On the Preprocessing Security of Nonzero Schnorr Signatures with Adaptive Corruptions

Jeremiah Blocki[1] and Seunghoon Lee*[2]

[1] Purdue University, West Lafayette, IN, USA
jblocki@purdue.edu
[2] University of Waterloo, Waterloo, ON, Canada
seunghoon.lee@uwaterloo.ca

Abstract. Modern cryptographic protocols rely on a relatively small collection of standardized hash functions and groups, raising the possibility of *preprocessing attacks* in which a computationally unbounded offline attacker generates a bounded-size hint that accelerates a later online attack. Blocki and Lee (EUROCRYPT 2022) analyzed the preprocessing security of *key-prefixed* short Schnorr signatures, where the public key is included in the random-oracle input. Their analysis imposed an a priori restriction on the number of oracle queries made during preprocessing and did not extend to standardized implementations of Schnorr, such as ISO/IEC 14888-3, which do not use key-prefixing but instead reject signatures whose hash challenge is zero. The preprocessing security of these *nonzero Schnorr signatures* was left as an open question.

We resolve this question affirmatively in the ROM+GGM. Moreover, our analysis holds in a stronger model that permits unrestricted offline computation and oracle access, subject only to the size of the resulting hint, as well as adaptive corruption queries during the online phase. Suppressing lower-order terms, an online attacker making q total oracle queries, including q_{H} random-oracle queries, forges a nonzero Schnorr signature for one of N users with probability at most $O\left(\sqrt{Sq^2/p} + q_{\mathsf{H}}N/2^{\lambda_{\mathsf{H}}}\right)$, where p is the group order and λ_{H} is the hash-output length.

We show that the factor-N loss is inherent by giving a matching *target-switching attack* with success probability $\Omega(q_{\mathsf{H}}N/2^{\lambda_{\mathsf{H}}})$. By contrast, key-prefixing prevents this attack and the corresponding term in the security bound is just $q_{\mathsf{H}}/2^{\lambda_{\mathsf{H}}}$. Consequently, at the same λ-bit security level, key-prefixing enables signatures that are approximately $\log N$ bits shorter than that of nonzero Schnorr.

To support these results, we extend the Bit-Fixing-to-Auxiliary-Input framework of Coretti et al. (CRYPTO/EUROCRYPT 2018) to applications involving multiple idealized primitives. We also address a subtle technical challenge introduced by rejection sampling in the standardized nonzero Schnorr signing algorithm: although long rejection sequences are exceptionally unlikely, a signing query has no deterministic worst-case bound on the number of random-oracle and generic-group queries made by the challenger.

Keywords: Preprocessing Attacks · Adaptive Corruption · Nonzero Schnorr Signatures · Key-Prefixing · Concrete Security · Bit-Fixing Model

1 Introduction

Many cryptographic protocols can be designed and analyzed from a handful of primitives. For example, the Schnorr signature scheme [Sch90] uses an elliptic curve group and a cryptographic hash function, while the simple oblivious transfer protocol [CO15] adds a block cipher. In practice, only a few instantiations of these primitives are widely used, e.g., AES, Triple DES, SHA-2/3, SHAKE, P-256/384, Curve25519/448. This raises the concern that an attacker with nation-state-level resources might attempt to execute a *preprocessing attack*, where a preprocessing attacker invests massive computational resources to generate a hint in the hope that this hint will help

* Part of this work was done while Seunghoon Lee was at Purdue University.

an online attacker quickly break cryptographic protocols that use the same instantiations of our primitives.

Blocki and Lee [BL22] analyzed the multi-user security[3] of (short) Schnorr signatures against preprocessing attacks in the Random Oracle Model (ROM) [BR93] plus the Generic Group Model (GGM) [Nec94, Sho97], where they gained *short* signatures by truncating the hash output by half. In particular, they established the multi-user security of *key-prefixed* short Schnorr signatures[4] against preprocessing attackers. However, their analysis involved a complicated compression argument and imposes a technical restriction on the power of the preproccessing attacker. In particular, they assume that preprocessing attacker can only examine the random oracle on a negligibly small fraction of inputs (e.g., $2^{3\lambda}$ out of the total possible $2^{4\lambda}$ input/output pairs). Furthermore, the multiuser signature forgery game that Blocki and Lee [BL22] analyzed did not model an online attacker who might be able to (adaptively) corrupt some of the users.

Blocki and Lee [BL22] also observed that key prefixing appears to be necessary to achieve preprocessing security. In particular, the original Schnorr signature scheme *without key-prefixing* since this scheme is trivially broken by preprocessing attacks that can find a pre-image of 0. In particular, if $\mathsf{H}(g^r \| m) = 0$ then the signature $\sigma = (r, 0)$ a universal forgery for the message m under *any* public key! A preprocessing attacker can simply find such (r, m) with $\mathsf{H}(g^r \| m) = 0$ and output (r, m) as a hint for the online signature forgery attacker.

Beyond key-prefixing, an alternative approach to protect against universal forgeries is to explicitly disallow signatures of form $\sigma = (\cdot, 0)$. While standardized implementations of Schnorr signatures (e.g., BSI-TR-03111 [fIS18] and ISO/IEC 14888-3 [fSC18]) do not adopt key-prefixing, they do explicitly disallow signatures of the form $\sigma = (\cdot, 0)$. We refer to such implementations as *nonzero Schnorr signatures* (see Definition 1 for the formal definition). Establishing the (in)security of (short) nonzero Schnorr signatures against preprocessing attacks was left as an open research direction [BL22].

This leads us to ask the following research questions:

- *Are standardized implementations of Schnorr signatures (no key-prefixing, and signatures of form $(\cdot, 0)$ explicitly disallowed) secure against preprocessing attacks?*
- *Does key-prefixing yield any provable security benefits?*
- *Can we establish multi-user security for both nonzero and key-prefixed Schnorr signatures in a stronger preprocessing model, where the online attacker may adaptively corrupt users and the preprocessing attacker is unrestricted in the number of random-oracle queries it makes before producing its hint?*

Coretti, Dodis, Guo, and Steinberger [CDGS18] introduced a comprehensive framework aimed at simplifying security analyses in the context of preprocessing in the ROM by proving improved bounds for Unruh's pre-sampling technique [Unr07]. In particular, they showed how to translate the security in the Bit-Fixing ROM (BF-ROM) [Unr07] into the Auxiliary-Input ROM (AI-ROM). BF-ROM allows the preprocessing attacker to fix P random oracle (RO) input/output pairs and the remaining outputs are selected uniformly at random. This is less realistic, but it is often easier

[3] We refer to the *1-out-of-N setting*, i.e., given N public keys, an adversary wins if it forges a valid signature under *any one* of these public keys.

[4] Key-prefixed Schnorr signatures salt the random oracle with the user's public key.

to upper-bound that attacker's advantage[5]. In contrast, AI-ROM generates an S-bit hint during the preprocessing phase after arbitrary RO access. This is a more realistic model of real-world attacks, but proving query complexity lower bounds in this model often involves notoriously challenging compression arguments. Coretti et al. [CDGS18] showed that upper bounds for the attacker's advantage in the BF-ROM translate to bounds in the AI-ROM with additive loss $O(Sq/P)$, where q denotes the number of online queries. As a concrete example, this immediately gives the familiar $O(\sqrt{Sq/2^\lambda})$-type bound for random-function inversion in the AI-ROM by setting $P = \sqrt{Sq2^\lambda}$, without requiring a direct compression argument.

Coretti et al. [CDG18] extended the BF-ROM framework [CDGS18] to encompass other idealized primitives including the GGM, Ideal Cipher Model (ICM) [Sha49], and Random Permutation Model (RPM). They proved that BF-GGM (resp. BF-ICM/BF-RPM) bounds yield AI-GGM (resp. AI-ICM/AI-RPM) bounds with a similar additive loss $O(Sq/P)$.

Technically, the results in Coretti et al. [CDGS18, CDG18] are only stated for a *single* idealized primitive, while many cryptographic constructions utilize several primitives, e.g., (nonzero) Schnorr signatures and hashed ElGamal KEMs such as PSEC-KEM utilize elliptic curve groups and cryptographic hash functions. Thus, analyzing their security against preprocessing attacks requires extending the BF-to-AI framework to settings with *multiple* idealized primitives.

This leads us to ask the additional research question:

– *Does Coretti et al.'s BF-to-AI framework extend to settings with* multiple *idealized models?*

While it is natural to expect such an extension to hold, it does not follow by simply applying the existing single-model results independently to each idealized primitive. It is therefore necessary to formally verify that the framework continues to hold in settings with multiple idealized models. In particular, the interaction between multiple idealized primitives must be incorporated into the BF-to-AI argument, and the resulting error terms must be carefully tracked.

1.1 Our Contributions

1.1.1 Supporting Machinery: Extension of BF-to-AI Framework to Multiple Idealized Models. To analyze our applications in multiple idealized models, e.g., ROM+GGM, we first verify that the results of Coretti et al. [CDGS18, CDG18] extend to settings with *multiple* idealized models. In particular, we verify that Bit-Fixing upper bounds yield upper bounds on the attacker's success rate in the Auxiliary-Input model with an additive loss $O(Sq/P)$. Here, S denotes the size of the hint generated by the offline attacker in the Auxiliary-Input model, P denotes the total number of input/output pairs fixed by our Bit-Fixing attacker across all idealized primitives, and q denotes the total number of oracle queries made by the online attacker across all idealized models.

Note that in the Auxiliary-Input model, the S-bit hint could encode information about relationships between the various ideal oracles, e.g., the hint could identify two generic group elements that hash to the same value under our random oracle. Thus, we cannot simply apply the results of Coretti et al. [CDG18] separately to each idealized primitive in a black-box manner. Although conceptually close to prior analyses, this extension is not stated in prior work [CDGS18, CDG18] and is required for our applications that combine multiple idealized primitives.

[5] For example, consider the function inversion problem where we are given a random string $y \in \{0,1\}^\lambda$ and the goal is to find an input x such that $f(x) = y$. In the BF-ROM, it is trivial to upper bound the attacker's success probability as $(P+q)/2^\lambda$ where q denotes the number of RO queries made by the online attacker.

Theorem 1 (informal). *If an application G is secure with probability ε' in the BF multiple idealized models (e.g., ROM+GGM+ICM) which fixes P coordinates of random oracles/generic group oracles/ideal ciphers in total, then it is also secure with probability ε in the corresponding AI multiple idealized model with an S-bit hint, with $\varepsilon \leq \varepsilon' + O(Sq/P)$ where q is the combined online query complexity corresponding to G across all idealized primitives used by G (see Theorem 4 and Theorem 10).*

While Theorem 1 applies more generally, Coretti et al. [CDGS18, CDG18] also established a tighter bound for *unpredictability applications* where a successful attacker must produce a particular target output (e.g., forged signature/MAC, preimage, hash collision, secret key) in contrast to a distinguishing attack where the attacker outputs a single bit. In unpredictability applications, we want to show that the attacker's success probability is at most ε for some small/negligible value ε. By contrast, in an indistinguishability game, we want to show that the attacker guesses the correct bit with probability at most $\frac{1}{2} + \varepsilon$ for some small/negligible ε. We verify that Coretti et al. [CDGS18, CDG18]'s result extends to our setting with multiple idealized models. This leads to tighter concrete bounds on the preprocessing security of short Schnorr signatures.

Theorem 2 (informal). *If an* unpredictability application *G is secure with probability ε' in the BF multiple idealized models (e.g., ROM+GGM+ICM) which fixes P coordinates of random oracles/generic group oracles/ideal ciphers in total, then it is also secure with probability ε in the corresponding AI multiple idealized model with an S-bit hint, with $\varepsilon \leq 2\varepsilon' + 2\gamma$, where γ satisfies $S + \log \gamma^{-1} \leq O(P/q)$ where q is the combined online query complexity corresponding to G across all idealized primitives used by G (see Theorem 5 and Theorem 11).*

In the formal statement of the theorem (Theorem 4, Theorem 5, Theorem 10, and Theorem 11), we establish slightly more refined bounds when we have separate upper bounds on the number of queries to each ideal oracle.

1.1.2 Multi-User Security of (Short) Nonzero Schnorr Signatures Against Preprocessing Attacks in the ROM+GGM.

We resolve the primary open question of whether standardized nonzero Schnorr signatures remain secure against preprocessing attacks. In comparison to prior work [BL22], our result is proved in a stronger multi-user unforgeability model in which the preprocessing attacker may perform an *unbounded* number of queries to both the random oracle and the generic group oracles before outputting an S-bit hint. During the online phase, the attacker may additionally make adaptive corruption queries $\texttt{Corrupt}(i)$, which reveal the secret key sk_i; as usual, the final forgery must be for an uncorrupted user. In particular, our bounds also rule out universal-forgery attacks with the same resource bounds, since a universal forgery is a special way of winning this stronger forgery game.

We first analyze the schemes in the Bit-Fixing model and then apply Theorem 2 to obtain bounds in the Auxiliary-Input model. For key-prefixed Schnorr, the combined query complexity of the security game is deterministically bounded, so the lifting theorem applies directly. For nonzero Schnorr, the rejection-sampling step in the standardized signing algorithm creates an additional technical challenge which we needed to address. Suppressing lower-order terms, our analysis gives the following bounds, where p is the order of the generic group, N is the number of users, $q_{\mathsf{H}} \gg N$ is the number of online random-oracle queries, q is the total number of online oracle queries, and λ_{H} is the output length of the random oracle:

$$\mathsf{Adv}_{\mathsf{NZ\text{-}Schnorr}} \leq O\Big(\sqrt{Sq^2/p} + q_{\mathsf{H}} N / 2^{\lambda_{\mathsf{H}}}\Big),$$

$$\mathsf{Adv}_{\mathsf{KP\text{-}Schnorr}} \le O\Big(Sq^2/p + q_{\mathsf{H}}/2^{\lambda_{\mathsf{H}}}\Big).$$

Thus, both nonzero Schnorr and key-prefixed Schnorr remain secure against preprocessing attacks, but their concrete security differs in the presence of adaptive corruptions.

Theorem 3 (informal). *Consider the multi-user chosen-message forgery game with N users and adaptive corruption queries. Any preprocessing attacker that performs arbitrary computation and oracle queries during preprocessing, outputs an S-bit hint, and makes at most q oracle queries during the online phase wins against nonzero Schnorr with probability at most $O\Big(\sqrt{Sq^2/p} + q_{\mathsf{H}}N/2^{\lambda_{\mathsf{H}}}\Big)$, in the ROM+GGM, up to lower-order terms. For key-prefixed Schnorr, the corresponding bound is $O\big(Sq^2/p + q_{\mathsf{H}}/2^{\lambda_{\mathsf{H}}}\big)$. See Theorem 9 and Theorem 13 for the formal statements.*

The factor-N difference between these bounds is inherent. We give a *target-switching attack* against nonzero Schnorr that succeeds with probability $\Omega\left(\min\left\{1, \frac{q_{\mathsf{H}}N}{2^{\lambda_{\mathsf{H}}}}\right\}\right)$. The attack uses adaptive corruptions to switch the eventual forgery target among the N public keys. It requires no preprocessing and therefore applies, a fortiori, in the preprocessing setting. This attack matches the $q_{\mathsf{H}}N/2^{\lambda_{\mathsf{H}}}$ term in our upper bound, showing that the dependence on both N and λ_{H} is tight. Key-prefixing binds each random-oracle query to a particular public key and prevents this target-switching attack; correspondingly, the security bound for key-prefixed Schnorr contains only $q_{\mathsf{H}}/2^{\lambda_{\mathsf{H}}}$.

The target-switching attack depends essentially on adaptive corruptions. If corruption queries are removed from the security game, we obtain the tighter term $q_{\mathsf{H}}/2^{\lambda_{\mathsf{H}}}$ for nonzero Schnorr as well as for key-prefixed Schnorr. Thus, without adaptive corruptions, key-prefixing does not yield the same concrete-security separation.

For the full-length challenges used by standardized nonzero Schnorr implementations, the target-switching term is not problematic. These implementations take $\lambda_{\mathsf{H}} \approx \log p$; with the usual choice $\log p \ge 2\lambda$, the term $q_{\mathsf{H}}N/2^{\lambda_{\mathsf{H}}}$ is at most $q_{\mathsf{H}}/2^{\lambda}$ for $N \le 2^{\lambda}$. Our results therefore establish λ-bit security for standardized nonzero Schnorr against preprocessing attacks—including universal-forgery attacks—when the group size is also selected to control the preprocessing term Sq^2/p. For example, taking $p \ge S2^{2\lambda}$ ensures that $\sqrt{Sq^2/p} \le q/2^{\lambda}$ for $q \le 2^{\lambda}$.

The distinction becomes significant when the challenge is shortened. To retain a target bound of order $q_{\mathsf{H}}/2^{\lambda}$ for nonzero Schnorr with adaptive corruptions, the tight target-switching term requires $\lambda_{\mathsf{H}} \ge \lambda + \lceil \log N \rceil$. Our upper bound shows that this choice is also sufficient, subject to the group-size requirement controlling the preprocessing term. For key-prefixed Schnorr, by contrast, it suffices to take $\lambda_{\mathsf{H}} = \lambda$. Consequently, at the same target security level, key-prefixing permits signatures that are approximately $\log N$ bits shorter than nonzero Schnorr in the adaptive-corruption setting.

As a concrete example, suppose that $S = 2^{\lambda/2}$ and $p \approx 2^{2.5\lambda}$. Key-prefixed Schnorr can use $\lambda_{\mathsf{H}} = \lambda$, resulting in signatures of length approximately 3.5λ bits. Nonzero Schnorr in the adaptive-corruption setting instead requires $\lambda_{\mathsf{H}} \approx \lambda + \log N$, resulting in signatures of length approximately $3.5\lambda + \log N$ bits. Both remain shorter than full-length Schnorr signatures for the same enlarged group, but key-prefixing yields the better signature-length guarantee.

For key-prefixed Schnorr, our result also strengthens the previous preprocessing bound of Blocki and Lee [BL22]. Their analysis gave a bound of $O\big(Sq^2 \log p/p + q_{\mathsf{H}}/2^{\lambda_{\mathsf{H}}}\big)$ and imposed a restriction on the preprocessing attacker's oracle access. Our bound removes the extraneous $\log p$ factor, allows unrestricted offline computation and oracle queries subject only to the S-bit hint bound, and continues to hold in the stronger multi-user security game with adaptive corruptions. In particular, the group-size requirement improves from approximately $p \approx S2^{2\lambda} \log p$ to $p \approx S2^{2\lambda}$.

2 Preliminaries

2.1 Notation and Idealized Models

Let $\mathbb{N}$ be the set of positive integers, and we define $[N] := \{1, \ldots, N\}$ for $N \in \mathbb{N}$. Throughout the paper, λ denotes the security parameter and $\mathbb{Z}_p$ denotes a cyclic group of integers modulo p. We say that $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{Z}_p^N$ is an N-dimensional vector over $\mathbb{Z}_p^N$, and for each $i \in [N]$, we define $\hat{u}_i$ to be the ith N-dimensional unit vector, i.e., the ith element of $\hat{u}_i$ is 1, and all other elements are 0 elsewhere. For $\mathbf{x} \in \mathbb{Z}_p^N$ and a subset $S \subseteq [N]$, $\mathbf{x}_S$ denotes a sub-vector of $\mathbf{x}$ of dimension $s := |S|$, i.e., if $\mathbf{x} = (x_1, \ldots, x_N)$ then $\mathbf{x}_S = (x_{i_1}, \ldots, x_{i_s})$ where $S = \{i_1, \ldots, i_s\}$ with $i_1 < \cdots < i_s$. For simplicity, all logarithms are base 2, i.e., $\log x := \log_2 x$. The notation $\leftarrow\!\!\$$ denotes uniform sampling.

The Random Oracle Model (ROM). The ROM is a type of an idealized model formalized by Bellare and Rogaway [BR93], in which we model a cryptographic hash function H as a truly random function. In particular, H can only be evaluated by *querying* an oracle that returns $\mathsf{H}(x)$ when an input x is given.

The Generic Group Model (GGM). The GGM is an idealized model proposed by Shoup [Sho97]. One key motivation of the GGM is that it can be used to analyze several computational hardness assumptions. In particular, Shoup [Sho97] proved that the discrete-log problem in a group of prime order p requires $\Omega(\sqrt{p})$ queries to solve, and the same lower bound holds for Computational Diffie-Hellman (CDH) problem and Decisional Diffie-Hellman (DDH) problem as well.

Given a cyclic group $G = \langle g \rangle$ of prime order p, a random injective encoding (labeling) map $\tau : \mathbb{Z}_p \to \mathbb{G}$, where $\mathbb{G}$ is the set of bit strings of length $\ell \geq \lceil \log p \rceil$, and we encode the discrete log of a group element instead of the group element itself for simplicity. In the GGM, the adversary is only given access to oracles $\mathtt{Mult}(\cdot, \cdot)$ and $\mathtt{Inv}(\cdot)$, which compute the group operation *indirectly* in G, as well as the encoding of the generator of G, i.e., $\mathfrak{g} = \tau(1)$. In particular, for $(\mathfrak{a}, \mathfrak{b}) \in \mathbb{G} \times \mathbb{G}$, the oracles work as $\mathtt{Mult}(\mathfrak{a}, \mathfrak{b}) = \tau(\tau^{-1}(\mathfrak{a}) + \tau^{-1}(\mathfrak{b}))$ and $\mathtt{Inv}(\mathfrak{a}) = \tau(-\tau^{-1}(\mathfrak{a}))$ if $\mathfrak{a}, \mathfrak{b} \in \tau(G)$. We can also define an oracle $\mathtt{Pow}(\mathfrak{a}, n) := \tau(n\tau^{-1}(\mathfrak{a}))$ for convenience, which can be computed by making $O(\log n)$ oracle calls to $\mathtt{Mult}$ using the standard modular exponentiation algorithm.

We remark that there are different GGMs; Schnorr and Jakobsson's model [SJ00] where the attacker is not directly given a labeling map, Maurer's model [Mau05] that uses pointers instead of random encoding of a group, and Kiltz et al.'s model [KMP16] in which the generic group oracle maintains a global counter. While those models are incomparable in general, Zhandry [Zha22] argued that Shoup's model should be preferred as Maurer's model fails to capture many generic techniques such as Merkle-Damgård transform, Feistel network, authenticated encryption, etc. Blocki and Lee [BL22] also noted that Schnorr and Jakobsson's model and Kiltz et al.'s model do not provide a natural way to analyze preprocessing security since the attacker is never given bit encodings of group elements which could be processed outside of the generic group oracles. As we focus on preprocessing attacks, we will utilize Shoup's GGM in this paper.

2.2 Nonzero and Key-Prefixed Schnorr Signatures

We consider two Schnorr variants, *nonzero* and *key-prefixed* Schnorr. In both variants, a signer samples an ephemeral exponent r, computes the commitment $I = g^r$, derives a challenge e from a random oracle, and outputs the signature $\sigma = (s, e)$, where $s = r + \mathsf{sk} \cdot e \pmod{p}$. They differ only in

the derivation and admissibility of the challenge. Nonzero Schnorr uses $e = \mathsf{H}(I\|m)$ and signatures with $e = 0$ are disallowed[6], whereas key-prefixed Schnorr uses $e = \mathsf{H}(\mathsf{pk}\|I\|m)$, and $e = 0$ is allowed. We write NZ for the *nonzero* variant and KP for the *key-prefixed* variant, and use $\mathsf{v} \in \{\mathsf{NZ}, \mathsf{KP}\}$ to denote either variant. The corresponding challenge function $\mathsf{Chal}_{\mathsf{v},\mathsf{pk}}$ and admissibility predicate $\mathsf{Adm}_{\mathsf{v}}$ are specified in Figure 1.

v	$\mathsf{Chal}_{\mathsf{v},\mathsf{pk}}(I, m)$	$\mathsf{Adm}_{\mathsf{v}}(e)$
NZ	$\mathsf{H}(I\|m)$	$e \neq 0$
KP	$\mathsf{H}(\mathsf{pk}\|I\|m)$	1

$\mathsf{Kg}(1^\lambda)$:

1 : $\mathsf{sk} \leftarrow\!\$\ \mathbb{Z}_p$
2 : $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$
3 : **return** $(\mathsf{pk}, \mathsf{sk})$

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

1 : $r \leftarrow\!\$\ \mathbb{Z}_p$
2 : $I \leftarrow g^r$
3 : $e \leftarrow \mathsf{Chal}_{\mathsf{v},\mathsf{pk}}(I, m)$
4 : **while** $\neg\mathsf{Adm}_{\mathsf{v}}(e)$ **do**
5 : Resample $r \leftarrow\!\$\ \mathbb{Z}_p$
6 : $I \leftarrow g^r$
7 : $e \leftarrow \mathsf{Chal}_{\mathsf{v},\mathsf{pk}}(I, m)$
8 : $s \leftarrow r + \mathsf{sk} \cdot e \mod p$
9 : **return** $\sigma = (s, e)$

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

1 : Parse $\sigma = (s, e)$
2 : **if** $\neg\mathsf{Adm}_{\mathsf{v}}(e)$ **then**
3 : **return** 0
4 : $R \leftarrow g^s \cdot \mathsf{pk}^{-e}$
5 : **if** $\mathsf{Chal}_{\mathsf{v},\mathsf{pk}}(R, m) = e$ **then**
6 : **return** 1
7 : **else return** 0

Figure 1. The nonzero (NZ) and key-prefixed (KP) Schnorr variants considered in this work.

Definition 1. *Given a cyclic group $G = \langle g \rangle$, a prime p, and a variant $\mathsf{v} \in \{\mathsf{NZ}, \mathsf{KP}\}$, let $\Pi_{\mathsf{v}} = (\mathsf{Kg}, \mathsf{Sign}_{\mathsf{v}}, \mathsf{Vfy}_{\mathsf{v}})$ denote the Schnorr signature scheme defined in Figure 1. We call Π_{NZ} the* nonzero Schnorr signature scheme *and Π_{KP} the* key-prefixed Schnorr signature scheme.

Short Schnorr Signatures. The term *short Schnorr* refers to shortening the challenge rather than changing the signing algorithm itself. A signature $\sigma = (s, e)$ consists of an element $s \in \mathbb{Z}_p$, requiring $\lceil \log_2 p \rceil$ bits to encode, together with a λ_{H}-bit challenge e. Schnorr originally suggested using a challenge shorter than the group-order representation, equivalently by truncating a longer hash output, while retaining the corresponding security level [Sch90]. Blocki and Lee [BL22] proved this intuition for multi-user security in the ROM+GGM. Accordingly, the signature length in our setting is $\lceil \log_2 p \rceil + \lambda_{\mathsf{H}}$ bits. In the usual parameter regime without preprocessing, $\lceil \log_2 p \rceil \approx 2\lambda$ and $\lambda_{\mathsf{H}} \approx \lambda$, yielding the familiar 3λ-bit signature. Under preprocessing attacks, however, the group order may need to be larger, so the total signature length need not be exactly 3λ bits. Throughout this work, we therefore keep the random-oracle output length λ_{H} explicit and derive the resulting signature length from our security bounds rather than fixing it a priori.

2.3 Multi-User Unforgeability

We consider both the standard multi-user forgery game and its adaptive-corruption extension [BHJ+15], which has been studied in both the ROM and the standard model [GJ18, HJK+21, HLG23]. In the

[6] We use rejection sampling to match the BSI and ISO specifications that motivate the nonzero variant. One could instead abort when $e = 0$, which occurs with probability $2^{-\lambda_{\mathsf{H}}}$, or avoid rejection entirely by setting $e = \mathsf{H}(I\|m)+1$ when $2^{\lambda_{\mathsf{H}}} < p$, which instead modifies the challenge distribution.

latter, an adversary may obtain the secret keys of adaptively chosen users while still being required to forge a signature for a user that remains uncorrupted. This strictly extends the multi-user notion considered by Blocki and Lee [BL22], since an adversary may always choose not to use the corruption oracle.

Let $\Pi = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vfy})$ be a signature scheme from Definition 1. Fix a labeling map $\tau : \mathbb{Z}_p \to \mathbb{G}$ for a generic group and a random oracle H. We define the signature forgery experiment with and without adaptive corruption in Figure 2.

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

1 : $\mathsf{C} \leftarrow \emptyset$; $\mathsf{Q}_\mathsf{S} \leftarrow \emptyset$
2 : **for** $i \in [N]$ **do**
3 : $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Kg}^\tau(1^\lambda)$
4 : $(j^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{H},\mathtt{GO},\mathtt{Signing},\mathtt{Corrupt}}(\tau(1), \mathsf{pk}_1, \ldots, \mathsf{pk}_N, p)$
5 : **if** $j^* \in [N] \setminus \mathsf{C} \wedge \mathsf{Vfy}(\mathsf{pk}_{j^*}, m^*, \sigma^*) = 1$
6 : $\wedge\, (j^*, m^*) \notin \mathsf{Q}_\mathsf{S}$ **then**
7 : **return** 1
8 : **else**
9 : **return** 0

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

1 : $\mathsf{Q}_\mathsf{S} \leftarrow \mathsf{Q}_\mathsf{S} \cup \{(i, m)\}$
2 : **return** $\mathsf{Sign}(\mathsf{sk}_i, m)$

$\mathtt{Corrupt}(i)$:

1 : **if** $i \notin [N]$ **then**
2 : **return** $\perp$
3 : $\mathsf{C} \leftarrow \mathsf{C} \cup \{i\}$
4 : **return** sk_i

Figure 2. The 1-out-of-N generic signature-forgery game with and without adaptive corruption. Omitting all gray-shaded components gives $\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi}(\lambda)$, and including all gives $\mathsf{SigForgeCor}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi}(\lambda)$.

