Rate-1 Statistical Non-Interactive Zero-Knowledge

Pedro Branco[1], Nico Döttling[2], and Akshayaram Srinivasan[3]

[1]Bocconi University
[2]Helmholtz Center for Information Security (CISPA)
[3]University of Toronto

Abstract

We give the first construction of a rate-1 statistical non-interactive zero-knowledge argument of knowledge. For the $\mathsf{circuitSAT}$ language, our construction achieves a proof length of $|w| + |w|^{\epsilon} \cdot \mathsf{poly}(\lambda)$ where w denotes the witness, λ is the security parameter, ϵ is a constant less than 1, and $\mathsf{poly}(\cdot)$ is a fixed polynomial that is independent of the instance or the witness size. The soundness of our construction follows from the sub-exponential hardness of either the LWE assumption, or the $O(1)$-LIN assumption on prime-order groups with efficiently computable bilinear maps, or the DDH assumption. Previously, Gentry et al. (Journal of Cryptology, 2015) achieved NIZKs with statistical soundness and computational zero-knowledge with the aforementioned proof length by relying on (circular-secure) Learning with Errors assumption.

1 Introduction

In a zero-knowledge (ZK) [GMR85] proof system for an NP language $\mathcal{L}$, a prover convinces a verifier that a statement $x \in \mathcal{L}$ without revealing any other information. In a bit more detail, a ZK proof system should satisfy three properties, namely, completeness, soundness, and zero-knowledge. The *completeness* property requires that the honest prover convinces the verifier with a proof if $x \in \mathcal{L}$. The *soundness* guarantees that an (unbounded) malicious prover cannot make the honest verifier accept the proof except with negligible probability if $x \notin \mathcal{L}$. The *zero-knowledge* property ensures that a malicious polynomial-time verifier cannot learn any other information except that $x \in \mathcal{L}$ from its interaction with an honest prover. If the soundness of the proof system is required to hold only against computationally-bounded provers, then it is called an *argument system* [BCC88] and if the zero-knowledge property is required to hold against unbounded verifiers, then we have *statistical zero-knowledge*.

In the non-interactive setting (NIZK) [BFM88], the prover sends a single message (or proof) π to the verifier and the verifier then checks the correctness of this proof. This setting has attracted significant attention, not only due to the ubiquity of NIZKs as fundamental building blocks in cryptography [NY90, DDN91, DN00], but also because it is notoriously hard to construct such schemes. It is well-known that NIZKs are impossible to achieve without any trusted setup [BFM88, GO94]. Hence, to build NIZKs, it is usually assumed that there exists a common reference string (CRS) generated by a trusted party. Throughout this work, we will refer to NIZK for all NP in the CRS model as simply NIZK.

By now, NIZKs are known from a plethora of hardness assumptions including factoring-based assumptions [BFM88, FLS90, BDSMP91], pairing-based assumptions [CHK03, GOS12], LWE [CCH$^+$19, PS19, Wat24, BCD$^+$24, WWW24], sub-exponential DDH [JJ21], DDH and LPN [BKM20, BCD$^+$24] and LPN and MQ [DJJ24].

Statistical Zero-Knowledge. In this work, we focus on achieving statistical zero-knowledge in the non-interactive setting. That is, we require the zero-knowledge property to hold against unbounded verifiers but

the soundness is required to hold only against computationally bounded provers. This setting is of particular interest as it achieves everlasting security: even if one of the hardness assumptions used to build the scheme gets broken in the future, the secret witness of the prover remains hidden.

Minimizing Proof Length. An important measure of efficiency for a NIZK scheme is its proof length. Starting from the seminal works of Kilian [Kil92] and Micali [Mic94], the problem of minimizing the proof length has been extensively studied for nearly three decades. For the case of statistically-sound systems (a.k.a. proofs), under standard complexity-theoretic assumptions, one cannot achieve proof size that is lesser than the witness size [GH98, GVW02]. Moreover, the work of Campanelli *et al.* [CGKS23] showed that under the exponential-time hypothesis, for the case of arguments of knowledge[1], it is impossible to achieve proof size $o(|w|)$ with black-box extraction.[2] Thus, there is little difference between the lower bounds on the communication complexity for proofs and arguments of knowledge with respect to black-box extraction. Hence, we define the rate of a NIZK proof or argument of knowledge to be $|w|/|\pi|$ and call constructions with a proof size that asymptotically approaches $|w|$ to be rate-1.

Prior Work. Gentry et al. [GGI+15] gave an elegant construction of rate-1 NIZKs based on fully-homomorphic encryption (FHE) [Gen09, BV11]. Specifically, for the $\mathsf{circuitSAT}$ language, their proof size is $|w| + \mathsf{poly}(\lambda)$ where w is the witness, λ is the security parameter, and $\mathsf{poly}(\cdot)$ is a fixed polynomial that is independent of the instance or the witness size. Their construction achieves rate-1 as the ratio between the size of the witness and the size of the proof approaches 1 asymptotically. This construction achieves statistical soundness but unfortunately, does not achieve statistical zero-knowledge property.

Another interesting line of work by Katsumata et al. [KNYY19, KNYY20] constructed NIZKs for the $\mathsf{circuitSAT}$ language with a proof size of $|C| + \mathsf{poly}(\lambda)$ where C is the verification circuit in the $\mathsf{circuitSAT}$ instance. The security (both soundness and zero-knowledge) of their construction is computational and is based on standard assumptions on bilinear maps. This construction does not achieve rate-1 as the size of the verification circuit C could be polynomial in the size of the witness w.

As none of these works achieve statistical zero-knowledge, we ask:

Can we build rate-1 NIZKs with statistical zero-knowledge?

This was stated as an open problem in [PS19].

1.1 Our Result

In this work, we give a new approach to build a statistical NIZK for NP satisfying argument of knowledge with rate-1. The soundness of our construction follows from the sub-exponential hardness of either the $O(1)$-LIN assumption on prime-order groups with efficiently computable bilinear maps, or the DDH assumption, or the LWE assumption. For the $\mathsf{circuitSAT}$ language, our construction achieves a proof length of $|w| + |w|^{\epsilon} \cdot \mathsf{poly}(\lambda)$ where w is the witness, λ is the security parameter, a constant $0 < \epsilon < 1$, and $\mathsf{poly}(\cdot)$ is a fixed polynomial that is independent of the instance or the witness size.

Theorem 1 (Informal)**.** *Assuming the sub-exponential hardness of either LWE, or $O(1)$-LIN on prime order groups with efficiently computable bilinear maps, or the DDH, there exists a statistical rate-1 NIZK argument of knowledge.*

1.2 Concurrent Work

In a concurrent and independent work, Afshar *et al.* [ACG+24] considered the problem of constructing rate-1 NIZKs. They provided constructions of rate-1 NIZKs with computational soundness and computational

[1] Argument of knowledge guarantees that if a polynomial-time prover creates a valid proof for a statement x, then there exists an extractor that can extract a valid witness from this proof.

[2] In a nutshell, if that was the case, then we could use the extractor for the argument of knowledge to solve NP-complete problems better than brute-forcing over all witnesses. This is not possible under the exponential-time hypothesis.

zero-knowledge from either LWE or $O(1)$-linear on prime order groups or the sub-exponential DDH. They considered from a new primitive called *encrypted RAM* delegation scheme and used this to construct a rate-1 NIZK, fully homomorphic NIZKs, rate preserving compiler from semi-honest to malicious secure protocols, and composable zero-knowledge BARGs. When compared to their work, our construction achieves statistical zero-knowledge. As we explain later, much of the technical depth in our work is to achieve statistical zero-knowledge property.

2 Technical Overview

In this section, we give a high-level overview of the techniques used in constructing a rate-1 statistical NIZK argument of knowledge for NP. Throughout this section, we will be interested in constructing a NIZK for the $\mathsf{circuitSAT}$ language. Recall that an instance of this language comprises of a Boolean circuit C and an input x. A valid witness is a string w is such that $C(x, w) = 1$.

Computational Setting. We give a simple approach to construct a rate-1 NIZK argument of knowledge that achieves computational zero-knowledge (rather than statistical zero-knowledge). This construction makes use of a primitive called RAM delegation scheme [CJJ22, KLVW23, BD24] (or RAM-SNARG).

Before sketching this approach, we briefly recall the definition of a RAM delegation scheme. A RAM delegation allows a prover to convince a verifier that a RAM machine M starting at configuration cf (including a large external memory/database) transitions to configuration cf' in T steps where the verifier is only given digests h and h' of the two configurations cf and cf' respectively. The soundness guarantees that it is infeasible for a malicious (computationally bounded) prover to come up with an initial configuration cf, a digest $\widetilde{h}$ and a proof Π such that i) the proof Π is accepting for the digests $h, \widetilde{h}$ where h is the digest for the initial configuration cf, and (ii) $\widetilde{h}$ is not the digest of the (unique) configuration of M obtained after T steps starting with cf. For efficiency, we require the size of Π to be $\mathsf{poly}(\lambda, \log T)$ and is otherwise independent of the size of the configuration cf. We refer the reader to [CJJ22] for detailed definitions and constructions of this primitive.

To construct a rate-1 NIZK, the prover given the instance (C, x) and the witness w first computes a rate-1 commitment $\mathsf{com}(w)$ to the witness w. Crucially, this commitment should have a "short" (i.e., independent of $|w|$) opening γ_w. Such a commitment can be easily obtained by masking the witness with a PRF output, generating the commitment to the PRF key and setting γ_w to be the opening to the commitment of the PRF key. The prover then constructs a program P that uses $(C, x)\|\mathsf{com}(w)$ as the external memory, takes as input γ_w and computes $C(x, w)$. The prover then computes a RAM delegation proof Π for: *When P takes as input γ_w with memory $(C, x)\|\mathsf{com}(w)$, it outputs 1 after $O(|C|)$ steps.* Finally, the prover uses a (rate-0) NIZK argument of knowledge to prove that: *There is a γ_w and Π such that the RAM delegation verifier outputs 1 when run on (γ_w, Π) and the hash of the external memory $(C, x)\|\mathsf{com}(w)$.* Let π be the resulting (rate-0) NIZK proof. The prover sends $(\mathsf{com}(w), \pi)$. The verifier computes the digest of $(C, x)\|\mathsf{com}(w)$ and checks if the NIZK verifier accepts π.

It is easy to see that the resulting proof has rate-1 since the (i) size of the commitment is $|w| + \mathsf{poly}(\lambda)$, and (ii) the size of π only grows with the size of γ_w (which is much shorter than $|w|$) and the RAM delegation verifier's runtime (which is grows poly logarithmically in $|C|$). The zero-knowledge property also follows from the hiding of the commitment and the zero-knowledge property of the underlying NIZK.

To prove argument of knowledge, we can rely on the soundness property guaranteed by the RAM delegation in [CJJ22] and argument of knowledge of the underlying (rate-0) NIZK. The main idea is to use the knowledge extractor for the underlying NIZK system to extract Π and γ_w. Using γ_w, we can extract w from $\mathsf{com}(w)$. If w is not a valid witness, then we can build an adversary that breaks the soundness of the RAM delegation scheme by providing $(C, x)\|\mathsf{com}(w)$ as the external memory, γ_w as the short input and Π as the cheating proof that makes the verifier accept.

Challenges in the Statistical Setting. To make the above construction satisfy statistical zero-knowledge, we need to make the following changes: i) replace $\mathsf{com}(w)$ by a statistically hiding rate-1 commitment, ii) replace the rate-0 NIZK scheme by one that provides statistical zero-knowledge argument of knowledge. With these changes, the resultant construction can be proved to satisfy both statistical zero-knowledge and argument of knowledge properties. However, once we do these two changes, the rate of the construction becomes 0. Specifically, note that for a statistically hiding commitment, the opening needs to be at least as large as the message (by standard incompressibility argument). Thus, the size of γ_w is at least as large as the witness size. This means that, if we apply the previous strategy, the size of the proof π will be as large as $\mathsf{poly}(|w|, \lambda)$ and the resulting proof is not rate-1 anymore.

2.1 SNARG on Committed Database

To overcome the above challenge, we construct a special type of SNARG proof that works over committed database. In the standard setting of RAM SNARG, we are given a database in the clear and a program performs a specific computation by making random accesses to the database. The goal is to succinctly prove that a given output is obtained after running the program for T steps on the database. In our setting, the database is not provided in the clear but instead committed via a special commitment scheme. We want to succinctly prove correctness of a program that access the database in the *clear*. The proof generation procedure could take as input the openings to the commitment. However, we want the proof to satisfy statistical zero-knowledge property and the size of the proof to be poly logarithmic in the running time of the program.[3]

We now explain that this special SNARG is sufficient to construct a rate-1 statistical NIZK argument of knowledge. This construction assumes the existence of dual-mode rate-1 statistically hiding commitment scheme. The CRS for the dual-mode commitment can be generated in two computationally indistinguishable modes. In one of the modes, the commitment to any message is statistically hiding. In the other mode, the committed value can be extracted using a trapdoor. Such a dual-mode commitment scheme can be constructed from a rate-1 statistical sender private oblivious transfer [ADD+22].

The prover in this construction generates a commitment $\mathsf{com}(w)$ to the witness w using the rate-1 dual-mode commitment scheme. It then generates a special SNARG proof for the program that accesses the witness in the clear and runs the NP verification circuit. Let π be the proof. The prover sends $(\mathsf{com}(w), \pi)$. The verifier runs the SNARG verifier and checks if it accepts π.

This construction satisfies rate-1 as the size of the proof π is poly logarithmic in the size of the NP verification circuit and the commitment is rate-1. The statistical zero-knowledge property follows from the statistical hiding of the commitment and the statistical zero-knowledge of the proof π. To see why it satisfies argument of knowledge, we first switch the CRS of the dual-mode commitment to the extraction mode. This switch is not detected by a computationally bounded adversary. We then use the trapdoor to extract the witness w. If it is invalid, we can use π to break the soundness of the special SNARG proof system.

Construction of Special SNARG. We now explain how to construct a SNARG for committed database. To keep the proof size small, we require the dual-mode commitment to have local openings. This property allows the committer to generate a succinct proof that the committed bit at any location is a specific value. Later, in section 2.3, we explain how to construct such a dual-mode commitment scheme.

Our constructions builds on the construction of flexible SNARGs for RAM programs given in [KLVW23].

- The CRS comprises of a hash key for a collision-resistant hash function, a hash key for a somewhere extractable hash function, CRS for a rate-1 dual-mode commitment with local openings and a CRS for a somewhere extractable BARG scheme. At a high-level, BARGs allow the prover to convince a verifier that a set of k NP statements all belong to an NP language $\mathcal{L}$ with a proof. The size of this proof only grows poly logarithmically in the batch size. A BARG scheme is said to be somewhere extractable if

[3]Note that for a RAM program, the access pattern could provide non-trivial information. However, we are not concerned with concealing the access pattern and will only utilize it in scenarios where the access pattern is predetermined, such as that of Turing machines.

the setup procedure takes as input a location i and outputs the CRS along with a trapdoor. Given an accepting proof for the BARG, it is possible to extract the witness for the i-th statement from the proof using the trapdoor. Further, the CRS hides this index i. Recent constructions have shown how to achieve this primitive under LWE [CJJ22], bilinear maps [WW22], and sub-exponential DDH [CGJ+23].

- The prover uses the witness w to evaluate the circuit C on the input (x, w) to generate the computation tableau T. T comprises of the input x along with values carried by each internal wire in the circuit (we will deal with the witness separately).

- Let n be the number of gates in the circuit C. Let $T_1, \ldots, T_n$ be the snapshots of the computation tableau at the end of evaluating gates $1, 2, \ldots, n$ respectively. In a bit more detail, to compute T_i, we evaluate the first i gates in the circuit using the inputs (x, w). Let W_i be the sequence of $|C|$ wire values where the wires corresponding to the outputs of the first i gates are correctly set and the rest of the values set to 0. T_i is computed as $x \| W_i$. Note that $T_n = T$ and let T_0 be the initial computation tableau that is populated with input x and with 0s for each internal wire.

- The prover hashes each T_i for $i \in [n] \cup \{0\}$ using a Merkle tree to generate the hash digests $h_0, h_1, \ldots, h_n$. The prover then hashes these values using a somewhere extractable hash function to generate a hash digest h.

