

Hidden-Bits Generators with Succinct CRS and Openings

Pedro Branco¹, Nico Döttling², Yao-Ching Hsieh³, Abhishek Jain⁴, Akshayaram Srinivasan⁵, and Brent Waters⁶

¹IE University

²Helmholtz Center for Information Security (CISPA)

³University of Washington

⁴NTT Research & JHU

⁵University of Toronto

⁶NTT Research & UT Austin

Abstract

Feige, Lapidot, and Shamir [FOCS'90] presented a generic approach for building non-interactive zero-knowledge proofs via the notion of hidden-bits generators (HBGs). At a high level, an HBG allows a prover to statistically commit to a k -bit pseudorandom string in the CRS model with a succinct commitment. Later, the prover can open this pseudorandom string at an arbitrary subset $S \subseteq [k]$ of positions by providing a proof π_S . The hiding property requires that unopened positions remain pseudorandom.

In this work, we study the efficiency of HBGs, focusing on *direct* constructions that make black-box use of cryptography. We obtain the following results:

- **Succinct CRS:** Assuming learning with errors (LWE), we construct an HBG with *succinct* CRS of size $\text{poly}(\lambda, \log k)$ for generating k hidden bits. This achieves an exponential improvement over a recent sequence of works by Waters [STOC'24], Branco et al. [EUROCRYPT'25], and Waters, Wee, Wu [EUROCRYPT'25] that required a CRS of size at least $k \cdot \text{poly}(\lambda)$.
- **Batch opening:** Using pairings, we construct an HBG with *batch* opening, where the proof size for any subset is $\text{poly}(\lambda)$. Previously, this was only known from the Naccache-Stern public-key cryptosystem [Groth, ASIACRYPT'10].
- **Succinct CRS and Batch opening:** Assuming the hardness of RSA, we construct an HBG where *both* the CRS and the opening proof for any subset are of size $\text{poly}(\lambda, \log k)$. No black-box constructions achieving this efficiency were previously known.

1 Introduction

Zero-knowledge (ZK) proofs [GMR85] allow a prover to convince a verifier about the validity of a statement without revealing any additional information. Initially envisioned as an interactive proof system [GMR85, Bab85], ZK proofs were subsequently studied in the non-interactive model where the proof consists of a single message from the prover to the verifier. This notion is referred to as *non-interactive zero knowledge* (NIZK) [DMP88, BFM88, FLS90], and is typically studied in the common reference string (CRS) model where both the prover and the verifier receive a common string sampled from some fixed distribution.

NIZKs via Hidden-Bits Generators. Feige, Lapidot, and Shamir (FLS) [FLS90] proposed a generic approach to NIZKs that consists of two steps: the first step involves the design of unconditionally-secure NIZKs in an idealized, so-called hidden-bits model, and the second step involves a transformation from this

model to statistically-sound NIZKs (aka, NIZK *proofs*) in the CRS model via the cryptographic notion of *hidden-bits generator* (HBG) [QRW19]. Informally speaking, an HBG allows one to generate a succinct commitment to a pseudorandom string r of size k and later release verifiable openings of bits of r . The key properties are statistical binding, namely, the commitment cannot information-theoretically encode “too many” strings, and computational hiding, namely, the unopened bits of r remain hidden even given the openings of the other bits.

An attractive feature of HBG is that it can yield (single-theorem) NIZKs that make *black-box* use of cryptography. This is in contrast to the correlation-intractable hash (CIH) [CGH04] framework for NIZKs, which inherently makes non-black-box use of cryptography [CCRR18, HL18, CCH⁺19, PS19, CKU20, BKM20, JJ21, DJJ24]. The only other approach to black-box NIZKs requires bilinear maps [GOS06b, GOS06a], and this is currently not known to be instantiable from any other assumptions.

Direct constructions of HBGs are known from several assumptions, including RSA (and more generally, enhanced trapdoor permutations) [FLS90], bilinear maps [CHK03], learning with errors (LWE) [Wat24, BCD⁺25, WWW25], and the joint hardness of decisional Diffie-Hellman and learning parity with noise [BCD⁺25]. This implies black-box NIZKs from the same set of assumptions. By making non-black-box use of cryptography, HBGs can also be indirectly constructed from any statistically-sound NIZK and one-way functions. The construction is simple: commit to the key of a pseudorandom function (PRF) and open any bit by providing a NIZK proof. This, however, requires non-black-box use of the commitment and the PRF.

Efficiency of Hidden-Bits Generators. In this work, we study the efficiency of hidden-bits generators, focusing on direct constructions that make black-box use of cryptography. There are two parameters of interest for HBGs: the size of the CRS and the size of the openings. In the transformation from HBG to NIZK proofs, the size of the HBG CRS (resp., openings) impacts the size of the NIZK CRS (resp., proof). Toward the goal of constructing efficient NIZK proofs, it is desirable to minimize both of these parameters.

- **CRS size:** All known direct constructions of HBGs require a CRS of size at least $k \cdot \text{poly}(\lambda)$ for generating k hidden bits. Ideally, we want the CRS size to grow only polylogarithmically in k and we refer to this as a succinct CRS. We ask:

Do there exist direct constructions of HBG with a succinct CRS?

- **Opening size:** Most known constructions of HBG require at least $\text{poly}(\lambda)$ communication for *each* opened bit, where λ is the security parameter. Assuming the security of Naccache-Stern PKE [NS98], Groth [Gro10] constructed the first HBG with *batch opening*, where any subset of bits can be opened with communication independent of the subset size. We ask:

Do there exist direct constructions of HBG with batch opening from other assumptions?

- **Succinct CRS and batch openings:** A natural question is to simultaneously achieve the best of both worlds: an HBG with succinct CRS and batch openings.

Do there exist direct constructions of HBG with succinct CRS and batch openings?

If we instantiate the indirect construction of HBG mentioned above with rate-1 NIZK proofs from the work of Gentry et al. [GGI⁺15], we can obtain an HBG with succinct CRS and batch openings. However, this instantiation makes *non-black-box* use of a fully homomorphic encryption scheme. As mentioned earlier, in this work, we focus on direct constructions.

1.1 Our Results

In this work, we resolve all the above questions in the affirmative.

HBG with Succinct CRS. Our first result is a construction of an HBG with a succinct CRS from LWE [Reg05] with polynomial modulus-to-noise ratio.

Theorem 1. *Assuming LWE with polynomial modulus-to-noise ratio, there is a direct construction of an HBG for generating k bits with a transparent CRS of size $\text{poly}(\lambda, \log k)$.*

Prior constructions of HBGs from lattices [Wat24, WWW25, BCD⁺25] required a CRS of size at least $k \cdot \text{poly}(\lambda)$. In fact, before our work, this was the best-known CRS size achieved by all known direct constructions of HBGs, regardless of the computational assumptions.

HBG with Batch Opening. Our next result is a black-box construction of an HBG from bilinear maps that achieves batch opening. The security of our construction relies on the computational Bilinear Diffie-Hellman (CBDH) assumption.

Theorem 2 (Informal). *Assuming the hardness of computational Bilinear Diffie-Hellman over pairing groups, there is a black-box construction of an HBG with opening size $\text{poly}(\lambda)$ for any subset of hidden bits.*

Previously, such a direct construction was only known [Gro10] from the security of Naccache-Stern PKE [NS98]. All other direct constructions of HBGs required opening size $\text{poly}(\lambda)$ for *each* hidden bit.

The size of the CRS of the HBG in the above theorem grows as $k^2 \cdot \text{poly}(\lambda)$, where k is the number of hidden bits. Under a falsifiable q -type assumption referred to as Assumption 2 (formally defined in Section 6), we give an improved construction where the CRS size is $k \cdot \text{poly}(\lambda)$. In Appendix A, we prove that the assumption holds in the generic group model.

Theorem 3 (Informal). *Suppose Assumption 2 holds. There exists a black-box construction of an HBG with opening size $\text{poly}(\lambda)$ for any subset of hidden bits and CRS size $k \cdot \text{poly}(\lambda)$, where k is the number of hidden bits.*

HBG with Succinct CRS and Batch Opening. Finally, we give a direct construction of an HBG that simultaneously achieves succinct CRS and batch openings. This construction is based on the RSA assumption.

Theorem 4 (Informal). *Assuming the hardness of the RSA problem, there is a black-box construction of HBG with a CRS size $\text{poly}(\lambda, \log k)$ and an opening size $\text{poly}(\lambda)$ for any subset of hidden bits.*

From HBG to NIZK Proofs. The original transformation from HBG to NIZK proofs given by Feige, Lapidot, and Shamir [FLS90] was rather inefficient and yielded NIZK proofs with CRS and proof size polynomial in the size of the NP relation circuit. Following [KP98], Groth [Gro10] presented an improved transformation that requires $\tilde{O}(|C|)$ hidden bits (as opposed to $\text{poly}(|C|)$), where C is the NP relation circuit and $\tilde{O}$ notation hides polylogarithmic factors. This constitutes the state of the art in this line of work. Applying this transformation to our HBG constructions from Theorems 1, 2, 3, and 4, we obtain new and improved constructions of NIZK proofs.

Corollary 1 (Informal). *We have the following:*

- *Assuming LWE with polynomial modulus-to-noise ratio, there exists a NIZK proof system in which the size of the CRS is $\tilde{O}(|C|) + \text{poly}(\lambda, \log |C|)$ and the size of the proof is $|C| \cdot \text{poly}(\lambda, \log |C|)$.*
- *Assuming the hardness of Computational Bilinear Diffie-Hellman over pairing groups, there exists a black-box NIZK proof system in which the size of the proof is $\tilde{O}(|C|) + \text{poly}(\lambda)$ and the size of the CRS is $|C|^2 \cdot \text{poly}(\lambda, \log |C|)$. Furthermore, under the hardness of Assumption 2, the size of the CRS can be reduced to $|C| \cdot \text{poly}(\lambda, \log |C|)$.*

- Assuming the hardness of the RSA problem, there exists a black-box NIZK proof system in which the size of the CRS and the size of the proof are both $\tilde{O}(|C|) + \text{poly}(\lambda, \log |C|)$.

Remark 1 (On the size of the CRS). *One would expect the CRS for a NIZK proof constructed from an HBG with succinct CRS to also be succinct. Unfortunately, this is not the case. The transformation from NIZKs in the hidden-bits model to NIZKs in the CRS model via HBGs [FLS90, QRW19] additionally requires the NIZK CRS to contain a random string of k bits (where k is the number of hidden bits). Currently, it is not known how to remove these additional bits from the CRS. We note that any improvement on this transformation would lead to a better CRS size for NIZKs when starting with our HBG.*

We now compare the above results with prior work. Our LWE-based construction of NIZK proofs achieves CRS size $\tilde{O}(|C|) + \text{poly}(\lambda, \log |C|)$, where C is the NP relation circuit. That is, the CRS size has *additive overhead* in the security parameter. All prior constructions of NIZK proofs via HBGs required CRS size at least $|C| \cdot \text{poly}(\lambda, \log |C|)$, that is, with *multiplicative overhead* in the security parameter. While NIZK proofs with shorter CRS (even size independent of $|C|$) exist from LWE via CI hashing [PS19], they make *non-black-box* use of trapdoor commitment schemes. On the other hand, our NIZK construction does not require such non-black-box use of cryptographic primitives. Furthermore, our HBG construction is simple and direct; in particular, it does not require any homomorphic properties that seem inherent in the CI hashing framework.

Our bilinear-map-based construction of NIZK proofs achieves proof size $\tilde{O}(|C|) + \text{poly}(\lambda)$. Previously, the best-known result from pairings due to [GOS06b, GOS06a] incurred a *multiplicative* overhead in λ , i.e., its proof size is $O(|C|\lambda)$. Furthermore, black-box NIZK proofs with our proof size were previously known only from the security of the Naccache-Stern cryptosystem [Gro10].

Finally, our RSA-based construction is the first black-box construction of NIZK proofs where both the CRS and proof are of size quasilinear in the NP relation circuit size. To the best of our knowledge, no black-box NIZK proof matching this efficiency was previously known from any assumption.

See Table 1 for a comparison of our results with the prior state-of-the-art black-box NIZK constructions.

	Proof size	CRS size	Assumption
[FLS90, Gro10]	$\tilde{O}(C) \cdot \lambda$	$\tilde{O}(C) \cdot \lambda$	Enhanced TDP
[GOS06b, GOS06a]	$O(C \cdot \lambda)$	$\text{poly}(\lambda)$	Subgroup Decision or DLIN
[Gro10]	$\tilde{O}(C) + \text{poly}(\lambda)$	$\tilde{O}(C) \cdot \lambda$	Naccache-Stern [NS98]
Corollary 3	$\tilde{O}(C) + \text{poly}(\lambda)$	$\tilde{O}(C ^2) \cdot \lambda$	CBDH
Corollary 4	$\tilde{O}(C) + \text{poly}(\lambda)$	$\tilde{O}(C) \cdot \lambda$	Assumption 2
Corollary 5	$\tilde{O}(C) + \text{poly}(\lambda)$	$\tilde{O}(C) + \text{poly}(\lambda, \log C)$	RSA

Table 1: Proof size and CRS size of statistically-sound black-box NIZKs.

1.2 Related Work

There is an extensive body of work studying the feasibility and efficiency of NIZKs. We do not aim for a comprehensive survey here; instead, we briefly discuss the works most relevant to ours.

Statistically-Sound NIZKs. Feige et al. [FLS90] constructed an HBG from enhanced trapdoor permutations with CRS and opening size $k \cdot \text{poly}(\lambda)$ for generating k bits. Furthermore, they presented a compiler from HBGs to NIZK proofs that required $O(\lambda \cdot |C|^{11/2})$ hidden bits, where C is the NP relation circuit. This resulted in NIZK proofs with proof and CRS sizes polynomial in the NP relation circuit size. Subsequent work achieved improved parameters from various assumptions; see [KP98] and references therein. The state-of-the-art efficiency in this line of work was achieved by Groth [Gro10], who constructed HBGs with batch

opening and NIZK proofs with quasilinear size (i.e., $\tilde{O}(|C|) + \text{poly}(\lambda)$) from the Naccache-Stern cryptosystem [NS98]. Finally, Gentry et al. [GGI⁺15] presented a generic construction of rate-1 NIZK proofs, i.e., with proof size $|w| + \text{poly}(\lambda)$ (where $|w|$ is the witness size) from fully-homomorphic encryption and any NIZK proof. In contrast to the schemes discussed above, this construction makes *non-black-box* use of the FHE scheme.

All of the above constructions, except [GGI⁺15], require a CRS of size at least $|C| \cdot \text{poly}(\lambda, \log |C|)$. Black-box NIZK proofs from LWE with the same CRS size were recently constructed in [WWW25] (building on [Wat24]). NIZK proofs with CRS size independent of the circuit size, i.e., $\text{poly}(\lambda)$, are currently known only via approaches different from HBG: either via a direct construction from bilinear maps [GOS06b], from FHE [GGI⁺15], or via the correlation-intractability framework based on LWE [PS19]. The last two approaches require non-black-box use of cryptography.

Computationally-Sound NIZKs. In the case of computationally-sound NIZKs, aka NIZK arguments, short proof size can be achieved from many assumptions. Katsumata *et al.* [KNYY19, KNYY20] presented NIZK schemes with proof size $|C| + \text{poly}(\lambda)$, first from the computational DH exponent and ratio assumption (CDHER, a non-standard q -type assumption on pairing groups) [KNYY19], and later from the DLIN assumption [KNYY20]. More recently, Afshar *et al.* [ACG⁺24] and Branco, Döttling, and Srinivasan [BDS25] constructed rate-1 NIZKs, that is, where the proof size is asymptotically the same as the witness size. The work of [ACG⁺24] provides a construction from batch arguments (BARGs) [CJJ21, CJJ22, HJKS22, WW22, CGJ⁺23] achieving computational soundness and zero-knowledge. On the other hand, the construction of [BDS25] relies on sub-exponential security of either LWE, DLIN, or DDH and achieves statistical ZK.

2 Technical Overview

NIZKs in the Hidden-Bits Model. The hidden-bits model is an idealized model where: i) the prover has full access to a binary string while this string remains hidden to the verifier, and ii) the prover can reveal to the verifier the bits at a subset of the positions of this string, while the remaining bits stay hidden. It is well known that, in this idealized model, we can build NIZKs for all NP unconditionally [FLS90].

Hidden-bits generators. In order to build NIZKs in the common reference string model, Feige, Lapidot, and Shamir [FLS90] introduced a primitive called a hidden-bits generator (HBG), which was later formalized by Quach, Rothblum, and Wichs [QRW19]. An HBG consists of:

- a setup algorithm **Setup** that generates a common reference string (CRS);
- a generation algorithm **GenBits** that, given the CRS, outputs k pseudorandom bits $\alpha_1, \dots, \alpha_k$, along with a “short” commitment **com** that statistically binds to these bits, and corresponding local proofs $\pi_1, \dots, \pi_k$;
- a verification algorithm **Verify** that, given the CRS, the commitment **com**, a bit α_i , and its proof π_i , checks that α_i is consistent with the commitment.

Intuitively, an HBG emulates the hidden-bits model: the prover can generate pseudorandom bits together with proofs that allow individual bits to be revealed when needed. If a proof is not revealed, the corresponding bit remains hidden, while the commitment statistically binds the prover to a particular bit string. The crucial property that enables us to transform a NIZK proof from the hidden-bits model to the CRS model is the *succinctness of the commitment*. Specifically, we require the size of the commitment to be sublinear in k .

In this work, we focus on minimizing the size of the CRS and batching the local proofs. In Section 2.1, we explain how to achieve a succinct CRS (i.e., with size $\text{poly}(\lambda, \log k)$) from LWE. In Section 2.2, we show how to use pairings to batch local proofs and achieve succinct openings. In Section 2.3, we show how to achieve both succinct CRS and succinct openings from the RSA assumption.

2.1 HBG with Succinct CRS from LWE

Before we discuss our construction of an HBG with succinct CRS, we first describe a construction with non-compact CRS. This construction builds upon ideas introduced in [Wat24] and expanded upon in [BCD⁺25, WWW25]. We will later show how to compress the CRS to achieve succinctness, and this is the main technical contribution of this subsection.

The Base Scheme. Our starting point is the basic SIS hash function, which is also the basis for the HBG constructions in the above-mentioned works. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a uniformly chosen matrix. To hash an $\mathbf{x} \in \{0, 1\}^m$, compute $\mathbf{h} = \mathbf{A} \cdot \mathbf{x} \in \mathbb{Z}_q^n$. We can now define a basic HBG with respect to $\mathbf{A}$ as follows. Compute an HBG key (which is part of the CRS) $\mathbf{d} = \mathbf{s}\mathbf{A} + \mathbf{e}$, where $\mathbf{s} \in \mathbb{Z}_q^n$ is an LWE secret and $\mathbf{e} \in \mathbb{Z}^m$ is an LWE error term chosen from a short distribution.

For the sake of presentation, in the following we will think of the modulus q as superpolynomial. We will later discuss the modifications needed in the polynomial modulus-to-noise regime.

For a commitment $\mathbf{h}$ with opening $\mathbf{x}$, we can now define the corresponding hidden bit α via $\alpha = \lceil \mathbf{d}\mathbf{x} \rceil$, where $\lceil \cdot \rceil$ is the rounding function that retrieves the most significant bit¹. We argue that $\mathbf{h}$ *statistically commits* to α , irrespective of $\mathbf{x}$, as it holds that

$$\alpha = \lceil \mathbf{d} \cdot \mathbf{x} \rceil = \lceil \mathbf{s}\mathbf{A}\mathbf{x} + \underbrace{\mathbf{e}\mathbf{x}}_{\text{short}} \rceil =_{\text{whp}} \lceil \mathbf{s}\mathbf{h} \rceil.$$

In other words, there is an equivalent private way of computing α via $\alpha = \lceil \mathbf{s}\mathbf{h} \rceil$ that does not depend on the opening $\mathbf{x}$ but only on $\mathbf{h}$ and the secret $\mathbf{s}$.

Notice that the key feature that enables establishing this fact is that the opening is linear. If $\mathbf{x} \in \{0, 1\}^m$ is chosen uniformly at random, we can easily argue that α is pseudorandom given only $\mathbf{h}$ and $\mathbf{d}$ by relying on LWE: under the decisional LWE assumption, it immediately holds that we can replace $\mathbf{d}$ with a uniformly random value, hence it follows that $\mathbf{d} \cdot \mathbf{x}$ is statistically close to uniform given $\mathbf{d}$ and $\mathbf{h} = \mathbf{A}\mathbf{x}$ via the leftover hash lemma².