Definition 2. *Consider the generic group model with a labeling map $\tau : \mathbb{Z}_p \to \mathbb{G}$. A signature scheme $\Pi = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vfy})$ is* $(N, q_\mathsf{H}, q_\mathtt{G}, q_\mathtt{S}, \varepsilon)$-MU-UF-CMA secure (multi-user unforgeable against chosen message attack) *if for every adversary $\mathcal{A}$ making at most q_H (resp. $q_\mathtt{G}$, $q_\mathtt{S}$) queries to the random (resp. generic group, signing) oracle, we have* $\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi}(\lambda) = 1\right] \leq \varepsilon$.

We define $(N, q_\mathsf{H}, q_\mathtt{G}, q_\mathtt{S}, \varepsilon)$-MU-UF-CMA security under adaptive corruption *analogously using* SigForgeCor*, where the adversary may additionally corrupt users and the forgery must target an uncorrupted user. That is,*

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

In both experiments, the randomness is taken over the selection of τ, the random coins of $\mathcal{A}$, the random coins of Kg*, and the selection of random oracle* H*.*

For *preprocessing security*, we consider an adversary $\mathcal{A} = (\mathcal{A}_\mathsf{pre}, \mathcal{A}_\mathsf{on})$ against either forgery experiment, where $\mathcal{A}_\mathsf{pre}$ performs an offline computation before the public keys are sampled and passes limited information to $\mathcal{A}_\mathsf{on}$. Then we upper bound the success probability of the resulting online forgery experiment. We consider both the bit-fixing and auxiliary-input formulations of this attack; their formal definitions are given in Section 3.

3 Background: Bit-Fixing/Auxiliary-Input ROM+GGM

Coretti et al. [CDGS18, CDG18] developed generic transformations from security in bit-fixing (BF) idealized models to security in the corresponding auxiliary-input (AI) models. Their results are

stated for a single idealized primitive. Since our Schnorr analysis simultaneously uses a random oracle and a generic group, we use the corresponding joint ROM+GGM extension. This does not follow by independently applying the single-model results: an auxiliary-input hint may depend jointly on the random oracle and the generic-group encoding and thereby encode correlations between them. We therefore extend the BF-to-AI argument to their joint distribution. Formal definitions and proofs are given in Appendix A.

BF- and AI-(ROM+GGM). We write $\mathsf{BF\text{-}RO\text{+}GG}(P_1, P_2, m, \lambda, p, \ell)$ for the bit-fixing model with a random oracle $\mathsf{H} : \{0,1\}^m \to \{0,1\}^\lambda$ and a generic group of order p represented by a random injection $\tau : \mathbb{Z}_p \to \{0,1\}^\ell$. The preprocessing adversary may fix at most P_1 random-oracle coordinates and P_2 coordinates of τ, and may pass an S-bit state to the online adversary; the remaining coordinates are completed uniformly at random, subject to injectivity of τ.

In the auxiliary-input model $\mathsf{AI\text{-}RO\text{+}GG}(m, \lambda, p, \ell)$, the preprocessing adversary may inspect the complete tables of H and τ, but is restricted to outputting an S-bit hint. In either model, an application is $(S, T_1, T_2, \varepsilon)$-secure if every such adversary whose online phase makes at most T_1 random-oracle queries and T_2 generic-group queries succeeds with probability at most ε. See Section A.2 for the formal definitions.

For an application G, let $(T_1)_G^{\mathsf{comb}}$ and $(T_2)_G^{\mathsf{comb}}$ denote the total numbers of random-oracle and generic-group queries, respectively, made by both the online adversary and the challenger.

Additive Lifting. The following joint ROM+GGM version of the BF-to-AI transformation applies to arbitrary applications.

Theorem 4. *For every $\gamma > 0$, if an application G is $(S, T_1, T_2, \varepsilon')$-secure in $\mathsf{BF\text{-}RO\text{+}GG}(P_1, P_2, m, \lambda, p, \ell)$ for any $P_1, P_2 \in \mathbb{N}$ such that $P_1\lambda + P_2 \log(p/e) \le \eta$, then it is $(S, T_1, T_2, \varepsilon)$-secure in $\mathsf{AI\text{-}RO\text{+}GG}(m, \lambda, p, \ell)$, for*

$$\varepsilon \le \varepsilon' + \frac{2(S + \log \gamma^{-1}) \left((T_1)_G^{\mathsf{comb}} \lambda + 3(T_2)_G^{\mathsf{comb}} \log p\right)}{\eta} + 2\gamma,$$

where e is the Euler's number.

Unpredictability Applications. For unpredictability applications, where the adversary wins by producing a particular target output (e.g., a forged signature, preimage, or collision) rather than by distinguishing a hidden bit, the multiplicative lifting of Coretti et al. [CDGS18, Theorem 6] gives a sharper bound. We use the following joint ROM+GGM extension. The proof of Theorem 5 can be found in Appendix A.

Theorem 5. *For every $\gamma > 0$, if an* unpredictability *application G is $(S, T_1, T_2, \varepsilon')$-secure in $\mathsf{BF\text{-}RO\text{+}GG}(P_1, P_2, m, \lambda, p, \ell)$ for any $P_1, P_2 \in \mathbb{N}$ satisfying $(S + \log \gamma^{-1}) \cdot \left((T_1)_G^{\mathsf{comb}} \lambda + 3(T_2)_G^{\mathsf{comb}} \log p\right) \le P_1\lambda + P_2 \log(p/e)$, then it is $(S, T_1, T_2, \varepsilon)$-secure in $\mathsf{AI\text{-}RO\text{+}GG}(m, \lambda, p, \ell)$, for $\varepsilon \le 2\varepsilon' + 2\gamma$, where e is the Euler's number.*

We also extend this transition to leverage additional idealized models (e.g., ROM+GGM+ICM). See Appendix B for details.

4 Preprocessing Security of Nonzero Schnorr Signatures

We apply the results from Section 3 (Theorem 4 and Theorem 5) to analyze the preprocessing security of Schnorr signatures utilizing multiple idealized models. While Blocki and Lee [BL22] analyzed preprocessing security of *key-prefixed* Schnorr signatures, several standardized implementations of Schnorr signatures, e.g., BSI-TR-03111 [fIS18] and ISO/IEC 14888-3 [fSC18], do not adopt key-prefixing but explicitly disallow $e = 0$ signatures. In this section, we analyze the multi-user preprocessing security of (short) nonzero Schnorr signatures. We begin with the standard multi-user forgery game, without adaptive corruptions.

4.1 Bridge Finding in the BF-(ROM+GGM)

Blocki and Lee [BL22] introduced the 1-*out-of-N generic* BRIDGE^N*-finding game* as an intermediate problem in their analysis of the multi-user security of Schnorr signatures against preprocessing attacks. Given challenge labels $\tau(x_1), \ldots, \tau(x_N)$, the challenger maintains symbolic representations $(\mathfrak{y}, \mathbf{a}, b)$ of the generic-group elements encountered during the execution. A bridge event occurs when the same label $\mathfrak{y}$ acquires two distinct symbolic representations $(\mathbf{a}, b) \neq (\mathbf{a}', b')$, yielding the nontrivial relation $(\mathbf{a} - \mathbf{a}') \cdot \mathbf{x} = b' - b$.

For our application, we formulate the game directly in the BF-(ROM+GGM). The preprocessing adversary fixes at most P_{H} random-oracle records and $P_{\mathtt{G}}$ generic-group records and may pass state z to the online adversary. The remaining random-oracle and generic-group coordinates, as well as the challenge secrets $x_1, \ldots, x_N$, are sampled only afterwards. We write $\mathcal{H}[F_{\mathsf{H}}]$ and $\mathcal{I}_{\mathcal{Y}}[F_{\mathtt{G}} \cup \{(1, \mathfrak{g})\}]$ for the corresponding conditional completions. Admissibility requires the fixed records to be distinct and consistent with the generator record. The resulting game is shown in Figure 3.

$\mathsf{BridgeChal}^{\tau,\mathsf{H},N}_{\mathcal{A}}(\lambda)$:

1 : $\mathcal{Y} \leftarrow\!\!\$\ \{Y \subseteq \mathbb{G} : |Y| = p\}$
2 : $\mathfrak{g} \leftarrow\!\!\$\ \mathcal{Y}$
3 : $(F_{\mathsf{H}}, F_{\mathtt{G}}, z) \leftarrow \mathcal{A}_{\mathsf{pre}}(\mathcal{Y}, \mathfrak{g})$
4 : **if** $(F_{\mathsf{H}}, F_{\mathtt{G}}, z)$ is not admissible **then**
5 : **return** 0
6 : $\mathsf{H} \leftarrow\!\!\$\ \mathcal{H}[F_{\mathsf{H}}]$
7 : $\tau \leftarrow\!\!\$\ \mathcal{I}_{\mathcal{Y}}[F_{\mathtt{G}} \cup \{(1, \mathfrak{g})\}]$
8 : $\mathbf{x} = (x_1, \ldots, x_N) \leftarrow\!\!\$\ \mathbb{Z}_p^N$
9 : $\mathcal{L} \leftarrow \emptyset$; $\mathsf{win} \leftarrow 0$
10 : $\mathtt{Record}(\mathfrak{g}, \mathbf{0}, 1)$
11 : **for** $i \in [N]$ **do**
12 : $\mathtt{Record}(\tau(x_i), \hat{u}_i, 0)$
13 : **for** $(t, \mathfrak{t}) \in F_{\mathtt{G}}$ **do**
14 : $\mathtt{Record}(\mathfrak{t}, \mathbf{0}, t)$
15 : $o \leftarrow \mathcal{A}^{\mathsf{H},\mathtt{GO}}_{\mathsf{on}}\big(F_{\mathsf{H}}, F_{\mathtt{G}}, z, \mathfrak{g}, \tau(x_1), \ldots, \tau(x_N)\big)$
16 : **return** win

$\mathtt{Record}(\mathfrak{y}, \mathbf{a}, b)$:

1 : **if** $\exists (\mathfrak{y}, \mathbf{a}', b') \in \mathcal{L}$ such that
2 : $(\mathbf{a}', b') \neq (\mathbf{a}, b)$ **then**
3 : $\mathsf{win} \leftarrow 1$
4 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathfrak{y}, \mathbf{a}, b)\}$

Figure 3. The 1-out-of-N generic bridge-finding game in the BF-(ROM+GGM). The terminal output o of $\mathcal{A}_{\mathsf{on}}$ is ignored. The challenger uses the internal procedure `Record` to maintain $\mathcal{L}$ and sets the sticky flag win as soon as a bridge collision occurs.

During the online phase, `Mult` and `Inv` are the standard GGM oracles from Section 2. The challenger additionally maintains the symbolic list $\mathcal{L}$ using `Record`: fresh generic-group labels are

registered with their affine representations, and multiplication and inversion propagate these representations in the natural way. The internal discrete logarithms used for this bookkeeping are never revealed to the adversary. Whenever a tuple is inserted into $\mathcal{L}$, if its label already occurs with a distinct affine representation, the challenger sets $\mathsf{win} \leftarrow 1$. Once set, this flag remains equal to 1 for the rest of the experiment.

Random oracle queries are answered by H in the usual way and do not themselves modify $\mathcal{L}$. A fixed RO input may syntactically contain one or more strings that are valid generic-group labels, and such labels may influence the online adversary's strategy. A label is nevertheless registered in $\mathcal{L}$ only when it is actually submitted to a generic-group oracle. Such registrations are already accounted for by the number of online generic-group queries. Thus, although the bridge event itself is defined through the generic group transcript, both the preprocessing state and the online strategy may depend on the fixed and completed random oracle.

Security of the Bridge-Finding Game. We next upper bound the probability of winning the bridge-finding game in the BF-(ROM+GGM). The random-oracle component does not incur an additional loss in the bridge bound. To see this, fix an admissible preprocessing view $\mathsf{view}_{\mathsf{pre}} = (\mathcal{Y}, \mathfrak{g}, F_{\mathsf{H}}, F_{\mathtt{G}}, z)$ and condition further on the completed random oracle H and on the random coins of the adversary. Under this conditioning, the fixed RO records, the RO responses, and the state z determine a possibly nonuniform and adaptive strategy for the online adversary. However, the remaining labeling map is still uniformly distributed over $\mathcal{I}_{\mathcal{Y}}[F_{\mathtt{G}} \cup \{(1, \mathfrak{g})\}]$. Thus, it suffices to analyze symbolic collisions over the remaining randomness of τ; the RO component introduces no additional constraint.

Relative to the bridge analysis of Blocki and Lee [BL22, Theorem 5], the initial symbolic list has size at most $N + P_{\mathtt{G}} + 1$. The P_{H} fixed RO records are not inserted into $\mathcal{L}$. A direct analysis based only on $|\mathcal{L}|^2$ would retain a quadratic $P_{\mathtt{G}}^2$ term in the denominator, indicating that the group size p should be at least $P_{\mathtt{G}}^2$. Considering the security loss of $O\big((S + \log \gamma^{-1}) q_{\mathtt{G}}^{\mathsf{on}})/P_{\mathtt{G}}\big)$ [CDG18, Theorem 1] that appears in the transition from the Bit-Fixing to the Auxiliary-Input model, we find that $P_{\mathtt{G}}$ must be $\Omega(2^{\lambda} S)$ to achieve λ bits of security. Consequently, $p = \Omega(P_{\mathtt{G}}^2) = \Omega(2^{2\lambda} S^2)$. Hence, we come up with a longer signature length of $\log p + \lambda \geq 3\lambda + 2 \log S$, when compared to the prior work that achieves λ bits of security with a Schnorr signature length of $3\lambda + \log S$ bits [BL22].

Consequently, a more sophisticated analysis is required to maintain the same signature length as demonstrated in [BL22] while ensuring the same security level. Recall that the bridge event occurs if we find two distinct tuples $(\mathfrak{y}_1, \mathbf{a}_1, b_1), (\mathfrak{y}_2, \mathbf{a}_2, b_2)$ such that $\mathfrak{y}_1 = \mathfrak{y}_2$ but $(\mathbf{a}_1, b_1) \neq (\mathbf{a}_2, b_2)$ which yields a non-trivial linear dependency on $\mathbf{x}$. Intuitively, we exploit the observation that if we pick two distinct tuples $(\mathfrak{y}_1, \mathbf{a}_1, b_1), (\mathfrak{y}_2, \mathbf{a}_2, b_2)$ such that $\mathbf{a}_1 = \mathbf{a}_2 = \mathbf{0}$ then we cannot produce the bridge event since otherwise we have a contradiction. Since all the pre-fixed $P_{\mathtt{G}}$ generic-group coordinates are of the form $(\cdot, \mathbf{0}, \cdot)$, we can get a tighter bound by splitting the list set $\mathcal{L}$ into two subsets $\mathcal{L}_0$ and $\mathcal{L}_1$ where $\mathcal{L}_0 \coloneqq \{(\mathfrak{y}, \mathbf{a}, b) \in \mathcal{L} : \mathbf{a} = \mathbf{0}\}$ and $\mathcal{L}_1 \coloneqq \{(\mathfrak{y}, \mathbf{a}, b) \in \mathcal{L} : \mathbf{a} \neq \mathbf{0}\}$. Two distinct tuples in $\mathcal{L}_0$ cannot create a bridge, since the injectivity of τ implies that $\tau(b) = \tau(b')$ only when $b = b'$. Hence, only pairs not entirely contained in $\mathcal{L}_0$ need to be counted. Since all fixed GGM records belong to $\mathcal{L}_0$, this eliminates the quadratic dependence on $P_{\mathtt{G}}$.

Lemma 1. *Let $p > 2^{2\lambda}$ be a prime, and $N, P_{\mathsf{H}}, P_{\mathtt{G}}$ be nonnegative integers. Let $\mathcal{A} \coloneqq \Big(\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}}, P_{\mathtt{G}})}, \mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}}, P_{\mathtt{G}})}\Big)$ be a bit-fixing adversary for the multi-user bridge-finding game in Figure 3. Suppose that $\mathcal{A}_{\mathsf{on}}$ makes at most $q_{\mathsf{H}}^{\mathsf{on}}$ random-oracle queries and at most $q_{\mathtt{G}}^{\mathsf{on}}$ generic-group queries.*

Define $D_{\mathsf{br}} \coloneqq p - 2P_{\mathtt{G}}(N + 3q_{\mathtt{G}}^{\mathsf{on}} + 1) - (N + 3q_{\mathtt{G}}^{\mathsf{on}} + 1)^2 - N$ *and assume that* $D_{\mathsf{br}} > 0$. *Then*

$$\Pr\left[\mathsf{BridgeChal}_{\mathcal{A}}^{\tau,\mathsf{H},N}(\lambda) = 1\right] \le \frac{\binom{N}{2} + N(P_{\mathtt{G}} + 1)}{p} + \frac{q_{\mathtt{G}}^{\mathsf{on}}(N + P_{\mathtt{G}}) + 3q_{\mathtt{G}}^{\mathsf{on}}(q_{\mathtt{G}}^{\mathsf{on}} + 1)/2}{D_{\mathsf{br}}}.$$

The probability is over all random choices in the experiment, including the completion of H *and* τ*, the challenge secrets, and the random coins of* $\mathcal{A}_{\mathsf{pre}}$ *and* $\mathcal{A}_{\mathsf{on}}$.

Proof sketch. Here, we only give a proof sketch and the full proof can be found in Appendix D. Let InitBr denote the event that the initial list already contains a bridge. By injectivity of τ, this can occur only if two challenge secrets coincide, or if some challenge secret coincides with the generator or with one of the preprocessing-fixed GGM coordinates. Hence, $\Pr[\mathsf{InitBr}] \le \frac{\binom{N}{2}+N(P_{\mathtt{G}}+1)}{p}$.

Condition on $\overline{\mathsf{InitBr}}$, on the preprocessing view, the completed random oracle, and the adversary's coins. Consider the ith online generic-group query, conditioned further on no bridge having occurred previously. Before the response to this query is sampled, the affine coefficients $(\mathbf{a}_i, b_i)$ of the resulting representation are already determined. If the exact tuple $(\mathbf{a}_i, b_i)$ has already been recorded, no bridge is created. Otherwise, a bridge with a previously recorded tuple $(\cdot, \mathbf{a}, b)$ requires $(\mathbf{a}-\mathbf{a}_i)\cdot\mathbf{x} = b_i - b$. If $\mathbf{a} = \mathbf{a}_i$, this equality is impossible since $b \neq b_i$. Otherwise, choose a coordinate r for which $a[r] - a_i[r] \neq 0$, and expose x_r last.

The previous bridge-free transcript imposes inequalities between the affine representations of distinct recorded labels. After fixing all coordinates except x_r, each such inequality involving x_r excludes at most one possible value of x_r. Crucially, pairs of tuples in $\mathcal{L}_0 = \{(\mathfrak{y}, \mathbf{a}, b) \in \mathcal{L} : \mathbf{a} = \mathbf{0}\}$ exclude no value at all. Thus, x_r remains uniform over a set of size at least $p - \left(|\mathcal{L}|^2 - |\mathcal{L}_0|^2\right) - (N-1)$, where the final $N-1$ accounts for the distinct challenge labels. Writing $p_{\mathtt{G}} = |F_{\mathtt{G}}| \le P_{\mathtt{G}}$, before any online query we have $|\mathcal{L}_0| \ge p_{\mathtt{G}}$, while after at most $q_{\mathtt{G}}^{\mathsf{on}}$ generic-group queries, $|\mathcal{L}| \le N + p_{\mathtt{G}} + 3q_{\mathtt{G}}^{\mathsf{on}} + 1$. Consequently, the denominator above is at least D_{br}. Hence, for each previously recorded tuple, the probability of producing the required linear relation is at most $1/D_{\mathsf{br}}$. At the ith query there are at most $N + P_{\mathtt{G}} + 3i$ relevant tuples, and therefore, $\Pr[B_i \mid \overline{\mathsf{BRIDGE}}_{<i}^{N}] \le \frac{N+P_{\mathtt{G}}+3i}{D_{\mathsf{br}}}$. A union bound over initialization and all $q_{\mathtt{G}}^{\mathsf{on}}$ online generic-group queries gives the stated bound. □

4.2 Nonzero Schnorr in the BF-(ROM+GGM)

We first analyze nonzero Schnorr signatures in the standard multi-user forgery game, without corruption. Our reduction uses the bridge-finding game from Section 4.1 to control collisions among symbolic representations. We keep the challenge length λ_{H} explicit throughout the analysis; this will allow us to derive both the security of the standard full-length variant and the parameters for shorter signatures.

Theorem 6. *Let* $p > 2^{2\lambda}$ *be prime and suppose that* $2^{\lambda_{\mathsf{H}}} < p$. *Let* $\mathcal{A}_{\mathsf{Sig}} = \left(\mathcal{A}_{\mathsf{Sig.pre}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}, \mathcal{A}_{\mathsf{Sig.on}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}\right)$ *be a bit-fixing adversary. Suppose that* $\mathcal{A}_{\mathsf{Sig.on}}$ *makes at most* $q_{\mathsf{H}}^{\mathsf{on}}$ *random-oracle queries,* $q_{\mathtt{G}}^{\mathsf{on}}$ *primitive generic-group queries (i.e.,* `Mult` *and* `Inv`*), and* $q_{\mathsf{S}}^{\mathsf{on}}$ *signing queries. Define* $\overline{q}_{\mathtt{G}}^{\mathsf{on}} \coloneqq q_{\mathtt{G}}^{\mathsf{on}} + (q_{\mathsf{S}}^{\mathsf{on}} + 1)(2\lceil \log_2 p \rceil + 2\lambda_{\mathsf{H}} + 2)$ *and* $D_{\mathsf{sig}} \coloneqq p - 2P_{\mathtt{G}}(N + 3\overline{q}_{\mathtt{G}}^{\mathsf{on}} + 1) - (N + 3\overline{q}_{\mathtt{G}}^{\mathsf{on}} + 1)^2 - N$.

Assume that $D_{\mathsf{sig}} > 0$. *Then,* $\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda)=1\right] \le \varepsilon$, *with*

$$\begin{aligned}\varepsilon &= \frac{\binom{N}{2}+N(P_{\mathtt{G}}+1)}{p} + \frac{\overline{q}^{\mathsf{on}}_{\mathtt{G}}(N+P_{\mathtt{G}})+3\overline{q}^{\mathsf{on}}_{\mathtt{G}}(\overline{q}^{\mathsf{on}}_{\mathtt{G}}+1)/2}{D_{\mathsf{sig}}} \\ &\quad + \frac{q^{\mathsf{on}}_{\mathsf{S}}(P_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{S}})}{p} + \frac{(P_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{H}})(N+\overline{q}^{\mathsf{on}}_{\mathtt{G}})}{p-(N+P_{\mathtt{G}}+3\overline{q}^{\mathsf{on}}_{\mathtt{G}}+1)} + \frac{q^{\mathsf{on}}_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{S}}+1}{2^{\lambda_{\mathsf{H}}}} \\ &= O\!\left(\frac{(N+P_{\mathsf{H}}+q\log p)(N+P_{\mathtt{G}}+q\log p)}{p} + \frac{q}{2^{\lambda_{\mathsf{H}}}}\right),\end{aligned}$$

where $q \coloneqq q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathtt{G}} + q^{\mathsf{on}}_{\mathsf{S}}$ *denotes the total online query complexity of* $\mathcal{A}_{\mathsf{sig.on}}$. *The asymptotic bound holds when* $D_{\mathsf{sig}} = \Omega(p)$. *The probability is over the ideal oracles and all random coins of the experiment and the adversary.*

Proof. Given a bit-fixing adversary $\mathcal{A}_{\mathsf{Sig}} = \left(\mathcal{A}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}_{\mathsf{Sig.pre}}, \mathcal{A}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}_{\mathsf{Sig.on}}\right)$ that attacks the nonzero Schnorr signature scheme, we construct an efficient generic algorithm $\mathcal{B} = \left(\mathcal{B}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}_{\mathsf{pre}}, \mathcal{B}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}_{\mathsf{on}}\right)$ which tries to win $\mathsf{BridgeChal}^{\tau,\mathsf{H},N}_{\mathcal{B}}(\lambda)$:

Algorithm $\mathcal{B} = \left(\mathcal{B}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}_{\mathsf{pre}}, \mathcal{B}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}_{\mathsf{on}}\right)$**:**

Preprocessing Phase:

1. On input $(\mathcal{Y},\mathfrak{g})$, algorithm $\mathcal{B}_{\mathsf{pre}}$ runs $(F_{\mathsf{H}},F_{\mathtt{G}},z) \leftarrow \mathcal{A}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathtt{G}})}_{\mathsf{Sig.pre}}(\mathcal{Y},\mathfrak{g})$.
2. $\mathcal{B}_{\mathsf{pre}}$ outputs the same tuple $(F_{\mathsf{H}},F_{\mathtt{G}},z)$. In particular, $\mathcal{B}_{\mathsf{pre}}$ does not introduce any additional fixed coordinate of τ.

Online Phase:

Algorithm $\mathcal{B}_{\mathsf{on}}$ receives $(F_{\mathsf{H}},F_{\mathtt{G}},z)$ as well as $\mathfrak{g} = \tau(1)$, $\mathsf{pk}_i = \tau(x_i)$ for $1 \le i \le N$ from the bridge-finding experiment. It has access to the standard generic-group oracles $\mathtt{GO} = (\mathtt{Mult},\mathtt{Inv})$ of Figure 3.

3. Initialize $\mathcal{T}_{\mathsf{H}} \leftarrow F_{\mathsf{H}}, \mathsf{Q}_{\mathsf{S}} \leftarrow \emptyset$. Here, $\mathcal{T}_{\mathsf{H}}$ is the table of an internally simulated programmable random oracle $\widetilde{\mathsf{H}}$.
4. The oracle $\widetilde{\mathsf{H}}$ is simulated as follows. On input w:
   (a) If $(w,R) \in \mathcal{T}_{\mathsf{H}}$, return R.
   (b) Otherwise, sample $R \leftarrow\!\!\$\ \{0,1\}^{\lambda_{\mathsf{H}}}$, update $\mathcal{T}_{\mathsf{H}} \leftarrow \mathcal{T}_{\mathsf{H}} \cup \{(w,R)\}$, and return R.
   The resulting oracle is distributed identically to a uniformly random oracle conditioned on the fixed records F_{H}. $\mathcal{B}_{\mathsf{on}}$ does not need to use its direct access to the random oracle sampled by the bridge experiment.
5. Algorithm $\mathcal{B}_{\mathsf{on}}$ invokes $\mathcal{A}^{\widetilde{\mathsf{H}},\mathtt{GO},\mathtt{Signing}}_{\mathsf{Sig.on}}(F_{\mathsf{H}},F_{\mathtt{G}},z,\mathfrak{g},\mathsf{pk}_1,\ldots,\mathsf{pk}_N,p)$. Generic-group queries are forwarded *unchanged* to the bridge challenger. The challenger maintains the symbolic list $\mathcal{L}$ and the sticky flag win as specified in Figure 3.
6. When $\mathcal{A}_{\mathsf{Sig.on}}$ submits a signing query $\mathtt{Signing}(j,m)$:
   (a) Update $\mathsf{Q}_{\mathsf{S}} \leftarrow \mathsf{Q}_{\mathsf{S}} \cup \{(j,m)\}$.
   (b) Sample $s \leftarrow\!\!\$\ \mathbb{Z}_p$ and $e \leftarrow\!\!\$\ \{0,1\}^{\lambda_{\mathsf{H}}}$.
   (c) If $e = 0$, abort the simulation and return $\perp$.

(d) Compute $\mathfrak{s} \leftarrow \mathtt{Pow}(\mathfrak{g}, s)$, $\mathfrak{e} \leftarrow \mathtt{Pow}(\mathsf{pk}_j, e)$, and $I \leftarrow \mathtt{Mult}(\mathfrak{s}, \mathtt{Inv}(\mathfrak{e})) = \tau(s - ex_j)$.
(e) If $(I\|m, R) \in \mathcal{T}_{\mathsf{H}}$ for some R, abort the simulation and return $\perp$.
(f) Otherwise, program $\mathcal{T}_{\mathsf{H}} \leftarrow \mathcal{T}_{\mathsf{H}} \cup \{(I\|m, e)\}$ and return $\sigma = (s, e)$.

7. Suppose that $\mathcal{A}_{\mathsf{Sig.on}}$ terminates with (j^*, m^*, σ^*). If $j^* \notin [N]$ or $(j^*, m^*) \in \mathsf{Q}_{\mathsf{S}}$, then return $\perp$. Parse $\sigma^* = (s^*, e^*)$. If $e^* = 0$, then return $\perp$.
8. Compute $\mathfrak{s}^* \leftarrow \mathtt{Pow}(\mathfrak{g}, s^*)$, $\mathfrak{e}^* \leftarrow \mathtt{Pow}(\mathsf{pk}_{j^*}, e^*)$, and then compute $I^* \leftarrow \mathtt{Mult}(\mathfrak{s}^*, \mathtt{Inv}(\mathfrak{e}^*)) = \tau(s^* - e^* x_{j^*})$. By the symbolic bookkeeping of the bridge challenger, this computation records $(I^*, -e^*\hat{u}_{j^*}, s^*)$ in $\mathcal{L}$. If $\widetilde{\mathsf{H}}(I^*\|m^*) \neq e^*$, then return $\perp$. If I^* already has a distinct symbolic representation, then `Record` sets the sticky flag $\mathsf{win} \leftarrow 1$.

We remark that the random oracle $\widetilde{\mathsf{H}}$ presented to $\mathcal{A}_{\mathsf{Sig.on}}$ is an internal lazy-sampled completion consistent with F_{H}. It is therefore distributed identically to the random oracle in the bit-fixing signature experiment. The random oracle sampled by the bridge-finding challenger can be ignored by $\mathcal{B}_{\mathsf{on}}$; it is included in the bridge game so that the preprocessing interface and the conditioning argument are stated in the same multiple idealized model. All generic-group queries made by $\mathcal{A}_{\mathsf{Sig.on}}$ are forwarded unchanged to the bridge challenger.