- The prover also generates a rate-1 dual-mode commitment to the witness w. Let us denote the commitment by $\mathsf{com}(w)$.

- The prover then considers a batch NP language consisting of n statements. The witness to the i-th instance comprises of openings to h_i, h_{i+1} w.r.t. to the hash digest h along with certain other openings ρ. If the i-th gate in the circuit reads particular bit(s) of the witness, then ρ includes those openings from $\mathsf{com}(w)$. Else, if it reads certain input wires or other internal wires, then the openings to those values w.r.t. to h_i are included in ρ. Finally, ρ also includes the opening to output wire of the i-th gate in h_i.

  The relation function checks if all the openings are correct and computes the output of the i-th gate using the values of the input wires. If it is the final output gate, it additionally checks if the computed output is 1. It then checks if the opening to the output wire value in h_i is the default value (which is 0). If it is the case, it checks if h_{i+1} is correctly obtained from h_i with the output wire value of the i-th gate correctly set to the above computed output. Note that this can be done for Merkle tree using the openings to the output wire of the i-th gate in h_i.

- It generates a batch proof π for the above batch language and sends $(\pi, \mathsf{com}(w), h)$ to the verifier along with local opening to h_0 w.r.t. to h. The verifier checks if the proof verifies, the local opening to h_0 is correct and if h_0 is the output of the Merkle tree on T_0 (which can be computed from the statement).

Proof Size. The size of $\mathsf{com}(w)$ is $|w| + \mathsf{poly}(\lambda)$ and the size of h is $\mathsf{poly}(\lambda)$. Note that size of each openings used in the witness of the BARG language is $\mathsf{poly}(\lambda)$ and therefore, the size of π is $\mathsf{poly}(\lambda)$. The size of the opening to h_0 is $\mathsf{poly}(\lambda)$ and hence, the overall proof size in our construction is $|w| + \mathsf{poly}(\lambda)$.

Argument of Knowledge. We give the description of a knowledge extractor. The extractor generates the CRS for the dual-mode commitment in the extractable mode. It first verifies if the proof is valid and if it the case, it uses the extractor for the dual-mode commitment to compute the candidate witness and outputs this. To prove that this extractor succeeds, we rely on techniques developed in the work of Kalai et al. [KLVW23] for constructing flexible SNARGs for RAM programs. Let us give more details.

Let $T_1, \ldots, T_n$ be the snapshots of the tableau generation procedure using the candidate witness extracted by the extractor. We maintain an invariant that if the hash key for the somewhere extractable hash function is extractable at location i and the prover produces an accepting proof, then h_i that is extracted from the

hash digest h is consistent with the tableau T_i (except with negligible probability). We prove this invariant using induction with the base case being $i = 0$.

The base case holds trivially (as the proof includes an opening to h_0) and suppose the invariant holds for the i-th step. Let us generate the hash key of the somewhere extractable hash function to be extractable at location $i + 1$ and let h_{i+1} be value that is extracted from h. We need to show that h_{i+1} is consistent with T_{i+1}. Below we provide a brief sketch of this argument.

- Since the invariant holds in the i-th step, it follows that if the hash key of the somewhere extractable hash function is set to be extractable at location i, then h_i extracted from h is consistent with T_i.
- We first change the CRS for the somewhere extractable BARG scheme to be binding at the i-th instance. It follows from the CRS indistinguishability property for the BARG scheme that h_i computed as above is still consistent with T_i.
- We now extract the witness for i-th BARG instance using the corresponding extractor. By somewhere extractability, this witness is a valid witness for the i-th instance except with negligible probability. Note that this witness corresponds to openings to h_i, h_{i+1} w.r.t. h along with certain other openings denoted by ρ.
- Since the somewhere extractable hash function is binding at location i and the witness is valid, it follows that h_i extracted from the digest h is identical to the h_i extracted from the witness of the BARG scheme except with negligible probability.
- We now rely on the collision-resistance of Merkle tree and the binding of the commitment to the witness w to show that values used in the computation of the i-th gate is consistent with T_i except with negligible probability. Since the extracted witness from the BARG is correct, it follows that the output of the i-th gate is computed correctly. Once again relying on the collision-resistance of Merkle tree, it follows that h_{i+1} that is extracted from the BARG scheme is consistent with T_{i+1}.
- Now, if we switch the hash key of the somewhere extractable hash function to be binding at location $i + 1$, we still maintain the invariant that h_{i+1} that is extracted from the BARG witness is consistent with T_{i+1} except with negligible probability. This property directly follows from the index hiding of the hash key.
- Once again relying on the binding property of somewhere extractable hash function at location $i + 1$, we infer that the invariant is maintained at step $i + 1$ as the BARG witness is valid.

This argument is reminiscent of the "two-key" trick [NY90] where we alternate between extracting the hash digests using somewhere extractable BARG and somewhere extractable hash. Extending this argument to the final step shows that h_n is consistent with T_n. We can now rely on the somewhere extractability of the BARG at the final instance along with the the collision-resistance of the Merkle tree to prove that the value carried by the output wire in T_n is 1. Thus, the witness extracted by us is valid.

Zero-Knowledge. An astute reader would have noticed that the above construction does not satisfy the zero-knowledge property. Specifically, the digest h could reveal information about some of the bits of some h_i which in turn leaks information about the tableau T_i. Since T_i consists of the internal wire values, we cannot afford to leak this information. Furthermore, the BARG proof π could leak information about the witness. We explain how to fix these leakages in the next subsection.

2.2 Achieving Statistical Zero-Knowledge

To fix the leakage from the BARG proof π, we use a zero-knowledge BARG. We observe that there is a very simple transformation from BARG to zero-knowledge BARG by additionally using NIZKs. Specifically, we consider a modified batch NP language where the instances are the same as before but the witnesses

correspond to NIZK proofs that attest that each instance belongs to the NP language. Since the new witnesses hide all information about the old witnesses, this transformation can be easily shown to satisfy zero-knowledge property. Additionally, the size of the new BARG proof is still a fixed polynomial in the size of the verification circuit of the underlying NP language. Furthermore, if the underlying NIZK satisfies statistical zero-knowledge property, then so does the BARG proof.

To prevent leakage of information from the hash digest h, we use a special SEH that we call *dual-mode SEH*. This SEH can be set in two different modes: a statistically hiding mode and a somewhere extractable mode. In the somewhere extractable mode, the hash key binds to a secret location i and an extractor can use the trapdoor to extract the input bit at location i from the digest. In the statistically hiding mode, the digest provides no information about the input. To construct such a SEH, we rely on the construction of SEH from the work of Kalai et al. [KLVW23] based on rate-1 OT. The hash key in this construction comprises of $\log k$ receiver OT messages where k is the length of the input. To hash an input, we first construct a complete binary tree where the leaves are assigned the values corresponding to the bits of the input and the value assigned to any node in level-ℓ is the sender OT message using the ℓ-th receiver message in the hash key and the sender strings being equal to the values of its children. The digest is the value assigned to the root. In the somewhere extractable mode, the choice bits in the hash key consists of the bits of the secret location i. To modify this construction to have a statistically hiding mode, we add an additional leaf with a default symbol and have the hash key point to this location. If the underlying OT has statistical sender privacy, then it follows that the input is statistically hidden in this mode.

2.3 Constructing Dual-Mode Rate-1 Commitment

The only missing piece in our work is the construction of dual-mode rate-1 commitment scheme. Recall that a dual-mode rate-1 commitment has a crs that can be generated in two different modes: a statistically hiding mode and an extractable mode. In the extractable mode, an extractor can use the trapdoor (sampled along with the CRS) to extract the committed message. In the statistically hiding mode, the committed message is statistically hidden. We additionally need this commitment to have a local opening property. In particular, we need the size of the opening to any location to be $|m|^{\epsilon} \cdot \mathsf{poly}(\lambda)$ (where m is the committed message) for a small enough constant $0 < \epsilon < 1$. This does not affect the rate of the NIZK construction as we can set ϵ to be an arbitrarily small constant and hence, the size of the BARG proof π is $|w|^{\epsilon'} \cdot \mathsf{poly}(\lambda)$ for an $\epsilon' < 1$.

To construct such a rate-1 commitment, we rely on a technique developed by Paneth and Pass [PP22] which uses on a rate-1 OT scheme. Specifically, we chop the message m into k blocks $m_1, \ldots, m_k$ for $k = |m|^{1-\epsilon}$. The commitment key in the extraction mode comprises of k receiver OT messages with choice bit being 0 in each of them. To commit to m, the committer computes k sender OT messages using $(m_1, \bot)$, $(m_2, \bot), \ldots, (m_k, \bot)$ as the sender messages. If the underlying OT has rate-1, then the size of the commitment is $|m| + |m|^{1-\epsilon} \cdot \mathsf{poly}(\lambda)$. Hence, the asymptotic rate achieved by this construction is 1. The local opening to some location i comprises $(m_{i^*}, \bot)$ along with the sender randomness used to generate the i^*-th OT message where i^* is the block that includes i. Thus, the size of the local opening is $|m|^{\epsilon} \cdot \mathsf{poly}(\lambda)$. Additionally, if the underlying OT scheme achieves statistical sender privacy, then we can set the commitment keys to be receiver OT messages with the choice bits set to 1 and we can show that the commitment statistically hides the message.

2.4 Need for Sub-Exponential Hardness

We haven't yet discussed why we need sub-exponential hardness for our assumptions. The reason is that we need the soundness error of (rate-0) NIZK used to upgrade a BARG to zero-knowledge BARG is sub-exponentially small in the security parameter.

Pass [Pas13] showed that it is impossible to construct a statistical NIZK with adaptive soundness based on any polynomial-time falsifiable assumption. One way to overcome this impossibility is to use complexity leveraging (as noted in [GOS12]) wherein we guess the statement to be proved by the adversary beforehand.

However, this requires the non-adaptive soundness error to be sub-exponentially small and requires the security parameter to scale on the instance size.

We need adaptive soundness for our underlying NIZK as the statement to be proved which comprises of the digest for the somewhere extractable hash and the commitment to the witness is generated adaptively by a cheating prover. The size of the hash digest is a fixed polynomial in the security parameter. However, the size of the commitment to the witness is at least as large as the witness itself and naïvely performing complexity leveraging will affect the rate of our construction. Fortunately, the size of the statement doesn't grow with the length of the entire commitment to the witness but instead only depends on at most two blocks of the witness (please refer to the above construction of the dual-mode rate-1 commitment scheme). This is because the statement only depends on the blocks in which the input wires to a specific gate lie. The size of each block is $|w|^{\epsilon} + \mathsf{poly}(\lambda)$. Crucially, we can set ϵ in this construction to be an arbitrary small constant. Therefore, to perform complexity leveraging, it is sufficient for us to guess this information. Hence, the security parameter for the NIZK can be set to at most $|w|^{\alpha\epsilon} \cdot \mathsf{poly}(\lambda)$ for some large constant α. The size of the NIZK verification circuit is $|w|^{\beta\epsilon} \cdot \mathsf{poly}(\lambda)$ and hence, the size of the BARG proof grows as $|w|^{\Delta\epsilon} \cdot \mathsf{poly}(\lambda)$ for some large constant Δ. We can set ϵ such that $\Delta \cdot \epsilon < 1$. This allows us to maintain the rate-1 feature of our construction.

3 Preliminaries

Let λ denote the cryptographic security parameter. We assume that all cryptographic algorithms implicitly take 1^{λ} as input.

3.1 Non-Interactive Zero-Knowledge

We recall the definition of non-interactive zero-knowledge (NIZK).

Definition 1 (Non-interactive zero-knowledge)**.** *A non-interactive zero-knowledge (NIZK) argument of knowledge (AoK) for a* NP *language* $\mathcal{L}$*, with corresponding instance-witness relation* $\mathcal{R} : \mathcal{X} \times \mathcal{W} \to \{0,1\}$*, is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *such that:*

- $\mathsf{Setup}(1^{\lambda}) \to \mathsf{crs}$: *It is a PPT algorithm that takes as input the unary encoding of the security parameter* 1^{λ} *and outputs a common reference string* crs*.*
- $\mathsf{P}(\mathsf{crs}, x, w) \to \pi$: *It 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*.*
- $\mathsf{V}(\mathsf{crs}, x, \pi) \to b$: *It 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-AoK 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:

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

- **(Non-adaptive) Argument of Knowledge.** There exists a PPT extractor $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$ such that for every non-uniform PPT prover P^* and all statements x, we have:

$$\Pr\left[\mathsf{V}(\mathsf{crs}, x, \pi) = 1 \wedge \mathcal{R}(x,w) = 0 \quad : \quad \begin{array}{l} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Ext}_1(1^{\lambda}), \\ \pi \leftarrow \mathsf{P}^*(\mathsf{crs}, x), \\ w \leftarrow \mathsf{Ext}_2(\mathsf{td}, x, \pi) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

- **Statistical Zero-knowledge.** There exists a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ such that for any computationally unbounded adversary $\mathcal{A}$, we have:

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

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

On Adaptive Soundness. The NIZK system is said to have adaptive argument of knowledge if the cheating prover P^* on input the CRS produces the statement x along with the prover π. Pass [Pas13] proved that it is impossible to construct a statistical NIZK that satisfies stronger form of adaptive soundness (where the adversary can choose the statement to prove after seeing the CRS) from any polynomial time falsifiable assumption via a black-box reduction. However, as noted in [GOS12], it is possible to achieve adaptive soundness by complexity leveraging. In order to use a rate-0 statistical NIZK in our construction, we need the non-adaptive soundness property to hold with sub-exponentially small error. This is achieved by relying on the sub-exponential hardness of the underlying assumptions used in proving non-adaptive soundness.

Theorem 2. *Assuming sub-exponential hardness of either $O(1)$-LIN assumption on prime-order groups with efficiently computable bilinear maps, or LWE, or DDH assumption, there exists a statistical NIZK with non-adaptive argument of knowledge where the soundness error is $2^{-\lambda^\epsilon}$ for some constant $\epsilon < 1$.*

3.2 Somewhere Extractable Hash Families

Definition 2 (Somewhere Extractable Hash)**.** *A somewhere extractable hash family consists of the following polynomial time algorithms:*

- $\mathsf{Gen}(1^\lambda, N, i^*) \to (\mathsf{hk}, \mathsf{td})$: *It is a probabilistic setup algorithm that takes as input the security parameter 1^λ, the message length N, and an index $i^* \in [N]$. It outputs a hashing key hk and a trapdoor td.*
- $\mathsf{Hash}(\mathsf{hk}, x) \to v$: *It is a deterministic algorithm that takes as input a hashing key hk and a message $x \in \{0,1\}^N$, and outputs a hash value $v \in \{0,1\}^{\ell_{hash}}$.*
- $\mathsf{Open}(\mathsf{hk}, x, j) \to (b, \rho)$: *It is a deterministic algorithm that takes as input a hashing key hk, a message $x \in \{0,1\}^N$ and an index $j \in [N]$. It outputs a bit $b \in \{0,1\}$ and an opening $\rho \in \{0,1\}^{\ell_{open}}$.*
- $\mathsf{Verify}(\mathsf{hk}, v, j, b, \rho) \to \{0,1\}$: *It is a deterministic algorithm that takes as input a hashing key hk, a hash value v, an index $j \in [N]$, a bit b and an opening ρ, and it outputs 1 (accept) or 0 (reject).*
- $\mathsf{Extract}(\mathsf{td}, v) \to u$: *It is a deterministic algorithm that takes as input the trapdoor td and a hash value v, and it outputs a bit $u \in \{0,1\}$.*

We want a somewhere extractable hash family to satisfy the following properties:

- ***Efficiency:*** *The size of the hashing key $|\mathsf{hk}|$, the size of the hash ℓ_{hash}, the size of the opening ℓ_{open} and the running time of Verify are all bounded by $\mathsf{poly}(\lambda, \log N)$.*
- ***Opening completeness:*** *For every $x \in \{0,1\}^N$, $j \in [N]$, we have*

$$\Pr\left[\begin{array}{l} b = x_j \\ \wedge\ \mathsf{Verify}(\mathsf{hk}, v, j, b, \rho) = 1 \end{array} : \begin{array}{l} (\mathsf{hk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, N, i^*), \\ v = \mathsf{Hash}(\mathsf{hk}, x), \\ (b, \rho) = \mathsf{Open}(\mathsf{hk}, x, j) \end{array}\right] = 1$$

