On the Multi-User Security of CSI-FiSh with Tight Reductions

Seunghoon Lee, Maher Mamah, and Bruno Sterner

University of Waterloo, Waterloo, Canada
{seunghoon.lee,mmamah,bsterner}@uwaterloo.ca

Abstract. The security of isogeny-based signatures is almost exclusively studied in the single-user setting, leaving a gap for realistic multi-user deployments. This gap is especially crucial for CSI-FiSh, where the small challenge space and parameter sensitivity directly impacts security estimates. We address the multi-user security of CSI-FiSh and obtain tight concrete classical multi-user bounds in the random-oracle model (ROM) plus generic group-action model (GGAM). Crucially, our analysis only incurs an *additive* degradation in the number of users rather than a multiplicative one. As a byproduct, we show that CSI-FiSh attains 125 bits of provable classical security even with 2^{46} users. We extend this analysis to preprocessing attacks, where an adversary with nation-state resources can perform offline precomputations. We capture this by extending the preprocessing security framework of Coretti et al. to ROM+GGAM and obtain concrete multi-user bounds for CSI-FiSh with preprocessing. Finally, we address the multi-user security of CSI-FiSh in the quantum random-oracle model and obtain results in the algebraic group action model which also avoids this multiplicative loss. This gives a unified concrete security treatment of plain CSI-FiSh in the deployment regimes where multi-user and preprocessing effects are unavoidable.

Keywords: Isogeny-Based Signatures · CSI-FiSh · Multi-User Security · Preprocessing Attacks · Quantum Random Oracle Model

1 Introduction

Replacing classical signature schemes has become a central goal in the transition to post-quantum cryptography. One notable candidate in isogeny-based cryptography is CSI-FiSh, viewed as a quantum-resistant analogue of Schnorr signatures. Its underlying hardness assumption is the *Group Action Inverse Problem* (GAIP): given two oriented supersingular elliptic curves obtained by the action of an ideal class, recover the ideal class inducing this action.

The versatility of CSI-FiSh lends itself well to further applications – including distributed signatures [ABCP23, CS20], blind signatures [KLLQ24, HLM+25], group/ring signatures [BKP20, BDK+22], threshold signatures [DM20], adaptor signatures [CDG+26], and many others. Therefore it has become quite crucial to understand the concrete security of CSI-FiSh. However it is well documented that CSI-FiSh has a substantial tightness loss in the security analysis coming from the Fiat Shamir's use of the forking lemma. As noted in [EKP20], with $q = 2^{40}$ random oracle queries, plain CSI-FiSh provides only about 44 bits of exact classical security[1], against modest adversaries under the known reductions. The situation is more delicate for a quantum security analysis. Rewinding-based arguments do not directly carry over to the quantum random-oracle model (QROM), and therefore the standard forking-lemma proof strategy gives no meaningful quantum security guarantee for CSI-FiSh. This motivated Lossy CSI-FiSh [EKP20], a variant designed to provide the necessary exact security at the cost of slower performance. Consequently, tight security

[1] A cryptographic scheme has λ *bits of provable security* if for any running time parameter $t \leq 2^{\lambda}$, any attacker running in time t succeeds with probability $O(t/2^{\lambda})$.

bounds for plain CSI-FiSh remain out of reach, and achieving NIST Level I security against classical adversaries appeared to require substantially larger parameters.

Moreover the security of isogeny-based signatures has largely been analyzed only in the single-user setting. This contrasts with real-world deployments, where many public keys are available simultaneously, and an adversary succeeds by forging under *any chosen one* of the T users' keys [Res18]. A naïve multi-user analysis via a union bound gives a valid guarantee, but incurs a multiplicative loss in T, making the resulting bounds too loose for concrete security.

Even less is known about multi-user *preprocessing* security, where high-resource adversaries may perform massive offline computation and can compress the result into a hint before any public keys are known. Then the online adversary uses this hint to forge under *any one* of the generated public keys. Such attacks are naturally captured by auxiliary-input (AI) idealized models, originating with Unruh's auxiliary-input random oracle model (AI-ROM) [Unr07] and later developed for other idealized models [CDG18]. However, proving security directly in the AI model is difficult, so the analysis is often carried out in the more tractable, but less realistic, bit-fixing (BF) model, where the preprocessing adversary is restricted to fixing a bounded number of input-output pairs of the ideal objects.

The breakthrough results of Coretti, Dodis, Guo, and Steinberger [CDGS18, CDG18] show how to transfer security bounds from the BF model to the AI model for single idealized models, such as the random oracle model (ROM) and generic group model (GGM) as well as other models. More recent work [BL25] extends this paradigm to combinations of idealized models, such as ROM+GGM. For CSI-FiSh, however, the relevant preprocessing setting is different. The Fiat-Shamir transform requires a random oracle, while the isogeny-based group action component is naturally analyzed in the generic group action model (GGAM), where group-action set elements are hidden behind random labels and the adversary only has access to group action oracles. Thus, the idealized setting for the preprocessing security of CSI-FiSh is ROM+GGAM. The existing BF-to-AI translations mentioned above do not cover this setting.

Our Contributions. In this paper, we prove the multi-user security of CSI-FiSh in many of the settings presented previously. First, in Section 3 and without preprocessing, we provide tight multi-user security bounds against *classical* adversaries breaking CSI-FiSh. As a result, we show that CSI-FiSh achieves NIST Level I classical provable security in the ROM+GGAM, even in the multi-user setting. Crucially, our bound incurs only an additive loss in the number of users, rather than a black-box multiplicative one. It also holds under adaptive corruptions, meaning that the adversary may adaptively learn some users' secret keys but must forge under an uncorrupted user. Second, in Section 4, we develop the preprocessing multi-user security for CSI-FiSh. To do so, we extend the BF-to-AI translations of Coretti et al. [CDGS18, CDG18] to the ROM+GGAM setting. This allows us to obtain tight multi-user security bounds for a slightly modified version of CSI-FiSh against preprocessing classical adversaries. The modification comes from a simple preprocessing attack on plain CSI-FiSh which we also detail. The main new tool in these results is the *Group Action Bridge-Finding* game – an algebraic intermediate game between multi-target GAIP (see Problem 2) and CSI-FiSh. We show that the security bounds for this game match those of multi-target GAIP asymptotically, which allows us to derive tight multi-user security bounds for CSI-FiSh.

We emphasize the point that these bounds for CSI-FiSh only apply to classical adversaries. Translating this to the quantum realm and get tight quantum bounds is rather nontrivial due to Kuperberg's quantum algorithm [Kup05].Therefore to deal with its post-quantum security, we

finally prove, in Section 5, the *quantum* multi-user security in the QROM and the algebraic group-action model (AGAM). We give a tight reduction from breaking CSI-FiSh in the multi-user setting to solving any-pair multi-target GAIP. This is achieved by combining tight adaptive reprogramming with a multi-user reduction tailored to CSI-FiSh. Once again, this only incurs an additive loss in the number of users.

Acknowledgment. The authors would like to thank David Jao, Youcef Mokrani, Nic Swanson and Mojtaba Fadavi for valuable discussions that helped shape this work. The authors also acknowledge support of the NSERC Alliance Consortia Quantum Grant "Accelerating the Transition to Quantum-Resistant Cryptography" (ALLRP 578463–2022) and the NSERC Discovery Grants program.

2 Preliminaries

2.1 Orientations and the Generic Group Action Model

We begin by recalling the definition of a group action from [DHK+23], and then introduce cyclic effective group actions with twists.

Definition 1 (Effective group actions with Twists). *Let $(\mathsf{G}, \circ)$ be a finite cyclic group with identity element $e \in \mathsf{G}$, and let $\mathcal{X}$ be a finite set. An* effective group action with twists *is a map $\star : \mathsf{G} \times \mathcal{X} \to \mathcal{X}$ such that we have the following:*

1. *(Group action): $e \star x = x$ and $(g \circ h) \star x = g \star (h \star x)$, for all $g, h \in \mathsf{G}$ and $x \in \mathcal{X}$.*
2. *(Testing and operations in G): There are efficient algorithms for membership testing, equality testing, (uniform) sampling and group operations in G.*
3. *(Testing in $\mathcal{X}$): There is an efficient algorithm for membership testing in $\mathcal{X}$.*
4. *(Group action operation): There is an efficient algorithm to compute $g \star x$ for any $g \in G$ and $x \in \mathcal{X}$.*
5. *(Distinguished element): There is a distinguished $\tilde{x}_0 \in \mathcal{X}$.*
6. *(Twist): There is an efficient algorithm to compute $g^{-1} \star \tilde{x}_0$ given $g \star \tilde{x}_0$.*

We write $(\mathsf{G}, \mathcal{X}, \star, \tilde{x}_0)$ for such an action. The action is said to be regular *if, for every $x, y \in \mathcal{X}$, there exists a unique element $g \in \mathsf{G}$ such that $y = g \star x$.*

2.2 Group Action via Isogenies

We recall the instantiation of effective group actions from oriented supersingular elliptic curves. Let F be a quadratic number field and $\mathfrak{O}$ an order in F. An F-orientation on a supersingular elliptic curve E is an embedding $\iota : F \hookrightarrow \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$. An $\mathfrak{O}$-orientation is an F-orientation such that $\iota(\mathfrak{O}) \subseteq \mathrm{End}(E)$. In this context, we call a pair (E, ι) an $\mathfrak{O}$-oriented curve. We assume from now on that all orientations are *primitive*, i.e., $\iota(\mathfrak{O}) = \iota(F) \cap \mathrm{End}(E)$.

Let $\mathrm{SS}_{\mathfrak{O}}(p)$ denote the set of isomorphism classes of $\mathfrak{O}$-oriented supersingular elliptic curves over $\overline{\mathbb{F}}_p$. Under the standard assumption that p does not divide the conductor of $\mathfrak{O}$ and does not split in F, $\mathrm{SS}_{\mathfrak{O}}(p)$ is nonempty [Onu21, Prop. 3.2]. We define ideal-induced isogenies as follows: for $(E, \iota) \in \mathrm{SS}_{\mathfrak{O}}(p)$ and an ideal $\mathfrak{a} \subseteq \mathfrak{O}$, set $E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}} \ker(\iota(\alpha))$. This subgroup determines an

isogeny $\phi_{\mathfrak{a}} : E \to E_{\mathfrak{a}}$ and induces an orientation on $E_{\mathfrak{a}}$ given by $(\phi_{\mathfrak{a}})_*(\iota)(\alpha) = \big(\phi_{\mathfrak{a}} \circ \iota(\alpha) \circ \hat{\phi}_{\mathfrak{a}}\big) \otimes \frac{1}{\deg \phi}$. In this way, one obtains an action

$$\mathrm{Cl}(\mathfrak{O}) \times \mathrm{SS}_{\mathfrak{O}}(p) \longrightarrow \mathrm{SS}_{\mathfrak{O}}(p), \qquad (\mathfrak{a}, (E, \iota)) \longmapsto \mathfrak{a} * (E, \iota) \coloneqq \big(E_{\mathfrak{a}}, (\phi_{\mathfrak{a}})_*(\iota)\big).$$

When p does not split in F and does not divide the conductor of $\mathfrak{O}$, the class group action of $\mathrm{Cl}(\mathfrak{O})$ on $\mathrm{SS}_{\mathfrak{O}}(p)$ is regular [Onu21, Theorem 3.4]. The notion of twist is used in the CSIDH setting where $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$), the distinguished (or auxiliary) curve is $E_0 : y^2 = x^3 + x$, and $E^t = \mathfrak{a}^{-1} \star E_0$ is efficiently computable from $E = [\mathfrak{a}] \star E_0$ through the quadratic twist of E. Moreover, recent advances show that this action on $\mathfrak{O}$-oriented curves can be evaluated efficiently, notably via the CLAPOTIS algorithm [PR23] and PEGASIS [DEF+25].

We are only concerned with the class group action where the group is cyclic (e.g. when $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$). So one may abstract away the underlying objects and view only the action itself without the heavy number-theoretic machinery.

We now recall the hard problems associated with these group actions:

Problem 1. The *Group Action Inverse Problem* (GAIP) states: given two elements $E, E' \in \mathcal{X}$, output an ideal $\mathfrak{a} \in \mathsf{G}$ such that $E' = \mathfrak{a} \star E$.

Problem 2. The *Multi-Target Group Action Inverse Problem* (MT-GAIP) states: given n elements $E_0, \ldots, E_{n-1} \in \mathcal{X}$, output two integers $i, j < n$, with $E_i \neq E_j$, and an ideal $\mathfrak{a} \in \mathsf{G}$ such that $E_i = \mathfrak{a} \star E_j$.

For an adversary $\mathcal{A}$ that solves MT-GAIP, we define the adversary's advantage by

$$\mathrm{Adv}_n^{\mathsf{MT\text{-}GAIP}}(\mathcal{A}) \coloneqq \Pr[\mathcal{A} \text{ solves } \mathsf{MT\text{-}GAIP}].$$

The Generic Group Action Model. The *Generic Group Action Model* (GGAM) [DHK+23] is the analogue of Generic Group Model (GGM) [Nec94, Sho97] for effective group actions. It is suited to settings where security relies on inverting the action of a group G on a set $\mathcal{X}$, rather than on a hard problem internal to G.

Following [DHK+23], only the set $\mathcal{X}$ is generically encoded, while elements of G remain explicit. This differs from Montgomery and Zhandry's model [MZ22], which generically encodes both G and $\mathcal{X}$. The former is more natural for the class-group actions underlying CSI-FiSh, where the intended hardness lies in inverting the action on $\mathcal{X}$, while the structure of the acting group need not be hidden.

Concretely, an injective random labeling map $\tau : \mathcal{X} \to \{0,1\}^\rho$ assigns a unique label $\mathcal{E} = \tau(E)$ to each $E \in \mathcal{X}$. The adversary manipulates these labels through the action oracle $\mathcal{O}^{\exp}(\mathfrak{g}, \mathcal{E}) = \tau(\mathfrak{g} \star E)$ and the twisting oracle $\mathcal{O}^{\mathsf{tw}}(\tau(\mathfrak{g} \star E)) = \tau(\mathfrak{g}^{-1} \star E)$. Thus, the adversary can evaluate the group action without access to the concrete representations of elements of $\mathcal{X}$.

We remark the following lemma that bounds the advantage of a GGAM adversary to solve the group action inverse problem GAIP.

Lemma 1 ([DHK+23, Corollary 1]). *Every adversary making at most q generic group action oracle queries solves* GAIP *with probability at most q^2/N.*

2.3 Sigma Protocols and Fiat-Shamir Signatures

We recall the formal definitions of sigma protocols and signature schemes. We only present signature schemes in the context relevant to this work – namely the Fiat-Shamir paradigm.

Definition 2 (Sigma Protocol). *A* Σ-protocol *for a relation* R *is a* 3*-message interactive protocol between a prover and a verifier, given by a tuple of PPT algorithms* $(\mathcal{P}_1, \mathcal{P}_2, \mathsf{Ver}_\Sigma)$ *as follows:*

1. *On input* $(x, w) \in R$*, the prover runs* $(\mathsf{com}, \mathsf{state}) \leftarrow \mathcal{P}_1(x, w)$*, obtaining a commitment* com *and the state* state *needed for the response computation. It sends* com *to the verifier.*
2. *The verifier samples a challenge* chal *uniformly from the challenge space* $\mathcal{C}$ *and sends it to the prover.*
3. *The prover replies to the verifier with the response* $\mathsf{resp} \leftarrow \mathcal{P}_2(\mathsf{state}, \mathsf{chal})$.
4. *Finally, the verifier runs the verification algorithm to obtain a bit* $b \leftarrow \mathsf{Ver}_\Sigma(x, \mathsf{com}, \mathsf{chal}, \mathsf{resp})$. *It accepts if and only if* $b = 1$.

Any Σ-protocol should satisfy the following security properties:

- *Correctness*: For every honest execution of the protocol, the verifier accepts the transcript with negligible probability;
- *Soundness*: Any dishonest prover can only produce an accepting transcript with negligible probability; and
- *Honest-Verifier Zero-Knowledge*: There exists a simulator that can generate transcripts of the protocol that are indistinguishable from honest transcripts.

Definition 3 (Fiat-Shamir Signature with Restorable Commitments). *Let* $\Sigma = (\mathcal{P}_1, \mathcal{P}_2, \mathsf{Ver}_\Sigma)$ *be a* Σ*-protocol with challenge space* $\mathcal{C}_\Sigma$*, and let* IG *be an instance generator that, on input* 1^λ*, outputs a pair* $(\mathsf{pk}, \mathsf{sk})$*, where* pk *is a valid statement for* Σ *and* sk *contains a witness for* pk*. Let* H *be a random oracle with range* $\mathcal{C}_\Sigma$*. Assume further that there exists an efficient deterministic algorithm* RecCom *such that, for every accepting transcript* $(\mathsf{pk}, \mathsf{com}, \mathsf{chal}, \mathsf{resp})$ *of* Σ*,* $\mathsf{RecCom}(\mathsf{pk}, \mathsf{chal}, \mathsf{resp}) = \mathsf{com}$*. We define the* Fiat-Shamir signature scheme with restorable commitments $\mathsf{SIG}[\Sigma, \mathsf{H}] = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ *as follows:*

1. $\mathsf{KeyGen}(1^\lambda)$*: outputs* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{IG}(1^\lambda)$.
2. $\mathsf{Sign}(\mathsf{sk}, m)$*: parse* sk *to recover the public statement* pk *and the corresponding witness. Run* $(\mathsf{com}, \mathsf{state}) \leftarrow \mathcal{P}_1(\mathsf{pk}, \mathsf{sk})$*,* $\mathsf{chal} \leftarrow \mathsf{H}(\mathsf{com}, m)$*,* $\mathsf{resp} \leftarrow \mathcal{P}_2(\mathsf{state}, \mathsf{chal})$*, and output a signature* $\sigma = (\mathsf{chal}, \mathsf{resp})$.
3. $\mathsf{Vfy}(\mathsf{pk}, m, \sigma = (\mathsf{chal}, \mathsf{resp}))$*: reconstruct* $\mathsf{com} \leftarrow \mathsf{RecCom}(\mathsf{pk}, \mathsf{chal}, \mathsf{resp})$*, recompute* $\mathsf{chal}' \leftarrow \mathsf{H}(\mathsf{com}, m)$*, and accepts if and only if* $\mathsf{chal}' = \mathsf{chal}$ *and* $\mathsf{Ver}_\Sigma(\mathsf{pk}, \mathsf{com}, \mathsf{chal}, \mathsf{resp}) = 1$.

2.4 CSI-FiSh

In this section, we recall CSI-FiSh [BKV19] – a Σ-protocol based on the isogeny-based group action mentioned in Section 2.1. From now on, we adopt CSI-Fish to a more scalable version we call scalable CSI-FiSh, where the protocol is similar to the regular CSI-FiSh but with larger class group orders. This is solely to allow the scheme scale to higher parameters provided that an explicit description of the cyclic class group is given, such as SCALLOP [DFK+23].

Afterwards, we describe the resulting signature scheme derived from applying the CSI-FiSh Σ-protocol to the Fiat-Shamir paradigm. We present this signature in the generic group action model (GGAM).

CSI-FiSh Σ-Protocol. For the distinguished $\mathfrak{O}$-oriented curve E_0, let $\mathfrak{a} \in \mathrm{Cl}(\mathfrak{O})$ and $E_1 = \mathfrak{a} \star E_0$. We want to prove knowledge of a witness $w = \mathfrak{a}$ to a statement $x = E_1$. To do the prover commits to a curve $\mathsf{com} = E' = \mathfrak{b} \star E_0$ for a uniformly sampled $\mathfrak{b} \in \mathrm{Cl}(\mathfrak{O})$. After receiving a challenge bit $\mathsf{chal} \in \{0, 1\}$ from the verifier, the prover responds with the ideal $\mathsf{resp} = \mathfrak{r} = \mathfrak{b} \cdot \mathfrak{a}^{-\mathsf{chal}}$ and the verifier checks that $\mathfrak{r} \star E_{\mathsf{chal}} = E'$. This gives a Σ-protocol that is 1/2-sound and repeating the protocol λ times amplifies the soundness.

One can also incorporate the twist to get a better soundness per iteration of the protocol. The commitment and response procedures are unchanged but the challenge now is $\mathsf{chal} \in \{-1, 0, 1\}$. In the verification procedure, the curve E_{-1} is $E_{-1} := E_1^{\mathsf{tw}}$. Furthermore one can attain more soundness by increasing the number of curves, $(E_1, \ldots, E_{S-1})$, in the public key and ask the prover to provide an ideal $\mathfrak{r}$ such that $\mathfrak{r} \star E_{\mathsf{chal}} = E'$ for a challenge $\mathsf{chal} \in \{-S+1, \ldots, S-1\}$. To sufficient soundness, one repeats this protocol t times where $(2S-1)^t \approx 2^\lambda$. A formal description of this Σ-protocol is given in Figure 1 and is denoted by $\Sigma_{S,t}^{\mathsf{CSI}}$. We write this description in the context where the challenge space is $\{-S+1, \ldots, S-1\}^t$ rather than $\{-S+1, \ldots, S-1\}$ so that all commitments and responses are collated together.

Prover $\mathcal{P}_1(x, w)$:

Parse $x = (E_1, \ldots, E_{S-1})$, $w = (\mathfrak{a}_1, \ldots, \mathfrak{a}_{S-1})$
for $j = 1, \ldots, t$ **do**
  $\mathfrak{b}_j \leftarrow\!\$\ G$
  $C_j \leftarrow \mathfrak{b}_j \star E_0$
$\mathsf{com} \leftarrow (C_1, \ldots, C_t)$
$\mathsf{state} \leftarrow \big((\mathfrak{b}_1, \ldots, \mathfrak{b}_t), (\mathfrak{a}_1, \ldots, \mathfrak{a}_{S-1})\big)$
return $(\mathsf{com}, \mathsf{state})$

Verifier $\mathsf{Ver}_\Sigma(x, \mathsf{com}, \mathsf{chal}, \mathsf{resp})$:

Parse $x = (E_1, \ldots, E_{S-1}), \mathsf{com} = (C_1, \ldots, C_t)$
Parse $\mathsf{chal} = (c_1, \ldots, c_t), \mathsf{resp} = (\mathfrak{r}_1, \ldots, \mathfrak{r}_t)$
for $k = 1, \ldots, S-1$ **do**
  $E_{-k} \leftarrow E_k^{\mathsf{tw}}$
for $j = 1, \ldots, t$ **do**
  if $C_j \neq \mathfrak{r}_j \star E_{c_j}$ **then**
    return 0
return 1

Prover $\mathcal{P}_2(\mathsf{state}, \mathsf{chal})$:

Parse $\mathsf{state} = \big((\mathfrak{b}_1, \ldots, \mathfrak{b}_t), (\mathfrak{a}_1, \ldots, \mathfrak{a}_{S-1})\big)$
Parse $\mathsf{chal} = (c_1, \ldots, c_t)$
for $j = 1, \ldots, t$ **do**
  if $c_j = 0$ **then**
    $\mathfrak{r}_j \leftarrow \mathfrak{b}_j$
  else
    $\mathfrak{r}_j \leftarrow \mathfrak{b}_j \cdot \mathfrak{a}_{|c_j|}^{-\mathrm{sign}(c_j)}$
$\mathsf{resp} \leftarrow (\mathfrak{r}_1, \ldots, \mathfrak{r}_t)$
return resp

Fig. 1: The t-parallel scalable CSI-FiSh Σ-protocol $\Sigma_{S,t}^{\mathsf{CSI}}$.