Query Accounting. Observe that $\mathtt{Pow}(\mathfrak{g}, s)$ uses at most $2\lceil \log_2 p \rceil$ primitive generic-group queries as $s \in \mathbb{Z}_p$, while $\mathtt{Pow}(\mathsf{pk}_j, e)$ uses at most $2\lambda_{\mathsf{H}}$ primitive queries as $1 \leq e < 2^{\lambda_{\mathsf{H}}}$. The final calls to `Inv` and `Mult` use two additional queries. Hence, each computation $I = \mathtt{Mult}(\mathtt{Pow}(\mathfrak{g}, s), \mathtt{Inv}(\mathtt{Pow}(\mathsf{pk}_j, e)))$ uses at most $2\lceil \log_2 p \rceil + 2\lambda_{\mathsf{H}} + 2$ primitive generic-group queries. Algorithm $\mathcal{B}_{\mathsf{on}}$ performs at most one such computation for each of the at most $q_{\mathsf{S}}^{\mathsf{on}}$ signing queries and one additional computation while processing the final forgery. Therefore, including the $q_{\mathsf{G}}^{\mathsf{on}}$ generic-group queries forwarded from $\mathcal{A}_{\mathsf{Sig.on}}$, the total number of primitive generic-group queries made in the bridge-finding experiment is at most $\overline{q}_{\mathsf{G}}^{\mathsf{on}} = q_{\mathsf{G}}^{\mathsf{on}} + (q_{\mathsf{S}}^{\mathsf{on}} + 1)(2\lceil \log_2 p \rceil + 2\lambda_{\mathsf{H}} + 2)$.

Analysis. Let Forge denote the event that $\mathcal{A}_{\mathsf{Sig.on}}$ outputs a valid forgery for some user j^* in the simulated signature forgery experiment. We first define the failure events that can occur in the reduction. The event $\mathsf{FailtoSign}$ occurs if, while answering a signing query, $\mathcal{B}_{\mathsf{on}}$ either samples $e = 0$ or attempts to program a random oracle coordinate that has already been defined in the RO table $\mathcal{T}_{\mathsf{H}}$.

Consider an execution in which $\mathsf{Forge} \wedge \overline{\mathsf{FailtoSign}}$ occurs. Write the forgery output by $\mathcal{A}_{\mathsf{Sig.on}}$ as (j^*, m^*, σ^*), parse $\sigma^* = (s^*, e^*)$, and let $I^* \coloneqq \tau(s^* - e^* x_{j^*})$. Thus, $e^* = \widetilde{\mathsf{H}}(I^*\|m^*)$ and the final generic-group computation performed by $\mathcal{B}_{\mathsf{on}}$ causes the bridge challenger to call $\mathtt{Record}(I^*, -e^*\hat{u}_{j^*}, s^*)$. We consider the following times:

- Let t_{H}^* denote the first time, before the forgery output, when the RO coordinate $I^*\|m^*$ is defined in the table $\mathcal{T}_{\mathsf{H}}$. We set $t_{\mathsf{H}}^* \coloneqq 0$ if $I^*\|m^*$ already belongs to the preprocessing table F_{H}, and set $t_{\mathsf{H}}^* \coloneqq \infty$ if it is not defined before $\mathcal{A}_{\mathsf{Sig.on}}$ outputs the forgery.
- Let t_{nc}^* denote the first time that I^* is recorded in $\mathcal{L}$ with a *nonzero coefficient* vector, including the final computation used for forgery verification. We observe that, since $e^* \neq 0$, this final verification records $(I^*, -e^*\hat{u}_{j^*}, s^*)$ to $\mathcal{L}$, so we always have $t_{\mathsf{nc}}^* < \infty$.

Now we define the remaining failure events as follows, conditioned on $\mathsf{Forge} \wedge \overline{\mathsf{FailtoSign}}$ as discussed. Also, assume that no bridge occurs.

(1) $[t^*_{\mathsf{H}} = \infty]$, i.e., the critical coordinate $I^*\|m^*$ has *not* been defined by the time $\mathcal{A}_{\mathsf{Sig.on}}$ outputs its purported forgery. Thus, the adversary chooses e^* without having observed $\widetilde{\mathsf{H}}(I^*\|m^*)$, so no bridge event occurs. We call this event NoQuery.

(2) $[t^*_{\mathsf{H}} < \infty]$ but $[t^*_{\mathsf{H}} < t^*_{\mathsf{nc}}]$, i.e., the critical RO coordinate $I^*\|m^*$ is defined *before* I^* acquires a symbolic representation with nonzero coefficient vector in $\mathcal{L}$. We call this event EarlyRO.

(3) Finally, $[t^*_H < \infty]$ and $[t^*_{\mathsf{nc}} \le t^*_{\mathsf{H}}]$, but when the reduction makes the finall call $\mathtt{Record}(I^*, -e^*\hat{u}_{j^*}, s^*)$, the list $\mathcal{L}$ *already* contains the exact tuple $(I^*, -e^*\hat{u}_{j^*}, s^*)$, so no bridge event occurs. We call this event BadQuery.

Then we observe the implication

$$\left\{\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi^{\mathsf{short}}_{\mathsf{NZ}}}(\lambda)=1\right\} \subseteq \mathsf{FailtoSign} \cup \left\{\mathsf{BridgeChal}^{\tau,\mathsf{H},N}_{\mathcal{B}}(\lambda)=1\right\} \cup \mathsf{NoQuery} \cup \mathsf{EarlyRO} \cup \mathsf{BadQuery}. \tag{1}$$

We next upper bound the probability of each bad event as follows.

(1) Bounding FailtoSign. Consider the kth signing query, conditioned on no previous FailtoSign. The sampled challenge $e_k \leftarrow\$ \{0,1\}^{\lambda_{\mathsf{H}}}$ is zero with probability $2^{-\lambda_{\mathsf{H}}}$. Otherwise, if $e_k \neq 0$, then conditioned on the preceding transcript, the exponent $s_k - e_k x_j$ is uniform in $\mathbb{Z}_p$, and hence $I_k = \tau(s_k - e_k x_j)$ is uniform over the p generic group labels. Since at most $P_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{H}} + k - 1$ RO coordinates have already been defined via preprocessing, online RO queries, and preceding signing queries, union bounding over the signing queries gives $\Pr[\mathsf{FailtoSign}] \le \sum_{k=1}^{q^{\mathsf{on}}_{\mathsf{S}}} \left(\frac{1}{2^{\lambda_{\mathsf{H}}}} + \frac{P_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{H}}+k-1}{p}\right) \le \frac{q^{\mathsf{on}}_{\mathsf{S}}}{2^{\lambda_{\mathsf{H}}}} + \frac{q^{\mathsf{on}}_{\mathsf{S}}(P_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{S}})}{p}$.

(2) Bounding NoQuery. If NoQuery occurs, then e^* is fixed before $\widetilde{\mathsf{H}}(I^*\|m^*)$ is sampled, which is fresh and uniform, so we have $\Pr[\mathsf{NoQuery}] \le \frac{1}{2^{\lambda_{\mathsf{H}}}}$.

(3) Bounding EarlyRO. At any time, there are at most $P_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{H}}$ labels I may have appeared in an RO input before being recorded in $\mathcal{L}$ with a nonzero coefficient vector, since there are at most P_{H} preprocessing inputs and $q^{\mathsf{on}}_{\mathsf{H}}$ online RO queries. Points programmed during signing simulation are excluded, since the corresponding label I is recorded in $\mathcal{L}$ before $\widetilde{\mathsf{H}}(I\|m)$ is programmed.

There are at most $N + \overline{q}^{\mathsf{on}}_{\mathsf{G}}$ opportunities for a label to acquire a nonzero coefficient vector for the first time, since there are N initial public key labels and at most one output per generic group query. If the underlying group element has not appeared previously in $\mathcal{L}$, then conditioned on the transcript so far, its label is uniform among the unused labels. Moreover, $|\mathcal{L}| \le N + P_{\mathtt{G}} + 3\overline{q}^{\mathsf{on}}_{\mathsf{G}} + 1$ since $\mathcal{L}$ initially contains the generator, the N public keys, and the $P_{\mathtt{G}}$ preprocessing labels, while each generic group query can register at most two fresh inputs and one output. Hence, each such registration hits one of the at most $P_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{H}}$ labels exposed earlier through the RO with probability at most $(P_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{H}})/(p - (N + P_{\mathtt{G}} + 3\overline{q}^{\mathsf{on}}_{\mathsf{G}} + 1))$. Union bounding over the $N + \overline{q}^{\mathsf{on}}_{\mathsf{G}}$ possible registration, we have $\Pr[\mathsf{EarlyRO}] \le \frac{(P_{\mathsf{H}}+q^{\mathsf{on}}_{\mathsf{H}})(N+\overline{q}^{\mathsf{on}}_{\mathsf{G}})}{p-(N+P_{\mathtt{G}}+3\overline{q}^{\mathsf{on}}_{\mathsf{G}}+1)}$.

(4) Bounding BadQuery. On BadQuery, the coordinate $I^*\|m^*$ must be first defined by an online RO query of $\mathcal{A}_{\mathsf{Sig.on}}$. Indeed, it cannot belong to F_{H}, since then $t^*_{\mathsf{H}} = 0 < t^*_{\mathsf{nc}}$. It also cannot first be programmed by signing oracle. To see this, suppose that it was programmed during a signing query (j, m), producing (s, e) and $I = \tau(s - ex_j)$. Then $I\|m = I^*\|m^*$, so $m = m^*$ and $e = e^*$. The simulator records $(I, -e\hat{u}_j, s)$ before programming the RO, whereas processing the forgery records $(I^*, -e^*\hat{u}_{j^*}, s^*)$. Since we condition on no bridge, these symbolic representations must agree. Since

$e^* \neq 0$, equality of the coefficient vectors forces $j = j^*$ and equality of the constant terms then gives $s = s^*$, contradicting the freshness of (j^*, m^*).

Let $(I^*, \mathbf{a}, b)$ be the unique symbolic representation of I^* when the critical RO response is sampled. On $\mathsf{BadQuery}$ and outside the bridge event, this representation must agree with the one produced by final verification: $(I^*, \mathbf{a}, b) = (I^*, -e^*\hat{u}_{j^*}, s^*)$. Hence, $\mathbf{a} = -e^*\hat{u}_{j^*}$ and $b = s^*$. For a fixed RO query and its already determined representation $(\mathbf{a}, b)$, there is therefore at most one RO response that can be consistent with a bridge-free successful forgery: if $\mathbf{a}$ has support other than a single coordinate, there is no such response; otherwise both the target user and the required response are uniquely determined. Since the RO response is uniform in $\{0,1\}^{\lambda_\mathsf{H}}$, a union bound over the q_H^on online RO queries gives $\Pr[\mathsf{BadQuery}] \leq \frac{q_\mathsf{H}^\mathsf{on}}{2^{\lambda_\mathsf{H}}}$.

Finally, define $D_{\mathsf{sig}} \coloneqq p - 2P_\mathsf{G}(N + 3\overline{q}_\mathsf{G}^\mathsf{on} + 1) - (N + 3\overline{q}_\mathsf{G}^\mathsf{on} + 1)^2 - N$. Applying Lemma 1 to $\mathcal{B}$, whose total number of primitive generic-group queries is at most $\overline{q}_\mathsf{G}^\mathsf{on}$, and combining the probability of bad events and applying to (1), we obtain

$$\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda) = 1\right] \leq \frac{\binom{N}{2} + N(P_\mathsf{G}+1)}{p} + \frac{\overline{q}_\mathsf{G}^\mathsf{on}(N+P_\mathsf{G}) + 3\overline{q}_\mathsf{G}^\mathsf{on}(\overline{q}_\mathsf{G}^\mathsf{on}+1)/2}{D_{\mathsf{sig}}}$$
$$+ \frac{q_\mathsf{S}^\mathsf{on}(P_\mathsf{H} + q_\mathsf{H}^\mathsf{on} + q_\mathsf{S}^\mathsf{on})}{p} + \frac{(P_\mathsf{H} + q_\mathsf{H}^\mathsf{on})(N + \overline{q}_\mathsf{G}^\mathsf{on})}{p - (N + P_\mathsf{G} + 3\overline{q}_\mathsf{G}^\mathsf{on} + 1)} + \frac{q_\mathsf{H}^\mathsf{on} + q_\mathsf{S}^\mathsf{on} + 1}{2^{\lambda_\mathsf{H}}}. \qquad \square$$

4.3 Lifting Nonzero Schnorr to the AI-(ROM+GGM)

We now lift the prior BF-(ROM+GGM) bound to the AI-(ROM+GGM). There is one technical issue in applying the BF-to-AI lifting theorem directly. Recall that the signing algorithm of Π_{NZ} uses rejection sampling: if the hash challenge is zero, the signer *resamples* a fresh nonce and repeats. Consequently, a signing query does not have a deterministic worst-case bound on the number of RO and generic group queries made by the challenger.

This matters because the BF-to-AI lifting theorems (Theorem 4 and Theorem 5) are parameterized by the *combined query complexity* of the application, which counts the queries made by the online adversary *together with* those made by the challenger. As noted in Footnote 6, loop-free variants are possible, but they do not exactly match the rejection-sampling formulation used in the BSI/ISO specifications [fIS18, fSC18] motivating our analysis. We therefore handle the unbounded retry loop directly rather than modifying the signature scheme.

We consider an intermediate forgery game in which each signing query is allowed only one signing attempt, and account separately for the event that this attempt produces the zero challenge. The truncated game has deterministically bounded combined query complexity, while the probability of requiring a retry is bounded through a separate application. Thus, the AI-(ROM+GGM) analysis reduces to two applications: the *truncated forgery application* and the *retry application*. For the former, the BF security bound follows immediately from Section 4.2, whereas the latter requires a separate BF bound. We then lift both bounds to the AI-(ROM+GGM) and combine them.

Truncating the Rejection Loop. Let $\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi_{\mathsf{NZ}}}(\lambda; 1)$ denote the single-attempt variant of $\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi_{\mathsf{NZ}}}(\lambda)$, i.e., for each signing query, the challenger samples one nonce and aborts the experiment with output 0 if the resulting challenge is zero. Let Retry denote the corresponding event in the original experiment that some signing query requires a second attempt. Then

$$\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi_{\mathsf{NZ}}}(\lambda) = 1\right] \leq \Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi_{\mathsf{NZ}}}(\lambda; 1) = 1\right] + \Pr[\mathsf{Retry}]\,. \tag{2}$$

Let $\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}$ denote the corresponding truncated signature forgery application. Then, in $\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}$, each signing query makes at most one RO query and one generic group query. Hence, $(T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}} \leq q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + 1$ and $(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}} \leq q_{\mathsf{G}}^{\mathsf{on}} + N + q_{\mathsf{S}}^{\mathsf{on}} + 1$. Here, the N generic group queries arise from key generation, and the final $+1$ term comes from the forgery verification.

The Retry Application. Define $\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}$ using the same interaction as the nonzero Schnorr forgery experiment, except that its winning condition is that some signing query produces the zero challenge on its first attempt. For each signing query, the challenger samples a single nonce and evaluates the corresponding challenge. If the challenge is zero, it outputs 1 and terminates; otherwise, it returns the resulting signature and continues the interaction. If no such event occurs before the adversary terminates, the challenger outputs 0, irrespective of the adversary's final output. By construction,

$$\Pr[\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}(\mathcal{A}) = 1] = \Pr[\mathsf{Retry}]\,. \tag{3}$$

Moreover, $(T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}} \leq q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}$ and $(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}} \leq q_{\mathsf{G}}^{\mathsf{on}} + N + q_{\mathsf{S}}^{\mathsf{on}}$ since the application omits final forgery verification.

BF Bounds for the Bounded-Query Applications. We first bound the two applications in the BF-(ROM+GGM). For the truncated forgery application $\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}$, no new BF analysis is needed since every successful execution of $\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}$ is also a successful execution of the original nonzero Schnorr forgery experiment. Hence, by Theorem 6, for a bit-fixing adversary $\mathcal{A}$,

$$\Pr[\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}(\mathcal{A}) = 1] \leq \varepsilon^{\mathsf{BF}}_{\mathsf{NZ}}, \tag{4}$$

where $\varepsilon^{\mathsf{BF}}_{\mathsf{NZ}}$ denotes the bound from Theorem 6.

It remains to bound the retry application $\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}$ in the BF-(ROM+GGM).

Lemma 2. *Let $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathsf{G}})}, \mathcal{A}_{\mathsf{on}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathsf{G}})})$ be a bit-fixing adversary. Suppose that $\mathcal{A}_{\mathsf{on}}$ makes at most $q_{\mathsf{H}}^{\mathsf{on}}$ RO queries, $q_{\mathsf{G}}^{\mathsf{on}}$ generic group queries, and $q_{\mathsf{S}}^{\mathsf{on}}$ signing queries. Then,*

$$\Pr[\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}(\mathcal{A}) = 1] \leq q_{\mathsf{S}}^{\mathsf{on}} \left(\frac{P_{\mathsf{H}} + q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}}{p} + \frac{1}{2^{\lambda_{\mathsf{H}}}} \right),$$

where the probability is over the ideal oracles and all random coins of the experiment and the adversary.

Proof. Fix a signing query (j, m). At its first signing attempt, the signer samples a fresh nonce $r \leftarrow\!\!\$\, \mathbb{Z}_p$ and computes $I = \tau(r)$. Throughout the application, at most P_{H} RO coordinates are fixed during preprocessing, at most $q_{\mathsf{H}}^{\mathsf{on}}$ are defined by the adversary's RO queries, and at most $q_{\mathsf{S}}^{\mathsf{on}}$ are defined by signing attempts. Hence the RO table contains at most $P_{\mathsf{H}} + q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}$ coordinates.

For the fixed message m, each previously defined coordinate $I' \| m$ can be hit by at most one value of r, by injectivity of τ. Thus, $\Pr[(I\|m) \text{ is already defined}] \leq \frac{P_{\mathsf{H}} + q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}}{p}$. Otherwise, the RO response at $I\|m$ is fresh and uniform and equals zero with probability $2^{-\lambda_{\mathsf{H}}}$. A union bound over the $q_{\mathsf{S}}^{\mathsf{on}}$ signing queries proves the claim. □

Lifting to the AI-(ROM+GGM). We are now ready to lift the preceding bounds to the AI-(ROM+GGM). The truncated forgery application $\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}$ is an unpredictability application, so we apply Theorem 5 together with the BF bound from Theorem 6. In contrast, the retry application $\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}$ only detects whether a particular event occurs during the interaction, so we apply Lemma 2 together with the general additive lifting theorem (Theorem 4).

Theorem 7. *Let $p > 2^{2\lambda}$ be prime and suppose that $2^{\lambda_{\mathsf{H}}} < p$. Let $\mathcal{A}_{\mathsf{Sig}} = \left(\mathcal{A}^{\mathsf{AI\text{-}RO+GG}}_{\mathsf{Sig.pre}}, \mathcal{A}^{\mathsf{AI\text{-}RO+GG}}_{\mathsf{Sig.on}}\right)$ be an auxiliary-input adversary such that $\mathcal{A}_{\mathsf{Sig.pre}}$ outputs an S-bit hint. Suppose that $\mathcal{A}_{\mathsf{Sig.on}}$ makes at most $q^{\mathsf{on}}_{\mathsf{H}}$ random-oracle queries, $q^{\mathsf{on}}_{\mathsf{G}}$ primitive generic-group queries (i.e., `Mult` and `Inv`), and $q^{\mathsf{on}}_{\mathsf{S}}$ signing queries, and let $q \coloneqq q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{G}} + q^{\mathsf{on}}_{\mathsf{S}}$. Then,*

$$\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda) = 1\right] \le \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Forge}} + \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}},$$

where the two terms $\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Forge}}$ and $\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}}$ are given explicitly in Equations (5) and (6) below. Moreover, whenever the denominators in the explicit bound are $\Omega(p)$,

$$\begin{aligned}&\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda) = 1\right]\\ &= O\left(\frac{S(N+q)(N+q\log p)+q^2\log^2 p}{p} + \frac{q}{2^{\lambda_{\mathsf{H}}}} + \sqrt{\frac{q(S+\lambda)(N+q)\log p}{p\lambda_{\mathsf{H}}}} + 2^{-2\lambda}\right),\end{aligned}$$

where the probability is over the ideal oracles and all random coins of the experiment and the adversary.

Proof. We first bound the truncated application. Recall that $(T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}} \le q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{S}} + 1 =: Q_{\mathsf{H}}$ and $(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}} \le q^{\mathsf{on}}_{\mathsf{G}} + N + q^{\mathsf{on}}_{\mathsf{S}} + 1 =: Q_{\mathtt{G}}$. Hence, for $\gamma > 0$, we can choose $P_{\mathsf{H}} = \lceil (S + \log\gamma^{-1})Q_{\mathsf{H}}\rceil$ and $P_{\mathtt{G}} = \lceil 6(S+\log\gamma^{-1})Q_{\mathtt{G}}\rceil$. Then since $p > 2^{2\lambda}$, we have $\log p/\log(p/e) \le 2$ and hence, $(S+\log\gamma^{-1})\left((T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}}\lambda_{\mathsf{H}} + 3(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Forge}}}\log p\right) \le P_{\mathsf{H}}\lambda_{\mathsf{H}} + P_{\mathtt{G}}\log(p/e)$. Therefore, we can apply Theorem 5 together with Equation (4) and set $\gamma = 2^{-2\lambda}$ to get $\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A},\Pi_{\mathsf{NZ}}}(\lambda;1) = 1\right] \le \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Forge}}$, where

$$\begin{aligned}&\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Forge}}\\ &= \frac{2\binom{N}{2} + 2N(\lceil 6(S+2\lambda)Q_{\mathtt{G}}\rceil + 1)}{p} + \frac{2\overline{q}^{\mathsf{on}}_{\mathtt{G}}(N + \lceil 6(S+2\lambda)Q_{\mathtt{G}}\rceil) + 3\overline{q}^{\mathsf{on}}_{\mathtt{G}}(\overline{q}^{\mathsf{on}}_{\mathtt{G}}+1)}{D_{\mathsf{sig}}}\\ &\quad + \frac{2q^{\mathsf{on}}_{\mathsf{S}}(\lceil (S+2\lambda)Q_{\mathsf{H}}\rceil + q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{S}})}{p} + \frac{2(\lceil (S+2\lambda)Q_{\mathsf{H}}\rceil + q^{\mathsf{on}}_{\mathsf{H}})(N+\overline{q}^{\mathsf{on}}_{\mathtt{G}})}{p - (N + \lceil 6(S+2\lambda)Q_{\mathtt{G}}\rceil + 3\overline{q}^{\mathsf{on}}_{\mathtt{G}} + 1)}\\ &\quad + \frac{q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{S}} + 1}{2^{\lambda_{\mathsf{H}}-1}} + 2^{-2\lambda+1}\\ &= O\left(\frac{S(N+q)(N+q\log p)+q^2\log^2 p}{p} + \frac{q}{2^{\lambda_{\mathsf{H}}}} + 2^{-2\lambda}\right).\end{aligned} \tag{5}$$

We next bound the retry application. Recall that $(T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}} \le q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{S}} =: Q'_{\mathsf{H}}$ and $(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}} \le q^{\mathsf{on}}_{\mathtt{G}} + N + q^{\mathsf{on}}_{\mathsf{S}} =: Q'_{\mathtt{G}}$. To apply Theorem 4, fix $\eta > 0$ and consider any $P_{\mathsf{H}}, P_{\mathtt{G}}$ satisfying $P_{\mathsf{H}}\lambda_{\mathsf{H}} +$

$P_{\mathsf{G}} \log(p/e) \le \eta$. Since $P_{\mathsf{G}} \ge 0$, every such choice satisfies $P_{\mathsf{H}} \le \eta/\lambda_{\mathsf{H}}$. Hence, applying Theorem 4 together with Lemma 2 gives $\Pr[\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}(\mathcal{A}) = 1] \le \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}}$, where

$$\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}} = q_{\mathsf{S}}^{\mathsf{on}} \left(\frac{\eta/\lambda_{\mathsf{H}} + q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}}{p} + \frac{1}{2^{\lambda_{\mathsf{H}}}} \right) + \frac{2(S + \log \gamma^{-1}) \left(Q'_{\mathsf{H}} \lambda_{\mathsf{H}} + 3 Q'_{\mathsf{G}} \log p \right)}{\eta} + 2\gamma.$$

For $q_{\mathsf{S}}^{\mathsf{on}} > 0$, set $\gamma := 2^{-2\lambda}$ and $\eta := \sqrt{\frac{(S+2\lambda)\left(Q'_{\mathsf{H}} \lambda_{\mathsf{H}} + 3 Q'_{\mathsf{G}} \log p\right) p \lambda_{\mathsf{H}}}{q_{\mathsf{S}}^{\mathsf{on}}}}$. Then

$$\begin{aligned} \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}} &\le q_{\mathsf{S}}^{\mathsf{on}} \left(\frac{q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}}}{p} + \frac{1}{2^{\lambda_{\mathsf{H}}}} \right) + 3 \sqrt{\frac{q_{\mathsf{S}}^{\mathsf{on}} (S + 2\lambda) \left(Q'_{\mathsf{H}} \lambda_{\mathsf{H}} + 3 Q'_{\mathsf{G}} \log p \right)}{p \lambda_{\mathsf{H}}}} + 2^{-2\lambda+1} \\ &= O\left(\frac{q^2}{p} + \frac{q}{2^{\lambda_{\mathsf{H}}}} + \sqrt{\frac{q(S+\lambda)(N+q)\log p}{p \lambda_{\mathsf{H}}}} + 2^{-2\lambda} \right). \end{aligned} \tag{6}$$

If $q_{\mathsf{S}}^{\mathsf{on}} = 0$, then $\Pr[\mathcal{G}_{\mathsf{NZ},\mathsf{Retry}}(\mathcal{A}) = 1] = 0$. Otherwise, the bound in Equation (6) applies. Finally, by combining Equation (2) with Equations (3), (5), and (6) gives the claimed bound. □

4.4 Short Signatures and Parameters

We now instantiate the bound of Theorem 7 and examine the resulting signature length. We first note that the theorem already establishes preprocessing security for the standardized nonzero Schnorr construction with a full-length challenge. We then show that, in the standard multi-user setting without corruption, the challenge can be shortened substantially without introducing an additional loss depending on the number of users. Suppose that $q_{\mathsf{H}}^{\mathsf{on}}, q_{\mathsf{G}}^{\mathsf{on}}, q_{\mathsf{S}}^{\mathsf{on}} = O(q)$, $N \le q \le 2^{\lambda}$, and assume $\log(S + \lambda) = O(\lambda)$. Choose $p \approx (S + \lambda)\lambda 2^{2\lambda}$. Then $\log p = 2\lambda + \log(S + \lambda) + \log \lambda + O(1) = O(\lambda)$. Substituting these relations into Theorem 7, the group-dependent terms satisfy $\frac{S(N+q)(N+q\log p)+q^2 \log^2 p}{p} = O\left(\frac{q}{2^{\lambda}}\right)$, and the retry term satisfies $\sqrt{\frac{q(S+\lambda)(N+q)\log p}{p\lambda_{\mathsf{H}}}} = O\left(\frac{q}{2^{\lambda}}\right)$ whenever $\lambda_{\mathsf{H}} = \Omega(\lambda)$.

Full-length challenges. The standardizations motivating the nonzero variant use a full-length hash challenge [fIS18, fSC18]; in the usual asymptotic parameterization this corresponds to taking λ_{H} on the order of 2λ (or of $\log p$, depending on the parameterization). For either choice, $\frac{q}{2^{\lambda_{\mathsf{H}}}} \le O\left(\frac{q}{2^{2\lambda}}\right)$, and hence Theorem 7 gives

$$\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda) = 1 \right] = O\left(\frac{q}{2^{\lambda}}\right).$$

Thus, the standardized nonzero Schnorr construction retains λ-bit preprocessing security for the above choice of group size.

Shortening the challenge. More interestingly, the same security level is obtained with $\lambda_{\mathsf{H}} \approx \lambda$. Indeed, the short-challenge term in Theorem 7 is only $q/2^{\lambda_{\mathsf{H}}}$, so setting $\lambda_{\mathsf{H}} = \lambda + O(1)$ yields

$$\Pr\left[\mathsf{SigForge}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda) = 1 \right] = O\left(\frac{q}{2^{\lambda}}\right).$$

In particular, no additional $\log N$ bits are required in the challenge in the standard multi-user game. A nonzero Schnorr signature consists of $s \in \mathbb{Z}_p$ and an λ_{H}-bit challenge, and therefore has

length $\lceil \log_2 p \rceil + \lambda_{\mathsf{H}}$. With the above parameter choice and $\lambda_{\mathsf{H}} \approx \lambda$, this becomes $3\lambda + \log(S + \lambda) + \log \lambda + O(1)$ bits. Compared with a full 2λ-bit challenge, this saves approximately λ bits while preserving the same asymptotic preprocessing-security guarantee. For comparison with the familiar 4λ-bit Schnorr signatures, the resulting signatures are shorter whenever $\log(S + \lambda) + \log \lambda < \lambda$ up to lower-order additive terms. For example, for $\lambda = 128$ and $S = 2^{72}$, the above choice gives $\log p \approx 2\lambda + \log S + \log \lambda = 335$, and a $\lambda_{\mathsf{H}} \approx 128$-bit challenge gives a signature of approximately $335 + 128 = 463$ bits.

5 Nonzero Schnorr Under Adaptive Corruptions

In Section 4, we established preprocessing security of nonzero Schnorr signatures in the standard multi-user forgery game. In particular, the short-challenge term is $O(q/2^{\lambda_{\mathsf{H}}})$, allowing $\lambda_{\mathsf{H}} \approx \lambda$ in the parameter regime considered there.

We now revisit the analysis when the adversary may *corrupt users adaptively* during the online attack. Most of the reduction from the standard multi-user setting remains unchanged. The key phenomenon for short challenges is that, after observing a response to a query $\mathsf{H}(I\|m)$, the adversary may reveal selected secret keys and thereby remove their coordinates from the symbolic representation of I. As a result, a single random-oracle query may remain compatible with several possible uncorrupted target users.