- ***Index hiding:*** *For any $N = \mathsf{poly}(\lambda)$, $(i_0, i_1) \in [N]$, we have:*

$$\left\{\mathsf{hk} : (\mathsf{hk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, N, i_0)\right\} \approx_c \left\{\mathsf{hk} : (\mathsf{hk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, N, i_1)\right\}$$

- ***Collision-Resistance:*** *For any non-uniform PPT adversary $\mathcal{A}$, we have:*

$$\Pr\left[\begin{array}{l} x \neq x' \\ \wedge\ \mathsf{Hash}(\mathsf{hk}, x) = \mathsf{Hash}(\mathsf{hk}, x') \end{array} \ : \ \begin{array}{l} (\mathsf{hk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, N, i^*), \\ (x, x') \leftarrow \mathcal{A}(\mathsf{hk}) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

- ***Somewhere statistically (resp. computational) binding w.r.t. opening:*** *For any all-powerful (resp. non-uniform PPT) adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we have:*

$$\Pr\left[\begin{array}{l} u \neq b \wedge \mathsf{Verify}(\mathsf{hk}, v, i^*, b, \rho) = 1 \end{array} \ : \ \begin{array}{l} (1^N, i^*, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda), \\ (\mathsf{hk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^n, N, i^*), \\ (v, i^*, b, \rho) \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{hk}), \\ u \leftarrow \mathsf{Extract}(\mathsf{td}, v) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

Remark 1 ([DGKV22, KLVW23])**.** *Notice that we can easily convert any such somewhere extractable hash family into one that is extractable on m indices $I = \{i_1, \ldots, i_m\}$ by running each algorithm m times and concatenating the outputs. Under this transformation, the sizes of ℓ_{hash}, ℓ_{open} and the efficiency of the* Verify *will be $|I| \cdot \mathsf{poly}(\lambda, \log N)$. We will use the shorthand notation $\mathsf{Gen}(1^n, N, I)$ to denote this construction, in which case $\mathsf{Extract}(\mathsf{td}, v)$ will output m bits $(u_i)_{i \in I}$.*

Remark 2. *We note that collision-resistance is implied by index hiding and somewhere statistical/computational binding w.r.t. to opening.*

[KLVW23] showed how to construct a somewhere extractable hash family satisfying the above properties assuming a rate-1 string OT. This can be constructed from DCR [DJ01], from QR/DDH/LWE [DGI+19] or from $O(1)$-LIN assumption on prime-order groups with efficiently computable bilinear maps.

Theorem 3. *Assuming either QR or DDH or LWE or $O(1)$-LIN assumption on prime-order groups with efficiently computable bilinear maps, there exists a construction of somewhere extractable hash family.*

3.2.1 Dual-Mode Somewhere Extractable Hash

We also define a variant of SEH where the hashing key can be set in a *statistically hiding mode*, meaning that the input $x \in \{0,1\}^N$ is statistically hidden given $v \leftarrow \mathsf{Hash}(\mathsf{hk}, x)$.

Definition 3. *A dual-mode somewhere extractable hash family consists of the following polynomial time algorithms:*

- $\mathsf{Gen}(1^\lambda, N) \to \mathsf{hk}$: *It is a probabilistic setup algorithm that takes as input the security parameter 1^λ and the message length N. It outputs a hashing key* hk*.*
- $\mathsf{ExtGen}(1^\lambda, N, i^*) \to (\mathsf{hk}, \mathsf{td})$: *It is a probabilistic setup algorithm that takes as input the security parameter 1^λ, the message length N, and an index $i^* \in [N]$. It outputs a hashing key* hk *and a trapdoor* td*.*
- $\mathsf{Hash}(\mathsf{hk}, x)$, $\mathsf{Open}(\mathsf{hk}, x, j)$, $\mathsf{Verify}(\mathsf{hk}, v, j, b, \rho)$ *and* $\mathsf{Extract}(\mathsf{td}, v)$ *are defined as in Definition 2.*

A dual-mode SEH should fulfill all the properties described in Definition 2 (with respect to ExtGen*) and additionally it fulfills the following properties:*

- ***Mode indistinguishability.*** *For every $\lambda, N \in \mathbb{N}$ and any $i \in [N]$ we have:*

$$\left\{\mathsf{hk} \leftarrow \mathsf{Gen}(1^\lambda, N)\right\} \approx_c \left\{\mathsf{hk} : (\mathsf{hk}, \mathsf{td}) \leftarrow \mathsf{ExtGen}(1^\lambda, N, i)\right\}.$$

  Note that index hiding is implied by mode indistinguishability.

- ***Statistical hiding:*** *For any inputs $x, x' \in \{0,1\}^N$ we have:*

$$\left\{(\mathsf{hk}, v) : \begin{array}{c} \mathsf{hk} \leftarrow \mathsf{Gen}(1^n, N) \\ v = \mathsf{Hash}(\mathsf{hk}, x) \end{array}\right\} \approx_s \left\{(\mathsf{hk}, v) : \begin{array}{c} \mathsf{hk} \leftarrow \mathsf{Gen}(1^n, N) \\ v = \mathsf{Hash}(\mathsf{hk}, x') \end{array}\right\}.$$

Construction from Rate-1 SSP OT. We observe that the construction of SEH from rate-1 OT from [KLVW23] can be easily extended into a dual-mode SEH by using a (semi-honest) statistically sender secure rate-1 OT [DGI+19, ADD+22, BBDP22, BDS23]. At a high level, we will start by hashing $\mathsf{Hash}(x||r)$ where $x \in \{0,1\}^N$ is the input to be hashed and r is some uniformly chosen bit. If we use a statistically sender secure rate-1 OT in the SEH construction of [KLVW23], we will get a SEH for which all the hashed bits of the input are statistically hidden, except for the extractable one. Hence, in the statistically hiding mode, we just set the extractable index to be $N+1$, which corresponds to r. Thus, by the statistical sender security of the underlying rate-1 OT we will have that $\mathsf{Hash}(\mathsf{hk}, x||r) \approx_s \mathsf{Hash}(\mathsf{hk}, x'||r)$ for any $x, x' \in \{0,1\}^N$.

3.3 Somewhere Extractable Batch Arguments

We recall the notion of batch arguments (BARGs). This is an argument system to succinctly prove that multiple instances $x_1, \ldots, x_k$ belong to an NP language $\mathcal{L}$ via witnesses $w_1, \ldots, w_k$, with the communication complexity less than $\sum |w_i|$. At the moment, we know BARG constructions from several standard assumptions such as LWE [CJJ22], pairings [WW22], subexponential DDH [CGJ+23], or a mixture of assumptions [CJJ21, HJKS22].

In particular, let $\mathsf{BatchCSAT}$ be the following language:

$$\mathsf{BatchCSAT} = \{(C, x_1, \ldots, x_k) \ : \ \exists w_1, \ldots, w_k \text{ s.t. } \forall i \in [k],\, C(x_i, w_i) = 1\},$$

where $C_n : \{0,1\}^n \times \{0,1\}^{m(n)} \to \{0,1\}$ is a Boolean circuit of size $s(n)$ that checks a relation with instance size n and witness size $m(n)$. For simplicity, we use s and m instead of $s(n)$ and $m(n)$.

Definition 4 (Somewhere extractable BARGs)**.** *A somewhere extractable batch argument for* $\mathsf{BatchCSAT}$ *with L-succinctness consists of the following polynomial time algorithms:*

- $\mathsf{Gen}(1^\lambda, 1^k, 1^s, i^*) \to (\mathsf{crs}, \mathsf{td})$ *: It is a PPT algorithm that takes as input the unary encoding of the security parameter 1^λ, the number of instances 1^k, the circuit size 1^s and an index $i^* \in [k]$. It outputs a common reference string* crs *and a trapdoor* td*.*
- $\mathsf{P}(\mathsf{crs}, C, \{x_i\}_{i\in[k]}, \{w_i\}_{i\in[k]}) \to \pi$*: It is a PPT algorithm that takes* crs*, a circuit C, k statements $x_1, \ldots, x_k \in \{0,1\}^n$ and k witnesses $w_1, \ldots, w_k \in \{0,1\}^m$. It outputs a proof π.*
- $\mathsf{V}(\mathsf{crs}, C, \{x_i\}_{i\in[k]}, \pi) \to b$*: It is a deterministic algorithm that takes* crs*, a circuit C, k statements $\{x_i\}_{i\in[k]}$ and a proof π. It outputs a bit $b \in \{0,1\}$.*
- $\mathsf{Extract}(\mathsf{td}, C, \{x_i\}_{i\in[k]}, \pi) \to w^*$*: It is a deterministic algorithm that takes a trapdoor* td*, a circuit C, k statements $\{x_i\}_{i\in[k]}$ and a proof π. It outputs a witness w^* for instance x_{i^*}.*

We want a somewhere extractable batch argument to satisfy the following properties:

- ***L-succinctness:*** *The* crs *and the proof π have length at most $L(k,\lambda) \cdot \mathsf{poly}(s)$, and the verifier runs in time $L(k,\lambda) \cdot \mathsf{poly}(s) + k \cdot \mathsf{poly}(n, \lambda)$.*
- ***Completeness:*** *For all $\lambda \in \mathbb{N}$, all $k, n \in \mathsf{poly}(\lambda)$, all circuits $C : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ at size most s and all $(x_1, \ldots, x_k)$ and $(w_1, \ldots, w_k)$ such that $C(x_i, w_i) = 1$ and any $i^* \in [k]$, we have:*

$$\Pr\left[1 \leftarrow \mathsf{V}(\mathsf{crs}, C, \{x_i\}_{i\in[k]}, \pi) : \begin{array}{c} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, 1^k, 1^s, i^*) \\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, C, \{x_i\}_{i\in[k]}, \{w_i\}_{i\in[k]}) \end{array}\right] = 1.$$

- ***Index hiding:*** *For all* $k, s = \mathsf{poly}(\lambda)$*,* $i_0, i_1 \in [k]$

$$\left\{\mathsf{crs} : (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, 1^k, 1^s, i_0)\right\} \approx_c \left\{\mathsf{crs} : (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, 1^k, 1^s, i_1)\right\}$$

- ***Somewhere argument of knowledge:*** *There exists a PPT extractor* Ext *such that for any PPT adversary* $\mathcal{A}$ *and for any* $k, s = \mathsf{poly}(\lambda)$*, and any index* $i^* \in [k]$ *we have that*

$$\Pr\left[\begin{array}{ccc} 1 \leftarrow \mathsf{V}(\mathsf{crs}, C, \{x_i\}_{i\in[k]}, \pi) & & (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda, 1^k, 1^s, i^*) \\ \wedge & : & (C, \{x_i\}_{i\in[k]}, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \\ C(x_{i^*}, w^*) \neq 1 & & w^* \leftarrow \mathsf{Ext}(\mathsf{td}, C, \{x_i\}_{i\in[k]}, \pi) \end{array}\right] \leq \mathsf{negl}.$$

Index BARGs. We say that a somewhere extractable BARG scheme is an index seBARG if the instances $x_1, \ldots, x_k$ are all of the form $x_i = i$; however, in the L-succinctness property we require that the verification algorithm runs in time $L(k, \lambda) \cdot \mathsf{poly}(s)$, since it doesn't have to read all the instances anymore.

Theorem 4 ([CJJ22, DGKV22, PP22, WW22, CGJ+23, KLVW23])**.** *Assuming either LWE or sub-exponential DDH or* $O(1)$*-*LIN *assumption on prime-order groups with efficiently computable bilinear maps, there exists a construction of somewhere extractable index BARGs with* $\mathsf{poly}(\lambda, \log k)$*-succinctness.*

4 Rate-1 Dual-Mode Fully Local Extractable Commitments

In this section we present a new commitment scheme with some special properties: i) It is dual-mode, meaning that we can set the CRS to be in either extractable or statistically hiding mode, ii) It is local openable, meaning that opening a bit of the message should only require a opening of sublinear size, and iii) It is rate 1, meaning that the size of the commitment is identical to the size of the message (up to sublinear additive factors).

Definition 5 (Dual-Mode Fully-local extractable commitment)**.** *A dual-mode fully-local extractable commitment (FLC) scheme parametrized by* $k, \ell \in \mathbb{N}$ *consists of the following polynomial time algorithms:*

- $\mathsf{Setup}(1^\lambda, k, \ell) \to \mathsf{crs}$ *: It is a PPT algorithm that takes in the unary encoding of the security parameter* 1^λ *and output a common reference string* crs*.*
- $\mathsf{ExtSetup}(1^\lambda, k, \ell) \to (\mathsf{crs}, \mathsf{td})$ *: It is a PPT algorithm that takes in the unary encoding of the security parameter* 1^λ *and output a common reference string* crs *and a trapdoor* td*.*
- $\mathsf{Com}(\mathsf{crs}, \mathbf{m}) \to (\mathsf{com}, \mathsf{st})$ *: It is a PPT algorithm that takes in* crs *and the message* $\mathbf{m} \in \{0,1\}^k$ *and outputs a commitment string* com *(which can be decomposed as* $\mathsf{com} = (\mathbf{d}_1, \ldots, \mathbf{d}_\ell)$*) and a state* st*.*
- $\mathsf{LocOpen}(\mathsf{crs}, \mathsf{com}, \mathsf{st}, m, i)$ *: It is a deterministic algorithm that takes in* crs*, a commitment* com*, a state* st*, a message* $m = (m_1, \ldots, m_k)$ *and an index* $i \in [k]$*, and outputs a opening* ρ*.*
- $\mathsf{LocVer}(\mathsf{crs}, \mathbf{d}_j, m_i, \rho) \to b$ *: It is a deterministic algorithm that takes in a* crs*, a commitment* $(c, \mathbf{d}_j)$ *where* $j \in [\ell]$*, a message* m_i *and the opening* ρ *and outputs a bit* $b \in \{0,1\}$*.*
- $\mathsf{Extract}(\mathsf{td}, \mathsf{com}) \to m$ *: It is a deterministic algorithm that takes in the trapdoor* td *and a commitment string* com *and outputs a message* m*.*

We require a dual-mode fully-local extractable commitment to satisfy the following properties:

- ***Mode indistinguishability:*** *For every* $\lambda, k, \ell \in \mathbb{N}$ *we have:*

$$\left\{\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, k, \ell)\right\} \approx_c \left\{\mathsf{crs} : (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda, k, \ell)\right\}$$

- ***Local Opening Completeness:*** *For every message $m = (m_1, \ldots, m_k)$ and any indices $j \in [\ell], i \in [k]$ such that $j \in \{(i-1)k/\ell, \ldots, ik/\ell\}$, we have:*

$$\Pr\left[\mathsf{LocVer}(\mathsf{crs}, \mathbf{d}_j, m_i, \rho) = 1 - \mathsf{negl}(\lambda) \quad : \quad \begin{array}{l} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda), \\ (\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{crs}, m) \\ \rho \leftarrow \mathsf{LocOpen}(\mathsf{crs}, \mathsf{com}, \mathsf{st}, m, i) \end{array} \right] = 1$$

  where $\mathsf{com} = (\mathbf{d}_1, \ldots, \mathbf{d}_\ell)$.

- ***Statistical Hiding:*** *For any two messages m_0, m_1 of equal length that is $\mathsf{poly}(\lambda)$, we have:*

$$\begin{aligned} &\left\{(\mathsf{crs}, \mathsf{com}) : (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\lambda), \mathsf{com} \leftarrow \mathsf{Com}(\mathsf{crs}, m_0)\right\} \approx_s \\ &\quad \left\{(\mathsf{crs}, \mathsf{com}) : (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\lambda), \mathsf{com} \leftarrow \mathsf{Com}(\mathsf{crs}, m_1)\right\} \end{aligned}$$

- ***Extraction Correctness:*** *For any unbounded adversary $\mathcal{A}$, we have:*

$$\Pr\left[\wedge_i \mathsf{LocVer}(\mathsf{crs}, \mathbf{d}_j, m_i, \rho_i) = 1 \wedge \mathbf{m} \neq \mathbf{m}' \quad : \quad \begin{array}{l} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda), \\ (\mathsf{com}, \mathbf{m}, \{\rho_i\}_i) \leftarrow \mathcal{A}(\mathsf{crs}), \\ \mathbf{m}' \leftarrow \mathsf{Extract}(\mathsf{td}, \mathsf{com}) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

  where $\mathsf{com} = (\mathbf{d}_1, \ldots, \mathbf{d}_\ell)$ *and* $\mathbf{m} = (m_1, \ldots, m_k)$.