CSI-FiSh Signature in the GGAM. One gets the CSI-FiSh signature scheme by applying the Fiat-Shamir transform to the Σ-protocol $\Sigma_{S,t}^{\mathsf{CSI}}$. We remark that the commitment reconstruction procedure in CSI-FiSh is defined directly from the verification equation. In such cases, the condition $\mathsf{Ver}_\Sigma(\mathsf{pk}, \mathsf{com}, \mathsf{chal}, \mathsf{resp}) = 1$ becomes redundant once $\mathsf{com} = \mathsf{RecCom}(\mathsf{pk}, \mathsf{chal}, \mathsf{resp})$ has been computed, so verification may equivalently be written as only checking $H(\mathsf{com}, m) = \mathsf{chal}$).

We reformulate the description of the resulting signature scheme, formally described in Figure 2, in the GGAM. Here the curves E are replaced by their label $\mathcal{E} = \tau(E)$ and group action operations

$\mathsf{KeyGen}(\mathcal{E}_0 \coloneqq \tau(E_0), G \cong \mathrm{Cl}(\mathfrak{O}))$:

1 : **for** $i = 1, \ldots, S-1$ **do**
2 : $\quad \mathfrak{a}_i \leftarrow\!\!\$\ G$
3 : $\quad \mathcal{E}_i \coloneqq \mathcal{O}^{\mathrm{exp}}(\mathfrak{a}_i, \mathcal{E}_0) = \tau(\mathfrak{a}_i \star E_0)$
4 : $\mathsf{sk} \leftarrow (\mathfrak{a}_1, \ldots, \mathfrak{a}_{S-1})$
5 : $\mathsf{pk} \leftarrow (\mathcal{E}_1, \ldots, \mathcal{E}_{S-1})$
6 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Vfy}(\mathsf{pk}, m, \sigma)$:

1 : Parse $\sigma = (\mathbf{r}, \mathbf{c})$, where
2 : $\quad \mathbf{r} \coloneqq (\mathfrak{r}_1, \ldots, \mathfrak{r}_t)$
3 : $\quad \mathbf{c} \coloneqq (c_1, \ldots, c_t)$
4 : **for** $i = 1, \ldots, S-1$ **do**
5 : $\quad \mathcal{E}_{-i} \coloneqq \mathcal{O}^{\mathrm{tw}}(\mathcal{E}_i)$
6 : **for** $i = 1, \ldots, t$ **do**
7 : $\quad \mathcal{E}^{(i)} \coloneqq \mathcal{O}^{\mathrm{exp}}(\mathfrak{r}_i, \mathcal{E}_{c_i})$
8 : $\mathbf{c}' \leftarrow H(\mathcal{E}^{(1)} \| \cdots \| \mathcal{E}^{(t)} \| m)$
9 : **if** $\mathbf{c} = \mathbf{c}'$ **then return** 1
10 : **return** 0

$\mathsf{Sign}(\mathsf{sk}, m)$:

1 : **for** $i = 1, \ldots, t$ **do**
2 : $\quad \mathfrak{b}_i \leftarrow\!\!\$\ G$
3 : $\quad \mathcal{E}^{(i)} \coloneqq \mathcal{O}^{\mathrm{exp}}(\mathfrak{b}_i, \mathcal{E}_0) = \tau(\mathfrak{b}_i \star E_0)$
4 : $\mathbf{c} \coloneqq (c_1, \ldots, c_t) \coloneqq \mathsf{H}(\mathcal{E}^{(1)} \| \cdots \| \mathcal{E}^{(t)} \| m) \in \mathcal{C}^t$
5 : **for** $i = 1, \ldots, t$ **do**
6 : $\quad$ **if** $c_i = 0$ **then**
7 : $\quad\quad \mathfrak{r}_i \leftarrow \mathfrak{b}_i$
8 : $\quad$ **else** $\mathfrak{r}_i \coloneqq \mathfrak{b}_i \cdot \mathfrak{a}_{|c_i|}^{-\mathrm{sign}(c_i)}$
9 : $\mathbf{r} \coloneqq (\mathfrak{r}_1, \ldots, \mathfrak{r}_t)$
10 : **return** $\sigma \coloneqq (\mathbf{r}, \mathbf{c})$

Fig. 2: Scalable CSI-FiSh $\Pi_{S,t}^{\text{CSI-FiSh}}$ in the Generic Group Action Model.

are replaced by the oracles $\mathcal{O}^{\mathrm{exp}}$ and $\mathcal{O}^{\mathrm{tw}}$. We denote this CSI-FiSh signature by $\Pi_{S,t}^{\text{CSI-FiSh}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ to emphasize the dependence on S and the security level desired. We keep this notation consistent throughout the work.

Compressed CSI-FiSh. As noted in [BKV19], a binding commitment scheme `Commit` can be used to reduce the public-key size. Recall that the public key contains all labels $(\mathcal{E}_1, \ldots, \mathcal{E}_{S-1})$, but only the labels indexed by the challenge values are needed during verification. Thus one can replace this list by a commitment `Comm` of these labels, $\texttt{Comm} \coloneqq \texttt{Commit}(\mathcal{E}_1, \ldots, \mathcal{E}_{S-1})$, implemented using a Merkle-tree. In particular assume for simplicity that $S - 1 = 2^d$ and let $h_{d,i} = H(\mathcal{E}_i \,\|\, 2^d + i \,\|\, \mathsf{MerkleKey})$, where $\mathsf{MerkleKey} \in \{0,1\}^\lambda$ is chosen uniformly at random during key generation and is used in both signing and verification. The internal nodes are then defined recursively by $h_{k,i} = H(h_{k+1,2i-1} \,\|\, h_{k+1,2i} \,\|\, 2^k + i \,\|\, \mathsf{MerkleKey})$, where the root of this tree is the commitment `Comm`. The signature then includes, for each challenge index $|c_i|$, an opening π_i proving that $\mathcal{E}_{|c_i|}$ is the value committed at that position. By the binding property of the commitment, the signer cannot open the same public key to inconsistent labels.

2.5 Multi-User Security Notions

We now define multi-user security notions for signature schemes. Throughout this section and the rest of the paper, $T \geq 1$ denotes the number of users. These notions reduce to their standard single-user counterparts when $T = 1$.

Definition 4 (Multi-User Signature Forgery Game). *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ be a signature scheme, λ be a security parameter, $T \geq 1$, and let $\mathcal{A}$ be an adversary. The* multi-user

signature-forgery experiment, *denoted by* $\mathsf{SigForge}^{T}_{\mathcal{A},\Pi}(\lambda)$ *is given in Figure 3 by omitting every gray-shaded component.*

The challenger generates an independent key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$ *for each user* $i \in [T]$ *and gives all public keys to* $\mathcal{A}$*. The adversary has access to the signing oracle* `Signing`*, and the set* $\mathcal{Q}_\mathsf{S}$ *records all pairs* (i, m) *submitted to this oracle. The adversary wins if it outputs* (i^*, m^*, σ^*) *such that* $i^* \in [T]$*,* $(i^*, m^*) \notin \mathcal{Q}_\mathsf{S}$*, and* $\mathsf{Vfy}(\mathsf{pk}_{i^*}, m^*, \sigma^*) = 1$*.*

$\mathsf{SigForge}\,\mathsf{Cor}^{T}_{\mathcal{A},\Pi}(\lambda)$:

$\mathcal{Q}_\mathsf{S} \leftarrow \varnothing$; $\mathsf{C} \leftarrow \varnothing$
for $i \in [T]$ **do**
  $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda)$
$(i^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathtt{Signing},\mathtt{Corrupt}}(\mathsf{pk}_1, \ldots, \mathsf{pk}_T)$
if $i^* \notin [T] \vee i^* \in \mathsf{C}$ **then**
  return 0
if $(i^*, m^*) \in \mathcal{Q}_\mathsf{S}$ **then**
  return 0
return $\mathsf{Vfy}(\mathsf{pk}_{i^*}, m^*, \sigma^*)$

`Signing`(i, m):

if $i \notin [T]$ **then**
  return $\perp$
$\mathcal{Q}_\mathsf{S} \leftarrow \mathcal{Q}_\mathsf{S} \cup \{(i, m)\}$
return $\mathsf{Sign}(\mathsf{sk}_i, m)$

`Corrupt`(i):

if $i \notin [T]$ **then**
  return $\perp$
$\mathsf{C} \leftarrow \mathsf{C} \cup \{i\}$
return sk_i

Fig. 3: The ordinary and adaptive-corruption multi-user signature-forgery games. Omitting all gray-shaded components gives $\mathsf{SigForge}^{T}_{\mathcal{A},\Pi}(\lambda)$; including all of them, including the entire `Corrupt` procedure, gives $\mathsf{SigForgeCor}^{T}_{\mathcal{A},\Pi}(\lambda)$.

We next extend the multi-user forgery game to *adaptive corruptions*. In this setting, the adversary may adaptively corrupt users and obtain their secret keys, but a successful forgery must target a user that remains *uncorrupted*.

Definition 5 (Multi-User Signature Forgery Game under Adaptive Corruption). *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ *be a signature scheme. The adaptive-corruption experiment* $\mathsf{SigForgeCor}^{T}_{\mathcal{A},\Pi}(\lambda)$ *is obtained from Figure 3 by including all gray-shaded components. In particular, the challenger additionally initializes the corruption set* $\mathsf{C} \leftarrow \varnothing$*, and* $\mathcal{A}$ *is additionally given access to the corruption oracle* `Corrupt`*. The adversary may issue corruption queries adaptively and interleave them with its signing queries. Upon a query* `Corrupt`(i) *for* $i \in [T]$*, the challenger updates* $\mathsf{C} \leftarrow \mathsf{C} \cup \{i\}$ *and returns* sk_i*. The adversary wins if it outputs* (i^*, m^*, σ^*) *such that* $i^* \in [T]$*,* $i^* \notin \mathsf{C}$*,* $(i^*, m^*) \notin \mathcal{Q}_\mathsf{S}$*, and* $\mathsf{Vfy}(\mathsf{pk}_{i^*}, m^*, \sigma^*) = 1$*.*

Definition 6 (Multi-User EUF-CMA Security in the ROM+GGAM). *Consider the GGAM with a labeling map* $\tau : \mathcal{X} \to \{0,1\}^\rho$*. In the ROM+GGAM instantiation of the experiment from Definition 4 and Definition 5, the adversary is additionally given access to the random oracle* H *and the generic group-action oracles* $\mathcal{O}$*. We denote the resulting experiment by* $\mathsf{SigForge}^{T,\mathsf{H},\mathcal{O}}_{\mathcal{A},\Pi}(\lambda)$ *and* $\mathsf{SigForgeCor}^{T,\mathsf{H},\mathcal{O}}_{\mathcal{A},\Pi}(\lambda)$*, respectively.*

Let $\mathcal{A}$ *make at most* q_H *queries to the random oracle* H*,* q_G *queries to the GGAM oracle* $\mathcal{O}$*, and* q_S *signing queries in total across all* T *users. A signature scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ *is said to be* $(T, q_\mathsf{H}, q_\mathsf{G}, q_\mathsf{S}, \varepsilon)$-MU-EUF-CMA secure *(multi-user unforgeable under chosen-message attack) in the ROM+GGAM if every such adversary* $\mathcal{A}$ *satisfies*

$$\Pr\left[\mathsf{SigForge}^{T,\mathsf{H},\mathcal{O}}_{\mathcal{A},\Pi}(\lambda) = 1\right] \le \varepsilon.$$

It is said to be $(T, q_{\mathsf{H}}, q_{\mathsf{G}}, q_{\mathsf{S}}, \varepsilon)$-MU-EUF-CMA secure under adaptive corruption *in the ROM+GGAM if every such adversary* $\mathcal{A}$*, additionally given adaptive oracle access to* `Corrupt`*, satisfies*

$$\Pr\left[\mathsf{SigForgeCor}^{T,\mathsf{H},\mathcal{O}}_{\mathcal{A},\Pi}(\lambda) = 1\right] \leq \varepsilon.$$

In both cases, the probability is taken over the choice of the labeling map τ*, the random oracle* H*, and the random coins of* KeyGen*,* Sign*, and* $\mathcal{A}$*.*

3 Multi-User Security of CSI-FiSh in the ROM+GGAM

Breaking single-user CSI-FiSh is known to reduce to MT-GAIP [BKV19], which tightly implies GAIP. However, the first reduction relies on the forking lemma: a forger with advantage ε yields a MT-GAIP solver with advantage only $\varepsilon^2/q_{\mathsf{H}}$, where q_{H} is the number of RO queries. This loss requires larger parameters at higher security levels, while increasing the class-group size becomes quickly impractical.

In this section, we establish *tight* multi-user security of CSI-FiSh with adaptive corruptions. We adapt the bridge-finding framework of [BL22] to group actions and reduce CSI-FiSh security directly to bridge finding. This avoids the forking lemma and yields multi-user bounds matching the magnitude of Lemma 1, recovering tight single-user security as a special case.

3.1 Group Action Bridge-Finding Game

We define the *group action bridge-finding game*, where the adversary aims to produce two algebraically distinct representations of the same group-action label. The challenger maintains a list $\mathcal{L}$ of tuples $(\mathcal{E}, \mathbf{a}, b)$ satisfying $\mathcal{E} = [\mathfrak{g}]^{\mathbf{a}\cdot\mathbf{x}+b} \star E_0$. To preserve this invariant, a restricted challenger-only oracle `GAI` is used: whenever a fresh valid label enters an action or twist query, `GAI` recovers $b \in \mathbb{Z}_N$ such that $E = [\mathfrak{g}]^b \star E_0$, and the challenger adds $(\mathcal{E}, \mathbf{0}, b)$ to $\mathcal{L}$. Since this only strengthens the adversary, any upper bound for this game also applies to the ordinary GGAM.

The game also allows *adaptive corruptions*. Upon corrupting user i, the challenger reveals $x_{i,1}, \ldots, x_{i,S-1}$ and absorbs these values into the constant terms of all tuples in $\mathcal{L}$, so that bridge finding is thereafter defined only with respect to the remaining hidden variables.

During the execution of $\mathcal{A}$, the challenger monitors the GGA oracle transcript and maintains $\mathcal{L}$ using `Record`. Before processing a group action or twist query whose input label $\mathcal{E}$ is not represented in $\mathcal{L}$, the challenger obtains the unique $b \in \mathbb{Z}_N$ satisfying $E = [\mathfrak{g}]^b \star E_0$ through the internal `GAI` oracle and calls $\texttt{Record}(\mathcal{E}, \mathbf{0}, b)$. If $(\mathcal{E}, \mathbf{a}, b) \in \mathcal{L}$ and $\mathcal{O}^{\exp}(\mathfrak{c}, \mathcal{E}) = \mathcal{E}'$, where $[\mathfrak{c}] = [\mathfrak{g}]^c$, the challenger calls $\texttt{Record}(\mathcal{E}', \mathbf{a}, b + c)$. Similarly, if $\mathcal{O}^{\mathsf{tw}}(\mathcal{E}) = \mathcal{E}'$, the challenger calls $\texttt{Record}(\mathcal{E}', -\mathbf{a}, -b)$. Thus, one online GGA oracle query creates at most *two* calls to `Record`, i.e., one for a fresh input and one for the output. Notice that the event $\mathsf{BRIDGE}^{N,(T,S)}_{\mathcal{A}}(\mathbf{x})$ implies that $[\mathfrak{g}]^{\mathbf{a}\cdot\mathbf{x}+b} \star E_0 = [\mathfrak{g}]^{\mathbf{a}'\cdot\mathbf{x}+b'} \star E_0$, which by the fact that the action is regular we have $\mathbf{a}\cdot\mathbf{x} + b = \mathbf{a}'\cdot\mathbf{x} + b'$, exactly as in the GGM setting. For compactness, we write $\mathcal{I} \coloneqq [T] \times [S-1]$ and $M \coloneqq |\mathcal{I}|$.

Theorem 1. *Let* $N = |\mathrm{Cl}(\mathfrak{O})|$*. If an adversary* $\mathcal{A}$ *makes at most* q_{G} *generic group action queries, then we have*

$$\Pr\left[\mathsf{CorrBRIDGE}^{N,(T,S)}_{\mathcal{A}}(\mathbf{x}) = 1\right] \;\leq\; \frac{M + \binom{M}{2}}{N} + \frac{q_{\mathsf{G}} M + q_{\mathsf{G}}(q_{\mathsf{G}}+1)}{N - (M + 2q_{\mathsf{G}} + 1)^2 - M},$$

where the randomness is taken over $x_1, \ldots, x_T$ *and the random coins of* $\mathcal{A}$*.*

$\mathsf{CorrBRIDGE}_{\mathcal{A}}^{N,(T,S)}(\mathbf{x})$:

Parse $\mathbf{x} = (x_{1,1}, \ldots, x_{T,S-1})$
$\mathcal{L} \leftarrow \varnothing$; $\mathsf{Revealed} \leftarrow \varnothing$; $\mathsf{win} \leftarrow 0$
$\texttt{Record}(\mathcal{E}_0, \mathbf{0}, 0)$
for $i \in [T]$ **do**
  for $j \in [S-1]$ **do**
    $E_j^{(i)} \leftarrow [\mathfrak{g}]^{x_{i,j}} \star E_0$
    $\mathcal{E}_j^{(i)} \leftarrow \tau(E_j^{(i)})$
    $\texttt{Record}(\mathcal{E}_j^{(i)}, \hat{u}_{i,j}, 0)$
$o \leftarrow \mathcal{A}^{\mathcal{O}^{\mathrm{exp}}, \mathcal{O}^{\mathrm{tw}}, \texttt{Corrupt}}\left(\mathcal{E}_0, \{\mathcal{E}_j^{(i)}\}_{i\in[T], j\in[S-1]}\right)$
return win

$\texttt{Record}(\mathcal{E}, \mathbf{a}, b)$:

if $\exists (\mathcal{E}, \mathbf{a}', b') \in \mathcal{L}$ such that
  $(\mathbf{a}', b') \neq (\mathbf{a}, b)$ **then**
  $\mathsf{win} \leftarrow 1$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathcal{E}, \mathbf{a}, b)\}$

$\texttt{Corrupt}(i)$:

if $i \notin [T]$ **then**
  return $\perp$
$\mathsf{Revealed} \leftarrow \mathsf{Revealed} \cup \{i\}$
$\mathcal{L} \leftarrow \big\{(\mathcal{E}, \mathbf{a} - \sum_{j=1}^{S-1} a_{i,j}\hat{u}_{i,j}, b + \sum_{j=1}^{S-1} a_{i,j}x_{i,j}) :$
  $(\mathcal{E}, \mathbf{a}, b) \in \mathcal{L}\big\}$
return $(x_{i,1}, \ldots, x_{i,S-1})$

Fig. 4: The group-action bridge-finding game under adaptive corruptions. The terminal output o of $\mathcal{A}$ is ignored. The flag win is sticky and may already be set during initialization.

Proof. Let InitBr denote the event that the initial list already contains a bridge, before the adversary makes any online oracle query. By injectivity of τ and regularity of the group action, this can occur only if $x_{i,j} = 0$ for some $(i,j) \in \mathcal{I}$, or if $x_{i,j} = x_{i',j'}$ for two distinct indices. Hence,

$$\Pr[\mathsf{InitBr}] \leq \frac{M + \binom{M}{2}}{N}.$$

Now we condition on $\overline{\mathsf{InitBr}}$. A bridge occurs if and only if there exist two distinct tuples $(\tau(E), \mathbf{a}, b), (\tau(E), \mathbf{a}', b') \in \mathcal{L}$ with $(\mathbf{a}, b) \neq (\mathbf{a}', b')$. Let B_i denote the event that the i^{th} oracle output causes a bridge, and let $\overline{\mathsf{BRIDGE}}_{<i}$ denote the event that no bridge occurred before step i, as defined in the above game, with respect to the coordinates belonging to unrevealed users, i.e., $([T] \setminus \mathsf{Revealed}) \times [S-1]$. By union bound,

$$\Pr[\mathsf{BRIDGE}_{\mathcal{A}}^{N,(T,S)}(\mathbf{x}) = 1] \;\leq\; \Pr[\mathsf{InitBr}] + \sum_{i=1}^{q_{\mathsf{G}}} \Pr\left[B_i \mid \overline{\mathsf{BRIDGE}}_{<i} \wedge \overline{\mathsf{InitBr}}\right].$$

Fix i and condition on $\overline{\mathsf{BRIDGE}}_{<i}$. Let $(\tau(E_i), \mathbf{a}_i, b_i)$ be the tuple appended to $\mathcal{L}$ for the i^{th} output. A bridge at step i can only happen if there exists some prior tuple $(\cdot, \mathbf{a}, b) \in \mathcal{L}$ with $(\mathbf{a}, b) \neq (\mathbf{a}_i, b_i)$ such that

$$\langle \mathbf{a} - \mathbf{a}_i, \mathbf{x} \rangle \equiv b_i - b \qquad \bmod \mathbb{Z}_N. \tag{1}$$

Let $\overline{\mathsf{Revealed}}$ be the set of unrevealed variables. Note that for each $r \in \mathsf{Revealed}$ and all $(\mathcal{E}, \mathbf{a}', b') \in \mathcal{L}$, we have $\mathbf{a}[r]{=}0$ (the r^{th} entry being zero). For every $r \in \overline{\mathsf{Revealed}}$, we can almost view $\mathbf{x}[r]$ as uniformly random from $\mathbb{Z}/N\mathbb{Z}$. However, this is not quite true since the $\mathbf{x}[r]$'s are subject to the constraint of $\overline{\mathsf{BRIDGE}}_{<i}$. Fix a prior tuple $(\mathcal{E}, \mathbf{a}, b)$ and set $\mathbf{d}_i = \mathbf{a} - \mathbf{a}_i \neq \mathbf{0}$. Choose an index r with $\mathbf{d}_i[r] \neq 0$, and sample $\mathbf{x}$ coordinate-wise from $\overline{\mathsf{Revealed}}$ with $\mathbf{x}[r]$ last. For fixed $\{\mathbf{x}[\ell] : \ell \neq r\}$, Equation (1) becomes a linear equation in the single variable $\mathbf{x}[r]$ with nonzero coefficient $\mathbf{d}_i[r]$, hence has at most one solution for $\mathbf{x}[r]$.