Now, recall that a critical feature of an HBG is that the commitment (in our case $\mathbf{h}$) is succinct; in particular, the size of the commitment should be considerably smaller than the number of hidden bits it commits to. Hence, repeating this in parallel, i.e., computing hashes $\mathbf{h}_1 = \mathbf{A}\mathbf{x}_1, \dots, \mathbf{h}_k = \mathbf{A}\mathbf{x}_k$ does not yield an HBG, as this makes the size of the commitment $\text{com} = (\mathbf{h}_1, \dots, \mathbf{h}_k)$ grow linearly with the number of hidden bits k .

The prior works [Wat24, BCD⁺25, WWW25] took a different approach than ours to compress the commitment and make it succinct. Here, we take a different route that will later help compress the public parameters.

Succinct Commitments. In order to compress these commitments, we will resort to a technique proposed in [DKL⁺23], which (implicitly) allows one to construct Merkle tree commitments with *linear local opening*. The basic idea is to use a small but crucial variation of the SIS hash function discussed above as a 2-to-1 Merkle tree compression function. In the following, let $\mathbf{G} \in \mathbb{Z}_q^{n \times n \log(q)}$ be the well-known gadget matrix and let $\mathbf{G}^{-1} : \mathbb{Z}_q^n \rightarrow \{0, 1\}^{n \log(q)}$ be the binary decomposition operation, i.e., it holds for every $\mathbf{z} \in \mathbb{Z}_q^n$ that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{z}) = \mathbf{z}$. From now on let $m = 2n \log(q)$.

Now, to compress two strings $\mathbf{z}_0, \mathbf{z}_1 \in \mathbb{Z}_q^n$ into a single $\mathbf{h} \in \mathbb{Z}_q^n$, compute

$$\pi = \pi(\mathbf{z}_0, \mathbf{z}_1) := \begin{pmatrix} \mathbf{G}^{-1}(\mathbf{z}_0) \\ \mathbf{G}^{-1}(\mathbf{z}_1) \end{pmatrix} \in \{0, 1\}^m$$

and set $\mathbf{h} = \mathbf{A} \cdot \pi$.

¹For the sake of this exposition, we can think of the rounding function as having a built-in random offset δ , such that any close values $x \approx x'$ round to the same value, except with small probability over the choice of the offset

²For the sake of this outline, we will ignore the additional entropy overhead of the LHL and refer the reader to Section 4 where the strings $\mathbf{x}_i$ are slightly longer and are hashed with a different matrix $\mathbf{A}_0$

For convenience, we will define the shorthand $\mathbf{G}_0 = (\mathbf{G}, \mathbf{0}) \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{G}_1 = (\mathbf{0}, \mathbf{G}) \in \mathbb{Z}_q^{n \times m}$. Notice now that it holds for the *opening value* π defined above that $\mathbf{G}_0 \pi = \mathbf{z}_0$ and $\mathbf{G}_1 \pi = \mathbf{z}_1$. In other words, given the opening π , we can compute the hash value $\mathbf{h} = \mathbf{A} \pi$ and the two components of the preimage $\mathbf{z}_0 = \mathbf{G}_0 \pi$ and $\mathbf{z}_1 = \mathbf{G}_1 \pi$ via simple linear operations on π .

Now, let $k = 2^{d-1}$ be a power of two and let $\mathbf{x}_0, \dots, \mathbf{x}_{k-1} \in \{0, 1\}^m$. We can use the 2-to-1 compression function defined above to compress the vector $(\mathbf{x}_0, \dots, \mathbf{x}_{k-1}) \in \{0, 1\}^{km}$ into a single vector $\mathbf{h} \in \mathbb{Z}_q^n$ via Merkle hashing. Specifically, for $i \in [k]$ ³ compute $\mathbf{h}_{1,i} = \mathbf{A} \mathbf{x}_i$. Then, for $i \in [k/2]$ compute $\pi_{1,i} = \pi(\mathbf{h}_{1,2i}, \mathbf{h}_{1,2i+1})$, $\mathbf{h}_{2,i} = \mathbf{A} \pi_{1,i}$ and so forth, until we obtain $\mathbf{h}_{d,0} =: \mathbf{h}$.

We will now briefly show that this Merkle hash function allows for linear opening. Fix an index $i \in [k] = [2^{d-1}]$ and let $(i_2, \dots, i_d)$ be the binary decomposition of i for bits $i_j \in \{0, 1\}$, i.e., we think of i represented as the leaf-to-root path $(i_2, \dots, i_d)$. For bookkeeping purposes, we will prefix this with $i_1 = 0$ and then use $(i_1, \dots, i_d)$ to represent i .

Now define the matrix $\mathbf{A}^{(i)} = (\mathbf{A}, \mathbf{A} - \mathbf{G}_{i_2}, \dots, \mathbf{A} - \mathbf{G}_{i_d}) \in \mathbb{Z}_q^{n \times dm}$. For $(\mathbf{x}_0, \dots, \mathbf{x}_{k-1}) \in \{0, 1\}^{km}$ with Merkle hash $\mathbf{h}$ and an index $i \in [k]$, let $\mathbf{x}_i, \pi_1, \dots, \pi_{d-1}$ be the openings along the leaf-to-root path for i with hashes $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_d = \mathbf{h}$, i.e., it holds that $\mathbf{A} \mathbf{x}_i = \mathbf{h}_1$ and $\mathbf{A} \pi_j = \mathbf{h}_{j+1}$ for $j = 1, \dots, d-1$. Setting

$$\pi^{(i)} = \begin{pmatrix} \mathbf{x}_i \\ \pi_1 \\ \vdots \\ \pi_{d-1} \end{pmatrix},$$

we observe that

$$\begin{aligned} \mathbf{A}^{(i)} \pi^{(i)} &= \underbrace{\mathbf{A} \mathbf{x}_i}_{=\mathbf{h}_1} + \sum_{j=1}^{d-1} (\mathbf{A} - \mathbf{G}_{i_{j+1}}) \pi_j \\ &= \mathbf{h}_1 + \sum_{j=1}^{d-1} (\underbrace{\mathbf{A} \pi_j}_{=\mathbf{h}_{j+1}} - \underbrace{\mathbf{G}_{i_{j+1}} \pi_j}_{=\mathbf{h}_j}) \\ &= \mathbf{h}_d \\ &= \mathbf{h}, \end{aligned}$$

where the step from the second to the third equation is a simple telescoping sum. Thus, we have established that this Merkle hash function allows for *linear local opening*.

We can now extend the basic HBG recipe sketched above to this hash function. For each index $i \in [k]$, define a key

$$\mathbf{d}^{(i)} = (\mathbf{d}_1^{(i)} = \mathbf{s}_1^{(i)} \mathbf{A} + \mathbf{e}_1^{(i)}, \mathbf{d}_2^{(i)} = \mathbf{s}_2^{(i)} \mathbf{A} - \mathbf{s}_1^{(i)} \mathbf{G}_{i_2} + \mathbf{e}_2^{(i)}, \dots, \mathbf{d}_d^{(i)} = \mathbf{s}_d^{(i)} \mathbf{A} - \mathbf{s}_{d-1}^{(i)} \mathbf{G}_{i_d} + \mathbf{e}_d^{(i)}),$$

where for all $i \in [k]$ the $\mathbf{s}_1^{(i)}, \dots, \mathbf{s}_d^{(i)} \in \mathbb{Z}_q^n$ are uniformly chosen LWE secrets and the $\mathbf{e}_1^{(i)}, \dots, \mathbf{e}_d^{(i)}$ are LWE error terms.

For an index $i \in [k]$ and a commitment $\mathbf{h} \in \mathbb{Z}_q^n$ with linear local opening $\pi^{(i)} = (\mathbf{x}_i, \pi_1, \dots, \pi_{d-1})$, we define the hidden bit α_i via $\alpha_i = \lceil \mathbf{d}^{(i)} \pi^{(i)} \rceil$.

Similarly as above, to argue that $\mathbf{h}$ statistically binds to α_i , we can argue that α_i can be *equivalently computed* given only the root hash $\mathbf{h} = \mathbf{h}_d$ and the LWE secret $\mathbf{s}_d^{(i)}$ via $\alpha_i = \lceil \mathbf{s}_d^{(i)} \mathbf{h} \rceil$. Specifically, it holds

³We use the convention $[k] = \{0, \dots, k-1\}$

that

$$\begin{aligned}
\mathbf{d}^{(i)} \pi^{(i)} &\approx \mathbf{s}_1^{(i)} \underbrace{\mathbf{A} \mathbf{x}_i}_{=\mathbf{h}_1} + \sum_{j=1}^{d-1} (\mathbf{s}_{j+1}^{(i)} \mathbf{A} - \mathbf{s}_j^{(i)} \mathbf{G}_{i_{j+1}}) \pi_j \\
&= \mathbf{s}_1^{(i)} \mathbf{h}_1 + \sum_{j=1}^{d-1} (\mathbf{s}_{j+1}^{(i)} \underbrace{\mathbf{A} \pi_j}_{=\mathbf{h}_{j+1}} - \mathbf{s}_j^{(i)} \underbrace{\mathbf{G}_{i_{j+1}} \pi_j}_{=\mathbf{h}_j}) \\
&= \mathbf{s}_d^{(i)} \mathbf{h}_d \\
&= \mathbf{s}_d^{(i)} \mathbf{h}.
\end{aligned}$$

Again, going from the second to the third line is just a telescopic sum in which all but the last term $\mathbf{s}_d^{(i)} \mathbf{h}_d$ cancel out. Consequently, it holds with high probability that

$$\alpha_i = \lceil \mathbf{d}^{(i)} \pi^{(i)} \rceil = \lceil \mathbf{s}_d^{(i)} \mathbf{h} \rceil$$

as desired.

As before, we can routinely argue that each $\mathbf{d}_j^{(i)}$ is pseudorandom under the decisional LWE assumption, and in particular $\mathbf{d}_1^{(i)}$ is pseudorandom. Hence, we can likewise argue that $\mathbf{d}^{(i)} \pi^{(i)} = \mathbf{d}_1^{(i)} \mathbf{x}_i + \sum_{j=1}^{d-1} \mathbf{d}_{j+1}^{(i)} \pi_j$ is statistically close to uniform given $\mathbf{h}$ via the leftover hash lemma, as $\mathbf{x}_i \in \{0, 1\}^m$ is chosen uniformly at random.

Compressing the Public Parameters. While the above scheme has compact commitments as required by an HBG, it still has large public parameters, as we need to provide one key component $\mathbf{d}^{(i)}$ per index $i \in [k]$ in the public parameters, which means that the size of the public parameters scales linearly with the number of committed bits k .

In order to compress the public parameters, we will not include the $\mathbf{d}^{(i)}$ explicitly in the public parameters, but instead provide a small number of components that enable the verifier to construct the $\mathbf{d}^{(i)}$ “on the fly.” Specifically, let $\mathbf{B}_0, \mathbf{B}_1 \in \{0, 1\}^{n \log(q) \times n}$ be two uniformly random *binary matrices* and let $\mathbf{b}^* = (1, 0, \dots, 0) \in \{0, 1\}^{n \log q}$ be a *bookend vector*. For $b \in \{0, 1\}$ compute

$$\mathbf{C}_{1,b} = \mathbf{B}_b(\mathbf{S}_1 \mathbf{A} + \mathbf{E}_1) + \mathbf{E}'_{1,b},$$

and for $j \in \{2, \dots, d\}$ and $b \in \{0, 1\}$ compute

$$\mathbf{C}_{j,b} = \mathbf{B}_b(\mathbf{S}_j \mathbf{A} + \mathbf{E}_j) + \mathbf{E}'_{j,b} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_b,$$

where as usual $\mathbf{S}_1, \dots, \mathbf{S}_d \in \mathbb{Z}_q^{n \times n}$ are LWE secrets and the $\mathbf{E}_j$ and $\mathbf{E}'_{j,b}$ are LWE error terms. Note that for $j \geq 2$ we can equivalently write

$$\mathbf{C}_{j,b} = \mathbf{B}_b \mathbf{S}_j \mathbf{A} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_b + \mathbf{B}_b \mathbf{E}_j + \mathbf{E}'_{j,b},$$

and as the error term $\mathbf{B}_b \mathbf{E}_j + \mathbf{E}'_{j,b}$ is short we will write $\mathbf{C}_{j,b} \approx \mathbf{B}_b \mathbf{S}_j \mathbf{A} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_b$. Likewise, it holds that $\mathbf{C}_{1,b} \approx \mathbf{B}_b \mathbf{S}_1 \mathbf{A}$.

First notice that the additional elements in the public parameters, namely $(\mathbf{b}^*, \mathbf{B}_0, \mathbf{B}_1, (\mathbf{C}_{j,b})_{j \in [d], b \in \{0,1\}})$, are succinct, i.e., they can be described by $\text{poly}(\lambda) \cdot d = \text{poly}(\lambda) \cdot \log(k)$ bits.

We briefly introduce some notation. For an index $i \in [k] = [2^{d-1}]$, we consider its binary representation $i = (i_1, \dots, i_d)$ where i_1 is fixed to 0. Essentially, $i_2, \dots, i_d$ are the actual binary representation of i and i_1 is a dummy entry. For indices $j \in [d]$ we define the matrices $\mathbf{H}_j^{(i)}$ recursively. For $j = 1$, we simply define

$$\mathbf{H}_1^{(i)} = \mathbf{B}_{i_1}.$$

For $1 < j \leq d$ we define

$$\mathbf{H}_j^{(i)} = \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{B}_{i_j}.$$

First note that each $\mathbf{H}_j^{(i)}$ is short: $\mathbf{H}_1^{(i)}$ is just a binary matrix and each other $\mathbf{H}_j^{(i)}$ is just the product of two binary matrices.

We can now define *derived LWE secrets* $\mathbf{s}_j^{(i)} = \mathbf{b}^* \cdot \mathbf{H}_j^{(i)} \cdot \mathbf{S}_j$ for $i \in [k]$ and $j \in [d]$. We further set

$$\mathbf{d}_1^{(i)} = \mathbf{b}^* \cdot \mathbf{C}_{1,i_1}.$$

and for $j \in \{2, \dots, d\}$

$$\mathbf{d}_j^{(i)} = \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{C}_{j,i_j}.$$

First note that the $\mathbf{d}_j^{(i)}$ can be publicly computed from $(\mathbf{b}^*, \mathbf{B}_0, \mathbf{B}_1, (\mathbf{C}_{j,b})_{j \in [d], b \in \{0,1\}})$. Critically, observe that it holds that

$$\mathbf{d}_1^{(i)} = \mathbf{b}^* \cdot \mathbf{C}_{1,i_1} \approx \underbrace{\mathbf{b}^* \cdot \mathbf{B}_{i_1} \mathbf{S}_1}_{=\mathbf{s}_1^{(i)}} \mathbf{A} = \mathbf{s}_1^{(i)} \mathbf{A}$$

and for $j \in \{2, \dots, d\}$ that

$$\begin{aligned} \mathbf{d}_j^{(i)} &= \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{C}_{j,i_j} \\ &\approx \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot (\mathbf{B}_{i_j} \mathbf{S}_j \mathbf{A} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_{i_j}) \\ &= \mathbf{b}^* \cdot \underbrace{\left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{B}_{i_j} \mathbf{S}_j \mathbf{A}}_{=\mathbf{H}_j^{(i)}} - \underbrace{\mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \mathbf{G}^\top}_{=\mathbf{H}_{j-1}^{(i)}} \cdot \mathbf{S}_{j-1} \mathbf{G}_{i_j} \\ &= \underbrace{\mathbf{b}^* \cdot \mathbf{H}_j^{(i)} \mathbf{S}_j \mathbf{A}}_{=\mathbf{s}_j^{(i)}} - \underbrace{\mathbf{b}^* \cdot \mathbf{H}_{j-1}^{(i)} \cdot \mathbf{S}_{j-1} \mathbf{G}_{i_j}}_{=\mathbf{s}_{j-1}^{(i)}} \\ &= \mathbf{s}_j^{(i)} \mathbf{A} - \mathbf{s}_{j-1}^{(i)} \mathbf{G}_{i_j}. \end{aligned}$$

In other words, the $\mathbf{d}_j^{(i)}$ and the $\mathbf{s}_j^{(i)}$ obey the same relation as above, and the binding property for this scheme follows *exactly* as above! We note that we only use the derived LWE secrets $\mathbf{s}_j^{(i)}$ to establish the binding property of our scheme. The hiding property, which we will turn to next, is established directly.

Concerning noise growth, note that as the matrices $\left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top$ are binary, the noise terms in the $\mathbf{d}_j^{(i)}$ are only a fixed small polynomial factor larger than those in the $\mathbf{C}_{j,b}$.

Finally, to establish the hiding property we need to show that the α_i are pseudorandom given $\mathbf{h}$ as well as $\mathbf{A}$ and $(\mathbf{b}^*, \mathbf{B}_0, \mathbf{B}_1, (\mathbf{C}_{j,b})_{j \in [d], b \in \{0,1\}})$. This proof is slightly more involved than in the case of non-compact public parameters. Using LWE, we will replace the $\mathbf{C}_{j,b}$ with uniform matrices starting with $j = d$. For this purpose we will make the noise terms explicit. It holds that

$$\begin{aligned} \mathbf{C}_{d,0} &= \mathbf{B}_0 \cdot (\mathbf{S}_d \mathbf{A} + \mathbf{E}_d) + \mathbf{E}'_{d,0} - \mathbf{G}^\top \mathbf{S}_{d-1} \mathbf{G}_0 \\ \mathbf{C}_{d,1} &= \mathbf{B}_1 \cdot (\mathbf{S}_d \mathbf{A} + \mathbf{E}_d) + \mathbf{E}'_{d,1} - \mathbf{G}^\top \mathbf{S}_{d-1} \mathbf{G}_1. \end{aligned}$$

Note that both $\mathbf{C}_{d,0}$ and $\mathbf{C}_{d,1}$ share the term $\mathbf{S}_d \mathbf{A} + \mathbf{E}_d$. As $\mathbf{S}_d$ and $\mathbf{E}_d$ appear nowhere else, we can routinely replace $\mathbf{S}_d \mathbf{A} + \mathbf{E}_d$ with a uniformly random $\mathbf{U}_d$ by appealing to decisional LWE. After this replacement we can write

$$\begin{aligned} \mathbf{C}_{d,0} &= \mathbf{B}_0 \mathbf{U}_d + \mathbf{E}'_{d,0} - \mathbf{G}^\top \mathbf{S}_{d-1} \mathbf{G}_0 \\ \mathbf{C}_{d,1} &= \mathbf{B}_1 \mathbf{U}_d + \mathbf{E}'_{d,1} - \mathbf{G}^\top \mathbf{S}_{d-1} \mathbf{G}_1. \end{aligned}$$

At this point, we can consider the matrix $\mathbf{U}_d$ as a new LWE secret, as $\mathbf{U}_d$ appears nowhere else. While the matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ are binary and public, a simple reduction using binary decompositions implies that $\mathbf{B}_0\mathbf{U}_d + \mathbf{E}'_{d,0}$ and $\mathbf{B}_1\mathbf{U}_d + \mathbf{E}'_{d,1}$ are pseudorandom under standard LWE (see e.g. [BLMR13]). It follows that we can replace both $\mathbf{C}_{d,0}$ and $\mathbf{C}_{d,1}$ with uniformly random matrices. As this in particular *erases* the dependency of $\mathbf{C}_{d,0}$ and $\mathbf{C}_{d,1}$ on $\mathbf{S}_{d-1}$, we can repeat this argument and also replace $\mathbf{C}_{d-1,0}$ and $\mathbf{C}_{d-1,1}$ with uniform matrices. Once all $\mathbf{C}_{j,b}$ are replaced with uniform, we can argue for all $i \in [k]$ that $\alpha_i = \lceil \mathbf{d}^{(i)}\pi^{(i)} \rceil$ is statistically close to uniform given $\mathbf{h}$ and the public parameters. To see this, first note that since $\mathbf{C}_{1,i_1}$ is uniformly random after the above replacements and $\mathbf{b}^* = (1, 0, \dots, 0)$, it also holds that $\mathbf{d}_1^{(i)} = \mathbf{b}^*\mathbf{C}_{1,i_1}$ is distributed uniformly at random. Hence, by the leftover hash lemma, it holds that $\mathbf{d}_1^{(i)}\mathbf{x}_i$ is statistically close to uniform given $\mathbf{h}$ and the public parameters, and we can conclude that the same holds for

$$\mathbf{d}^{(i)}\pi^{(i)} = \mathbf{d}_1^{(i)}\mathbf{x}_i + \sum_{j=1}^{d-1} \mathbf{d}_j^{(i)}\pi_j^{(i)}.$$

It follows that α_i is statistically close to uniform, which concludes this proof.