- ***Efficiency:*** *The rate-1 property states that $|\mathsf{com}| = k + o(k) \cdot \mathsf{poly}(\lambda)$. Additionally, we require that* LocVer *runs in time $o(k) \cdot \mathsf{poly}(\lambda)$, which also implies that $|\rho| = o(k) \cdot \mathsf{poly}(\lambda)$.*

Remark 3 (Simple construction achieving computational hiding)**.** *If we drop the statistically hiding property, then this notion can be easily achieved with an extractable commitment and a PRF. To commit to the message, simply compute a commitment to the PRF key K and compute $m' = m \oplus (\mathsf{PRF}(K, 1), \ldots, \mathsf{PRF}(K, k))$. It is easy to see that this construction is extractable and local openable. This construction only achieves computational hiding, as the PRF only provides computational pseudorandomness.*

4.1 Dual-Mode Fully Local Extractable Commitment from Rate-1 OT

We present a generic construction from rate-1 oblivious transfer with (semi-honest) statistical sender security [DGI+19, ADD+22, BBDP22, BDS23]. At the moment, we know how to build this primitive from hardness assumptions such as DCR [DJ01], LWE [BDGM19], DDH, QR [DGI+19], k-LIN [KLVW23] or a combination of DDH and LPN [BBDP22, BDS23].

Definition 6 (Rate-1 oblivious transfer)**.** *A rate-1 oblivious transfer (OT) consists of the following algorithms:*

- $\mathsf{OTR}(1^\lambda, b) \to (\mathsf{st}, \mathsf{ot}_1)$: *It is a PPT algorithm that takes in the unary encoding of the security parameter 1^λ and a bit $b \in \{0, 1\}$, and outputs a private state* st *and an OT receiver message* ot_1.

- $\mathsf{OTS}(\mathsf{ot}_1, (m_0, m_1); r) \to \mathsf{ot}_2$: *It is a PPT algorithm that takes in an OT message* ot_1 *and a pair of messages (m_0, m_1) (using random coins $r \in \lambda$), and outputs an OT sender message* ot_2

- $\mathsf{OTD}(\mathsf{st}, \mathsf{ot}_2) \to m$: *It is a deterministic algorithm that takes in a private state* st *and an OT sender message* ot_2*, and outputs a message m.*

- ***Correctness:*** *For every $\lambda \in \mathbb{N}$ $b \in \{0, 1\}$ and every pair of messages (m_0, m_1) we have:*

$$\Pr\left[\mathsf{OTD}(\mathsf{st}, \mathsf{ot}_2) = m_b : \begin{array}{c} (\mathsf{st}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, b) \\ \mathsf{ot}_2 \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (m_0, m_1)) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

- ***Receiver security:*** *For every $\lambda \in \mathbb{N}$ we have that:*

$$\left\{\mathsf{ot}_1 : (\mathsf{st}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, 0)\right\} \approx_c \left\{\mathsf{ot}_1 : (\mathsf{st}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, 1)\right\}$$

- ***Semi-honest statistical sender security:*** *For every $\lambda \in \mathbb{N}$, any $b \in \{0,1\}$ and any pair of messages (m_0, m_1) we have:*

$$\left\{(\mathsf{ot}_2, \mathsf{st}) : \begin{array}{c}(\mathsf{st}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, b) \\ \mathsf{ot}_2 \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (m_0, m_1))\end{array}\right\} \approx_s \left\{(\mathsf{ot}_2, \mathsf{st}) : \begin{array}{c}(\mathsf{st}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, b) \\ \mathsf{ot}_2 \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (m_b, m_b))\end{array}\right\}.$$

- ***Efficiency:*** *We require that $|\mathsf{ot}_2| = |m_b| + \mathsf{poly}(\lambda)$.*

We need an additional property for the rate-1 OT called verifiable correctness, which was defined in [KLVW23].

Definition 7. *An OT scheme $(\mathsf{OTR}, \mathsf{OTS}, \mathsf{OTD})$ is said to be verifiable correct if there is polynomial-time deterministic algorithm $\mathsf{Valid}(\mathsf{ot}_1, m_0, m_1, r) \to \{0,1\}$ such that:*

- *If $1 = \mathsf{Valid}(\mathsf{ot}_1, m_0, m_1, r)$ then*

$$\Pr\left[\mathsf{OTD}(\mathsf{st}, \mathsf{ot}_2) = m_b : \begin{array}{c}(\mathsf{st}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, b) \\ \mathsf{ot}_2 \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (m_0, m_1); r)\end{array}\right] = 1.$$

- *For all b, m_0, m_1, r,*

$$\Pr\left[1 = \mathsf{Valid}(\mathsf{ot}_1, m_0, m_1, r) : (\mathsf{st}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, b)\right] = 1 - \mathsf{negl}(\lambda).$$

Rate-1 OT schemes with (semi-honest) statistical sender security can be built from DDH, QR or LWE [DGI+19, BBDP22]. We now sketch a construction of a (semi-honest) statistical sender secure rate-1 OT from $\mathcal{O}(1)-$LIN assumption. This follows from a simple modification of the construction given in [KLVW23]. Like in [KLVW23], we give a construction of a rate-1 batch OT scheme which implies a rate-1 string OT.

Rate-1 OT with semi-honest statistical sender security from $\mathcal{O}(1)-$LIN. We show how to modify the rate-1 OT from $\mathcal{O}(1)-$LIN assumption given in [KLVW23] to provide semi-honest statistical sender security.

- **Receiver's message**: Let $\mathbf{b} \in \{0,1\}^{n-k}$ be the vector of receiver's choice bits. Pad these bits with default bits to obtain a vector of length n. We will slightly abuse the notation and use $\mathbf{b}$ to denote this padded vector. Let $\mathbf{X} \in \{0,1\}^{n \times n}$ be a matrix whose diagonal elements are given by $\mathbf{b}$. Let $\mathbf{Y} \in \{0,1\}^{(k+n)\times n}$ be another matrix that as $\mathbf{X}$ in the bottom block and zeroes elsewhere. The receiver chooses a random matrix $\mathbf{B} \in \mathbb{Z}_p^{(k+n)\times n}$ with rank k. This is done by choosing the first k rows of $\mathbf{B}$ uniformly at random and then choosing the last n rows to be random linear combinations of the first k rows. Let the $(k+i)$-th row be given by $\mathbf{v}_i \cdot \mathbf{B}[1, \ldots, k]$ where $\mathbf{B}[1, \ldots, k]$ denotes the first k rows of $\mathbf{B}$. The receiver's message is given by $g^{(\mathbf{B}+\mathbf{Y})}$. Note that the k-LIN assumption implies that $g^{\mathbf{B}}$ chosen as above is indistinguishable from $g^{\mathbf{U}}$ where $\mathbf{U}$ is uniformly chosen from $\mathbb{Z}_p^{(k+n)\times n}$. This ensures that the receiver's choice bits are computationally hidden.

- **Sender's message.** Let $\mathbf{s}_0, \mathbf{s}_1 \in \{0,1\}^{n-k}$ be the vector of sender's inputs in the batch OT. The sender samples $2k$ random elements from $\mathbb{Z}_p$ and extends $\mathbf{s}_0, \mathbf{s}_1$ with k elements each to obtain new vectors in $\mathbb{Z}_p^n$. We slightly abuse the notation and denote the new vectors as $\mathbf{s}_0, \mathbf{s}_1$ respectively. The sender computes

$$\mathbf{h} = g^{(\mathbf{B}+\mathbf{Y})\cdot(\mathbf{s}_1 - \mathbf{s}_0)} \odot g^{0^k \| \mathbf{s}_0}.$$

  where $\odot$ denotes point-wise product. It then parses $\mathbf{h} = (\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3) \in \mathbb{G}^k \times \mathbb{G}^{n-k} \times \mathbb{G}^k$. For each $i \in [n-k]$, it computes $z_i = \mathsf{DDLog}(h_{2,i})$ where DDLog is the Distributed Discrete Log introduced in [BGI16]. It sends $(\mathbf{h}_1, (z_1, \ldots, z_{n-k}))$.

- **Output Computation.** Let $\mathbf{u}_i = (-\mathbf{v}_i, \mathbf{e}_i)$ be the vector that is in the left kernel of $\mathbf{B}$ for each $i \in [n]$ (where $\mathbf{e}_i$ is the i-th unit vector of length n). This implies that for each $i \in [n-k]$,

$$((-\mathbf{v}_i) \diamond \mathbf{h}_1) \cdot h_{2,i} = g^{\mathbf{u}_i\left((\mathbf{B}+\mathbf{Y})\cdot(\mathbf{s}_1-\mathbf{s}_0)+0^k\|\mathbf{s}_0\right)} = g^{s_{b_i,i}}$$

  where $(-\mathbf{v}_i) \diamond \mathbf{h}_1$ denotes applying the linear map $-\mathbf{v}_i$ to the exponent of the group elements $\mathbf{h}_1$. This means that with probability at least $1 - 1/n\lambda$, for each $i \in [n-k]$, $\mathsf{DDLog}((-\mathbf{v}_i) \diamond \mathbf{h}_1)) \oplus z_i = s_{b_i,i}$. Therefore, by union bound, the probability of correctness of the complete construction is $1 - 1/\lambda$. This can be bootstrapped to $1 - \mathsf{negl}(\lambda)$ correctness via the same techniques in [DGI+19].

- **Statistical Sender Privacy.** Note that except with negligible probability, $z_i = \mathsf{DDLog}((-\mathbf{v}_i) \diamond \mathbf{h}_1)) \oplus s_{b_i,i}$. This can be simulated given the knowledge of the output and $\mathbf{h}_1$. Furthermore, since the last k elements of $\mathbf{s}_1 - \mathbf{s}_0$ are uniformly chosen from $\mathbb{Z}_p^k$, it follows that $\mathbf{B}[1, \ldots, k] \cdot (\mathbf{s}_1 - \mathbf{s}_0)$ is uniformly distributed with overwhelming probability. This means that $\mathbf{h}_1$ is uniformly distributed and this completes the statistical sender security proof.

Construction. We give a formal description of a rate-1 FLC from rate-1 OT. Let $(\mathsf{OTR}, \mathsf{OTS}, \mathsf{OTD})$ be a rate-1 OT which is verifiable correct.

- $\mathsf{Setup}(1^\lambda)$:
  1. Run $(\mathsf{st}_{OT}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, 1)$.
  2. Sample $\mathbf{u} \leftarrow\!\$\ \{0,1\}^t$.
  3. Output $\mathsf{crs} = (\mathsf{ot}_1, \mathbf{u})$.
- $\mathsf{ExtSetup}(1^\lambda)$:
  1. Run $(\mathsf{st}_{OT}, \mathsf{ot}_1) \leftarrow \mathsf{OTR}(1^\lambda, 0)$.
  2. Sample $\mathbf{u} \leftarrow\!\$\ \{0,1\}^t$.
  3. Output $\mathsf{crs} = (\mathsf{ot}_1, \mathbf{u})$ and $\mathsf{td} = \mathsf{st}_{OT}$.
- $\mathsf{Com}(\mathsf{crs}, \mathbf{m})$:
  1. Parse $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_\ell)$ where each $\mathbf{m}_i \in \{0,1\}^t$ and $t = k/\ell$.
  2. For all $i \in [\ell]$ compute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (\mathbf{m}_i, \mathbf{u}); r_i)$ using random coins $r_i \leftarrow\!\$\ \{0,1\}^\lambda$.
  3. Output $\mathsf{com} = \{\mathsf{ot}_{2,i}\}_{i\in[\ell]}$ and $\mathsf{st} = \{r_i\}_{i\in[\ell]}$.
- $\mathsf{LocOpen}(\mathsf{crs}, \mathsf{com}, \mathsf{st}, \mathbf{m}, i \in [k])$:
  1. Parse $\mathsf{com} = \{\mathsf{ot}_{2,i}\}_{i\in[\ell]}$, $\mathsf{st} = \{r_i\}_{i\in[\ell]}$, and $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_j)$
  2. Let $j \in [\ell]$ such that $i \in \{(j-1)k/\ell, jk/\ell\}$.
  3. Output $\rho = (r_j, \mathbf{m}_j)$.
- $\mathsf{LocVer}(\mathbf{d}_j, m_i, \rho)$:
  1. Parse $\mathbf{d}_j = \mathsf{ot}_{2,j}$, $\rho = (r_j, \mathbf{m}_j)$ and $\mathbf{m}_j = (m_{j,1}, \ldots, m_{j,k/\ell})$.
  2. If $\mathsf{ot}_{2,j} \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (\mathbf{m}_j, \mathbf{u}); r_j)$, if $m_{j,i \mod k/\ell} = m_i$ and if $1 = \mathsf{Valid}(\mathsf{ot}_1, \mathbf{m}_j, \mathbf{u}, r_j)$, output 1.
- $\mathsf{Extract}(\mathsf{td}, \mathsf{com})$:
  1. Parse $\mathsf{td} = \mathsf{st}_{OT}$ and $\mathsf{com} = \{\mathsf{ot}_{2,i}\}_{i\in[\ell]}$.
  2. For all $i \in [\ell]$ compute $\mathbf{m}_i \leftarrow \mathsf{OTD}(\mathsf{st}_{OT}, \mathsf{ot}_{2,i})$.
  3. Output $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_\ell)$

Efficiency. Set $\ell = k^{1-\epsilon}$ for $\epsilon > 0$. The commitment com is composed by $\{\mathsf{ot}_{2,i}\}_{i\in[\ell]}$ where

- $|\mathsf{ot}_{2,i}| = k/\ell + \mathsf{poly}(\lambda) = k^{\epsilon} + \mathsf{poly}(\lambda)$.

Hence the rate ϱ_{com} is

$$\varrho_{\mathsf{com}} = \frac{|\mathsf{com}|}{|\mathbf{m}|} = \frac{k^{1-\epsilon}\cdot|\mathsf{ot}_{2,i}|}{k} = 1 + \frac{\mathsf{poly}(\lambda)}{k^{\epsilon}}.$$

On the other hand, the size of the opening ρ for a position $i \in [k]$ is

$$|\rho| = |r_j| + \mathbf{m}_j = k^{\epsilon} + \mathsf{poly}(\lambda)$$

which is sublinear in the size of the message k. Finally, the algorithm LocVer runs in time $P(k^{\epsilon}, \lambda)$ for a fixed polynomial P. For a proper choice of ϵ we have that LocVer runs in time $o(k)\cdot\mathsf{poly}(\lambda)$.

Remark 4. *Note that the choice of ϵ can be tuned at setup time. In particular, the parameter $\epsilon > 0$ (which determines the size of the opening ρ and the runtime of LocVer) can be as small as we want. This will be important to set up the parameters that allow us to achieve rate-1 NIZKs with statistical zero-knowledge (see Section 5).*

Analysis. We now proceed with the analysis of the scheme. Local opening completeness follows easily from the verifiable correctness of the underlying OT scheme. We proceed to show all other security properties of the scheme.

Lemma 1. *The construction presented above is dual-mode indistinguishable given that the underlying rate-1 OT is receiver secure.*

Proof. Assume that there is an adversary $\mathcal{A}$ that breaks the mode indistinguishability of the commitment scheme. This adversary can be directly used as an adversary against the receiver security of the underlying OT. □

Lemma 2. *The construction presented above has extraction correctness given that the underlying rate-1 OT is verifiable correct.*

Proof. Let $(\mathsf{com}, \mathbf{m}, \{\rho_i\})$ be the output of the adversary in the extraction correctness game. If $\mathsf{LocVer}(\mathsf{crs}, \mathbf{d}_j, m_i, \rho_i) = 1$ for all $i \in [k]$ then this means that $\mathsf{ot}_{2,j} \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (\mathbf{m}_j, \mathbf{u}); r_j)$, $m_{j,i \mod k/\ell} = m_i$ and $1 = \mathsf{Valid}(\mathsf{ot}_1, \mathbf{m}_j, \mathbf{u}, r_j)$. By the verifiable correctness of the underlying OT scheme, this immediately implies that for all $j \in [\ell]$ compute $\mathbf{m}_j \leftarrow \mathsf{OTD}(\mathsf{st}_{OT}, \mathsf{ot}_{2,j})$. □

Lemma 3. *The construction presented above is statistical hiding given that the underlying rate-1 OT is semi-honest statistical sender secure.*

Proof. In the statistically hiding mode, the crs is composed by ot_1 which is the output of $\mathsf{OTR}(1^{\lambda}, 1)$ and the commitment is of the form $\mathsf{ot}_{2,i} \leftarrow \mathsf{OTS}(\mathsf{ot}_1, (\mathbf{m}_i, \mathbf{u}); r_i)$. Thus, breaking statistically hiding of the commitment is equivalent to breaking statistical sender security of the underlying OT. □

5 Rate-1 Statistical Non-Interactive Zero-Knowledge

In this section we show how to build a rate-1 NIZK for which the zero-knowledge property holds statistically.

