixed when its j-invariant is 1728, or equivalently, this can be done by sampling from the other $\mathbb{Z}/\ell_i$-linear eigen-subspace of π_p in $E[\ell_i]$, which sits in the quadratic extension $E(\mathbb{F}_{p^2})$.

We also list here some well-known properties for the considered class group action. First, the class group $\mathsf{Cl}(\mathcal{O})$ *commutes*, which is a direct result of the fact that the $\mathbb{F}_p$-rational endomorphism ring $\mathsf{End}_p(E)$ commutes. Second, as noted in [12, Theorem 7], $\mathsf{Cl}(\mathcal{O})$ acts *freely and transitively* on $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$, which means that for all $E_1, E_2 \in \mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$, there exists a unique $\mathfrak{a} \in \mathsf{Cl}(\mathcal{O})$ such that $\mathfrak{a} \cdot E_1 = E_2$. Finally, elements in $\mathcal{E}\ell\ell_p(\mathcal{O}, \pi_p)$ can be *efficiently verified*. We note that a curve E is supersingular if and only if it has $p+1$ points over $\mathbb{F}_p$. This can be efficiently tested by finding some $P \in E(\mathbb{F}_p)$ with order $\mathsf{ord}(P) \geq 4\sqrt{p}$ dividing $p+1$. A random point P sampled from $E(\mathbb{F}_p)$ satisfies such a condition with high probability if E is supersingular, and whether it does can be verified efficiently as follows. If $(p+1)P \neq 0$, then $\mathsf{ord}(P)$ does not divide $p+1$ and E is ordinary. Otherwise, we can perform the so-called *batch co-factor multiplication* computing $P_i = \frac{p+1}{\ell_i} P$ for each i, by using convention that $\ell_0 = 4$. This allows us to determine $\mathsf{ord}(P) = \prod_i \mathsf{ord}(P_i)$.

For typical cryptographic constructions such as CSIDH, additional heuristic assumptions are required to sample a random element from the class group (as in Definition 2). This is because the "CSIDH-way" for doing this is by sampling exponents $(e_1, \ldots, e_n)$ satisfying $\forall i : |e_i| \leq b_i$, and the resulting distribution for ideals $\mathfrak{l}_1^{e_1} \ldots \mathfrak{l}_n^{e_n}$ is generally non-uniform within $\mathsf{Cl}(\mathcal{O})$. To get rid of such heuristics, one could instead work with specific parameters, where a bijective (yet efficient) representation of ideals is known. For instance, in [6], the structure of $\mathsf{Cl}(\mathcal{O})$ is computed, including a full generating set of ideals $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ and the entire lattice $\Lambda := \{(e_1, \ldots, e_n) | \mathfrak{l}_1^{e_1} \ldots \mathfrak{l}_n^{e_n} = \mathsf{id}\}$. Evaluating the group action is just a matter of approximating a *closest vector* and then evaluating the residue as in CSIDH. In this work, we will be working with such a "perfect" representation of ideals, unless otherwise specified.

As a remark, we note that the D-CSIDH problem for characteristic $p = 1 \mod 4$ is known to be broken [13]. Nevertheless, the attack is not applicable to the standard CSIDH setting where $p = 3 \mod 4$.

D Sigma protocol

A sigma protocol should satisfy the following three properties.

Definition 13. *(Correctness) A sigma protocol is correct if for any $(x, w) \in R$, the probability*

$$\Pr\left[(\mathsf{com}, st) \leftarrow P_1(x, w), \mathsf{ch} \xleftarrow{\$} \mathcal{C}, \mathsf{resp} \leftarrow P_2(st, \mathsf{ch}), 0 \leftarrow V(x, \mathsf{com}, \mathsf{ch}, \mathsf{resp})\right]$$

is negligible.

Definition 14. *(Honest Verifier Zero Knowledge/HVZK) Let* $\mathbf{Trans}(x, w) \rightarrow (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ *be a function that honestly executes the sigma protocol and outputs a transcript. We say that the sigma protocol is HVZK if there exists a simulator* $\mathbf{Sim}(x) \rightarrow (\mathsf{com}, \mathsf{ch}, \mathsf{resp})$ *such that the output distribution of* $\mathbf{Trans}(x, w)$ *and* $\mathbf{Sim}(x)$ *is indistinguishable.*