Achieving Polynomial Modulus-to-Noise Ratio. In the polynomial modulus-to-noise regime, we cannot prevent the values $\alpha_i = \lceil \mathbf{d}^{(i)}\pi^{(i)} \rceil = \lceil \mathbf{s}_d^{(i)}\mathbf{h} + e^{(i)} \rceil$ and $\alpha'_i = \lceil \mathbf{s}_d^{(i)}\mathbf{h} \rceil$ from disagreeing with some inverse polynomial probability. Hence we need a mechanism that limits the number of positions i at which these two values disagree. We achieve this by introducing a random shift $\delta(i)$ into the computation of α_i , computed by a t -wise independent function $\delta : [k] \rightarrow \mathbb{Z}_q$ for a $t = \text{poly}(\lambda)$ that is independent of k , that is, we set

$$\alpha_i = \lceil \mathbf{d}^{(i)}\pi^{(i)} + \delta(i) \rceil.$$

This allows us to use strong concentration bounds to bound the number of *faulty* indices i at which a disagreement may occur to a small polynomial number (independent of k), which suffices to achieve the sparsity condition of HBGs. For details, refer to Section 4.

2.2 HBG with Batch Opening from Pairings

To reduce the size of the local proofs, we consider an HBG with *batch opening*. A batch opening is a proof that allows one to open an arbitrary subset S of positions of the hidden string with communication $\text{poly}(\lambda)$, independent of $|S|$. In contrast, standard HBG constructions do not support batch proofs: in almost all known constructions of HBG, each local proof has size $\text{poly}(\lambda)$. Therefore, opening an arbitrary subset S of positions requires communication $|S| \cdot \text{poly}(\lambda)$. The only known exception is the construction of HBG in [Gro10] based on the Naccache-Stern cryptosystem.

Our construction of an HBG with batch opening is obtained through a simple (only in hindsight!) connection to another cryptographic object called *projective* pseudorandom generators [ABI⁺23].

Projective pseudorandom generators. Projective PRGs were introduced by Applebaum et al. [ABI⁺23] to construct a computational secret sharing scheme. A projective PRG is a PRG (that is, it expands a short seed s into k pseudorandom bits) for which we can create a short projective key K_S from the seed s that allows one to evaluate the PRG on the subset S . Crucially, the bits on the subset $[k] \setminus S$ should remain pseudorandom even given the projective key K_S . The efficiency property requires that the size of the projective key K_S is independent of the size of the subset S .

To be a bit more formal, there is a setup phase that outputs a master secret key K along with a set of public parameters pp . Using K , we can derive a projective seed K_S for an arbitrary subset $S \subseteq [k]$. Given the public parameters pp and the projective seed K_S , we can evaluate the output of the PRG on the seed K at locations given by S . The pseudorandomness property says that given K_S , the output of the PRG on the seed K at locations $[k] \setminus S$ is pseudorandom.

The work of Applebaum *et al.* [ABI⁺23] provided constructions for projective PRGs from several assumptions, including DDH, DBDH, LWE, and factoring-based assumptions, each providing different levels of efficiency.

Connection between HBG and Projective PRGs. To see how a projective PRG might be useful in constructing an HBG with batch opening, consider the following idea. The prover samples a PRG seed K and sets the hidden bits of the HBG to be the expanded output of the seed K . In order to open to a subset S of the hidden bits, the prover can simply release a projective seed K_S corresponding to K . The hiding property of HBG follows easily from the pseudorandomness property of the projective PRG.

Despite this superficial similarity, projective PRGs do not by themselves imply an HBG, as we explain below.

- **Reusability:** The setup phase in the case of projective PRGs samples both the public parameters pp and the PRG seed K . This means that the public parameters are not reusable. In contrast, the CRS in HBG must be reusable, i.e., the prover should be able to sample the master seed K given a CRS (but not the random coins used to sample the CRS).
- **Commitments:** More importantly, projective PRGs do not provide any mechanism to commit to the seed and release verifiable local openings to subsets of the expanded output. Recall that the statistical binding property of an HBG requires the prover to send a succinct commitment that statistically binds the prover to all the pseudorandom bits. In the case of projective PRGs, the master seed K itself could serve as such a commitment. However, giving out the master seed in the clear would completely compromise the hiding property. Naive approaches such as committing to the master seed K using a standard commitment scheme do not work. To see this, note that the opening to a subset S would now require releasing not only the projective seed K_S but also a consistency proof that the projective seed is derived from the committed master seed. This consistency proof should be both statistically sound and zero knowledge in order to achieve binding and hiding properties. However, designing such consistency proofs without relying on NIZKs seems nontrivial.

In this work, we show that by using pairings, we can overcome both these bottlenecks and obtain a construction of HBG with batch opening.

2.2.1 Construction

The starting point of our HBG construction is the projective PRG construction from pairings given in [ABI⁺23]. We recall the technical details below.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two groups of prime order p (with generators g and $g_T = e(g, g)$) and let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be the pairing operation.

The setup algorithm starts by generating a matrix

$$\mathbf{M} = \begin{pmatrix} g^{r_1 x_1} \cdot g^{y_1} & g^{r_1 x_2} & \dots & g^{r_1 x_k} \\ g^{r_2 x_1} & g^{r_2 x_2} \cdot g^{y_2} & \dots & g^{r_2 x_k} \\ \vdots & & \ddots & \vdots \\ g^{r_k x_1} & g^{r_k x_2} & \dots & g^{r_k x_k} \cdot g^{y_k} \end{pmatrix}$$

where all elements x_i , r_i and y_i are uniformly chosen at random, together with elements g^{x_i} and g^{r_i} for each $i \in [k]$. The public parameters include $(\mathbf{M}, \{g^{r_i}, g^{x_i}\}_{i \in [k]})$ and the master seed $s \leftarrow \mathbb{Z}_p$ is uniformly chosen.

The i -th pseudorandom bit α_i is computed as follows:

$$\alpha_i = \frac{e(g^{r_i x_i + y_i}, g^s)}{e(g^{r_i}, g^{x_i s})} = \frac{g_T^{s(r_i x_i + y_i)}}{g_T^{s r_i x_i}} = g_T^{s y_i}$$

for all $i \in [k]$.

Finally, a projective key K_S for a subset $S \subseteq [k]$ is computed as $K_S = (g^s, \pi_S)$ where

$$\pi_S = \left(\prod_{j \in S} g^{x_j} \right)^s.$$

The projective evaluation of the PRG for every $i \in S$ is done by computing

$$\alpha_i = \frac{e\left(g^s, g^{r_i x_i + y_i} \prod_{j \in S \setminus \{i\}} g^{r_i x_j}\right)}{e(\pi_S, g^{r_i})} = \frac{e(g, g)^{s y_i + \sum_{j \in S} s r_i x_j}}{e(g, g)^{\sum_{j \in S} s r_i x_j}} = g_T^{s y_i}.$$

It was already noted by Applebaum et al. [ABI⁺23] that the setup phase in this construction is reusable and the master seed can be chosen independently of the public parameters. This already solves the first bottleneck. We explain below how to overcome the second bottleneck.

Proof of Statistical Binding. To build our HBG, we start by setting the CRS to be the public parameters of the above reusable projective PRGs. Specifically, the CRS consists of the matrix $\mathbf{M}$ together with the elements g^{x_i} and g^{r_i} for each $i \in [k]$.

The prover samples the master seed $s \leftarrow \mathbb{Z}_p$. The next step is to define a succinct commitment to this master seed. We define this commitment to be $\text{com} = g^s$. The batch opening proof to a subset S is the value π_S defined in the construction above. For the sake of simplicity, we describe our construction assuming the decisional Bilinear Diffie-Hellman assumption. In the main body, we replace this with the computational Bilinear Diffie-Hellman assumption. The hidden bits $\{\alpha_i\}_{i \in [k]}$ ⁴ are set as

$$\alpha_i = \text{LSB}(g_T^{s y_i})$$

for all $i \in [k]$.

This construction seems promising. First, given the commitment g^s , the hidden bits $\alpha_1, \dots, \alpha_k$ are statistically determined by the CRS. However, we are not quite done. Recall that the statistical binding property requires that, given the commitment, a malicious prover cannot open any location $i \in [k]$ to any value other than α_i . The problem is that a malicious prover can set π_S to be an arbitrary group element, and thus we cannot ensure this property. In other words, we need a way to connect the commitment to the seed and the projective key.

Our main observation is that, if the verifier is guaranteed that com and π_S *share* the same exponent s , then the verifier is guaranteed that the “projective” evaluation is identical to the real one. We show that in the pairing world, this can be achieved via the following checks: given a commitment $\text{com} = g^s$, a projective key π_S , and a value α_i , the verifier performs two checks:

1. The elements com and π_S *share* the same exponent s . This can be checked by $e(\text{com}, g^{\sum_{i \in S} x_i}) \stackrel{?}{=} e(\pi_S, g)$.
2. Check whether α_i is computed correctly, that is, $\alpha_i \stackrel{?}{=} \text{LSB}\left(\frac{e(\text{com}, g^{r_i x_i + y_i} \prod_{j \in S \setminus \{i\}} g^{r_i x_j})}{e(\pi_S, g^{r_i})}\right)$. This step uses the projective evaluation from the construction described above.

We can now argue statistical binding. Note that if the first check passes, then π_S is $g^s \sum_{i \in S} x_i$. This means

⁴Here, we implicitly assume that the LSB of a random group element is uniform.

that if the second check also passes, then

$$\begin{aligned}
\alpha_i &= \text{LSB} \left(\frac{e \left(\text{com}, g^{r_i x_i + y_i} \prod_{j \in S \setminus \{i\}} g^{r_i x_j} \right)}{e(\pi_S, g^{r_i})} \right) \\
&= \text{LSB} \left(\frac{e \left(g^s, g^{\sum_{j \in S} r_i x_j} \right) g_T^{s y_i}}{e \left(g^s, g^{\sum_{j \in S} x_j r_i} \right)} \right) \\
&= \text{LSB} (g_T^{s y_i})
\end{aligned}$$

This shows that if both checks pass, then $\alpha_i = \text{LSB} (g_T^{s y_i})$. Note that in this case, α_i is only a function of the commitment and the CRS. This is sufficient to prove statistical binding.

Proof of Hiding. Finally, we briefly describe how to achieve hiding. Suppose that we want to prove hiding for the first value α_1 . We have to show that α_1 is a pseudorandom group element, even given all other $\alpha_2, \dots, \alpha_k$ and local proofs $\pi_2 = g^{s x_2}, \dots, \pi_k = g^{s x_k}$. Note that given the local proofs, we can generate a batch proof for any subset S that does not contain the first element.

For this, we rely on the decisional bilinear Diffie-Hellman (DBDH) assumption, which states that

$$(g, g^a, g^b, g^c, e(g, g)^{abc}) \approx_c (g, g^a, g^b, g^c, e(g, g)^u)$$

where a, b, c, u are uniform values.

Let $g^{r_1}, g^{x_1}, g^s = \text{com}, e(g, g)^w$ be a DBDH challenge where either $w = r_1 x_1 s$ or w is uniformly chosen. We set

$$\alpha_1 = \text{LSB} \left(\frac{e(\text{com}, g^v)}{e(g, g)^w} \right)$$

where $v = r_1 x_1 + y_1$ is a uniformly chosen value (the uniformity of v follows since y_1 is uniformly distributed). We can generate the other parts of the CRS and the local proofs $\pi_2, \dots, \pi_k$ using the DBDH instance. Now, if we start with a valid DBDH instance, that is, if $w = r_1 x_1 s$, then $\alpha_1 = \text{LSB} (e(g, g)^{s y_1})$. On the other hand, if it is not a DBDH instance, then α_1 is uniform and we have hiding.

We give the formal construction and analysis in Section 5 where we rely on the computational Bilinear Diffie-Hellman assumption. Looking ahead, we replace LSB with the Goldreich-Levin [GL89] hardcore predicate in our construction.

Reducing the public parameter size with q -type assumptions. The public parameters in the above construction consist of $O(k^2)$ group elements, as they include the $k \times k$ matrix $\mathbf{M}$, containing all the *cross terms* $g^{r_i x_j}$ for all $i, j \in [k]$ for verification purposes. Several existing works have shown techniques to “compress” these cross terms while keeping the diagonal terms $g^{r_i x_i}$ pseudorandom (e.g., [BW10, GLWW24]). In this work, we adopt the partial-fraction techniques inspired by [JNR26]. Specifically, we correlate all the terms r_i, x_i by setting $r_i = x_i = \frac{1}{x+i}$ for a random $x \leftarrow \mathbb{Z}_p$, with the core idea that

$$g^{\frac{1}{(x+i)(x+j)}} = g^{\frac{1}{j-i} \left(\frac{1}{x+i} - \frac{1}{x+j} \right)} = \left(g^{\frac{1}{x+i}} / g^{\frac{1}{x+j}} \right)^{\frac{1}{j-i}},$$

while the diagonal terms $g^{r_i x_i} = g^{1/(x+i)^2}$ remain hard to compute. Therefore, by replacing the matrix $\mathbf{M}$ with only the diagonal terms $g^{r_i x_i + y_i} = g^{1/(x+i)^2 + y_i}$ and the linear terms $g^{r_i} = g^{x_i} = g^{1/(x+i)}$, one can fully recover the previous construction with only $O(k)$ group elements in the public parameters.

To argue security, we will rely on a q -type assumption adapted from [JNR26], which we call the q -Computational Bilinear Square-Pole eXternal Diffie-Hellman (q -XSqPole) assumption (Assumption 2). We refer the reader to Section 6 for the formal definition of this assumption and the security analysis of the construction. We also give a proof of the q -XSqPole assumption in the generic group model in Appendix A.

2.3 HBG with Succinct CRS and Batch Openings from RSA

We now explain how to achieve both succinct CRS and batch openings from the RSA assumption. Our starting point is again the projective PRG construction from the RSA assumption given by Applebaum et al. [ABI⁺23], which we recall below.

The setup algorithm samples an RSA modulus N and k primes $e_1, \dots, e_k$. The public parameters include $(N, e_1, \dots, e_k)$. The master seed is set to a randomly chosen $u \leftarrow \mathbb{Z}_N^*$. The i -th pseudorandom bit is set to $\text{hc}(u^{1/e_i})$ where hc is the Goldreich-Levin hardcore predicate. The batch opening to a set $W \subseteq [k]$ is set to $\pi_W = u^{\prod_{j \in W} 1/e_j}$. From the batch opening proof and the public parameters, it is possible to derive the i -th pseudorandom bit by computing $(\pi_W)^{\prod_{j \in W, j \neq i} e_j} = u^{1/e_i}$.

An immediate problem we face when trying to lift this construction to an HBG is that to generate the i -th pseudorandom bit and the batch opening proof, the prover needs access to the factorization of N . Indeed, computing u^{1/e_i} is done by first computing d_i such that $d_i \cdot e_i = 1 \pmod{\phi(N)}$ and then computing u^{d_i} . This is a fundamental issue for the HBG setting, as the public parameters are not reusable.

Our Construction. Our approach to overcoming this obstacle is to modify the construction so that the prover can compute the hidden bits and the batch opening proof without having access to the factorization of N . We explain this below.

The setup algorithm remains unchanged. The prover samples $x \leftarrow \mathbb{Z}_N^*$ and sets $\text{com} = x^{\prod_{j \in [k]} e_j}$. Looking ahead, this com will serve the same purpose as u in the projective PRG construction of [ABI⁺23]. To compute the i -th pseudorandom bit, the projective PRG construction computes $\text{hc}(u^{1/e_i})$. Recall that this required factorization of N . However, in our case, com is set to $x^{\prod_{j \in [k]} e_j}$. Therefore, com^{1/e_i} is essentially equal to $x^{\prod_{j \neq i} e_j}$. This can be computed from x and the public parameters. The batch opening proof to a subset $W \subseteq [k]$ in the projective PRG construction is given by $u^{\prod_{j \in W} 1/e_j}$. When translated to our construction, this value is equal to $x^{\prod_{j \notin W} e_j}$. This can also be computed from x and the public parameters.

The verification performs two checks. First, it checks whether $(\pi_W)^{\prod_{j \in W} e_j} \stackrel{?}{=} \text{com}$ and whether the i -th opened bit is equal to $\text{hc}((\pi_W)^{\prod_{j \in W, j \neq i} e_j})$. These two checks can be easily shown to pass when π_W and the i -th hidden bit are computed honestly.

To prove statistical binding, observe that if these two checks pass, then the i -th opened bit is essentially equal to $\text{hc}(\text{com}^{1/e_i})$. Therefore, this value is completely determined by the public parameters and com . The hiding property is proven by providing a reduction to the RSA assumption, and this part is similar to [ABI⁺23].

Reducing the public parameter size. We rely on an observation from earlier works [HW09, ABI⁺23] to reduce the size of the public parameters. Specifically, Hohenberger and Waters [HW09] showed how to plant a prime using a short PRF seed. Subsequently, Applebaum et al. [ABI⁺23] observed that the primes $(e_1, \dots, e_k)$ can be generated using an $O(\lambda)$ -wise independent hash function, and this is sufficient for the hiding argument to go through. This observation helps in reducing the randomness complexity of sampling $(e_1, \dots, e_k)$. Thus, it is sufficient to include the description of this hash function in the public parameters rather than giving the entire set of k primes.

We give the formal description of our construction and its analysis in Section 7.

3 Preliminaries

Let λ denote the cryptographic security parameter. We assume that all cryptographic algorithms implicitly take 1^λ as input. We will use $\text{negl}(\cdot)$ to denote an unspecified negligible function and $\text{poly}(\cdot)$ to denote an unspecified polynomial function. Let X and Y be two distributions. Then, (i) $X \approx_c Y$ means that the distributions are computationally indistinguishable, (ii) $X \approx_\epsilon Y$ means that their statistical distance is at most ϵ , and (iii) $X \approx_s Y$ means that their statistical distance is at most $\text{negl}(\lambda)$. For a finite set S , we write $x \leftarrow S$ for the process of sampling x uniformly from the set S .

Definition 1 (Randomness extractor). A (k, ϵ) -strong randomness extractor $\text{Ext} : \mathcal{S} \times \mathcal{X} \rightarrow \mathcal{Y}$ is a deterministic algorithm with domain $\mathcal{X}$, seed space $\mathcal{S}$, and range $\mathcal{Y}$ with the following property: For every distribution X with support $\mathcal{X}$ and min-entropy at least k ,

$$(s, \text{Ext}(s, x)) \approx_\epsilon (s, y)$$

where $x \leftarrow_{\$} X$ and $y \leftarrow_{\$} \mathcal{Y}$.

Definition 2 (Average Min-Entropy). Given two jointly distributed random variables (X, Y) , the average min-entropy of X conditioned on Y is defined as

$$\tilde{H}_\infty(X|Y) = -\log \left(\mathbb{E}_{y \leftarrow_{\$} Y} \left[\max_x \Pr[X = x \mid Y = y] \right] \right).$$

Lemma 1 (Chain rule). For random variables X, Y, Z where Y is supported over a set of size T , we have

$$\tilde{H}_\infty(X|(Y, Z)) \geq \tilde{H}_\infty((X, Y)|Z) - \log(T) \geq \tilde{H}_\infty(X|Z) - \log(T).$$

Lemma 2 (Leftover Hash Lemma). Let X be a random variable supported on a finite set $\mathcal{X}$ with $\tilde{H}_\infty(X) \geq k$. Suppose $h : \mathcal{S} \times \mathcal{X} \rightarrow \{0, 1\}^\ell$ is a universal hash function, where $\ell \leq k - 2\log(1/\epsilon)$. Then, for a uniformly chosen $S \leftarrow \mathcal{S}$,

$$(h(S, X), S) \approx_\epsilon (z, S),$$

where $z \leftarrow \{0, 1\}^\ell$.