Building Blocks. The construction uses the following building blocks:

- A rate-1 dual-mode fully local extractable commitment scheme $(\mathsf{Setup}_C, \mathsf{ExtSetup}_C, \mathsf{Com}, \mathsf{LocOpen}, \mathsf{LocVer}, \mathsf{Extract}_C)$ (see Definition 5), for which the local openings have sublinear size $m^{\epsilon} \cdot \mathsf{poly}(\lambda)$ where ϵ is set later (see Remark 4).
- A dual-mode somewhere extractable hash family $(\mathsf{Gen}_H, \mathsf{ExtGen}_H, \mathsf{Hash}_H, \mathsf{Open}_H, \mathsf{Verify}_H, \mathsf{Extract}_H)$ with statistical hiding (Definition 5).
- A somewhere extractable batch argument $(\mathsf{Gen}_B, \mathsf{P}_B, \mathsf{V}_B, \mathsf{Extract}_B)$ for the language defined below. Let s be the size of the relation circuit.
- A NIZK argument $(\mathsf{Setup}_N, \mathsf{P}_N, \mathsf{V}_N)$ with statistical zero knowledge for an NP language $\mathcal{L}'_N$ described along with the language for the BARGs. We assume that the non-adaptive soundness error with security parameter λ_1 for this construction is $2^{-\lambda_1^{\epsilon_1}}$ for some fixed $\epsilon_1 < 1$. We will set λ_1 later.
- A $(\mathsf{MTGen}, \mathsf{MTHash}, \mathsf{MTRead}, \mathsf{MTVerify}, \mathsf{MTWrite})$ be a Merkle hash tree using a collision resistant hash key hk. MTRead gives a local opening to any location i. $\mathsf{MTVerify}$ checks if this opening is correct. $\mathsf{MTWrite}$ to write to a location i first takes takes a local opening to location i, runs $\mathsf{MTVerify}$ and then, updates the hash using the updated value at location i.
- A NIZK-AoK $(\mathsf{Setup}_{N'}, \mathsf{P}_{N'}, \mathsf{V}_{N'})$ with statistical zero knowledge for the language

$$\mathcal{L}' = \{(\mathsf{hk}_H, v, 0, \mathsf{st}_0) : \exists \rho_0 \text{ s.t. } \mathsf{Verify}_H(\mathsf{hk}_0, v, 0, \mathsf{st}_0, \rho_0) = 1\}.$$

  We set the security parameter for this NIZK to be λ_2. We again assume that this NIZK has non-adaptive soundness error that is $2^{-\lambda_2^{\epsilon_2}}$. We will set λ_2 such that $\lambda_2^{\epsilon_2}$ is a large polynomial that is greater than the output size of the Hash_H.

Language for BARGs. We describe a language for our BARG scheme.

- **Input:** The CRS for a NIZK scheme crs_N, the CRS for the commitment scheme crs_C, the hash digests v, hash key hk for SEH, a hash key hk' for MTHash, $\mathsf{st}_0^* \in \{0,1\}^{\lambda}$, an index $g \in [T]$ and two blocks $\mathbf{d}_i^g, \mathbf{d}_j^g$ of the commitment.
- **Witness:** A commitment $\mathsf{com}(\tilde{w}_g)$ and proof π_N.
- **Description of Relation Circuit:** The circuit checks if $V_N(\mathsf{crs}_N, \tilde{x}_g = (\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0^*, \mathsf{com}, g, \mathbf{d}_i^g, \mathbf{d}_j^g), \mathsf{com}(\tilde{w}_g), \pi_N) = 1$ for the NP language $\mathcal{L}'_N$ whose instance is $(\tilde{x}_g, \mathsf{com}(\tilde{w}_g))$ and the witness is an opening to $\mathsf{com}(\tilde{w}_g)$ such that $\tilde{x}_g \in \mathcal{L}_N$ defined below with witness $\tilde{w}_g$. Let s be the size of this relation circuit.
- **Description of $\mathcal{L}_N$:** $\tilde{x}_g = (\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0^*, g, \mathbf{d}_i^g, \mathbf{d}_j^g) \in \mathcal{L}_N$ if there exists a witness $\tilde{w}_g = (\mathsf{st}_{g-1}, \rho_{g-1})$, (st_g, ρ_g), (b_i^g, ρ_i^g), $(\mathbf{d}_i^g, \delta_i^g)$, (b_j^g, ρ_j^g), $(\mathbf{d}_j^g, \delta_j^g)$, (b_ℓ^g, ρ_ℓ^g) such that:
  1. If $g = 1$, check if $\mathsf{st}_0^* = \mathsf{st}_0$.
  2. Check if $\mathsf{Verify}_H(\mathsf{hk}, v, g-1, \mathsf{st}_{g-1}, \rho_{g-1}) = 1$ and $\mathsf{Verify}_H(\mathsf{hk}, v, g, \mathsf{st}_g, \rho_g) = 1$.
  3. Let i and j be the input wires to g-th gate in C and let ℓ be the output wire.
  4. If $i \in [m]$ (that is, if i corresponds to a witness wire that is given as input to the circuit), check if $\mathsf{LocVer}(\mathsf{crs}_C, \mathbf{d}_i^g, b_i^g, \delta_i^g) = 1$. Similarly, if $j \in [m]$ (that is, if i corresponds to a witness wire that is given as input to the circuit), check if $\mathsf{LocVer}(\mathsf{crs}_C, \mathbf{d}_j^g, b_j^g, \delta_j^g) = 1$.
  5. Check if $\mathsf{MTVerify}(\mathsf{hk}', \mathsf{st}_{g-1}, b_i^g, \rho_i^g) = 1$, $\mathsf{MTVerify}(\mathsf{hk}', \mathsf{st}_{g-1}, b_j^g, \rho_j^g) = 1$ and $\mathsf{MTVerify}(\mathsf{hk}', \mathsf{st}_{g-1}, b_\ell^g, \rho_\ell^g) = 1$.
  6. Check if $b_\ell^g = 0$.
  7. Compute $b_\ell = \mathsf{NAND}(b_i^g, b_j^g)$. If $g = T$, check if $b_\ell = 1$.
  8. Check if $\mathsf{st}_g = \mathsf{MTWrite}(\mathsf{hk}', v, \ell, (b_\ell^g, \rho_\ell^g), b_\ell)$.

Construction. We give the formal description of a rate-1 NIZK-AoK for CSAT. Recall that the instance is (C, x) and assume that T is the size of C.

- $\mathsf{Setup}(1^\lambda)$:
  1. Sample $\mathsf{crs}_C \leftarrow \mathsf{Setup}_C(1^\lambda)$.
  2. Sample $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
  3. Sample $\mathsf{hk} \leftarrow \mathsf{Gen}_H(1^\lambda, N, 1)$ where $N = (T+1)\cdot\lambda$.
  4. Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
  5. Sample $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, 1)$ where $k = T$.
  6. Output $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$.
- $\mathsf{P}(\mathsf{crs}, x, w)$: Let $x \in \{0,1\}^n$ and $w \in \{0,1\}^m$.
  1. Compute $(\mathsf{com}, \mathsf{st}_w) \leftarrow \mathsf{Com}(\mathsf{crs}_C, w)$.
  2. Let $\mathsf{init}_0 = x \| 0^S$ and compute $\mathsf{st}_0 = \mathsf{MTHash}(\mathsf{hk}', \mathsf{init}_0)$.
  3. Evaluate $\mathcal{C}_n(w, x)$ to compute all the values carried by each wire. Let $\mathsf{wire}_{n+m+1}, \ldots, \mathsf{wire}_{n+m+S}$ be the values carried by the internal wires.
  4. For every g from 1 to T:
     (a) Let i and j be the input wires to g-th gate and let ℓ be the output wire.
     (b) Set $(b_i^g, \rho_i^g) = \mathsf{MTRead}(\mathsf{hk}', \mathsf{st}_{g-1}, \mathsf{init}_{g-1}, i)$, $(b_j^g, \rho_j^g) = \mathsf{MTRead}(\mathsf{hk}', \mathsf{st}_{g-1}, \mathsf{init}_{g-1}, j)$ and $(b_\ell^g, \rho_\ell^g) = \mathsf{MTRead}(\mathsf{hk}', \mathsf{st}_{g-1}, \mathsf{init}_{g-1}, \ell)$.
     (c) Let init_g be same as init_{g-1} except that ℓ-th entry is set to wire_ℓ.
     (d) Compute $\mathsf{st}_g = \mathsf{MTHash}(\mathsf{hk}', \mathsf{init}_g)$.
  5. Compute $v = \mathsf{Hash}(\mathsf{hk}, \mathsf{st}_0 \| \ldots \| \mathsf{st}_T)$ and for each $i \in [0, T]$, compute $(\mathsf{st}_i, \rho_i) \leftarrow \mathsf{Open}(\mathsf{hk}, v, \mathsf{st}_0 \| \ldots \| \mathsf{st}_T, i)$.
  6. Decompose com into $(\mathbf{d}_1, \ldots, \mathbf{d}_m)$ where each block $\mathbf{d}_i$ contains w_i.[4]
  7. For each $g \in [T]$, compute the witness $\tilde{w}_g$ for the language $\mathcal{L}_N$ with the instance being $\tilde{x}_g = (\mathsf{crs}_N, v, \mathsf{hk}, \mathsf{st}_0, g, \mathbf{d}_i^g, \mathbf{d}_j^g)$.
  8. For each $g \in [T]$, compute $\mathsf{com}(\tilde{w}_g) \leftarrow \mathsf{Com}(\mathsf{crs}_C, \tilde{w}_g)$ and generate the NIZK proof π_g for the language $\mathcal{L}'_N$.
  9. Compute $\pi_B = \mathsf{P}_B(\mathsf{crs}_B, (\tilde{x}_1, \ldots, \tilde{x}_T), ((\mathsf{com}(\tilde{w}_1), \pi_1), \ldots, (\mathsf{com}(\tilde{w}_T), \pi_T)))$.
  10. Set $x' = (\mathsf{hk}, v, 0, \mathsf{st}_0)$ and $w' = \rho_0$. Compute $\sigma \leftarrow \mathsf{P}_{N'}(\mathsf{crs}_{N'}, x', w')$.
  11. Output $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
- $\mathsf{V}(\mathsf{crs}, x, \pi)$:
  1. Parse π as $(\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$
  2. Compute $\mathsf{init}_0 = x \| 0^S$ and check if $\mathsf{st}_0 = \mathsf{MTHash}(\mathsf{hk}', \mathsf{init}_0)$.
  3. Decompose com into $(\mathbf{d}_1, \ldots, \mathbf{d}_m)$ where each block $\mathbf{d}_i$ contains the bit w_i.
  4. Set $x' = (\mathsf{hk}, v, 0, \mathsf{st}_0)$ and check if $\mathsf{V}_{N'}(\mathsf{crs}_{N'}, x', \sigma) = 1$.
  5. For each $g \in T$, set $\tilde{x}_g = (\mathsf{crs}_N, v, \mathsf{hk}, \mathsf{st}_0, g, \mathbf{d}_i^g, \mathbf{d}_j^g)$.
  6. Check if $\mathsf{V}_B(\mathsf{crs}_B, (\tilde{x}_1, \ldots, \tilde{x}_T), \pi_B) = 1$.
  7. If all the checks pass, V outputs 1.

[4] This can be done by decomposing $(\mathbf{c}_1, \ldots, \mathbf{c}_\ell)$ (as per Definition 5) and computing $(\mathbf{d}_1, \ldots, \mathbf{d}_m)$ where $\mathbf{d}_{i(j-1)+1} = \cdots = \mathbf{d}_{i(j-1)} = \mathbf{c}_j$ for all $i \in [k/\ell]$ and $j \in [\ell]$. In other words, each block $\mathbf{d}_i$ is identical to the block $\mathbf{c}_j$ for which we can open w_i.

5.1 Proof Size

Setting the parameters. We set the security parameter for all the primitives except the statistical NIZK arguments $(\mathsf{Setup}_N, \mathsf{P}_N, \mathsf{V}_N)$ and $(\mathsf{Setup}_{N'}, \mathsf{P}_{N'}, \mathsf{V}_{N'})$ to be λ. For the first NIZK, we set the security parameter to be λ_1 and for the second, we set it to λ_2. We have already explained how to set λ_2. We now explain how to set λ_1 and ϵ. Let $\tilde{x}_g$ be an instance of $\mathcal{L}_N$ and let $\tilde{w}_g$ be the corresponding witness. We parse $\tilde{x}_g$ as $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0^*, g, \mathbf{d}_i^g, \mathbf{d}_j^g)$. From the property of the dual-mode commitment scheme and the SEH function, we infer that $\ell_1 = (|v| + |\mathbf{d}_i^g| + |\mathbf{d}_j^g|) \leq |w|^\epsilon \cdot \mathsf{poly}(\lambda)$ and $\ell_2 = |\tilde{w}_g| \leq |w|^\epsilon \cdot \mathsf{poly}(\lambda)$. We will set λ_1 such that $\lambda_1^{\epsilon_1} \gg 2^{\ell_1+\ell_2}$. Let s be the size of the relation circuit for the BARG scheme. Note that $s = \mathsf{poly}(|\tilde{x}_g|, |\tilde{w}_g|, \lambda_1) = |w|^{\alpha\epsilon} \cdot \mathsf{poly}(\lambda)$ for some constant $\alpha > 1$. The size of the BARG proof is $\mathsf{poly}(\lambda, \log T) \cdot \mathsf{poly}(s)$ and this is at most $|w|^{\beta\epsilon} \cdot \mathsf{poly}(\lambda)$ for some constant $\beta > 1$. We set ϵ such that $\beta\epsilon < 1$. Hence, the size of the BARG proof can be upper bounded by $|w|^{\epsilon'} \cdot \mathsf{poly}(\lambda)$ for some $\epsilon' < 1$.

Size of the proof π. The proof π is composed by $(\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$ where:

- $|\mathsf{com}| = |w| + |w|^{1-\epsilon} \cdot \mathsf{poly}(\lambda)$ where $\epsilon > 0$.
- $|v| = \mathsf{poly}(\lambda)$, since we just need to extract st_i (which is of size $\mathsf{poly}(\lambda)$).
- $|\mathsf{st}_0| = \mathsf{poly}(\lambda)$ and $|\sigma| = \mathsf{poly}(\lambda_2) = \mathsf{poly}(\lambda)$.
- $|\pi_B| = |w|^{\epsilon'} \cdot \mathsf{poly}(\lambda)$ for some $\epsilon' < 1$.

Thus, we have that the rate ϱ_π of the proof is

$$\begin{aligned}\varrho_\pi = \frac{\pi}{|w|} &= \frac{|\mathsf{com}| + |v| + |\mathsf{st}_0| + |\sigma| + |\pi_B|}{|w|} = \frac{|w| + o(|w|) \cdot \mathsf{poly}(\lambda)}{|w|} \\ &= 1 + o(|w|) \cdot \mathsf{poly}(\lambda)/|w|\end{aligned}$$

which approaches 1 for large enough $|w|$.

5.2 Analysis

We now proceed with the analysis of our scheme.

Theorem 5 (Completeness)**.** *The NIZK construction presented above is complete given that the underlying fully-local extractable commitment is local opening complete, the dual-mode SEH is opening complete and the underlying NIZKs and BARG scheme are complete.*

Proof. Assume that $x \in \mathcal{L}$ and that w is the corresponding witness. We will show that $1 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$ where $\pi = (w', v, (\mathsf{st}_0, \rho_0), \mathsf{com}, \pi_B)$.