Definition 15. *(μ-special soundness) A sigma protocol is μ-special sound if there exist an efficient extractor* $\mathbf{Ext}$ *such that, for any set of μ transcripts with the same (x, com), denoted as $(x, \mathsf{com}, \{\mathsf{ch}_i\}_{i\in[\mu]}, \{\mathsf{resp}_i\}_{i\in[\mu]})$, where every ch_i is distinct, the probability*

$$\Pr\left[(x, s) \notin R \wedge \forall i \in [\mu], \mathsf{acc}_i = 1 : \begin{smallmatrix}\forall i\in[\mu],\ \mathsf{acc}_i \leftarrow V(x,\mathsf{com},\mathsf{ch}_i,\mathsf{resp}_i),\\ s\leftarrow\mathbf{Ext}(x,\mathsf{com},\{\mathsf{ch}_i\}_{i\in[\mu]},\{\mathsf{resp}_i\}_{i\in[\mu]})\end{smallmatrix}\right]$$

is negligible.

Here, we formulate a more general form of special soundness. While most sigma protocol constructions in the literature adopt 2-special soundness, any μ-special sound protocol with constant μ can be similarly transformed into a signature scheme, simply by applying more rewinding trials.

E Analysis in classical ROM

E.1 The forking lemma

The concept of the forking lemma is as follows. In the random oracle model, let A be an adversary that can with non-negligible probability generate valid transcripts $(m, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$ with $\mathsf{ch} = H(m, \mathsf{com})$. Since H is a random oracle, for some (m, com), A should be able to succeed on sufficiently many different ch' from H in order to achieve an overall non-negligible success probability. If we can rewind and rerun A with different oracle outputs on $H(m, \mathsf{com})$, we should be able to get multiple accepting transcripts.

To dig a little bit deeper, we can construct an efficient algorithm B that runs A as a subroutine, where $A \rightarrow (m, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$ has at most Q oracle queries. The tuple (m, com) should, with all but negligible probability, be among one of the Q queries. B first guesses the *critical query* $i \in [Q]$, the index where $\mathcal{Q}_i = (m, \mathsf{com})$ is being queried. Then, B replays A with fixed random tape, fixed oracle outputs for the first $i - 1$ queries, and fresh random oracle outputs

for the remaining queries. If the query guess i and fixed randomness are "good," which should happen with non-negligible probability, then among sufficiently many retries we should get t successful outputs of A, which are transcripts with identical (m, com) with distinct challenges ch's. For a rigorous proof, we refer the reader to [33, 34] for the forking lemma with 2 transcripts and [10] for a μ-transcript version.

Here, we give a reformulated version of the improved forking lemma proposed by [10]. We renamed the variables to fit our notion and restricted parameters to the range that is sufficient for our proof.

Theorem 4. *(The Improved Forking Lemma [10], Reformulated) Let A be a probabilistic polynomial-time algorithm and Sim be a probabilistic polynomial-time simulator which can be queried by A. Let H be a random oracle with image size $|H| \geq 2^\lambda$. If A can output some valid tuple $(m, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$ with non-negligible probability $\varepsilon \geq 1/poly(\lambda)$ within less than Q queries to the random oracle, then with $O(Q\mu \log \mu/\varepsilon)$ rewinds of A with different random oracles, A will, with at least constant probability, output μ valid tuples $(m, \mathsf{com}, \mathsf{ch}_i, \mathsf{resp}_i)$ with identical (m, com) and pairwise distinct ch_i's.*

E.2 Proof of Lemma 9

Proof. Assume that there exists an efficient adversary A' that wins $G'^{\mathsf{UF}}_{A'}(\mathsf{mpk}, \mathsf{msk})$ on some valid key pair $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$ with non-negligible probability. We aim to show that we can construct some algorithm B which runs A' as a subroutine and extract an un-corrupted secret key.

As it doesn't hurt for a signing oracle to produce the challenges, let's abuse the notation as say the signing oracle returns not only the signature, but also those corresponding challenges. First, we replace the **Sign** oracle with a simulator, so that $A^H := A'^{\mathbf{Sim},H}$ can emulate the oracle responses to A'. We consider a modified game $G'^{\mathsf{UF},1}_{A'}$ which replaces the signing oracle $\mathbf{Sign}(\bullet, \bullet, \bullet, \mathsf{sk})$ by a simulator **Sim** defined as follows:

- $\mathbf{Sim}(\mathsf{mpk}, S, m, \mathsf{pk} \in S)$:
  1: $t := t(\lambda, |S|)$
  2: for $j \in [t]$, $(\mathsf{com}_j, \mathsf{ch}_j, \mathsf{resp}_j) \leftarrow \Sigma_{GA}.\mathbf{Sim}(\mathsf{mpk}, S, \mathsf{pk} \in S)$
  3: **program** $H(m, S, \mathsf{com}_1, \ldots, \mathsf{com}_t) := (\mathsf{ch}_1, \ldots, \mathsf{ch}_t)$
  4: **return** $\sigma = (\mathsf{com}, \mathsf{resp}) := ((\mathsf{com}_1, \ldots, \mathsf{com}_t), (\mathsf{resp}_1, \ldots, \mathsf{resp}_t))$

Since $\Sigma.\mathbf{Sim}$ is a statistical HVZK simulator, any adversary with $Q = \mathsf{poly}(\lambda)$ queries to H cannot distinguish **Sign** from **Sim** with non-negligible probability. Without loss of generality we assume $A'^{\mathbf{Sim},H}$ never produces a signature σ^* from previous queries to $\mathbf{Sim}(\mathsf{mpk}, S^*, m^*, \mathsf{pk})$. Therefore for an forgery $(\mathsf{com}^*, \mathsf{resp}^*) \leftarrow A'^{\mathbf{Sim},H}$ accepted with respect to the potentially reprogrammed H, $H(S^*, m^*, \mathsf{com}^*)$ must have not been reprogrammed, otherwise there must

be a prior query of form $(\mathsf{com}^*, \mathsf{resp}) \leftarrow \mathbf{Sim}(\mathsf{mpk}, S^*, m^*, \mathsf{pk})$ for some resp, but then $\mathsf{resp} \neq \mathsf{resp}^*$ is hard to find due to the computational unique-response property. Thus, A should also win $G_A^{\mathsf{UF},1}$ with non-negligible probability.

Now, since A wins $G_A^{\mathsf{UF},1}(\mathsf{mpk}, \mathsf{msk})$ only if it outputs some (R, m^*, σ^*) such that $\mathsf{out}^* \leftarrow \mathsf{Open}(\mathsf{msk}, R, m, \sigma^*)$ satisfies $\mathsf{out}^* = \mathsf{pk}$ or $\mathsf{out}^* = \perp$, either A wins with non-negligible probability with $\mathsf{out}^* = \perp$, or A wins with non-negligible probability with $\mathsf{out}^* = \mathsf{pk}$. We deal with these cases separately.

We first prove that there cannot exist efficient $A_\perp$ that wins $G_A^{\mathsf{UF},1}(\mathsf{mpk}, \mathsf{msk})$ with non-negligible probability with $\mathsf{out}^* = \perp$. If such $A_\perp$ exists, we can construct an algorithm B that honestly generates $(\mathsf{pk}, \mathsf{sk})$ and runs $A_\perp(\mathsf{pk})$. With non-negligible probability, $A_\perp$ will output valid $(S, m, \sigma = (\mathsf{com}, \mathsf{ch}, \mathsf{resp}))$ such that $\perp \leftarrow \mathbf{Open}(\mathsf{msk}, S, m, \sigma)$. By applying the improved forking lemma (Theorem 4), with $r = O(Q/\varepsilon)$ rewinds of $A_\perp$, it would, with constant probability, output μ valid signatures $(S, m, \sigma^1, \ldots, \sigma^\mu)$ with identical com and pairwise distinct ch^c, and that $\perp \leftarrow \mathbf{Open}(\mathsf{msk}, S, m, \sigma^c)$ for all $c \in [\mu]$. We now claim that with high probability, we can find some parallel session $j \in [t]$ such that $\perp \leftarrow \Sigma.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}_j)$ and $\mathsf{ch}_j^1, \ldots, \mathsf{ch}_j^\mu$ are distinct. Note that this is not trivially true, as the forking lemma only promises that $\mathsf{ch}^1, \ldots, \mathsf{ch}^\mu$ are pairwise distinct as vectors, so they might not be pairwise distinct on any index j.

Let T be the set of indices j where $\perp \leftarrow \Sigma.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}_j)$. Since $\perp \leftarrow \mathbf{Open}(\mathsf{msk}, S, m, \sigma)$, by the definition of $\mathbf{Open}$, $\perp$ must be (one of) the majority output among the t parallel sessions. Thus $|T| \geq t/(|S|+1) \geq \lambda$. We say that μ challenges $\mathsf{ch}'^1, \ldots, \mathsf{ch}'^\mu$ are **good** on T if there exists some $j \in T$ such that $\mathsf{ch}_j'^1, \ldots, \mathsf{ch}_j'^\mu$ are distinct. For μ independently random challenges in $[\mu]^t$, the probability that they are good on T is $1 - (1 - (\mu!/\mu^\mu))^{|T|} = 1 - \mathsf{negl}(\lambda)$.