Under $\overline{\mathsf{InitBr}} \wedge \overline{\mathsf{BRIDGE}}_{<i}$, all existing affine forms $\langle \mathbf{a}, \mathbf{x} \rangle + b$ indexed by distinct symbolic representations in $\mathcal{L}$ are pairwise distinct. After fixing $\{\mathbf{x}[\ell] : \ell \neq r\}$, each resulting nontrivial inequality forbids at most *one* value of $x[r]$, so these constraints forbid at most $|\mathcal{L}|^2$ values of $\mathbf{x}[r]$.

In addition, conditioning on $\overline{\mathsf{InitBr}}$ requires $\mathbf{x}[r] \notin \{0\} \cup \{\mathbf{x}[\ell] : \ell \neq r\}$. These initial non-collision constraints forbid at most M additional values, i.e., the value 0 and the other $M-1$ challenge exponents. Therefore, conditioned on $\overline{\mathsf{InitBr}} \wedge \overline{\mathsf{BRIDGE}}_{<i}$ and on $\{\mathbf{x}[\ell] : \ell \neq r\}$ from $\overline{\mathsf{Revealed}}$, the remaining variable $\mathbf{x}[r]$ is uniform over a set of size at least $N - |\mathcal{L}|^2 - M$, and thus

$$\Pr\big[\langle \mathbf{a} - \mathbf{a}_i, \mathbf{x} \rangle \equiv b_i - b \mod \mathbb{Z}_N \mid \overline{\mathsf{InitBr}} \wedge \overline{\mathsf{BRIDGE}}_{<i}\big] \;\leq\; \frac{1}{N - |\mathcal{L}|^2 - M}.$$

Union bounding over all prior tuples in $\mathcal{L}$ yields

$$\Pr\big[B_i \mid \overline{\mathsf{InitBr}} \wedge \overline{\mathsf{BRIDGE}}_{<i}\big] \;\leq\; \frac{|\mathcal{L}|}{N - |\mathcal{L}|^2 - M}.$$

Finally, since a single oracle query may introduce at most two tuples (register a fresh input via `GAI` and append the output), we have $|\mathcal{L}| \leq M + 1 + 2i \leq M + 1 + 2q_{\mathsf{G}}$ prior to step i. Thus, for every $i \leq q_{\mathsf{G}}$,

$$\Pr\big[B_i \mid \overline{\mathsf{InitBr}} \wedge \overline{\mathsf{BRIDGE}}_{<i}\big] \;\leq\; \frac{M + 2i}{N - (M + 2q_{\mathsf{G}} + 1)^2 - M}.$$

Summing over $i = 1, \ldots, q_{\mathsf{G}}$ gives

$$\begin{aligned}
\Pr[\mathsf{CorrBRIDGE}^{N,(T,S)}_{\mathcal{A}}(\mathbf{x}) = 1] \;&\leq\; \Pr[\mathsf{InitBr}] + \sum_{i=1}^{q_{\mathsf{G}}} \big[B_i \mid \overline{\mathsf{InitBr}} \wedge \overline{\mathsf{BRIDGE}}_{<i}\big] \\
&\leq \frac{M + \binom{M}{2}}{N} + \sum_{i=1}^{q_{\mathsf{G}}} \frac{M + 2i}{N - (M + 2q_{\mathsf{G}} + 1)^2 - M} \\
&= \frac{M + \binom{M}{2}}{N} + \frac{q_{\mathsf{G}} M + q_{\mathsf{G}}(q_{\mathsf{G}} + 1)}{N - (M + 2q_{\mathsf{G}} + 1)^2 - M}.
\end{aligned}$$

□

As a corollary, we give a bound for the MT-GAIP problem in the GGAM.

Corollary 1. *Let $\mathcal{A}$ be an adversary making at most q_{G} queries to the generic group-action oracle and is given $M+1$ targets for* MT-GAIP*. Then,*

$$\mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{M+1}(\mathcal{A}) \leq \frac{M + \binom{M}{2}}{N} + \frac{(q_{\mathsf{G}} + 1)(M + q_{\mathsf{G}} + 2)}{N - \big(M + 2(q_{\mathsf{G}} + 1) + 1\big)^2 - M}.$$

The probability is taken over the choice of τ, the challenges $x_1, \ldots, x_M$, and the internal randomness of $\mathcal{A}$.

Proof. Define $\mathcal{B}$ to run the MT-GAIP adversary $\mathcal{A}$ on $\tau(E_0), \ldots, \tau(E_M)$, obtaining (i, j, c) such that $E_i = [\mathfrak{g}]^c \star E_j$, and then query $\mathcal{O}^{\exp}(c, \tau(E_j))$. Since $(\tau(E_j), \mathbf{u}_j, 0) \in \mathcal{L}$, the challenger calls $\mathtt{Record}(\tau([\mathfrak{g}]^c \star E_j), \mathbf{u}_j, c) = \mathtt{Record}(\tau(E_i), \mathbf{u}_j, c)$. But $\mathcal{L}$ already contains $(\tau(E_i), \mathbf{u}_i, 0)$, and $(\mathbf{u}_j, c) \neq (\mathbf{u}_i, 0)$ because $i \neq j$. Hence, a collision occurs, and $\mathcal{B}$ wins whenever $\mathcal{A}$ wins. □

3.2 Tight Reduction of CSI-FiSh without Preprocessing

We give a *tight* reduction from CSI-FiSh security to group-action bridge finding, avoiding the forking lemma. The reduction preserves success probability and incurs only a t-fold overhead in group-action queries: one action query per repetition for each simulated signing query and for reconstructing the final forgery, with at most one additional twist query for each previously unseen negative challenge curve. The bound is already tight in the single-user setting and extends additively to the multi-user setting, including adaptive corruptions. Here, t denotes the number of parallel repetitions used to reduce the soundness error.

Theorem 2. *$\Pi_{S,\lambda}^{\text{CSI-FiSh}}$ is $(T, q_{\mathsf{H}}, q_{\mathsf{G}}, q_{\mathsf{S}}, \varepsilon)$-MU-EUF-CMA secure under adaptive corruption with*

$$\varepsilon \le \frac{M + \binom{M}{2}}{N} + \frac{\widehat{q}_{\mathsf{G}} M + \widehat{q}_{\mathsf{G}}(\widehat{q}_{\mathsf{G}}+1)}{N - (M + 2\widehat{q}_{\mathsf{G}} + 1)^2 - M} + \frac{q_{\mathsf{S}}(q_{\mathsf{H}} + q_{\mathsf{S}})}{N} + \frac{q_{\mathsf{H}} + q_{\mathsf{S}}}{N - (M + 2\widehat{q}_{\mathsf{G}} + 1)} + \frac{q_{\mathsf{H}}(\kappa_{\mathsf{BQ}} + 1) + q_{\mathsf{S}} + 3}{(2S-1)^t}$$
$$= O\Big(\frac{M^2}{N} + \frac{tq(tq+M)}{N} + \frac{q\kappa_{\mathsf{BQ}}}{2^\lambda}\Big),$$

in the ROM and the GGAM, where $M := T(S-1), \kappa_{\mathsf{BQ}} := \min\{T, t+1\}$, $\widehat{q}_{\mathsf{G}} := q_{\mathsf{G}} + t(q_{\mathsf{S}}+1) + \min\{M, t(q_{\mathsf{S}}+1)\}$, $N = |\mathrm{Cl}(\mathcal{O})|$ and $q := q_{\mathsf{H}} + q_{\mathsf{G}} + q_{\mathsf{S}}$ denotes the online query budget of the signature attacker.

Proof. Let $\mathcal{B}$ be a PPT adversary against scalable CSI-FiSh in the MU-EUF-CMA game, and let $\mathcal{A}_{\mathsf{Bridge}}$ be an adversary in the group action bridge-finding game. We first treat the simplified scheme without commitment compression, but with challenge space $\mathcal{C}$ of size $|\mathcal{C}| = 2S - 1$ and t parallel repetitions (so the FS challenge is in $\mathcal{C}^t$).

Simulation. $\mathcal{A}_{\mathsf{Bridge}}$ runs $\mathcal{B}$ and answers its oracle queries as follows.

- *Random oracle.* $\mathcal{A}_{\mathsf{Bridge}}$ lazy-samples H with a table Hresp. On query w, if $w \in \mathsf{Hresp}$ return $\mathsf{Hresp}[w]$; otherwise sample $\mathbf{c} \leftarrow\!\$\, \mathcal{C}^t$, set $\mathsf{Hresp}[w] \leftarrow \mathbf{c}$, and return $\mathbf{c}$.
- *Corruption oracle.* On a corruption query for user i, $\mathcal{A}_{\mathsf{Bridge}}$ queries $\mathtt{Corrupt}(i)$ to the bridge challenger, obtains $(x_{i,1}, \ldots, x_{i,S-1})$, and forwards the corresponding secret key to $\mathcal{B}$. The challenger updates every tuple in $\mathcal{L}$ by absorbing the revealed variables into the constant term, as specified in the CorrBRIDGE game. We write $\mathsf{Revealed}$ for the set of users corrupted so far.
- *Signing oracle.* On a signing query for user i and message m, if $i \in \mathsf{Revealed}$, then $\mathcal{A}_{\mathsf{Bridge}}$ uses the revealed secret key to answer the query honestly. Otherwise, $\mathcal{A}_{\mathsf{Bridge}}$ samples $\mathbf{c} = (c_1, \ldots, c_t) \leftarrow\!\$\, \mathcal{C}^t$ and $r_1, \ldots, r_t \leftarrow\!\$\, \mathbb{Z}_N$. For each $j \in [t]$, it defines the commitment by one GGAM action query $\tau(C_j) \leftarrow \mathcal{O}^{\exp}\big(r_j, \tau(E_{c_j}^{(i)})\big)$, and sets $w := \tau(C_1)\| \cdots \|\tau(C_t)\|m$. Here, if $c_j < 0$ and $\tau(E_{c_j}^{(i)})$ has not yet been obtained, the simulator first queries $\tau(E_{c_j}^{(i)}) \leftarrow \mathcal{O}^{\mathsf{tw}}(\tau(E_{|c_j|}^{(i)}))$ to obtain the label. If $w \in \mathsf{Hresp}$, it aborts and sets $\mathsf{FailToSign}$; otherwise it programs $\mathsf{Hresp}[w] \leftarrow \mathbf{c} = (c_1, \ldots, c_t)$ and returns $\sigma := \big((c_1, \ldots, c_t), (r_1, \ldots, r_t)\big)$. This produces the correct distribution by the Honest-verifier Zero-Knowledge property.

Each query made by $\mathcal{B}$ to the GGAM is forwarded by $\mathcal{A}_{\mathsf{Bridge}}$. Plus, simulating a signing query requires at most one action query for each of the t repetitions, and reconstructing the final forgery requires at most t further group action queries. Finally, whenever a negative challenge curve is needed for the first time, $\mathcal{A}_{\mathsf{Bridge}}$ obtains it by a twist query. There are at most $M = T(S-1)$ such curves and at most $t(q_{\mathsf{S}}+1)$ occasions when one is needed. Thus, $\mathcal{A}_{\mathsf{Bridge}}$ makes at most $\widehat{q}_{\mathsf{G}} := q_{\mathsf{G}} + t(q_{\mathsf{S}}+1) + \min\{M, t(q_{\mathsf{S}}+1)\}$ group action queries in total.

Forgery implies BRIDGE modulo bad events. Assume $\mathcal{B}$ outputs a valid forgery (m^*, σ^*) under pk_{i^*} with $i^* \notin \mathsf{Revealed}$, where $\sigma^* = \big((c_1^*, \ldots, c_t^*), (r_1^*, \ldots, r_t^*)\big)$, with the challenge $(c_1^*, \ldots, c_t^*) = \mathsf{H}(w^*)$, $w^* = \tau(C_1^*)\| \ldots \|\tau(C_t^*)||m^*$, and C_i^*'s are the commitments reconstructed by the verifier. We distinguish how w^* was *first* entered into Hresp: it was either not present before the forgery, first entered upon an RO query, or first programmed while answering a signing query.

(i) Missing hash entry. If w^* was not previously recorded in Hresp, then $\mathbf{c}^*$ is fresh and uniform in $\mathcal{C}^t$. Thus, the probability that σ^* verifies is at most $|\mathcal{C}|^{-t} = (2S-1)^{-t}$. Hence, we may assume w^* is a query answered by $\mathcal{A}_{\mathsf{Bridge}}$ and pay the probability of this event $\mathsf{MissQuery}$ in the final bound.

(ii) RO hash entry. Suppose that $w = \tau(C_1)\| \ldots \|\tau(C_t)\|m$ is first entered into Hresp upon a fresh RO query. If some $\tau(C_j)$ has not yet been recorded in $\mathcal{L}$, we call this bad event $\mathsf{FailToFind}$, analyzed below. Hence, assume that all commitment labels are already recorded in $\mathcal{L}$.

Condition on no prior BRIDGE. For every $j \in [t]$, let $(\tau(C_j), \mathbf{a}_j, b_j) \in \mathcal{L}$ be its unique symbolic representation when $\mathsf{H}(w)$ is answered. The structure of the GGAM implies that each $\mathbf{a}_j$ is either $\mathbf{0}$ or a signed unit vector $s\hat{u}_{i,h}$ for some currently uncorrupted user $i \in [T] \setminus \mathsf{Revealed}$, $h \in [S-1]$, and $s \in \{-1, +1\}$: GGAM operations preserve this form, while corruption only removes the corresponding user's coefficients. For such a query w, define

$$\mathsf{Users}(w) \coloneqq \{i \in [T] \setminus \mathsf{Revealed} : \mathbf{a}_j = s\hat{u}_{i,h} \text{ for some } j \in [t],\, h \in [S-1],\, s \in \{-1, +1\}\}.$$

Intuitively, $\mathsf{Users}(w)$ denotes the set of uncorrupted users represented in the commitment coordinates of w. Since there are only t commitment coordinates, $|\mathsf{Users}(w)| \le t$. Fix a potential target $i \in \mathsf{Users}(w)$ that remains uncorrupted. A valid forgery whose verification uses the hash input w reconstructs the commitment labels $\tau(C_1), \ldots, \tau(C_t)$ recorded in $\mathcal{L}$. To avoid a bridge, the verification computation must also produce the same symbolic representations, after accounting for any subsequent corruption queries. In particular, the coefficient vectors must agree at every commitment coordinate:

- If $\mathbf{a}_j = s\hat{u}_{i,h}$, this requires challenge sh.
- If $\mathbf{a}_j = \mathbf{0}$, it requires challenge 0.
- If $\mathbf{a}_j = s\hat{u}_{k,h}$ for another user $k \neq i$, then k must subsequently be corrupted so that its coefficient is absorbed into the constant term; the corresponding challenge must then be 0.

Thus, there is at most one candidate challenge vector for target i. Define $\hat{\mathbf{c}}^{(i)}(w) = (c_1^{(i)}, \ldots, c_t^{(i)}) \in \mathcal{C}^t$ by

$$c_j^{(i)}(w) = \begin{cases} sh & \text{if } \mathbf{a}_j = s\hat{u}_{i,h} \text{ for some } h \in [S-1], s \in \{-1, +1\}, \\ 0 & \text{otherwise.} \end{cases}$$

We now collect the candidates over all possible targets. Define $\mathsf{Cand}(w) \coloneqq \{\hat{\mathbf{c}}^{(i)}(w) : i \in \mathsf{Users}(w)\}$, and additionally include $\mathbf{0}^t$ in $\mathsf{Cand}(w)$ if there is an uncorrupted user outside $\mathsf{Users}(w)$. The zero vector accounts for the possibility that all users represented among the commitments are subsequently corrupted and an unrepresented user is chosen as the target. There are at most $|\mathsf{Users}(w)| \le t$ candidates from represented users and at most one additional candidate from unrepresented users. The latter is included only when $|\mathsf{Users}(w)| < T$. Therefore, $|\mathsf{Cand}(w)| \le \min\{T, t+1\} = \kappa_{\mathsf{BQ}}$.

Suppose $\mathbf{c}^* \notin \mathsf{Cand}(w^*)$. Since $i^* \notin \mathsf{Revealed}$, there exists an index $j^* \in [t]$ for which the coefficient vector stored with $\tau(C_{j^*}^*)$, after all subsequent corruption updates, is incompatible with

verification for target user i^* and challenge $c^*_{j^*}$. Let $\mathbf{a}^{\mathrm{fin}}_{j^*}$ denote this coefficient vector at the time of forgery. More precisely, the coefficient vector required by verification is

$$\mathbf{v}^*_{j^*} = \begin{cases} \mathbf{0} & \text{if } c^*_{j^*} = 0, \\ \mathrm{sign}(c^*_{j^*})\hat{u}_{i^*,|c^*_{j^*}|} & \text{if } c^*_{j^*} \neq 0. \end{cases}$$

By the definition of $\mathsf{Cand}(w^*)$, and since subsequent corruptions can only remove coefficients belonging to corrupted users while i^* remains uncorrupted, we have $\mathbf{a}^{\mathrm{fin}}_{j^*} \neq \mathbf{v}^*_{j^*}$. When reconstructing the commitments of the final forgery, $\mathcal{A}_{\mathsf{Bridge}}$ makes the query $\tau(D) \leftarrow \mathcal{O}^{\exp}\big(r^*_{j^*}, \tau(E^{(i^*)}_{c^*_{j^*}})\big)$. By validity of σ^*, we have $D = C^*_{j^*}$, hence $\tau(D) = \tau(C^*_{j^*})$. By the GGAM list-update rule, the challenger appends a tuple $(\tau(C^*_{j^*}), \mathbf{a}', b')$ consistent with the input curve. The newly appended tuple has coefficient vector $\mathbf{v}^*_{j^*}$, whereas the tuple already stored with $\tau(C^*_{j^*})$ has coefficient vector $\mathbf{a}^{\mathrm{fin}}_{j^*}$. Since $\mathbf{a}^{\mathrm{fin}}_{j^*} \neq \mathbf{v}^*_{j^*}$, these are two algebraically distinct representations of the same encoding. Hence, BRIDGE^N occurs.

It remains to consider the case $\mathbf{c}^* \in \mathsf{Cand}(w^*)$. For every RO query w, the set $\mathsf{Cand}(w)$ is determined before the fresh response $\mathsf{H}(w)$ is sampled. Future corruption queries may remove secret variables represented in the symbolic tuples, but cannot introduce a new nonzero user component into any commitment coordinate. We call this event $\mathsf{BadQuery}$. Since $|\mathsf{Cand}(w)| \leq \kappa_{\mathsf{BQ}} = \min\{T, t+1\}$ and $\mathsf{H}(w)$ is uniform in $\mathcal{C}^t$, each RO query is bad with probability at most $\frac{\kappa_{\mathsf{BQ}}}{(2S-1)^t}$. Union bounding over at most q_{H} RO queries gives

$$\Pr[\mathsf{BadQuery}] \leq \frac{q_{\mathsf{H}}\kappa_{\mathsf{BQ}}}{(2S-1)^t}.$$

(iii) Hash entry from signing queries. Suppose that w^* was first programmed while answering a signing query for user $j \neq i^*$ and message m^*.

Suppose that $c^*_\ell \neq 0$ for some $\ell \in [t]$. Since the commitment C^*_ℓ was generated using the challenge curve $E^{(j)}_{c^*_\ell}$, conditioned on no prior bridge, the symbolic representation of $\tau(C^*_\ell)$ has a coefficient vector $\mathrm{sign}(c^*_\ell)\hat{u}_{j,|c^*_\ell|}$. It either remains the same, or becomes $\mathbf{0}$, if j is later corrupted. On the other hand, validity of forgery implies $\tau(C^*_\ell) = \mathcal{O}^{\exp}(r^*_\ell, \tau(E^{(i^*)}_{c^*_\ell}))$. Therefore, the corresponding GGAM query made while reconstructing the final forgery adds the tuple $(\tau(C^*_\ell), \mathrm{sign}(c^*_\ell)\hat{u}_{i^*,|c^*_\ell|}, r^*_\ell)$ to $\mathcal{L}$. Since $\hat{u}_{i^*,|c^*_\ell|} \neq \hat{u}_{j,|c^*_\ell|}$ and $\hat{u}_{i^*,|c^*_\ell|} \neq \mathbf{0}$, in either case (whether or not j is corrupted), $\mathcal{L}$ contains two distinct symbolic representations of $\tau(C^*_\ell)$, and a bridge event occurs.

It remains to consider the case $\mathbf{c}^* = \mathbf{0}$. In this case, every commitment in the signing transcript has the form $\tau(C^*_\ell) = \mathcal{O}^{\exp}(r^*_\ell, \tau(E_0))$, and hence contains *no* coefficient corresponding to the signing user. Consequently, the same transcript may be reused for a different target user without creating a second symbolic representation. Let $\mathsf{ZeroChal}$ denote this event. Since every signing query uses a uniform challenge vector,

$$\Pr[\mathsf{ZeroChal}] \leq \frac{q_{\mathsf{S}}}{(2S-1)^t}.$$

Bad Events. Besides BRIDGE, the preceding analysis uses the following events:

- $\mathsf{FailToSign}$: the signing simulation aborts since w was already queried to H.

– FailToFind: $\mathcal{B}$ outputs a valid forgery but at least one C_j^* was not recorded in $\mathcal{L}$ when $H(w^*)$ was recorded. (equivalently, the verifier-reconstructed objects are "fresh").
– MissQuery: w^* was not previously recorded in Hresp.
– BadQuery: w^* was first entered on an RO query and $\mathbf{c}^* = \mathsf{H}(w^*) \in \mathsf{Cand}(w^*)$.
– ZeroChal: w^* was first programmed during a signing query and $\mathbf{c}^* = \mathbf{0}$.