The following result, due to Regev [Reg05], shows that the LHL can be used when the input domain is binary.

Lemma 3 ([Reg05]). Fix a positive integer q , and let $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n \times m}$ be sampled uniformly at random. Then the map $h : \mathbb{Z}_q^{n \times m} \times \{0, 1\}^m \rightarrow \mathbb{Z}_q^n$ defined by $h(\mathbf{A}, \mathbf{x}) = \mathbf{A}\mathbf{x}$ is a universal hash function.

3.1 Non-Interactive Zero-Knowledge (NIZKs)

In this section, we recall the notion of non-interactive zero-knowledge.

Definition 3. A non-interactive zero-knowledge (NIZK) proof for an NP language $\mathcal{L}$, with corresponding instance-witness relation $\mathcal{R} : \mathcal{X} \times \mathcal{W} \rightarrow \{0, 1\}$, is a tuple of algorithms $(\text{Setup}, \text{P}, \text{V})$ such that:

- $\text{Setup}(1^\lambda) \rightarrow \text{crs}$: This is a PPT algorithm that takes as input the unary encoding of the security parameter 1^λ and outputs a common reference string crs .
- $\text{P}(\text{crs}, x, w) \rightarrow \pi$: This is a PPT algorithm that takes as input a common reference string crs , a statement $x \in \mathcal{X}$, and a witness $w \in \mathcal{W}$. It outputs a proof π .
- $\text{V}(\text{crs}, x, \pi) \rightarrow b$: This is a deterministic algorithm that takes as input a common reference string crs , a statement $x \in \mathcal{X}$, and a proof π . It outputs a bit $b \in \{0, 1\}$.

A NIZK proof system satisfies the following properties:

- **Completeness.** For every $x \in \mathcal{X}$ and $w \in \mathcal{W}$ such that $\mathcal{R}(x, w) = 1$, we have:

$$\Pr \left[\text{V}(\text{crs}, x, \pi) = 1 \quad : \quad \begin{array}{l} \text{crs} \leftarrow \text{Setup}(1^\lambda), \\ \pi \leftarrow \text{P}(\text{crs}, x, w) \end{array} \right] = 1$$

- **Soundness.** For every unbounded prover P^* , we have:

$$\Pr \left[\text{V}(\text{crs}, x, \pi) = 1 \wedge x \notin \mathcal{L} \quad : \quad \begin{array}{l} \text{crs} \leftarrow \text{Setup}(1^\lambda), \\ (x, \pi) \leftarrow \text{P}^*(\text{crs}) \end{array} \right] \leq \text{negl}(\lambda)$$

- **Zero-knowledge.** There exists a simulator $\text{Sim} = (\text{Sim}_1, \text{Sim}_2)$ such that for any non-uniform PPT adversary $\mathcal{A}$, we have:

$$\left| \Pr[1 \leftarrow \mathcal{A}^{\text{Real}(\cdot, \cdot)}(\text{crs}) : \text{crs} \leftarrow \text{Setup}(1^\lambda)] - \Pr[1 \leftarrow \mathcal{A}^{\text{Ideal}(\cdot, \cdot)}(\text{crs}) : (\text{crs}, \text{td}) \leftarrow \text{Sim}_1(1^\lambda)] \right| \leq \text{negl}(\lambda)$$

where Real takes as input (x, w) and if $R(x, w) = 1$, outputs $\pi \leftarrow \text{P}(\text{crs}, x, w)$. Ideal takes as input (x, w) and if $R(x, w) = 1$, outputs $\pi \leftarrow \text{Sim}_2(\text{td}, x)$.

3.2 NIZKs in the Hidden-Bits Model

We recall the definition of NIZKs in the hidden-bits model.

Definition 4. Let $\mathcal{L}$ be an NP language and n be an integer. A non-interactive zero-knowledge proof in the hidden-bits model for $\mathcal{L}$ is given by a pair of PPT algorithms (P, V) and a polynomial $k(\lambda, n)$, where:

- $\text{P}(1^\lambda, r, x, w)$ takes as input a string $r \in \{0, 1\}^{k(\lambda, n)}$, a statement x of size $|x| = n$, and a witness w . It outputs a set of indices $I \subseteq [k]$ and a proof π .
- $\text{V}(1^\lambda, I, r_I, x, \pi)$ takes as input a subset $I \subseteq [k]$, a string r_I , a statement x , and a proof π . It outputs 0 or 1.

A NIZK in the Hidden-Bits Model should fulfill the following properties.

- **Completeness:** For all $x \in \mathcal{L}$ of size $|x| = n$ with witness w we have:

$$\Pr \left[\text{V}(1^\lambda, I, r_I, x, \pi) = 1 : \begin{array}{l} r \leftarrow \{0, 1\}^{k(\lambda, n)} \\ (I, \pi) \leftarrow \text{P}(1^\lambda, r, x, w) \end{array} \right] = 1.$$

- **Soundness:** For all polynomial $n = n(\lambda)$, and all unbounded cheating provers $\tilde{\text{P}}$, we have:

$$\Pr \left[\text{V}(1^\lambda, I, r_I, x, \pi) = 1 \wedge x \notin \mathcal{L} \wedge |x| = n : \begin{array}{l} r \leftarrow \{0, 1\}^{k(\lambda, n)} \\ (x, \pi, I) \leftarrow \tilde{\text{P}}(1^\lambda, r) \end{array} \right] \leq \text{negl}(\lambda)$$

- **Zero-Knowledge:** There exists an efficient simulator Sim such that for any adversary $\mathcal{A}$ the following two distributions are statistically indistinguishable:

$$(I, r_I, \pi) \approx_s (I', r_{I'}, \pi')$$

where $r \leftarrow \{0, 1\}^{k(\lambda, n)}$, $(I, \pi) \leftarrow \text{P}(1^\lambda, r, x, w)$, $(I', r_{I'}, \pi') \leftarrow \text{Sim}(1^\lambda, x)$.

The following theorem was proved in [FLS90].

Theorem 5 ([FLS90]). For some polynomial $k'(\lambda, n)$ and for any polynomial $q(\lambda, n)$, there exists a NIZK in the hidden-bits model with $k(\lambda, n) = k'(\lambda, n) \cdot q(\lambda, n)$ and soundness error $2^{-q(\lambda, n)} \cdot \text{negl}(\lambda)$.

The following theorem is taken from [Gro10]. It states that there is a NIZK scheme in the hidden-bits model whose proof size is quasilinear.

Theorem 6 ([Gro10]). Let $\mathcal{L}$ be an NP language with verification circuit C . There exists a NIZK in the hidden-bits model for $\mathcal{L}$ where the sizes of the hidden string and proof are at most $|C| \cdot \text{polylog}(|C|) + \text{poly}(\lambda)$ with soundness error $2^{-\lambda}$.

3.3 Hidden-Bits Generator

We now define a Hidden-Bits Generator (HBG) [QRW19] following [Wat24]. For a string $\mathbf{r} \in \{0, 1\}^k$ and a set of indices $I \subseteq [k]$, we denote by $\mathbf{r}_I$ the substring corresponding to the bits of $\mathbf{r}$ indexed by I .

Definition 5 (Hidden-Bits Generator). *A Hidden-Bits Generator (HBG) is given by a set of PPT algorithms (Setup, GenBits, Verify):*

- **Setup**($1^\lambda, 1^k$) takes as input a security parameter and 1^k . It outputs public parameters pp .
- **GenBits**(pp) takes as input the public parameters and outputs a triple $(\text{com}, \mathbf{r}, \{\pi_i\}_{i \in [k]})$ where $\mathbf{r} \in \{0, 1\}^k$.
- **Verify**($\text{pp}, \text{com}, i, \mathbf{r}_i, \pi_i$) takes as input the public parameters, a commitment com , an index $i \in [k]$, a bit $\mathbf{r}_i$ and a local proof π_i . It outputs a bit $b \in \{0, 1\}$.

We require a Hidden-Bits Generator to satisfy the following properties.

- **Completeness.** We require that for every polynomial $k = k(\lambda)$ and for all $i \in [k]$, we have:

$$\Pr \left[\text{Verify}(\text{pp}, \text{com}, i, \mathbf{r}_i, \pi) = 1 \quad : \quad \begin{array}{l} \text{pp} \leftarrow \text{Setup}(1^\lambda, 1^k) \\ (\text{com}, \mathbf{r}, \{\pi_i\}_{i \in [k]}) \leftarrow \text{GenBits}(\text{pp}) \end{array} \right] = 1.$$

- **Binding.** For every pp in the support of the algorithm $\text{Setup}(1^\lambda, k)$, there exists a set $\mathcal{V}^{\text{pp}}$ with the following properties.

- **Output sparsity.** There exists a fixed polynomial $p(\cdot)$ such that for every polynomial $k = k(\lambda)$, and every pp in the support of $\text{Setup}(1^\lambda, 1^k)$,

$$|\mathcal{V}^{\text{pp}}| \leq 2^{p(\lambda)}$$

- **Statistical binding.** For every (possibly inefficient) stateful adversary $\mathcal{A}$

$$\Pr \left[\begin{array}{l} \mathbf{r}_I \notin \mathcal{V}_I^{\text{pp}} \\ \forall i \in I, \text{Verify}(\text{pp}, \text{com}, i, \mathbf{r}_i, \pi_i) = 1 \end{array} \quad : \quad \begin{array}{l} k \leftarrow \mathcal{A}(1^\lambda) \\ \text{pp} \leftarrow \text{Setup}(1^\lambda, 1^k) \\ (\text{com}, I, \mathbf{r}_I, \{\pi_i\}_{i \in I}) \leftarrow \mathcal{A}(\text{pp}) \end{array} \right] = \text{negl}(\lambda).$$

By $\mathcal{V}_I^{\text{pp}}$ we denote the set $\{\mathbf{r}_I : \mathbf{r} \in \mathcal{V}^{\text{pp}}\}$.

- **Hiding.** There is an alternate setup algorithm Setup^* that satisfies the following properties:

- **CRS Indistinguishability.** $\{\text{pp} : \text{pp} \leftarrow \text{Setup}(1^\lambda, 1^k)\} \approx_c \{\text{pp} : \text{pp} \leftarrow \text{Setup}^*(1^\lambda, 1^k)\}$.
- **Pseudorandomness.** We have that for all efficient and stateful adversaries $\mathcal{A}$,

$$\left| \Pr[\mathcal{A}(\text{pp}, \text{com}, \{\mathbf{r}_i, \pi_i\}_{i \neq i^*}, \mathbf{r}_{i^*})] - \Pr[\mathcal{A}(\text{pp}, \text{com}, \{\mathbf{r}_i, \pi_i\}_{i \neq i^*}, u)] \right| = \text{negl}(\lambda)$$

where $(k, i^*) \leftarrow \mathcal{A}(1^n)$, $\text{pp} \leftarrow \text{Setup}^*(1^n, 1^k)$, $(\text{com}, \mathbf{r}, \{\pi_i\}_{i \in [k]}) \leftarrow \text{GenBits}(\text{pp})$ and $u \leftarrow \{0, 1\}$.

[QRW19] showed how to transform a NIZK in the hidden-bits model to the common reference string model using an HBG. We get the following theorem when using the construction of NIZKs in the hidden-bits model given in Theorem 6.

Theorem 7 ([QRW19]). *Let $\mathcal{L}$ be an NP language with verification circuit C . Assume the existence of a hidden-bits generator. There exists a NIZK proof system in the CRS model for $\mathcal{L}$ where the size of crs is $|\text{pp}| + |C| \text{poly log } |C| + \text{poly}(\lambda)$ and the proof size is given by $\max_I (|I| + \sum_{i \in I} |\pi_i|)$ where I is a subset of size at most $|C| \text{poly log } |C| + \text{poly}(\lambda)$.*

Batch Openings. We say that an HBG has batch openings if for any subset $W \subseteq [k]$, there exists a short opening proof π_W that convinces the verifier about the correctness of opened bits $\{\mathbf{r}_i\}_{i \in W}$. In this case, we modify `Verify` to take $(\mathbf{pp}, \mathbf{com}, W, \{\mathbf{r}_i\}_{i \in W}, \pi_W)$ as inputs. The completeness and binding properties are modified accordingly. The hiding game requires the adversary to declare a subset $W \subset [k]$ along with (k, i^*) such that $i^* \notin W$ and the adversary is given the openings $\{r_i\}_{i \in W}, \pi_W$ (rather than all other openings).

4 Hidden-Bits Generator from LWE

For this construction, we rely on the LWE assumption which was first introduced by Regev [Reg05].

Definition 6 (Learning with Errors). *Let $n, q \in \mathbb{Z}$. For all $m = \text{poly}(n)$, we say that the LWE assumption $\text{LWE}_{n,q,\sigma}$ holds if*

$$(\mathbf{A}, \mathbf{s}^T \mathbf{A} + \mathbf{e}^T) \approx_c (\mathbf{A}, \mathbf{u}^T)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow D_{\mathbb{Z}, \sigma}^m, \mathbf{u} \leftarrow \mathbb{Z}_q^m$ and where $D_{\mathbb{Z}, \sigma}^m$ is the discrete Gaussian distribution with parameter σ .

When we consider $\sigma = \zeta q \geq 2\sqrt{n}$, the LWE problem is at least as hard as solving the approximate shortest independent vector problem to within a factor of $\tilde{O}(n/\zeta)$ [Reg05].

The following lemma states that the LWE assumption still holds even if the LWE matrix is a uniformly chosen binary matrix.

Lemma 4 ([BLMR13]). *Let $n, q \in \mathbb{Z}$. For all $m = \text{poly}(n)$, assuming that the $\text{LWE}_{n,q,\sigma}$ assumption holds, we have that*

$$(\mathbf{B}, \mathbf{s}^T \mathbf{B} + \mathbf{e}^T) \approx_c (\mathbf{B}, \mathbf{u}^T)$$

where $\mathbf{B} \leftarrow \{0, 1\}^{n \log q \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^{n \log q}, \mathbf{e} \leftarrow D_{\mathbb{Z}, \sigma}^m, \mathbf{u} \leftarrow \mathbb{Z}_q^m$.

For a modulus q , let $\mathbf{g}^T = (1, 2, 2^2, \dots, 2^{\lceil \log(q) \rceil}) \in \mathbb{Z}^{1 \times \lceil \log(q) \rceil}$ be the powers-of-two vector (as a row vector). For a dimension n , let $\mathbf{I}$ be the $n \times n$ identity matrix. The gadget matrix $\mathbf{G}$ [MP12] is given by

$$\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^T = \begin{bmatrix} \mathbf{g}^T & & & \\ & \mathbf{g}^T & & \\ & & \ddots & \\ & & & \mathbf{g}^T \end{bmatrix} \in \mathbb{Z}^{n \times n \lceil \log(q) \rceil}.$$

Define $\mathbf{g}^{-1} : \mathbb{Z}_q \rightarrow \{0, 1\}^{\lceil \log(q) \rceil}$ as the function that computes the binary decomposition of $\mathbb{Z}_q$ elements (as a column vector). Note that it holds $\mathbf{g}^T \cdot \mathbf{g}^{-1}(x) = x$. This induces a function $\mathbf{G}^{-1} : \mathbb{Z}_q^n \rightarrow \{0, 1\}^{n \lceil \log(q) \rceil}$ given by

$$\mathbf{G}^{-1}(\mathbf{x}) = \begin{bmatrix} \mathbf{g}^{-1}(x_1) \\ \vdots \\ \mathbf{g}^{-1}(x_n) \end{bmatrix}.$$

Note that it holds that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{x}) = \mathbf{x}$. For a matrix $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_m) \in \mathbb{Z}_q^{n \times m}$ we define $\mathbf{G}^{-1}(\mathbf{A})$ via

$$\mathbf{G}^{-1}(\mathbf{A}) = (\mathbf{G}^{-1}(\mathbf{a}_1), \dots, \mathbf{G}^{-1}(\mathbf{a}_m)).$$

Note finally that it holds that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A}$.

We will introduce some additional shorthand notation. Let $m = n \log q$. We will define two additional matrices $\mathbf{G}_0, \mathbf{G}_1 \in \mathbb{Z}_q^{n \times 2m}$ via

$$\mathbf{G}_0 = (\mathbf{G}, \mathbf{0}) \quad \mathbf{G}_1 = (\mathbf{0}, \mathbf{G})$$

where $\mathbf{0} \in \mathbb{Z}_q^{n \times m}$ is the all-zero matrix.

Rounding. We additionally define the usual rounding function $\lfloor \cdot \rfloor_2 : \mathbb{Z}_q \rightarrow \{0, 1\}$ where $\lfloor x \rfloor_2 = \lfloor (2/q)x \rfloor \bmod 2$. We will make use of the following simple *noise-resilience property* of the rounding function. Let $B > 0$ and assume that for $x \in \mathbb{Z}_q$ it holds that $x \notin [q/4 - B, q/4 + B] \cup [3q/4 - B, 3q/4 + B]$. Then it holds for every error $e \in \mathbb{Z}$ with $|e| < B$ that $\lfloor x + e \rfloor_2 = \lfloor x \rfloor_2$.

Discrete Gaussians. Let $\rho_\sigma(\mathbf{x})$ be the probability distribution of the Gaussian distribution over $\mathbb{R}^n$ with parameter σ and centered at 0. As usual, we define the discrete Gaussian distribution $D_{S, \sigma}$ over S and with parameter σ by the probability distribution $\rho_\sigma(\mathbf{x})/\rho_\sigma(S)$ for all $\mathbf{x} \in S$ (where $\rho_\sigma(S) = \sum_{\mathbf{x} \in S} \rho_\sigma(\mathbf{x})$).

We will use the following lemma due to [MP12] which provides a tail bound for the matrix norm of a matrix chosen component-wise from a discrete Gaussian.

Lemma 5 ([MP12], Lemma 2.8, 2.9). *Let $\sigma > \omega(\log(\lambda))$ and let $\mathbf{E} \leftarrow D_{\mathbb{Z}^{n \times m}, \sigma}$. There exists a universal constant $C > 0$ such that $\|\mathbf{E}\| \leq C \cdot \sigma \sqrt{\max(m, n)}$, except with probability $2^{-\max(m, n)}$.*

We will need a similar bound for random binary matrices.

Lemma 6 ([Ver] Thm. 4.3.5). *Assume that $m > n$ and let $\mathbf{B} \leftarrow \{0, 1\}^{m \times n}$ be chosen uniformly at random. There exists a universal constant C , such that $\|\mathbf{B}\| \leq C \cdot \sqrt{m}$, except with probability 2^{-m} .*

Limited-wise Independence. We say that random variables $Y_1, \dots, Y_k$ are t -wise independent if it holds for any subset $S = \{i_1, \dots, i_t\} \subseteq [k]$ that $(Y_{i_1}, \dots, Y_{i_t})$ are independent. It holds that t -wise independent functions $[k] \rightarrow \{0, 1\}^m$ (for $k < 2^m$) can be constructed by taking random degree t polynomials over the field $\mathbb{F}_{2^m}$ together with a suitable injective embedding $[k] \hookrightarrow \mathbb{F}_{2^m}$. Such a function can be described by $t \cdot m$ uniform bits. We will use the following simple concentration bound.

Lemma 7. *Let $Y_1, \dots, Y_k \in \{0, 1\}$ be t -wise independent and let $Y = \sum_{i=1}^k Y_i$. Then it holds that*

$$\Pr[Y \geq t] \leq (E[Y])^t.$$

Proof. First note that if $Y \geq t$, then there must exist t distinct indices $i_1, \dots, i_t$ such that $Y_{i_1} = 1, \dots, Y_{i_t} = 1$. Hence it holds by a union bound that

$$\begin{aligned} \Pr[Y \geq t] &\leq \Pr[\exists \text{ pairwise distinct } i_1, \dots, i_t \text{ s.t. } Y_{i_1} = 1, \dots, Y_{i_t} = 1] \\ &\leq \sum_{i_1, \dots, i_t \text{ pairwise distinct}} \Pr[Y_{i_1} = 1, \dots, Y_{i_t} = 1] \\ &= \sum_{i_1, \dots, i_t \text{ pairwise distinct}} \prod_{j=1}^t \Pr[Y_{i_j} = 1] \\ &= \sum_{i_1, \dots, i_t \text{ pairwise distinct}} \prod_{j=1}^t E[Y_{i_j}] \\ &\leq \sum_{i=1}^k E[Y_i]^t \\ &= (E[Y])^t. \end{aligned}$$

Here the second inequality is the union bound, the third line follows by t -independence of the Y_i , and the third inequality follows from the fact that expanding $(\sum_{i=1}^k E[Y_i])^t$ will lead to a sum of monomials in which each monomial is ≥ 0 (as each $E[Y_i] \geq 0$), and all monomials of the form $\prod_{j=1}^t E[Y_{i_j}]$ appear. $\square$

Hash functions. To set up notation, we also briefly describe how to construct a hash function by instantiating a Merkle tree using the Ajtai hash function [Ajt96].