Unfortunately, the challenges $\mathsf{ch}^1, \ldots, \mathsf{ch}^\mu$ obtained from rewinding A are not necessarily independent. To cope with this, we will need the fact that in each rewind of A, the *valid* ch is a new random output from the new random oracle H. Thus, the finally output μ-tuple $\mathsf{ch}^1, \ldots, \mathsf{ch}^\mu$ must be a subset of $r = O(Q/\varepsilon)$ independent random samples from $[\mu]^t$. By the union bound, the probability that *all* μ-tuples in the r samples are good on T is $1 - \binom{r}{\mu}\mathsf{negl}(|T|) \geq 1 - \mathsf{negl}(\lambda)$. Thus we can find $j \in T$ such that $\mathsf{ch}_j^1, \ldots, \mathsf{ch}_j^\mu$ are distinct with probability $1 - \mathsf{negl}(\lambda)$.

For such j, we without loss of generality let $(\mathsf{ch}_j^1, \ldots, \mathsf{ch}_j^\mu) = (1, \ldots, \mu)$ and consider $(S, \mathsf{com}_j, \mathsf{resp}_j^1, \ldots, \mathsf{resp}_j^\mu)$. Now B achieves $\forall c \in [\mu], 1 \leftarrow \Sigma.\mathbf{Verify}(S, \mathsf{com}_j, c, \mathsf{resp}_j^c)$, and $\perp \leftarrow \Sigma.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}_j)$. Thus B violates the μ-special soundness property of Σ and brings a contradiction. Hence such $A_\perp$ cannot exist.

Now we consider the case where some A wins $G_A^{\mathsf{UF},1}(\mathsf{mpk}, \mathsf{msk})$ with non-negligible probability with $\mathsf{out}^* = \mathsf{pk}$.

For such A, we can similarly construct an algorithm B that runs A with input pk. Then again by applying the improved forking lemma, with the same probability, $r = O(Q/\varepsilon)$ rewinds of A will output μ valid signatures $(S, m, \sigma^1, \ldots, \sigma^\mu)$ with identical com and pairwise distinct ch^c, so that $\mathsf{pk} \leftarrow \mathbf{Open}(\mathsf{msk}, S, m, \sigma^c)$

for all $c \in [\mu]$. Again by the same argument as in the case of $A_\perp$, we can with high probability find some $j \in [t]$ such that $\mathsf{pk} \leftarrow \Sigma_{GA}.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}_j)$ and $\mathsf{ch}_j^1, \ldots, \mathsf{ch}_j^\mu$ are distinct.

Now, without loss of generality let $(\mathsf{ch}_j^1, \ldots, \mathsf{ch}_j^\mu) = (1, \ldots, \mu)$ and consider $(S, \mathsf{com}_j, \mathsf{resp}_j^1, \ldots, \mathsf{resp}_j^\mu)$. We have $\forall c \in [\mu], 1 \leftarrow \Sigma_{GA}.\mathbf{Verify}(S, \mathsf{com}_j, c, \mathsf{resp}_j^c)$, and that the challenge statement $\mathsf{pk} \leftarrow \Sigma_{GA}.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}_j)$. Thus by the μ-special soundness property of Σ_{GA}, we can extract the matching secret key $\mathsf{sk} \leftarrow \Sigma_{GA}.\mathbf{Ext}(S, \mathsf{com}_j, \mathsf{resp}_j^1, \ldots, \mathsf{resp}_j^\mu)$, such that $(\mathsf{pk}, \mathsf{sk}) \in R$.

From the previous arguments, we see that if such efficient A exists, then we can obtain an algorithm B based on A that, on inputting random $\mathsf{pk} \in \mathcal{PK}$, output sk such that $(\mathsf{pk}, \mathsf{sk}) \in R$ with non-negligible probability. Thus, we successfully construct a secret extractor from adversary A that wins the unforgeability game, which concludes the proof that our $\mathcal{ARS}_\Sigma$ is unforgeable assuming the instance relations are hard (to extract witness) for Σ.