We first isolate this phenomenon through an explicit attack that does not use preprocessing. We then extend the bridge-finding game of Section 4.1 to adaptive corruptions, use it to derive the corresponding BF-(ROM+GGM) and AI-(ROM+GGM) bounds, and finally examine the resulting short-signature parameters.

5.1 Adaptive Target Switching

The key difference from the standard multi-user game is that a nonzero Schnorr RO query $\mathsf{H}(I\|m)$ does not identify the public key that will eventually be targeted by a forgery. Under adaptive corruption, the adversary can therefore prepare a commitment involving several users, observe the RO response, and only then decide which user will remain uncorrupted. We call this strategy *adaptive target switching*.

Related analyses of Schnorr signatures under adaptive corruption already allow the final forgery to be matched to any uncorrupted user after the challenge queries; in particular, this flexibility is captured by the $\mathsf{CorrIDLOG}$ framework of Pan, Qian, and Ringerud [PQR22]. Our focus here is on the effect of this flexibility for a shortened challenge space: it gives rise to an explicit multi-target attack whose success probability grows with the number of candidate users.

Importantly, this phenomenon is not caused by preprocessing. It already gives an attack in the ordinary ROM+GGM with adaptive corruptions. Fix $k \le \min\left\{N, \left\lfloor \frac{q_{\mathsf{G}}^{\mathsf{on}}+1}{2} \right\rfloor, 2^{\lambda_{\mathsf{H}}} - 1\right\}$, and consider the first k users. The adversary computes the suffix products $S_r := \prod_{i=r}^{k} \mathsf{pk}_i$ for each $r \in [k]$, followed by $J := \prod_{r=1}^{k} S_r$ and $I := J^{-1}$. This requires $2k - 1$ generic group queries and yields $I = \tau\left(-\sum_{i=1}^{k} i x_i\right)$. The adversary now queries $e \leftarrow \mathsf{H}(I\|m)$. If $e = j$ for some $j \in [k]$, it corrupts every user in $[k] \setminus \{j\}$ and sets $s := -\sum_{i \in [k]\setminus\{j\}} i x_i$. Then $I = \tau(s - e x_j)$, and therefore (s, e) is a valid nonzero Schnorr signature on m for the remaining uncorrupted user j.

A single RO query succeeds with probability $k/2^{\lambda_{\mathsf{H}}}$. Repeating the attack on independent messages gives success probability $1 - \left(1 - \frac{k}{2^{\lambda_{\mathsf{H}}}}\right)^{q_{\mathsf{H}}^{\mathsf{on}}} = \Theta\left(\min\left\{1, \frac{k q_{\mathsf{H}}^{\mathsf{on}}}{2^{\lambda_{\mathsf{H}}}}\right\}\right)$. Thus, up to a constant factor

in the generic group query complexity, adaptive corruption introduces a multi-target loss of order $\frac{q_{\mathsf{H}}^{\mathsf{on}} \min\{N, q_{\mathsf{G}}^{\mathsf{on}}+1\}}{2^{\lambda_{\mathsf{H}}}}$. This explains precisely what changes from Section 4. Without corruption, a bridge-free critical RO query can be consistent with at most one eventual target and one challenge value. Under adaptive corruption, the adversary may instead remove other coordinates after seeing the RO response, leaving any one of several users as the final uncorrupted target. Section 5.2 extends the bridge-finding game to capture this operation.

5.2 Bridge Finding Under Adaptive Corruptions

We extend the bridge-finding game of Section 4.1 to adaptive corruptions. The experiment is identical to Figure 3, except that the online adversary is additionally given access to a corruption oracle. The challenger maintains the set $\mathsf{C} \subseteq [N]$ of corrupted users, initially $\mathsf{C} = \emptyset$. On a query $\texttt{Corrupt}(i)$, it returns x_i and substitutes the revealed value into every symbolic representation in $\mathcal{L}$: $(\mathfrak{y}, \mathbf{a}, b) \longmapsto (\mathfrak{y}, \mathbf{a} - a_i\hat{u}_i, b + a_i x_i)$. We write $\mathsf{BridgeChalCor}_{\mathcal{A}}^{\tau,\mathsf{H},N}(\lambda)$ for the resulting experiment.

The substitution preserves the group element represented by every tuple in $\mathcal{L}$. It does not increase $|\mathcal{L}|$, and after user i is corrupted, every tuple has zero coefficient in coordinate i. Moreover, corruption itself cannot create a new bridge: applying the same substitution to two identical symbolic representations keeps them identical, while any bridge that occurred earlier has already set the sticky win flag.

Security of the Adaptive Bridge-Finding Game. The adaptive corruption oracle does not worsen the bridge-finding bound. Intuitively, corruption only substitutes already revealed secret values into the symbolic representations: it neither enlarges $\mathcal{L}$ nor introduces additional constraints on the remaining uncorrupted secrets. The following lemma therefore gives exactly the same bound as in Lemma 1.

Lemma 3. *Under the assumptions of Lemma 1, the same bound holds for the adaptive-corruption bridge game:*

$$\Pr\left[\mathsf{BridgeChalCor}_{\mathcal{A}}^{\tau,\mathsf{H},N}(\lambda) = 1\right] \leq \frac{\binom{N}{2} + N(P_{\mathsf{G}}+1)}{p} + \frac{q_{\mathsf{G}}^{\mathsf{on}}(N+P_{\mathsf{G}}) + 3q_{\mathsf{G}}^{\mathsf{on}}(q_{\mathsf{G}}^{\mathsf{on}}+1)/2}{D_{\mathsf{br}}},$$

where D_{br} is defined as in Lemma 1.

Proof. We follow the proof of Lemma 1 and describe only the effect of corruption queries. A corruption query substitutes a revealed value x_i into every tuple in $\mathcal{L}$. This operation preserves the represented group elements and does not increase $|\mathcal{L}|$. It can only remove nonzero coefficients from symbolic representations; in particular, $\mathcal{L}_0 = \{(\mathfrak{y}, \mathbf{a}, b) \in \mathcal{L} : \mathbf{a} = \mathbf{0}\}$ can only grow. Hence, the bounds on $|\mathcal{L}|$ and $|\mathcal{L}_0|$ used in the proof of Lemma 1 continue to hold.

Now consider an online generic-group query, conditioned on no bridge having occurred previously. As in Lemma 1, a prospective bridge with a previously recorded tuple gives an equation $(\mathbf{a}-\mathbf{a}_i)\cdot\mathbf{x} = b_i - b$. All corrupted coordinates have zero coefficient in both representations. Therefore, if $\mathbf{a} \neq \mathbf{a}_i$, there exists an uncorrupted index $r \notin \mathsf{C}$ such that $a[r] - a_i[r] \neq 0$.

Condition on the revealed corrupted secrets and expose this x_r last. The exclusion argument from Lemma 1 then applies unchanged: corruption neither adds tuples to $\mathcal{L}$ nor creates any additional exclusion on the remaining secret coordinates. Thus the same denominator D_{br} and the same per-query bridge bound hold. The initialization bound is unchanged as well, since no corruption has occurred at initialization. Applying the same union bound as in Lemma 1 proves the claim. □

5.3 Nonzero Schnorr Under Adaptive Corruptions in the BF-(ROM+GGM)

We now revisit the BF-(ROM+GGM) analysis of Section 4.2 in the presence of adaptive corruptions. The reduction, signing simulation, and query accounting remain unchanged. By Lemma 3, adaptive corruption also does not worsen the bridge-finding bound. The only substantive change is the analysis of the critical random-oracle query: because a query $\mathsf{H}(I\|m)$ does not bind a target user, subsequent corruptions may leave several possible users compatible with the same RO response.

Theorem 8. *Under the assumptions of Theorem 6, let $\mathcal{A}_{\mathsf{Sig}}$ additionally have adaptive access to the corruption oracle, and let $\varepsilon^{\mathsf{BF}}_{\mathsf{NZ}}$ denote the bound of Theorem 6. Then*

$$\Pr\left[\mathsf{SigForgeCor}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda)=1\right] \le \varepsilon^{\mathsf{BF}}_{\mathsf{NZ}} + \frac{q^{\mathsf{on}}_{\mathsf{H}}\big(\min\{N, q^{\mathsf{on}}_{\mathsf{G}}+1\}-1\big)}{2^{\lambda_{\mathsf{H}}}}.$$

Hence, when $D_{\mathsf{sig}} = \Omega(p)$,

$$\Pr\left[\mathsf{SigForgeCor}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda)=1\right] = O\left(\frac{(N+P_{\mathsf{H}}+q\log p)(N+P_{\mathsf{G}}+q\log p)}{p} + \frac{q\min\{N,q\}}{2^{\lambda_{\mathsf{H}}}}\right).$$

Proof. We use the reduction $\mathcal{B}$ from the proof of Theorem 6, with one modification: whenever $\mathcal{A}_{\mathsf{Sig.on}}$ makes a corruption query $\mathtt{Corrupt}(i)$, the reduction forwards it to the adaptive bridge challenger of Section 5.2, obtains x_i, and returns it to the adversary. The corresponding substitution in $\mathcal{L}$ is performed by the challenger. All other parts of the simulation are unchanged. In particular, the signing simulation and final-forgery processing are identical to those in the proof of Theorem 6, and the reduction still makes at most $\overline{q}^{\mathsf{on}}_{\mathsf{G}} = q^{\mathsf{on}}_{\mathsf{G}} + (q^{\mathsf{on}}_{\mathsf{S}}+1)\big(2\lceil\log_2 p\rceil + 2\lambda_{\mathsf{H}} + 2\big)$ generic group queries.

Define Forge, $\mathsf{FailtoSign}$, $\mathsf{NoQuery}$, $\mathsf{EarlyRO}$, and $\mathsf{BadQuery}$ exactly as in the proof of Theorem 6, except that a successful forgery *must target an uncorrupted user*. The implication $\mathsf{Forge} \subseteq \mathsf{FailtoSign} \cup \{\mathsf{Bridge} = 1\} \cup \mathsf{NoQuery} \cup \mathsf{EarlyRO} \cup \mathsf{BadQuery}$ continues to hold. The bounds for $\mathsf{FailtoSign}$, $\mathsf{NoQuery}$, and $\mathsf{EarlyRO}$ are unchanged. Corruption queries neither define RO coordinates nor add tuples to $\mathcal{L}$, and Lemma 3 gives the same bridge bound as in the standard multi-user analysis. It therefore remains only to reconsider $\mathsf{BadQuery}$.

As in the proof of Theorem 6, on $\mathsf{BadQuery}$, the coordinate $I^*\|m^*$ must be first defined by an online RO query of $\mathcal{A}_{\mathsf{Sig.on}}$. It cannot originate from preprocessing, since then the RO coordinate would precede any nonzero symbolic representation of I^*. Nor can it have been programmed during signing. Indeed, suppose a signing query (j,m) produced (s,e) and $I = \tau(s - ex_j)$ at the same RO coordinate as the final forgery. Outside the bridge event, the signing and forgery representations must agree after all corruption substitutions. Since the target j^* remains uncorrupted and $e^* \neq 0$, this forces $j = j^*$ and $s = s^*$, contradicting freshness. Let $(I^*, \mathbf{a}, b)$ be the unique representation of I^* when the RO response is sampled. Define the support of $\mathbf{a}$ by $\mathrm{supp}(\mathbf{a}) \coloneqq \{i \in [N] : \mathbf{a}[i] \neq 0\}$. Every label available to the adversary without using one of its own generic group queries has coefficient support of size at most one, and each generic group query can increase the number of such components combined in an expression by at most one. Hence, after at most $q^{\mathsf{on}}_{\mathsf{G}}$ such queries, since there are at most N users, $|\mathrm{supp}(\mathbf{a})| \le \min\{N, q^{\mathsf{on}}_{\mathsf{G}}+1\}$.

Subsequent corruptions can only remove coordinates from $\mathrm{supp}(\mathbf{a})$. Hence, if the online RO query that first defines $I^*\|m^*$ is later used in a forgery without bridge for an uncorrupted user j^*, then $j^* \in \mathrm{supp}(\mathbf{a})$ and $e^* = -\mathbf{a}[j^*]$. Thus, before that RO response is sampled, there are at most $|\mathrm{supp}(\mathbf{a})| \le \min\{N, q^{\mathsf{on}}_{\mathsf{G}}+1\}$ responses that can avoid a bridge. Since the response is uniform in $\{0,1\}^{\lambda_{\mathsf{H}}}$, union bounding over the $q^{\mathsf{on}}_{\mathsf{H}}$ online RO queries gives $\Pr[\mathsf{BadQuery}] \le \frac{q^{\mathsf{on}}_{\mathsf{H}}\min\{N, q^{\mathsf{on}}_{\mathsf{G}}+1\}}{2^{\lambda_{\mathsf{H}}}}$.

All other bad-event bounds are identical to those in Theorem 6, and the bridge term is bounded by Lemma 3. Replacing the standard bound $q_{\mathsf{H}}^{\mathsf{on}}/2^{\lambda_{\mathsf{H}}}$ for $\mathsf{BadQuery}$ by the bound above gives exactly the stated expression. □

Together with the adaptive target-switching attack of Section 5.1, this shows that the short-challenge term is *tight* up to constant factors: in the nontrivial regime, the attack achieves success probability $\Theta\left(\frac{q_{\mathsf{H}}^{\mathsf{on}} \min\{N, q_{\mathsf{G}}^{\mathsf{on}}\}}{2^{\lambda_{\mathsf{H}}}}\right)$, up to a constant factor in generic-group query complexity.

5.4 Nonzero Schnorr Under Adaptive Corruptions in the AI-(ROM+GGM)

The single-attempt truncation and retry analysis of Section 4.3 is unchanged under adaptive corruption. In particular, corruption queries neither define random-oracle coordinates nor change the combined RO and generic-group query complexities of the two applications. Hence, the retry application and its AI-(ROM+GGM) bound are exactly the same as in Section 4.3. The only change is that the truncated forgery application now uses the adaptive-corruption BF bound of Theorem 8.

Theorem 9. *Under the same parameter and query assumptions as Theorem 7, let* $\mathcal{A}_{\mathsf{Sig}} = (\mathcal{A}_{\mathsf{Sig.pre}}^{\mathsf{AI\text{-}RO+GG}}, \mathcal{A}_{\mathsf{Sig.on}}^{\mathsf{AI\text{-}RO+GG}})$ *be an auxiliary-input adversary with adaptive access to the corruption oracle. Then,*

$$\Pr\left[\mathsf{SigForgeCor}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda) = 1\right] \le \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{ForgeCor}} + \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}},$$

where $\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}}$ *is exactly the retry bound from Equation* (6)*, and*

$$\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{ForgeCor}} = \varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Forge}} + \frac{q_{\mathsf{H}}^{\mathsf{on}}\left(\min\{N, q_{\mathsf{G}}^{\mathsf{on}}+1\}-1\right)}{2^{\lambda_{\mathsf{H}}-1}}.$$

Here, $\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Forge}}$ *is the bound from Equation* (5)*. Consequently, whenever the denominators in the explicit bound are* $\Omega(p)$*,*

$$\begin{aligned}&\Pr\left[\mathsf{SigForgeCor}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{NZ}}}(\lambda) = 1\right]\\ &= O\left(\frac{S(N+q)(N+q\log p)+q^2\log^2 p}{p} + \frac{q\min\{N,q\}}{2^{\lambda_{\mathsf{H}}}} + \sqrt{\frac{q(S+\lambda)(N+q)\log p}{p\lambda_{\mathsf{H}}}} + 2^{-2\lambda}\right).\end{aligned}$$

Proof. Apply the same single-attempt truncation as in the proof of Theorem 7. Adaptive corruption does not affect the retry application: corruption queries neither define RO coordinates nor change its RO or generic-group query complexity. Hence, $\varepsilon^{\mathsf{AI}}_{\mathsf{NZ},\mathsf{Retry}}$ remains exactly as in Equation (6). For the truncated forgery application, the combined query complexities are also unchanged. We therefore apply Theorem 5 exactly as in Theorem 7, but use Theorem 8 in place of Theorem 6. By Theorem 8, the only change in the BF bound is that $\frac{q_{\mathsf{H}}^{\mathsf{on}}+q_{\mathsf{S}}^{\mathsf{on}}+1}{2^{\lambda_{\mathsf{H}}}}$ is replaced by $\frac{q_{\mathsf{H}}^{\mathsf{on}}\min\{N,q_{\mathsf{G}}^{\mathsf{on}}+1\}+q_{\mathsf{S}}^{\mathsf{on}}+1}{2^{\lambda_{\mathsf{H}}}}$. The lifting multiplies this BF contribution by two, yielding the additional term $\frac{q_{\mathsf{H}}^{\mathsf{on}}\left(\min\{N,q_{\mathsf{G}}^{\mathsf{on}}+1\}-1\right)}{2^{\lambda_{\mathsf{H}}-1}}$. All remaining terms are identical to those in Theorem 7, which proves the claim. □

Parameters Under Adaptive Corruption. Compared with the standard multi-user analysis of Section 4.4, the group-dependent and retry terms are unchanged. The only new asymptotic contribution is $\frac{q \min\{N,q\}}{2^{\lambda_{\mathsf{H}}}}$. Suppose, as in Section 4.4, that $q_{\mathsf{H}}^{\mathsf{on}}, q_{\mathsf{G}}^{\mathsf{on}}, q_{\mathsf{S}}^{\mathsf{on}} = O(q)$ and $N \le q \le 2^{\lambda}$, and retain $p \approx (S+\lambda)\lambda 2^{2\lambda}$. To make the additional challenge term $O(q/2^{\lambda})$, it suffices to choose $\lambda_{\mathsf{H}} \approx \lambda + \log N$. Thus, adaptive corruption costs approximately $\log N$ additional challenge bits compared with the standard multi-user setting.

The resulting signature length is $\log p + \lambda_{\mathsf{H}} \approx 3\lambda + \log(S+\lambda) + \log\lambda + \log N$. For example, when $\lambda = 128$, $S = 2^{72}$, and $N = 2^{32}$, we obtain $\log p \approx 335$ and $\lambda_{\mathsf{H}} \approx 160$, giving signatures of approximately 495 bits. For these parameters, the adaptive-corruption cost is therefore 32 additional bits relative to the 463-bit standard multi-user instantiation of Section 4.4.

Key-prefixing. In Appendix C, we show that key-prefixing eliminates the adaptive target-switching loss even in the presence of preprocessing: the short-challenge term becomes $O(q/2^{\lambda_{\mathsf{H}}})$, rather than $O(q \min\{N,q\}/2^{\lambda_{\mathsf{H}}})$. Thus, one may again take $\lambda_{\mathsf{H}} \approx \lambda$ under the parameter regime above.

References

AFOS26. Ojaswi Acharya, Georg Fuchsbauer, Adam O'Neill, and Marek Sefranek. Revisiting the security of sparkle. Cryptology ePrint Archive, Paper 2026/431, 2026.

BHJ+15. Christoph Bader, Dennis Hofheinz, Tibor Jager, Eike Kiltz, and Yong Li. Tightly-secure authenticated key exchange. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 629–658. Springer, Berlin, Heidelberg, March 2015.

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.

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

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.

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.

CO15. Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In *Proceedings of the 4th International Conference on Progress in Cryptology – LATINCRYPT 2015 - Volume 9230*, page 40–58, Berlin, Heidelberg, 2015. Springer-Verlag.

fIS18. Federal Office for Information Security. Elliptic Curve Cryptography, Version 2.1. *Technical Guideline BSI TR-03111*, Jun 2018.

fSC18. International Organization for Standardization and International Electrotechnical Commission. IT Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms. *ISO/IEC 14888-3*, Nov 2018.

GJ18. Kristian Gjøsteen and Tibor Jager. Practical and tightly-secure digital signatures and authenticated key exchange. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 95–125. Springer, Cham, August 2018.

HJK+21. Shuai Han, Tibor Jager, Eike Kiltz, Shengli Liu, Jiaxin Pan, Doreen Riepel, and Sven Schäge. Authenticated key exchange and signatures with tight security in the standard model. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 670–700, Virtual Event, August 2021. Springer, Cham.

HLG23. Shuai Han, Shengli Liu, and Dawu Gu. Almost tight multi-user security under adaptive corruptions & leakages in the standard model. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 132–162. Springer, Cham, April 2023.

KMP16. Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal security proofs for signatures from identification schemes. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 33–61. Springer, Berlin, Heidelberg, August 2016.

Mau05. Ueli Maurer. Abstract models of computation in cryptography. In *Proceedings of the 10th International Conference on Cryptography and Coding*, IMA'05, page 1–12, Berlin, Heidelberg, 2005. Springer-Verlag.

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

PQR22. Jiaxin Pan, Chen Qian, and Magnus Ringerud. Signed (group) Diffie-Hellman key exchange with tight security. *Journal of Cryptology*, 35(4):26, October 2022.

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

Sha49. Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.

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.

SJ00. Claus-Peter Schnorr and Markus Jakobsson. Security of signed ElGamal encryption. In Tatsuaki Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 73–89. Springer, Berlin, Heidelberg, December 2000.

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.

Zha22. Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Cham, August 2022.

A Formal Definition of Bit-Fixing/Auxiliary-Input ROM+GGM

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

We first define an (H, τ)-source for a random oracle H and a generic-group random encoding map τ as follows.

Definition 3. *An* (H, τ)-source *is a pair of random variables* (X, Y) *where* X *corresponds to the function table of a random oracle* $\mathsf{H} : \{0,1\}^m \to \{0,1\}^\lambda$*, with range* $(\{0,1\}^\lambda)^{2^m}$*, and* Y *corresponds to the function table of an injection* $\tau : \mathbb{Z}_p \to \mathbb{G}$*. A source* (X, Y) *is called*

- $(P_1, P_2, 1-\delta)$-dense *if for* $(P_1, P_2) \in \mathbb{Z}_{2^m} \times \mathbb{Z}_p$ *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* $(I_1, I_2) \subseteq \{0,1\}^m \times \mathbb{Z}_p$ *of non-fixed coordinates,*
$$H_\infty(X_{I_1}, Y_{I_2}) \geq (1-\delta)\left[|I_1|\lambda + \log(p - P_2)^{\underline{|I_2|}}\right],$$
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 on the rest.*

Lemma 4. *Let* (X_1, X_2) *be a pair of random variables such that* X_1 *is distributed uniformly over* $(\{0,1\}^\lambda)^{2^m}$ *and* X_2 *is a uniformly random injection that takes on as value function tables corresponding to an injection* $\tau : \mathbb{Z}_p \to \mathbb{G}$*. Let* $Z \coloneqq f(X_1, X_2)$*, where* $f : (\{0,1\}^\lambda)^{2^m} \times \mathbb{G}^p \to \{0,1\}^S$ *is an arbitrary function. For any* $\gamma > 0$*, there exists a family* $\{(Y_1, Y_2)^{Z,\mathcal{Y}}\}$*, indexed by* $Z \in \{0,1\}^S$ *and size-*p *subsets* $\mathcal{Y}$ *of* $\mathbb{G}$*, of convex combinations of* (P_1, P_2)*-Bit-Fixing* (H, τ)*-sources satisfying* $P_1\lambda + P_2 \log(p/e) \leq W$ *such that for any distinguisher* $\mathcal{D}$ *taking an* S*-bit input and querying at most* $T_1 < P_1$ *coordinates and* $T_2 < P_2$ *coordinates of each oracle,*
$$\left|\Pr\left[\mathcal{D}^{X_1,X_2}(f(X_1,X_2)) = 1\right] - \Pr\left[\mathcal{D}^{(Y_1,Y_2)^{f(X_1,X_2),\mathsf{im}(X_2)}}(f(X_1,X_2)) = 1\right]\right| \leq \frac{(S + \log\gamma^{-1})(T_1\lambda + T_2 \log p)}{W} + \gamma,$$
and
$$\Pr\left[\mathcal{D}^{X_1,X_2}(f(X_1,X_2)) = 1\right] \leq 2^{(S+\log\gamma^{-1})(T_1\lambda + T_2\log p)/W} \cdot \Pr\left[\mathcal{D}^{(Y_1,Y_2)^{f(X_1,X_2),\mathsf{im}(X_2)}}(f(X_1,X_2)) = 1\right] + 2\gamma,$$
where e *is the Euler's number.*

Proof. Fix an arbitrary $z \in \{0,1\}^S$ and let (X_1^z, X_2^z) be the distribution of (X_1, X_2) conditioned on $f(X_1, X_2) = z$. Let $S_z = 2^m\lambda + \log p! - H_\infty(X_1, X_2)$ be the min-entropy deficiency of (X_1^z, X_2^z). Let $\gamma > 0$ be arbitrary.

Claim A.1. For every $\delta > 0$, any leaky source (X_1^z, X_2^z) is γ-close to a source (Y_1^z, Y_2^z) which is a convex combination of sources such that each source in the convex combination is $(P_1', P_2', 1-\delta)$-dense sources for some P_1' and P_2' which satisfy
$$P_1'\lambda + P_2' \log(p/e) \leq \frac{S_z + \log\gamma^{-1}}{\delta}.$$

Proof of Claim A.1. Without loss of generality, we can assume that (X_1^z, X_2^z) is not $(1-\delta)$-dense since if it is $(1-\delta)$-dense then we can trivially find such $(P_1', P_2', 1-\delta)$-dense sources. Let $(Y_1^z, Y_2^z) \coloneqq (X_1^z, X_2^z)$ and $I_1 \subseteq \{0,1\}^m$ and $I_2 \subseteq \mathbb{Z}_p$ be the largest subset for which there exists a violation, i.e., there exists $y_1^{I_1} \in \left(\{0,1\}^\lambda\right)^{|I_1|}$ and $y_2^{I_2} \in \mathbb{G}^{|I_2|}$ such that

$$\Pr\left[(Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2}\right] > 2^{-(1-\delta)\left[|I_1|\lambda + \log p^{\underline{|I_2|}}\right]}.$$

Let $(\widetilde{Y}_1^z, \widetilde{Y}_2^z)$ be the distribution of (Y_1^z, Y_2^z) conditioned on the event $(Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2}$.

(1) Now we claim that $(\widetilde{Y}_1^z, \widetilde{Y}_2^z)$ is $(P_1', P_2', 1-\delta)$-dense with $P_1' = |I_1|$ and $P_2' = |I_2|$. Suppose that $(\widetilde{Y}_1^z, \widetilde{Y}_2^z)$ is not $(P_1', P_2', 1-\delta)$-dense. Then there exists a pair of non-empty sets $J_1 \subseteq \overline{I_1}$ and $J_2 \subseteq \overline{I_2}$ and $(y_1^{J_1}, y_2^{J_2})$ such that

$$\begin{aligned}
&\Pr\left[(\widetilde{Y}_1^z)_{J_1} = y_1^{J_1} \wedge (\widetilde{Y}_2^z)_{J_2} = y_2^{J_2}\right] \\
&= \Pr\left[(Y_1^z)_{J_1} = y_1^{J_1} \wedge (Y_2^z)_{J_2} = y_2^{J_2} \;\middle|\; (Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2}\right] \\
&> 2^{-(1-\delta)\left[|J_1|\lambda + \log p^{\underline{|J_2|}}\right]}.
\end{aligned}$$

The set $I_1 \cup J_1$ and $I_2 \cup J_2$ now form subsets for which

$$\begin{aligned}
&\Pr\left[(Y_1^z)_{I_1 \cup J_1} = y_1^{I_1 \cup J_1} \wedge (Y_2^z)_{I_2 \cup J_2} = y_2^{I_2 \cup J_2}\right] \\
&= \Pr\left[(Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_1^z)_{J_1} = y_1^{J_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2} \wedge (Y_2^z)_{J_2} = y_2^{J_2}\right] \\
&= \Pr\left[(Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2}\right] \\
&\qquad \cdot \Pr\left[(Y_1^z)_{J_1} = y_1^{J_1} \wedge (Y_2^z)_{J_2} = y_2^{J_2} \;\middle|\; (Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2}\right] \\
&> 2^{-(1-\delta)\left[|I_1|\lambda + \log p^{\underline{|I_2|}}\right]} \cdot 2^{-(1-\delta)\left[|J_1|\lambda + \log p^{\underline{|J_2|}}\right]} \\
&= 2^{-(1-\delta)\left[|I_1 \cup J_1|\lambda + \log p^{\underline{|I_2 \cup J_2|}}\right]},
\end{aligned}$$

since I_i's and J_i's are disjoint for $i = 1, 2$. However, this contradicts the maximality of I_1 and I_2.

(2) Next, we claim that $|I_1|\lambda + |I_2|\log(p/e) \leq S_z/\delta$. Let $\mathcal{B}_{y_2^{I_2}}$ be the set of vectors $y_2^{\overline{I_2}}$ such that $y_2^{I_2}$ and $y_2^{\overline{I_2}}$ are a valid injection. Since $H_\infty(Y_1^z, Y_2^z) \geq 2^m\lambda + \log p! - S_z$, we observe that for any $y_1^{I_1}$ and $y_2^{I_2}$,

$$\begin{aligned}
&\Pr\left[(Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2}\right] \\
&= \sum_{\substack{y_1^{\overline{I_1}} \in \left(\{0,1\}^\lambda\right)^{2^m - |I_1|}, \\ y_2^{\overline{I_2}} \in \mathcal{B}_{y_2^{I_2}}}} \Pr\left[(Y_1^z)_{I_1} = y_1^{I_1} \wedge (Y_2^z)_{I_2} = y_2^{I_2} \wedge (Y_1^z)_{\overline{I_1}} = y_1^{\overline{I_1}} \wedge (Y_2^z)_{\overline{I_2}} = y_2^{\overline{I_2}}\right] \\
&\leq 2^{(2^m - |I_1|)\lambda} \cdot 2^{\log(p - |I_2|)!} \cdot 2^{-(2^m\lambda + \log p! - S_z)}
\end{aligned}$$

$= 2^{-(|I_1|\lambda + \log p^{\underline{|I_2|}} - S_z)}$,

and, hence,

$$H_\infty((Y_1^z)_{I_1}, (Y_2^z)_{I_2}) \geq |I_1|\lambda + \log p^{\underline{|I_2|}} - S_z. \tag{7}$$

On the other hand, because $((Y_1^z)_{I_1}, (Y_2^z)_{I_2})$ is not $(1-\delta)$-dense, we have

$$H_\infty((Y_1^z)_{I_1}, (Y_2^z)_{I_2}) < (1-\delta)\left[|I_1|\lambda + \log p^{\underline{|I_2|}}\right]. \tag{8}$$

Combining equation (7) and (8) together, we have

$$\begin{aligned} S_z &\geq \delta\left[|I_1|\lambda + \log p^{\underline{|I_2|}}\right] \\ &\geq \delta\left[|I_1|\lambda + |I_2|\log(p/e)\right], \end{aligned}$$

where the last inequality comes from [CDG18, Proposition 38][7]. Divided by $\delta > 0$, we get the result that $|I_1|\lambda + |I_2|\log(p/e) \leq S_z/\delta$.