First, by the completeness of N', we have that $\mathsf{V}_{N'}(\mathsf{crs}_{N'}, x', \sigma) = 1$ where $x' = (\mathsf{hk}, v, 0, \mathsf{st}_0) \in \mathcal{L}'$ by construction.

By construction and opening completeness of the underlying extractable commitment and SEH, we have that $\tilde{x}_g = (\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0^*, \mathsf{com}, g, \mathbf{d}_i^g, \mathbf{d}_i^g, \mathbf{d}_j^g) \in \mathcal{L}_N$ by all $g \in [T]$. Let $\tilde{w}_g$ be the corresponding witness. Then, $\mathsf{V}_N(\mathsf{crs}_N, \tilde{x}_g,$
$(\mathsf{com}(\tilde{w}_g), \pi_g)) = 1$ by the completeness of the underlying NIZK, where π_g is an honestly generated proof. Moreover, by the completeness of the underlying (index) BARG, we have that

$$\mathsf{V}_B(\mathsf{crs}_B, (\mathsf{crs}_N, \mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0, \mathsf{com}), \pi_B) = 1$$

where $\pi_B = \mathsf{P}_B(\mathsf{crs}_B, (\mathsf{crs}_N, \mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0, \mathsf{com}), ((\mathsf{com}(\tilde{w}_1), \pi_1), \ldots, (\mathsf{com}(\tilde{w}_T), \pi_T)))$. □

We show that our NIZK system satisfies adaptive argument of knowledge. We note that this does not contradict the result of Pass [Pas13] as we rely on a rate-0 NIZK with sub-exponentially small non-adaptive soundness error.

Theorem 6 (Adaptive Argument of knowledge)**.** *The NIZK construction presented above satisfies adaptive argument of knowledge given that the underlying SEH is index hiding and SSB, the underlying BARG is index hiding and somewhere extractable, the underlying NIZK has sub-exponentially small non-adaptive soundness error and the underlying Merkle hash function is collision resistant.*

Proof. Let $(\mathsf{Ext}_{C,1}, \mathsf{Ext}_{C,2})$ be the extractor of the commitment scheme. We will start by describing the extractor $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$.

$\mathsf{Ext}_1(1^\lambda)$:

- $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$.
- $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
- $(\mathsf{hk}_H, \mathsf{td}_H) \leftarrow \mathsf{ExtGen}_H(1^\lambda, N, T)$.
- Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
- $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, T)$.
- Output $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ and $\mathsf{td} = (\mathsf{td}_C, \mathsf{td}_H, \mathsf{td}_B)$.

$\mathsf{Ext}_2(\mathsf{td}, x, \pi)$:

- Check if $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$. If not, output $\perp$.
- Else, parse π as $(\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
- Extract $w \leftarrow \mathsf{Extract}_C(\mathsf{td}_C, \mathsf{com})$.

We start by moving into a hybrid where both the extractable commitment and the SEH are moved from the statistially hiding mode into the extractable mode. The proof follows the following sequence of hybrids.

- Hyb_0 : This is the argument of knowledge game described in Definition 1.
- Hyb_1 : This hybrid is identical to the previous one except that we set $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$

  Claim 1. Hyb_0 *and* Hyb_1 *are indistinguishable given that the FLC is dual-mode indistinguishable.*

  The hybrids are identical except for the mode in which the setup of the FLC is set to (either statistically hiding mode or extractable mode). Hence, we could use an adversary that distinguishes the hybrids to break the mode indistinguishability of the underlying FLC.
- Hyb_2 : This hybrid is identical to the previous one except that we set $(\mathsf{hk}_H, \mathsf{td}_H) \leftarrow \mathsf{ExtGen}_H(1^\lambda, N, T)$.

  Claim 2. Hyb_1 *and* Hyb_2 *are indistinguishable given that the SEH is dual-mode indistinguishable.*

  The hybrids are identical except for the mode in which the setup of the SEH is set to (either statistically hiding mode or somewhere extractable mode). Hence, we could use an adversary that distinguishes the hybrids to break the mode indistinguishability of the underlying SEH.

To prove argument of knowledge, it is sufficient to prove the following statement for hybrid Hyb_2:

If output of Ext_2 it not $\perp$, then $\mathcal{R}(x, w) = 1$ except with negligible probability.

Let (C, x) be the statement output by the prover and let w be the witness extracted by the extractor described above. Let us define st_i that is computed as follows.

1. Set $\mathsf{init}_0 = x \| 0^S$.

2. For every g from 1 to i:
   (a) Let i' and j' be the input wires to g-th gate and let ℓ be the output wire.
   (b) Let $\alpha = \mathsf{init}_{g,i'}$ and $\beta = \mathsf{init}_{g,j'}$.
   (c) Compute $\gamma = \mathsf{NAND}(\alpha, \beta)$.
   (d) Let init_g be same as init_{g-1} except that ℓ-th entry is set to γ.
3. Compute $\mathsf{st}_i = \mathsf{MTHash}(\mathsf{hk}, \mathsf{init}_i)$.

We prove the following claim about the above computed st_i.

Lemma 4. *For any $i \in [N]$ and for any PPT adversary P^*, consider the following experiment.*

1. *Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where*
   (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
   (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ *and* $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
   (c) $(\mathsf{hk}_H, \mathsf{td}_H) \leftarrow \mathsf{ExtGen}_H(1^\lambda, N, i)$ *where* $N = (T+1) \cdot \lambda$,
   (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ *where* $k = T$.
   (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, *where* $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma_0), \pi_B)$

Then, we have

$$\Pr\left[\mathsf{st}_i \neq \mathsf{st}_i^* \wedge \mathsf{V}(\mathsf{crs}, x, \pi) = 1 : \mathsf{st}_i^* \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)\right] \leq \mathsf{negl}(\lambda).$$

Proof. We will prove this claim via induction on i. We start by analyzing the base case where $i = 0$.

Base case $i = 0$. We prove that the invariant holds when $i = 0$ except with probability $2^{|v|} \cdot 2^{-\lambda_2^{\epsilon_2}} + \mu(\lambda) \leq \mathsf{negl}(\lambda)$ where $\mu(\lambda)$ is the statistical error in the binding game in the SEH. Assume for the sake of contradiction that this is not the case. That is, the invariant does not hold at $i = 0$ with probability at least $2^{|v|} \cdot 2^{-\lambda_2^{\epsilon_2}} + \mu(\lambda)$. We arrive at a contradiction to the non-adaptive soundness of the the NIZK proof system $(\mathsf{Setup}_{N'}, \mathsf{P}_{N'}, \mathsf{V}_{N'})$.

- Hyb_1 : We condition on the event that for v that is output by the cheating prover P^* has a unique opening at location 0. By the statistical binding property of the SEH, we have that the invariant does not hold in Hyb_1 with probability at least $2^{|v|} \cdot 2^{-\lambda_2^{\epsilon_2}}$.
- Hyb_2 : We first make a random guess for v before the prover generates the proof and say that the invariant does not hold only if our guess is correct. The probability that invariant does not hold in Hyb_2 is at least $2^{-\lambda_2^{\epsilon_2}}$. Note that if the invariant does not hold, then $x' = (\mathsf{hk}, v, 0, \mathsf{st}_0)$ is not in the language $\mathcal{L}'$ and we break the soundness of the NIZK in Hyb_2 with probability at least $2^{-\lambda_2^{\epsilon_2}}$. This is a contradiction.

This proves that the invariant holds for $i = 0$.

Induction step. We will now show that if the induction hypothesis holds for $i - 1$ then it also holds for i. We will prove this via a hybrid argument.

- Hyb_0 : Consider the following experiment:
  1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
     (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
     (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.

(c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{ExtGen}_H(1^\lambda, N, i-1)$ where $N = (T+1)\cdot\lambda$,
(d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i-1)$ where $k = T$.
(e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.

2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
3. $\mathsf{st}^*_{i-1} \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.
4. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions hold.
   (a) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$

Note that by induction hypothesis, the probability that we output 1 in Hyb_0 is negligible.

- Hyb_1 : This hybrid is identical to the previous one except that we generate crs_B as $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$. Concretely,

1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
   (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
   (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
   (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i-1)$ where $N = (T+1)\cdot\lambda$,
   (d) $\underline{(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)}$ where $k = T$.
   (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
3. $\mathsf{st}^*_{i-1} \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.
4. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions hold.
   (a) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$

Claim 3. *Hybrids* Hyb_0 *and* Hyb_1 *are indistinguishable given that the BARG scheme is index hiding.*

We show from the index hiding property of the BARG scheme that the probability that we output 1 in Hyb_1 is negligible. Suppose for the sake of contradiction assume that the probability we output 1 in Hyb_1 is non-negligible. We show that we can break the index hiding property of BARG. We interact with this challenger and send i and $i-1$ as the challenge index locations. We receive crs_B from the challenger. We generate the rest of the components of the CRS as before and send this to the prover. After receiving the proof, we compute st^*_{i-1} and st_i as before. If the proof verifies and $\mathsf{st}^*_{i-1} \neq \mathsf{st}_i$, we output 1. Eles, we output 0. Note that if crs_B is generated with respect to location i, then the reduction mimics Hyb_1. Else, it mimics Hyb_0. Therefore, this reduction can break the index hiding property of the BARG scheme with non-negligible advantage and this is a contradiction.

- Hyb_2 : In this hybrid, we do the following:

1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
   (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
   (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
   (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i-1)$ where $N = (T+1)\cdot\lambda$,
   (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
   (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
3. Compute $\underline{(\mathsf{com}(\tilde{w}_i), \pi_i) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)}$ and $\underline{\tilde{w}_i \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_i))}$.
4. $\mathsf{st}^*_{i-1} \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.

5. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
   (a) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$
   (b) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$

Claim 4. *Hybrids* Hyb_1 *and* Hyb_2 *are indistinguishable except with advantage* $\mathsf{negl}(\lambda) + 2^{\ell_1+\ell_2} \cdot 2^{-\lambda_1^{\epsilon_1}}$ *(where* ℓ_1, ℓ_2 *are defined in Section 5.1) given that the underlying BARG is somewhere extractable and the non-adaptive soundness of NIZK* $(\mathsf{Setup}_N, \mathsf{P}_N, \mathsf{V}_N)$.

Let ϵ_1 be the probability that the experiment outputs 1 in Hyb_1. We consider the following sequence of sub-hybrids:

- $\mathsf{Hyb}_{1,1}$: This hybrid is same as Hyb_1 except that if we output 1 only if $(\mathsf{com}(\tilde{w}_i), \pi_i)$ is a valid witness for the BARG relation. Let $\epsilon_{1,1}$ be the probability that the experiment outputs 1 in $\mathsf{Hyb}_{1,1}$ is $\epsilon_{1,1}$. From the somewhere extractability of the BARG scheme, we have $|\epsilon_{1,1} - \epsilon_1| \leq \mathsf{negl}(\lambda)$.
- $\mathsf{Hyb}_{1,2}$: In this hybrid, we only output 1 if $\mathsf{com}(\tilde{w}_i)$ has an unique opening. Let $\epsilon_{1,2}$ be the probability that the output of the experiment is 1 in this hybrid. From the extraction correctness of the dual-mode extractable commitment, it follows that $|\epsilon_{1.2} - \epsilon_{1,1}| \leq \mathsf{negl}(\lambda)$.
- $\mathsf{Hyb}_{1,3}$: In this hybrid, we make random guess for $(v, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ and $\mathsf{com}(\tilde{w}_i)$ and output 1 only in this case. Let ℓ_1 be the length of $(v, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ and ℓ_2 be the length of $\mathsf{com}(\tilde{w}_i)$. The probability that we output 1 in this hybrid is $\epsilon_{1,3} \leq \frac{\epsilon_{1,2}}{2^{\ell_1+\ell_2}}$.
- $\mathsf{Hyb}_{1,4}$: In this hybrid, we also output 1 if $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ as in Hyb_2. It follows from the non-adaptive soundness of the NIZK system $(\mathsf{Setup}_N, \mathsf{P}_N, \mathsf{V}_N)$ that $|\epsilon_{1,4} - \epsilon_{1,3}| \leq 2^{-\lambda_1^{\epsilon_1}}$ where $\epsilon_{1,4}$ is the probability that the output of the experiment is 1 in $\mathsf{Hyb}_{1,4}$.
- $\mathsf{Hyb}_{1,5}$: In this hybrid, we reverse the change made in $\mathsf{Hyb}_{1,3}$. It follows that $\epsilon_{1,4} \leq \frac{\epsilon_{1,5}}{2^{\ell_1+\ell_2}}$ where $\epsilon_{1,5}$ is the probability that the output of the experiment is 1 in $\mathsf{Hyb}_{1,5}$.
- $\mathsf{Hyb}_{1,6}$ and $\mathsf{Hyb}_{1,7}$: We reverse the changes made in $\mathsf{Hyb}_{1,2}$ and $\mathsf{Hyb}_{1,1}$ respectively. Note that $\mathsf{Hyb}_{1,7}$ is identical to Hyb_2.

This proves that no adversary can distinguish between Hyb_1 and Hyb_2 except with advantage $\mathsf{negl}(\lambda) + 2^{\ell_1+\ell_2} \cdot 2^{-\lambda_1^{\epsilon_1}} \leq \mathsf{negl}(\lambda)$.

- Hyb_3 : This hybrid is identical to the previous one except that we parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c^g_{i'}, \nu^g_{i'}), \delta^g_{i'}, \delta^g_{j'}, (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell))$. Additionally, we output 1 if either $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$ or $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ or $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$. Concretely,

  1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
     (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
     (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
     (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i-1)$ where $N = (T+1) \cdot \lambda$,
     (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
     (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
  2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
  3. Compute $(\mathsf{com}(\tilde{w}_i), \pi_i) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)$ and $\tilde{w}_i \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_i))$.
  4. Parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c^g_{i'}, \nu^g_{i'}), \delta^g_{i'}, \delta^g_{j'}, (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell))$.
  5. $\mathsf{st}^*_{i-1} \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.

6. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
   (a) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$
   (b) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$
   (c) $\underline{\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}}$.

Claim 5. *Hybrids* Hyb_2 *and* Hyb_3 *are indistinguishable given that the SEH is somewhere extractable.*

We prove using the somewhere binding property of the SEH that the probability that we output 1 in Hyb_3 is negligible. Assume for the sake of contradiction that the probability we output 1 in Hyb_3 is non-negligible. Since the probability we output 1 in Hyb_2 is negligible, we infer that the following event happens with non-negligible probability: (i) the proof π verifies, (ii) $\mathsf{st}_{i-1} = \mathsf{st}^*_{i-1}$, (iii) $\tilde{w}_i$ is a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$, and (iv) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$. Since $\tilde{w}_i$ is a valid witness, it follows that $(\mathsf{st}'_{i-1}, \rho'_{i-1})$ is a valid opening to v. If $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$, then this breaks the somewhere binding property of SEH and this is a contradiction.

- Hyb_4 : This hybrid is identical to the previous one except that we compute $\hat{w} \leftarrow \mathsf{Ext}_C(\mathsf{td}_C, \mathsf{com})$ we output 1 if either i) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$ or ii) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ or iii) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$ or iv) $c^g_i \neq \hat{w}_i$ if $i \in [m]$ or $c^g_j \neq \hat{w}_j$ if $j \in [m]$. Concretely,

  1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
     (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
     (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
     (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i-1)$ where $N = (T+1) \cdot \lambda$,
     (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
     (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
  2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
  3. Compute $(\mathsf{com}(\tilde{w}_i), \pi_i) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)$ and $\tilde{w}_i \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_i))$.
  4. Parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c^g_{i'}, \nu^g_{i'}), \delta^g_{i'}, \delta^g_{j'}, (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell))$.
  5. $\underline{\hat{w} \leftarrow \mathsf{Ext}_C(\mathsf{td}_C, \mathsf{com})}$
  6. $\mathsf{st}^*_{i-1} \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.
  7. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
     (a) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$
     (b) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$
     (c) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$.
     (d) $\underline{c^g_i \neq \hat{w}_i \text{ if } i \in [m] \text{ or } c^g_j \neq \hat{w}_j \text{ if } j \in [m]}$.