□

F Analysis in QROM

F.1 Proof of Lemma 14

Proof. We adopt the generalized Unruh's rewinding, as described in [18, Lemma 29]. Let $B(\mathsf{pk})$ run as follows. First, execute $(S, m, \mathsf{com}, \mathsf{st}_0) \leftarrow A(\mathsf{pk})$ as usual. Then, perform the following computation for μ times. For the jth time, freshly sample a challenge $\mathsf{ch}_j \leftarrow \Sigma^{\otimes t}.\mathcal{C}$ and then produce $\mathsf{resp}_j \leftarrow A(\mathsf{st}_{j-1}, \mathsf{ch}_j)$, where the computation is *projectively executed*, i.e. after resp_j is produced, the computation is rewinded to where it started with st_{j-1}, but with the internal state collapsed to st_j for the next run. After μ trials of rewinding, B obtains μ samples of transcripts $\mathsf{com}, \{\mathsf{ch}_j, \mathsf{resp}_j\}_{j\in[\mu]}$ sharing the same first message com. Denote $\mathsf{com}^i, \mathsf{ch}_j^i, \mathsf{resp}_j^i$ to be the ith repetition of the jth rewinding. If there is some repetition (the ith) such that the corresponding transcript $(\mathsf{com}^i, \mathsf{ch}_j^i, \mathsf{resp}_j^i)$ are distinct valid responses opened to pk or $\perp$ for all $j \in [\mu]$, then output $s \leftarrow \Sigma.\mathbf{Ext}(S, \mathsf{com}^i, \{\mathsf{ch}_j^i\}_{j\in\mu}, \{\mathsf{resp}_j^i\}_{j\in[\mu]})$, and abort otherwise.

As described earlier, by (1), we know that the output s of $B(\mathsf{pk})$ is always such that $(\mathsf{pk}, s) \in R$ whenever B does not abort. Let out, out^i and acc_j respectively be the output of $\Sigma^{\otimes t}.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com})$, $\Sigma.\mathbf{Open}(\mathsf{msk}, S, \mathsf{com}^i)$ and $\Sigma^{\otimes t}.\mathbf{Verify}(\mathsf{mpk}, S, \mathsf{com}, \mathsf{ch}_j, \mathsf{resp}_j)$.

The non-abort probability of B can be union-bounded by two parts, namely

$$\Pr[B(\mathsf{pk}) \text{ non-abort}] \geq \Pr\left[\begin{smallmatrix}\mathsf{out}\in\{\mathsf{pk},\perp\} \text{ and} \\ \forall j\in[\mu]:\mathsf{acc}_j=1\end{smallmatrix}\right] - \Pr\left[\begin{smallmatrix}\forall i\in[\mu]:\mathsf{out}^i\notin\{\mathsf{pk},\perp\} \text{ or} \\ \mathsf{ch}_1^i,\ldots,\mathsf{ch}_\mu^i \text{ not distinct}\end{smallmatrix}\middle|\mathsf{out}\in\{\mathsf{pk},\perp\}\right].$$

We bound $\Pr\left[\begin{smallmatrix}\mathsf{out}\in\{\mathsf{pk},\perp\} \text{ and} \\ \forall j\in[\mu]:\mathsf{acc}_j=1\end{smallmatrix}\right]$ first. For every fixed choice $x^\circ := (\mathsf{pk}^\circ, S^\circ, \mathsf{com}^\circ)$ of the joint random variable $x := (\mathsf{pk}, S, \mathsf{com})$, identifying the (mixed) state of

st_0 conditioned on $x = x^\circ$ as ρ_{x°, and thus the un-conditioned state would be $\rho := \sum_{x^\circ} \Pr[x = x^\circ]\rho_{x^\circ}$. Due to the perfect unique-response property, every time when a valid response resp_j is produced, it only disturbs the running state as a projector. Thus, we can define the projective measurement $\{P^{x^\circ}_{\mathsf{ch}^\circ}\}_{\mathsf{ch}^\circ \in \Sigma^{\otimes t}.\mathcal{C}}$ where each projector $P^{x^\circ}_{\mathsf{ch}^\circ}$ on input st_{j-1} serve as the predicate that $(\mathsf{com}^\circ, \mathsf{ch}_j, \mathsf{resp}_j)$ is an accepted transcript, i.e. $1 \leftarrow \Sigma^{\otimes t}.\mathbf{Verify}(\mathsf{mpk}, S^\circ, \mathsf{com}^\circ, \mathsf{ch}_j, \mathsf{resp}_j)$. Then

$$\Pr\left[\begin{matrix}\forall j\in[\mu]:\\ 1\leftarrow \Sigma^{\otimes t}.\mathbf{Verify}(\mathsf{mpk},S^\circ,\mathsf{com}^\circ,\mathsf{ch}_j,\mathsf{resp}_j)\end{matrix}\middle| x = x^\circ\right] = \sum_{\mathsf{ch}^\circ_1,\ldots,\mathsf{ch}^\circ_\mu \in \Sigma^{\otimes t}.\mathcal{C}} \mathsf{tr}\left(P^{x^\circ}_{\mathsf{ch}^\circ_1} \ldots P^{x^\circ}_{\mathsf{ch}^\circ_\mu} \rho_{x^\circ}\right) .$$