- **Base hash function.** Define $\text{Hash}_{\mathbf{A}}$ as the algorithm that takes an input $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{Z}_q^{2n}$ (where $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{Z}_q^n$), computes $\pi = (\mathbf{G}^{-1}(\mathbf{x}_0) \mathbf{G}^{-1}(\mathbf{x}_1))$, and outputs $\mathbf{h} = \mathbf{A} \cdot \pi$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times 2m}$ is a compressing matrix.
- **Merkle hash.** Let $(\text{MerkleHash}_{\mathbf{A}}, \text{MerkleVer}_{\mathbf{A}})$ be a Merkle hash tree using a collision resistant hash function $\text{Hash}_{\mathbf{A}}$. MerkleHash hashes the input $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_k)$ into a digest com using a Merkle tree hash (using $\text{Hash}_{\mathbf{A}}$ as base hash function) and computes local proofs $\pi^{(1)}, \dots, \pi^{(k)}$ for each of the blocks $\mathbf{x}_1, \dots, \mathbf{x}_k$.

Recall that a local proof $\pi^{(i)}$ for the block $\mathbf{x}_i$ is composed of $(\mathbf{x}_i, \pi_{i,1}, \dots, \pi_{i,d-1})$, where each $\pi_{i,j}$ corresponds to the openings along the leaf-to-root path of the Merkle tree.

MerkleVer takes as input the hash value com , an index i , and a local proof π_i , and it checks if π_i is a valid Merkle tree root-to-leaf path for the block $\mathbf{x}_i$ with respect to com .

Remark 2. In our construction, we will use a slight variation of the Merkle hash, denoted by $(\text{MerkleHash}_{\mathbf{A}_0, \mathbf{A}}, \text{MerkleVer}_{\mathbf{A}_0, \mathbf{A}})$. This variation uses the hash function $\text{Hash}_{\mathbf{A}_0}$ at the leaf level and then uses $\text{Hash}_{\mathbf{A}}$.

4.1 Construction

We now present the HBG construction based on the LWE assumption. Let $n = \text{poly}(\lambda)$ be the dimension of the LWE problem and q be a modulus which is a power of 2. We set $m = n \log q$. Let $(\text{MerkleHash}_{\mathbf{A}}, \text{MerkleVer}_{\mathbf{A}})$ be the algorithms described above. Let $k = 2^{d-1} = \text{poly}(\lambda)$ (i.e. $d = O(\log(\lambda))$) be a parameter which defines the number of hidden bits.

We will first define some helper functions and constants. Denote by $\mathbf{b}^* = (1, 0, \dots, 0) \in \{0, 1\}^m$ the first unit vector. Fix two matrices $\mathbf{B}_0, \mathbf{B}_1 \in \{0, 1\}^{m \times n}$. For an index $i \in [k] = [2^{d-1}]$, we consider its binary representation $i = (i_1, \dots, i_d)$ where $i_1 = 0$ and $i_2, \dots, i_d$ are its actual binary representation. For indices $j \in [d]$ we will define the matrices $\mathbf{H}_j^{(i)}$ recursively. For $j = 1$, we define

$$\mathbf{H}_1^{(i)} = \mathbf{B}_{i_1}.$$

For $1 < j \leq d$ we define

$$\mathbf{H}_j^{(i)} = \left(\mathbf{G}^{-1} \left(\left(\mathbf{H}_{j-1}^{(i)} \right)^\top \right) \right)^\top \cdot \mathbf{B}_{i_j}.$$

Note that this recursive definition allows us to efficiently compute the matrices $\mathbf{H}_j^{(i)}$. Furthermore, since both $\left(\mathbf{G}^{-1} \left(\left(\mathbf{H}_{j-1}^{(i)} \right)^\top \right) \right)^\top$ and $\mathbf{B}_{i_j}$ are binary matrices, we can bound the norm of the vector $\mathbf{b}^* \mathbf{H}_j^{(i)}$ by $m \cdot \sqrt{n}$ via the Cauchy-Schwarz inequality.

Setup($1^\lambda, 1^k$):

- Sample matrices $\mathbf{B}_0, \mathbf{B}_1 \leftarrow \{0, 1\}^{m \times n}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times 2m}$ and $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times 3m}$. Sample a m -wise independent function $\delta : [k] \rightarrow \mathbb{Z}_q$.
- For $j \in [d]$, sample $\mathbf{S}_j \leftarrow \mathbb{Z}_q^{n \times n}$, $\mathbf{E}_1 \leftarrow D_{\mathbb{Z}^{n \times 3m}, \sigma}$ and $\mathbf{E}_j \leftarrow D_{\mathbb{Z}^{n \times 2m}, \sigma}$. Additionally, for $b \in \{0, 1\}$ sample $\mathbf{E}'_{1,b} \leftarrow D_{\mathbb{Z}^{m \times 3m}, \sigma}$ and $\mathbf{E}'_{j,b} \leftarrow D_{\mathbb{Z}^{m \times 2m}, \sigma}$. Compute

$$\mathbf{C}_{1,b} = \mathbf{B}_b \cdot (\mathbf{S}_1 \cdot \mathbf{A}_0 + \mathbf{E}_1) + \mathbf{E}'_{1,b}$$

and for $j = 2, \dots, d$ compute

$$\mathbf{C}_{j,b} = \mathbf{B}_b \cdot (\mathbf{S}_j \cdot \mathbf{A} + \mathbf{E}_j) + \mathbf{E}'_{j,b} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_b$$

- Output $\text{pp} = (\mathbf{B}_0, \mathbf{B}_1, \mathbf{A}_0, \mathbf{A}, \{\mathbf{C}_{i,b}\}_{i \in [d], b \in \{0,1\}}, \delta)$.

GenBits(pp):

- Parse pp as $(\mathbf{B}_0, \mathbf{B}_1, \mathbf{A}_0, \mathbf{A}, \{\mathbf{C}_{i,b}\}_{i \in [d], b \in \{0,1\}}, \delta)$.
- Sample $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_k)$ where each $\mathbf{x}_i \leftarrow \{0,1\}^{3m}$.
- Compute $(\text{com}, \pi^{(1)}, \dots, \pi^{(k)}) \leftarrow \text{MerkleHash}_{\mathbf{A}_0, \mathbf{A}}(\mathbf{x})$.
- For all $i \in [k]$, let $\mathbf{i} = (i_1, \dots, i_d)$ be the corresponding binary decomposition, i.e., the leaf-to-root path, and proceed as follows.
 - Compute $\mathbf{d}_1^{(i)} = \mathbf{b}^* \cdot \mathbf{C}_{1,i_1}$ and for $j = 2, \dots, d$ compute $\mathbf{d}_j^{(i)} = \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{C}_{j,i_j}$.
 - Set $\mathbf{d}^{(i)} = (\mathbf{d}_1^{(i)}, \dots, \mathbf{d}_d^{(i)})$.
 - Parse $\pi^{(i)}$ as a column vector $(\mathbf{x}_i, \pi_{i,1}, \dots, \pi_{i,d-1})$.
 - Compute $\alpha_i = \lfloor \mathbf{d}^{(i)} \pi^{(i)} + \delta(i) \rfloor_2$
- Output $(\text{com}, \pi^{(1)}, \dots, \pi^{(k)}, \alpha_1, \dots, \alpha_k)$.

Verify(pp, com, $i, \beta, \pi^{(i)}$):

- Parse pp as $(\mathbf{B}_0, \mathbf{B}_1, \mathbf{A}_0, \mathbf{A}, \{\mathbf{C}_{i,b}\}_{i \in [d], b \in \{0,1\}}, \delta)$.
- Compute $\mathbf{d}_1^{(i)} = \mathbf{b}^* \cdot \mathbf{C}_{1,i_1}$ and for $j = 2, \dots, d$ compute $\mathbf{d}_j^{(i)} = \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{C}_{j,i_j}$.
- Set $\mathbf{d}^{(i)} = (\mathbf{d}_1^{(i)}, \dots, \mathbf{d}_d^{(i)})$.
- Compute $v_i = \mathbf{d}^{(i)} \pi^{(i)} + \delta(i)$. Accept if i) $\text{MerkleVer}(\text{com}, i, \pi^{(i)}) = 1$ and ii) $\beta = \lceil v_i \rceil$. Otherwise, reject.

We now show that the scheme presented above achieves correctness, statistical binding and hiding.

Lemma 8. *The scheme presented above is complete.*

Proof. Let $\text{com}, i, \pi^{(i)}, \alpha_i$ be an honestly computed tuple of local proof $\pi^{(i)}$ for the bit α_i with respect to commitment com . By the correctness of the Merkle tree, we have that $1 \leftarrow \text{MerkleVer}(\text{com}, i, \pi^{(i)})$. It furthermore holds immediately that $\beta = \lceil v_i \rceil = \alpha_i$, as these values are computed in the same way by GenBits and Verify. Hence the verifier accepts. $\square$

Theorem 8. *If $q > \omega(k^2 \cdot \sigma \cdot d \cdot m^2)$, then the scheme presented above is statistically binding.*

Proof. Recall that for $b \in \{0,1\}$,

$$\mathbf{C}_{1,b} = \mathbf{B}_b \cdot (\mathbf{S}_1 \cdot \mathbf{A}_0 + \mathbf{E}_1) + \mathbf{E}'_{1,b}$$

and for $j \in [d] \setminus \{1\}$ and $b \in \{0,1\}$,

$$\mathbf{C}_{j,b} = \mathbf{B}_b \cdot (\mathbf{S}_j \cdot \mathbf{A} + \mathbf{E}_j) + \mathbf{E}'_{j,b} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_b.$$

By Lemma 5 and a union bound, it holds for each $j \in [d]$ and each $b \in \{0,1\}$ that $\|\mathbf{E}_j\| \leq O(\sigma\sqrt{m})$ and $\|\mathbf{E}'_{j,b}\| \leq O(\sigma\sqrt{m})$, except with negligible probability over all these random choices. Moreover, by Lemma 6 and a union bound, it holds that $\|\mathbf{B}_0\| \leq O(\sqrt{m})$ and $\|\mathbf{B}_1\| \leq O(\sqrt{m})$, except with negligible probability.

Hence, assume in the following that $\|\mathbf{E}_j\| \leq O(\sigma\sqrt{m})$, $\|\mathbf{E}'_{j,b}\| \leq O(\sigma\sqrt{m})$, $\|\mathbf{B}_0\| \leq O(\sqrt{m})$ and $\|\mathbf{B}_1\| \leq O(\sqrt{m})$.

For each $i \in [k]$ and $j \in [d]$ denote $\mathbf{s}_j^{(i)} = \mathbf{b}^* \cdot \mathbf{H}_j^{(i)} \cdot \mathbf{S}_j$. We will first show that each $\mathbf{d}_j^{(i)}$ is of the form

$$\mathbf{d}_1^{(i)} = \mathbf{s}_1^{(i)} \mathbf{A}_0 + \mathbf{e}_1^{(i)}$$

or for $j = 2, \dots, d$ of the form

$$\mathbf{d}_j^{(i)} = \mathbf{s}_j^{(i)} \mathbf{A} - \mathbf{s}_{j-1}^{(i)} \mathbf{G}_{i_j} + \mathbf{e}_j^{(i)}$$

for error terms $\mathbf{e}_j^{(i)}$. This will also allow us to bound the size of the error terms $\mathbf{e}_j^{(i)}$. Consequently, observe that

$$\begin{aligned} \mathbf{d}_1^{(i)} &= \mathbf{b}^* \cdot \mathbf{C}_{1,i_1} \\ &= \mathbf{b}^* \cdot (\mathbf{B}_{i_1} (\mathbf{S}_1 \mathbf{A}_0 + \mathbf{E}_1) + \mathbf{E}'_{1,i_1}) \\ &= \mathbf{b}^* \underbrace{\mathbf{B}_{i_1}}_{=\mathbf{H}_1^{(i)}} \mathbf{S}_1 \mathbf{A}_0 + \underbrace{\mathbf{b}^* (\mathbf{B}_{i_1} \cdot \mathbf{E}_1 + \mathbf{E}'_{1,i_1})}_{=:\mathbf{e}_j^{(i)}} \\ &= \underbrace{\mathbf{b}^* \mathbf{H}_1^{(i)}}_{\mathbf{s}_1^{(i)}} \mathbf{S}_1 \mathbf{A}_0 + \mathbf{e}_j^{(i)} \\ &= \mathbf{s}_1^{(i)} \mathbf{A}_0 + \mathbf{e}_j^{(i)} \end{aligned}$$

that is, it holds that $\mathbf{e}_j^{(i)} = \mathbf{b}^* (\mathbf{B}_{i_1} \cdot \mathbf{E}_1 + \mathbf{E}'_{1,i_1})$. We can bound the matrix-norm of $\mathbf{B}_{i_1} \cdot \mathbf{E}_1 + \mathbf{E}'_{1,i_1}$ via

$$\|\mathbf{B}_{i_1} \cdot \mathbf{E}_1 + \mathbf{E}'_{1,i_1}\| \leq \underbrace{\|\mathbf{B}_{i_1}\|}_{\leq O(\sqrt{m})} \cdot \underbrace{\|\mathbf{E}_1\|}_{\leq O(\sigma\sqrt{m})} + \underbrace{\|\mathbf{E}'_{1,i_1}\|}_{\leq O(\sqrt{m})} \leq O(\sigma m).$$

Hence, it follows that

$$\|\mathbf{e}_1^{(i)}\| \leq \underbrace{\|\mathbf{b}^*\|}_{=1} \cdot \|\mathbf{B}_{i_1} \cdot \mathbf{E}_1 + \mathbf{E}'_{1,i_1}\| \leq O(\sigma m)$$

For $j = 2, \dots, d$ it holds that

$$\begin{aligned} \mathbf{d}_j^{(i)} &= \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{C}_{j,i_j} \\ &= \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot (\mathbf{B}_{i_j} (\mathbf{S}_j \mathbf{A} + \mathbf{E}_j) + \mathbf{E}'_{j,i_j} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_{i_j}) \\ &= \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot (\mathbf{B}_{i_j} \mathbf{S}_j \mathbf{A} - \mathbf{G}^\top \cdot \mathbf{S}_{j-1} \mathbf{G}_{i_j}) + \underbrace{\mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top (\mathbf{B}_{i_j} \mathbf{E}_j + \mathbf{E}'_{j,i_j})}_{=:\mathbf{e}_j^{(i)}} \\ &= \mathbf{b}^* \cdot \underbrace{\left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \cdot \mathbf{B}_{i_j} \mathbf{S}_j \mathbf{A}}_{=\mathbf{H}_j^{(i)}} - \underbrace{\mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \mathbf{G}^\top \cdot \mathbf{S}_{j-1} \mathbf{G}_{i_j}}_{=\mathbf{H}_{j-1}^{(i)}} + \mathbf{e}_j^{(i)} \\ &= \underbrace{\mathbf{b}^* \cdot \mathbf{H}_j^{(i)} \mathbf{S}_j \mathbf{A}}_{=\mathbf{s}_j^{(i)}} - \underbrace{\mathbf{b}^* \cdot \mathbf{H}_{j-1}^{(i)} \cdot \mathbf{S}_{j-1} \mathbf{G}_{i_j}}_{=\mathbf{s}_{j-1}^{(i)}} + \mathbf{e}_j^{(i)} \\ &= \mathbf{s}_j^{(i)} \mathbf{A} - \mathbf{s}_{j-1}^{(i)} \mathbf{G}_{i_j} + \mathbf{e}_j^{(i)}, \end{aligned}$$

where it holds that

$$\mathbf{e}_j^{(i)} = \mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top (\mathbf{B}_{i_j} \mathbf{E}_j + \mathbf{E}'_{j,i_j}).$$

Since $\left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top \in \{0, 1\}^{m \times m}$ it follows that

$$\|\mathbf{e}_j^{(i)}\| \leq \underbrace{\|\mathbf{b}^* \cdot \left(\mathbf{G}^{-1}((\mathbf{H}_{j-1}^{(i)})^\top) \right)^\top\|}_{\leq \sqrt{m}} \cdot \underbrace{\|\mathbf{B}_{i_j} \cdot \mathbf{E}_j + \mathbf{E}'_{j,i_j}\|}_{\leq O(\sigma m)} \leq O(\sigma \cdot m^{3/2}).$$

Recall that $\alpha_i = \lfloor \mathbf{d}^{(i)} \pi^{(i)} + \delta(i) \rfloor_2$. Now that we have bounded the errors in the $\mathbf{d}_j^{(i)}$, we are ready to show that, with high probability over the choice of $\delta(\cdot)$, there is an alternative way of computing α_i via

$$\alpha_i = \lfloor \mathbf{s}_d^{(i)} \cdot \text{com} + \delta(i) \rfloor_2,$$

That is, the value of α_i only depends on the hash value com , but not on the opening $\pi^{(i)}$. To see this, first note that

$$\begin{aligned} \mathbf{d}^{(i)} \pi^{(i)} &= (\mathbf{s}_1^{(i)} \mathbf{A}_0 + \mathbf{e}_1^{(i)}) \mathbf{x}_i + \sum_{j=1}^{d-1} (\mathbf{s}_{j+1}^{(i)} \mathbf{A} - \mathbf{s}_j^{(i)} \mathbf{G}_{i_{j+1}} + \mathbf{e}_{j+1}^{(i)}) \pi_j \\ &= \mathbf{s}_1^{(i)} \underbrace{\mathbf{A}_0 \mathbf{x}_i}_{=\mathbf{h}_1} + \sum_{j=1}^{d-1} (\mathbf{s}_{j+1}^{(i)} \underbrace{\mathbf{A} \pi_j}_{=\mathbf{h}_{j+1}} - \mathbf{s}_j^{(i)} \underbrace{\mathbf{G}_{i_{j+1}} \pi_j}_{=\mathbf{h}_j}) + \underbrace{\mathbf{e}_1^{(i)} \mathbf{x}_i + \sum_{j=1}^{d-1} \mathbf{e}_{j+1}^{(i)} \cdot \pi_j}_{=:\mathbf{e}_{\pi^{(i)}}} \\ &= \mathbf{s}_1^{(i)} \mathbf{h}_1 + \sum_{j=1}^{d-1} (\mathbf{s}_{j+1}^{(i)} \mathbf{h}_{j+1} - \mathbf{s}_j^{(i)} \mathbf{h}_j) + \mathbf{e}_{\pi^{(i)}} \\ &= \mathbf{s}_d^{(i)} \mathbf{h}_d + \mathbf{e}_{\pi^{(i)}} \\ &= \mathbf{s}_d^{(i)} \cdot \text{com} + \mathbf{e}_{\pi^{(i)}}. \end{aligned}$$

We can bound the norm of the error term $\mathbf{e}_{\pi^{(i)}}$ by

$$\begin{aligned} |\mathbf{e}_{\pi^{(i)}}| &= \|\mathbf{e}_1^{(i)} \mathbf{x}_i + \sum_{j=1}^{d-1} \mathbf{e}_j^{(i)} \cdot \pi_j\| \\ &\leq \underbrace{\|\mathbf{e}_1^{(i)}\|}_{\leq O(\sigma \cdot m^{3/2})} \cdot \underbrace{\|\mathbf{x}_i\|}_{\leq \sqrt{3m}} + \sum_{j=1}^{d-1} \underbrace{\|\mathbf{e}_j^{(i)}\|}_{\leq O(\sigma \cdot m^{3/2})} \cdot \underbrace{\|\pi_j\|}_{\leq \sqrt{2m}} \\ &\leq O(\sigma \cdot d \cdot m^2) =: B. \end{aligned}$$

Now fix $\mathbf{s}_d^{(i)}$, com and a function $\delta(\cdot)$ such that $\mathbf{s}_d^{(i)} \cdot \text{com} + \delta(i) \notin [q/4 - B, q/4 + B] \cup [3q/4 - B, 3q/4 + B]$. Then it holds by the above that

$$\begin{aligned} \alpha_i &= \lfloor \mathbf{d}^{(i)} \pi^{(i)} + \delta(i) \rfloor_2 \\ &= \lfloor \mathbf{s}_d^{(i)} \cdot \text{com} + \mathbf{e}_{\pi^{(i)}} + \delta(i) \rfloor_2 \\ &= \lfloor \mathbf{s}_d^{(i)} \cdot \text{com} + \delta(i) \rfloor_2, \end{aligned}$$

as $|\mathbf{e}_{\pi^{(i)}}| < B$.