Claim 6. *Hybrids* Hyb_3 *and* Hyb_4 *are indistinguishable given that the underlying FLC has extraction correctness.*

Assume for the sake of contradiction that we output 1 in Hyb_4 is non-negligible probability. Since the probability we output 1 in Hyb_3 is negligible, the following event happens with non-negligible probability: (i) the proof π verifies, ii) $\mathsf{st}_{i-1} = \mathsf{st}^*_{i-1}$, iii) $\tilde{w}_i$ is a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$, iii) $\mathsf{st}'_{i-1} = \mathsf{st}^*_{i-1}$. Then if we output 1 in this hybrid it is because $c^g_i \neq \hat{w}_i$ or $c^g_j \neq \hat{w}_j$, which breaks the extraction correctness of the underlying fully-local extractable commitment.

- Hyb_5 : This hybrid is identical to the previous one except that we output 1 if either i) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$ or ii) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ or iii) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$ or iv) $c^g_i \neq \hat{w}_i$ if $i \in [m]$ or $c^g_j \neq \hat{w}_j$ if $j \in [m]$, or v) $(b^g_{i'}, \rho^g_{i'}) \neq (c^g_{i'}, \nu^g_{i'})$ or $(b^g_{j'}, \rho^g_{j'}) \neq (c^g_{j'}, \nu^g_{j'})$ or $(b^g_\ell, \rho^g_\ell) \neq (c^g_\ell, \nu^g_\ell)$. Concretely,

  1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
     (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
     (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
     (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i-1)$ where $N = (T+1)\cdot\lambda$,
     (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
     (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
  2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
  3. Compute $(\mathsf{com}(\tilde{w}_i), \pi_i) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)$ and $\tilde{w}_i \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_i))$.
  4. Parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c^g_{i'}, \nu^g_{i'}), \delta^g_{i'}, \delta^g_{j'}, (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell))$.
  5. Let $(b^g_{i'}, \rho^g_{i'})$, $(b^g_{j'}, \rho^g_{j'})$, (b^g_ℓ, ρ^g_ℓ) be the local openings to the i'-th, j'-th and the ℓ-th locations in st_{i-1}.
  6. $\hat{w} \leftarrow \mathsf{Ext}_C(\mathsf{td}_C, \mathsf{com})$
  7. $\mathsf{st}^*_{i-1} \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.
  8. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
     (a) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$
     (b) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_n$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$
     (c) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$.
     (d) $c^g_i \neq \hat{w}_i$ if $i \in [m]$ or $c^g_j \neq \hat{w}_j$ if $j \in [m]$.
     (e) $(b^g_{i'}, \rho^g_{i'}) \neq (c^g_{i'}, \nu^g_{i'})$ or $(b^g_{j'}, \rho^g_{j'}) \neq (c^g_{j'}, \nu^g_{j'})$ or $(b^g_\ell, \rho^g_\ell) \neq (c^g_\ell, \nu^g_\ell)$.

Claim 7. *Hybrids* Hyb_4 *and* Hyb_5 *are indistinguishable given that* MTHash *is collision-resistant.*

We prove using the collision-resistance of the MTHash that the probability we output 1 in Hyb_5 is negligible. Assume for the sake of contradiction that we output 1 in Hyb_5 is non-negligible probability. Since the probability we output 1 in Hyb_4 is negligible, the following event happens with non-negligible probability: (i) the proof π verifies, (ii) $\mathsf{st}_{i-1} = \mathsf{st}^*_{i-1}$, (iii) $\tilde{w}_i$ is a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$, (iv) $\mathsf{st}'_{i-1} = \mathsf{st}^*_{i-1}$, and (v) $(b^g_{i'}, \rho^g_{i'}) \neq (c^g_{i'}, \nu^g_{i'})$ or $(b^g_{j'}, \rho^g_{j'}) \neq (c^g_{j'}, \nu^g_{j'})$ or $(b^g_\ell, \rho^g_\ell) \neq (c^g_\ell, \nu^g_\ell)$. It follows from conditions (ii) and (iv) that $\mathsf{st}_{i-1} = \mathsf{st}'_{i-1}$. Since $\tilde{w}_i$ is a valid witness, it follows that $(c^g_{i'}, \nu^g_{i'}), (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell)$ are valid openings to st_{i-1}. However, if condition (v) happens, then we have found a collision to the MTHash function family and this is a contradiction.

- Hyb_6 : This hybrid is identical to the previous one except that we output 1 if either i) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$ or ii) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ or iii) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$ or iv) $c^g_i \neq \hat{w}_i$ if $i \in [m]$ or $c^g_j \neq \hat{w}_j$ if $j \in [m]$, or v) $(b^g_{i'}, \rho^g_{i'}) \neq (c^g_{i'}, \nu^g_{i'})$ or $(b^g_{j'}, \rho^g_{j'}) \neq (c^g_{j'}, \nu^g_{j'})$ or $(b^g_\ell, \rho^g_\ell) \neq (c^g_\ell, \nu^g_\ell)$, or vi) $\mathsf{st}'_i \neq \mathsf{st}_i$. Concretely,

  1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
     (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
     (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
     (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i-1)$ where $N = (T+1)\cdot\lambda$,

(d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
(e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.

2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
3. Compute $(\mathsf{com}(\tilde{w}_i), \pi_i) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)$ and $\tilde{w}_i \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_i))$.
4. Parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c^g_{i'}, \nu^g_{i'}), \delta^g_{i'}, \delta^g_{j'}, (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell))$.
5. Let $(b^g_{i'}, \rho^g_{i'})$, $(b^g_{j'}, \rho^g_{j'})$, (b^g_ℓ, ρ^g_ℓ) be the local openings to the i'-th, j'-th and the ℓ-th locations in st_{i-1}.
6. $\hat{w} \leftarrow \mathsf{Ext}_C(\mathsf{td}_C, \mathsf{com})$
7. $\mathsf{st}^*_{i-1} \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.
8. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
   (a) $\mathsf{st}_{i-1} \neq \mathsf{st}^*_{i-1}$
   (b) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_n$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$
   (c) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$.
   (d) $(b^g_{i'}, \rho^g_{i'}) \neq (c^g_{i'}, \nu^g_{i'})$ or $(b^g_{j'}, \rho^g_{j'}) \neq (c^g_{j'}, \nu^g_{j'})$ or $(b^g_\ell, \rho^g_\ell) \neq (c^g_\ell, \nu^g_\ell)$.
   (e) $c^g_i \neq \hat{w}_i$ if $i \in [m]$ or $c^g_j \neq \hat{w}_j$ if $j \in [m]$.
   (f) $\mathsf{st}'_i \neq \mathsf{st}_i$.

Claim 8. *Hybrids* Hyb_5 *and* Hyb_6 *are indistinguishable by the correctness of* MTWrite.

Since $\tilde{w}_i$ is a valid witness and $(b^g_i, b^g_j) = (c^g_i, c^g_j)$, it follows from the perfect extraction of com that the output of the gate i that is computed in $\tilde{w}_i$ is same as the output computed as part of init_i. Since init_{i-1} and init_i only differ at the ℓ-th location, it follows from the correctness of MTWrite that $\mathsf{st}_i = \mathsf{st}'_i$. Hence, the probability that we output 1 in Hyb_6 is negligible.

- Hyb_7 : This hybrid is identical to the previous one except that we output 1 if either i) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ or ii) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$ or iii) $c^g_i \neq \hat{w}_i$ if $i \in [m]$ or $c^g_j \neq \hat{w}_j$ if $j \in [m]$, or iv) $(b^g_{i'}, \rho^g_{i'}) \neq (c^g_{i'}, \nu^g_{i'})$ or $(b^g_{j'}, \rho^g_{j'}) \neq (c^g_{j'}, \nu^g_{j'})$ or $(b^g_\ell, \rho^g_\ell) \neq (c^g_\ell, \nu^g_\ell)$, or v) $\mathsf{st}'_i \neq \mathsf{st}_i$. Concretely,

1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
   (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
   (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
   (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i-1)$ where $N = (T+1) \cdot \lambda$,
   (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
   (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
3. Compute $(\mathsf{com}(\tilde{w}_i), \pi_i) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)$ and $\tilde{w}_i \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_i))$.
4. Parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c^g_{i'}, \nu^g_{i'}), \delta^g_{i'}, \delta^g_{j'}, (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell))$.
5. Let $(b^g_{i'}, \rho^g_{i'})$, $(b^g_{j'}, \rho^g_{j'})$, (b^g_ℓ, ρ^g_ℓ) be the local openings to the i'-th, j'-th and the ℓ-th locations in st_{i-1}.
6. $\hat{w} \leftarrow \mathsf{Ext}_C(\mathsf{td}_C, \mathsf{com})$
7. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
   (a) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_n$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$

(b) $c_i^g \neq \hat{w}_i$ if $i \in [m]$ or $c_j^g \neq \hat{w}_j$ if $j \in [m]$.
(c) $(b_{i'}^g, \rho_{i'}^g) \neq (c_{i'}^g, \nu_{i'}^g)$ or $(b_{j'}^g, \rho_{j'}^g) \neq (c_{j'}^g, \nu_{j'}^g)$ or $(b_\ell^g, \rho_\ell^g) \neq (c_\ell^g, \nu_\ell^g)$.
(d) $\mathsf{st}'_i \neq \mathsf{st}_i$.

Notice that the only difference in Hyb_5 and Hyb_6 is that in Hyb_6, we no longer extract st^*_{i-1} from v and do not check if $\mathsf{st}_{i-1} \stackrel{?}{=} \mathsf{st}^*_{i-1}$ and if $\mathsf{st}'_{i-1} \stackrel{?}{=} \mathsf{st}^*_{i-1}$. Hence, the probability that we output 1 in Hyb_6 is at most the probability that we output 1 in Hyb_5 and this is negligible.

- Hyb_8 : This hybrid is identical to the previous one except that we set hk to be $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i)$.

  1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
     (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
     (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
     (c) $\underline{(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i)}$ where $N = (T+1) \cdot \lambda$,
     (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
     (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
  2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
  3. Compute $(\mathsf{com}(\tilde{w}_i), \pi_i) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)$ and $\tilde{w}_i \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_i))$.
  4. Parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c_{i'}^g, \nu_{i'}^g), \delta_{i'}^g, \delta_{j'}^g, (c_{j'}^g, \nu_{j'}^g), (c_\ell^g, \nu_\ell^g))$.
  5. Let $(b_{i'}^g, \rho_{i'}^g)$, $(b_{j'}^g, \rho_{j'}^g)$, (b_ℓ^g, ρ_ℓ^g) be the local openings to the i'-th, j'-th and the ℓ-th locations in st_{i-1}.
  6. $\hat{w} \leftarrow \mathsf{Ext}_C(\mathsf{td}_C, \mathsf{com})$
  7. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
     (a) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_n$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{st}^*_0, \mathsf{com}, i)$
     (b) $c_i^g \neq \hat{w}_i$ if $i \in [m]$ or $c_j^g \neq \hat{w}_j$ if $j \in [m]$.
     (c) $(b_{i'}^g, \rho_{i'}^g) \neq (c_{i'}^g, \nu_{i'}^g)$ or $(b_{j'}^g, \rho_{j'}^g) \neq (c_{j'}^g, \nu_{j'}^g)$ or $(b_\ell^g, \rho_\ell^g) \neq (c_\ell^g, \nu_\ell^g)$.
     (d) $\mathsf{st}'_i \neq \mathsf{st}_i$.

Claim 9. *Hybrids* Hyb_7 *and* Hyb_8 *are indistinguishable given that the underlying SEH is index hiding.*

We show from the index hiding property of SEH that the probability that we output 1 in Hyb_8 is negligible. Assume for the sake of contradiction that the probability that we output 1 in Hyb_8 is negligible. We break the index hiding property of SEH. We interact with the challenger and provide $(i-1)$ and i as the challenge locations and obtain crs_H. We generate the rest of the components in the CRS as before and send to the prover. On obtaining the proof, we make the same checks as in Hyb_7 and output 1 if and only if the output of the experiment described in the hybrids outputs 1. Note that if crs_H is generated with respect to location i, the this reduction mimics Hyb_8. Else, it mimics Hyb_7. Therefore, this reduction contradicts the index hiding property of SEH.

- Hyb_9 : This hybrid is identical to the previous one except that we compute $\mathsf{st}^*_i \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$ and output 1 if either i) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_N$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}^*_0, \mathsf{com}, i, \mathbf{d}^i_j, \mathbf{d}^j_\ell)$ or ii) $\mathsf{st}'_{i-1} \neq \mathsf{st}^*_{i-1}$ or iii) $c_i^g \neq \hat{w}_i$ if $i \in [m]$ or $c_j^g \neq \hat{w}_j$ if $j \in [m]$, or iv) $(b_{i'}^g, \rho_{i'}^g) \neq (c_{i'}^g, \nu_{i'}^g)$ or $(b_{j'}^g, \rho_{j'}^g) \neq (c_{j'}^g, \nu_{j'}^g)$ or $(b_\ell^g, \rho_\ell^g) \neq (c_\ell^g, \nu_\ell^g)$, or v) $\mathsf{st}'_i \neq \mathsf{st}_i$, or vi) $\mathsf{st}'_i \neq \mathsf{st}_i$. Concretely,

  1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
     (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
     (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
     (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i)$ where $N = (T+1) \cdot \lambda$,

(d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
(e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.

2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
3. Parse $\tilde{w}_i$ as $((\mathsf{st}'_{i-1}, \rho'_{i-1}), (\mathsf{st}'_i \rho'_i), (c^g_{i'}, \nu^g_{i'}), \delta^g_{i'}, \delta^g_{j'}, (c^g_{j'}, \nu^g_{j'}), (c^g_\ell, \nu^g_\ell))$.
4. Let $(b^g_{i'}, \rho^g_{i'})$, $(b^g_{j'}, \rho^g_{j'})$, (b^g_ℓ, ρ^g_ℓ) be the local openings to the i'-th, j'-th and the ℓ-th locations in st_{i-1}.
5. $\hat{w} \leftarrow \mathsf{Ext}_C(\mathsf{td}_C, \mathsf{com})$
6. $\underline{\mathsf{st}^*_i \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)}$.
7. If $\mathsf{V}(\mathsf{crs}, x, \pi) = 1$, then we output 1 if any of the following conditions holds.
   (a) $\tilde{w}_i$ is not a valid witness for the instance of $\mathcal{L}_n$ given by $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{st}^*_0, \mathsf{com}, i)$
   (b) $c^g_i \neq \hat{w}_i$ if $i \in [m]$ or $c^g_j \neq \hat{w}_j$ if $j \in [m]$.
   (c) $(b^g_{i'}, \rho^g_{i'}) \neq (c^g_{i'}, \nu^g_{i'})$ or $(b^g_{j'}, \rho^g_{j'}) \neq (c^g_{j'}, \nu^g_{j'})$ or $(b^g_\ell, \rho^g_\ell) \neq (c^g_\ell, \nu^g_\ell)$.
   (d) $\mathsf{st}'_i \neq \mathsf{st}_i$.
   (e) $\underline{\mathsf{st}^*_i \neq \mathsf{st}'_i}$.

Claim 10. *Hybrids* Hyb_8 *and* Hyb_9 *are indistinguishable given that the underlying SEH is is somewhere extractable.*

Observe that since w^*_i is a valid witness, it follows from the somewhere binding property of SEH that the probability that we output 1 in Hyb_9 is negligible. This argument is similar to Claim 5.

- Hyb_{10} : This hybrid is identical to the previous one except that we output 1 if $\mathsf{st}^*_i \neq \mathsf{st}_i$.

1. Generate $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{crs}_N, \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{hk}, \mathsf{hk}')$ where
   (a) $(\mathsf{crs}_C, \mathsf{td}_C) \leftarrow \mathsf{ExtSetup}_C(1^\lambda)$
   (b) $\mathsf{crs}_N \leftarrow \mathsf{Setup}_N(1^{\lambda_1})$ and $\mathsf{crs}_{N'} \leftarrow \mathsf{Setup}_{N'}(1^{\lambda_2})$.
   (c) $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N, i)$ where $N = T \cdot \lambda$,
   (d) $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, i)$ where $k = T$.
   (e) Sample $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$.
2. $((C, x), \pi) \leftarrow \mathsf{P}^*(\mathsf{crs})$, where $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.
3. $\mathsf{st}^*_i \leftarrow \mathsf{Extract}_H(\mathsf{td}_H, v)$.
4. $\underline{\text{If } \mathsf{V}(\mathsf{crs}, x, \pi) = 1, \text{ then we output 1 if any of the following conditions holds.}}$
   (a) $\underline{\mathsf{st}^*_i \neq \mathsf{st}_i}$.