Hence, $(\widetilde{Y}_1^z, \widetilde{Y}_2^z)$ is a $(P_1', P_2', 1-\delta)$-dense source such that $P_1'\lambda + P_2'\log(p/e) \leq S_z/\delta$. Set (Y_1^z, Y_2^z) now to be (Y_1^z, Y_2^z) conditioned on $((Y_1^z)_{I_1}, (Y_2^z)_{I_2}) \neq (y_1^{I_1}, y_2^{I_2})$ and recursively decompose (Y_1^z, Y_2^z) as long as

$$\Pr\left[X_1^z \in \mathrm{supp}(Y_1^z) \wedge X_2^z \in \mathrm{supp}(Y_2^z)\right] > \gamma.$$

Observe that $H_\infty(Y_1^z, Y_2^z) \geq 2^m\lambda + \log p! - (S_z + \log\gamma^{-1})$ at any point in this decomposition process since

$$\begin{aligned} \Pr\left[Y_1^z = y_1 \wedge Y_2^z = y_2\right] &= \Pr\left[X_1^z = y_1 \wedge X_2^z = y_2 | X_1^z \in \mathrm{supp}(Y_1^z) \wedge X_2^z \in \mathrm{supp}(Y_2^z)\right] \\ &\leq \frac{\Pr[X_1^z = y_1 \wedge X_2^z = y_2]}{\Pr[X_1^z \in \mathrm{supp}(Y_1^z) \wedge X_2^z \in \mathrm{supp}(Y_2^z)]} \\ &\leq \frac{2^{-(2^m\lambda + \log p! - S_z)}}{\gamma} \\ &= 2^{-(2^m\lambda + \log p! - (S_z + \log\gamma^{-1}))}. \end{aligned}$$

Note that $|\mathrm{supp}(Y_1^z)|$ and $|\mathrm{supp}(Y_2^z)|$ decreases in every step, and since $\mathrm{supp}(X_1^z)$ and $\mathrm{supp}(X_2^z)$ are finite, after finitely many steps, this process ends with $(Y_{1,\mathsf{final}}^z, Y_{2,\mathsf{final}}^z)$ with

$$\Pr\left[X_1^z \in \mathrm{supp}(Y_{1,\mathsf{final}}^z) \wedge X_2^z \in \mathrm{supp}(Y_{2,\mathsf{final}}^z)\right] \leq \gamma.$$

Hence, (X_1^z, X_2^z) is a convex combination of finitely many $(P_1', P_2', 1-\delta)$-dense sources and $(Y_{1,\mathsf{final}}^z, Y_{2,\mathsf{final}}^z)$, where P_1' and P_2' satisfy the inequality

$$P_1'\lambda + P_2'\log(p/e) \leq (S_z + \log\gamma^{-1})/\delta.$$

This implies that (X_1^z, X_2^z) is γ-close to a convex combination of $(P_1', P_2', 1-\delta)$-dense sources where $P_1'\lambda + P_2'\log(p/e) \leq (S_z + \log\gamma^{-1})/\delta$. □

[7] [CDG18, Proposition 38] says that $N^{\underline{j}} \geq (N/e)^j$, where $a^{\underline{b}} \coloneqq a!/(a-b)!$.

Let $(\widetilde{X}_1^z, \widetilde{X}_2^z)$ be the convex combination of $(P_1', P_2', 1-\delta)$-dense sources that is γ-close to (X_1^z, X_2^z) for a $\delta = \delta_z$ to be determined later. For every $(P_1', P_2', 1-\delta)$ source $(\widetilde{X}_1, \widetilde{X}_2)$ in said convex combination, let $(\widetilde{Y}_1, \widetilde{Y}_2)$ be the corresponding (P_1', P_2')-Bit-Fixing source, i.e., $(\widetilde{X}_1, \widetilde{X}_2)$ and $(\widetilde{Y}_1, \widetilde{Y}_2)$ are fixed on the same coordinates to the same values. The following claim bounds the distinguishing advantage between $(\widetilde{X}_1, \widetilde{X}_2)$ and $(\widetilde{Y}_1, \widetilde{Y}_2)$ for any (T_1, T_2)-query distinguisher.

Claim A.2. For any $(P_1', P_2', 1-\delta)$-dense source $(\widetilde{X}_1, \widetilde{X}_2)$ and its corresponding (P_1', P_2')-Bit-Fixing source $(\widetilde{Y}_1, \widetilde{Y}_2)$, it holds that for any (adaptive) distinguisher $\mathcal{D}$ that queries at most T_1 coordinates of the random oracle H and makes at most T_2 forward or inverse queries to the injection map τ,

$$\left|\Pr\left[\mathcal{D}^{\widetilde{X}_1,\widetilde{X}_2} = 1\right] - \Pr\left[\mathcal{D}^{\widetilde{Y}_1,\widetilde{Y}_2} = 1\right]\right| \leq \delta(T_1\lambda + T_2 \log p),$$

and

$$\Pr\left[\mathcal{D}^{\widetilde{X}_1,\widetilde{X}_2} = 1\right] \leq 2^{\delta(T_1\lambda + T_2 \log p)} \cdot \Pr\left[\mathcal{D}^{\widetilde{Y}_1,\widetilde{Y}_2} = 1\right].$$

Proof of Claim A.2. Without loss of generality, assume that $\mathcal{D}$ is deterministic and does not query any of the fixed positions, make the same query twice, or make an inverse query after making the corresponding forward query or vice-versa. Let $T_{\widetilde{X}_1,\widetilde{X}_2}$ and $T_{\widetilde{Y}_1,\widetilde{Y}_2}$ be the random variables corresponding to the transcripts containing the query/answer pairs resulting from $\mathcal{D}$'s interaction with $(\widetilde{X}_1, \widetilde{X}_2)$ and $(\widetilde{Y}_1, \widetilde{Y}_2)$, respectively.

For a fixed transcript t, denote by $\mathsf{p}_{\widetilde{X}_1,\widetilde{X}_2}(t)$ and $\mathsf{p}_{\widetilde{Y}_1,\widetilde{Y}_2}(t)$ the probabilities that $(\widetilde{X}_1, \widetilde{X}_2)$ and $(\widetilde{Y}_1, \widetilde{Y}_2)$, respectively, produce the answers in t if the queries in t are asked. Since $\mathcal{D}$ is deterministic, $\Pr\left[T_{\widetilde{X}_1,\widetilde{X}_2} = t\right] \in \left\{0, \mathsf{p}_{\widetilde{X}_1,\widetilde{X}_2}(t)\right\}$, and similarly, $\Pr\left[T_{\widetilde{Y}_1,\widetilde{Y}_2} = t\right] \in \left\{0, \mathsf{p}_{\widetilde{Y}_1,\widetilde{Y}_2}(t)\right\}$. Denote by $\mathcal{T}_{X_1,X_2}$ the set of all transcripts t for which $\Pr\left[T_{\widetilde{X}_1,\widetilde{X}_2} = t\right] > 0$. For such t, $\Pr\left[T_{\widetilde{X}_1,\widetilde{X}_2} = t\right] = \mathsf{p}_{\widetilde{X}_1,\widetilde{X}_2}(t)$ and also $\Pr\left[T_{\widetilde{Y}_1,\widetilde{Y}_2} = t\right] = \mathsf{p}_{\widetilde{Y}_1,\widetilde{Y}_2}(t)$. Observe that for every transcript t,

$$\mathsf{p}_{\widetilde{X}_1,\widetilde{X}_2}(t) \leq 2^{-(1-\delta)\left[T_1\lambda + \log(p-P_2')\frac{T_2}{}\right]} \text{ and } \mathsf{p}_{\widetilde{Y}_1,\widetilde{Y}_2}(t) \leq 2^{-\left[T_1\lambda + \log(p-P_2')\frac{T_2}{}\right]}, \tag{9}$$

as $(\widetilde{X}_1, \widetilde{X}_2)$ is $(P_1', P_2', 1-\delta)$-dense and $(\widetilde{Y}_1, \widetilde{Y}_2)$ is (P_1', P_2')-fixed.

Towards proving the first part of the claim, observe that $\mathcal{D}$'s output can be computed from the transcript (including whether a query was a forward for an inverse query) by just running $\mathcal{D}$ and providing the answers to its queries from the transcript. Hence,

$$\begin{aligned}
\left|\Pr\left[\mathcal{D}^{\widetilde{X}_1,\widetilde{X}_2} = 1\right] - \Pr\left[\mathcal{D}^{\widetilde{Y}_1,\widetilde{Y}_2} = 1\right]\right| &\leq \mathsf{SD}\left(T_{\widetilde{X}_1,\widetilde{X}_2}, T_{\widetilde{Y}_1,\widetilde{Y}_2}\right) \\
&= \sum_t \max\left\{0, \Pr\left[T_{\widetilde{X}_1,\widetilde{X}_2} = t\right] - \Pr\left[T_{\widetilde{Y}_1,\widetilde{Y}_2} = t\right]\right\} \\
&= \sum_{t \in \mathcal{T}_{X_1,X_2}} \max\left\{0, \mathsf{p}_{\widetilde{X}_1,\widetilde{X}_2}(t) - \mathsf{p}_{\widetilde{Y}_1,\widetilde{Y}_2}(t)\right\} \\
&= \sum_{t \in \mathcal{T}_{X_1,X_2}} \mathsf{p}_{\widetilde{X}_1,\widetilde{X}_2}(t) \cdot \max\left\{0, 1 - \frac{\mathsf{p}_{\widetilde{Y}_1,\widetilde{Y}_2}(t)}{\mathsf{p}_{\widetilde{X}_1,\widetilde{X}_2}(t)}\right\} \\
&\leq 1 - 2^{-\delta\left[T_1\lambda + \log(p-P_2')\frac{T_2}{}\right]}
\end{aligned}$$

$$\leq 1 - 2^{-\delta(T_1\lambda + T_2 \log p)} \leq \delta(T_1\lambda + T_2 \log p),$$

where the first sum is over all possible transcripts and where the last inequality uses $2^{-x} \geq 1 - x$ for $x \geq 0$ and $a^{\underline{b}} \leq a^b$ for $a, b \in \mathbb{N}$.

As for the second part of the claim, observe that due to the equation (9) and the support of $T_{\widetilde{X}_1,\widetilde{X}_2}$ being a subset of $T_{\widetilde{Y}_1,\widetilde{Y}_2}$,

$$\begin{aligned}\Pr\left[T_{\widetilde{X}_1,\widetilde{X}_2} = t\right] &\leq 2^{\delta\left[T_1\lambda + \log(p-P_2')^{\underline{T_2}}\right]} \cdot \Pr\left[T_{\widetilde{Y}_1,\widetilde{Y}_2} = t\right] \\ &\leq 2^{\delta(T_1\lambda + T_2\log p)} \cdot \Pr\left[T_{\widetilde{Y}_1,\widetilde{Y}_2} = t\right],\end{aligned}$$

for any transcript t. Let $\mathcal{T}_{\mathcal{D}}$ be the set of transcripts where $\mathcal{D}$ outputs 1. Then we have

$$\begin{aligned}\Pr\left[\mathcal{D}^{\widetilde{X}_1,\widetilde{X}_2} = 1\right] &= \sum_{t\in\mathcal{T}_{\mathcal{D}}} \Pr\left[T_{\widetilde{X}_1,\widetilde{X}_2} = t\right] \\ &\leq 2^{\delta(T_1\lambda + T_2\log p)} \cdot \sum_{t\in\mathcal{T}_{\mathcal{D}}} \Pr\left[T_{\widetilde{Y}_1,\widetilde{Y}_2} = t\right] \\ &= 2^{\delta(T_1\lambda + T_2\log p)} \cdot \Pr\left[\mathcal{D}^{\widetilde{Y}_1,\widetilde{Y}_2} = 1\right],\end{aligned}$$

which concludes the proof. □

Let $(\widetilde{Y}_1^z, \widetilde{Y}_2^z)$ be obtained by replacing $(\widetilde{X}_1, \widetilde{X}_2)$ by $(\widetilde{Y}_1, \widetilde{Y}_2)$ in $(\widetilde{X}_1^z, \widetilde{X}_2^z)$. Setting

$$\delta_z = \frac{S_z + \log\gamma^{-1}}{P_1 k + P_2 \log(p/e)},$$

Claim A.1 and Claim A.2 imply

$$\begin{aligned}&\left|\Pr\left[\mathcal{D}^{X_1^z,X_2^z}(z) = 1\right] - \Pr\left[\mathcal{D}^{\widetilde{Y}_1^z,\widetilde{Y}_2^z}(z) = 1\right]\right| \\ &\leq \frac{(S_z + \log\gamma^{-1})(T_1\lambda + T_2\log p)}{(P_1\lambda + P_2\log(p/e))} + \gamma,\end{aligned} \tag{10}$$

as well as

$$\begin{aligned}&\Pr\left[\mathcal{D}^{X_1^z,X_2^z}(z) = 1\right] \\ &\leq 2^{\frac{(S_z+\log\gamma^{-1})(T_1\lambda+T_2\log p)}{(P_1\lambda+P_2\log(p/e))}} \cdot \Pr\left[\mathcal{D}^{\widetilde{Y}_1^z,\widetilde{Y}_2^z}(z) = 1\right] + \gamma.\end{aligned} \tag{11}$$

Moreover, note that for the above choice of δ_z, we have $P_1' = P_1$ and $P_2' = P_2$, i.e., the sources $(\widetilde{Y}_1^z, \widetilde{Y}_2^z)$ are (P_1, P_2)-fixed, as desired.

Claim A.3. $\mathbb{E}_z[S_z] \leq S$ and $\Pr[S_{f(X_1,X_2)} > S + \log\gamma^{-1}] \leq \gamma$.

Proof of Claim A.3. Observe that $H_\infty(X_1^z, X_2^z) = H_\infty((X_1, X_2)|Z = z) = H((X_1, X_2)|Z = z)$ since, conditioned on $Z = z$, (X_1, X_2) is distributed uniformly over all values (x_1, x_2) with $f(x_1, x_2) = z$. Hence,

$$\mathbb{E}_z[S_z] = 2^m\lambda + \log p! - \mathbb{E}_z\left[H_\infty((X_1, X_2)|Z = z)\right]$$

$$= 2^m\lambda + \log p! - \mathbb{E}_z\left[H((X_1,X_2)|Z=z)\right]$$
$$= 2^m\lambda + \log p! - H((X_1,X_2)|Z=z) \le S.$$

Again, due to the uniformity of (X_1, X_2), $\Pr[f(X_1,X_2)=z] = 2^{-S_z}$. Hence,

$$\Pr[S_{f(X_1,X_2)} > S + \log\gamma^{-1}] = \sum_{z\in\{0,1\}^S : S_z > S+\log\gamma^{-1}} \Pr[f(X_1,X_2)=z]$$
$$\le 2^S \cdot 2^{-(S+\log\gamma^{-1})} \le \gamma. \qquad \square$$

Now the first part of the lemma follows (using $(Y_1^z, Y_2^z) \coloneqq (\widetilde{Y}_1^z, \widetilde{Y}_2^z)$) by taking expectations over z of the equation (10) and applying the first part of Claim A.3. The second part of the lemma is proved as follows:

$$\begin{aligned}
&\Pr\left[\mathcal{D}^{X_1,X_2}(f(X_1,X_2))=1\right] \\
&\le \Pr\left[\mathcal{D}^{X_1,X_2}(f(X_1,X_2))=1, S_{f(X_1,X_2)} \le S+\log\gamma^{-1}\right] \\
&\qquad + \Pr\left[S_{f(X_1,X_2)} > S+\log\gamma^{-1}\right] \\
&\le \Big(2^{(S+\log\gamma^{-1})(T_1\lambda+T_2\log p)/W} \cdot \Pr\Big[\mathcal{D}^{X_1,X_2}(f(X_1,X_2))=1, \\
&\qquad S_{f(X_1,X_2)} \le S+\log\gamma^{-1}\Big] + \gamma\Big) + \gamma \\
&\le 2^{(S+\log\gamma^{-1})(T_1\lambda+T_2\log p)/W} \cdot \Pr\Big[\mathcal{D}^{X_1,X_2}(f(X_1,X_2))=1, \\
&\qquad S_{f(X_1,X_2)} \le S+\log\gamma^{-1}\Big] + 2\gamma,
\end{aligned}$$

where the second inequality follows by taking expectations over z of the equation (11) (together with the condition $S_z \le S + \log\gamma^{-1}$) and the second part of Claim A.3. $\square$

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

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 in this work as follows:

- **Random Oracle + Generic Group Oracle** $\mathsf{RO{+}GG}(m,\lambda,p,\ell)$: Samples a random oracle function table $\mathsf{H} \leftarrow \mathcal{H}_{m,\lambda}$, where $\mathcal{H}_{m,\lambda}$ is the set of all functions from $\{0,1\}^m \to \{0,1\}^\lambda$, and samples a random injection $\tau \leftarrow \mathcal{I}_{p,\ell}$, where $\mathcal{I}_{p,\ell}$ is the set of all injections from $\mathbb{Z}_p \to \mathbb{G}$ where $\mathbb{G}$ is the set of bitstrings of length $\ell \ge \log p$; offers no functionality at $(\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$; answers queries $x \in \{0,1\}^m$ to H at $\mathcal{O}_1.\mathsf{on}$ by the corresponding value $\mathsf{H}(x) \in \{0,1\}^\lambda$; answers *forward* queries $x' \in \mathbb{Z}_p$ to τ at $\mathcal{O}_2.\mathsf{on}$ by the corresponding value $\tau(x') \in \mathbb{G}$; answers *group-operation* queries (s,s') at $\mathcal{O}_2.\mathsf{on}$ as follows: if $s = \tau(x)$ and $s' = \tau(x')$ for some x, x', the oracle replies by $\tau(x+x')$ and by $\perp$ otherwise; answers *inverse* queries s at $\mathcal{O}_2.\mathsf{on}$ by returning $\tau^{-1}(s)$ if s is in the range of τ and $\perp$ otherwise.
- **Auxiliary-Input Random Oracle + Generic Group Oracle** $\mathsf{AI{-}RO{+}GG}(m,\lambda,p,\ell)$: Samples a random oracle function table $\mathsf{H} \leftarrow \mathcal{H}_{m,\lambda}$ and samples a random injection $\tau \leftarrow \mathcal{I}_{p,\ell}$ as explained in $\mathsf{RO{+}GG}(m,\lambda,p,\ell)$; 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 $\mathsf{RO{+}GG}(m,\lambda,p,\ell)$. When the parameters (m,λ,p,ℓ) are clear in context, we sometimes abuse the notation and simply say $\mathsf{AI{-}RO{+}GG}$.

– **Bit-Fixing Random Oracle + Generic Group Oracle** $\mathsf{BF\text{-}RO{+}GG}(P_1, P_2, m, \lambda, p, \ell)$: Samples a random oracle function table $\mathsf{H} \leftarrow \mathcal{H}_{m,\lambda}$ at $\mathcal{O}_1.\mathsf{pre}$ and samples a random size-p subset $\mathcal{Y}$ of $\mathbb{G}$ and outputs $\mathcal{Y}$ at $\mathcal{O}_2.\mathsf{pre}$; 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 $\mathcal{Y}$; samples a random injection $\tau \leftarrow \mathcal{I}_{p,\ell}$ with range $\mathcal{Y}$ and consistent with said list; $(\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$ behaves the same as $\mathsf{RO{+}GG}(m, \lambda, p, \ell)$. When the parameters (m, λ, p, ℓ) are clear in context, we sometimes abuse the notation and simply say $\mathsf{BF\text{-}RO{+}GG}(P_1, P_2)$.

Attackers with Oracle-Dependent Advice. We define the attackers $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$ similar to that of prior work [CDGS18, CDG18], which consist of a preprocessing attacker $\mathcal{A}_{\mathsf{pre}}$ and an online attacker $\mathcal{A}_{\mathsf{on}}$, which carries out the actual attack using the output of $\mathcal{A}_{\mathsf{pre}}$. The difference is that we consider a pair of oracles instead of a single oracle. More precisely, in the presence of an oracle pair $(\mathcal{O}_1, \mathcal{O}_2)$, $\mathcal{A}_{\mathsf{pre}}$ interacts with $(\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$ and $\mathcal{A}_{\mathsf{on}}$ with $(\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$.

Definition 4. *An* (S, 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 an* S*-bit string as hint, and*
– $\mathcal{A}_{\mathsf{on}}$*, which takes an* S*-bit auxiliary input and makes at most* T_1 *queries to* $\mathcal{O}_1.\mathsf{on}$ *and* T_2 *queries to* $\mathcal{O}_2.\mathsf{on}$*.*

For an arbitrary oracle $\mathcal{O}$, Coretti et al. [CDGS18, CDG18] defined an *application* G in the $\mathcal{O}$-model by specifying a challenger C that has access to $\mathcal{O}.\mathsf{on}$, interacts with the online attacker $\mathcal{A}_{\mathsf{on}}$ of an attacker $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$, and outputs a bit at the end of the interaction. Then they defined the *success* of $\mathcal{A}$ on G in the $\mathcal{O}$-model as follows:

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

where $\mathcal{A}_{\mathsf{on}}^{\mathcal{O}.\mathsf{on}}\left(\mathcal{A}_{\mathsf{pre}}^{\mathcal{O}.\mathsf{pre}}\right) \leftrightarrow \mathsf{C}^{\mathcal{O}.\mathsf{on}}$ denotes the bit output by the challenger C after its interaction with $\mathcal{A}$. It can be extended to the oracle pair setting in the same manner when $\mathcal{O} = (\mathcal{O}_1, \mathcal{O}_2)$, by defining $\mathcal{O}.\mathsf{pre} \coloneqq (\mathcal{O}_1.\mathsf{pre}, \mathcal{O}_2.\mathsf{pre})$ and $\mathcal{O}.\mathsf{on} \coloneqq (\mathcal{O}_1.\mathsf{on}, \mathcal{O}_2.\mathsf{on})$.

Definition 5. *For an* indistinguishability *application* G *in the* $(\mathcal{O}_1, \mathcal{O}_2)$*-model, the* advantage *of an attacker* $\mathcal{A}$ *is defined as*

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

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

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

Additive Error for Arbitrary Applications in the ROM+GGM. Using Lemma 4, one can extend [CDGS18, Theorem 5] and [CDG18, Theorem 1] to the translation from the Bit-Fixing model with multiple idealized models (ROM+GGM) to the corresponding Auxiliary-Input model at the cost of an additive term which is similar to those from [CDGS18, Theorem 5] and [CDG18, Theorem 1].

Reminder of Theorem 4. *For every $\gamma > 0$, if an application G is $(S, T_1, T_2, \varepsilon')$-secure in $\mathsf{BF\text{-}RO\text{+}GG}(P_1, P_2, m, \lambda, p, \ell)$ for any $P_1, P_2 \in \mathbb{N}$ such that $P_1\lambda + P_2 \log(p/e) \leq \eta$, then it is $(S, T_1, T_2, \varepsilon)$-secure in $\mathsf{AI\text{-}RO\text{+}GG}(m, \lambda, p, \ell)$, for*

$$\varepsilon \leq \varepsilon' + \frac{2(S + \log \gamma^{-1}) \left((T_1)_G^{\mathsf{comb}} \lambda + 3(T_2)_G^{\mathsf{comb}} \log p\right)}{\eta} + 2\gamma,$$

where e is the Euler's number.

Proof. Fix P_1, P_2 as well as γ. Let G be an arbitrary application in the $(\mathsf{BF\text{-}RO}, \mathsf{BF\text{-}GG})$-model (or in the $(\mathsf{AI\text{-}RO}, \mathsf{AI\text{-}GG})$-model) and C be the corresponding challenger. Moreover, fix an (S, T_1, T_2)-attacker $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$, and let $\{(Y_1, Y_2)^{z,\mathcal{Y}}\}$, indexed by $z \in \{0,1\}^S$ and size-p subsets $\mathcal{Y} \subseteq \mathbb{G}$, be the family of distributions guaranteed to exist by Lemma 4. Consider the following (S, T_1, T_2)-attacker $\mathcal{A}' = (\mathcal{A}'_{\mathsf{pre}}, \mathcal{A}'_{\mathsf{on}})$ (expecting to interact with $\mathsf{BF\text{-}RO\text{+}GG}$):

- $\mathcal{A}'_{\mathsf{pre}}$ obtains the set $\mathcal{Y}$ from $\mathsf{BF\text{-}GG.pre}$ and internally simulates $\mathcal{A}_{\mathsf{pre}}$ on a uniformly random input $X_1 \in \{0,1\}^m$ and a uniformly random injection $X_2 \in \mathbb{Z}_p$ with range $\mathcal{Y}$ to obtain $z \leftarrow \mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO\text{+}GG.pre}}$. Then, it samples one of the (P_1, P_2)-bit-fixing sources (Y'_1, Y'_2) making up $(Y_1, Y_2)^{z,\mathcal{Y}}$ and presets $\mathsf{BF\text{-}RO\text{+}GG}$ 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-operation, and inverse 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 group-operation queries (s, s') (to the generic group oracle) by making two backward queries to its own oracle for s and s', obtaining i and j, respectively, making a forward query $i + j$, and passing the answer to $\mathcal{A}_{\mathsf{on}}$ or C (unless one of the answers to the backward queries was $\perp$, in which case $\perp$ is returned).

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

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

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

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

Furthermore, $\mathcal{D}$ is a distinguisher taking an S-bit input and making at most $(T_1)_G^{\mathsf{comb}}$ to its random oracle and $3(T_2)_G^{\mathsf{comb}}$ to its generic group oracle. Therefore, by Lemma 4,

$$\begin{aligned}\mathrm{Succ}_{G,\mathsf{AI\text{-}RO\text{+}GG}}(\mathcal{A}) \leq\ & \mathrm{Succ}_{G,\mathsf{BF\text{-}RO\text{+}GG}}(\mathcal{A}') \\ & + \frac{(S + \log \gamma^{-1}) \left((T_1)_G^{\mathsf{comb}} \lambda + 3(T_2)_G^{\mathsf{comb}} \log p\right)}{\eta} + \gamma,\end{aligned}$$

for any P_1, P_2 satisfying $P_1\lambda + P_2 \log(p/e)$. Since there is only an additive term between the two success probabilities, the above inequality implies

$$\mathrm{Adv}_{G,\mathsf{AI\text{-}RO+GG}}(\mathcal{A}) \le \mathrm{Adv}_{G,\mathsf{BF\text{-}RO+GG}}(\mathcal{A}') + \frac{2(S+\log\gamma^{-1})\left((T_1)_G^{\mathsf{comb}}\lambda + 3(T_2)_G^{\mathsf{comb}}\log p\right)}{\eta} + 2\gamma,$$

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

Reminder of Theorem 5. *For every $\gamma > 0$, if an* unpredictability *application G is $(S, T_1, T_2, \varepsilon')$-secure in $\mathsf{BF\text{-}RO+GG}(P_1, P_2, m, \lambda, p, \ell)$ for any $P_1, P_2 \in \mathbb{N}$ satisfying $(S+\log\gamma^{-1}) \cdot \left((T_1)_G^{\mathsf{comb}}\lambda + 3(T_2)_G^{\mathsf{comb}}\log p\right) \le P_1\lambda + P_2\log(p/e)$, then it is $(S, T_1, T_2, \varepsilon)$-secure in $\mathsf{AI\text{-}RO+GG}(m, \lambda, p, \ell)$, for $\varepsilon \le 2\varepsilon' + 2\gamma$, where e is the Euler's number.*

Proof. Using the same attacker $\mathcal{A}'$ as in the proof of Theorem 4 and applying the second part of Lemma 4, for any $P_1, P_2 \in \mathbb{N}$ such that $(S+\log\gamma^{-1})\left((T_1)_G^{\mathsf{comb}}\lambda + 3(T_2)_G^{\mathsf{comb}}\log p\right) \le P_1\lambda + P_2\log(p/e) \le \eta$, we have

$$\begin{aligned}
&\mathrm{Succ}_{G,\mathsf{AI\text{-}RO+GG}}(\mathcal{A}) \\
&\le 2^{\frac{(S+\log\gamma^{-1})\left((T_1)_G^{\mathsf{comb}}\lambda + 3(T_2)_G^{\mathsf{comb}}\log p\right)}{\eta}} \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GG}}(\mathcal{A}') + 2\gamma \\
&\le 2 \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GG}}(\mathcal{A}') + 2\gamma,
\end{aligned}$$

which translates into

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

for unpredictability applications. □

B Bit-Fixing/Auxiliary-Input Multiple Idealized Models

In Section 3, we studied the Bit-Fixing to Auxiliary-Input transition with two idealized models at once — the Random Oracle Model plus the Generic Group Model. We can further extend this approach and consider multiple idealized models, including the Ideal Cipher Model and the Random Permutation Model as well.