Hence, for fixed $\mathbf{s}_d^{(i)}$, a commitment com and function $\delta(\cdot)$ call an index $i \in [k]$ *bad*, if $\mathbf{s}_d^{(i)} \cdot \text{com} + \delta(i) \in [q/4 - B, q/4 + B] \cup [3q/4 - B, 3q/4 + B]$. As a shorthand, we define the random variables $Y_1, \dots, Y_k$, where $Y_i = 1$ if and only if i is bad, otherwise $Y_i = 0$. Now, as $\delta(1), \dots, \delta(k)$ are m -wise independent, so are $Y_1, \dots, Y_k$. Consequently, it holds for all $i \in [k]$ that

$$E[Y_i] = \Pr[Y_i = 1] = \frac{4B}{q} \leq 1/k^2,$$

where the probability is taken over the choice of δ . This holds as $q = \omega(k^2 \cdot \sigma \cdot d \cdot m^2) > 4B \cdot k^2$.

Let $Y = \sum_{i=1}^k Y_i$ be the random variable that counts the number of bad indices. By linearity of expectation, it holds that

$$E[Y] = \sum_{i=1}^k E[Y_i] \leq 1/k.$$

Consequently, as the Y_i are m -wise independent, it holds by Lemma 7 that

$$\Pr[Y \geq m] \leq k^{-m},$$

where we recall that the probability is taken over the choice of the m -wise independent function $\delta(\cdot)$.

Now recall that $\text{com} \in \mathbb{Z}_q^n$, i.e., there are at most $q^n = 2^{n \log(q)} = 2^m$ syntactically correct choices for com . Hence we can conclude by a union bound that

$$\Pr[\exists \text{com s.t. } Y \geq m] \leq 2^m \cdot k^{-m} = (2/k)^m,$$

which is negligible. We conclude that it holds for all $\text{com} \in \mathbb{Z}_q^n$ that the number of bad indices is smaller than m , except with negligible probability over the choice of pp . Call such public parameters pp *good*.

For given good public parameters pp , set $\mathcal{V}^{\text{pp}}$ as the set of all tables $(\lfloor \mathbf{d}^{(i)} \tilde{\pi}^{(i)} + \delta(i) \rfloor_2)$, where the $\mathbf{d}^{(i)}$ and the $\delta(i)$ are derived from pp , but the $\tilde{\pi}^{(i)}$ are arbitrary binary strings for which there exists a single commitment com such that $\text{MerkleVer}(\text{com}, i, \mathbf{x}_i, \pi^{(i)}) = 1$ for all $i \in [k]$. By our discussion above, for good public parameters pp , the commitment $\text{com} \in \mathbb{Z}_q^n$ fixes the entire table $(\lfloor \mathbf{d}^{(i)} \tilde{\pi}^{(i)} + \delta(i) \rfloor_2)$, except for at most m bad positions. Counting conservatively, we may assume that these m bad positions can be chosen arbitrarily by choosing the corresponding openings $\tilde{\pi}^{(i)}$ appropriately. Hence, we can conclude that for good pp the set $\mathcal{V}^{\text{pp}}$ contains at most $2^m \cdot q^n = 2^{2m}$ strings. For bad public parameters we can, without loss of generality, choose $\mathcal{V}^{\text{pp}} = \emptyset$. Hence, we achieve output sparsity 2^{2m} , and the statistical binding property follows. This concludes the proof. $\square$

Theorem 9. *The scheme presented above is computationally hiding assuming the LWE assumption.*

Proof. We first show that the public parameters pp are pseudorandom under the LWE assumption.

Claim 1. *The public parameters pp are pseudorandom assuming the LWE assumption.*

- **Hybrid 0.** This is the real execution of the protocol. Note that in this hybrid the public parameters are of the form $\text{pp} = (\mathbf{B}_0, \mathbf{B}_1, \mathbf{A}, \mathbf{A}_0, \{\mathbf{C}_{i,b}\}_{i \in [d], b \in \{0,1\}}, \delta)$ where $\mathbf{B}_0, \mathbf{B}_1 \leftarrow \{0,1\}^{m \times n}$, $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times 3m}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times 2m}$, $\delta : [k] \rightarrow \mathbb{Z}_q$ is an m -wise independent function and

$$\mathbf{C}_{1,b} = \mathbf{B}_b \cdot (\mathbf{S}_1 \cdot \mathbf{A}_0 + \mathbf{E}_1) + \mathbf{E}'_{1,b}$$

and for $j = 2, \dots, d$

$$\mathbf{C}_{j,b} = \mathbf{B}_b \cdot (\mathbf{S}_j \cdot \mathbf{A} + \mathbf{E}_j) + \mathbf{E}'_{j,b} - \mathbf{G}^\top \mathbf{S}_{j-1} \mathbf{G}_b.$$

Now repeat the following sequence of hybrids for all $i = d, \dots, 1$ (starting in $i = d$).

- **Hybrid 1.** In this hybrid, for all $b \in \{0,1\}$ we compute $\mathbf{C}_{i,b}$ as

$$\mathbf{C}_{i,b} = \mathbf{B}_b \cdot \mathbf{U}_i + \mathbf{E}'_{i,b} - \mathbf{G}^\top \mathbf{S}_{i-1} \mathbf{G}_b$$

where $\mathbf{U}_i$ is a uniform matrix over $\mathbb{Z}_q^{n \times 3m}$ if $i = 1$ or $\mathbf{U}_i$ is a uniform matrix over $\mathbb{Z}_q^{n \times 2m}$ otherwise.

Indistinguishability of hybrids follows from the LWE assumption since

$$\mathbf{S}_i \cdot \mathbf{A} + \mathbf{E}_i \approx_c \mathbf{U}_i.$$

- **Hybrid 2.** In this hybrid, for all $b \in \{0, 1\}$ we set $\mathbf{C}_{i,b} \leftarrow \mathbb{Z}_q^{m \times 3m}$, if $i = 1$, or $\mathbf{C}_{i,b} \leftarrow \mathbb{Z}_q^{m \times 2m}$ otherwise. Indistinguishability of hybrids follows from Lemma 4. To see this, first note that for all $i \in [d]$

$$\begin{pmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \end{pmatrix} \cdot \mathbf{U}_i + \begin{pmatrix} \mathbf{E}'_{i,0} \\ \mathbf{E}'_{i,1} \end{pmatrix} \approx_c \begin{pmatrix} \mathbf{V}_{i,0} \\ \mathbf{V}_{i,1} \end{pmatrix}$$

where $\mathbf{V}_{i,0}, \mathbf{V}_{i,1} \leftarrow \mathbb{Z}_q^{m \times 3m}$ (resp., $\leftarrow \mathbb{Z}_q^{m \times 2m}$ if $i > 1$). This follows directly from Lemma 4. It follows that $\mathbf{C}_{i,b}$ is pseudorandom over $\mathbb{Z}_q^{m \times 3m}$ (resp., over $\mathbb{Z}_q^{m \times 2m}$ if $i > 1$).

Now fix any position $i^* \in [k]$. At this point, we use an information-theoretic argument to argue that the bit α_{i^*} is uniform even given all other hidden bits and corresponding local proofs $\{\alpha_i, \pi_i\}_{i \neq i^*}$. First note that $\pi_{i^*,0} = \mathbf{x}_{i^*}$ is contained only in π_{i^*} and not in other proofs $\{\pi_i\}_{i \neq i^*}$. Moreover, the only leakage on $\mathbf{x}_{i^*}$ that the adversary gets to see as part of the other proofs $\{\pi_i\}_{i \neq i^*}$ is $\mathbf{t} = \mathbf{A}_0 \cdot \mathbf{x}_{i^*} \in \mathbb{Z}_q^n$. Furthermore, in the opening π_{i^*} , all other values $\pi_{i^*,1}, \dots, \pi_{i^*,d}$ can be derived using this leakage.

Since $\mathbf{x}_{i^*} \leftarrow \{0, 1\}^{3m}$ and the leakage $\mathbf{t}$ is of size $n \log q$, we can use the chain rule for average min-entropy (Lemma 1) to argue that

$$\tilde{H}_\infty(\mathbf{x}_{i^*} | \mathbf{t}) \geq 2m.$$

By the argument above $\mathbf{C}_{1,i_1^*}$ is a uniform matrix. Hence $\mathbf{b}^* \cdot \mathbf{C}_{1,i_1^*}$, which is just the first row of $\mathbf{C}_{1,i_1^*}$, is a uniformly random vector in $\mathbb{Z}_q^{3m}$.

Hence we are in the conditions of the Leftover Hash Lemma 3 and thus we have that

$$\mathbf{b}^* \cdot \mathbf{C}_{1,i_1^*} \cdot \mathbf{x}_{i^*} \approx_s w$$

where $w \leftarrow \mathbb{Z}_q$ is distributed uniformly at random. Hence it follows that

$$\begin{aligned} \alpha_{i^*} &= \left\lfloor \mathbf{d}^{(i^*)} \pi^{(i^*)} + \delta(i^*) \right\rfloor_2 \\ &= \left\lfloor \underbrace{\mathbf{b}^* \cdot \mathbf{C}_{1,i_1^*} \mathbf{x}_{i^*}}_{\approx_s w} + (\mathbf{d}_2^{(i^*)}, \dots, \mathbf{d}_d^{(i^*)})^T (\pi_{i^*,1}, \dots, \pi_{i^*,d-1}) + \delta(i^*) \right\rfloor_2 \\ &\approx_s \lfloor w' \rfloor_2 \\ &\approx_s z \end{aligned}$$

where $w' \leftarrow \mathbb{Z}_q$ and $z \leftarrow \{0, 1\}$, and thus the bit α_{i^*} is statistically close to a uniform bit. This concludes the proof. $\square$

4.2 Efficiency

We now analyze the communication efficiency of the scheme presented in the previous section. We first analyze the size of public parameters and then the proof.

Size of public parameters. The public parameters $\mathbf{pp}$ consist of

$$(\mathbf{B}_0, \mathbf{B}_1, \mathbf{A}, \mathbf{A}_0, \{\mathbf{C}_{i,b}\}_{i \in [d], b \in \{0,1\}}, \delta)$$

where $\mathbf{B}_0, \mathbf{B}_1 \in \{0, 1\}^{m \times n}$, $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times 3m}$, $\mathbf{A} \in \mathbb{Z}_q^{n \times 2m}$, $\mathbf{C}_{1,b} \in \mathbb{Z}_q^{m \times 3m}$, $\mathbf{C}_{i,b} \in \mathbb{Z}_q^{m \times 2m}$ and $|\delta| = m \cdot \text{poly}(\lambda)$. Hence we have that $|\mathbf{pp}| = d \cdot \text{poly}(\lambda) = \log k \cdot \text{poly}(\lambda)$.

Size of proof. A proof π_i for an index i grows with $\text{poly}(\lambda)$. A proof for a subset of positions W consists of the concatenation of all the local proofs, hence it grows with $|W| \cdot \text{poly}(\lambda)$.

4.3 Lattice-Based NIZKs with Shorter CRS

Combining the above HBG construction with Theorem 7, we obtain a NIZK proof scheme (i.e., with statistical soundness) where the size of the CRS is $|C| \cdot \text{poly} \log |C| + \log |C| \cdot \text{poly}(\lambda)$.

Corollary 2. *Let $\mathcal{L}$ be an NP language with relation circuit C . Assuming that the LWE assumption holds, there exists a NIZK scheme for $\mathcal{L}$ with statistical soundness where CRS size is $|C| \cdot \text{polylog}(|C|) + \log |C| \cdot \text{poly}(\lambda)$ and the proof size is $|C| \cdot \text{poly}(\lambda, \log |C|)$.*

Remark 3. *We remark that the resulting NIZK scheme is actually dual-mode. That is, if we set the public parameters as a uniform string (as in the proof of Theorem 9), then the resulting NIZK scheme satisfies statistical zero knowledge.*

5 Hidden-Bits Generator with Batch Openings from Pairings

In this work, we use symmetric bilinear groups. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two groups of prime order p . We will denote by g and $g_T = e(g, g)$ their respective generators. Additionally, let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be the pairing operation.

For this construction, we rely on the computational Bilinear Diffie-Hellman (CBDH) assumption. We start by recalling this assumption.

Assumption 1 (Computational Bilinear Diffie-Hellman). *Let $(\mathbb{G}, \mathbb{G}_T, e)$ be a symmetric bilinear group of order p , where g is a generator of $\mathbb{G}$. We say that the computational Bilinear Diffie-Hellman (CBDH) assumption holds if for all non-uniform PPT adversaries $\mathcal{A}$, we have:*

$$\Pr [\mathcal{A}(g, g^\alpha, g^\beta, g^\gamma) = e(g, g)^{\alpha\beta\gamma}] \leq \text{negl}(\lambda)$$

where $\alpha, \beta, \gamma \leftarrow \mathbb{Z}_p$ are chosen uniformly at random.

We prove the following lemma that is useful in proving the security of our construction.

Lemma 9. *Assume that the computational Bilinear Diffie-Hellman assumption holds. Then, for any non-uniform PPT adversary $\mathcal{A}$, we have*

$$\Pr \left[\mathcal{A}(g, g^\alpha, g^\beta, g^\gamma, g^v) = \frac{e(g^\gamma, g^v)}{e(g, g)^{\alpha\beta\gamma}} \right] \leq \text{negl}(\lambda)$$

where $\alpha, \beta, \gamma, v \leftarrow \mathbb{Z}_p$ are chosen uniformly at random.

Proof. This follows directly from the CBDH assumption. The reduction chooses v uniformly at random and runs the adversary $\mathcal{A}$ on input $(g, g^\alpha, g^\beta, g^\gamma, g^v)$. Let z_T be the output of $\mathcal{A}$. The reduction computes $\frac{e(g^\gamma, g^v)}{z_T}$ as the CBDH solution. $\square$

5.1 Construction

We now present the construction of an HBG from pairings. In the following, let $(\mathbb{G}, \mathbb{G}_T, e)$ be a symmetric bilinear group of order p , where g is a generator of $\mathbb{G}$. Let hc_t be the Goldreich-Levin hardcore predicate with extraction randomness t .

Setup($1^\lambda, 1^k$):

- For $i \in [k]$ sample $x_i, r_i, y_i \leftarrow \mathbb{Z}_p$.

- Compute the matrix

$$\mathbf{M} = \begin{pmatrix} g^{r_1 x_1} \cdot g^{y_1} & g^{r_1 x_2} & \dots & g^{r_1 x_k} \\ g^{r_2 x_1} & g^{r_2 x_2} \cdot g^{y_2} & \dots & g^{r_2 x_k} \\ \vdots & & \ddots & \vdots \\ g^{r_k x_1} & g^{r_k x_2} & \dots & g^{r_k x_k} \cdot g^{y_k} \end{pmatrix}.$$

- Sample the Goldreich-Levin extraction randomness $t \leftarrow \{0, 1\}^{\lambda_T}$, where λ_T is the description length of $\mathbb{G}_T$ elements.
- Output $\text{pp} = (\mathbf{M}, \{g^{x_i}, g^{r_i}\}_{i \in [k]}, t)$.

GenBits(pp):

- Parse pp as $(\mathbf{M}, \{g^{x_i}, g^{r_i}\}_{i \in [k]}, t)$.
- Sample $s \leftarrow \mathbb{Z}_p$ and compute $\text{com} = g^s$.
- For $i \in [k]$ compute $\alpha_i = \text{hc}_t \left(\left(\frac{e(\mathbf{M}_{i,i}, g)}{e(g^{r_i}, g^{x_i})} \right)^s \right)$.
- For any subset⁵ W of $[k]$, compute the proof $\pi_W = \left(\prod_{j \in W} g^{x_j} \right)^s$.
- Output $(\text{com}, \alpha_1, \dots, \alpha_k, \pi_W)$.

Verify(pp, com, W , $\{\beta_i\}_{i \in W}$, π_W):

- Parse pp as $(\mathbf{M}, \{g^{x_i}, g^{r_i}\}_{i \in [k]}, t)$.
- Accept if
 1. $e(\text{com}, \prod_{j \in W} g^{x_j}) = e(\pi_W, g)$, and
 2. For every $i \in W$, check whether $\beta_i = \text{hc}_t \left(\frac{e(\text{com}, \prod_{j \in W} \mathbf{M}_{i,j})}{e(\pi_W, g^{r_i})} \right)$.

If any of the checks fail, reject.

5.2 Analysis

Theorem 10. *The scheme presented above is complete.*

Proof. Let $\text{com} = g^s$ and $\pi_W = \left(\prod_{j \in W} g^{x_j} \right)^s = g^{s \cdot \sum_{j \in W} x_j}$. First, we have that

$$e \left(\text{com}, \prod_{j \in W} g^{x_j} \right) = e \left(g^s, \prod_{j \in W} g^{x_j} \right) = e(g, g)^{s \cdot \sum_{j \in W} x_j} = e(\pi_W, g)$$

and the first check succeeds. Additionally, we have that

$$\frac{e(\text{com}, \prod_{j \in W} \mathbf{M}_{i,j})}{e(\pi_W, g^{r_i})} = \frac{e(g, g)^{sr_i \sum_{j \in W} x_j + sy_i}}{e(g, g)^{sr_i \sum_{j \in W} x_j}} = e(g, g)^{sy_i}$$

⁵In the transformation from HBG to NIZKs in the CRS model [FLS90, QRW19], the set of indices that need to be opened is determined by $(\alpha_1, \dots, \alpha_k)$.

and that

$$\left(\frac{e(\mathbf{M}_{i,i}, g)}{e(g^{r_i}, g^{x_i})} \right)^s = \frac{e(g, g)^{sr_i x_i + sy_i}}{e(g, g)^{sr_i x_i}} = e(g, g)^{sy_i}.$$

Hence $\beta_i = \alpha_i$ and the scheme is correct. $\square$

Theorem 11. *The scheme presented above is statistically binding.*

Proof. First, note that the commitment consists of a single group element g^s . Hence, the number of possible commitments is 2^λ . We show that for a particular $\mathbf{pp}$ and commitment $\mathbf{com}$, an index i can be opened to just one value α_i .

If verification succeeds, then $e(\mathbf{com}, \prod_{j \in W} g^{x_j}) = e(g, g)^{s \sum_{j \in W} x_j} = e(\pi_W, g)$ for some $s \in \mathbb{Z}_p$, which means that $\pi_W = \mathbf{com}^{\sum_{j \in W} x_j}$.

Then, we have that

$$\frac{e(\mathbf{com}, \prod_{j \in W} \mathbf{M}_{i,j})}{e(\pi_W, g^{r_i})} = \frac{e(g, g)^{sr_i \sum_{j \in W} x_j + sy_i}}{e(g, g)^{sr_i \sum_{j \in W} x_j}} = e(g, g)^{sy_i}$$

where we replace $\pi_W = \mathbf{com}^{\sum_{j \in W} x_j}$. This means that the bit α_i is fully determined given the $\mathbf{pp}$ and the commitment $\mathbf{com}$.