Claim 11. *Hybrids* Hyb_9 *and* Hyb_{10} *are indistinguishable.*

Observe that since $\mathsf{st}'_i = \mathsf{st}_i$ and $\mathsf{st}'_i = \mathsf{st}^*_i$ happens with overwhelming probability in Hyb_9, it follows that $\mathsf{st}_i = \mathsf{st}^*_i$ with overwhelming probability. As we are making fewer checks in Hyb_{10}, the probability that we output 1 in Hyb_{10} is at most the probability that we output 1 in Hyb_9. Hence, the probability that we output 1 in Hyb_{10} is negligible. This completes the induction step.

This completes the proof of the Lemma. □

It remains to show that output of the final gate (i.e., the T-th gate) computed in init_T is 1 except with negligible probability if the proof π is accepting. Suppose this is not the case. Let st_T be the digest of init_T. Let $(0, \rho_T)$ be the opening to the T-th location in st_T. By applying Lemma 4 at location T, we have that st^*_T (extracted from v) is same as st_T except with negligible probability. Let $(\mathsf{com}(\tilde{w}_T), \pi_T) \leftarrow \mathsf{Extract}_B(\mathsf{td}_B, \pi_B)$ and $\tilde{w}_T \leftarrow \mathsf{Ext}_{C,2}(\mathsf{td}_C, \mathsf{com}(\tilde{w}_T))$ (where π_B is part of π).

As in Claim 4, we can show $\tilde{w}_T$ is a valid witness to the language $\mathcal{L}_N$ with the instance being $(\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0^*, T, \mathbf{d}_i^T, \mathbf{d}_j^T)$ except with negligible probability. Parse this witness as

$$(\mathsf{st}'_{T-1}, \rho'_{T-1}), (\mathsf{st}'_T \rho'_T), (c^g_{i'}, \nu^g_{i'}), (\mathbf{e}^g_{i'}, \sigma^g_{i'}), (c^g_{j'}, \nu^g_{j'}), (\mathbf{e}^g_{j'}, \sigma^g_{j'}), (c^g_\ell, \nu^g_\ell)$$

where (i', j') are the input wires to the T-th gate and ℓ is the output wire. It follows from somewhere binding property of SEH that $\mathsf{st}'_T = \mathsf{st}^*_T$ except with negligible probability. Since $\tilde{w}_T$ is a valid witness, it follows that there is a valid opening $(1, \rho'_T)$ to the T-th location in st^*_T. However, this contradicts the collision resistance of the MTHash.

This completes the proof of the theorem. □

We now show that this construction satisfies statistical zero-knowledge property.

Theorem 7 (Zero-knowledge)**.** *The NIZK construction presented above is statistical zero-knowledge given that the underlying NIZKs have statistical zero-knowledge and the fully local extractable commitment is statistically hiding.*

Proof. We will first describe the simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$. Let $\mathsf{Sim}_N = (\mathsf{Sim}_{N,1}, \mathsf{Sim}_{N,2})$ be the simulator for the underlying NIZK scheme and $\mathsf{Sim}_{N'}$ be the simulator for the NIZK for language $\mathcal{L}'$.

$\mathsf{Sim}_1(1^\lambda)$: Sample $\mathsf{crs}_C \leftarrow \mathsf{Setup}_C(1^\lambda)$, $(\mathsf{crs}_{N'}, \mathsf{td}_{N'}) \leftarrow \mathsf{Sim}_{N',1}(1^{\lambda_2})$, $(\mathsf{hk}, \mathsf{td}_H) \leftarrow \mathsf{Gen}_H(1^\lambda, N)$, $\mathsf{hk}' \leftarrow \mathsf{MTGen}(1^\lambda)$, $(\mathsf{crs}_B, \mathsf{td}_B) \leftarrow \mathsf{Gen}_B(1^\lambda, 1^k, 1^s, 1)$ and $(\mathsf{crs}_N, \mathsf{td}_N) \leftarrow \mathsf{Sim}_{N,1}(1^{\lambda_1})$. Set $\mathsf{crs} = (\mathsf{crs}_C, \mathsf{hk}, \mathsf{hk}', \mathsf{crs}_B, \mathsf{crs}_N, \mathsf{crs}_{N'})$.

$\mathsf{Sim}_2(C, x, \mathsf{td})$:

- Compute $(\mathsf{com}, \rho) \leftarrow \mathsf{Com}_C(\mathsf{crs}_C, 0^m)$.
- Set $\mathsf{init}_0 = x || 0^S$. Let $\mathsf{st}_0 = \mathsf{MTHash}(\mathsf{hk}', \mathsf{init}_0)$.
- For each $g \in [T]$,
  - Set init_g exactly as init_{g-1} except that ℓ-th coordinate is set to $d \leftarrow\!\$\ \{0,1\}$.
  - Compute $\mathsf{st}_g = \mathsf{MTHash}(\mathsf{hk}, \mathsf{init}_g)$.
- Compute $v = \mathsf{Hash}(\mathsf{hk}, \mathsf{st}_0 \| \ldots \| \mathsf{st}_T)$.
- For each $g \in [T]$, generate $\mathsf{com}(\tilde{w}_g)$ as $\mathsf{Com}(\mathsf{crs}_C, \mathbf{0})$.
- For each $g \in [T]$, compute $\pi_g \leftarrow \mathsf{Sim}_{N,2}(\mathsf{td}_N, (\tilde{x}_g, \mathsf{com}(\tilde{w}_g)))$ where $\tilde{x}_g = (\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0, g, \mathbf{d}_i^g, \mathbf{d}_j^g)$.
- Compute π_B as in the real scheme.
- Set $x' = (\mathsf{hk}, v, 0, \mathsf{st}_0)$ and compute $\sigma \leftarrow \mathsf{Sim}_{N',2}(\mathsf{td}_{N'}, x')$.
- Output $\pi = (\mathsf{com}, v, (\mathsf{st}_0, \sigma), \pi_B)$.

We now show that for any (C, x) and $w \in \{0,1\}^m$ such that $C(w, x) = 1$, we have:

$$\left\{(\mathsf{crs}, \mathsf{Real}(\cdot,\cdot)) : \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)\right\} \approx_c \left\{(\mathsf{crs}, \mathsf{Ideal}(\mathsf{td}, \cdot)) :: (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Sim}_1(1^\lambda, (C, x))\right\}.$$

The proof follows from the following sequence of hybrids.

- Hyb_0 : This is the distribution of the honestly generated crs and proofs π.
- Hyb_1 : This hybrid is identical to the previous one except that we simulate the NIZK proofs $\pi_1, \ldots, \pi_T$ and crs_N using $\mathsf{Sim}_{N,2}$ and $\mathsf{Sim}_{N,1}$ respectively.

Claim 12. *Hybrids* Hyb_0 *and* Hyb_1 *are statistically indistinguishable given that the underlying NIZK for language* $\mathcal{L}_N$ *is statistical zero-knowledge.*

Assume for the sake of contradiction that Hyb_0 and Hyb_1 are distinguishable with non-negligible advantage. We give a reduction that breaks the zero-knowledge property of the NIZK. The reduction generates $\mathsf{hk}, \mathsf{hk}', \mathsf{crs}_{N'}, \mathsf{crs}_B, \mathsf{crs}_C$ as in the previous hybrid. It obtains crs_N from the challenger and uses it to generate the crs for the scheme. It then computes v, com as in Hyb_0. For each $g \in [T]$, it generates the witness $\tilde{w}_g$ for the instance $\tilde{x}_g = (\mathsf{crs}_C, v, \mathsf{hk}, \mathsf{hk}', \mathsf{st}_0^*,$
$\mathsf{com}, g, \mathbf{d}_i^g, \mathbf{d}_j^g)$ as in Hyb_0 and computes $\mathsf{com}(\tilde{w}_g)$. It gives $(\tilde{x}_1, \mathsf{com}(\tilde{w}_1)), \ldots,$
$(\tilde{x}_T, \mathsf{com}(\tilde{w}_T))$ as the instances and the openings to $(\mathsf{com}(\tilde{w}_1), \ldots, \mathsf{com}(\tilde{w}_T))$ as the witnesses to the challenger. The challenger responds with $(\pi_1, \ldots, \pi_N)$. It uses this to compute π as before. Observe that if the challenger generated the CRS and the proofs using the honest prover algorithm, then the output of the above reduction is identical to Hyb_0. Else, it is distributed identically to Hyb_1. Therefore, this reduction breaks the zero-knowledge property of the NIZK scheme and this is a contradiction.

- Hyb_2 : This hybrid is identical to the previous one except that we generate $(\mathsf{crs}_{N'}, \mathsf{td}_{N'}) \leftarrow \mathsf{Sim}_{N'}(1^{\lambda_2})$ and generate $\sigma \leftarrow \mathsf{Sim}_{N',2}(\mathsf{td}_{N'}, x')$ (instead of computing σ honestly).

  Claim 13. *Hybrids* Hyb_1 *and* Hyb_2 *are statistically indistinguishable given that the underlying NIZK for language* $\mathcal{L}'$ *is statistical zero-knowledge.*

  The proof of this claim is identical to the proof of Claim 12.

- Hyb_3 : This hybrid is identical to the previous one except that we set com to be $\mathsf{Com}_C(\mathsf{crs}_C, 0^m)$ and $\mathsf{com}(\tilde{w}_g)$ for each $g \in [T]$ as $\mathsf{Com}(\mathsf{crs}_C, \mathbf{0})$.

  Claim 14. *Hybrids* Hyb_2 *and* Hyb_3 *are statistically indistinguishable given that the underlying FLC is statistically hiding.*

  If Hyb_2 is distinguishable from Hyb_3 with non-negligible advantage, we show that this contradicts the statistical hiding of the commitment scheme. The reduction receives crs_C. It uses this to generate the CRS for the scheme. The reduction then gives $(w,\ 0^m)$ and $(\tilde{w}_g, \mathbf{0})$ for each $g \in [T]$ as the challenge tuples. The challenger responds with com (either committing to w or 0^m) and $\mathsf{com}(\tilde{w}_g)$ (either committing to $\tilde{w}_g$ or $\mathbf{0}$) for each $g \in [T]$. The reduction behaves exactly as in the previous hybrid to create a proof π. If com is generated as a commitment to w and $\mathsf{com}(\tilde{w}_g)$ is generated as a commitment to $\tilde{w}_g$ for each $g \in [T]$, then the output of the reduction is identical to Hyb_2. Else, it is distributed identically to Hyb_3. Therefore, the reduction breaks the hiding property of the commitment scheme with non-negligible advantage and this is a contradiction.

- Hyb_4 : This hybrid is identical to the previous one except that for each $g \in [T]$, we set init_g to be same as init_{g-1} except that the ℓ-th coordinate is set to a uniformly chosen random bit.

  Claim 15. *Hybrids* Hyb_3 *and* Hyb_4 *are statistically indistinguishable given that the underlying dual-mode SEH is statistically hiding.*

  If Hyb_3 is distinguishable from Hyb_4 with non-negligible advantage, we show that this contradicts the statistical hiding of the dual-mode SEH scheme. The reduction receives hk and the challenge v (either hashing the honestly generated init_g or a uniformly generated init'_g). Then it behaves exactly as in the previous hybrid to create a proof π and crs. If v is generated as a hashing of init_g, then the output of the reduction is identical to Hyb_3. Else, it is distributed identically to Hyb_4.

Note that the last hybrid corresponds to the simulation described above, and this concludes the proof of the theorem. □

Acknowledgments

We thank the anonymous reviewers for their comments and feedback.

Pedro Branco is supported by the European Research Council through an ERC Starting Grant (Grant agreement No. 101077455, ObfusQation). Akshayaram Srinivasan is supported in part by a NSERC Discovery Grant RGPIN-2024-03928. 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.

References

[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, Paper 2024/1806, 2024.

[ADD+22] Divesh Aggarwal, Nico Döttling, Jesko Dujmovic, Mohammad Hajiabadi, Giulio Malavolta, and Maciej Obremski. Algebraic Restriction Codes and Their Applications. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:15, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[BBDP22] Zvika Brakerski, Pedro Branco, Nico Döttling, and Sihang Pu. Batch-OT with optimal rate. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 157–186, Trondheim, Norway, May 30 – June 3, 2022. Springer, Heidelberg, Germany.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[BCD+24] Pedro Branco, Arka Rai Choudhuri, Nico Döttling, Abhishek Jain, Giulio Malavolta, and Akshayaram Srinivasan. Black-box non-interactive zero knowledge from vector trapdoor hash. Cryptology ePrint Archive, Paper 2024/1514, 2024.

[BD24] Shany Ben-David. Probabilistically checkable arguments for all np. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 345–374, Cham, 2024. Springer Nature Switzerland.

[BDGM19] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 407–437, Nuremberg, Germany, December 1–5, 2019. Springer, Heidelberg, Germany.

[BDS23] Pedro Branco, Nico Döttling, and Akshayaram Srinivasan. A framework for statistically sender private OT with optimal rate. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 548–576, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Heidelberg, Germany.

[BDSMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, 1991.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112, Chicago, IL, USA, May 2–4, 1988. ACM Press.

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 509–539, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

[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, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106, Palm Springs, CA, USA, October 22–25, 2011. IEEE Computer Society Press.

[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, *51st ACM STOC*, pages 1082–1090, Phoenix, AZ, USA, June 23–26, 2019. ACM Press.

[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, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Heidelberg, Germany.

[CGKS23] Matteo Campanelli, Chaya Ganesh, Hamidreza Khoshakhlagh, and Janno Siim. Impossibilities in succinct arguments: Black-box extraction and more. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *Progress in Cryptology - AFRICACRYPT 2023*, pages 465–489, Cham, 2023. Springer Nature Switzerland.

[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, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.

[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 16–20, 2021. Springer, Heidelberg, Germany.

[CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd FOCS*, pages 68–79, Denver, CO, USA, February 7–10, 2022. IEEE Computer Society Press.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd ACM STOC*, pages 542–552, New Orleans, LA, USA, May 6–8, 1991. ACM Press.

[DGI+19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 3–32, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[DGKV22] Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. Rate-1 non-interactive arguments for batch-NP and applications. In *63rd FOCS*, pages 1057–1068, Denver, CO, USA, October 31 – November 3, 2022. IEEE Computer Society Press.

[DJ01] Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 119–136, Cheju Island, South Korea, February 13–15, 2001. Springer, Heidelberg, Germany.

[DJJ24] Quang Dao, Aayush Jain, and Zhengzhong Jin. Non-interactive zero-knowledge from LPN and MQ. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IX*, volume 14928 of *Lecture Notes in Computer Science*, pages 321–360. Springer, 2024.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st FOCS*, pages 283–293, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.

[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, St. Louis, MO, USA, October 22–24, 1990. IEEE Computer Society Press.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

[GGI+15] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam D. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, 28(4):820–843, October 2015.

[GH98] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Inf. Process. Lett.*, 67(4):205–214, 1998.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304, Providence, RI, USA, May 6–8, 1985. ACM Press.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.

[GVW02] Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Comput. Complex.*, 11(1-2):1–53, 2002.

[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, Trondheim, Norway, May 30 – June 3, 2022. Springer, Heidelberg, Germany.

[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, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732, Victoria, BC, Canada, May 4–6, 1992. ACM Press.

[KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1545–1552. ACM, 2023.

[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, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[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, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437, Baltimore, MD, USA, May 14–16, 1990. ACM Press.

[Pas13] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 334–354, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany.

[PP22] Omer Paneth and Rafael Pass. Incrementally verifiable computation via rate-1 batch arguments. In *63rd FOCS*, pages 1045–1056, Denver, CO, USA, October 31 – November 3, 2022. IEEE Computer Society Press.

[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, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[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, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 399–410. ACM, 2024.

[WW22] Brent Waters and David J. Wu. Batch arguments for sfNP 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, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Heidelberg, Germany.

[WWW24] Brent Waters, Hoeteck Wee, and David J. Wu. New techniques for preimage sampling: Improved NIZKs and more from LWE. Cryptology ePrint Archive, Paper 2024/1401, 2024.