B.1 Replacing Auxiliary Information by Bit-Fixing

Definition 6. *A* (n_1, n_2, n_3)-source *is a tuple of vector random variables $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$, where*

- *$\mathbf{X} = (X_1, \ldots, X_{n_1})$ is an n_1-dimensional vector where for each $i \in [n_1]$, an element X_i is a random variable that corresponds to the function table of a random oracle $\mathsf{H}_i : [M_i] \to [N_i]$ with range $[N_i]^{M_i}$,*
- *$\mathbf{Y} = (Y_1, \ldots, Y_{n_2})$ is an n_2-dimensional vector where for each $i \in [n_2]$, an element Y_i is a random variable that corresponds to the function table of an injection $\tau_i : \mathbb{Z}_{p_i} \to \mathbb{G}_i$, where $\mathbb{G}_i$ is the set of bitstrings of length ℓ_i (with $\ell_i \ge \log p_i$), and*
- *$\mathbf{Z} = (Z_1, \ldots, Z_{n_3})$ is an n_3-dimensional vector where for each $i \in [n_3]$, an element Z_i is a random variable that corresponds to the function table of a cipher $F_i : [K_i] \times [C_i] \to [C_i]$.*

Further, a (n_1, n_2, n_3)-source is called:

– $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, 1-\delta)$-dense *if for vectors* $\mathbf{P} = (P_1, \ldots, P_{n_1}) \in \mathbb{Z}_{M_1} \times \cdots \times \mathbb{Z}_{M_{n_1}}$, $\mathbf{Q} = (Q_1, \ldots, Q_{n_2}) \in \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_{n_1}}$, *and* $\mathbf{R} = (\mathbf{R}_1, \ldots, \mathbf{R}_{n_3}) \in (\mathbb{Z}_{C_1})^{K_1} \times \cdots \times (\mathbb{Z}_{C_{n_3}})^{K_{n_3}}$ *where* $\mathbf{R}_k = (R_{k,1}, \ldots, R_{k,K_k})$ *for each* $k \in [n_3]$*, if* H_i *is fixed on at most* P_i *coordinates for each* $i \in [n_1]$*,* τ_j *is fixed on at most* Q_j *coordinates for each* $j \in [n_2]$*, and for each cipher* F_k *with* $k \in [n_3]$*, it is fixed on at most* $R_{k,s}$ *coordinates for each* $s \in [K_k]$*, and if for all tuples of vectors of subsets* $(\mathbf{U}, \mathbf{V}, \mathbf{W})$ *of non-fixed coordinates where* $\mathbf{U} = (U_1, \ldots, U_{n_1}) \subseteq \mathbb{Z}_{M_1} \times \cdots \times \mathbb{Z}_{M_{n_1}}$, $\mathbf{V} = (V_1, \ldots, V_{n_2}) \subseteq \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_{n_1}}$, *and* $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{n_3}) \subseteq (\mathbb{Z}_{C_1})^{K_1} \times \cdots \times (\mathbb{Z}_{C_{n_3}})^{K_{n_3}}$ *where* $\mathbf{W}_k = (W_{k,1}, \ldots, W_{k,K_k})$ *for each* $k \in [n_3]$*,*

$$H_\infty(\mathbf{X}_\mathbf{U}, \mathbf{Y}_\mathbf{V}, \mathbf{Z}_\mathbf{W}) \geq (1-\delta)\left[\sum_{i=1}^{n_1} |U_i| \log N_i + \sum_{j=1}^{n_2} \log(p_j - Q_j)^{\underline{|V_j|}} + \sum_{k=1}^{n_3}\sum_{s=1}^{K_k} \log(C_k - R_{k,s})^{\underline{|W_{k,s}|}}\right],$$

where $a^{\underline{b}} \coloneqq a!/(a-b)!$ *and* $\mathbf{X}_\mathbf{U}$ *is* $\mathbf{X}$ *restricted to the coordinates in* $\mathbf{U}$*,*

– $(1-\delta)$-dense *if it is* $(\mathbf{0}, \mathbf{0}, \mathbf{0}, 1-\delta)$*-dense, and*

– $(\mathbf{P}, \mathbf{Q}, \mathbf{R})$-bit-fixing *if it is* $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, 1)$*-dense, i.e.,* $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, 1-\delta)$*-dense with* $\delta = 0$*.*

In Definition 6, we remark that $\mathbf{X}$ captures a source with multiple random oracles, $\mathbf{Y}$ captures a source with multiple generic groups, and $\mathbf{Z}$ captures a source with multiple ideal ciphers. In the case of $\mathbf{Z}$, if $K_i = 1$ for some $i \in [n_3]$, it can be considered as a random permutation. Hence, Definition 6 captures the usage of multiple idealized models at once, including the random oracle model, generic group model, ideal cipher model, and random permutation model.

Lemma 5. *Let* $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ *be a tuple of vector random variables which satisfies the following.*

– $\mathbf{X} = (X_1, \ldots, X_{n_1})$ *is an* n_1*-dimensional vector where* X_i *is distributed uniformly over* $[N_i]^{M_i}$ *for each* $i \in [n_1]$*,*

– $\mathbf{Y} = (Y_1, \ldots, Y_{n_2})$ *is an* n_2*-dimensional vector where* Y_i *is a uniformly random injection that takes on as value function tables corresponding to an injection* $\tau_i : \mathbb{Z}_{p_i} \to \mathbb{G}_i$ *for each* $i \in [n_2]$*, and*

– $\mathbf{Z} = (Z_1, \ldots, Z_{n_3})$ *is an* n_3*-dimensional vector where* Z_i *takes on as value function tables corresponding to a cipher* $F_i : [K_i] \times [C_i] \to [C_i]$ *for each* $i \in [n_3]$*.*

Let $W = f(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$*, where*

$$f : \Big(\prod_{i=1}^{n_1} [M_i]\Big) \times \Big(\prod_{j=1}^{n_2} \mathbb{Z}_{p_j}\Big) \times \Big(\prod_{k=1}^{n_3} [K_k] \times [C_k]\Big) \to \{0,1\}^S$$

is an arbitrary function. For any $\gamma > 0$ *and* $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ *that were defined as in Definition 6, there exists a family* $\{(\mathbf{X}', \mathbf{Y}', \mathbf{Z}')\}_{W,\mathcal{Y}}$*, indexed by* $W \in \{0,1\}^S$ *and* $\mathcal{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_{n_2})$ *where* $\mathcal{Y}_i$ *is a size-*p_i *subset of* $\mathbb{G}_i$ *for each* $i \in [n_2]$*, of convex combinations of* $(\mathbf{P}, \mathbf{Q}, \mathbf{R})$*-bit-fixing* (n_1, n_2, n_3)*-sources satisfying*

$$\sum_{i=1}^{n_1} P_i \log N_i + \sum_{i=1}^{n_2} Q_i \log(p_i/e) + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} R_{i,j} \log(C_i/e) \leq \eta,$$

such that for any distinguisher $\mathcal{D}$ taking an S-bit input and querying at most $\mathbf{S} = (S_1, \ldots, S_{n_1})$ coordinates of each oracle that corresponds to $\mathbf{X}$ where $S_i < P_i$ for each $i \in [n_1]$, $\mathbf{T} = (T_1, \ldots, T_{n_2})$ coordinates of each oracle that corresponds to $\mathbf{Y}$ where $T_i < Q_i$ for each $i \in [n_2]$, and $\mathbf{L} = (\mathbf{L}_1, \ldots, \mathbf{L}_{n_3})$ coordinates of each oracle that corresponds to $\mathbf{Z}$ where $\mathbf{L}_k = (L_{k,1}, \ldots, L_{k,K_k})$ and $L_{i,j} < R_{i,j}$ for each $j \in [K_i]$ and $i \in [n_3]$, we have (here, e is the Euler's number)

$$\begin{aligned}&\left|\Pr\left[\mathcal{D}^{\mathbf{X},\mathbf{Y},\mathbf{Z}}(f(\mathbf{X},\mathbf{Y},\mathbf{Z})) = 1\right] - \Pr\left[\mathcal{D}^{(\mathbf{X}',\mathbf{Y}',\mathbf{Z}')_{f(\mathbf{X},\mathbf{Y},\mathbf{Z}),\mathsf{im}(\mathbf{Y})}}(f(\mathbf{X},\mathbf{Y},\mathbf{Z})) = 1\right]\right| \\ &\leq \frac{\left(S + \log\gamma^{-1}\right)\left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j}\log C_i\right)}{\eta} + \gamma,\end{aligned}$$

and

$$\begin{aligned}&\Pr\left[\mathcal{D}^{\mathbf{X},\mathbf{Y},\mathbf{Z}}(f(\mathbf{X},\mathbf{Y},\mathbf{Z})) = 1\right] \\ &\leq 2^{\frac{\left(S+\log\gamma^{-1}\right)\left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j}\log C_i\right)}{\eta}} \cdot \Pr\left[\mathcal{D}^{(\mathbf{X}',\mathbf{Y}',\mathbf{Z}')_{f(\mathbf{X},\mathbf{Y},\mathbf{Z}),\mathsf{im}(\mathbf{Y})}}(f(\mathbf{X},\mathbf{Y},\mathbf{Z})) = 1\right] + 2\gamma.\end{aligned}$$

Proof. Fix an arbitrary $w \in \{0,1\}^S$ and let $(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$ be the distribution of $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ conditioned on $f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = w$. Let $S_z = \sum_{i=1}^{n_1} M_i \log N_i + \sum_{i=1}^{n_2} \log p_i! + \sum_{i=1}^{n_3} K_i \log C_i! - H_\infty(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ be the min-entropy deficiency of $(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$. Let $\gamma > 0$ be arbitrary.

Claim B.1. For every $\delta > 0$, any leaky source $(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$ is γ-close to a source $(\mathbf{X'}^w, \mathbf{Y'}^w, \mathbf{Z'}^w)$ which is a convex combination of $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense sources for

$$\sum_{i=1}^{n_1} P'_i \log N_i + \sum_{i=1}^{n_2} Q'_i \log(p_i/e) + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} R'_{i,j}\log(C_i/e) \leq \frac{S_z + \log\gamma^{-1}}{\delta}.$$

Proof of Claim B.1. Without loss of generality, we can assume that $(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$ is not $(1-\delta)$-dense since if it is $(1-\delta)$-dense then we can trivially find such $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense sources. Let $(\mathbf{X'}^w, \mathbf{Y'}^w, \mathbf{Z'}^w) \coloneqq (\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$ and $\mathbf{U} = (U_1, \ldots, U_{n_1}) \in \mathbb{Z}_{M_1} \times \cdots \times \mathbb{Z}_{M_{n_1}}$, $\mathbf{V} = (V_1, \ldots, V_{n_2}) \in \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_{n_1}}$, and $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{n_3}) \in (\mathbb{Z}_{C_1})^{K_1} \times \cdots \times (\mathbb{Z}_{C_{n_3}})^{K_{n_3}}$ (where $\mathbf{W}_k = (W_{k,1}, \ldots, W_{k,K_k})$ for each $k \in [n_3]$) be the largest subset for which there exists a violation, i.e., there exists $\mathbf{x'}_\mathbf{U}, \mathbf{y'}_\mathbf{V}, \mathbf{z'}_\mathbf{W}$ such that

$$\begin{aligned}&\Pr\left[(\mathbf{X'}^w)_\mathbf{U} = \mathbf{x'}_\mathbf{U} \wedge (\mathbf{Y'}^w)_\mathbf{V} = \mathbf{y'}_\mathbf{V} \wedge (\mathbf{Z'}^w)_\mathbf{W} = \mathbf{z'}_\mathbf{W}\right] \\ &> 2^{-(1-\delta)\left[\sum_{i=1}^{n_1}|U_i|\log N_i + \sum_{j=1}^{n_2}\log p_j^{\frac{|V_j|}{}} + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}\log C_k^{\frac{|W_{k,j}|}{}}\right]}.\end{aligned}$$

Let $(\widetilde{\mathbf{X}'}^w, \widetilde{\mathbf{Y}'}^w, \widetilde{\mathbf{Z}'}^w)$ be the distribution of $(\mathbf{X'}^w, \mathbf{Y'}^w, \mathbf{Z'}^w)$ conditioned on the event $(\mathbf{X'}^w)_\mathbf{U} = \mathbf{x'}_\mathbf{U} \wedge (\mathbf{Y'}^w)_\mathbf{V} = \mathbf{y'}_\mathbf{V} \wedge (\mathbf{Z'}^w)_\mathbf{W} = \mathbf{z'}_\mathbf{W}$.

(1) Now we claim that $(\widetilde{\mathbf{X}'}^w, \widetilde{\mathbf{Y}'}^w, \widetilde{\mathbf{Z}'}^w)$ is $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense with $P'_i = |U_i|$ for $i \in [n_1]$, $Q'_j = |V_j|$ for $j \in [n_2]$, and $R'_{k,j} = |W_{k,j}|$ for $k \in [n_3], j \in [K_k]$. Suppose that $(\widetilde{\mathbf{X}'}^w, \widetilde{\mathbf{Y}'}^w, \widetilde{\mathbf{Z}'}^w)$

is not $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense. Then there exists a pair of non-empty of sets $\mathbf{U}' \subseteq \overline{\mathbf{U}}$ (here, $\subseteq$ refers to an pairwise subset notation), $\mathbf{V}' \subseteq \overline{\mathbf{V}}$, and $\mathbf{W}' \subseteq \overline{\mathbf{W}}$ such that

$$\begin{aligned}
&\Pr\left[(\widetilde{\mathbf{X}'}^w)_{\mathbf{U}'} = \mathbf{x}'_{\mathbf{U}'} \wedge (\widetilde{\mathbf{Y}'}^w)_{\mathbf{V}'} = \mathbf{y}'_{\mathbf{V}'} \wedge (\widetilde{\mathbf{Z}'}^w)_{\mathbf{W}'} = \mathbf{z}'_{\mathbf{W}'}\right] \\
&= \Pr\Big[(\mathbf{X}'^w)_{\mathbf{U}'} = \mathbf{x}'_{\mathbf{U}'} \wedge (\mathbf{Y}'^w)_{\mathbf{V}'} = \mathbf{y}'_{\mathbf{V}'} \wedge (\mathbf{Z}'^w)_{\mathbf{W}'} = \mathbf{z}'_{\mathbf{W}'} \Big| \\
&\qquad\qquad (\mathbf{X}'^w)_{\mathbf{U}} = \mathbf{x}'_{\mathbf{U}} \wedge (\mathbf{Y}'^w)_{\mathbf{V}} = \mathbf{y}'_{\mathbf{V}} \wedge (\mathbf{Z}'^w)_{\mathbf{W}} = \mathbf{z}'_{\mathbf{W}}\Big] \\
&> 2^{-(1-\delta)\left[\sum_{i=1}^{n_1} |U'_i| \log N_i + \sum_{j=1}^{n_2} \log p_j^{|V'_j|} + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k^{|W'_{k,j}|}\right]}.
\end{aligned}$$

The set $\mathbf{U} \cup \mathbf{U}'$ (also pairwise), $\mathbf{V} \cup \mathbf{V}'$, and $\mathbf{W} \cup \mathbf{W}'$ now form subsets for which

$$\begin{aligned}
&\Pr\left[(\mathbf{X}'^w)_{\mathbf{U}\cup\mathbf{U}'} = \mathbf{x}'_{\mathbf{U}\cup\mathbf{U}'} \wedge (\mathbf{Y}'^w)_{\mathbf{V}\cup\mathbf{V}'} = \mathbf{y}'_{\mathbf{V}\cup\mathbf{V}'} \wedge (\mathbf{Z}'^w)_{\mathbf{W}\cup\mathbf{W}'} = \mathbf{z}'_{\mathbf{W}\cup\mathbf{W}'}\right] \\
&= \Pr\Big[(\mathbf{X}'^w)_{\mathbf{U}} = \mathbf{x}'_{\mathbf{U}} \wedge (\mathbf{X}'^w)_{\mathbf{U}'} = \mathbf{x}'_{\mathbf{U}'} \wedge (\mathbf{Y}'^w)_{\mathbf{V}} = \mathbf{y}'_{\mathbf{V}} \wedge (\mathbf{Y}'^w)_{\mathbf{V}'} = \mathbf{y}'_{\mathbf{V}'} \\
&\qquad\qquad \wedge (\mathbf{Z}'^w)_{\mathbf{W}} = \mathbf{z}'_{\mathbf{W}} \wedge (\mathbf{Z}'^w)_{\mathbf{W}'} = \mathbf{z}'_{\mathbf{W}'}\Big] \\
&= \Pr\left[(\mathbf{X}'^w)_{\mathbf{U}} = \mathbf{x}'_{\mathbf{U}} \wedge (\mathbf{Y}'^w)_{\mathbf{V}} = \mathbf{y}'_{\mathbf{V}} \wedge (\mathbf{Z}'^w)_{\mathbf{W}} = \mathbf{z}'_{\mathbf{W}}\right] \\
&\quad \cdot \Pr\Big[(\mathbf{X}'^w)_{\mathbf{U}'} = \mathbf{x}'_{\mathbf{U}'} \wedge (\mathbf{Y}'^w)_{\mathbf{V}'} = \mathbf{y}'_{\mathbf{V}'} \wedge (\mathbf{Z}'^w)_{\mathbf{W}'} = \mathbf{z}'_{\mathbf{W}'} \Big| \\
&\qquad\qquad (\mathbf{X}'^w)_{\mathbf{U}} = \mathbf{x}'_{\mathbf{U}} \wedge (\mathbf{Y}'^w)_{\mathbf{V}} = \mathbf{y}'_{\mathbf{V}} \wedge (\mathbf{Z}'^w)_{\mathbf{W}} = \mathbf{z}'_{\mathbf{W}}\Big] \\
&> 2^{-(1-\delta)\left[\sum_{i=1}^{n_1} |U_i| \log N_i + \sum_{j=1}^{n_2} \log p_j^{|V_j|} + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k^{|W_{k,j}|}\right]} \\
&\quad \cdot 2^{-(1-\delta)\left[\sum_{i=1}^{n_1} |U'_i| \log N_i + \sum_{j=1}^{n_2} \log p_j^{|V'_j|} + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k^{|W'_{k,j}|}\right]} \\
&= 2^{-(1-\delta)\left[\sum_{i=1}^{n_1} |U_i \cup U'_i| \log N_i + \sum_{j=1}^{n_2} \log p_j^{|V_j \cup V'_j|} + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k^{|W_{k,j} \cup W'_{k,j}|}\right]},
\end{aligned}$$

since $(\mathbf{U}, \mathbf{V}, \mathbf{W})$ and $(\mathbf{U}', \mathbf{V}', \mathbf{W}')$ are pairwise disjoint. However, this contradicts the maximality of $(\mathbf{U}, \mathbf{V}, \mathbf{W})$.

(2) Next, we claim that

$$\sum_{i=1}^{n_1} |U_i| \log N_i + \sum_{j=1}^{n_2} |V_j| \log(p_j/e) + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} |W_{k,j}| \log(C_k/e) \le S_z/\delta.$$

Let $\mathcal{B}_{\mathbf{y}'_{\mathbf{V}}}$ be the set of vectors $\mathbf{y}'_{\overline{\mathbf{V}}}$ such that $\mathbf{y}'_{\mathbf{V}}$ and $\mathbf{y}'_{\overline{\mathbf{V}}}$ are a valid injection. Similarly, let $\mathcal{C}_{\mathbf{z}'_{\mathbf{W}}}$ be the set of vectors $\mathbf{z}'_{\overline{\mathbf{W}}}$ such that $\mathbf{z}'_{\mathbf{W}}$ and $\mathbf{z}'_{\overline{\mathbf{W}}}$ are a valid IC. Since $H_\infty(\mathbf{X}'^w, \mathbf{Y}'^w, \mathbf{Z}'^w) \ge \sum_{i=1}^{n_1} M_i \log N_i + \sum_{j=1}^{n_2} \log p_j! + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k! - S_z$, we observe that for any $\mathbf{x}'_{\mathbf{U}}, \mathbf{y}'_{\mathbf{V}}, \mathbf{z}'_{\mathbf{W}}$,

$$\begin{aligned}
&\Pr\left[(\mathbf{X}'^w)_{\mathbf{U}} = \mathbf{x}'_{\mathbf{U}} \wedge (\mathbf{Y}'^w)_{\mathbf{V}} = \mathbf{y}'_{\mathbf{V}} \wedge (\mathbf{Z}'^w)_{\mathbf{W}} = \mathbf{z}'_{\mathbf{W}}\right] \\
&= \sum_{\mathbf{x}'_{\overline{\mathbf{U}}} \in \prod_{i=1}^{n_1} [N_i]^{M_i - |U_i|}} \sum_{\mathbf{y}'_{\overline{\mathbf{V}}} \in \mathcal{B}_{\mathbf{y}'_{\mathbf{V}}}} \sum_{\mathbf{z}'_{\overline{\mathbf{W}}} \in \mathcal{C}_{\mathbf{z}'_{\mathbf{W}}}} \Pr\left[(\mathbf{X}^w)_{\mathbf{U}} = \mathbf{x}'_{\mathbf{U}}\right.
\end{aligned}$$

$$
\begin{aligned}
&\wedge (\mathbf{Y}^w)_{\mathbf{V}} = \mathbf{y'}_{\mathbf{V}} \wedge (\mathbf{Z}^w)_{\mathbf{W}} = \mathbf{z'}_{\mathbf{W}} \wedge (\mathbf{X}^w)_{\overline{\mathbf{U}}} = \mathbf{x'}_{\overline{\mathbf{U}}} \\
&\wedge (\mathbf{Y}^w)_{\overline{\mathbf{V}}} = \mathbf{y'}_{\overline{\mathbf{V}}} \wedge (\mathbf{Z}^w)_{\overline{\mathbf{W}}} = \mathbf{z'}_{\overline{\mathbf{W}}}\Big] \\
&\leq 2^{\sum_{i=1}^{n_1}(M_i - |U_i|)\log N_i} \cdot 2^{\sum_{j=1}^{n_2}\log(p_j - |V_j|)!} \cdot 2^{\sum_{k=1}^{n_3}\sum_{j=1}^{K_k}\log(C_k - |W_{j,k}|)!} \\
&\qquad\qquad \cdot 2^{-(\sum_{i=1}^{n_1} M_i \log N_i + \sum_{j=1}^{n_2}\log p_j! + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}\log C_k! - S_z)} \\
&= 2^{-(\sum_{i=1}^{n_1}|U_i|\log N_i + \sum_{j=1}^{n_2}\log p_j^{\underline{|V_j|}} + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}\log C_k^{\underline{|W_{k,j}|}} - S_z)},
\end{aligned}
$$

and hence,

$$
\begin{aligned}
&H_\infty((\mathbf{X'}^w)_{\mathbf{U}}, (\mathbf{Y'}^w)_{\mathbf{V}}, (\mathbf{Z'}^w)_{\mathbf{W}}) \\
&\qquad \geq \sum_{i=1}^{n_1}|U_i|\log N_i + \sum_{j=1}^{n_2}\log p_j^{\underline{|V_j|}} + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}\log C_k^{\underline{|W_{k,j}|}} - S_z.
\end{aligned}
$$

On the other hand, because $((\mathbf{X'}^w)_{\mathbf{U}}, (\mathbf{Y'}^w)_{\mathbf{V}}, (\mathbf{Z'}^w)_{\mathbf{W}})$ is not $(1-\delta)$-dense, we have

$$
\begin{aligned}
&H_\infty((\mathbf{X'}^w)_{\mathbf{U}}, (\mathbf{Y'}^w)_{\mathbf{V}}, (\mathbf{Z'}^w)_{\mathbf{W}}) \\
&\qquad \leq (1-\delta)\left[\sum_{i=1}^{n_1}|U_i|\log N_i + \sum_{j=1}^{n_2}\log p_j^{\underline{|V_j|}} + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}\log C_k^{\underline{|W_{k,j}|}}\right].
\end{aligned}
$$

Combining those two equations together, we have

$$
\begin{aligned}
S_z &\geq \delta\left[\sum_{i=1}^{n_1}|U_i|\log N_i + \sum_{j=1}^{n_2}\log p_j^{\underline{|V_j|}} + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}\log C_k^{\underline{|W_{k,j}|}}\right] \\
&\geq \delta\left[\sum_{i=1}^{n_1}|U_i|\log N_i + \sum_{j=1}^{n_2}|V_j|\log(p_j/e) + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}|W_{k,j}|\log(C_k/e)\right],
\end{aligned}
$$

where the last inequality comes from [CDG18, Proposition 38][8]. Divided by $\delta > 0$, we get the desired result.

Hence, $(\widetilde{\mathbf{X}'}^w, \widetilde{\mathbf{Y}'}^w, \widetilde{\mathbf{Z}'}^w)$ is a $(\mathbf{P'}, \mathbf{Q'}, \mathbf{R'}, 1-\delta)$-dense source such that

$$
\sum_{i=1}^{n_1}P'_i\log N_i + \sum_{j=1}^{n_2}Q'_j\log(p_j/e) + \sum_{k=1}^{n_3}\sum_{j=1}^{K_k}R'_{k,j}\log(C_k/e) \leq S_z/\delta.
$$

Set $(\mathbf{X'}^w, \mathbf{Y'}^w, \mathbf{Z'}^w)$ now to be $(\mathbf{X'}^w, \mathbf{Y'}^w, \mathbf{Z'}^w)$ conditioned on $(\mathbf{X'}^w)_{\mathbf{U}} \neq \mathbf{x'}_{\mathbf{U}}$, $(\mathbf{Y'}^w)_{\mathbf{V}} \neq \mathbf{y'}_{\mathbf{V}}$, and $(\mathbf{Z'}^w)_{\mathbf{W}} \neq \mathbf{z'}_{\mathbf{W}}$ and recursively decompose $(\mathbf{X'}^w, \mathbf{Y'}^w, \mathbf{Z'}^w)$ as long as

$$
\Pr\left[\mathbf{X}^w \in \mathrm{supp}(\mathbf{X'}^w) \wedge \mathbf{Y}^w \in \mathrm{supp}(\mathbf{Y'}^w) \wedge \mathbf{Z}^w \in \mathrm{supp}(\mathbf{Z'}^w)\right] > \gamma.
$$

Observe that

$$
H_\infty(\mathbf{X'}^w, \mathbf{Y'}^w, \mathbf{Z'}^w)
$$

[8] [CDG18, Proposition 38] says that $N^{\underline{j}} \geq (N/e)^j$, where $a^{\underline{b}} \coloneqq a!/(a-b)!$.

$$\geq \sum_{i=1}^{n_1} M_i \log N_i + \sum_{j=1}^{n_2} \log p_j! + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k! - (S_z + \log \gamma^{-1})$$

at any point in this decomposition process since

$$\begin{aligned}
&\Pr\left[\mathbf{X}'^w = \mathbf{x}' \wedge \mathbf{Y}'^w = \mathbf{y}' \wedge \mathbf{Z}'^w = \mathbf{z}'\right] \\
&= \Pr\Big[\mathbf{X}^w = \mathbf{x}' \wedge \mathbf{Y}^w = \mathbf{y}' \wedge \mathbf{Z}^w = \mathbf{z}' \Big| \\
&\qquad \mathbf{X}^w \in \mathrm{supp}(\mathbf{X}'^w) \wedge \mathbf{Y}^w \in \mathrm{supp}(\mathbf{Y}'^w) \wedge \mathbf{Z}^w \in \mathrm{supp}(\mathbf{Z}'^w)\Big] \\
&\leq \frac{\Pr\left[\mathbf{X}^w = \mathbf{x}' \wedge \mathbf{Y}^w = \mathbf{y}' \wedge \mathbf{Z}^w = \mathbf{z}'\right]}{\Pr\left[\mathbf{X}^w \in \mathrm{supp}(\mathbf{X}'^w) \wedge \mathbf{Y}^w \in \mathrm{supp}(\mathbf{Y}'^w) \wedge \mathbf{Z}^w \in \mathrm{supp}(\mathbf{Z}'^w)\right]} \\
&\leq \frac{2^{-(\sum_{i=1}^{n_1} M_i \log N_i + \sum_{j=1}^{n_2} \log p_j! + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k! - S_z)}}{\gamma} \\
&= 2^{-(\sum_{i=1}^{n_1} M_i \log N_i + \sum_{j=1}^{n_2} \log p_j! + \sum_{k=1}^{n_3} \sum_{j=1}^{K_k} \log C_k! - (S_z + \log \gamma^{-1}))}.
\end{aligned}$$

Note that $\mathrm{supp}(\mathbf{X}'^w), \mathrm{supp}(\mathbf{Y}'^w), \mathrm{supp}(\mathbf{Z}'^w)$ decreases in every step, and since $\mathrm{supp}(\mathbf{X}^w), \mathrm{supp}(\mathbf{Y}^w)$, $\mathrm{supp}(\mathbf{Z}^w)$ are finite, after finitely many steps, this process ends with $(\mathbf{X}'^w_{\mathsf{final}}, \mathbf{Y}'^w_{\mathsf{final}}, \mathbf{Z}'^w_{\mathsf{final}})$ with

$$\Pr\left[\mathbf{X}^w \in \mathrm{supp}(\mathbf{X}'^w_{\mathsf{final}}) \wedge \mathbf{Y}^w \in \mathrm{supp}(\mathbf{Y}'^w_{\mathsf{final}}) \wedge \mathbf{Z}^w \in \mathrm{supp}(\mathbf{Z}'^w_{\mathsf{final}})\right] > \gamma.$$

Hence, $(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$ is a convex combination of finitely many $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense sources and $(\mathbf{X}'^w_{\mathsf{final}}, \mathbf{Y}'^w_{\mathsf{final}}, \mathbf{Z}'^w_{\mathsf{final}})$, where $\mathbf{P}', \mathbf{Q}', \mathbf{R}'$ satisfy the inequality

$$\sum_{i=1}^{n_1} P'_i \log N_i + \sum_{i=1}^{n_2} Q'_i \log(p_i/e) + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} R'_{i,j} \log(C_i/e) \leq \frac{S_z + \log \gamma^{-1}}{\delta}.$$

This implies that $(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$ is γ-close to a convex combination of $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense sources where $\sum_{i=1}^{n_1} P'_i \log N_i + \sum_{i=1}^{n_2} Q'_i \log(p_i/e) + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} R'_{i,j} \log(C_i/e) \leq (S_z + \log \gamma^{-1})/\delta$. □

Let $(\widetilde{\mathbf{X}}^w, \widetilde{\mathbf{Y}}^w, \widetilde{\mathbf{Z}}^w)$ be the convex combination of $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense sources that is γ-close to $(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w)$ for a $\delta = \delta_z$ to be determined later. For every $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$ source $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}})$ in said convex combination, let $(\widetilde{\mathbf{X}'}, \widetilde{\mathbf{Y}'}, \widetilde{\mathbf{Z}'})$ be the corresponding $(\mathbf{P}', \mathbf{Q}', \mathbf{R}')$-bit-fixing source, i.e., $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}})$ and $(\widetilde{\mathbf{X}'}, \widetilde{\mathbf{Y}'}, \widetilde{\mathbf{Z}'})$ are fixed on the same coordinates to the same values. The following claim bounds the distinguishing advantage between $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}})$ and $(\widetilde{\mathbf{X}'}, \widetilde{\mathbf{Y}'}, \widetilde{\mathbf{Z}'})$ for any $(\mathbf{S}, \mathbf{T}, \mathbf{L})$-query distinguisher.