From the discussion above, we have the implication

$$\mathsf{Forge} \Rightarrow \mathsf{BRIDGE} \vee \mathsf{FailToSign} \vee \mathsf{FailToFind} \vee \mathsf{MissQuery} \vee \mathsf{BadQuery} \vee \mathsf{ZeroChal}.$$

Moreover, $\Pr[\mathsf{MissQuery}] \leq (2S-1)^{-t}$ as shown in (i), $\Pr[\mathsf{BadQuery}] \leq \frac{q_\mathsf{H}\kappa_{\mathsf{BQ}}}{(2S-1)^t}$ from (ii), and $\Pr[\mathsf{ZeroChal}] \leq \frac{q_\mathsf{S}}{(2S-1)^t}$ from (iii). Now we analyze the event FailToFind with condition on some $\tau(C^*) \notin \mathcal{L}$ before the last step. Then $\tau(C^*)$ is a fresh encoding, hence can be viewed as uniformly random among at least $N - |\mathcal{L}|$ unused encodings. Therefore the probability that the RO input $w^* = (\tau(I^*)\|m^*)$ was already recorded (by one of the q_H RO queries or q_S programmed signing points) is at most $(q_\mathsf{H} + q_\mathsf{S})/(N - |\mathcal{L}|)$. If w^* was not previously recorded, then $\mathsf{H}(w^*)$ is fresh uniform in $\mathcal{C}^t$, so the probability it equals the required challenge vector is at most $|\mathcal{C}^t|^{-1} = (2S-1)^{-t}$. Furthermore, since $\mathcal{A}_{\mathsf{Bridge}}$ makes at most $\widehat{q}_\mathsf{G}$ group action queries, Theorem 1 gives $|\mathcal{L}| \leq M + 2\widehat{q}_\mathsf{G} + 1$. A union bound yields

$$\Pr[\mathsf{FailToFind}] \leq \frac{q_\mathsf{H} + q_\mathsf{S}}{N - (M + 2\widehat{q}_\mathsf{G} + 1)} + (2S-1)^{-t}.$$

Finally, the standard estimate for programming collisions as in the proof of claim 1 in [BL22, Appendix C.2] gives

$$\Pr[\mathsf{FailToSign}] \leq \frac{q_\mathsf{S}(q_\mathsf{H} + q_\mathsf{S})}{N}.$$

Combining the bounds above with Theorem 1, instantiated with $\widehat{q}_\mathsf{G}$ group action queries, gives the claimed bound.

Commitment Compression. We now consider the Merkle-compressed variant. The bridge challenger is unchanged and provides $\mathcal{A}_{\mathsf{Bridge}}$ with all challenge encodings $\{\mathcal{E}_h^{(i)} : i \in [T],\ h \in [S-1]\}$, which are initialized in the list $\mathcal{L}$ as before. The difference is that these encodings are not forwarded to $\mathcal{B}$. For each user i, $\mathcal{A}_{\mathsf{Bridge}}$ samples a Merkle key $K_i \leftarrow\!\$\ \{0,1\}^\lambda$, builds a Merkle tree over $(\mathcal{E}_1^{(i)}, \ldots, \mathcal{E}_{S-1}^{(i)})$, and gives $\mathcal{B}$ only the compressed public key $\mathsf{pk}_i = (\mathtt{Comm}_i, K_i)$, where $\mathtt{Comm}_i$ is the root.

The random-oracle and corruption simulations are unchanged. For a signing query for an unrevealed user i, the simulator samples $\mathbf{c} = (c_1, \ldots, c_t) \xleftarrow{\$} \mathcal{C}^t$ and $r_1, \ldots, r_t \leftarrow\!\$\ \mathbb{Z}_N$, and computes $\tau(C_j) \leftarrow \mathcal{O}^{\exp}(r_j, \tau(E_{c_j}^{(i)}))$ as before. For every $c_j \neq 0$, it additionally includes the committed leaf $\mathcal{E}_{|c_j|}^{(i)}$ and its Merkle authentication path π_j. Since $\mathcal{A}_{\mathsf{Bridge}}$ constructed the tree, these openings can be generated without knowing the secret exponents. The point $w = C_1\| \cdots \|C_t\|m$ is then programmed exactly as in the uncompressed proof. Signing queries to corrupted users are answered honestly as well.

Let BreakBind denote the event that, for some user i and position h, the adversary produces an accepting Merkle opening $(\widetilde{\mathcal{E}}, \widetilde{\pi})$ with $\widetilde{\mathcal{E}} \neq \mathcal{E}_h^{(i)}$. Condition on $\overline{\mathsf{BreakBind}}$. Suppose that $\mathcal{B}$ outputs a valid forgery (m^*, σ^*) under an uncorrupted user i^*. For every coordinate with $c_j^* \neq 0$, validity of

the Merkle proof and $\overline{\mathsf{BreakBind}}$ imply that the opened curve is precisely $\mathcal{E}^{(i^*)}_{|c^*_j|}$. Hence the verifier reconstructs $C^*_j = r^*_j \star E^{(i^*)}_{c^*_j}$, with the usual twist convention for negative challenges. Therefore the symbolic coefficient required by verification is exactly $\mathbf{v}^*_j = \mathbf{0}$ if $c^*_j = 0$, and $\mathbf{v}^*_j = \mathsf{Sign}(c^*_j)\hat{u}_{i^*,|c^*_j|}$ otherwise. We therefore obtain

$$\mathsf{Forge} \Rightarrow \mathsf{BRIDGE} \vee \mathsf{FailToSign} \vee \mathsf{FailToFind} \vee \mathsf{MissQuery} \vee \mathsf{BadQuery} \vee \mathsf{ZeroChal} \vee \mathsf{BreakBind}.$$

The GGAM query complexity is unchanged from the uncompressed case, so the same bound $\widehat{q}_{\mathsf{G}} = q_{\mathsf{G}} + tq_{\mathsf{S}} + t$ applies. Finally, the binding security of the Merkle commitment bounds $\Pr[\mathsf{BreakBind}]$ by the corresponding Merkle-binding advantage $\frac{q_{\mathsf{H}}+1}{2^\lambda}$ [BKV19, section 5.2]. □

Tightness of the Target-Switching Loss. The factor $\kappa_{\mathsf{BQ}} = \min\{T, t+1\}$ in the $\mathsf{BadQuery}$ bound is not merely an artifact of the proof for the uncompressed version of CSI-FiSh. We consider the following matching attack. Let $k := \min\{T-1, t\}$ and consider k distinct users. The adversary chooses a fresh message m and makes an RO query $w = C_1\| \dots \|C_t\|m$ such that $C_j = E_1^{(j)}$ for $j \in [k]$ and $C_j = E_0$ for $j \in \{k+1, \dots, t\}$. For every $j \in [k]$, consider the challenge vector $\mathbf{c}^{(j)} = (0, \dots, 0, 1, 0, \dots, 0)$ (i.e., 1 in the j^{th} position). If $\mathsf{H}(w) = \mathbf{c}^{(j)}$, the adversary leaves user j uncorrupted and corrupts the other $k-1$ represented users. For the j^{th} coordinate, it uses response 0, so that $\mathfrak{e} \star E_1^{(j)} = E_1^{(j)} = C_j$. For every other represented user $j' \neq j$, the challenge is 0; after corrupting user j', the adversary knows $\mathfrak{a}_{j',1} = [\mathfrak{g}]^{x_{j',1}}$ and uses this value as the response, yielding $\mathfrak{a}_{j',1} \star E_0 = E_1^{(j')} = C_{j'}$. The remaining coordinates are reconstructed from E_0 using response 0.

In addition, if there is an unrepresented user and $\mathsf{H}(w) = \mathbf{0}^t$, the adversary corrupts all k represented users and chooses an unrepresented user as the forgery target. The revealed secret values again reconstruct the first k commitments from E_0, while the remaining commitments are already equal to E_0. Thus, one RO query has $k+1 = \min\{T, t+1\}$ favorable challenge vectors, showing that the factor κ_{BQ} in the $\mathsf{BadQuery}$ bound cannot in general be removed. The same construction can use negatively indexed public-key curves and the corresponding negative challenges via the twist. This does not yield an additional factor of two, since the sign of each commitment coordinate is fixed before the RO response is sampled. It remains open whether the same matching attack carries over unchanged to the Merkle-compressed variant of CSI-FiSh.

	$N \approx 2^{256}$			$N \approx 2^{512}$			$N \approx 2^{1024}$		
T	$\lambda = 128$	256	512	$\lambda = 128$	256	512	$\lambda = 128$	256	512
2^{24}	125	124	123	125	252	251	125	252	507
2^{36}	125	124	123	125	252	251	125	252	507
2^{46}	125	124	123	125	252	251	125	252	507

Table 1: Concrete security bits for $S = 2^{18}$, where $(2(S-1))^t \approx 2^\lambda$.

Concrete Security Estimates. We now estimate the provable security of scalable CSI-FiSh under several parameter choices. For the ranges considered below, the multi-user term M^2/N remains non-dominant up to about $T = 2^{46}$. The original CSI-FiSh parameters [BKV19] target 128-bit security by choosing t such that $(2(S-1))^t \approx 2^\lambda$ with $\lambda = 128$. Our bound in Theorem 2 contains the terms

t^2q^2/N and $q\kappa_{\mathsf{BQ}}/2^\lambda$, where $\kappa_{\mathsf{BQ}} = \min\{T, t+1\}$. For $S = 2^{18}$ and $\lambda = 128$, we have $t \approx 7$ and $\kappa_{\mathsf{BQ}} = 8$, yielding security limited by approximately

$$\frac{\log_2 N}{2} - \log_2 t \text{ and } \lambda - \log_2 \kappa_{\mathsf{BQ}}.$$

Thus, for $N \approx 2^{256}$, the original parameters provide approximately 125 bits of provable security, rather than the nominal 128-bit target. This still exceeds the approximately 44 bits of security reported in [EKP20], while extending the analysis to the multi-user setting with as many as 2^{46} users—roughly 8,600 times the current world population.

Increasing N alone cannot overcome the $q\kappa_{\mathsf{BQ}}/2^\lambda$ term, while increasing λ alone leaves the t^2q^2/N term limiting when $N \approx 2^{256}$. To obtain λ_0 bits of provable security, it suffices approximately that

$$\log_2 N \geq 2\lambda_0 + 2\log_2 t, \;\; \lambda \geq \lambda_0 + \log_2 \kappa_{\mathsf{BQ}},$$

together with the multi-user constraint from M^2/N. Hence, unlike the previous bound, the revised analysis does not require doubling λ; only a modest increase in challenge entropy is needed once N is sufficiently large. Table 1 summarizes the resulting provable security for several choices of N, T, and λ, assuming $S = 2^{18}$.

4 Multi-User Security of CSI-FiSh with Preprocessing

In this section, we study CSI-FiSh under preprocessing attacks. We first show that plain CSI-FiSh, despite its tight online multi-user security from Section 3, admits a simple preprocessing attack using a zero challenge. We then develop the ROM+GGAM preprocessing framework needed to analyze CSI-FiSh variants that prevent this attack, and prove preprocessing security for such schemes.

4.1 Preprocessing Models and an Attack on Plain CSI-FiSh

The auxiliary-input (AI) model [Unr07] captures non-uniform preprocessing by allowing an unbounded preprocessing phase to inspect the ideal objects and output a bounded hint to the online adversary. Since this correlates the ideal objects with the hint, direct analysis is difficult.

Following Unruh's presampling technique and subsequent work [CDGS18, CDG18], we first analyze a structured *bit-fixing* (BF) model, where preprocessing fixes only a bounded number of input-output pairs and the remaining values are sampled uniformly subject to consistency. Security can then be transferred from BF to AI; recent work [BL25] extends this approach to combined idealized models.

For CSI-FiSh, the relevant model is ROM+GGAM, with ideal objects H and the labeling map $\tau : \mathcal{X} \to \{0,1\}^\rho$. We let P_1 and P_2 denote the numbers of fixed pairs for H and τ, respectively. Before giving the BF-to-AI translation, we show that plain CSI-FiSh already admits a simple preprocessing attack.

A Zero-Challenge Preprocessing Attack. For any message m and responses $r_1, \ldots, r_t \in \mathbb{Z}_N$, let $C_i = r_i \star E_0$ for $i \in [t]$ and $w = \tau(C_1) \| \ldots \| \tau(C_t) \| m$. We observe that if $\mathsf{H}(w) = \mathbf{0}$, then $\sigma = (\mathbf{r}, \mathbf{c})$ with $\mathbf{r} = (r_1, \ldots, r_t)$ and $\mathbf{c} = (c_1, \ldots, c_t) = (0, \ldots, 0)$ is a valid signature under *any* public key, since for every coordinate, verification reconstructs $C_i = r_i \star E_{c_i} = r_i \star E_0$ and hence does not use

any user-specific challenge curve. The same observation applies to the Merkle-compressed variant, since no public-key leaf opening is required for a zero challenge coordinate.

This gives a direct attack on the AI model, which works as follows. Let K be a parameter and fix a family $\{(m_\ell, r_{\ell,1}, \ldots, r_{\ell,t})\}_{\ell \in [L]}$, with $L \leq 2^K$, independently of H, with distinct messages m_ℓ. For each $\ell \in [L]$, define $C_{\ell,i} = r_{\ell,i} \star E_0$ and $w_\ell = \tau(C_{\ell,1}) \| \ldots \| \tau(C_{\ell,t}) \| m_\ell$. During preprocessing, the adversary searches for an index ℓ such that $\mathsf{H}(w_\ell) = \mathbf{0}$ and, if one exists, stores ℓ in its K-bit hint. Otherwise, the attack fails. Given a successful hint, the online adversary reconstructs the corresponding tuple and outputs the resulting signature with zero challenge.

Since the w_ℓ's are distinct, their RO responses are independent and uniform in $\mathcal{C}^t$, so the attack succeeds with probability $1 - \left(1 - \frac{1}{(2S-1)^t}\right)^L$. In particular, taking $L = 2^K$, its failure probability is at most $\exp\left(-2^{K - t\log_2(2S-1)}\right)$.[2] Thus, the attack already has constant success probability around $K = t\log_2(2S-1)$ and rapidly approaches probability 1 beyond it; for CSI-FiSh, this threshold is approximately λ since $(2S-1)^t \approx 2^\lambda$.

This vulnerability is even more immediate in the BF model. There, the preprocessing phase can *fix* a RO input-output pair $\mathsf{H}(\tau(C_1) \| \ldots \| \tau(C_t) \| m) = \mathbf{0}$ for commitments $C_i = r_i \star E_0$ chosen during preprocessing and fix the corresponding labels when necessary. The online adversary can then use the resulting tuple $(m, r_1, \ldots, r_t)$ to forge under any public key with probability 1.

Preventing the Attack. There are two natural ways to prevent the zero-challenge attack. The first is to exclude the zero challenge from the Fiat-Shamir challenge space. We refer to the resulting scheme as *nonzero CSI-FiSh*. The second is to include the public key in the Fiat-Shamir hash input, i.e., $\mathbf{c} = \mathsf{H}(\mathsf{pk} \| \tau(C_1) \| \ldots \| \tau(C_t) \| m)$. We refer to this variant as *key-prefixed CSI-FiSh*.

We focus on the key-prefixed variant in the main text. Besides preventing the preprocessing attack above, key-prefixing binds each hash query to its intended target user. Consequently, a hash point prepared *before* the public keys are generated cannot be reused for a later target unless its fixed prefix matches that target's public key. Moreover, the target-switching and cross-user reuse from signing queries from Section 3 disappear. The modification only incurs the cost of hashing the public key together with the existing Fiat-Shamir input. We discuss the nonzero variant separately in Appendix B.

4.2 BF-to-AI Translation in the ROM+GGAM

The ROM+GGAM Preprocessing Models. We recall only the part of the BF-to-AI framework needed for our application. A *bit-fixing* source for the ROM+GGAM fixes at most P_1 input-output pairs of the RO H and at most P_2 input-output pairs of the injective labeling map $\tau : \mathcal{X} \to \{0,1\}^\rho$, while the remaining values are sampled uniformly, subject to τ remaining injective. This is the direct analogue of the ROM+GGM source used in [BL25]. We give formal source definitions and the proof of the dense-source lemma in Appendix A.1.

We write $\mathsf{BF\text{-}RO\text{+}GGA}(P_1, P_2, m, \mathcal{Y}, \mathcal{X}, \rho)$ for the BF model in which the preprocessing phase may fix such P_1 and P_2 points, and $\mathsf{AI\text{-}RO\text{+}GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$ for the AI model in which the preprocessing adversary may inspect the full tables of H and τ and output a K-bit hint. In particular, We write $\mathcal{F}_\mathsf{H} \subseteq \{0,1\}^m \times \mathcal{Y}$ and $\mathcal{F}_\tau \subseteq \mathcal{X} \times \{0,1\}^\rho$ for the sets of RO and labeling-map input/output pairs fixed during preprocessing, respectively, with $|\mathcal{F}_\mathsf{H}| \leq P_1$ and $|\mathcal{F}_\tau| \leq P_2$. In both models,

[2] Using $1 - x \leq e^{-x}$ with $x = (2S-1)^t$, we have $(1 - 1/(2S-1)^t)^{2^K} \leq \exp(-2^K/(2S-1)^t) = \exp(-2^{K - t\log_2(2S-1)})$.

the online phase gives the adversary ordinary access to the random oracle and the generic group-action oracles, together with a K-bit hint produced in the preprocessing phase. Thus, BF restricts preprocessing to the fixed entries, while AI allows preprocessing over the entire tables of H and τ.

Security Convention. Following [BL25], we write $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$ for a preprocessing/online attacker, and $\mathrm{Adv}_{\mathcal{G},\mathsf{M}}(\mathcal{A})$ for its probability in application $\mathcal{G}$ and model M. We say that $\mathcal{G}$ is $(K, T_1, T_2, \varepsilon)$-secure in model M if every attacker with K-bit preprocessing hint and at most T_1 online RO queries and T_2 online generic group-action queries satisfies $\mathrm{Adv}_{\mathcal{G},\mathsf{M}}(\mathcal{A}) \leq \varepsilon$. Formal definitions are given in Appendix A.2.

The quantities $(T_1)^{\mathsf{comb}}_{\mathcal{G}}$ and $(T_2)^{\mathsf{comb}}_{\mathcal{G}}$ denote the corresponding combined query complexities of the online adversary and the challenger in the game $\mathcal{G}$.

From Bit-Fixing to Auxiliary-Input. We use the multiplicative BF-to-AI translation for *unpredictability applications*, namely security games in which the adversary must output a specific winning object, such as a preimage, a collision, or a forged signature, rather than distinguish between two experiments. Since signature forgery is an unpredictability application, we use the multiplicative version of the BF-to-AI translation for the ROM+GGAM.

Theorem 3 gives the ROM+GGAM analogue of the ROM+GGM statement of [BL25]. Its proof follows the same dense-source strategy, but with one model-specific change: each group-action or twist query requires one inverse lookup to recover the underlying $\mathcal{X}$-element and one forward lookup to label the resulting action output. This accounts for the $2(T_2)^{\mathsf{comb}}_{\mathcal{G}} \log|\mathcal{X}|$ term in the condition below. The formal proof is deferred to Appendix A.2.

Theorem 3. *For all $\gamma > 0$, if an* unpredictability *application $\mathcal{G}$ is $(K, T_1, T_2, \varepsilon')$-secure in* BF-RO+GGA *$(P_1, P_2, m, \mathcal{Y}, \mathcal{X}, \rho)$ for every $P_1, P_2 \in \mathbb{N}$ satisfying*

$$(K + \log \gamma^{-1}) \cdot \Big((T_1)^{\mathsf{comb}}_{\mathcal{G}} \log|\mathcal{Y}| + 2(T_2)^{\mathsf{comb}}_{\mathcal{G}} \log|\mathcal{X}| \Big) \leq P_1 \log|\mathcal{Y}| + P_2 \log(|\mathcal{X}|/e), \tag{2}$$

then it is $(K, T_1, T_2, \varepsilon)$-secure in AI-RO+GGA*$(m, \mathcal{Y}, \mathcal{X}, \rho)$, for $\varepsilon \leq 2\varepsilon' + 2\gamma$, where e is Euler's constant.*

4.3 Multi-User Security of Key-Prefixed CSI-FiSh with Preprocessing

We now establish the multi-user security of *key-prefixed* CSI-FiSh with preprocessing, allowing adaptive corruption of users. We first extend the group action bridge-finding game of Section 3 to the BF-(ROM+GGAM), and then reduce the key-prefixed CSI-FiSh security to this game. Finally, we lift the resulting BF bound to the AI model using Theorem 3.

Bridge Finding in the BF-(ROM+GGAM). We extend the bridge-finding game of Figure 4 to the BF-(ROM+GGAM) setting. The only new ingredients are preprocessing-fixed entries of the RO H and the injective labeling map τ. Accordingly, we keep the procedures `Record`, `Corrupt`, and the group-action and twist oracles exactly as in Figure 4, and describe only the additional preprocessing specific initialization below in Figure 5.

The game differs from Figure 4 only through the preprocessing initialization: the at most P_2 labeling entries are inserted into $\mathcal{L}$ with zero coefficient vector. We remark that the fixed RO entries during preprocessing are handled separately in the key-prefixed CSI-FiSh reduction.

$\mathsf{BF\text{-}CorrBRIDGE}_{\mathcal{A}}^{N,(T,S)}(P_1,P_2)$:

1 : $(\mathcal{F}_{\mathsf{H}},\mathcal{F}_\tau,z)\leftarrow\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}(\mathcal{E}_0)$
2 : Initialize $\mathcal{L}$, Revealed, win and $\{\mathcal{E}_j^{(i)}\}_{i\in[T],j\in[S-1]}$ as in Figure 4
3 : **for** $(E',\mathcal{E}')\in\mathcal{F}_\tau$ **do**
4 : $\quad b\leftarrow\mathtt{GAI}(\mathcal{E}')$
5 : $\quad\mathtt{Record}(\mathcal{E}',\mathbf{0},b)$
6 : $o\leftarrow\mathcal{A}_{\mathsf{on}}^{\mathsf{H},\mathcal{O}^{\exp},\mathcal{O}^{\mathsf{tw}},\mathtt{Corrupt}}(z,\mathcal{E}_0,\{\mathcal{E}_j^{(i)}\}_{i\in[T],j\in[S-1]})$
7 : **return** win

Fig. 5: The preprocessing extension of the group action bridge-finding game. `Record`, `Corrupt`, and the handling of $\mathcal{O}^{\exp}$ and $\mathcal{O}^{\mathsf{tw}}$ are exactly as in Figure 4.