Expanding $\rho_{x^\circ} := \sum_i \alpha_i |\psi_i\rangle\langle\psi_i|$ via singular-value decomposition, we get

$$\sum_{\mathsf{ch}^\circ_1,\ldots,\mathsf{ch}^\circ_\mu \in \Sigma^{\otimes t}.\mathcal{C}} \mathsf{tr}\left(P^{x^\circ}_{\mathsf{ch}^\circ_1} \ldots P^{x^\circ}_{\mathsf{ch}^\circ_\mu} \rho_{x^\circ}\right) = \sum_i \alpha_i \sum_{\mathsf{ch}^\circ_1,\ldots,\mathsf{ch}^\circ_\mu \in \Sigma^{\otimes t}.\mathcal{C}} \left\| P^{x^\circ}_{\mathsf{ch}^\circ_1} \ldots P^{x^\circ}_{\mathsf{ch}^\circ_\mu} |\psi_i\rangle \right\|^2$$

$$\geq \sum_i \alpha_i \left(\sum_{\mathsf{ch}^\circ \in \Sigma^{\otimes t}.\mathcal{C}} \left\| P^{x^\circ}_{\mathsf{ch}^\circ} |\psi_i\rangle \right\|^2 \right)^{2\mu-1} \geq \left(\sum_{\mathsf{ch}^\circ \in \Sigma^{\otimes t}.\mathcal{C}} \mathsf{tr}\left(P^{\mathsf{pk}^\circ}_{\mathsf{ch}^\circ} \rho_{x^\circ}\right) \right)^{2\mu-1} ,$$

where the first inequality is by [18, Lemma 29] and the second inequality is by Jensen's inequality. Summing over $x^\circ = (\mathsf{pk}^\circ, S^\circ, \mathsf{com}^\circ)$ such that $\{\mathsf{pk}^\circ, \bot\} \ni \mathsf{out}^\circ := \Sigma^{\otimes t}.\mathbf{Open}(\mathsf{msk}, S^\circ, \mathsf{com}^\circ)$ with suitable probability, we obtain

$$\Pr\left[\begin{matrix}\mathsf{out}\in\{\mathsf{pk},\bot\} \text{ and}\\ \forall j\in[\mu]:\mathsf{acc}_j=1\end{matrix}\right] \geq \sum_{x^\circ \text{ s.t. } \mathsf{out}^\circ\in\{\mathsf{pk}^\circ,\bot\}} \Pr[x = x^\circ] \left(\sum_{\mathsf{ch}^\circ \in \Sigma^{\otimes t}.\mathcal{C}} \mathsf{tr}\left(P^{x^\circ}_{\mathsf{ch}^\circ} \rho_{x^\circ}\right) \right)^{2\mu-1}$$

$$\geq \left(\sum_{\substack{x^\circ \text{ s.t. } \mathsf{out}^\circ\in\{\mathsf{pk}^\circ,\bot\}\\ \mathsf{ch}^\circ\in\Sigma^{\otimes t}.\mathcal{C}}} \Pr[x = x^\circ] \mathsf{tr}\left(P^{x^\circ}_{\mathsf{ch}^\circ} \rho_{x^\circ}\right) \right)^{2\mu-1} = \Pr\left[A \text{ wins } G^{\mathsf{int}}_A(\mathsf{mpk}, \mathsf{msk})\right]^{2\mu-1} ,$$

where the second inequality is again via Jensen's inequality.

Next, for every $x^\circ = (\mathsf{pk}^\circ, S^\circ, \mathsf{com}^\circ)$ we define

$$\mathcal{I}_{x^\circ} := \left\{ i \in [\mu] \,\middle|\, \mathsf{pk}^\circ \text{ or } \bot \leftarrow \Sigma.\mathbf{Open}(\mathsf{msk}, S^\circ, \mathsf{com}^\circ) \right\} ,$$

in order to bound $\Pr\left[\begin{matrix}\forall i\in[\mu]:\mathsf{out}^i\notin\{\mathsf{pk},\bot\} \text{ or}\\ \mathsf{ch}^i_1,\ldots,\mathsf{ch}^i_\mu \text{ not distinct}\end{matrix}\middle| \mathsf{out} \in \{\mathsf{pk}, \bot\}\right]$