Claim B.2. For any $(\mathbf{P}', \mathbf{Q}', \mathbf{R}', 1-\delta)$-dense source $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}})$ and its corresponding $(\mathbf{P}', \mathbf{Q}', \mathbf{R}')$-bit-fixing source $(\widetilde{\mathbf{X}'}, \widetilde{\mathbf{Y}'}, \widetilde{\mathbf{Z}'})$, it holds that for any (adaptive) distinguisher $\mathcal{D}$ that queries at most $\mathbf{S} = (S_1, \ldots, S_{n_1})$ coordinates of each oracle that corresponds to $\widetilde{\mathbf{X}}$ where $S_i < P'_i$ for each $i \in [n_1]$, $\mathbf{T} = (T_1, \ldots, T_{n_2})$ coordinates of each oracle that corresponds to $\widetilde{\mathbf{Y}}$ where $T_i < Q'_i$ for each $i \in [n_2]$, and $\mathbf{L} = (\mathbf{L}_1, \ldots, \mathbf{L}_{n_3})$ coordinates of each oracle that corresponds to $\widetilde{\mathbf{Z}}$ where $\mathbf{L}_k = (L_{k,1}, \ldots, L_{k,K_k})$ and $L_{i,j} < R'_{i,j}$ for each $j \in [K_i]$ and $i \in [n_3]$,

$$\left|\Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}}} = 1\right] - \Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}'}, \widetilde{\mathbf{Y}'}, \widetilde{\mathbf{Z}'}} = 1\right]\right|$$

$$\leq \delta \left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} L_{i,j} \log C_i \right),$$

and

$$\begin{aligned}
&\Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = 1\right] \\
&\quad \leq 2^{\delta\left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} L_{i,j} \log C_i\right)} \\
&\qquad \cdot \Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = 1\right].
\end{aligned}$$

Proof of Claim B.2. Without loss of generality, assume that $\mathcal{D}$ is deterministic and does not query any of the fixed positions, make the same query twice, or make an inverse query after making the corresponding forward query or vice-versa. Let $T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}$ and $T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}$ be the random variables corresponding to the transcripts containing the query/answer pairs resulting from $\mathcal{D}$'s interaction with $(\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}})$ and $(\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'})$, respectively.

For a fixed transcript t, denote by $\mathsf{p}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}(t)$ and $\mathsf{p}_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}(t)$ the probabilities that $(\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}})$ and $(\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'})$, respectively, produce the answers in t if the queries in t are asked. Since $\mathcal{D}$ is deterministic, $\Pr\left[T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = t\right] \in \left\{0, \mathsf{p}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}(t)\right\}$, and similarly, $\Pr\left[T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = t\right] \in \left\{0, \mathsf{p}_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}(t)\right\}$. Denote by $\mathcal{T}_{\mathbf{X},\mathbf{Y},\mathbf{Z}}$ the set of all transcripts t for which $\Pr\left[T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = t\right] > 0$. For such t, $\Pr\left[T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = t\right] = \mathsf{p}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}(t)$ and also $\Pr\left[T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = t\right] = \mathsf{p}_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}(t)$. Observe that for every transcript t,

$$\begin{aligned}
&\mathsf{p}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}(t) \\
&\leq 2^{-(1-\delta)\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} \log(p_i - Q'_i)^{\underline{T_i}} + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} \log(C_i - R'_{i,j})^{\underline{L_{i,j}}}\right]},
\end{aligned} \tag{12}$$

and

$$\begin{aligned}
&\mathsf{p}_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}(t) \\
&\leq 2^{-\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} \log(p_i - Q'_i)^{\underline{T_i}} + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} \log(C_i - R'_{i,j})^{\underline{L_{i,j}}}\right]},
\end{aligned} \tag{13}$$

as $(\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}})$ is $(\mathbf{P}',\mathbf{Q}',\mathbf{R}',1-\delta)$-dense and $(\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'})$ is $(\mathbf{P}',\mathbf{Q}',\mathbf{R}')$-fixed.

Towards proving the first part of the claim, observe that $\mathcal{D}$'s output can be computed from the transcript (including whether a query was a forward for an inverse query) by just running $\mathcal{D}$ and providing the answers to its queries from the transcript. Hence,

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = 1\right] - \Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = 1\right]\right| \leq \mathsf{SD}\left(T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}, T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}\right) \\
&\quad = \sum_t \max\left\{0, \Pr\left[T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = t\right] - \Pr\left[T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = t\right]\right\} \\
&\quad = \sum_{t \in \mathcal{T}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}} \max\left\{0, \mathsf{p}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}(t) - \mathsf{p}_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}(t)\right\} \\
&\quad = \sum_{t \in \mathcal{T}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}} \mathsf{p}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}(t) \cdot \max\left\{0, 1 - \frac{\mathsf{p}_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}(t)}{\mathsf{p}_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}(t)}\right\}
\end{aligned}$$

$$
\begin{aligned}
&\leq 1 - 2^{-\delta\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} \log(p_i - Q_i')^{\underline{T_i}} + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} \log(C_i - R_{i,j}')^{\underline{L_{i,j}}}\right]} \\
&\leq 1 - 2^{-\delta\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right]} \\
&\leq -\delta \left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right],
\end{aligned}
$$

where the first sum is over all possible transcripts and where the last inequality uses $2^{-x} \geq 1 - x$ for $x \geq 0$ and $a^{\underline{b}} \leq a^b$ for $a, b \in \mathbb{N}$.

As for the second part of the claim, observe that due to the equation (12),(13), and the support of $T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}}$ being a subset of $T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}}$,

$$
\begin{aligned}
&\Pr\left[T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = t\right] \\
&\leq 2^{\delta\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} \log(p_i - Q_i')^{\underline{T_i}} + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} \log(C_i - R_{i,j}')^{\underline{L_{i,j}}}\right]} \\
&\qquad \cdot \Pr\left[T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = t\right] \\
&\leq 2^{\delta\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right]} \cdot \Pr\left[T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = t\right],
\end{aligned}
$$

for any transcript t. Let $\mathcal{T}_{\mathcal{D}}$ be the set of transcripts where $\mathcal{D}$ outputs 1. Then we have

$$
\begin{aligned}
&\Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = 1\right] = \sum_{t \in \mathcal{T}_{\mathcal{D}}} \Pr\left[T_{\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}},\widetilde{\mathbf{Z}}} = t\right] \\
&\leq 2^{\delta\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right]} \\
&\qquad \cdot \sum_{t \in \mathcal{T}_{\mathcal{D}}} \Pr\left[T_{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = t\right] \\
&= 2^{\delta\left[\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right]} \cdot \Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}'},\widetilde{\mathbf{Y}'},\widetilde{\mathbf{Z}'}} = 1\right],
\end{aligned}
$$

which concludes the proof. □

Let $(\widetilde{\mathbf{X}'}^w, \widetilde{\mathbf{Y}'}^w, \widetilde{\mathbf{Z}'}^w)$ be obtained by replacing $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}, \widetilde{\mathbf{Z}})$ by $(\widetilde{\mathbf{X}'}, \widetilde{\mathbf{Y}'}, \widetilde{\mathbf{Z}'})$ in $(\widetilde{\mathbf{X}}^w, \widetilde{\mathbf{Y}}^w, \widetilde{\mathbf{Z}}^w)$. Setting

$$
\delta_z = \frac{S_z + \log \gamma^{-1}}{\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i},
$$

Claim B.1 and Claim B.2 imply

$$
\left|\Pr\left[\mathcal{D}^{\mathbf{X}^w,\mathbf{Y}^w,\mathbf{Z}^w}(z) = 1\right] - \Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}'}^w,\widetilde{\mathbf{Y}'}^w,\widetilde{\mathbf{Z}'}^w}(z) = 1\right]\right| \leq \Lambda + \gamma, \tag{14}
$$

as well as

$$
\Pr\left[\mathcal{D}^{\mathbf{X}^w,\mathbf{Y}^w,\mathbf{Z}^w}(z) = 1\right] \leq 2^{\Lambda} \cdot \Pr\left[\mathcal{D}^{\widetilde{\mathbf{X}'}^w,\widetilde{\mathbf{Y}'}^w,\widetilde{\mathbf{Z}'}^w}(z) = 1\right] + \gamma, \tag{15}
$$

where

$$
\Lambda = \frac{\left(S + \log \gamma^{-1}\right)\left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right)}{\sum_{i=1}^{n_1} P_i \log N_i + \sum_{i=1}^{n_2} Q_i \log(p_i/e) + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} R_{i,j} \log(C_i/e)}.
$$

Moreover, note that for the above choice of δ_z, we have $(\mathbf{P}', \mathbf{Q}', \mathbf{R}') = (\mathbf{P}, \mathbf{Q}, \mathbf{R})$, i.e., the sources $(\widetilde{\mathbf{X}'}^w, \widetilde{\mathbf{Y}'}^w, \widetilde{\mathbf{Z}'}^w)$ are $(\mathbf{P}, \mathbf{Q}, \mathbf{R})$-fixed, as desired.

Claim B.3. $\mathbb{E}_z[S_z] \leq S$ and $\Pr[S_{f(\mathbf{X},\mathbf{Y},\mathbf{Z})} > S + \log \gamma^{-1}] \leq \gamma$.

Proof of Claim B.3. Observe that $H_\infty(\mathbf{X}^w, \mathbf{Y}^w, \mathbf{Z}^w) = H_\infty((\mathbf{X}, \mathbf{Y}, \mathbf{Z})|W = w) = H((\mathbf{X}, \mathbf{Y}, \mathbf{Z})|W = w)$ since, conditioned on $W = w$, $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ is distributed uniformly over all values $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ with $f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = w$. Hence,

$$\begin{aligned}
\mathbb{E}_z[S_z] &= \sum_{i=1}^{n_1} M_i \log N_i + \sum_{i=1}^{n_2} \log p_i! + \sum_{i=1}^{n_3} K_i \log C_i! - \mathbb{E}_z\left[H_\infty((\mathbf{X}, \mathbf{Y}, \mathbf{Z})|W = w)\right] \\
&= \sum_{i=1}^{n_1} M_i \log N_i + \sum_{i=1}^{n_2} \log p_i! + \sum_{i=1}^{n_3} K_i \log C_i! - \mathbb{E}_z\left[H((\mathbf{X}, \mathbf{Y}, \mathbf{Z})|W = w)\right] \\
&= \sum_{i=1}^{n_1} M_i \log N_i + \sum_{i=1}^{n_2} \log p_i! + \sum_{i=1}^{n_3} K_i \log C_i! - H((\mathbf{X}, \mathbf{Y}, \mathbf{Z})|W = w) \leq S.
\end{aligned}$$

Again, due to the uniformity of $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$, $\Pr[f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = z] = 2^{-S_z}$. Hence,

$$\begin{aligned}
&\Pr[S_{f(\mathbf{X},\mathbf{Y},\mathbf{Z})} > S + \log \gamma^{-1}] \\
&\qquad = \sum_{w \in \{0,1\}^S : S_z > S + \log \gamma^{-1}} \Pr[f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = w] \\
&\qquad \leq 2^S \cdot 2^{-(S + \log \gamma^{-1})} \leq \gamma.
\end{aligned}$$

□

Now the lemma follows (using $(\mathbf{X}'^w, \mathbf{Y}'^w, \mathbf{Z}'^w) \coloneqq (\widetilde{\mathbf{X}'}^w, \widetilde{\mathbf{Y}'}^w, \widetilde{\mathbf{Z}'}^w)$) by taking expectations over w of the equation (14) and applying the first part of Claim B.3. The second part of the Lemma is proved as follows:

$$\begin{aligned}
&\Pr\left[\mathcal{D}^{\mathbf{X},\mathbf{Y},\mathbf{Z}}(f(\mathbf{X}, \mathbf{Y}, \mathbf{Z})) = 1\right] \\
&\leq \Pr\left[\mathcal{D}^{\mathbf{X},\mathbf{Y},\mathbf{Z}}(f(\mathbf{X}, \mathbf{Y}, \mathbf{Z})) = 1, S_{f(\mathbf{X},\mathbf{Y},\mathbf{Z})} \leq S + \log \gamma^{-1}\right] + \Pr\left[S_{f(\mathbf{X},\mathbf{Y},\mathbf{Z})} > S + \log \gamma^{-1}\right] \\
&\leq \left(2^{\frac{\left(S + \log \gamma^{-1}\right)\left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right)}{\eta}} \cdot \Pr\Big[\mathcal{D}^{\mathbf{X},\mathbf{Y},\mathbf{Z}}(f(\mathbf{X}, \mathbf{Y}, \mathbf{Z})) = 1,\right. \\
&\qquad\qquad \left. S_{f(\mathbf{X},\mathbf{Y},\mathbf{Z})} \leq S + \log \gamma^{-1}\Big] + \gamma\right) + \gamma \\
&\leq 2^{\frac{\left(S + \log \gamma^{-1}\right)\left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right)}{\eta}} \cdot \Pr\Big[\mathcal{D}^{\mathbf{X},\mathbf{Y},\mathbf{Z}}(f(\mathbf{X}, \mathbf{Y}, \mathbf{Z})) = 1, \\
&\qquad\qquad S_{f(\mathbf{X},\mathbf{Y},\mathbf{Z})} \leq S + \log \gamma^{-1}\Big] + 2\gamma,
\end{aligned}$$

where the second inequality follows by taking expectations over w of the equation (15) (together with the condition $S_z \leq S + \log \gamma^{-1}$) and the second part of Claim B.3. □

B.2 From the Bit-Fixing Model to the Auxiliary-Input Model

Capturing the Models. A tuple of oracles $\left(\{\mathcal{O}_{1,i}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}\}_{k=1}^{n_3}\right)$ has two interfaces $(\{\mathcal{O}_{1,i}.\mathsf{pre}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{pre}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{pre}\}_{k=1}^{n_3})$ that is accessible during a preprocessing phase, and $(\{\mathcal{O}_{1,i}.\mathsf{on}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{on}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{on}\}_{k=1}^{n_3})$ that is accessible during an online phase, where $\Big(\{\mathcal{O}_{1,i}.\mathsf{pre}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{pre}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{pre}\}_{k=1}^{n_3}\Big)$ is accessible only once before any calls to $\Big(\{\mathcal{O}_{1,i}.\mathsf{on}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{on}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{on}\}_{k=1}^{n_3}\Big)$ are made. We consider the oracles as follows:

- **Random Oracle + Generic Group Oracle + Ideal Cipher** $\mathsf{RO{+}GG{+}IC}(\{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$**:** Samples random oracle function tables $\mathsf{H}_i \leftarrow \mathcal{H}_{M_i,N_i}$ for each $i \in [n_1]$, where $\mathcal{H}_{M_i,N_i}$ is the set of all functions from $[M_i] \to [N_i]$, samples random injections $\tau_j \leftarrow \mathcal{I}_{p_j,\ell_j}$ for each $j \in [n_2]$, where $\mathcal{I}_{p_j,\ell_j}$ is the set of all injections from $\mathbb{Z}_{p_j} \to \mathbb{G}_j$ where $\mathbb{G}_j$ is the set of bitstrings of length $\ell_j \geq \log p_j$, and samples random permutations $\pi_s \leftarrow \mathcal{P}_{C_k}$ for each $s \in [K_k]$, where $\mathcal{P}_{C_k}$ is the set of all permutations from $[C_k] \to [C_k]$; offers no functionality at $(\{\mathcal{O}_{1,i}.\mathsf{pre}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{pre}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{pre}\}_{k=1}^{n_3})$; for the online queries,
  - for each $i \in [n_1]$, answers queries $x \in [M_i]$ to H_i at $\mathcal{O}_{1,i}.\mathsf{on}$ by the corresponding value $\mathsf{H}_i(x) \in [N_i]$,
  - for each $j \in [n_2]$, answers *forward* queries $x' \in \mathbb{Z}_{p_j}$ to τ_j at $\mathcal{O}_{2,j}.\mathsf{on}$ by the corresponding value $\tau_j(x') \in \mathbb{G}_j$; answers *group-operation* queries (s, s') at $\mathcal{O}_{2,j}.\mathsf{on}$ as follows: if $s = \tau_j(x)$ and $s' = \tau_j(x')$ for some x, x', the oracle replies by $\tau_j(x + x')$ and by $\perp$ otherwise; answers *inverse* queries s at $\mathcal{O}_{2,j}.\mathsf{on}$ by returning $\tau_j^{-1}(s)$ if s is in the range of τ_j and $\perp$ otherwise, and
  - for each $k \in [n_3]$, answers both *forward* and *backward* queries $(s, x) \in [K_k] \times [C_k]$ at $\mathcal{O}_{3,k}.\mathsf{on}$ by the corresponding value $\pi_s(x) \in [C_k]$ or $\pi_s^{-1}(x) \in [C_k]$, respectively.
- **Auxiliary-Input Random Oracle + Generic Group Oracle + Ideal Cipher** $\mathsf{AI{-}RO{+}GG{+}IC}$ $(\{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$**:** Samples $\mathsf{H}_i, \tau_j, \pi_s$ for each $i \in [n_1], j \in [n_2]$, and $s \in [K_k]$ where $k \in [n_3]$ as explained in $\mathsf{RO{+}GG{+}IC}(\{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$ above; outputs all of H_i, τ_j, and π_s at $(\{\mathcal{O}_{1,i}.\mathsf{pre}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{pre}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{pre}\}_{k=1}^{n_3})$; $(\{\mathcal{O}_{1,i}.\mathsf{on}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{on}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{on}\}_{k=1}^{n_3})$ behaves the same as $\mathsf{RO{+}GG{+}IC}(\{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$. When the parameters $(\{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$ are clear in context, we sometimes abuse the notation and simply say $\mathsf{AI{-}RO{+}GG{+}IC}$.
- **Bit-Fixing Random Oracle + Generic Group Oracle + Ideal Cipher** $\mathsf{BF{-}RO{+}GG{+}IC}(\mathbf{P}, \mathbf{Q}, \mathbf{R}, \{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$**:** Samples a random oracle function table $\mathsf{H}_i \leftarrow \mathcal{H}_{M_i,N_i}$ at $\mathcal{O}_{1,i}.\mathsf{pre}$ for each $i \in [n_1]$, where $\mathcal{H}_{M_i,N_i}$ is the set of all functions from $[M_i] \to [N_i]$, samples a random size-p_j subset $\mathcal{Y}_j$ of $\mathbb{G}_j$ for each $j \in [n_2]$ and outputs $\mathcal{Y}_j$ at $\mathcal{O}_{2,j}.\mathsf{pre}$, and samples a random permutation $\pi_s \leftarrow \mathcal{P}_{C_k}$ at $\mathcal{O}_{3,k}.\mathsf{pre}$ for each $s \in [K_k]$, where $\mathcal{P}_{C_k}$ is the set of all permutations from $[C_k] \to [C_k]$.
  - Given $\mathbf{P} = (P_1, \ldots, P_{n_1}) \in \mathbb{Z}_{M_1} \times \cdots \times \mathbb{Z}_{M_{n_1}}$, for each $i \in [n_1]$, takes a list at $\mathcal{O}_{1,i}.\mathsf{pre}$ of at most P_i query/answer pairs that override H_i in the corresponding positions,
  - given $\mathbf{Q} = (Q_1, \ldots, Q_{n_2}) \in \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_{n_1}}$, for each $j \in [n_2]$, takes a list at $\mathcal{O}_{2,j}.\mathsf{pre}$ of at most Q_j query/answer pairs without collisions and all answers in $\mathcal{Y}_j$; samples a random injection $\tau_j \leftarrow \mathcal{I}_{p_j,\ell_j}$ with range $\mathcal{Y}_j$ and consistent with said list, and
  - given $\mathbf{R} = (\mathbf{R}_1, \ldots, \mathbf{R}_{n_3}) \in (\mathbb{Z}_{C_1})^{K_1} \times \cdots \times (\mathbb{Z}_{C_{n_3}})^{K_{n_3}}$ where $\mathbf{R}_k = (R_{k,1}, \ldots, R_{k,K_k})$ for each $k \in [n_3]$, takes a list at $\mathcal{O}_{3,k}.\mathsf{pre}$ of at most $R_{k,s}$ query/answer pairs (without collisions

for each $s \in [K_k]$); samples a random permutation $\pi_s \leftarrow \mathcal{P}_{C_k}$ consistent with said list for each $s \in [K_k]$.

- $(\{\mathcal{O}_{1,i}.\mathsf{on}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{on}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{on}\}_{k=1}^{n_3})$ behaves the same as $\mathsf{RO{+}GG{+}IC}(\{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$. When the parameters $(\{M_i, N_i\}_{i=1}^{n_1}, \{p_j, \ell_j\}_{j=1}^{n_2}, \{K_k, C_k\}_{k=1}^{n_3})$ are clear in context, we sometimes abuse the notation and simply say $\mathsf{BF\text{-}RO{+}GG{+}IC}(\mathbf{P}, \mathbf{Q}, \mathbf{R})$.

Attackers with Oracle-Dependent Advice. We define the attackers $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$ similar to that of prior work [CDGS18, CDG18], which consist of a preprocessing attacker $\mathcal{A}_{\mathsf{pre}}$ and an online attacker $\mathcal{A}_{\mathsf{on}}$, which carries out the actual attack using the output of $\mathcal{A}_{\mathsf{pre}}$. The difference is that we consider a tuple of oracles instead of a single oracle. More precisely, in the presence of a tuple of oracles $\left(\{\mathcal{O}_{1,i}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}\}_{k=1}^{n_3}\right)$, $\mathcal{A}_{\mathsf{pre}}$ interacts with $(\{\mathcal{O}_{1,i}.\mathsf{pre}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{pre}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{pre}\}_{k=1}^{n_3})$ and $\mathcal{A}_{\mathsf{on}}$ with $\left(\{\mathcal{O}_{1,i}.\mathsf{on}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{on}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{on}\}_{k=1}^{n_3}\right)$.

Definition 7. *Given* $\mathbf{S} = (S_1, \ldots, S_{n_1}), \mathbf{T} = (T_1, \ldots, T_{n_2})$, *and* $\mathbf{L} = (\mathbf{L}_1, \ldots, \mathbf{L}_{n_3})$ *where* $\mathbf{L}_k = (L_{k,1}, \ldots, L_{k,K_k})$ *for* $k \in [n_3]$, *an* $(S, \mathbf{S}, \mathbf{T}, \mathbf{L})$-attacker $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$ in the $\left(\{\mathcal{O}_{1,i}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}\}_{k=1}^{n_3}\right)$-model *consists of two procedures*

- $\mathcal{A}_{\mathsf{pre}}$, *which is computationally unbounded, interacts with* $(\{\mathcal{O}_{1,i}.\mathsf{pre}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{pre}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{pre}\}_{k=1}^{n_3})$, *and outputs an* S*-bit string as hint, and*
- $\mathcal{A}_{\mathsf{on}}$, *which takes an* S*-bit auxiliary input and makes at most* $(\mathbf{S}, \mathbf{T}, \mathbf{L})$ *queries to* $(\{\mathcal{O}_{1,i}.\mathsf{on}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{on}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{on}\}_{k=1}^{n_3})$. *That is,* $\mathcal{A}_{\mathsf{on}}$ *makes at most* S_i *queries to* $\mathcal{O}_{1,i}.\mathsf{on}$ *for each* $i \in [n_1]$, T_j *queries to* $\mathcal{O}_{2,j}.\mathsf{on}$ *for each* $j \in [n_2]$, *and* $L_{k,s}$ *queries to* $\mathcal{O}_{3,k}.\mathsf{on}$ *for each* $k \in [n_3]$ *and for each key* $s \in [K_k]$.

The definition of an *application* G and the *success* of $\mathcal{A}$ on G (see Definition 4) can be easily extended by defining $\mathcal{O} \coloneqq \left(\{\mathcal{O}_{1,i}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}\}_{k=1}^{n_3}\right)$, $\mathcal{O}.\mathsf{pre} \coloneqq \left(\{\mathcal{O}_{1,i}.\mathsf{pre}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{pre}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{pre}\}_{k=1}^{n_3}\right)$, and $\mathcal{O}.\mathsf{on} \coloneqq (\{\mathcal{O}_{1,i}.\mathsf{on}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}.\mathsf{on}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}.\mathsf{on}\}_{k=1}^{n_3})$. The *advantage* of an attacker $\mathcal{A}$ for an *indistinguishability/unpredictability* application G can also be defined accordingly for $\mathcal{O} = \left(\{\mathcal{O}_{1,i}\}_{i=1}^{n_1}, \{\mathcal{O}_{2,j}\}_{j=1}^{n_2}, \{\mathcal{O}_{3,k}\}_{k=1}^{n_3}\right)$ as defined in Definition 5.

Additive Error for Arbitrary Applications in the ROM+GGM+ICM. Using Lemma 5, one can extend [CDGS18, Theorem 5] and [CDG18, Theorem 1] to the translation from bit-fixing model with multiple idealized models (ROM+GGM+ICM) to the corresponding auxiliary-input model at the cost of an additive term which is similar to those from [CDGS18, Theorem 5] and [CDG18, Theorem 1].

Theorem 10. *For every* $\gamma > 0$, *if an application* G *is* $(S, \mathbf{S}, \mathbf{T}, \mathbf{L}, \varepsilon')$*-secure in the* $\mathsf{BF\text{-}RO{+}GG{+}IC}$ $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, \{M_i, N_i\}_{i=1}^{n_1}, \{p_i, \ell_i\}_{i=1}^{n_2}, \{K_i, C_i\}_{i=1}^{n_3})$*-model for any* $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ *as introduced in Definition 6 which satisfies*

$$\sum_{i=1}^{n_1} P_i \log N_i + \sum_{i=1}^{n_2} Q_i \log(p_i/e) + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} R_{i,j} \log(C_i/e) \leq \eta,$$

then it is $(S, \mathbf{S}, \mathbf{T}, \mathbf{L}, \varepsilon)$*-secure in the* $\mathsf{AI\text{-}RO{+}GG{+}IC}(\{M_i, N_i\}_{i=1}^{n_1}, \{p_i, \ell_i\}_{i=1}^{n_2}, \{K_i, C_i\}_{i=1}^{n_3})$*-model, for* $\varepsilon \leq \varepsilon' + \Lambda + 2\gamma$*, where*

$$\Lambda = \frac{2(S + \log \gamma^{-1}) \left(\sum_{i=1}^{n_1} (S_i)_G^{\mathsf{comb}} \log N_i + \sum_{i=1}^{n_2} 3(T_i)_G^{\mathsf{comb}} \log p_i + \sum_{i=1}^{n_3} \sum_{j=1}^{K_i} 2(L_{i,j})_G^{\mathsf{comb}} \log C_i \right)}{\eta},$$

e is the Euler constant, and $(S_i)_G^{\mathsf{comb}}, (T_i)_G^{\mathsf{comb}}$ *and* $(L_{i,j})_G^{\mathsf{comb}}$ *are the combined query complexity corresponding to* G *that corresponds to the random oracles, the generic group oracles, and the ideal ciphers, respectively.*

Proof. Fix $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ as well as γ. Let G be an arbitrary application and C be the corresponding challenger. Moreover, fix an $(S, \mathbf{S}, \mathbf{T}, \mathbf{L})$-attacker $\mathcal{A} = (\mathcal{A}_{\mathsf{pre}}, \mathcal{A}_{\mathsf{on}})$, and let $\{(\mathbf{X}', \mathbf{Y}', \mathbf{Z}')^{w,\mathcal{Y}}\}$, indexed by $w \in \{0,1\}^S$ and $\mathcal{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_{n_2})$ where $\mathcal{Y}_i$ is a size-p_i subset of $\mathbb{G}_i$ for each $i \in [n_2]$, be the family of distributions guaranteed to exist by Lemma 5. Consider the following $(S, \mathbf{S}, \mathbf{T}, \mathbf{L})$-attacker $\mathcal{A}' = (\mathcal{A}'_{\mathsf{pre}}, \mathcal{A}'_{\mathsf{on}})$ that is expected to interact with $\mathsf{BF\text{-}RO{+}GG{+}IC}$:

- $\mathcal{A}'_{\mathsf{pre}}$ obtains the set $\mathcal{Y}$ from $\{\mathsf{BF\text{-}GG.pre}\}_{i=1}^{n_2}$ and internally simulates $\mathcal{A}_{\mathsf{pre}}$ on a uniformly random input $\mathbf{X} \in \prod_{i=1}^{n_1}[M_i]$, a uniformly random injection $\mathbf{Y} \in \prod_{j=1}^{n_2} \mathbb{Z}_{p_j}$ with range $\mathcal{Y}$, a uniformly random IC $\mathbf{Z} \in \prod_{k=1}^{n_3}[K_k] \times [C_k]$ to obtain $w \leftarrow \mathcal{A}_{\mathsf{pre}}^{\mathsf{AI\text{-}RO{+}GG{+}IC.pre}}$. Then, it samples one of the $(\mathbf{P}, \mathbf{Q}, \mathbf{R})$-bit-fixing sources $(\mathbf{X}'', \mathbf{Y}'', \mathbf{Z}'')$ making up $(\mathbf{X}', \mathbf{Y}', \mathbf{Z}')^{w,\mathcal{Y}}$ and presets $\mathsf{BF\text{-}RO{+}GG{+}IC}$ to match $(\mathbf{X}'', \mathbf{Y}'', \mathbf{Z}'')$ on the at most $(\mathbf{P}, \mathbf{Q}, \mathbf{R})$ points (pairwise and element-wise) where $(\mathbf{X}'', \mathbf{Y}'', \mathbf{Z}'')$ is fixed. The output of $\mathcal{A}'_{\mathsf{pre}}$ is w.
- $\mathcal{A}'_{\mathsf{on}}$ works exactly the same as $\mathcal{A}_{\mathsf{on}}$.