Theorem 4. *For any adversary* $\mathcal{A}=\left(\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)},\mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}\right)$, *where* $\mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}$ *makes at most* $q_{\mathsf{G}}^{\mathsf{on}}$ *group-action oracle queries, we have*

$$\Pr\left[\mathsf{BF\text{-}CorrBRIDGE}_{\mathcal{A}}^{N,(T,S)}(P_1,P_2)=1\right]\le\varepsilon,$$

with

$$\varepsilon=\frac{M+\binom{M}{2}+MP_2}{N}+\frac{q_{\mathsf{G}}^{\mathsf{on}}(M+P_2)+q_{\mathsf{G}}^{\mathsf{on}}(q_{\mathsf{G}}^{\mathsf{on}}+1)}{N-2P_2(M+2q_{\mathsf{G}}^{\mathsf{on}}+1)-(M+2q_{\mathsf{G}}^{\mathsf{on}}+1)^2-M-P_2},$$

where $N=|\mathrm{Cl}(\mathfrak{O})|$ *and* $M\coloneqq T(S-1)$.

Proof. As in the proof of Theorem 1, we first account for bridges that occur during initialization, which we denote InitBr. Besides the base tuple $(\mathcal{E}_0,\mathbf{0},0)$, the challenger records the M hidden public-key tuples and at most P_2 pre-fixed tuples $(\mathcal{E}'_k,\mathbf{0},b_k)$, where $E'_k=[\mathfrak{g}]^{b_k}\star E_0$ and $\mathcal{E}'_k=\tau(E'_k)$. An initial bridge can therefore occur only if a hidden exponent equals 0, equals another hidden exponent, or equals one of the b_k's. Hence, $\Pr[\mathsf{InitBr}]\le\frac{M+\binom{M}{2}+MP_2}{N}$.

Conditioning on $\overline{\mathsf{InitBr}}$, the online analysis follows the proof of Theorem 1; we only update the counting to account for the preprocessing tuples.

Let $\mathcal{L}_0:=\{(\mathcal{E},\mathbf{a},b)\in\mathcal{L}:\mathbf{a}=\mathbf{0}\}$. Throughout the online phase, $|\mathcal{L}|\le 1+M+P_2+2q_{\mathsf{G}}^{\mathsf{on}}$ and $|\mathcal{L}_0|\ge P_2$. As in the bit-fixing analysis of [BL25, Lemma 1], pairs of tuples both lying in $\mathcal{L}_0$ impose no restriction on the unrevealed coordinate $x[r]$ that is sampled last in the argument of Theorem 1. Hence,

$$|\mathcal{L}|^2-|\mathcal{L}_0|^2\le(1+M+P_2+2q_{\mathsf{G}}^{\mathsf{on}})^2-P_2^2=(M+2q_{\mathsf{G}}^{\mathsf{on}}+1)^2+2P_2(M+2q_{\mathsf{G}}^{\mathsf{on}}+1).$$

Moreover, conditioning on $\overline{\mathsf{InitBr}}$ excludes at most $M+P_2$ further values of $x[r]$. Thus, $x[r]$ remains uniform over a set of size at least $N-2P_2(M+2q_{\mathsf{G}}^{\mathsf{on}}+1)-(M+2q_{\mathsf{G}}^{\mathsf{on}}+1)^2-M-P_2$.

The remainder is identical to Theorem 1. The P_2 preprocessing tuples add at most P_2 possible prior representations against which each online output may collide. Therefore, summing over the at most $q_{\mathsf{G}}^{\mathsf{on}}$ online queries gives

$$\frac{q_{\mathsf{G}}^{\mathsf{on}}(M+P_2)+q_{\mathsf{G}}^{\mathsf{on}}(q_{\mathsf{G}}^{\mathsf{on}}+1)}{N-2P_2(M+2q_{\mathsf{G}}^{\mathsf{on}}+1)-(M+2q_{\mathsf{G}}^{\mathsf{on}}+1)^2-M-P_2}.$$

Adding the initialization bound $\Pr[\mathsf{InitBr}]$ proves the theorem. □

Multi-User Security in the BF-(ROM+GGAM). We analyze the uncompressed key-prefixed variant of CSI-FiSh, which we denote by $\Pi_{S,t}^{\text{KP-CSI}}$. For brevity, let $\varepsilon_{\mathsf{Br}}(P_2, q_{\mathsf{G}}^{\mathsf{on}})$ denote the upper bound in Theorem 4.

Theorem 5. *Let* $\mathcal{A} = \left(\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}, \mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}\right)$ *be a bit-fixing adversary, where* $\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}$ *fixes at most* P_1 *input/output pairs of the random oracle* H *and at most* P_2 *input/output pairs of the labeling map* τ*. Suppose that* $\mathcal{A}_{\mathsf{on}} := \mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}$ *makes at most* $q_{\mathsf{G}}^{\mathsf{on}}$ *generic group action queries,* $q_{\mathsf{H}}^{\mathsf{on}}$ *random oracle queries, and* $q_{\mathsf{S}}^{\mathsf{on}}$ *signing queries.*

Then we have $\Pr\left[\mathsf{SigForgeCor}_{\mathcal{A}_{\mathsf{on}},\Pi_{S,t}^{\text{KP-CSI}}}^{T,\mathsf{H},\mathcal{O}}(\lambda) = 1\right] \le \varepsilon$*, where*

$$\begin{aligned}\varepsilon &= \varepsilon_{\mathsf{Br}}(P_2, \widehat{q}_{\mathsf{G}}^{\mathsf{on}}) + \frac{TP_1}{N^{S-1}} + \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}})}{N} + \frac{q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}}{N - (M + P_2 + 2\widehat{q}_{\mathsf{G}}^{\mathsf{on}} + 1)} + \frac{q_{\mathsf{H}}^{\mathsf{on}} + 2}{(2S-1)^t} \\ &= O\left(\frac{M(M+P_2)}{N} + \frac{tq^{\mathsf{on}}(tq^{\mathsf{on}} + M + P_2)}{N} + \frac{q^{\mathsf{on}}}{2^\lambda}\right),\end{aligned}$$

with $N = |\mathrm{Cl}(\mathfrak{O})|$*,* $M = T(S-1)$*,* $\widehat{q}_{\mathsf{G}}^{\mathsf{on}} := q_{\mathsf{G}}^{\mathsf{on}} + t(q_{\mathsf{S}}^{\mathsf{on}}+1) + \min\{M, t(q_{\mathsf{S}}^{\mathsf{on}}+1)\}$*, and* $q^{\mathsf{on}} = q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{G}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}$ *denotes the online query budget of the signature attacker.*

Proof. We follow the reduction of Theorem 2, with two changes: (1) the preprocessing-fixed entries must be accounted for, and (2) every Fiat-Shamir hash input is prefixed by the signing public key. Given a BF adversary $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$ against $\Pi_{S,t}^{\text{KP-CSI}}$, we construct an adversary $\mathcal{B} = (\mathcal{B}_{\mathsf{pre}}, \mathcal{B}_{\mathsf{on}})$ against the BF group action bridge-finding game. The preprocessing algorithm $\mathcal{B}_{\mathsf{pre}}$ runs $\mathcal{A}_{\mathsf{pre}}$ and forwards the same $(\mathcal{F}_{\mathsf{H}}, \mathcal{F}_\tau, z)$ to the bridge game. The bridge challenger incorporates at most P_2 entries of $\mathcal{F}_\tau$ into $\mathcal{L}$ as in Figure 5. The online algorithm $\mathcal{B}_{\mathsf{on}}$ runs the same simulation as in Theorem 2, initializing Hresp with $\mathcal{F}_{\mathsf{H}}$ and using $w = \mathsf{pk}\|\tau(C_1)\|\ldots\|\tau(C_t)\|m$ for every signing or verification hash input. As in Theorem 2, $\mathcal{B}_{\mathsf{on}}$ makes at most $\widehat{q}_{\mathsf{G}}^{\mathsf{on}} = q_{\mathsf{G}}^{\mathsf{on}} + t(q_{\mathsf{S}}^{\mathsf{on}}+1) + \min\{M, t(q_{\mathsf{S}}^{\mathsf{on}}+1)\}$ group action queries to the bridge challenger.

Let KeyHit denote the event that the public key prefix of some input fixed in $\mathcal{F}_{\mathsf{H}}$ equals an honestly generated public key pk_i for some $i \in [T]$. For any fixed preprocessing view and any fixed public key prefix, injectivity of τ and freeness of the action determine at most one value of each of the $S-1$ independent secret exponents $x_{i,1}, \ldots, x_{i,S-1}$. Hence, a fresh public key matches this prefix with probability at most $N^{-(S-1)}$, and union bounding over the T users and P_1 fixed RO inputs gives $\Pr[\mathsf{KeyHit}] \le \frac{TP_1}{N^{S-1}}$.

Now, condition on $\overline{\mathsf{KeyHit}}$. Then no preprocessing-fixed RO entry can coincide with a signing point or with the forgery hash input for an honest target. Hence, the bounds for $\mathsf{FailToSign}$, $\mathsf{FailToFind}$, and $\mathsf{MissQuery}$ from Theorem 2 apply with the preprocessing-fixed RO entries omitted. Since $\mathcal{L}$ contains at most P_2 *additional* preprocessing tuples and $\mathcal{B}_{\mathsf{on}}$ makes at most $\widehat{q}_{\mathsf{G}}^{\mathsf{on}} = q_{\mathsf{G}}^{\mathsf{on}} + tq_{\mathsf{S}}^{\mathsf{on}} + t$ group action queries, we have $\Pr[\mathsf{FailToSign}] \le \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}}+q_{\mathsf{S}}^{\mathsf{on}})}{N}$, $\Pr[\mathsf{FailToFind}] \le \frac{q_{\mathsf{H}}^{\mathsf{on}}+q_{\mathsf{S}}^{\mathsf{on}}}{N-(M+P_2+2\widehat{q}_{\mathsf{G}}^{\mathsf{on}}+1)} + \frac{1}{(2S-1)^t}$, and $\Pr[\mathsf{MissQuery}] \le \frac{1}{(2S-1)^t}$.

It remains to revisit the two cases in Theorem 2 affected by key-prefixing.

1. First, if the forgery hash input w^* was first defined by an online RO query, then its prefix already fixes the target user i^*. Thus, there is *at most one* challenge vector compatible with the symbolic representations of the reconstructed commitments, rather than κ_{BQ} possible target-dependent candidates. Hence, $\Pr[\mathsf{BadQuery}] \le \frac{q_{\mathsf{H}}^{\mathsf{on}}}{(2S-1)^t}$, and if the sampled response is different, we can apply the same argument of Theorem 2 to extract a bridge.

2. Second, suppose w^* was first programmed by a signing query for user j and message m. Equality of the key-prefixed hash inputs implies $\mathsf{pk}_j = \mathsf{pk}_{i^*}$ and $m = m^*$. Conditioned on no bridge, distinct honest users have distinct public keys, so we have $j = i^*$, which contradicts freshness of the forgery. Hence, the zero-challenge bad event disappears entirely.

Therefore, we have

$$\mathsf{Forge} \Rightarrow \mathsf{BRIDGE} \vee \mathsf{KeyHit} \vee \mathsf{FailToSign} \vee \mathsf{FailToFind} \vee \mathsf{MissQuery} \vee \mathsf{BadQuery}.$$

Applying Theorem 4 with $\widehat{q}_{\mathsf{G}}^{\mathsf{on}}$ group action queries and combining the bounds above proves the claim. □

Multi-User Security in the AI-(ROM+GGAM) We now lift the preceding BF security bound to the AI model using Theorem 3.

Theorem 6. *Let* $\mathcal{A} = \left(\mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO+GGA}}, \mathcal{A}_{\mathsf{on}}^{\mathsf{AI\text{-}RO+GGA}}\right)$ *be a pair of auxiliary-input adversaries where* $\mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO+GGA}}$ *outputs a* K*-bit hint* $\mathsf{str}_{\mathsf{H},\tau}$*. Suppose* $\mathcal{A}_{\mathsf{on}}^{\mathsf{AI\text{-}RO+GGA}}$ *takes the hint* $\mathsf{str}_{\mathsf{H},\tau}$*, and makes at most* $q_{\mathsf{G}}^{\mathsf{on}}$ *generic group action queries,* $q_{\mathsf{H}}^{\mathsf{on}}$ *random oracle queries and* $q_{\mathsf{S}}^{\mathsf{on}}$ *signing queries. Let* $\widehat{q}_{\mathsf{G}}^{\mathsf{on}} = q_{\mathsf{G}}^{\mathsf{on}} + t(q_{\mathsf{S}}^{\mathsf{on}} + 1) + \min\{M, t(q_{\mathsf{S}}^{\mathsf{on}} + 1)\}$ *and* $q^{\mathsf{on}} := q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{G}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}$ *denotes the online query budget of the signature attacker.*

Then we have $\Pr\left[\mathsf{SigForgeCor}^{T,\mathsf{H},\mathcal{O}}_{\mathcal{A}_{\mathsf{on}}^{\mathsf{AI\text{-}RO+GGA}},\Pi_{S,t}^{\text{KP-CSI}}}(\lambda) = 1\right] \le 2\varepsilon + 2^{-\lambda+1}$*, where*

$$\varepsilon = \varepsilon_{\mathsf{Br}}(2(K+\lambda)Q, \widehat{q}_{\mathsf{G}}^{\mathsf{on}}) + \frac{2TQ(K+\lambda)}{N^{S-1}} + \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}})}{N} + \frac{q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}}{N - (M + 2(K+\lambda)Q + 2\widehat{q}_{\mathsf{G}}^{\mathsf{on}} + 1)} + \frac{q_{\mathsf{H}}^{\mathsf{on}} + 2}{(2S-1)^t}$$
$$= O\left(\frac{(K+\lambda)(tq^{\mathsf{on}} + M)^2}{N} + \frac{q^{\mathsf{on}}}{2^\lambda}\right),$$

with $Q := t(q^{\mathsf{on}} + 1) + M + S$, $N = |\mathrm{Cl}(\mathfrak{O})|$ *and* $M = T(S-1)$.

Proof. We apply Theorem 3 to Theorem 5 with $\gamma = 2^{-\lambda}$. Let $\mathcal{G}_{\mathsf{KP}}$ denote the adaptive-corruption signature forgery application for $\Pi_{S,t}^{\text{KP-CSI}}$ defined by $\mathsf{SigForgeCor}$. Recall that $q^{\mathsf{on}} = q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{G}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}$ denotes the online query budget of the signature attacker. Then we observe that

- the combined RO query complexity of the forgery game is at most $(T_1)_{\mathcal{G}_{\mathsf{KP}}}^{\mathsf{comb}} \le q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + 1 \le t(q^{\mathsf{on}} + 1) + M + S = Q$ since $\mathcal{A}_{\mathsf{on}}$ makes at most $q_{\mathsf{H}}^{\mathsf{on}}$ RO queries, the signing oracle makes at most $q_{\mathsf{S}}^{\mathsf{on}}$ RO queries, and forgery verification requires one RO query.
- Plus, the combined group action query complexity is at most $(T_2)_{\mathcal{G}_{\mathsf{KP}}}^{\mathsf{comb}} \le q_{\mathsf{G}}^{\mathsf{on}} + M + tq_{\mathsf{S}}^{\mathsf{on}} + t + \min\{t, S-1\} \le t(q^{\mathsf{on}} + 1) + M + S = Q$ since $\mathcal{A}_{\mathsf{on}}$ makes at most $q_{\mathsf{G}}^{\mathsf{on}}$ group action queries, key generation algorithm makes at most M action queries by making $(S-1)$ $\mathcal{O}^{\text{exp}}$ calls per user, the signing oracle makes at most $tq_{\mathsf{S}}^{\mathsf{on}}$ action queries since each signing query makes t $\mathcal{O}^{\text{exp}}$ calls, and forgery verification makes at most $\min\{t, S-1\}$ twist calls and t $\mathcal{O}^{\text{exp}}$ calls.

Hence, we can set $P_1 = P_2 = 2(K+\lambda)Q$; then Equation (2) holds and hence we get the intended result. □

Theorem 6 gives concrete insight into the preprocessing security of CSI-FiSh. For a hint of size[3] $K = 2^{70}$, $S = 2^{18}$, and $N = 2^{256}$, Table 2 gives about 90 bits for $T = 2^{24}$, 78 bits for $T = 2^{36}$, and

[3] For scale, in 2025 Google stated that many Colossus filesystems have multiple exabytes of storage, including two exceeding 10 EB each, giving 20 EB total ($\approx 2^{67.1}$ bits).

58 bits for $T = 2^{46}$. In the latter two cases, the query-independent term $(K+\lambda)M^2/N$ becomes dominant; for $T = 2^{46}$,

$$\frac{(K+\lambda)M^2}{N} \approx \frac{2^{70} \cdot 2^{128}}{2^{256}} = 2^{-58}.$$

Increasing N removes this multi-user bottleneck. For $N = 2^{512}$, the attainable security is about 128, 217, and 216 bits for $\lambda = 128, 256, 512$, while for $N = 2^{1024}$ it is 128, 256, and 472 bits. The slight decrease for larger λ is due to $t = \lambda/\log_2(2S)$, which increases the quadratic $t^2(q^{\mathsf{on}})^2$ term. Thus, N, K, t, and M must be scaled jointly.

	$N = 2^{256}$			$N = 2^{512}$			$N = 2^{1024}$		
T	$\lambda = 128$	256	512	$\lambda = 128$	256	512	$\lambda = 128$	256	512
2^{24}	90	89	88	128	217	216	128	256	472
2^{36}	78	78	78	128	217	216	128	256	472
2^{46}	58	58	58	128	217	216	128	256	472

Table 2: Estimated provable bits of security for $S = 2^{18}$ and $K = 2^{70}$.

Remark 1. The bounds above apply only classically. The quantum parameter selection for CSI-FiSh is more delicate due to Kuperberg's algorithm [Kup05]. So obtaining concrete quantum bounds for MT-GAIP is left for future work.

5 Quantum Security of CSI-FiSh in the QROM+AGAM

In this section, we prove the multi-user security of scalable CSI-FiSh in the QROM against AGAM adversaries. First, following the tight adaptive reprogramming approach of Grilo, Hövelmanns, Hülsing, and Majenz [GHHM21], we reduce MU-EUF-CMA security to a multi-user key-only forgery game. Second, using the measure-and-reprogram technique of Don, Fehr, Majenz, and Schaffner [DFMS19, DFM20], we reduce key-only forgeries to an adaptive-target multi-user ID-soundness game. Finally, the algebraic representations supplied by the AGAM allow us to turn a successful adversary against the Σ-protocol Σ_{CSI} (see Figure 1) into an any-pair MT-GAIP adversary. Thus, the CSI-FiSh forgery advantage in the QROM is bounded in terms of the advantage against MT-GAIP.

5.1 The Algebraic Group Action Model

We first recall the algebraic group action model (AGAM) [DHK+23].

Definition 7 ((Quantum) Algebraic Group Action Algorithms [DHK+23]). *Let $G \cong \mathrm{Cl}(\mathfrak{O})$ act on $\mathcal{X}$ via $\star : G \times \mathcal{X} \to \mathcal{X}$. An algorithm $\mathcal{A}$ is* algebraic *if every output $y \in \mathcal{X}$ is accompanied by an explanation relative to a previously received element: if $x_1, \ldots, x_n \in \mathcal{X}$ are the received elements so far, then $\mathcal{A}$ outputs $(y, i, \mathfrak{a}, b) \in \mathcal{X} \times [n] \times G \times \{0,1\}$ such that*

$$y = y_{(i,\mathfrak{a},b)} := \begin{cases} \mathfrak{a} \star x_i, & b = 0, \\ \mathfrak{a} \star x_i^t, & b = 1. \end{cases}$$

A quantum adversary is algebraic if each output state $|y\rangle$ is accompanied by a quantum explanation $|i, \mathfrak{a}, b\rangle$ satisfying the analogous relation; we write this output as $\left|y_{(i,\mathfrak{a},b)}\right\rangle$.

We require the adversary to be algebraic because the AGAM explanations are used in the extraction step. When the adversary commits to group-action outputs in the identification game, it must provide representations of these outputs relative to previously received public curves. The reduction compares these representations with those induced by verification; if the two use different base curves, their equality yields an explicit relation between the corresponding public curves, and hence a solution to MT-GAIP.

5.2 Reduction to Multi-User Key-Only Forgery Game

We first reduce multi-user EUF-CMA security of CSI-FiSh to a key-only forgery game. This is the multi-user analogue of the intermediate step in [GHHM21, Theorem 3]: after replacing real signatures by simulated transcripts and accounting for adaptive random-oracle reprogramming, signing-oracle access can be removed. Since the adversary receives T public keys and may forge under any of them, we use the following multi-user variant of EUF-CMA$_0$.

Definition 8 (Multi-User Key-Only Forgery Game). *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vfy})$ and $T \geq 1$. In the game $\mathsf{MU\text{-}EUF\text{-}CMA}_0^{\Pi,T}(\mathcal{A})$, the challenger samples $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$ for all $i \in [T]$ and gives $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_T)$ to $\mathcal{A}$. The adversary has access to the random oracle H, but not to a signing oracle, and eventually outputs (i^*, m^*, σ^*). The game outputs 1 iff $i^* \in [T]$ and $\mathsf{Vfy}(\mathsf{pk}_{i^*}, m^*, \sigma^*) = 1$. We define*

$$\mathrm{Adv}_{\Pi,T}^{\mathsf{MU\text{-}EUF\text{-}CMA}_0}(\mathcal{A}) \coloneqq \Pr[\mathsf{MU\text{-}EUF\text{-}CMA}_0^{\Pi,T}(\mathcal{A}) = 1].$$

Theorem 7 gives the CSI-FiSh-specific multi-user analogue of the chosen-message-to-key-only reduction of Grilo et al. [GHHM21, Theorem 3]. Before we prove Theorem 7, we recall the commitment min-entropy definition and analyze the commitment min-entropy of CSI-FiSh Σ-protocol.

Definition 9 (Commitment Min-Entropy, [AABN02, ABD+25]). *For a sigma protocol Σ,*

$$\mathsf{MinEnt}(\Sigma) \coloneqq \max_{(x,w)} \max_{\mathsf{com}} \Pr[\mathsf{com} = \mathsf{com}' \mid (\mathsf{com}', \mathsf{state}) \leftarrow \mathcal{P}_1(x, w)],$$

where the first max ranges over the pairs that might be output by Gen_R. The commitment min-entropy of Σ is defined by $-\log(\mathsf{MinEnt}(\Sigma))$.

Lemma 2. $\mathsf{MinEnt}(\Sigma_{\mathrm{CSI}}) \leq N^{-t}$.