$$\leq \max_{x^\circ \text{ s.t. } \mathsf{out}^\circ\in\{\mathsf{pk}^\circ,\bot\}} \Pr\left[\begin{matrix}\forall i\in\mathcal{I}_{x^\circ}:\\ \mathsf{ch}^i_1,\ldots,\mathsf{ch}^i_\mu \text{ not distinct}\end{matrix}\middle| x = x^\circ\right] = \max_{x^\circ \text{ s.t. } \mathsf{out}^\circ\in\{\mathsf{pk}^\circ,\bot\}} \Pr\left[\begin{matrix}\forall i\in\mathcal{I}_{x^\circ}:\\ \mathsf{ch}^i_1,\ldots,\mathsf{ch}^i_\mu \text{ not distinct}\end{matrix}\right] ,$$

where the last equality is due to the freshly sampled $\{\mathsf{ch}^i_j\}_{(i,j)\in[\mu]\times[\mu]}$ being independent with x. Note that, by the pigeonhole principle, $\mathsf{out}^\circ \in \{\mathsf{pk}^\circ, \bot\}$

implies $\#\mathcal{I}_{x^\circ} \geq \kappa$, thus the above can be bounded by

$$\leq \left(1 - \frac{\binom{C}{\mu}}{C^\mu}\right)^\kappa \leq \exp\left(\frac{-\kappa}{\mu^\mu}\right) ,$$

where $C := \#\Sigma.\mathcal{C}$ is the size of the challenge space.

Putting things together,

$$\begin{aligned}
&\Pr\left[\mathsf{sk} \leftarrow B(\mathsf{pk})\right] \geq \Pr[B(\mathsf{pk}) \text{ non-abort}] \\
&\geq \Pr\left[A \text{ wins } G_A^{\mathsf{int}}(\mathsf{mpk}, \mathsf{msk})\right]^{2\mu-1} - \exp\left(\frac{-\kappa}{\mu^\mu}\right) ,
\end{aligned}$$

we conclude the proof. □

F.2 Proof of Lemma 15

Proof. Let $t(\lambda, n) = (n+1)\kappa$ where $\kappa = \mathsf{poly}(\lambda)$ and A be an efficient quantum adversary against $G_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})$, making at most q queries to the random oracle H. By Lemma 12, 13, 14, we know that for every $(\mathsf{mpk}, \mathsf{msk}) \in \mathcal{KP}_m$, there exists efficient quantum adversaries A_1, A_2, A_3 respectively such that

$$\begin{aligned}
&\Pr\left[A \text{ wins } G_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})\right] \leq \Pr\left[A_1 \text{ wins } \widetilde{G}_A^{\mathsf{UF}}(\mathsf{mpk}, \mathsf{msk})\right] + \mathsf{negl}(\lambda) \\
&\leq \Pr\left[A_2 \text{ wins } \widetilde{G}_A^{\mathsf{int}}(\mathsf{mpk}, \mathsf{msk})\right] (2q+1)^2 + \mathsf{negl}(\lambda) \\
&\leq \left(\Pr\left[\substack{(\mathsf{pk},\mathsf{sk}) \leftarrow R(1^\lambda) \\ \mathsf{sk} \leftarrow A_3(\mathsf{pk})}\right] + \exp\left(\frac{-\kappa}{\mu^\mu}\right)\right)^{\frac{1}{2\mu-1}} (2q+1)^2 + \mathsf{negl}(\lambda) .
\end{aligned}$$

By assumption R is hard and $\kappa \geq \mathsf{poly}(\lambda)$, making the right-most term negligible. This concludes the proof. □

G Group signature

A group signature scheme consists of one manager and n parties. The manager can set up a group and provide secret keys to each party. Every party is allowed to generate signatures on behalf of the whole group. Any party can verify the signature for the group without knowing the signer, while the manager party can open the signer's identity with his master secret key.

Syntax. A group signature scheme $\mathcal{GS}$ consists of the following four algorithms.

- **GKeygen**$(1^\lambda, 1^n) \to (\mathsf{gpk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk})$: The key generation algorithm **GKeygen** takes 1^λ and 1^n as inputs where λ is the security parameter and $n \in \mathbb{N}$ is the number of parties in the group, and outputs $(\mathsf{gpk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk})$

where gpk is the public key for the group, sk_i being the secret key of the i-th player for each $i \in [n]$, and msk is the master secret key held by the manager for opening.