This means that we can set

$$\mathcal{V}^{\mathbf{pp}} = \{ (\text{hc}_t(e(g, \mathbf{com})^{y_1}), \dots, \text{hc}_t(e(g, \mathbf{com})^{y_k})) : \mathbf{com} \in \mathbb{G} \}.$$

The size of $\mathcal{V}^{\mathbf{pp}}$ is at most 2^λ . Thus, it satisfies output sparsity. The statistical binding property was argued above. $\square$

Theorem 12. *The scheme presented above is hiding assuming the CBDH assumption over $(\mathbb{G}, \mathbb{G}_T, e)$.*

Proof. We will show that if there is an adversary $\mathcal{A}$ that breaks the hiding game with non-negligible probability ϵ , then there is a distinguisher $\mathcal{D}$ that, given (g, g^a, g^b, g^c, g^v) , can distinguish between $\text{hc}_t(\frac{e(g^c, g^v)}{e(g, g)^{abc}})$ and a uniformly chosen bit with advantage ϵ . From Lemma 9 and the Goldreich-Levin theorem, this is impossible.

The distinguisher $\mathcal{D}$ obtains $(g, g^a, g^b, g^c, g^v, t)$ and a bit z and does the following:

- Run $\mathcal{A}$ to obtain an index i^* and a subset $W \subset [k]$ such that $i^* \notin W$.
- Generate $\mathbf{pp}$ as in the real scheme except that it sets $g^{r_{i^*}} = g^a$, $g^{x_{i^*}} = g^b$, and $\mathbf{M}_{i^*, i^*} = g^v$ where we interpret $v = r_{i^*} x_{i^*} + y_{i^*}$ for randomly chosen $y_{i^*} \in \mathbb{Z}_p$. For every $i \neq i^*$, we sample x_i, y_i, r_i uniformly from $\mathbb{Z}_p$. We generate $\mathbf{pp}$ using these values.
- Set $\mathbf{com} = g^c$ and $\pi_W = g^{c \sum_{j \in W} x_j}$.
- For every $i \in W$, compute $\beta_i = \text{hc}_t\left(\frac{e(\mathbf{com}, \prod_{j \in W} \mathbf{M}_{i,j})}{e(\pi_W, g^{r_i})}\right)$ as in the verification procedure.
- Set $\alpha_{i^*} = z$.
- Output whatever $\mathcal{A}$ outputs.

We now analyze the advantage of the distinguisher $\mathcal{D}$. If $z = \text{hc}_t(\frac{e(g^c, g^v)}{e(g, g)^{abc}})$, then

$$\frac{e(\mathbf{com}, g^v)}{e(g, g)^{abc}} = \frac{e(g, g)^{cab + cy_{i^*}}}{e(g, g)^{abc}} = e(g, g)^{sy_{i^*}}$$

since $c = s$. In this case, the distribution of the game is identical to the real scheme.

On the other hand, if z is uniform, then the view of the adversary is identical to the case where the hidden bit is replaced with a uniform bit.

Thus, we conclude that the advantage of the distinguisher is exactly the same as the advantage of the adversary $\mathcal{A}$. $\square$

5.3 Efficiency

We now analyze the efficiency of the scheme in terms of communication.

Size of public parameters. The public parameters $\mathbf{pp}$ consist of the matrix $\mathbf{M}$, elements $\{g^{x_i}, g^{r_i}\}_{i \in [k]}$, and the string t . Hence, we have that

$$|\mathbf{pp}| = k^2 \cdot \text{poly}(\lambda).$$

Size of proof. We now analyze the size of the commitment and local proofs. The commitment $\mathbf{com}$ consists of a single group element from the source group, hence $|\mathbf{com}| = \text{poly}(\lambda)$. A local proof π_W for a subset $W \subseteq [k]$ consists of a single group element from the source group, hence $|\pi_W| = \text{poly}(\lambda)$.

5.4 Pairing-Based NIZKs with Shorter Proofs

Combining the above HBG construction with Theorem 7, we obtain a NIZK proof scheme (i.e., with statistical soundness) where proofs are of size $|C| \cdot \text{polylog}(|C|) + \text{poly}(\lambda)$.

Corollary 3. *Let $\mathcal{L}$ be an NP language with relation circuit C . Assuming that the CBDH assumption holds, there exists a NIZK scheme for $\mathcal{L}$ with statistical soundness where proofs are of size $|C| \cdot \text{polylog}(|C|) + \text{poly}(\lambda)$ and the CRS size is $|C|^2 \cdot \text{poly}(\lambda, \log |C|)$.*

6 Hidden-Bits Generators with Linear-Sized CRS and Batch Openings from Pairings

In this section, we improve the CRS size of the HBG construction from pairings so that it is linear in the output length k , using a stronger q -type assumption.

We start by introducing our new assumption, the q -Computational Bilinear Square-Pole eXternal Diffie-Hellman (q -XSqPole) assumption, which is adapted from the q -computational Bilinear Square-Pole Diffie-Hellman assumption introduced in [JNR26]. We justify the q -XSqPole assumption with a proof in the generic bilinear group model, provided in Appendix A.

Assumption 2 (q -Computational Bilinear Square-Pole eXternal Diffie-Hellman (q -XSqPole) assumption). *Let $(\mathbb{G}, \mathbb{G}_T, e)$ be a symmetric bilinear group of order p , where g is a generator of $\mathbb{G}$. We say that the q -XSqPole assumption holds if, for all non-uniform PPT adversaries $\mathcal{A}$ and all $i^* \in [q]$, we have:*

$$\Pr \left[\mathcal{A} \left(g, g^s, \left\{ g^{\frac{1}{x+i}} \right\}_{i \in [q]}, \left\{ g^{\frac{s}{x+i}} \right\}_{i \in [q] \setminus \{i^*\}} \right) = e(g, g)^{\frac{s}{(x+i^*)^2}} \right] \leq \text{negl}(\lambda)$$

where $s, x \leftarrow \mathbb{Z}_p$ are chosen uniformly at random, and the fractions are defined in the ring $\mathbb{Z}_p$.

We also prove the following lemma tailored to the security proof of our construction, which is a direct consequence of the q -XSqPole assumption.

Lemma 10. *Assume that the q -XSqPole assumption holds. Then, for any non-uniform PPT adversary $\mathcal{A}$ and any $i^* \in [q]$, we have*

$$\Pr \left[\mathcal{A} \left(g, g^s, \left\{ g^{\frac{1}{x+i}} \right\}_{i \in [q]}, \left\{ g^{\frac{s}{x+i}} \right\}_{i \in [q] \setminus \{i^*\}}, g^{\frac{1}{(x+i^*)^2} + y_{i^*}} \right) = e(g, g)^{sy_{i^*}} \right] \leq \text{negl}(\lambda)$$

where $s, x, y_{i^*} \leftarrow \mathbb{Z}_p$ are chosen uniformly at random.

Proof. Since y_{i^*} is chosen uniformly at random, the distribution of $g^{\frac{1}{(x+i^*)^2} + y_{i^*}}$ is uniform in $\mathbb{G}$ and independent of the other group elements in the adversary's input. Therefore, we can reduce the above guarantee to the q -XSqPole assumption by forwarding the q -XSqPole challenge to $\mathcal{A}$ and sampling $g^v = g^{\frac{1}{(x+i^*)^2} + y_{i^*}}$ with random v . Whenever the adversary wins with some output Y , it must satisfy

$$Y = e(g, g)^{sy_{i^*}} = e(g, g)^{sv - \frac{s}{(x+i^*)^2}} = e(g, g^s)^v / e(g, g)^{\frac{s}{(x+i^*)^2}}.$$

By computing $(e(g, g^s)^v) / Y$, the reduction obtains $e(g, g)^{\frac{s}{(x+i^*)^2}}$ and wins the q -XSqPole challenge. $\square$

6.1 Construction

We now present the HBG construction from pairings with improved CRS size. In the following, let $(\mathbb{G}, \mathbb{G}_T, e)$ be a symmetric bilinear group of order p , where g is a generator of $\mathbb{G}$. Let hc_t be the Goldreich-Levin hardcore predicate with extraction randomness t .

Setup($1^\lambda, 1^k$):

- Sample $x \leftarrow \mathbb{Z}_p \setminus \{-1, \dots, -k\}$. Set $r_i = \frac{1}{x+i}$ for $i \in [k]$.
- For each $i \in [k]$, sample $y_i \leftarrow \mathbb{Z}_p$, and set $m_i = \frac{1}{(x+i)^2} + y_i$.
- Sample the Goldreich-Levin extraction randomness $t \leftarrow \{0, 1\}^{\lambda_T}$, where λ_T is the description length of $\mathbb{G}_T$ elements.
- Output $\text{pp} = (\{g^{m_i}, g^{r_i}\}_{i \in [k]}, t)$.

GenBits(pp):

- Parse pp as $(\{g^{m_i}, g^{r_i}\}_{i \in [k]}, t)$.
- Sample $s \leftarrow \mathbb{Z}_p$ and compute $\text{com} = g^s$.
- For $i \in [k]$ compute $\alpha_i = \text{hc}_t \left(\left(\frac{e(g^{m_i}, g)}{e(g^{r_i}, g^{r_i})} \right)^s \right)$.
- For any subset⁶ W of $[k]$, compute the proof $\pi_W = \left(\prod_{j \in W} g^{r_j} \right)^s$.
- Output $(\text{com}, \alpha_1, \dots, \alpha_k, \pi_W)$.

Verify($\text{pp}, \text{com}, W, \{\beta_i\}_{i \in W}, \pi_W$):

- Parse pp as $(\{g^{m_i}, g^{r_i}\}_{i \in [k]}, t)$.
- Accept if
 1. $e(\text{com}, \prod_{j \in W} g^{r_j}) = e(\pi_W, g)$, and
 2. For every $i \in W$, check if $\beta_i = \text{hc}_t \left(\frac{e(\text{com}, g^{m_i}) \cdot \prod_{j \in W \setminus \{i\}} e(\text{com}, (g^{r_j} / g^{r_i})^{\frac{1}{i-j}})}{e(\pi_W, g^{r_i})} \right)$

If any of the checks fail, reject.

⁶In the transformation from HBG to NIZKs in the CRS model [FLS90, QRW19], the set of indices that need to be opened is determined by $(\alpha_1, \dots, \alpha_k)$.

6.2 Analysis

Theorem 13. *The scheme presented above is complete.*

Proof. Let $\text{com} = g^s$ and $\pi_W = \left(\prod_{j \in W} g^{r_j}\right)^s$ be honestly computed. Then

$$e\left(\text{com}, \prod_{j \in W} g^{r_j}\right) = e(g, g)^{s \cdot \sum_{j \in W} r_j} = e(\pi_W, g),$$

and the first check always passes. For the second check, we have that for every $i \in W$, the honest hidden bit α_i is computed as

$$\alpha_i = \text{hc}_t\left(\left(\frac{e(g^{m_i}, g)}{e(g^{r_i}, g^{r_i})}\right)^s\right) = \text{hc}_t\left(e(g, g)^{s \cdot m_i - s \cdot r_i^2}\right) = \text{hc}_t\left(e(g, g)^{s \cdot y_i}\right).$$

On the other hand, the right-hand side of the second check is computed as

$$\begin{aligned} \frac{e(\text{com}, g^{m_i}) \cdot \prod_{j \in W \setminus \{i\}} e\left(\text{com}, (g^{r_j} / g^{r_i})^{\frac{1}{i-j}}\right)}{e(\pi_W, g^{r_i})} &= \frac{e(g, g)^{s \cdot m_i} \cdot \prod_{j \in W \setminus \{i\}} e(g, g)^{s \cdot \frac{r_j - r_i}{i-j}}}{e(g, g)^{s \cdot r_i \sum_{j \in W} r_j}} \\ &= e(g, g)^{s \cdot m_i + s \cdot \sum_{j \in W \setminus \{i\}} \frac{r_j - r_i}{i-j} - s \cdot r_i \sum_{j \in W} r_j} \\ &= e(g, g)^{s \cdot m_i + s \cdot \sum_{j \in W \setminus \{i\}} r_i r_j - s \cdot r_i \sum_{j \in W} r_j} \\ &= e(g, g)^{s \cdot m_i - s \cdot r_i^2} = e(g, g)^{s \cdot y_i}, \end{aligned}$$

where the third equality follows from the fact that $\frac{r_j - r_i}{i-j} = \frac{1}{i-j} \cdot \left(\frac{1}{x+j} - \frac{1}{x+i}\right) = \frac{1}{(x+i)(x+j)} = r_i r_j$. Hence, $\beta_i = \text{hc}_t(e(g, \text{com})^{y_i}) = \alpha_i$, so the second check always passes. Combining the above, we conclude that the scheme is complete. $\square$

Theorem 14. *The scheme presented above is statistically binding.*

Proof. First, we show that for a particular pp and commitment com , an index i can be opened to just one value α_i .

Given $(\text{pp}, \text{com}, W, \{\beta_i\}_{i \in W}, \pi_W)$ that passes verification, and $\text{com} = g^s$, by the first verification check we have that

$$e\left(\text{com}, \prod_{j \in W} g^{r_j}\right) = e(g, g)^{s \cdot \sum_{j \in W} r_j} = e(\pi_W, g),$$

which means that $\pi_W = g^{s \cdot \sum_{j \in W} r_j}$. Then, for every $i \in W$, plugging in the value of π_W into the second verification check gives

$$\begin{aligned} \beta_i &= \text{hc}_t\left(\frac{e(\text{com}, g^{m_i}) \cdot \prod_{j \in W \setminus \{i\}} e\left(\text{com}, (g^{r_j} / g^{r_i})^{\frac{1}{i-j}}\right)}{e(\pi_W, g^{r_i})}\right) \\ &= \text{hc}_t\left(e(g, g)^{s \cdot m_i + s \cdot \sum_{j \in W \setminus \{i\}} \frac{r_j - r_i}{i-j} - s \cdot r_i \sum_{j \in W} r_j}\right) \\ &= \text{hc}_t\left(e(g, g)^{s \cdot m_i - s \cdot r_i^2}\right) = \text{hc}_t\left(e(g, g)^{s \cdot y_i}\right) = \text{hc}_t\left(e(g, \text{com})^{y_i}\right), \end{aligned}$$

which is uniquely determined by pp and com .

For output sparsity, we can set $\mathcal{V}^{\text{pp}} = \{(\text{hc}_t(e(g, \text{com})^{y_1}), \dots, \text{hc}_t(e(g, \text{com})^{y_k})) : \text{com} \in \mathbb{G}\}$. The size of $\mathcal{V}^{\text{pp}}$ is $|\mathbb{G}|$, which is $2^{\text{poly}(\lambda)}$. Thus, it satisfies output sparsity. This concludes the proof of statistical binding. $\square$

Theorem 15. *The scheme presented above is hiding assuming the q -XSqPole assumption over $(\mathbb{G}, \mathbb{G}_T, e)$.*

Proof. We will show that an adversary $\mathcal{A}$ that breaks the hiding property of the scheme implies a distinguisher $\mathcal{D}$ that distinguishes the distributions

$$D_b = \left(g, g^s, \left\{ g^{\frac{1}{x+i}} \right\}_{i \in [q]}, \left\{ g^{\frac{s}{x+i}} \right\}_{i \in [q] \setminus \{i^*\}}, g^{\frac{1}{(x+i^*)^2} + y_{i^*}}, t, \alpha_{i^*} \right),$$

where $\alpha_{i^*} = \text{hc}_t(e(g, g)^{sy_{i^*}})$ if $b = 0$, and $\alpha_{i^*} \leftarrow \{0, 1\}$ is uniform if $b = 1$.

The distinguisher $\mathcal{D}$ works as follows.

- Run $\mathcal{A}$ to obtain an index i^* and a subset $W \subset [k]$ such that $i^* \notin W$.
- Receive from its challenger the challenge tuple

$$\left(g, g^s, \left\{ g^{\frac{1}{x+i}} \right\}_{i \in [q]}, \left\{ g^{\frac{s}{x+i}} \right\}_{i \in [q] \setminus \{i^*\}}, g^{\frac{1}{(x+i^*)^2} + y_{i^*}}, t, \alpha_{i^*} \right).$$

- To generate pp , first set $g^{m_{i^*}} = g^{\frac{1}{(x+i^*)^2} + y_{i^*}}$ from the challenge. For the remaining indices $i \neq i^*$, sample random $m_i \leftarrow \mathbb{Z}_p$ and set g^{m_i} , implicitly defining $y_i = m_i - \frac{1}{(x+i)^2}$. For every $i \in [k]$, set $g^{r_i} = g^{\frac{1}{x+i}}$ from the challenge. Also, set the extraction randomness t from the challenge.
- Set $\text{com} = g^s$ and $\pi_W = g^{\sum_{j \in W} \frac{s}{x+j}} = \prod_{j \in W} g^{\frac{s}{x+j}}$ with elements from the challenge.
- For each $i \in W$, compute $\alpha_i = \text{hc}_t \left(\frac{e(\text{com}, g^{m_i}) \cdot \prod_{j \in W \setminus \{i\}} e(\text{com}, (g^{r_j} / g^{r_i})^{\frac{1}{x-j}})}{e(\pi_W, g^{r_i})} \right)$ as in the verification condition.
- Run the adversary $\mathcal{A}(\text{pp}, \text{com}, \{\alpha_i\}_{i \in W}, \pi_W, \alpha_{i^*})$ and output whatever $\mathcal{A}$ outputs.

It is easy to see that the distinguisher $\mathcal{D}$ simulates the view of $\mathcal{A}$, except with negligible probability in λ . In particular, conditioning on $x \notin \{-1, \dots, -k\}$ (except with negligible probability in λ), the term g^{m_i} is distributed identically in the simulation and in the real scheme, and the bits α_i for $i \in W$ are equal to $\text{hc}_t(e(g, g)^{sm_i - sr_i^2})$, which is exactly the same as in the real scheme given $y_i = m_i - \frac{1}{(x+i)^2}$. When $\alpha_{i^*} = \text{hc}_t(e(g, g)^{sy_{i^*}})$, the distribution of the simulation is identical to the real scheme. On the other hand, when α_{i^*} is uniform, the view of $\mathcal{A}$ is identical to the case where the hidden bit is a uniform bit. Thus, we conclude that the advantage of $\mathcal{D}$ in distinguishing D_0 and D_1 is exactly the same (up to an additive negligible factor in λ) as the advantage of $\mathcal{A}$ in breaking the hiding property of the scheme.

By Lemma 10 and the Goldreich-Levin theorem, no distinguisher $\mathcal{D}$ can distinguish D_0 and D_1 with non-negligible advantage. This implies that no adversary $\mathcal{A}$ can break the hiding property of the scheme with non-negligible advantage assuming the q -XSqPole assumption, concluding the proof. $\square$

6.3 Efficiency

We now analyze the efficiency of the scheme in terms of communication.

Size of public parameters. The public parameters pp consist of $2k$ group elements $(\{g^{m_i}, g^{r_i}\}_{i \in [k]})$ and the extraction randomness t . Hence, we have that

$$|\text{pp}| = k \cdot \text{poly}(\lambda).$$

Size of proof. We now analyze the size of the commitment and local proofs. Both the commitment $\text{com} = g^s$ and the local proof $\pi_W = g^{\sum_{j \in W} \frac{s}{x+j}}$ consist of a single group element from the source group, hence $|\text{com}| = \text{poly}(\lambda)$ and $|\pi_W| = \text{poly}(\lambda)$.

6.4 Pairing-Based NIZKs with Shorter Proofs

Combining the above HBG construction with Theorem 7, we obtain a NIZK proof scheme (i.e., with statistical soundness) where proofs are of size $|C| \cdot \text{polylog}(|C|) + \text{poly}(\lambda)$.

Corollary 4. *Let $\mathcal{L}$ be an NP language with relation circuit C . Assuming that the q -XSqPole assumption holds, there exists a NIZK scheme for $\mathcal{L}$ with statistical soundness where proofs are of size $|C| \cdot \text{polylog}(|C|) + \text{poly}(\lambda)$ and the CRS size is $|C| \cdot \text{poly}(\lambda, \log |C|)$.*

7 Hidden-Bits Generators with Short CRS and Batch Openings from RSA

In this section, we give a construction of a hidden-bits generator with batch openings from the RSA [RSA78] assumption. When compared to the pairing-based construction, the size of the public parameters in this case is $\text{poly}(\lambda)$. We first recall this assumption, tailored to the case where e is chosen to be prime. This version was shown to be reducible to the original RSA assumption in [ABI⁺23]. In the following, all group operations are done over the multiplicative group $\mathbb{Z}_N^*$.