Proof. For CSi-FiSh, the commitment distribution is independent of the statement or witness, since the commitment generation procedure only uses fresh uniform randomness $\mathfrak{b}_1, \ldots, \mathfrak{b}_t$ for t parallel repetitions. Now let $\mathsf{com} = (X_1, \ldots, X_t) \in \mathcal{X}^t$ be any candidate commitment tuple. Since the action is free and transitive, for each $i \in [t]$, there exists unique $\mathfrak{p}_i \in G$ such that $X_i = \mathfrak{p}_i \star E_0$. Due to the uniqueness of $\mathfrak{p}_i$, for each $i \in [t]$,

$$\Pr[C_i = X_i] = \Pr[\mathfrak{b}_i = \mathfrak{p}_i] = \frac{1}{|G|} = \frac{1}{N}.$$

Also, since the random variables $\mathfrak{b}_1, \ldots, \mathfrak{b}_t$ are sampled independently, we have

$$\Pr[\mathsf{com} = \mathsf{com}'] = \prod_{i=1}^{t} \Pr[C_i = X_i] = \frac{1}{N^t}.$$

Since we have the same probability for every commitment tuple $\mathsf{com} \in \mathcal{X}^t$, we have

$$\max_{\mathsf{com}} \Pr[\mathsf{com} = \mathsf{com}'] = N^{-t}.$$

And since the commitment distribution is independent of the statement or witness, taking the maximum over (x, w) does not change the value. This completes the proof. □

Theorem 7. *Let T be the number of users and $\mathcal{A}$ be a (quantum) adversary against the* MU-EUF-CMA *security of scalable CSI-FiSh, making at most q_{S} (classical) signing queries to the signing oracle* `Signing` *and at most q_{H} quantum queries to the random oracle H across all users. Then there exist quantum adversaries $\mathcal{B}$ and $\mathcal{D}$ such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}}_{\Pi^{\mathsf{CSI\text{-}FiSh}}_{S,t},T}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}_0}_{\Pi^{\mathsf{CSI\text{-}FiSh}}_{S,t},T}(\mathcal{B}) + \mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{U+1}(\mathcal{D}) \\ + \frac{3q_{\mathsf{S}}}{2}\sqrt{\frac{q_{\mathsf{H}}+q_{\mathsf{S}}+1}{N^t}} + \frac{q_{\mathsf{S}}}{(2S-1)^t} + \frac{T^2 S}{N},\end{aligned}$$

where $U := T(2S-1)$ and the running time of $\mathcal{B}$ and $\mathcal{D}$ is about that of $\mathcal{A}$.

Proof. We follow the proof structure of [GHHM21, Theorem 3], but modify it in exactly the places where the *multi-user* CSI-FiSh setting differs from the single-user Fiat-Shamir setting. Concretely, there are four changes.

1. We work in the multi-user game and therefore generate T key pairs $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_T, \mathsf{sk}_T)$ and maintain T message lists $\mathcal{L}_1, \ldots, \mathcal{L}_T$ per each user, rather than a single list $\mathcal{L}_{\mathcal{M}}$.
2. We work with the compressed Fiat-Shamir syntax from Definition 3 and a forgery has the form (i^*, m^*, σ^*) with $\sigma^* = (\mathsf{chal}^*, \mathsf{resp}^*)$, and the corresponding commitment is reconstructed as $\mathsf{com}^* := \mathsf{RecCom}(\mathsf{pk}_{i^*}, \mathsf{chal}^*, \mathsf{resp}^*)$.
3. In the transition from G_1 to G_2, the multi-user HVZK term disappears because for CSI-FiSh, the signing simulation is exact and we achieve perfect zero-knowledge.
4. Most importantly, the final step $G_2 \to \mathsf{MU\text{-}EUF\text{-}CMA}_0$ no longer follows immediately from message freshness, since the hash input is $\mathsf{H}(\mathsf{com}, m)$ and does not contain the user identity. Instead, we perform a case analysis according to whether the final forgery point (com^*, m^*) was already programmed by a signing query to another user.

   We now formalize this argument through the sequence of games in Figure 6.

Game G_0. This is the original MU-EUF-CMA game for scalable CSI-FiSh. Hence,

$$\Pr[G_0 \Rightarrow 1] = \mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}}_{\Pi^{\mathsf{CSI\text{-}FiSh}}_{S,t},T}(\mathcal{A}).$$

Game G_1. In game G_1, we modify the signing oracle similarly as in [GHHM21, Theorem 3] except that we take user as input as well. On a signing query (j, m), the transcript is first sampled honestly according to the underlying ID scheme, with a fresh uniform challenge $\mathsf{chal} \leftarrow\!\$\ \{-S+1, \ldots, S-1\}^t$, and only afterwards is the random oracle reprogrammed to be a-posteriori consistent with that transcript.

The adaptive reprogramming position is the actual hash input (com, m). The random part is the commitment com, and for scalable CSI-FiSh, the commitment distribution is uniform over $\mathcal{X}^t$. Therefore, its maximal point probability is N^{-t}, which is exactly the commitment min-entropy we get from Lemma 2. Thus, the same argument as in [GHHM21, Theorem 3] yields

$$|\Pr[G_0 \Rightarrow 1] - \Pr[G_1 \Rightarrow 1]| \leq \frac{3q_{\mathsf{S}}}{2}\sqrt{\frac{q_{\mathsf{H}}+q_{\mathsf{S}}+1}{N^t}}.$$

Games $G_0 - G_2$:

1 : **for** $i = 1, \ldots, T$ **do**
2 : $\quad (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3 : $\mathbf{pk} \coloneqq (\mathsf{pk}_1, \ldots, \mathsf{pk}_T)$
4 : $(i^*, m_{i^*}, \sigma_{i^*}) \leftarrow \mathcal{A}^{\mathtt{Signing},|\mathsf{H}\rangle}(\mathbf{pk})$
5 : **if** $m_{i^*} \in \mathcal{L}_{i^*}$ **return** 0
6 : Parse $(\mathsf{chal}^*, \mathsf{resp}^*) \coloneqq \sigma_{i^*}$
7 : $\mathsf{com}^* \coloneqq \mathsf{RecCom}(\mathsf{pk}_{i^*}, \mathsf{chal}^*, \mathsf{resp}^*)$
8 : **if** $\mathsf{H}(\mathsf{com}^*, m^*) = \mathsf{chal}^*$ **return** 1
9 : **return** 0

$\mathtt{Signing}(j, m)$:

1 : $\mathcal{L}_j \coloneqq \mathcal{L}_j \cup \{m\}$
2 : $(\mathsf{com}, \mathsf{chal}, \mathsf{resp}) \leftarrow \mathsf{getTrans}(j, m)$ // G_0-G_1
3 : $(\mathsf{com}, \mathsf{chal}, \mathsf{resp}) \leftarrow \mathsf{Sim}(\mathsf{pk}_j)$ // G_2
4 : $\mathsf{H} \coloneqq \mathsf{H}^{(\mathsf{com}, m) \mapsto \mathsf{chal}}$ // G_1-G_2
5 : **return** $\sigma \coloneqq (\mathsf{chal}, \mathsf{resp})$

$\mathsf{getTrans}(j, m)$:

1 : $(\mathsf{com}, \mathsf{state}) \leftarrow \mathsf{Commit}(\mathsf{sk}_j)$
2 : $\mathsf{chal} \coloneqq \mathsf{H}(\mathsf{com}, m)$ // G_0
3 : $\mathsf{chal} \leftarrow\!\!\$\, \mathcal{C}^t$ // G_1
4 : $\mathsf{resp} \leftarrow \mathsf{Respond}(\mathsf{sk}_j, \mathsf{com}, \mathsf{chal}, \mathsf{state})$
5 : **return** $(\mathsf{com}, \mathsf{chal}, \mathsf{resp})$

Fig. 6: Games G_0-G_2 for the proof of Theorem 7.

Game G_2. In game G_2, we replace the honest signing transcripts by simulated transcripts. For CSI-FiSh, the signing simulation is exact, so the distributions in G_1 and G_2 are identical. Hence,

$$\Pr[G_1 \Rightarrow 1] = \Pr[G_2 \Rightarrow 1].$$

From G_2 *to* MU-EUF-CMA$_0$ *and* MT-GAIP. Let $\mathcal{B}$ be the adversary that is run in the MU-EUF-CMA$_0$ game and simulates G_2 for $\mathcal{A}$. Since G_2 uses only simulated transcripts, $\mathcal{B}$ needs no secret keys and can maintain the same programmed random oracle table $\mathcal{L}_\mathsf{H}$ as in G_2.

Suppose $\mathcal{A}$ outputs a forgery (i^*, m^*, σ^*) with $\sigma^* = (\mathsf{chal}^*, \mathsf{resp}^*)$, and let $\mathsf{com}^* \coloneqq \mathsf{RecCom}(\mathsf{pk}_{i^*}, \mathsf{chal}^*, \mathsf{resp}^*)$. We consider two cases.

1. If $(\mathsf{com}^*, m^*) \notin \mathcal{L}_\mathsf{H}$, then the point (com^*, m^*) was never programmed by the signing simulation. Consequently, the forged signature is also valid in the MU-EUF-CMA$_0$ game simulated by $\mathcal{B}$. Thus,
$$\Pr[G_2 \Rightarrow 1 \wedge (\mathsf{com}^*, m^*) \notin \mathcal{L}_\mathsf{H}] \leq \mathrm{Adv}^{\text{MU-EUF-CMA}_0}_{\Pi^{\text{CSI-FiSh}}_{S,t}, T}(\mathcal{B}).$$
2. If $(\mathsf{com}^*, m^*) \in \mathcal{L}_\mathsf{H}$, then (com^*, m^*) was already programmed by some previous signing query (j, m^*) for some $j \neq i^*$. Let the earlier simulated signature for user j on message m^* be $(\mathsf{chal}, \mathsf{resp})$, where $\mathsf{chal} = (c_1, \ldots, c_t)$ and $\mathsf{resp} = (\mathfrak{r}_1, \ldots, \mathfrak{r}_t)$. Because both signatures correspond to the same programmed point (com^*, m^*), they share the same challenge vector, i.e., $\mathsf{chal}^* = \mathsf{chal}$. Write $\mathsf{com}^* = (C_1^*, \ldots, C_t^*)$ and $\mathsf{resp}^* = (\mathfrak{r}_1^*, \ldots, \mathfrak{r}_t^*)$. Then for every coordinate $i \in [t]$, the earlier valid transcript and the final forgery satisfy $C_i^* = \mathfrak{r}_i \star E_{c_i}^{(j)}$ and $C_i^* = \mathfrak{r}_i^* \star E_{c_i}^{(i^*)}$. If there exists some i with $c_i \neq 0$, then we can extract an MT-GAIP solution as long as $E_{c_i}^{(j)} \neq E_{c_i}^{(i^*)}$, since we get $E_{c_i}^{(j)} = (\mathfrak{r}_i^{-1}\mathfrak{r}_i^*) \star E_{c_i}^{(i^*)}$ which is a valid MT-GAIP solution. Now we consider two failure events:
   - If $c_i \neq 0$ but $E_{c_i}^{(j)} = E_{c_i}^{(i^*)}$, then we cannot solve for an MT-GAIP. Let CurveColl be the event where this could happen, i.e.,
$$\mathsf{CurveColl} = \left\{ \exists i \neq j \in [T], \exists c \in \mathcal{C} \setminus \{0\} : E_c^{(i)} = E_c^{(j)} \right\}.$$

For a fixed (i, j, c), $\Pr\left[E_c^{(i)} = E_c^{(j)}\right] = 1/|G| = 1/N$. Union bounding over the number of user pairs $\binom{T}{2}$ and nonzero challenges $(2S-2)$, we get $\Pr[\mathsf{CurveColl}] \le \binom{T}{2}(2S-2)/N \le \frac{T^2 S}{N}$.

– The algorithm also fails to produce an MT-GAIP solution if $\mathsf{chal} = (0, \ldots, 0)$. We call this event ZeroChal. Since each simulated signing query samples $\mathsf{chal} \leftarrow\!\$\ \{-S+1, \ldots, S-1\}^t$, a fixed signing query produces all-zero challenge vector with probability $(2S-1)^{-t}$. By union bound over at most q_{S} signing queries, the probability that the final forgery reuses a programmed point from all-zero challenge vector is at most $\Pr[\mathsf{ZeroChal}] \le q_{\mathsf{S}}/(2S-1)^t$.

Taken together, we have

$$\begin{aligned}\Pr[G_2 \Rightarrow 1 \wedge (\mathsf{com}^*, m^*) \in \mathcal{L}_{\mathsf{H}}] &\le \mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{U+1}(\mathcal{D}) + \Pr[\mathsf{CurveColl}] + \Pr[\mathsf{ZeroChal}] \\ &\le \mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{U+1}(\mathcal{D}) + \frac{T^2 S}{N} + \frac{q_{\mathsf{S}}}{(2S-1)^t}.\end{aligned}$$

Finally, collecting all the bounds together, we get the desired result. □

Remark 2. T does not multiply the adaptive-reprogramming term, since reprogramming is indexed by signing queries rather than users. Its effect appears instead through the MT-GAIP instance and the setup-collision term.

5.3 From Multi-User Key-Only Forgery to Adaptive-Target Multi-User ID Soundness

We now consider the key-only forgery game. Since $\mathsf{MU\text{-}EUF\text{-}CMA}_0$ gives no signing-oracle access, any successful Fiat–Shamir forgery must be consistent with the random oracle at its reconstructed query (com, m). Using the multi-user measure-and-reprogram theorem (Lemma 3), adapted from [DFMS19, DFM20], we measure such a query and replace its response by a fresh challenge. Because the adversary receives all public keys and may choose the target user only after seeing the challenge, we obtain the adaptive-target multi-user identification-soundness game $\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}^T_{\Sigma,\mathsf{H}}(\mathcal{A})$ shown in Figure 7. Here, $\mathcal{QO}_{\mathrm{SWAP}}$ lets the adversary fix a commitment, possibly in superposition, before $\mathcal{O}_{\mathrm{CHAL}}$ samples the challenge. We define

$$\mathrm{Adv}^{\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}}_{\Sigma,\mathsf{H},T}(\mathcal{A}) := \Pr[\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}^T_{\Sigma,\mathsf{H}}(\mathcal{A}) = 1].$$

Lemma 3 (Adapted from [ABR26, Lemma C.3]). *Let $\mathcal{A}$ be a (quantum) adversary for $\mathsf{MU\text{-}EUF\text{-}CMA}_0$ making at most q_{H} quantum queries to the random oracle H. Then there exists a (quantum) adversary $\mathcal{B}$ for $\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}$ such that*

$$\mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}_0}_{\mathsf{SIG}[\Sigma,\mathsf{H}],T}(\mathcal{A}) \le (2q_{\mathsf{H}}+1)^2 \cdot \mathrm{Adv}^{\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}}_{\Sigma,\mathsf{H},T}(\mathcal{B}).$$

Proof. We follow the same measure-and-reprogram argument as in the proof of [ABR26, Theorem C.3], combined with Theorem 8. Let $\mathcal{A}$ be a quantum adversary against $\mathsf{MU\text{-}EUF\text{-}CMA}_0$ making at most q_{H} quantum queries to the random oracle H. We first modify $\mathcal{A}$ such that its output is of the form (query, z), where $\mathsf{query} = (\mathsf{com}, m)$ is the corresponding Fiat-Shamir query point. More precisely, suppose that $\mathcal{A}$ outputs a forgery (i^*, m^*, σ^*), where $\sigma^* = (\mathsf{chal}^*, \mathsf{resp}^*)$. From this output,

Game $\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}^T_\Sigma(\mathcal{A})$:

1 : **for** $i = 1, \ldots, T$ **do**
2 : $\quad (x_i, w_i) \leftarrow \mathsf{IG}(1^\lambda)$
3 : **endfor**
4 : $\mathtt{Q}.\mathsf{com} \coloneqq \bot$
5 : $\mathsf{flag} \coloneqq 0$
6 : $(i^*, \mathsf{resp}) \leftarrow \mathcal{A}^{\mathcal{O}_{\textsc{Chal}}, \mathcal{QO}_{\textsc{Swap}}}(x_1, \ldots, x_T)$
7 : Measure $\mathtt{Q}.\mathsf{com}$ to obtain com
8 : **return** $\mathsf{Ver}_\Sigma(x_{i^*}, \mathsf{com}, \mathsf{chal}, \mathsf{resp})$

Oracle $\mathcal{O}_{\textsc{Chal}}$:

1 : **if** $\mathsf{flag} = 0$ **then**
2 : $\quad \mathsf{chal} \leftarrow\!\!\$\, \mathcal{C}$
3 : $\quad \mathsf{flag} \coloneqq 1$
4 : **endif**
5 : **return** chal

Oracle $\mathcal{QO}_{\textsc{Swap}}(\mathtt{Q}.R)$:

1 : **if** $\mathsf{flag} = 0$ **then**
2 : $\quad$ Swap $\mathtt{Q}.\mathsf{com}$ and $\mathtt{Q}.R$
3 : **endif**

Fig. 7: An adaptive-target T-user MU-ID-Sound game $\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}^T_{\Sigma,\mathsf{H}}(\mathcal{A})$ for a Σ-protocol Σ.

we reconstruct the commitment $\mathsf{com}^* \coloneqq \mathsf{RecCom}(\mathsf{pk}_{i^*}, \mathsf{chal}^*, \mathsf{resp}^*)$, and define $\mathsf{query}' \coloneqq (\mathsf{com}^*, m^*)$ and $z \coloneqq (i^*, \mathsf{chal}^*, \mathsf{resp}^*)$. Thus, we may equivalently view the output of $\mathcal{A}$ as (query', z).

We define

$$V(\mathsf{query}, \mathsf{chal}, z) \coloneqq \mathbf{1}\Big[z = (i^*, \mathsf{chal}^*, \mathsf{resp}^*) \wedge \mathsf{chal} = \mathsf{chal}^* \\ \wedge \mathsf{Ver}_\Sigma(\mathsf{pk}_{i^*}, \mathsf{com}, \mathsf{chal}, \mathsf{resp}^*) = 1\Big].$$

Then $\mathcal{A}$ wins the $\mathsf{MU\text{-}EUF\text{-}CMA}_0$ game if $V(\mathsf{query}', \mathsf{H}(\mathsf{query}'), z) = 1$.

We define the adversary $\mathcal{B}$ for ad-MU-ID-Sound using the two-stage algorithm $\mathcal{T}^{\mathcal{A}}$ from Theorem 8. $\mathcal{B}$ runs the first stage of $\mathcal{T}^{\mathcal{A}}$ to obtain $\mathsf{query} = (\mathsf{com}, m)$. Next, it calls $\mathcal{QO}_{\textsc{Swap}}(\mathsf{com})$ to send com to the game. Then it obtains $\mathsf{chal} \leftarrow \mathcal{O}_{\textsc{Chal}}$ and runs the second stage of $\mathcal{T}^{\mathcal{A}}$ with input chal. Finally, $\mathcal{T}^{\mathcal{A}}$ outputs (query', z), where $z = (i^*, \mathsf{chal}^*, \mathsf{resp}^*)$, and $\mathcal{B}$ outputs (i^*, resp^*). $\mathcal{B}$ wins whenever $\mathsf{query} = \mathsf{query}'$ and $V(\mathsf{query}, \mathsf{chal}, z) = 1$. Indeed, if $\mathsf{query} = \mathsf{query}'$, then the commitment stored in the game is exactly the reconstructed commitment that appears in the forgery. Moreover, if $V(\mathsf{query}, \mathsf{chal}, z) = 1$, then $\mathsf{Ver}_\Sigma(\mathsf{pk}_{i^*}, \mathsf{com}, \mathsf{chal}, \mathsf{resp}^*) = 1$. Hence, (i^*, resp^*) is a valid winning output for the game ad-MU-ID-Sound. Now we can conclude by applying Theorem 8. □

Theorem 8 ([DFM20, Theorem 2]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box two-stage quantum algorithm $\mathcal{T}$ with the following property. Let $\mathcal{A}$ be a quantum oracle algorithm making at most q_H quantum queries to a uniformly random $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$ and that outputs some classical $x \in \mathcal{X}$ and a (possibly quantum) output z. Then, the two-stage algorithm $\mathcal{T}^{\mathcal{A}}$ outputs some $x \in \mathcal{X}$ in the first stage and, upon a random $y \in \mathcal{Y}$ as input to the second stage, a (possibly quantum) output z, so that for any $x_0 \in \mathcal{X}$ and any (possibly quantum) predicate V:*

$$\Pr_y\Big[x = x_0 \wedge V(x, y, z) = 1 \,|\, (x, z) \leftarrow \langle \mathcal{T}^{\mathcal{A}}, y\rangle\Big] \\ \geq \frac{1}{(2q_\mathsf{H} + 1)^2} \cdot \Pr_\mathsf{H}\Big[x = x_0 \wedge V(x, \mathsf{H}(x), z) = 1 \,|\, (x, z) \leftarrow \mathcal{A}^\mathsf{H}\Big].$$

Furthermore, $\mathsf{Time}(\mathcal{T}^{\mathcal{A}}) = \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\log|\mathcal{X}|, \log|\mathcal{Y}|, q_\mathsf{H})$.

5.4 From Adaptive-Target ID Soundness to Multi-Target GAIP

We instantiate adaptive-target MU-ID-Sound with the t-parallel CSI-FiSh Σ-protocol $\Sigma_{S,t}^{\mathsf{CSI}}$, as shown in Figure 8. In the AGAM, commitments have algebraic explanations relative to the public curves; whenever such an explanation differs from the curve used in the verifier equation, it yields a relation between public curves, reducing to any-pair MT-GAIP. This proves Lemma 4.