- $\mathbf{GSign}(\mathsf{gpk}, m, \mathsf{sk}_k) \to \sigma$: The signing algorithm **GSign** takes a secret key sk_k and a message m as inputs, and outputs a signature σ of m using sk_k.
- $\mathbf{GVerify}(\mathsf{gpk}, m, \sigma) \to y \in \{0, 1\}$: The verification algorithm **GVerify** takes the public key gpk, a message m, and a candidate signature σ as inputs, and outputs either 1 for accept or 0 for reject.
- $\mathbf{GOpen}(\mathsf{gpk}, \mathsf{msk}, m, \sigma) \to k \in [n]$: The open algorithm **GOpen** takes the public key gpk, the manager's master secret key msk, a message m, and a signature σ as inputs, and outputs an identity k or abort with output $\perp$.

A group signature scheme should satisfy the following security properties.

Correctness. A group signature scheme is said to be correct if every honest signature can be correctly verified and opened.

Definition 16. *A group signature scheme $\mathcal{GS}$ is correct if for any tuple of keys $(\mathsf{gpk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathbf{GKeygen}(1^\kappa, 1^n)$, any $i \in [n]$ and any message m,*

$$\Pr\left[\mathsf{acc}{=}1 \wedge \mathsf{out}{=}i : \begin{array}{c}\sigma \leftarrow \mathbf{GSign}(\mathsf{gpk}, m, \mathsf{sk}_i),\\ \mathsf{acc} \leftarrow \mathbf{GVerify}(\mathsf{gpk}, m, \sigma),\\ \mathsf{out} \leftarrow \mathbf{GOpen}(\mathsf{gpk}, \mathsf{msk}, m, \sigma)\end{array}\right] > 1 - \mathsf{negl}(\lambda)$$

Anonymity. A group signature is said to be anonymous if no adversary can determine the signer's identity among the group of signers given a signature, without using the master's secret key (msk).

Definition 17. *A group signature scheme $\mathcal{GS}$ is anonymous if for any PPT adversary A and any $n = poly(\lambda)$,*

$$\left|\Pr[1 \leftarrow G_{A,0}^{\mathsf{Anon}}(\lambda, n)] - \Pr[1 \leftarrow G_{A,1}^{\mathsf{Anon}}(\lambda, n)]\right| \leq \mathsf{negl}(\lambda),$$

where the game $G_{A,b}^{\mathsf{Anon}}(\lambda, n)$ is defined below.

$G_{A,b}^{\mathsf{Anon}}(\lambda, n)$: Anonymity game

1: $(\mathsf{gpk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathbf{GKeygen}(1^\lambda, 1^n)$
2: $(st, i_0, i_1) \leftarrow A(\mathsf{gpk}, \{\mathsf{sk}_i\}_{i\in[n]})$
3: $b \leftarrow \{0, 1\}$
4: **return** $\mathsf{out} \leftarrow A^{\mathbf{GSign}(\mathsf{gpk}, \cdot, \mathsf{sk}_{i_b})}(st)$

Unforgeability. A group signature is said to be unforgeable if no adversary can forge a valid signature that fails to open or opens to some non-corrupted parties, even if the manager has also colluded.

Definition 18. *A group signature scheme $\mathcal{GS}$ is unforgeable if for any PPT adversary A and any $n = poly(\lambda)$,*

$$\Pr[A \text{ wins } G_A^{\mathsf{UF}}(\lambda, n)] < \mathsf{negl}(\lambda),$$

where the game $G_A^{\mathsf{UF}}(\lambda, n)$ is defined below.

$G_A^{\mathsf{UF}}(\lambda, n)$: Unforgeability game

1: $(\mathsf{gpk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathbf{GKeygen}(1^\lambda, 1^n)$, $\mathsf{Cor} = \{\}$
2: $(m^*, \sigma^*) \leftarrow A^{\mathbf{GSign}(\mathsf{gpk},\bullet,\mathsf{sk}_i\notin\mathsf{Cor}),\mathbf{Corrupt}(\bullet)}(\mathsf{gpk}, \mathsf{msk})$
$\{\mathbf{Corrupt}(i)$ returns sk_i stores query i in list $\mathsf{Cor}\}$
3: A wins if σ^* is not produced by querying $\mathbf{GSign}(\mathsf{gpk}, m^*, \mathsf{sk}_i \notin \mathsf{Cor})$, $1 \leftarrow \mathbf{GVerify}(\mathsf{gpk}, m^*, \sigma^*)$
and $i \leftarrow \mathbf{GOpen}(\mathsf{gpk}, \mathsf{msk}, m^*, \sigma^*)$ satisfies $i \notin \mathsf{Cor}$