Assumption 3 (RSA assumption). *For every non-uniform PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the following game is negligible.*

- Sample two random λ -bit primes p and q . Set $N = p \cdot q$.
- Sample e to be a random λ -bit prime.
- Sample $x \leftarrow \mathbb{Z}_N^*$.
- Run $\mathcal{A}(N, e, x^e)$. Let x' be the output of $\mathcal{A}$.
- $\mathcal{A}$ wins the game if $x = x'$.

We recall the following lemma about a special sampler from [HW09, ABI⁺23].

Lemma 11 ([HW09, ABI⁺23]). *There exist PPT algorithms **Samp** and **Plant** with the following properties:*

- The output of **Samp** $(1^\lambda, 1^k)$ is a tuple of k distinct λ -bit primes $(e_1, \dots, e_k)$ with probability $1 - \text{negl}(\lambda)$.
- The randomness complexity of **Samp** is $\lambda^3 \cdot \log k$.
- The marginal distribution of e_i for any $i \in [k]$ output by **Samp** is uniform over λ -bit primes.
- The **Plant** procedure takes $(1^\lambda, 1^k)$, an index i and a λ -bit prime e and produces a random string ρ with the following properties:
 - The outcome of **Samp** $(1^\lambda, 1^k; \rho)$ is $(e_1, \dots, e_k)$ such that $e_i = e$.
 - ρ is statistically close to uniform if the input e to **Plant** is a uniformly chosen λ -bit prime.

7.1 Construction

We are now ready to describe our HBG construction from the RSA assumption. Let hc_t be the Goldreich-Levin hardcore predicate with extraction randomness t .

Setup($1^\lambda, 1^k$):

- Sample two random λ -bit primes p and q and compute $N = p \cdot q$.
- Sample $\rho \leftarrow \{0, 1\}^{\lambda^3 \log k}$ as the random string for **Samp**.
- Sample the Goldreich-Levin extraction randomness $t \leftarrow \{0, 1\}^{\log N}$.
- Output $\text{pp} = (N, \rho, t)$.

GenBits(pp):

- Parse pp as (N, ρ, t) .
- Compute $(e_1, \dots, e_k) \leftarrow \text{Samp}(1^\lambda, 1^k; \rho)$.
- Sample $u \leftarrow \mathbb{Z}_N^*$.
- Compute $\text{com} = u^{\prod_{i \in [k]} e_i}$.
- For every $i \in [k]$, compute $\alpha_i = \text{hc}_t(u^{\prod_{j \neq i} e_j})$.
- For any subset⁷ $W \subseteq [k]$, compute $\pi_W = u^{\prod_{i \notin W} e_i}$.
- Output $(\text{com}, \alpha_1, \dots, \alpha_k, \pi_W)$.

Verify($\text{pp}, \text{com}, W, \{\beta_i\}_{i \in W}, \pi_W$):

- Parse pp as (N, ρ, t) .
- Compute $(e_1, \dots, e_k) \leftarrow \text{Samp}(1^\lambda, 1^k; \rho)$.
- Check if $\text{com} \stackrel{?}{=} (\pi_W)^{\prod_{j \in W} e_j}$.
- Check if for each $i \in W$, $\beta_i \stackrel{?}{=} \text{hc}_t(\pi_W^{\prod_{j \in W, j \neq i} e_j})$.

7.2 Analysis

Theorem 16. *The scheme presented above is complete.*

Proof. Let us fix an arbitrary subset $W \subseteq [k]$. By definition, $\text{com} = u^{\prod_{i \in [k]} e_i}$ and $\pi_W = u^{\prod_{j \notin W} e_j}$. Hence,

$$\begin{aligned} \pi_W^{\prod_{j \in W} e_j} &= u^{\prod_{j \in W} e_j \prod_{j \notin W} e_j} \\ &= \text{com} \end{aligned}$$

Let $i \in W$. In our construction, we set $\alpha_i = \text{hc}_t(u^{\prod_{j \neq i} e_j})$. Observe that

$$\begin{aligned} \pi_W^{\prod_{j \in W, j \neq i} e_j} &= u^{\prod_{j \in W, j \neq i} e_j \prod_{j \notin W} e_j} \\ &= u^{\prod_{j \neq i} e_j} \end{aligned}$$

Therefore, both checks done by **Verify** pass on honestly computed α_i and π_W . □

Theorem 17. *The scheme presented above satisfies statistical binding.*

⁷In the transformation from HBG to NIZKs in the CRS model [FLS90, QRW19], the set of indices that need to be opened is determined by $(\alpha_1, \dots, \alpha_k)$.

Proof. Note that com consists of an element in $\mathbb{Z}_N^*$. The length of com is 2λ bits. Now, let us fix a CRS (N, ρ) and a com . We will prove that any $i \in [k]$ can only be opened to a single value.

Let W be any subset of $[k]$ such that $i \in W$. Let π_W be the purported opening proof provided by a cheating prover. Note that the verifier performs two checks:

- $\text{com} \stackrel{?}{=} (\pi_W)^{\prod_{j \in W} e_j}$.
- $\beta_i \stackrel{?}{=} \text{hc}_t(\pi_W^{\prod_{j \in W, j \neq i} e_j})$.

These two checks imply that $\beta_i = \text{hc}_t(\text{com}^{1/e_i})$. Thus, when we fix com and the CRS, which fixes $(N, e_1, \dots, e_k)$, the prover can only open each index to $\text{hc}_t(\text{com}^{1/e_i})$. Thus, we can set

$$\mathcal{V}^{\text{pp}} = \left\{ (\text{hc}_t(\text{com}^{1/e_1}), \dots, \text{hc}_t(\text{com}^{1/e_k})) : \text{com} \in \mathbb{Z}_N^* \right\}.$$

The size of $\mathcal{V}^{\text{pp}}$ is at most $2^{2\lambda}$. Thus, it satisfies output sparsity. The statistical binding property was argued above. $\square$

Theorem 18. *Under the RSA assumption, the scheme presented above satisfies computational hiding.*

Proof. Suppose there is a non-uniform PPT adversary $\mathcal{A}$ that breaks the computational hiding property at index $i \in [k]$ and for any subset $W \subseteq [k]$ such that $i \notin W$. We give a reduction that breaks the RSA assumption.

We are given an RSA challenge $(N, e, z = y^e)$ and are tasked with finding y . We give a reduction that finds y :

- The reduction uses the **Plant** procedure on input $(1^\lambda, 1^k, i, e)$ to sample ρ .
- It computes $(e_1, \dots, e_k) \leftarrow \text{Samp}(1^\lambda, 1^k; \rho)$. Note that we have $e_i = e$.
- It computes $\text{com} = z^{\prod_{j \neq i} e_j}$, $\pi_W = z^{\prod_{j \in W, j \neq i} e_j}$, and $\alpha_j = \text{hc}_t(z^{\prod_{\ell \notin \{i, j\}} e_\ell})$ for all $j \in W$.
- It runs the Goldreich-Levin decoder on input (N, e, com) . Whenever the decoder queries its distinguishing oracle with (com, b) and randomness t , the reduction runs $\mathcal{A}(\text{pp} = (N, \rho, t), \text{com}, \{\alpha_j\}_{j \in W}, \pi_W, b)$. It returns whatever $\mathcal{A}$ outputs to the decoder. Let y' be the output of the decoder.
- If $(y')^e \neq \text{com}$, we abort and output failure.
- Else, the reduction uses the extended Euclidean algorithm to find a, b such that $a \prod_{j \neq i} e_j + be = 1$ (note that the gcd of $\prod_{j \neq i} e_j$ and e_i is 1). It outputs $y = (y')^a \cdot z^b$.

We first note that if $(y')^e = \text{com}$, then $y' = y^{\prod_{j \neq i} e_j}$. Thus, the last step of the reduction outputs the correct y . The running time of the reduction is polynomial since both Goldreich-Levin decoding and the extended Euclidean algorithm run in polynomial time.

Let us now analyze the success probability of the reduction. First, from Lemma 11, we infer that the output of the **Plant** procedure is statistically close to uniform. Note that we are guaranteed that:

$$\left| \Pr[\mathcal{A}(\text{pp}, \text{com}, \{\alpha_j\}_{j \in W}, \pi_W, \text{hc}_t(\text{com}^{1/e_i})) = 1] - \Pr[\mathcal{A}(\text{pp}, \text{com}, \{\alpha_j\}_{j \in W}, \pi_W, b) = 1] \right| \geq \epsilon(\lambda)$$

where $b \leftarrow \{0, 1\}$ and $\epsilon(\lambda)$ is non-negligible. Furthermore, (N, e, com) is distributed identically to a random RSA challenge as $y^{\prod_{j \neq i} e_j}$ is uniformly distributed in $\mathbb{Z}_N^*$ (this follows from the fact that the RSA function is a permutation). Therefore, with non-negligible probability $\mu(\lambda)$, the Goldreich-Levin decoder outputs y' such that $(y')^e = \text{com}$. If this is the case, then the last step always outputs the correct y . Therefore, the success probability of our reduction is $\mu(\lambda) - \text{negl}(\lambda)$ (where $\text{negl}(\lambda)$ is to account for the **Plant** procedure) which is non-negligible. $\square$

7.3 Efficiency

We now analyze the efficiency of the scheme.

Size of public parameters. The public parameters pp consist of (N, ρ, t) . Hence, $|\text{pp}| = \text{poly}(\lambda, \log k)$.

Size of proof. We now analyze the size of the commitment and batch proofs. The commitment com consists of a single element from $\mathbb{Z}_N^*$, hence $|\text{com}| = \text{poly}(\lambda)$. A local proof π_W for a subset $W \subseteq [k]$ consists of a single element from $\mathbb{Z}_N^*$, and hence $|\pi_W| = \text{poly}(\lambda)$.

7.4 NIZKs with Shorter Proofs and CRS

Combining the above HBG construction with Theorem 7, we obtain a NIZK proof scheme (i.e., with statistical soundness) where proofs and the CRS are of size $|C| \cdot \text{polylog}(|C|) + \text{poly}(\lambda, \log |C|)$.

Corollary 5. *Let $\mathcal{L}$ be an NP language with relation circuit C . Assuming that the RSA assumption holds, there exists a NIZK scheme for $\mathcal{L}$ with statistical soundness where proofs and the CRS are of size $|C| \cdot \text{polylog}(|C|) + \text{poly}(\lambda, \log |C|)$.*

Acknowledgements

Pedro Branco is supported by the Stellar Development Foundation. Nico Döttling is funded by the European Union (ERC, LACONIC, 101041207). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. Abhishek Jain is supported in part by NSF CAREER 1942789, JP Morgan Faculty Award and research gifts from Ethereum Foundation, Stellar Development Foundation, and Cisco. Akshayaram Srinivasan was supported in part by a NSERC Discovery Grant RGPIN-2024-03928, a Stellar Development Foundation Academic Research award, and an Amazon Research award. Brent Waters is supported by CNS-231870 and a Simons Investigator.

References

- [ABI⁺23] Benny Applebaum, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, Tianren Liu, and Vinod Vaikuntanathan. Succinct computational secret sharing. In Barna Saha and Rocco A. Servedio, editors, *55th ACM STOC*, pages 1553–1566. ACM Press, June 2023.
- [ACG⁺24] Abtin Afshar, Jiaqi Cheng, Rishab Goyal, Aayush Yadav, and Saikumar Yadugiri. Encrypted RAM delegation: Applications to rate-1 extractable arguments, homomorphic NIZKs, MPC, and more. Cryptology ePrint Archive, Report 2024/1806, 2024.
- [Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.
- [Bab85] László Babai. Trading group theory for randomness. In *17th ACM STOC*, pages 421–429. ACM Press, May 1985.
- [BCD⁺25] Pedro Branco, Arka Rai Choudhuri, Nico Döttling, Abhishek Jain, Giulio Malavolta, and Akshayaram Srinivasan. Black-box non-interactive zero knowledge from vector trapdoor hash. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 64–92. Springer, Cham, May 2025.

- [BDS25] Pedro Branco, Nico Döttling, and Akshayaram Srinivasan. Rate-1 statistical non-interactive zero-knowledge. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 514–547. Springer, Cham, August 2025.
- [BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.
- [BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 738–767. Springer, Cham, August 2020.
- [BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, Berlin, Heidelberg, August 2013.
- [BW10] Xavier Boyen and Brent Waters. Shrinking the keys of discrete-log-type lossy trapdoor functions. In Jianying Zhou and Moti Yung, editors, *ACNS 2010*, volume 6123 of *LNCS*, pages 35–52. Springer, Berlin, Heidelberg, June 2010.
- [CCH⁺19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23–26, 2019*, pages 1082–1090. ACM, 2019.
- [CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 91–122. Springer, Cham, April / May 2018.
- [CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, July 2004.
- [CGJ⁺23] Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. Correlation intractability and SNARGs from sub-exponential DDH. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 635–668. Springer, Cham, August 2023.
- [CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, Berlin, Heidelberg, May 2003.
- [CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 394–423, Virtual Event, August 2021. Springer, Cham.
- [CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd FOCS*, pages 68–79. IEEE Computer Society Press, February 2022.
- [CKU20] Geoffroy Couteau, Shuichi Katsumata, and Bogdan Ursu. Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 442–471. Springer, Cham, May 2020.
- [DJJ24] Quang Dao, Aayush Jain, and Zhengzhong Jin. Non-interactive zero-knowledge from LPN and MQ. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IX*, volume 14928 of *LNCS*, pages 321–360. Springer, Cham, August 2024.

- [DKL⁺23] Nico Döttling, Dimitris Kolonelos, Russell W. F. Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. Efficient laconic cryptography from learning with errors. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 417–446. Springer, Cham, April 2023.
- [DMP88] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In Carl Pomerance, editor, *CRYPTO’87*, volume 293 of *LNCS*, pages 52–72. Springer, Berlin, Heidelberg, August 1988.
- [FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990.
- [GGI⁺15] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam D. Smith. Using fully homomorphic hybrid encryption to minimize non-interactive zero-knowledge proofs. *Journal of Cryptology*, 28(4):820–843, October 2015.
- [GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.
- [GLWW24] Rachit Garg, George Lu, Brent Waters, and David J. Wu. Reducing the CRS size in registered ABE systems. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 143–177. Springer, Cham, August 2024.
- [GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.
- [GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Berlin, Heidelberg, August 2006.
- [GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Berlin, Heidelberg, May / June 2006.
- [Gro10] Jens Groth. Short non-interactive zero-knowledge proofs. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 341–358. Springer, Berlin, Heidelberg, December 2010.
- [HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. SNARGs for P from sub-exponential DDH and QR. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 520–549. Springer, Cham, May / June 2022.
- [HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th FOCS*, pages 850–858. IEEE Computer Society Press, October 2018.
- [HW09] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 654–670. Springer, 2009.
- [JJ21] Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 3–32. Springer, Cham, October 2021.

- [JNR26] Charanjit S. Jutla, Rohit Nema, and Arnab Roy. Partial fraction techniques for cryptography. Eurocrypt, 2026.
- [KNYY19] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 622–651. Springer, Cham, May 2019.
- [KNYY20] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Compact NIZKs from standard assumptions on bilinear maps. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 379–409. Springer, Cham, May 2020.
- [KP98] Joe Kilian and Erez Petrank. An efficient noninteractive zero-knowledge proof system for NP with general assumptions. *Journal of Cryptology*, 11(1):1–27, January 1998.
- [MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012.
- [NS98] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In Li Gong and Michael K. Reiter, editors, *ACM CCS 98*, pages 59–66. ACM Press, November 1998.
- [PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Cham, August 2019.
- [QRW19] Willy Quach, Ron D. Rothblum, and Daniel Wichs. Reusable designated-verifier NIZKs for all NP from CDH. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 593–621. Springer, Cham, May 2019.
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
- [RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
- [Ver] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
- [Wat24] Brent Waters. A new approach for non-interactive zero-knowledge from learning with errors. In Bojan Mohar, Igor Shinkar, and Ryan O’Donnell, editors, *56th ACM STOC*, pages 399–410. ACM Press, June 2024.
- [WW22] Brent Waters and David J. Wu. Batch arguments for NP and more from standard bilinear group assumptions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 433–463. Springer, Cham, August 2022.
- [WWW25] Brent Waters, Hoeteck Wee, and David J. Wu. New techniques for preimage sampling: Improved NIZKs and more from LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 3–33. Springer, Cham, May 2025.

A Generic hardness of the q -Computational Bilinear Square-Pole eXternal Diffie-Hellman assumption

In this section, we provide a generic hardness proof for the q -XSqPole assumption (Assumption 2).

Theorem 19. *Assumption 2 holds in the generic bilinear group model.*

Proof. In Assumption 2, the adversary $\mathcal{A}$ is given access to the following group elements in the source group $\mathbb{G}$:

$$g, g^s, \{g^{\frac{1}{x+i}}\}_{i \in [q]}, \{g^{\frac{s}{x+i}}\}_{i \in [q] \setminus \{i^*\}}.$$

Without loss of generality, we assume that the adversary $\mathcal{A}$ wins immediately if $-x \in \{1, \dots, q\}$, which happens with negligible probability. This ensures all the fractions are well-defined.

Let X be the formal variable corresponding to x , and let S be the formal variable corresponding to s . Every source group element that a generic adversary can compute must be of the form $g^{f(S,X)}$, where the function f is defined by

$$f(S, X) = a + bS + \sum_{i \in [q]} \frac{c_i}{X+i} + \sum_{i \in [q] \setminus \{i^*\}} \frac{d_i S}{X+i}$$

for coefficients a, b, c_i, d_i in $\mathbb{Z}_p$. Therefore, every target group element that the adversary can compute must be of the form $e(g, g)^{F(S,X)}$, where $F(S, X) = f_{a,b,c,d}(S, X) \cdot f'_{a',b',c',d'}(S, X)$ is a product of two functions f, f' of the form defined by two different source group queries. We expand the function $F(S, X)$ as

$$F(S, X) = H_0(X) + H_1(X) \cdot S + H_2(X) \cdot S^2$$

where $H_0(X), H_1(X), H_2(X)$ are rational functions of X , with

$$H_1(X) = \alpha + \sum_{i \in [q]} \frac{\beta_i}{X+i} + \sum_{i \in [q], j \in [q] \setminus \{i^*\}} \frac{\gamma_{i,j}}{(X+i)(X+j)}.$$

The description of H_0 and H_2 is irrelevant to the proof, and we omit it for simplicity.

We also write the target element

$$e(g, g)^{\frac{s}{(x+i^*)^2}} = e(g, g)^{\overline{H}_1(X) \cdot S}, \quad \text{where } \overline{H}_1(X) = \frac{1}{(X+i^*)^2}.$$

Now, fix any $x \in \mathbb{Z}_p$. Using the Schwartz-Zippel lemma over the variable S , we know that for $s \leftarrow \mathbb{Z}_p$,

$$\Pr [F(s, x) = \overline{H}_1(x) \cdot s \mid (H_0(x), H_1(x), H_2(x)) \neq (0, \overline{H}_1(x), 0)] \leq \frac{2}{p}.$$

Furthermore, we can argue that the probability of $H_1(x) = \overline{H}_1(x)$ is negligible. To see this, we first define the polynomial $P(X) = \prod_{i \in [q]} (X+i)^2$ of degree $2q$. It is easy to verify that $P(X) \cdot H_1(X)$ and $P(X) \cdot \overline{H}_1(X)$ are both polynomials of degree at most $2q$. Furthermore, it is easy to verify that $(X+i^*) \mid P(X) \cdot H_1(X)$, while $(X+i^*) \nmid P(X) \cdot \overline{H}_1(X)$. Therefore, $P(X) \cdot (H_1(X) - \overline{H}_1(X))$ is a non-zero polynomial of degree at most $2q$, and thus, for random $x \leftarrow \mathbb{Z}_p$,

$$\Pr [H_1(x) = \overline{H}_1(x)] \leq \Pr [P(x) \cdot (H_1(x) - \overline{H}_1(x)) = 0] \leq \frac{2q}{p}$$

again by the Schwartz-Zippel lemma. Therefore, we can conclude that for random $s, x \leftarrow \mathbb{Z}_p$,

$$\Pr [F(s, x) = \overline{H}_1(x) \cdot s] \leq \frac{2}{p} + \frac{2q}{p} = \frac{2(q+1)}{p},$$

which is negligible. Thus, the probability that $\mathcal{A}$ outputs the label of the target group element $e(g, g)^{\frac{s}{(x+i^*)^2}}$ is negligible, and the theorem follows. $\square$