Game $\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}^{T}_{\Sigma_{\mathsf{CSI}}}(\mathcal{A})$:

1 : **for** $i = 1, \ldots, T$ **do**
2 : $\quad(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$
3 : **endfor**
4 : $\mathtt{Q}.C_1 := \bot, \ldots, \mathtt{Q}.C_t := \bot$
5 : $\mathsf{flag} := 0$
6 : $(i^*, r_1, \ldots, r_t) \leftarrow \mathcal{A}^{\mathcal{O}_{\textsc{Chal}}, \mathcal{QO}_{\textsc{Swap}}}(\mathsf{pk}_1, \ldots, \mathsf{pk}_T)$
7 : Measure $\mathtt{Q}.C_1, \ldots, \mathtt{Q}.C_t$ to obtain $C_1, \ldots, C_t$
8 : **for** $j = 1, \ldots, t$ **do**
9 : $\quad$**if** $C_j \neq r_j \star E_{c_j}^{(i^*)}$ **then**
10 : $\quad\quad$**return** 0
11 : $\quad$**endif**
12 : **endfor**
13 : **return** 1

Oracle $\mathcal{O}_{\textsc{Chal}}$:

1 : **if** $\mathsf{flag} = 0$ **then**
2 : $\quad\mathbf{c} = (c_1, \ldots, c_t) \leftarrow\!\!\$\ \mathcal{C}^t$
3 : $\quad\mathsf{flag} := 1$
4 : **endif**
5 : **return** c

Oracle $\mathcal{QO}_{\textsc{Swap}}(\mathtt{Q}.D_1, \ldots, \mathtt{Q}.D_t)$:

1 : **if** $\mathsf{flag} = 0$ **then**
2 : $\quad$**for** $j = 1, \ldots, t$ **do**
3 : $\quad\quad$Swap $\mathtt{Q}.C_j$ and $\mathtt{Q}.D_j$
4 : $\quad$**endfor**
5 : **endif**

Fig. 8: The algebraic adaptive-target T-user MU-ID-Sound game for CSI-FiSh: after seeing $\mathbf{c} = (c_1, \ldots, c_t)$, the adversary adaptively chooses i^* and must output $r_1, \ldots, r_t$ such that $C_j = r_j \star E_{c_j}^{(i^*)}$ for all $j \in [t]$.

Lemma 4. *Let $\mathcal{A}$ be a (quantum) algebraic adversary against the adaptive-target* MU-ID-Sound *game for $\Sigma_{S,t}^{\mathsf{CSI}}$. Then there exists a (quantum) algebraic adversary $\mathcal{B}$ for any-pair* MT-GAIP *such that*

$$\mathrm{Adv}^{\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}}_{\Sigma_{S,t}^{\mathsf{CSI}},\mathsf{H},T}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{U+1}(\mathcal{B}) + \frac{T}{(2S-1)^t},$$

where $U := T(2S-1)$ and the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Proof. $\mathcal{B}$ is given the initial public curve list $\mathcal{L} = \{E_0\} \cup \{E_j^{(i)} : i \in [T], j \in \{-S+1, \ldots, S-1\}\}$. Then $\mathcal{B}$ simulates the game $\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}^{T}_{\Sigma_{S,t}^{\mathsf{CSI}},\mathsf{H}}(\mathcal{A})$ for $\mathcal{A}$ using those public curves. At the end of the game, $\mathcal{B}$ measures the commitment registers and obtains $C_1, \ldots, C_t$, along with the explanations $C_j = \alpha_j \star E_{u_j}$ for some $E_{u_j} \in \mathcal{L}$ for all $j \in [t]$. Since $\mathcal{A}$ outputs $(i^*, r_1, \ldots, r_t)$ and wins the ad-MU-ID-Sound game, $C_j = r_j \star E_{c_j}^{(i^*)}$ holds for all $j \in [t]$. If for some j we have $E_{u_j} \neq E_{c_j}^{(i^*)}$, then $E_{u_j} = (\alpha_j^{-1} r_j) \star E_{c_j}^{(i^*)}$, which is a valid any-pair MT-GAIP solution among $U+1$ candidate curves.

Let bad be the event that $\mathcal{B}$ fails to win. Then bad occurs if and only if $E_{u_j} = E_{c_j}^{(i^*)}$ for all $j \in [t]$. For a fixed user $i \in [T]$, there is at most one challenge vector $\mathbf{c} \in \mathcal{C}^t$ for which this can hold. This occurs with probability at most $1/|\mathcal{C}|^t = 1/(2S-1)^t$, and since there are at most T users, by union bound we have $\Pr[\mathsf{bad}] \leq \frac{T}{(2S-1)^t}$. This completes the proof. □

5.5 Quantum Multi-User Security of CSI-FiSh in the QROM+AGAM

We are now ready to combine the reductions from the previous subsections and establish the quantum multi-user security of CSI-FiSh in the QROM+AGAM.

Theorem 9. *Let $\mathcal{A}$ be a (quantum) algebraic* MU-EUF-CMA *adversary for* CSI-FiSh *with T users, making at most q_{S} classical queries to the signing oracle and at most q_{H} quantum queries to the random oracle across all users. Then there exists a (quantum) algorithm $\mathcal{B}$ such that*

$$\mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}}_{\Pi^{\mathsf{CSI\text{-}FiSh}}_{S,t},T}(\mathcal{A}) \leq \varepsilon,$$

with

$$\varepsilon = 4(q_{\mathsf{H}}+1)^2\left(\mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{U+1}(\mathcal{B}) + \frac{T}{(2S-1)^t}\right) + \frac{3q_{\mathsf{S}}}{2}\sqrt{\frac{q_{\mathsf{S}}+q_{\mathsf{H}}+1}{N^t}} + \frac{q_{\mathsf{S}}}{(2S-1)^t} + \frac{T^2S}{N},$$

where $U \coloneqq T(2S-1)$ where the running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Proof. Let $\mathcal{A}$ be a (quantum) algebraic adversary for the MU-EUF-CMA game of CSI-FiSh. By Theorem 7, we obtain two algebraic algorithms $\mathcal{F}$ and $\mathcal{D}_1$ such that

$$\mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}}_{\Pi^{\mathsf{CSI\text{-}FiSh}}_{S,t},T}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}_0}_{\Pi^{\mathsf{CSI\text{-}FiSh}}_{S,t},T}(\mathcal{F}) + \mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{U+1}(\mathcal{D}_1) + \frac{3q_{\mathsf{S}}}{2}\sqrt{\frac{q_{\mathsf{H}}+q_{\mathsf{S}}+1}{N^t}} + \frac{q_{\mathsf{S}}}{(2S-1)^t} + \frac{T^2S}{N}.$$

By Lemma 3, there exists an algebraic adversary $\mathcal{R}$ such that

$$\mathrm{Adv}^{\mathsf{MU\text{-}EUF\text{-}CMA}_0}_{\Pi^{\mathsf{CSI\text{-}FiSh}}_{S,t},T}(\mathcal{F}) \leq (2q_{\mathsf{H}}+1)^2 \cdot \mathrm{Adv}^{\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}}_{\Sigma^{\mathsf{CSI}}_{S,t},\mathsf{H},T}(\mathcal{R}).$$

And by Lemma 4, there exists an algebraic adversary $\mathcal{D}_2$ for MT-GAIP such that

$$\mathrm{Adv}^{\mathsf{ad\text{-}MU\text{-}ID\text{-}Sound}}_{\Sigma^{\mathsf{CSI}}_{S,t},\mathsf{H},T}(\mathcal{R}) \leq \mathrm{Adv}^{\mathsf{MT\text{-}GAIP}}_{U+1}(\mathcal{D}_2) + \frac{T}{(2S-1)^t}.$$

Out of $\mathcal{D}_1$ and $\mathcal{D}_2$, let $\mathcal{B}$ be the algorithm that has the largest advantage for the Multi-GAIP. Using $(2q_{\mathsf{H}}+1)^2+1 \leq 4(q_{\mathsf{H}}+1)^2$ and putting the inequalities together, we get the desired result. □

Remark 3. The parameter S is typically fixed independently of the security parameter, with suggested values such as $S \approx 2^{12}$. The bound in Theorem 9 avoids a black-box multiplicative loss in the number of users T: the T public-key families are absorbed into an any-pair MT-GAIP instance, and the remaining T-dependence appears only through explicit error terms. For instance, even with $T = 2^{40}$ users and the high-end choice $S = 2^{18}$, the setup-collision term T^2S/N is only 2^{-158} when $N \approx 2^{256}$.

The q_{S}-dependent terms are negligible in the intended regimes: if $q_{\mathsf{H}} + q_{\mathsf{S}} + 1 \leq 2^{60}$ and $N \approx 2^{256}$, then $\frac{3q_{\mathsf{S}}}{2}\sqrt{(q_{\mathsf{H}}+q_{\mathsf{S}}+1)/N^t} \leq 2^{-165}$ already for $t = 2$, and for $S = 2^{18}$, $t = 12$, one has $\frac{q_{\mathsf{S}}}{(2S-1)^t} \leq 2^{-132}$. The factor $4(q_{\mathsf{H}}+1)^2$ is inherited from the QROM measure-and-reprogram step. Thus, our result avoids an additional multiplicative loss in T.

References

AABN02. Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, Berlin, Heidelberg, April / May 2002.

ABCP23. Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. CSI-SharK: CSI-FiSh with sharing-friendly keys. In Leonie Simpson and Mir Ali Rezazadeh Baee, editors, *ACISP 23*, volume 13915 of *LNCS*, pages 471–502. Springer, Cham, July 2023.

ABD+25. Marius A. Aardal, Andrea Basso, Luca De Feo, Sikhar Patranabis, and Benjamin Wesolowski. A complete security proof of SQIsign. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 190–222. Springer, Cham, August 2025.

ABR26. Marius A. Aardal, Andrea Basso, and Doreen Riepel. The algebraic isogeny model: A general model with applications to SQIsign and key exchanges. Cryptology ePrint Archive, Paper 2026/032, 2026.

BDK+22. Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 95–126. Springer, Cham, May / June 2022.

BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 464–492. Springer, Cham, December 2020.

BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Cham, December 2019.

BL22. Jeremiah Blocki and Seunghoon Lee. On the multi-user security of short Schnorr signatures with preprocessing. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 614–643. Springer, Cham, May / June 2022.

BL25. Jeremiah Blocki and Seunghoon Lee. On the preprocessing security of (nonzero) Schnorr signatures and PSEC-KEM. Cryptology ePrint Archive, Report 2025/138, 2025.

CDG18. Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 693–721. Springer, Cham, August 2018.

CDG+26. Ryann Cartor, Nathan Daly, Giulia Gaggero, Jason T. LeGrow, Andrea Sanguineti, and Silvia Sconza. Post-quantum adaptor signatures with strong security from cryptographic group actions. In *Post-Quantum Cryptography*, volume 16491 of *Lecture Notes in Computer Science*, pages 370–405. Springer, 2026.

CDGS18. Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 227–258. Springer, Cham, April / May 2018.

CS20. Daniele Cozzo and Nigel P. Smart. Sashimi: Cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol. In Jintai Ding and Jean-Pierre Tillich, editors, *PQCrypto 2020*, pages 169–186. Springer, Cham, 2020.

DEF+25. Pierrick Dartois, Jonathan Komada Eriksen, Tako Boris Fouotsa, Arthur Herlédan Le Merdy, Riccardo Invernizzi, Damien Robert, Ryan Rueger, Frederik Vercauteren, and Benjamin Wesolowski. PEGASIS: Practical effective class group action using 4-dimensional isogenies. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part I*, volume 16000 of *LNCS*, pages 67–99. Springer, Cham, August 2025.

DFK+23. Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: Scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 345–375. Springer, Cham, May 2023.

DFM20. Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round Fiat-Shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 602–631. Springer, Cham, August 2020.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Cham, August 2019.

DHK+23. Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Generic models for group actions. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 406–435. Springer, Cham, May 2023.

DM20. Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 187–212. Springer, Cham, May 2020.

EKP20. Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 157–186. Springer, Cham, May 2020.

GHHM21. Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 637–667. Springer, Cham, December 2021.

HLM+25. Lucjan Hanzlik, Yi-Fu Lai, Marzio Mula, Eugenio Paracucchi, Daniel Slamanig, and Gang Tang. Tanuki: New frameworks for (concurrently secure) blind signatures from post-quantum group actions. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 35–69. Springer, Singapore, December 2025.

KLLQ24. Shuichi Katsumata, Yi-Fu Lai, Jason T. LeGrow, and Ling Qin. CSI-otter: isogeny-based (partially) blind signatures from the class group action with a twist. *DCC*, 92(11):3587–3643, 2024.

Kup05. Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.

MZ22. Hart Montgomery and Mark Zhandry. Full quantum equivalence of group action DLog and CDH, and more. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 3–32. Springer, Cham, December 2022.

Nec94. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Math Notes*, 55:165, 1994.

Onu21. Hiroshi Onuki. On oriented supersingular elliptic curves. *Finite Fields and Their Applications*, 69:101777, 2021.

PR23. Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766, 2023.

Res18. Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018.

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

Unr07. Dominique Unruh. Random oracles and auxiliary input. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 205–223. Springer, Berlin, Heidelberg, August 2007.

A Bit-Fixing/Auxiliary-Input ROM+GGAM

A.1 Replacing Auxiliary Information by Bit-Fixing in the ROM+GGAM

We can slightly tweak the definition of (H, τ)-source from [BL25, Definition 4] to accommodate an arbitrary injection $\tau : \mathcal{X} \to \mathcal{S}$, where $\mathcal{X}$ is a set (not necessarily a cyclic group $\mathbb{Z}_p$).

Definition 10 ([BL25, Definition 4]). *An* (H, τ)-source *is a pair of random variables* (X, Y) *where* X *corresponds to the function table of a random oracle* $\mathsf{H} : \{0,1\}^m \to \mathcal{Y}$*, with range* $\mathcal{Y}^{2^m}$*, and* Y *corresponds to the function table of an injection* $\tau : \mathcal{X} \to \mathcal{S}$*. A source* (X, Y) *is called*

- $(P_1, P_2, 1-\delta)$-dense *if for* $(P_1, P_2) \in \mathbb{Z}_{2^m} \times \mathbb{Z}_{|\mathcal{X}|}$ *if* H *is fixed on at most* P_1 *coordinates,* τ *is fixed on at most* P_2 *coordinates, and if for all pairs of subsets* $(I_1, I_2) \subseteq \{0,1\}^m \times \mathcal{X}$ *of non-fixed coordinates,*
$$H_\infty(X_{I_1}, Y_{I_2}) \geq (1-\delta)\left[|I_1| \log |\mathcal{Y}| + \log(|\mathcal{X}| - P_2)^{\underline{|I_2|}}\right],$$
where $a^{\underline{b}} \coloneqq a!/(a-b)!$ *and* X_I *is* X *restricted to the coordinates in* I*.*
- $(1-\delta)$-dense *if it is* $(0, 0, 1-\delta)$*-dense, and*
- (P_1, P_2)-Bit-Fixing *if it is fixed on at most* P_1 *coordinates on* H *and* P_2 *coordinates on* τ *and uniform over the rest.*

Then [BL25, Lemma 3] also holds with Definition 10. The only difference is that since we replaced $\mathbb{Z}_p$ (in [BL25, Definition 4]) with an arbitrary set $\mathcal{X}$, the group order $|\mathbb{Z}_p| = p$ is updated to the cardinality of the set $|\mathcal{X}|$. The proof of [BL25, Lemma 3] uses only the cardinality of the domain of τ, not any group operation in $\mathbb{Z}_p$. Hence, the same argument applies with $\mathbb{Z}_p$ replaced by the arbitrary group-action set $\mathcal{X}$.

Lemma 5 ([BL25, Lemma 3]). *Let* (X_1, X_2) *be a pair of random variables such that* X_1 *is distributed uniformly over* $\mathcal{Y}^{2^m}$ *and* X_2 *is a uniformly random injection that takes on as value function tables corresponding to an injection* $\tau : \mathcal{X} \to \mathcal{S}$*. Let* $Z \coloneqq f(X_1, X_2)$*, where* $f : \mathcal{Y}^{2^m} \times \mathcal{S}^{|\mathcal{X}|} \to \{0,1\}^S$ *is an arbitrary function. For any* $\gamma > 0$*, there exists a family* $\{(Y_1, Y_2)^{Z,\mathcal{W}}\}$*, indexed by* $Z \in \{0,1\}^S$ *and size-*$|\mathcal{X}|$ *subsets* $\mathcal{W}$ *of* $\mathcal{S}$ *of convex combinations of* (P_1, P_2)*-Bit-Fixing* (H, τ)*-sources satisfying* $P_1 \log |\mathcal{Y}| + P_2 \log(|\mathcal{X}|/e) \leq W$ *such that for any distinguisher* $\mathcal{D}$ *taking a* K*-bit input and querying at most* $T_1 < P_1$ *and* $T_2 < P_2$ *coordinates of each oracle,*

$$\left|\Pr\left[\mathcal{D}^{X_1,X_2}(f(X_1,X_2)) = 1\right] - \Pr\left[\mathcal{D}^{(Y_1,Y_2)^{f(X_1,X_2),\mathsf{im}(X_2)}}(f(X_1,X_2)) = 1\right]\right| \leq \frac{(K + \log \gamma^{-1})(T_1 \log |\mathcal{Y}| + T_2 \log |\mathcal{X}|)}{W} + \gamma,$$

and

$$\Pr\left[\mathcal{D}^{X_1,X_2}(f(X_1,X_2)) = 1\right] \leq 2^{(K+\log\gamma^{-1})(T_1 \log|\mathcal{Y}| + T_2 \log|\mathcal{X}|)/W} \cdot \Pr\left[\mathcal{D}^{(Y_1,Y_2)^{f(X_1,X_2),\mathsf{im}(X_2)}}(f(X_1,X_2)) = 1\right] + 2\gamma,$$

where e *is Euler's number.*

Proof. The proof works essentially the same as [BL25, Lemma 3]; we only replace the map $\tau : \mathbb{Z}_p \to \mathbb{G}$ with $\tau : \mathcal{X} \to \mathcal{S}$ while keeping same distributions. □

A.2 From the BF-ROM+GGAM to the AI-ROM+GGAM

Capturing the Models. An oracle pair $(\mathcal{O}_1, \mathcal{O}_2)$ has two interfaces $(\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$ and $(\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$ where $(\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$ is accessible only once before any calls to $(\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$ are made. We consider the oracles as follows for the ROM+GGAM:

- **Random Oracle + Generic Group Action Oracle** $\mathsf{RO{+}GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$**:** Samples a random oracle function table $H \leftarrow \mathcal{H}_{m,\mathcal{Y}}$, where $\mathcal{H}_{m,\mathcal{Y}}$ denotes the set of all functions from $\{0,1\}^m \to \mathcal{Y}$, and samples a random injection labeling map $\tau \leftarrow \mathcal{I}_{\mathcal{X},\rho}$, where $\mathcal{I}_{\mathcal{X},\rho}$ denotes the set of all injections from $\mathcal{X} \to \{0,1\}^\rho$ where $\rho \geq \lceil \log_2 |\mathcal{X}| \rceil$.
  - It offers no functionality at $(\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$.
  - It answers queries $x \in \{0,1\}^m$ to H at $\mathcal{O}_1.\mathsf{on}$ by the corresponding value $H(x) \in \mathcal{Y}$.
  - It answers *forward* queries $E \in \mathcal{X}$ to τ at $\mathcal{O}_2.\mathsf{on}$ by the corresponding value $\tau(E) \in \{0,1\}^\rho$.
  - It answers *group action* queries $(\mathfrak{a}, \mathcal{E})$ at $\mathcal{O}_2.\mathsf{on}$ as follows: if $\mathrm{Cl}(\mathcal{O}) = \langle [\mathfrak{g}] \rangle$ be a cyclic class group of order N, $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$, and $\mathcal{E} = \tau(E)$ for some $E \in \mathcal{X}$, the oracle replies by $\tau(\mathfrak{a} \star E)$ and by $\perp$ otherwise.
  - It answers *twist* queries $\mathcal{E}$ at $\mathcal{O}_2.\mathsf{on}$ as follows: if $\mathcal{E} = \tau(\mathfrak{a} \star E)$ for some $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$ and $E \in \mathcal{X}$, the oracle replies by $\tau(\mathfrak{a}^{-1} \star E)$, and by $\perp$ otherwise.
- **Auxiliary-Input Random Oracle + Generic Group Action Oracle** $\mathsf{AI{-}RO{+}GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$**:** Samples a random oracle function table $H \leftarrow \mathcal{H}_{m,\mathcal{Y}}$ and a random injection $\tau \leftarrow \mathcal{I}_{\mathcal{X},\rho}$ as described in $\mathsf{RO{+}GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$.
  - It outputs all of H at $\mathcal{O}_1.\mathsf{pre}$ and all of τ at $\mathcal{O}_2.\mathsf{pre}$.
  - $(\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$ behaves the same as $\mathsf{RO{+}GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$.
- **Bit-Fixing Random Oracle + Generic Group Action Oracle** $\mathsf{BF{-}RO{+}GGA}(P_1, P_2, m, \mathcal{Y}, \mathcal{X}, \rho)$**:** Samples a random oracle function table $H \leftarrow \mathcal{H}_{m,\mathcal{Y}}$ at $\mathcal{O}_1.\mathsf{pre}$ and samples a random subset $\mathcal{W}$ of $\{0,1\}^\rho$ of size $|\mathcal{X}|$ and outputs $\mathcal{W}$ at $\mathcal{O}_2.\mathsf{pre}$.
  - It takes a list at $(\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$ of at most P_1 query/answer pairs that override H in the corresponding positions and at most P_2 query/answer pairs without collisions and all answers in $\mathcal{W}$; samples a random injection $\tau \leftarrow \mathcal{I}_{\mathcal{X},\rho}$ with range $\mathcal{W}$ and consistent with said list.
  - $(\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$ behaves the same as $\mathsf{RO{+}GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$.

Attackers with Oracle-Dependent Advice. We define the attackers $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$ following the same definition of prior work [BL25], where $\mathcal{A}_{\mathsf{pre}}$ denotes a preprocessing attacker and $\mathcal{A}_{\mathsf{on}}$ denotes an online attacker that carries out the actual attack using the output of $\mathcal{A}_{\mathsf{pre}}$. For completeness, we recall the definition of an (K, T_1, T_2)-*attacker* in Definition 11 below.