Let $\mathcal{D}$ be a distinguisher — making forward, group-operation, inverse, and backward queries to an (n_1, n_2, n_3)-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 and backward queries to and back from its own oracle.
(2) It answers group-operation queries (s, s') (to the generic group oracle) by making two backward queries to its own oracle for s and s', obtaining i and j, respectively, making a forward query $i + j$, and passing the answer to $\mathcal{A}_{\mathsf{on}}$ or C (unless one of the answers to the backward queries was $\perp$, in which case $\perp$ is returned).

Note that $\mathcal{D}^{\mathbf{X},\mathbf{Y},\mathbf{Z}}(f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}))$ is identical to

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

and $\mathcal{D}^{(\mathbf{X}',\mathbf{Y}',\mathbf{Z}')^{f(\mathbf{X},\mathbf{Y},\mathbf{Z}),\mathsf{im}(\mathbf{Y})}}(f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}))$ is identical to

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

Furthermore, $\mathcal{D}$ is a distinguisher taking an S-bit input and making at most $(S_i)_G^{\mathsf{comb}}$ queries to the ith random oracle for $i \in [n_1]$, $3(T_j)_G^{\mathsf{comb}}$ queries to the jth generic group oracle for $j \in [n_2]$, and $2(L_{k,j})_G^{\mathsf{comb}}$ queries to the kth ideal cipher for $k \in [n_3]$ and for each key $j \in [K_k]$. Therefore, by Lemma 5,

$$
\begin{aligned}
&\mathrm{Succ}_{G,\mathsf{AI\text{-}RO+GG+IC}}(\mathcal{A}) \\
&\le \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GG+IC}}(\mathcal{A}') \\
&+ \frac{\left(S+\log\gamma^{-1}\right)\left(\sum_{i=1}^{n_1}(S_i)_G^{\mathsf{comb}}\log N_i + \sum_{i=1}^{n_2} 3(T_j)_G^{\mathsf{comb}}\log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} 2(L_{k,j})_G^{\mathsf{comb}}\log C_i\right)}{\eta} \\
&+ \gamma,
\end{aligned}
$$

for any $\mathbf{P},\mathbf{Q},\mathbf{R}$ satisfying

$$
\sum_{i=1}^{n_1} P_i \log N_i + \sum_{i=1}^{n_2} Q_i \log(p_i/e) + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} R_{i,j}\log(C_i/e) \le \eta.
$$

Since there is only an additive term between the two success probabilities, the above inequality implies

$$
\begin{aligned}
&\mathrm{Adv}_{G,\mathsf{AI\text{-}RO+GG+IC}}(\mathcal{A}) \\
&\le \mathrm{Adv}_{G,\mathsf{BF\text{-}RO+GG+IC}}(\mathcal{A}') \\
&+ \frac{\left(S+\log\gamma^{-1}\right)\left(\sum_{i=1}^{n_1}(S_i)_G^{\mathsf{comb}}\log N_i + \sum_{i=1}^{n_2} 3(T_j)_G^{\mathsf{comb}}\log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} 2(L_{k,j})_G^{\mathsf{comb}}\log C_i\right)}{\eta} \\
&+ 2\gamma,
\end{aligned}
$$

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+GGM+ICM.

Theorem 11. *For every $\gamma > 0$, if an* unpredictability *application G is $(S,\mathbf{S},\mathbf{T},\mathbf{L},\varepsilon')$-secure in the $\mathsf{BF\text{-}RO+GG+IC}(\mathbf{P},\mathbf{Q},\mathbf{R},\{M_i,N_i\}_{i=1}^{n_1},\{p_i,\ell_i\}_{i=1}^{n_2},\{K_i,C_i\}_{i=1}^{n_3})$-model for any $\mathbf{P},\mathbf{Q},\mathbf{R}$ as introduced in Definition 6 satisfying*

$$
\begin{aligned}
&(S+\log\gamma^{-1})\Bigg(\sum_{i=1}^{n_1}(S_i)_G^{\mathsf{comb}}\log N_i + \sum_{i=1}^{n_2} 3(T_i)_G^{\mathsf{comb}}\log p_i \\
&\qquad + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} 2(L_{i,j})_G^{\mathsf{comb}}\log C_i\Bigg) \\
&\le \sum_{i=1}^{n_1} P_i\log N_i + \sum_{i=1}^{n_2} Q_i\log(p_i/e) + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} R_{i,j}\log(C_i/e),
\end{aligned}
$$

then it is $(S,\mathbf{S},\mathbf{T},\mathbf{L},\varepsilon)$-secure in the $\mathsf{AI\text{-}RO+GG+IC}(\{M_i,N_i\}_{i=1}^{n_1},\{p_i,\ell_i\}_{i=1}^{n_2},\{K_i,C_i\}_{i=1}^{n_3})$-model, for

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

where e is the Euler's number, and $(S_i)_G^{\mathsf{comb}},(T_i)_G^{\mathsf{comb}}$ and $(L_{i,j})_G^{\mathsf{comb}}$ are the combined query complexity corresponding to G that corresponds to the random oracles, the generic group oracles, and the ideal ciphers, respectively.

Proof. Using the same attacker $\mathcal{A}'$ as in the proof of Theorem 10 and applying the second part of Lemma 5, for any $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ such that $\sum_{i=1}^{n_1} P_i \log N_i + \sum_{i=1}^{n_2} Q_i \log(p_i/e) + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} R_{i,j} \log(C_i/e) \le \eta$, we have

$$\begin{aligned}
&\mathrm{Succ}_{G,\mathsf{AI\text{-}RO+GG+IC}}(\mathcal{A}) \\
&\le 2^{\frac{\left(S+\log\gamma^{-1}\right)\left(\sum_{i=1}^{n_1} S_i \log N_i + \sum_{i=1}^{n_2} T_i \log p_i + \sum_{i=1}^{n_3}\sum_{j=1}^{K_i} L_{i,j} \log C_i\right)}{\eta}} \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GG+IC}}(\mathcal{A}') + 2\gamma \\
&\le 2 \cdot \mathrm{Succ}_{G,\mathsf{BF\text{-}RO+GG+IC}}(\mathcal{A}') + 2\gamma,
\end{aligned}$$

which translates into

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

for unpredictability applications. □

C Key-Prefixed Schnorr Under Preprocessing and Adaptive Corruptions

Key-prefixed Schnorr signatures have previously been studied in two complementary settings. Blocki and Lee [BL22] established preprocessing security for short key-prefixed Schnorr signatures without adaptive corruptions. More recently, Acharya et al. [AFOS26] established tight adaptive multi-user security for a key-prefixed formulation of Schnorr signatures, without considering preprocessing attacks.

Here, we consider the combination of these settings, while keeping the challenge length λ_{H} explicit. This is particularly relevant for short challenges. For nonzero Schnorr, Section 5.1 and Section 5.3 showed that adaptive corruption introduces the tight multi-target factor $\min\{N, q_{\mathsf{G}}^{\mathsf{on}}+1\}$ in the short-challenge term. We show that key-prefixing eliminates this factor even in the presence of preprocessing: the challenge term returns to $O(q/2^{\lambda_{\mathsf{H}}})$.

C.1 Key-Prefixed Schnorr in the BF-(ROM+GGM)

We first analyze key-prefixed Schnorr in the BF-(ROM+GGM) with adaptive corruptions. The reduction closely follows the adaptive nonzero-Schnorr analysis of Section 5.3, yet there are two differences. First, the hash input $\mathsf{H}(\mathsf{pk}\|I\|m)$ fixes the target user before the RO response is sampled, preventing adaptive target switching. Second, the key-prefixed variant allows the zero challenge and therefore does not require rejection sampling.

Theorem 12. *Let $p > 2^{2\lambda}$ be prime and suppose that $2^{\lambda_{\mathsf{H}}} < p$. Let $\mathcal{A}_{\mathsf{Sig}} = \left(\mathcal{A}_{\mathsf{Sig.pre}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathsf{G}})}, \mathcal{A}_{\mathsf{Sig.on}}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathsf{G}})}\right)$ be a bit-fixing adversary. Suppose that $\mathcal{A}_{\mathsf{Sig.on}}$ makes at most $q_{\mathsf{H}}^{\mathsf{on}}$ random-oracle queries, $q_{\mathsf{G}}^{\mathsf{on}}$ primitive generic-group queries (i.e., `Mult` and `Inv`), and $q_{\mathsf{S}}^{\mathsf{on}}$ signing queries. Define $\overline{q}_{\mathsf{G}}^{\mathsf{on}} := q_{\mathsf{G}}^{\mathsf{on}} + (q_{\mathsf{S}}^{\mathsf{on}}+1)(2\lceil\log_2 p\rceil + 2\lambda_{\mathsf{H}} + 2)$ and $D_{\mathsf{sig}} := p - 2P_{\mathsf{G}}(N + 3\overline{q}_{\mathsf{G}}^{\mathsf{on}} + 1) - (N + 3\overline{q}_{\mathsf{G}}^{\mathsf{on}} + 1)^2 - N$. Assume that $D_{\mathsf{sig}} > 0$. Then, $\Pr\left[\mathsf{SigForgeCor}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{KP}}}(\lambda) = 1\right] \le \varepsilon$, with*

$$\begin{aligned}
\varepsilon = \frac{\binom{N}{2} + N(P_{\mathsf{G}}+1)}{p} + \frac{\overline{q}_{\mathsf{G}}^{\mathsf{on}}(N+P_{\mathsf{G}}) + 3\overline{q}_{\mathsf{G}}^{\mathsf{on}}(\overline{q}_{\mathsf{G}}^{\mathsf{on}}+1)/2}{D_{\mathsf{sig}}} \\
+ \frac{NP_{\mathsf{H}}}{p} + \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}})}{p} + \frac{q_{\mathsf{H}}^{\mathsf{on}}(N + \overline{q}_{\mathsf{G}}^{\mathsf{on}})}{p - (N + P_{\mathsf{G}} + 3\overline{q}_{\mathsf{G}}^{\mathsf{on}} + 1)} + \frac{q_{\mathsf{H}}^{\mathsf{on}}+1}{2^{\lambda_{\mathsf{H}}}},
\end{aligned}$$

where the probability is over the ideal oracles and all random coins of the experiment and the adversary. Moreover, when $D_{\mathsf{sig}} = \Omega(p)$,

$$\varepsilon = O\left(\frac{(N + P_{\mathsf{H}} + q\log p)(N + P_{\mathsf{G}} + q\log p)}{p} + \frac{q}{2^{\lambda_{\mathsf{H}}}}\right).$$

Proof. We follow the reduction of Theorem 8 and describe only the changes caused by key-prefixing and by allowing the zero challenge. Corruption queries are forwarded to the adaptive bridge challenger exactly as before. For every signing and verification query, the simulated RO input is now $\mathsf{pk}_j \| I \| m$. Since zero is a valid challenge, the signing simulation does not abort when the RO response is zero. The generic-group query accounting is unchanged, so the reduction makes at most $\overline{q}_{\mathsf{G}}^{\mathsf{on}} = q_{\mathsf{G}}^{\mathsf{on}} + (q_{\mathsf{S}}^{\mathsf{on}} + 1)(2\lceil \log_2 p \rceil + 2\lambda_{\mathsf{H}} + 2)$ generic group queries.

Let KeyHit denote the event that the public key prefix of some input fixed in F_{H} matches an honestly generated public key pk_i for some user $i \in [N]$. Fix the preprocessing view and a public key prefix occurring in a fixed RO input. Since τ is injective, this prefix corresponds to at most one exponent in $\mathbb{Z}_p$. Since each honest secret key x_i is sampled uniformly from $\mathbb{Z}_p$ after preprocessing, the corresponding public key matches the fixed prefix with probability at most $1/p$. Union bounding over N users and P_{H} fixed RO inputs, $\Pr[\mathsf{KeyHit}] \le \frac{NP_{\mathsf{H}}}{p}$.

Condition on $\overline{\mathsf{KeyHit}}$. No preprocessing-fixed RO input can then coincide with a signing point or with the forgery hash input for an honest target. The signing simulation can fail only if a point that it attempts to program has already been defined online; there is no zero-challenge failure in the key-prefixed scheme. The same programming-collision argument as in Theorem 8 therefore gives $\Pr[\mathsf{FailtoSign}] \le \frac{q_{\mathsf{S}}^{\mathsf{on}}(q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}})}{p}$. Likewise, because preprocessing-fixed RO points are excluded by $\overline{\mathsf{KeyHit}}$, the early-RO argument gives $\Pr[\mathsf{EarlyRO}] \le \frac{q_{\mathsf{H}}^{\mathsf{on}}(N + \overline{q}_{\mathsf{G}}^{\mathsf{on}})}{p - (N + P_{\mathsf{G}} + 3\overline{q}_{\mathsf{G}}^{\mathsf{on}} + 1)}$. The no-query case is unchanged: $\Pr[\mathsf{NoQuery}] \le 2^{-\lambda_{\mathsf{H}}}$.

It remains to bound $\mathsf{BadQuery}$. As in Theorem 8, the critical RO coordinate cannot have been first defined by the signing simulation. Indeed, the public-key prefix fixes the signing user to j^*; outside the bridge event, equality of the signing and forgery representations then forces the same response and the same scalar s, contradicting freshness. This argument also covers $e^* = 0$. Let $I^* = \tau(\mathbf{a} \cdot \mathbf{x} + b)$ be its symbolic representation when the critical RO response is sampled. Outside the bridge event, final verification requires $e^* = -\mathbf{a}[j^*]$. Unlike nonzero Schnorr, however, the RO input itself contains pk_{j^*}. Thus the target j^* is already fixed before the RO response is sampled. Consequently, there is at most one response consistent with a bridge-free forgery, namely $e^* = -\mathbf{a}[j^*]$; when $\mathbf{a}[j^*] = 0$, this unique response is simply $e^* = 0$. Hence, $\Pr[\mathsf{BadQuery}] \le \frac{q_{\mathsf{H}}^{\mathsf{on}}}{2^{\lambda_{\mathsf{H}}}}$. Finally, Lemma 3 gives the same bridge-finding bound as before. Combining the bounds for KeyHit, $\mathsf{FailtoSign}$, $\mathsf{NoQuery}$, $\mathsf{EarlyRO}$, and $\mathsf{BadQuery}$ proves the theorem. □

C.2 Key-Prefixed Schnorr in the AI-(ROM+GGM)

Unlike nonzero Schnorr, key-prefixed Schnorr does not use rejection sampling. Its combined query complexity is therefore deterministically bounded, so the BF-to-AI lifting applies directly, without the retry application of Section 4.3. Let $\mathcal{G}_{\mathsf{KP}}$ denote the adaptive-corruption signature forgery application for Π_{KP} defined by $\mathsf{SigForgeCor}$. Then we have $(T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{KP}}} \le q_{\mathsf{H}}^{\mathsf{on}} + q_{\mathsf{S}}^{\mathsf{on}} + 1$ and $(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{KP}}} \le q_{\mathsf{G}}^{\mathsf{on}} + N + q_{\mathsf{S}}^{\mathsf{on}} + 1$, where the N term comes from key generation and the final $+1$ terms from forgery verification. Now, we have the following lifting theorem.

Theorem 13. *Let $p > 2^{2\lambda}$ be prime and suppose that $2^{\lambda_{\mathsf{H}}} < p$. Let $\mathcal{A}_{\mathsf{Sig}} = \left(\mathcal{A}^{\mathsf{AI\text{-}RO+GG}}_{\mathsf{Sig.pre}}, \mathcal{A}^{\mathsf{AI\text{-}RO+GG}}_{\mathsf{Sig.on}}\right)$ be an auxiliary-input adversary such that $\mathcal{A}_{\mathsf{Sig.pre}}$ outputs an S-bit hint. Suppose that $\mathcal{A}_{\mathsf{Sig.on}}$ makes at most $q^{\mathsf{on}}_{\mathsf{H}}$ random-oracle queries, $q^{\mathsf{on}}_{\mathsf{G}}$ primitive generic-group queries (i.e., `Mult` and `Inv`), and $q^{\mathsf{on}}_{\mathsf{S}}$ signing queries, and let $q \coloneqq q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{G}} + q^{\mathsf{on}}_{\mathsf{S}}$, $Q_{\mathsf{H}} \coloneqq q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{S}} + 1$, and $Q_{\mathsf{G}} \coloneqq q^{\mathsf{on}}_{\mathsf{G}} + N + q^{\mathsf{on}}_{\mathsf{S}} + 1$. Define $\overline{q}^{\mathsf{on}}_{\mathsf{G}} \coloneqq q^{\mathsf{on}}_{\mathsf{G}} + (q^{\mathsf{on}}_{\mathsf{S}} + 1)(2\lceil \log_2 p \rceil + 2\lambda_{\mathsf{H}} + 2)$ and $D_{\mathsf{sig}} \coloneqq p - 2\lceil 6(S + 2\lambda)Q_{\mathsf{G}} \rceil (N + 3\overline{q}^{\mathsf{on}}_{\mathsf{G}} + 1) - (N + 3\overline{q}^{\mathsf{on}}_{\mathsf{G}} + 1)^2 - N$. Assume that $D_{\mathsf{sig}} > 0$. Then, $\Pr\left[\mathsf{SigForgeCor}^{\tau,\mathsf{H},N}_{\mathcal{A}_{\mathsf{Sig.on}},\Pi_{\mathsf{KP}}}(\lambda) = 1\right] \le \varepsilon$, with*

$$\begin{aligned}
\varepsilon &= \frac{2\binom{N}{2} + 2N(\lceil 6(S+2\lambda)Q_{\mathsf{G}} \rceil + 1)}{p} + \frac{2\overline{q}^{\mathsf{on}}_{\mathsf{G}}(N + \lceil 6(S+2\lambda)Q_{\mathsf{G}} \rceil) + 3\overline{q}^{\mathsf{on}}_{\mathsf{G}}(\overline{q}^{\mathsf{on}}_{\mathsf{G}} + 1)}{D_{\mathsf{sig}}} + 2^{-2\lambda+1} \\
&\quad + \frac{2N\lceil (S+2\lambda)Q_{\mathsf{H}} \rceil}{p} + \frac{2q^{\mathsf{on}}_{\mathsf{S}}(q^{\mathsf{on}}_{\mathsf{H}} + q^{\mathsf{on}}_{\mathsf{S}})}{p} + \frac{2q^{\mathsf{on}}_{\mathsf{H}}(N + \overline{q}^{\mathsf{on}}_{\mathsf{G}})}{p - (N + \lceil 6(S+2\lambda)Q_{\mathsf{G}} \rceil + 3\overline{q}^{\mathsf{on}}_{\mathsf{G}} + 1)} + \frac{q^{\mathsf{on}}_{\mathsf{H}} + 1}{2^{\lambda_{\mathsf{H}}-1}} \\
&= O\left(\frac{(S+\lambda)(N+q)(N + q\log p) + q^2 \log^2 p}{p} + \frac{q}{2^{\lambda_{\mathsf{H}}}} + 2^{-2\lambda}\right),
\end{aligned}$$

where the asymptotic bound holds when $D_{\mathsf{sig}} = \Omega(p)$ and the probability is over the ideal oracles and all random coins of the experiment and the adversary.

Proof. Recall that $(T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{KP}}} \le Q_{\mathsf{H}}$ and $(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{KP}}} \le Q_{\mathsf{G}}$. Hence, for $\gamma > 0$, we can choose $P_{\mathsf{H}} = \lceil (S + \log \gamma^{-1})Q_{\mathsf{H}} \rceil$ and $P_{\mathsf{G}} = \lceil 6(S + \log \gamma^{-1})Q_{\mathsf{G}} \rceil$. Then since $p > 2^{2\lambda}$, we have $\log p / \log(p/e) \le 2$ and hence,

$$(S + \log \gamma^{-1}) \left((T_1)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{KP}}} \lambda_{\mathsf{H}} + 3(T_2)^{\mathsf{comb}}_{\mathcal{G}_{\mathsf{KP}}} \log p\right) \le P_{\mathsf{H}} \lambda_{\mathsf{H}} + P_{\mathsf{G}} \log(p/e).$$

Therefore, we can apply Theorem 5 to Theorem 12 with $\gamma = 2^{-2\lambda}$ and get the desired result. □

Parameter Consequence. Key-prefixing removes exactly the adaptive target-switching loss identified in Section 5.1. The short-challenge term is $O\left(\frac{q}{2^{\lambda_{\mathsf{H}}}}\right)$, rather than $O\left(\frac{q \min\{N,q\}}{2^{\lambda_{\mathsf{H}}}}\right)$ for nonzero Schnorr under adaptive corruption. Consequently, under the same parameter regime as in Section 4.4, one may again take $\lambda_{\mathsf{H}} \approx \lambda$ instead of $\lambda + \log N$. For example, with $\lambda = 128$, $S = 2^{72}$, and $N = 2^{32}$, the choice $\log p \approx 335$ and $\lambda_{\mathsf{H}} \approx 128$ gives a key-prefixed signature of approximately 463 bits, compared with approximately 495 bits for nonzero Schnorr under adaptive corruption.

D Missing Proofs

Reminder of Lemma 1. *Let $p > 2^{2\lambda}$ be a prime, and $N, P_{\mathsf{H}}, P_{\mathsf{G}}$ be nonnegative integers. Let $\mathcal{A} \coloneqq \left(\mathcal{A}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathsf{G}})}_{\mathsf{pre}}, \mathcal{A}^{\mathsf{BF\text{-}RO+GG}(P_{\mathsf{H}},P_{\mathsf{G}})}_{\mathsf{on}}\right)$ be a bit-fixing adversary for the multi-user bridge-finding game in Figure 3. Suppose that $\mathcal{A}_{\mathsf{on}}$ makes at most $q^{\mathsf{on}}_{\mathsf{H}}$ random-oracle queries and at most $q^{\mathsf{on}}_{\mathsf{G}}$ generic-group queries. Define $D_{\mathsf{br}} \coloneqq p - 2P_{\mathsf{G}}(N + 3q^{\mathsf{on}}_{\mathsf{G}} + 1) - (N + 3q^{\mathsf{on}}_{\mathsf{G}} + 1)^2 - N$ and assume that $D_{\mathsf{br}} > 0$. Then*

$$\Pr\left[\mathsf{BridgeChal}^{\tau,\mathsf{H},N}_{\mathcal{A}}(\lambda) = 1\right] \le \frac{\binom{N}{2} + N(P_{\mathsf{G}} + 1)}{p} + \frac{q^{\mathsf{on}}_{\mathsf{G}}(N + P_{\mathsf{G}}) + 3q^{\mathsf{on}}_{\mathsf{G}}(q^{\mathsf{on}}_{\mathsf{G}} + 1)/2}{D_{\mathsf{br}}}.$$

The probability is over all random choices in the experiment, including the completion of H and τ, the challenge secrets, and the random coins of $\mathcal{A}_{\mathsf{pre}}$ and $\mathcal{A}_{\mathsf{on}}$.

Proof. Let InitBr denote the event that the initial list already contains a bridge. By injectivity of τ, this event can occur only if two challenge secrets coincide or if some challenge secret belongs to $\{1\} \cup \mathrm{domain}(F_{\mathtt{G}})$. Therefore, $\Pr[\mathsf{InitBr}] \le \frac{\binom{N}{2}+N(P_{\mathtt{G}}+1)}{p}$.

We now condition on $\overline{\mathsf{InitBr}}$, on an admissible preprocessing view $(\mathcal{Y}, \mathfrak{g}, F_{\mathsf{H}}, F_{\mathtt{G}}, z)$, on the completed random oracle H, and on the random coins of the adversary. Under this conditioning, the online adversary is a fixed adaptive strategy, while the remaining labeling map is uniform over $\mathcal{I}_{\mathcal{Y}}[F_{\mathtt{G}} \cup \{(1, \mathfrak{g})\}]$. It therefore suffices to analyze the symbolic collisions created by the online generic-group queries, starting from the $P_{\mathtt{G}}$ fixed GGM records.

For $i \in [q_{\mathtt{G}}^{\mathsf{on}}]$, let $\overline{\mathsf{BRIDGE}}_{<i}^{N}$ denote the event that no bridge has occurred until the $(i-1)^{th}$ online generic group query, and let B_i denote the event that the ith query creates a bridge. Condition on $\overline{\mathsf{BRIDGE}}_{<i}^{N}$. Before the response $\mathfrak{y}_i$ to the ith generic-group query is sampled, the coefficients $\mathbf{a}_i, b_i$ of the tuple $(\mathfrak{y}_i, \mathbf{a}_i, b_i)$ to be recorded are already determined. If this exact tuple already belongs to $\mathcal{L}$, no bridge is created. Otherwise, B_i can occur only if there is a previously recorded tuple $(\cdot, \mathbf{a}, b) \in \mathcal{L}$ with $(\mathbf{a}, b) \neq (\mathbf{a}_i, b_i)$ such that $(\mathbf{a} - \mathbf{a}_i) \cdot \mathbf{x} = b_i - b$. If $\mathbf{a} = \mathbf{a}_i$, this equality is impossible because $b \neq b_i$. Otherwise, choose an index $r \in [N]$ such that $a[r] - a_i[r] \neq 0$ and expose x_r last.

Define $\mathcal{L}_0 := \{(\mathfrak{y}, \mathbf{a}, b) \in \mathcal{L} : \mathbf{a} = \mathbf{0}\}$ and $\mathcal{L}_1 := \mathcal{L} \setminus \mathcal{L}_0$. Conditioning on the absence of previous bridges imposes constraints of the form $\mathbf{a}_j \cdot \mathbf{x} + b_j \neq \mathbf{a}_k \cdot \mathbf{x} + b_k$ for distinct recorded labels. For fixed values of all coordinates except x_r, any such constraint whose coefficient of x_r is nonzero excludes at most one value of x_r. Pairs of tuples both belonging to $\mathcal{L}_0$ exclude no value, since both coefficient vectors are zero. In addition, the challenge labels are distinct under $\overline{\mathsf{InitBr}}$, contributing at most $N-1$ further excluded values. Hence x_r remains uniform over a set of size at least $p - \left(|\mathcal{L}|^2 - |\mathcal{L}_0|^2\right) - (N-1)$.

Write $p_{\mathtt{G}} := |F_{\mathtt{G}}| \le P_{\mathtt{G}}$. We have $|\mathcal{L}| \le N + p_{\mathtt{G}} + 3q_{\mathtt{G}}^{\mathsf{on}} + 1$ and $|\mathcal{L}_0| \ge p_{\mathtt{G}}$, since the list initially contains the $p_{\mathtt{G}}$ fixed generic-group records and each generic group oracle query adds at most three new tuples to $\mathcal{L}$ — exactly three when $\mathtt{Mult}(\mathfrak{y}_1, \mathfrak{y}_2)$ is queried on two fresh input labels. Thus, the probability that $(\mathbf{a} - \mathbf{a}_i) \cdot \mathbf{x} = b_i - b$ is at most $\frac{1}{p-(N+p_{\mathtt{G}}+3q_{\mathtt{G}}^{\mathsf{on}}+1)^2+p_{\mathtt{G}}^2-(N-1)} \le \frac{1}{D_{\mathsf{br}}}$. Indeed, the denominator on the left-hand side is equal to $D_{\mathsf{br}} + 2(P_{\mathtt{G}} - p_{\mathtt{G}})(N + 3q_{\mathtt{G}}^{\mathsf{on}} + 1) + 1$, which is at least D_{br}. Union bounding over all tuples $(\cdot, \mathbf{a}, b) \in \mathcal{L}$, we have

$$\Pr\left[B_i \,\middle|\, \overline{\mathsf{BRIDGE}}_{<i}^{N}\right] \le \frac{N + p_{\mathtt{G}} + 3i}{D_{\mathsf{br}}} \le \frac{N + P_{\mathtt{G}} + 3i}{D_{\mathsf{br}}}.$$

Finally, union bounding over initialization and the online generic group queries,

$$\begin{aligned}
\Pr\left[\mathsf{BridgeChal}_{\mathcal{A}}^{\tau,\mathsf{H},N}(\lambda) = 1\right] &\le \Pr[\mathsf{InitBr}] + \sum_{i=1}^{q_{\mathtt{G}}^{\mathsf{on}}} \Pr\left[B_i \cap \overline{\mathsf{BRIDGE}}_{<i}^{N}\right] \\
&\le \frac{\binom{N}{2} + N(P_{\mathtt{G}}+1)}{p} + \sum_{i=1}^{q_{\mathtt{G}}^{\mathsf{on}}} \frac{N + P_{\mathtt{G}} + 3i}{D_{\mathsf{br}}} \\
&= \frac{\binom{N}{2} + N(P_{\mathtt{G}}+1)}{p} + \frac{q_{\mathtt{G}}^{\mathsf{on}}(N + P_{\mathtt{G}}) + 3q_{\mathtt{G}}^{\mathsf{on}}(q_{\mathtt{G}}^{\mathsf{on}}+1)/2}{D_{\mathsf{br}}}\,.
\end{aligned}$$

□