Definition 11 ([BL25, Definition 5]). *An* (K, T_1, T_2)-attacker $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$ in the $(\mathcal{O}_1, \mathcal{O}_2)$-model *consists of two procedures*

- $\mathcal{A}_{\mathsf{pre}}$*, which is computationally unbounded, interacts with* $(\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$*, and outputs a* K*-bit string as hint, and*
- $\mathcal{A}_{\mathsf{on}}$*, which takes a* K*-bit auxiliary input and makes at most* T_1 *queries to* $\mathcal{O}_1.\mathsf{on}$ *and* T_2 *queries to* $\mathcal{O}_2.\mathsf{on}$*.*

Prior work [BL25] defined an *application* G in the $(\mathcal{O}_1, \mathcal{O}_2)$-model by specifying a challenger C that has access to $(\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$, interacts with the online attacker $\mathcal{A}_{\mathsf{on}}$ of $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$, and

outputs a bit at the end of the interaction. Then the *success* of G in the $(\mathcal{O}_1, \mathcal{O}_2)$-model is defined by

$$\mathrm{Succ}_{\mathsf{G},(\mathcal{O}_1,\mathcal{O}_2)}(\mathcal{A}) := \Pr\left[\mathcal{A}_{\mathsf{on}}^{(\mathcal{O}_1.\mathsf{on},\mathcal{O}_2.\mathsf{on})}\left(\mathcal{A}_{\mathsf{pre}}^{(\mathcal{O}_1.\mathsf{pre},\mathcal{O}_2.\mathsf{pre})}\right) \leftrightarrow \mathsf{C}^{(\mathcal{O}_1.\mathsf{on},\mathcal{O}_2.\mathsf{on})} = 1\right],$$

where $\mathcal{A}_{\mathsf{on}}^{(\mathcal{O}_1.\mathsf{on},\mathcal{O}_2.\mathsf{on})}\left(\mathcal{A}_{\mathsf{pre}}^{(\mathcal{O}_1.\mathsf{pre},\mathcal{O}_2.\mathsf{pre})}\right) \leftrightarrow \mathsf{C}^{(\mathcal{O}_1.\mathsf{on},\mathcal{O}_2.\mathsf{on})}$ denotes the bit output by the challenger C after interacting with $\mathcal{A}$.

As defined above, for an arbitrary application, the attacker's goal is to win a distinguishing game by outputting a single bit. Prior work [CDGS18, CDG18, BL25] also considered an *unpredictability* application, where the attacker must output a particular output (e.g., forged signature, hash collision, or preimage) to win the game.

Definition 12 ([BL25, Definition 6]). *For an* indistinguishability *application* G *in the* $(\mathcal{O}_1, \mathcal{O}_2)$*-model, the* advantage *of an attacker* $\mathcal{A}$ *is defined as*

$$\mathrm{Adv}_{\mathsf{G},(\mathcal{O}_1,\mathcal{O}_2)}(\mathcal{A}) := 2\left|\mathrm{Succ}_{\mathsf{G},(\mathcal{O}_1,\mathcal{O}_2)}(\mathcal{A}) - \frac{1}{2}\right|.$$

In the case of an unpredictability *application* G*, the advantage is defined as* $\mathrm{Adv}_{\mathsf{G},(\mathcal{O}_1,\mathcal{O}_2)}(\mathcal{A}) := \mathrm{Succ}_{\mathsf{G},(\mathcal{O}_1,\mathcal{O}_2)}(\mathcal{A})$*. An application* G *is said to be* $(K, T_1, T_2, \varepsilon)$-secure *in the* $(\mathcal{O}_1, \mathcal{O}_2)$*-model if for every* (K, T_1, T_2)*-attacker* $\mathcal{A}$*,*

$$\mathrm{Adv}_{\mathsf{G},(\mathcal{O}_1,\mathcal{O}_2)}(\mathcal{A}) \leq \varepsilon.$$

Additive Error for Arbitrary Applications in the ROM+GGAM. In the following theorem, we show that upper bounds for an arbitrary application in Bit-fixing model can be translated to an additive bound in the Auxiliary input model. This result is merely of an interest to see if theorems in previous papers like [BL25] extend to the GGAM.

Theorem 10. *For every* $\gamma > 0$*, if an application* G *is* $(K, T_1, T_2, \varepsilon')$*-secure in* $\mathsf{BF\text{-}RO\text{+}GGA}(P_1, P_2, m, \mathcal{Y}, \mathcal{X}, \rho)$ *for any* $P_1, P_2 \in \mathbb{N}$ *such that* $P_1 \log|\mathcal{Y}| + P_2 \log(|\mathcal{X}|/e) \leq \eta$*, then it is* $(K, T_1, T_2, \varepsilon)$*-secure in* $\mathsf{AI\text{-}RO\text{+}GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$*, for*

$$\varepsilon \leq \varepsilon' + \frac{2(K + \log\gamma^{-1})\left((T_1)_G^{\mathsf{comb}} \log|\mathcal{Y}| + 2(T_2)_G^{\mathsf{comb}} \log|\mathcal{X}|\right)}{\eta} + 2\gamma,$$

where e *is Euler's constant, and* $(T_1)_G^{\mathsf{comb}}, (T_2)_G^{\mathsf{comb}}$ *are the combined query complexity corresponding to* G *across the random oracle and the generic group action oracle.*

Proof. The proof follows essentially the same logic as [BL25, Theorem 5]. Fix P_1, P_2 as well as γ. Let G be an arbitrary application in the $(\mathsf{BF\text{-}RO}, \mathsf{BF\text{-}GGA})$-model (or in the $(\mathsf{AI\text{-}RO}, \mathsf{AI\text{-}GGA})$-model) and C be the corresponding challenger. Moreover, fix an (K, T_1, T_2)-attacker $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$, and let $\{(Y_1, Y_2)^{z,\mathcal{W}}\}$, indexed by $z \in \{0,1\}^S$ and size-$|\mathcal{X}|$ subsets $\mathcal{W} \subseteq \mathcal{S}$, be the family of distributions guaranteed to exist by Lemma 5. Consider the following (K, T_1, T_2)-attacker $\mathcal{A}' = (\mathcal{A}'_{\mathsf{pre}}, \mathcal{A}'_{\mathsf{on}})$ (expecting to interact with $\mathsf{BF\text{-}RO\text{+}GGA}$):

- $\mathcal{A}'_{\mathsf{pre}}$ obtains the set $\mathcal{W}$ from $\mathsf{BF\text{-}GGA.pre}$ and internally simulates $\mathcal{A}_{\mathsf{pre}}$ on a uniformly random input $X_1 \in \{0,1\}^m$ and a uniformly random injection $X_2 \in \mathcal{X}$ with range $\mathcal{W}$ to obtain $z \leftarrow \mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO\text{+}GGA.pre}}$. Then, it samples one of the (P_1, P_2)-bit-fixing sources (Y'_1, Y'_2) making up $(Y_1, Y_2)^{z,\mathcal{W}}$ and presets $\mathsf{BF\text{-}RO\text{+}GGA}$ to match (Y'_1, Y'_2) on the at most P_1 (resp. P_2) points where Y'_1 (resp. Y'_2) is fixed. The output of $\mathcal{A}'_{\mathsf{pre}}$ is z.

– $\mathcal{A}'_{\mathsf{on}}$ works exactly the same as $\mathcal{A}_{\mathsf{on}}$.

Let $\mathcal{D}$ be a distinguisher — making forward, group action operation, and twist queries to a (H, τ)-source — that internally runs the combination of $\mathcal{A}_{\mathsf{on}} = \mathcal{A}'_{\mathsf{on}}$ and C. It answers their queries as follows:

1. It passes forward queries to and back from its own oracle.
2. It answers a group action query $(\mathfrak{a}, \mathcal{E})$ by making backward query to its own oracle for $\mathcal{E}$, obtaining E, making a forward query $\mathfrak{a} \star E$, and passing the answer to $\mathcal{A}_{\mathsf{on}}$ or C (unless the answer to the backward query was $\perp$, in which case $\perp$ is returned).
3. It answers a twist query $\mathcal{E}$ by making backward query to obtain E, making a forward query $\mathfrak{a}^{-1} \star E$, and passing the answer to $\mathcal{A}_{\mathsf{on}}$ or C (unless the answer to the backward query was $\perp$, in which case $\perp$ is returned).

Note that $\mathcal{D}^{X_1,X_2}(f(X_1, X_2))$ is identical to

$$\mathcal{A}_{\mathsf{on}}^{\mathsf{AI\text{-}RO+GGA.on}}(\mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO+GGA.pre}}) \leftrightarrow \mathsf{C}^{\mathsf{AI\text{-}RO+GGA.on}},$$

and $\mathcal{D}^{(Y_1,Y_2)^{f(X_1,X_2),\mathsf{im}(X_2)}}(f(X_1, X_2))$ is identical to

$$\mathcal{A}_{\mathsf{on}}^{\prime\mathsf{BF\text{-}RO+GGA.on}}(\mathcal{A}_{\mathsf{pre}}^{\prime\mathsf{BF\text{-}RO+GGA.pre}}) \leftrightarrow \mathsf{C}^{\mathsf{BF\text{-}RO+GGA.on}}.$$

Furthermore, $\mathcal{D}$ is a distinguisher taking a K-bit input and making at most $(T_1)_G^{\mathsf{comb}}$ to its random oracle and $2(T_2)_G^{\mathsf{comb}}$ to its generic group action/twist oracle. Therefore, by Lemma 5,

$$\begin{aligned}\mathrm{Succ}_{G,\mathsf{AI\text{-}RO+GGA}}(\mathcal{A}) \le \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GGA}}(\mathcal{A}') \\ + \frac{(K + \log \gamma^{-1})\left((T_1)_G^{\mathsf{comb}} \log|\mathcal{Y}| + 2(T_2)_G^{\mathsf{comb}} \log|\mathcal{X}|\right)}{\eta} + \gamma,\end{aligned}$$

for any P_1, P_2 satisfying $P_1 \log|\mathcal{Y}| + P_2 \log(|\mathcal{X}|/e)$. Since there is only an additive term between the two success probabilities, the above inequality implies

$$\begin{aligned}\mathrm{Adv}_{G,\mathsf{AI\text{-}RO+GGA}}(\mathcal{A}) \le \mathrm{Adv}_{G,\mathsf{BF\text{-}RO+GGA}}(\mathcal{A}') \\ + \frac{2(K + \log \gamma^{-1})\left((T_1)_G^{\mathsf{comb}} \log|\mathcal{Y}| + 2(T_2)_G^{\mathsf{comb}} \log|\mathcal{X}|\right)}{\eta} + 2\gamma,\end{aligned}$$

for both indistinguishability and unpredictability applications. Note that the extra factor of 2 is technically only necessary for indistinguishability applications [CDGS18]. □

Multiplicative Error for Unpredictability Applications in the ROM+GGAM. Next we translate bounds for unpredictability applications (like signature forgeries) in the Bit-Fixing model to Auxiliary-Input model with constant multiplicative factor.

Reminder of Theorem 3. *For all $\gamma > 0$, if an* unpredictability *application $\mathcal{G}$ is $(K, T_1, T_2, \varepsilon')$-secure in $\mathsf{BF\text{-}RO+GGA}(P_1, P_2, m, \mathcal{Y}, \mathcal{X}, \rho)$ for every $P_1, P_2 \in \mathbb{N}$ satisfying*

$$(K + \log \gamma^{-1}) \cdot \left((T_1)_{\mathcal{G}}^{\mathsf{comb}} \log|\mathcal{Y}| + 2(T_2)_{\mathcal{G}}^{\mathsf{comb}} \log|\mathcal{X}|\right) \le P_1 \log|\mathcal{Y}| + P_2 \log(|\mathcal{X}|/e),$$

then it is $(K, T_1, T_2, \varepsilon)$-secure in $\mathsf{AI\text{-}RO+GGA}(m, \mathcal{Y}, \mathcal{X}, \rho)$, for

$$\varepsilon \le 2\varepsilon' + 2\gamma,$$

where e is Euler's constant.

Proof. Using the same attacker $\mathcal{A}'$ as in the proof of Theorem 10 and applying the second part of Lemma 5, for any $P_1, P_2 \in \mathbb{N}$ such that

$$\left(K + \log \gamma^{-1}\right)\left((T_1)_G^{\mathsf{comb}} \log |\mathcal{Y}| + 2(T_2)_G^{\mathsf{comb}} \log |\mathcal{X}|\right) \leq P_1 \log |\mathcal{Y}| + P_2 \log(|\mathcal{X}|/e) \leq \eta,$$

we have

$$\begin{aligned}\mathrm{Succ}_{G,\mathsf{AI\text{-}RO+GGA}}(\mathcal{A}) &\leq 2^{\frac{(K+\log\gamma^{-1})\left((T_1)_G^{\mathsf{comb}} \log |\mathcal{Y}| + 2(T_2)_G^{\mathsf{comb}} \log |\mathcal{X}|\right)}{\eta}} \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GGA}}(\mathcal{A}') + 2\gamma \\ &\leq 2 \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GGA}}(\mathcal{A}') + 2\gamma,\end{aligned}$$

which translates into

$$\mathrm{Adv}_{G,\mathsf{AI\text{-}RO+GGA}}(\mathcal{A}) \leq 2 \cdot \mathrm{Adv}_{G,\mathsf{BF\text{-}RO+GGA}}(\mathcal{A}') + 2\gamma$$

for unpredictability applications. □

B Preprocessing Security of Nonzero CSI-FiSh

Let $\mathcal{C}_{\mathsf{NZ}}^t := \mathcal{C}^t \setminus \{\mathbf{0}\}$. The *nonzero CSI-FiSh*, denoted by $\Pi_{S,t}^{\text{NZ-CSI}}$, is obtained from $\Pi_{S,t}^{\text{CSI-FiSh}}$ by sampling every Fiat-Shamir challenge uniformly from $\mathcal{C}_{\mathsf{NZ}}^t$. In this section, we analyze the preprocessing security of an uncompressed variant, as in Section 4.3.

Tracking Preprocessing-Fixed RO Points. We use the BF group action bridge-finding game of Figure 5, with the following additional preprocessing initialization. For every $(w, \mathbf{c}) \in \mathcal{F}_{\mathsf{H}}$, parse $w = \mathcal{E}_1 \| \ldots \| \mathcal{E}_t \| m$ whenever possible. Since $\mathbf{c} \in \mathcal{C}_{\mathsf{NZ}}^t$, choose $\ell = \min\{j \in [t] : c_j \neq 0\}$. If $\mathcal{E}_\ell \in \mathrm{Im}(\tau)$, let $b \leftarrow \mathtt{GAI}(\mathcal{E}_\ell)$ and record $(\mathcal{E}_\ell, \mathbf{0}, b) \in \mathcal{L}$. This adds at most P_1 tuples with zero coefficient vector to the at most P_2 preprocessing tuples already present in Figure 5. Therefore, the proof of Theorem 4 applies verbatim with P_2 replaced by $P_1 + P_2$.

Multi-User Security in the BF-(ROM+GGAM).

Theorem 11. *Let* $\mathcal{A} = \left(\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}, \mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}\right)$ *be a bit-fixing adversary, where* $\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}$ *fixes at most* P_1 *input/output pairs of the random oracle* H *and at most* P_2 *input/output pairs of the labeling map* τ*. Suppose that* $\mathcal{A}_{\mathsf{on}} := \mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GGA}(P_1,P_2)}$ *makes at most* $q_{\mathsf{G}}^{\mathsf{on}}$ *generic group action queries,* $q_{\mathsf{H}}^{\mathsf{on}}$ *random oracle queries, and* $q_{\mathsf{S}}^{\mathsf{on}}$ *signing queries. Then*

$$\Pr\left[\mathsf{SigForgeCor}_{\mathcal{A}_{\mathsf{on}},\Pi_{S,t}^{\text{NZ-CSI}}}^{T,\mathsf{H},\mathcal{O}}(\lambda) = 1\right] \leq \varepsilon,$$

where

$$\begin{aligned}\varepsilon &= \varepsilon_{\mathsf{Br}}(P_1 + P_2, \widehat{q}_{\mathsf{G}}^{\mathsf{on}}) + \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + P_1)}{N} + \frac{q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + P_1}{N - (M + P_1 + P_2 + 2\widehat{q}_{\mathsf{G}}^{\mathsf{on}} + 1)} + \frac{q_{\mathsf{H}}^{\mathsf{on}} \kappa_{\mathsf{NZ}} + 2}{(2S-1)^t - 1} \\ &= O\left(\frac{M(M + P_1 + P_2)}{N} + \frac{tq^{\mathsf{on}}(tq^{\mathsf{on}} + M + P_1 + P_2)}{N} + \frac{q^{\mathsf{on}}\kappa_{\mathsf{NZ}}}{2^\lambda}\right),\end{aligned}$$

with $N = |\mathrm{Cl}(\mathfrak{O})|$*,* $M = T(S-1)$*,* $\kappa_{\mathsf{NZ}} := \min\{T, t\}$*,* $\widehat{q}_{\mathsf{G}}^{\mathsf{on}} := q_{\mathsf{G}}^{\mathsf{on}} + t(q_{\mathsf{S}}^{\mathsf{on}} + 1) + \min\{M, t(q_{\mathsf{S}}^{\mathsf{on}} + 1)\}$*, and* $q^{\mathsf{on}} = q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{G}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}$ *denotes the online query budget of the signature attacker.*

Proof. We follow the reduction of Theorem 2, with the random oracle now taking values uniformly in $\mathcal{C}_{\mathsf{NZ}}^t$ and with Hresp initialized by $\mathcal{F}_{\mathsf{H}}$. The preprocessing-fixed RO entries are tracked as described above, so the bridge bound of Theorem 4 applies with P_2 replaced by $P_1 + P_2$. Note that the group action query complexity is unchanged and is bounded by $\widehat{q}_{\mathsf{G}}^{\mathsf{on}}$. We highlight the following differences from the analysis of Theorem 5:

1. *Fixed RO Entry.* In nonzero variant, we do not have the bad event KeyHit as appeared in Theorem 5. Instead, suppose that the forgery hash input w^* belongs to $\mathcal{F}_{\mathsf{H}}$. Let ℓ be the nonzero coordinate selected during preprocessing. The tuple recorded for $\tau(C_\ell^*)$ has coefficient vector $\mathbf{0}$. On the other hand, validity of a forgery under the uncorrupted target i^* reconstructs the same encoding from $E_{c_\ell^*}^{(i^*)}$, producing coefficient vector $\mathrm{sign}(c_\ell^*)\hat{u}_{i^*,|c_\ell^*|} \neq \mathbf{0}$. Thus, the final verification produces a bridge.
2. *Online RO Query.* Consider a hash point first defined by an online RO query. The candidate set argument of Theorem 2 applies, except that the all-zero candidate corresponding to an unrepresented target is no longer in the challenge space. Hence, $|\mathsf{Cand}(w) \cap \mathcal{C}_{\mathsf{NZ}}^t| \leq \min\{T, t\} = \kappa_{\mathsf{NZ}}$. Since $|\mathcal{C}_{\mathsf{NZ}}^t| = (2S-1)^t - 1$, we have $\Pr[\mathsf{BadQuery}] \leq \frac{q_{\mathsf{H}}^{\mathsf{on}} \kappa_{\mathsf{NZ}}}{(2S-1)^t - 1}$.
3. *Hash Entry from Signing Query.* Finally, suppose w^* was first programmed by a signing query for another user. Since every challenge vector in $\mathcal{C}_{\mathsf{NZ}}^t$ has a nonzero coordinate, the bridge extraction in case (iii) of Theorem 2 always applies. Thus, we do not have the event $\mathsf{ZeroChal}$.

Consequently, we have

$$\mathsf{Forge} \Rightarrow \mathsf{BRIDGE} \vee \mathsf{FailToSign} \vee \mathsf{FailToFind} \vee \mathsf{MissQuery} \vee \mathsf{BadQuery}.$$

The remaining bad event bounds are the same as in Theorem 2, except that the P_1 preprocessing-fixed RO points must be included among already defined hash inputs and the list $\mathcal{L}$ contains at most $P_1 + P_2$ additional preprocessing tuples. Therefore,

$$\Pr[\mathsf{FailToSign}] \leq \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + P_1)}{N}$$

and

$$\Pr[\mathsf{FailToFind}] \leq \frac{q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + P_1}{N - (M + P_1 + P_2 + 2\widehat{q}_{\mathsf{G}}^{\mathsf{on}} + 1)} + \frac{1}{(2S-1)^t - 1},$$

while

$$\Pr[\mathsf{MissQuery}] \leq \frac{1}{(2S-1)^t - 1}.$$

Combining these bounds concludes the proof. □

Multi-User Security in the AI-(ROM+GGAM).

Theorem 12. *Let* $\mathcal{A} = \left(\mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO+GGA}}, \mathcal{A}_{\mathsf{on}}^{\mathsf{AI\text{-}RO+GGA}}\right)$ *be a pair of auxiliary-input adversaries where* $\mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO+GGA}}$ *outputs a* K*-bit hint* $\mathsf{str}_{\mathsf{H},\tau}$*. Suppose* $\mathcal{A}_{\mathsf{on}}^{\mathsf{AI\text{-}RO+GGA}}$ *takes the hint* $\mathsf{str}_{\mathsf{H},\tau}$*, and makes at most* $q_{\mathsf{G}}^{\mathsf{on}}$ *generic group action queries,* $q_{\mathsf{H}}^{\mathsf{on}}$ *random oracle queries and* $q_{\mathsf{S}}^{\mathsf{on}}$ *signing queries. Let* $\widehat{q}_{\mathsf{G}}^{\mathsf{on}} = q_{\mathsf{G}}^{\mathsf{on}} + t(q_{\mathsf{S}}^{\mathsf{on}} + 1) + \min\{M, t(q_{\mathsf{S}}^{\mathsf{on}} + 1)\}$ *and* $q^{\mathsf{on}} := q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{G}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}$ *denotes the online query budget of the signature attacker. Then we have*

$$\Pr\left[\mathsf{SigForgeCor}_{\mathcal{A}_{\mathsf{on}}^{\mathsf{AI\text{-}RO+GGA}}, \Pi_{S,t}^{\mathsf{NZ\text{-}CSI}}}^{T,\mathsf{H},\mathcal{O}}(\lambda) = 1\right] \leq 2\varepsilon + 2^{-\lambda+1},$$

where

$$\varepsilon = \varepsilon_{\mathsf{Br}}(4(K+\lambda)Q, \widehat{q}_{\mathsf{G}}^{\mathsf{on}}) + \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + 2(K+\lambda)Q)}{N} + \frac{q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + 2(K+\lambda)Q}{N - (M + 4(K+\lambda)Q + 2\widehat{q}_{\mathsf{G}}^{\mathsf{on}} + 1)} + \frac{q_{\mathsf{H}}^{\mathsf{on}}\kappa_{\mathsf{NZ}} + 2}{(2S-1)^t - 1}$$
$$= O\left(\frac{(K+\lambda)(tq^{\mathsf{on}} + M)^2}{N} + \frac{q^{\mathsf{on}} \min\{T, t\}}{2^\lambda}\right),$$

with $Q \coloneqq t(q^{\mathsf{on}} + 1) + M + S$, $N = |\mathrm{Cl}(\mathfrak{O})|$, $\kappa_{\mathsf{NZ}} \coloneqq \min\{T, t\}$, *and* $M = T(S-1)$.

Proof. The claim follows from Theorem 3 and Theorem 11 exactly as in the proof of Theorem 6 by setting $P_1 = P_2 = 2(K+\lambda)Q$, since the combined query complexities are unchanged. □