Non-interactive Distributional Indistinguishability (NIDI) and Non-Malleable Commitments

Dakshita Khurana[⋆]

Abstract. We introduce *non-interactive distributionally indistinguishable arguments* (NIDI) to address a significant weakness of NIWI proofs: namely, the lack of meaningful secrecy when proving statements about NP languages with unique witnesses.

NIDI arguments allow a prover $\mathcal{P}$ to send a single message to verifier $\mathcal{V}$, from which $\mathcal{V}$ obtains a sample d from a (secret) distribution $\mathcal{D}$, together with a proof of membership of d in an NP language $\mathcal{L}$.

The soundness guarantee is that if the sample d obtained by the verifier $\mathcal{V}$ is not in $\mathcal{L}$, then $\mathcal{V}$ outputs $\perp$. The privacy guarantee is that secrets about the distribution remain hidden: for every pair of (sufficiently) hard-to-distinguish distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ with support in NP language $\mathcal{L}$, a NIDI that outputs samples from $\mathcal{D}_0$ with proofs of membership in $\mathcal{L}$ is indistinguishable from one that outputs samples from $\mathcal{D}_1$ with proofs of membership in $\mathcal{L}$.

- We build NIDI arguments for superpolynomially hard-to-distinguish distributions, assuming sub-exponential indistinguishability obfuscation and sub-exponentially secure (variants of) one-way functions.
- We demonstrate preliminary applications of NIDI and of our techniques to obtaining the first (relaxed) non-interactive constructions in the plain model, from well-founded assumptions, of:
  - Commit-and-prove that provably hides the committed message
  - CCA-secure commitments against non-uniform adversaries.

  The commit phase of our commitment schemes consists of a single message from the committer to the receiver, followed by a randomized output by the receiver (that need not be returned to the committer).

1 Introduction

Can one non-interactively commit to a plaintext and prove that it satisfies a predicate (e.g., the plaintext is larger than 0*) while also ensuring that the plaintext is hidden?*

[⋆] Email: `dakshita@illinois.edu`. University of Illinois, Urbana-Champaign. Work done in part during a visit to the Simons institute, Berkeley. This material is based upon work supported in part by DARPA under Contract No. HR001120C0024. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

More generally, can a prover send a statement to a verifier and demonstrate that the statement is true without revealing secrets about it? An *interactive* solution to this problem can be obtained via the use of zero-knowledge proofs. These were first introduced in an influential work of Goldwasser, Micali and Rackoff [38], and it was subsequently shown that all languages in NP admit *interactive* ZK proofs [36]. An interactive proof is said to be zero-knowledge if there exist a simulator that can simulate the behavior of any verifier, without having access to the prover, in such a way that its output is indistinguishable from the output of the verifier after having interacted with an honest prover.

Understanding the round complexity of zero knowledge has been an important problem. In particular, zero-knowledge arguments for languages outside BPP, and without any trusted setup, are known to require at least three messages of interaction [37]. This leads to a natural question: *what meaningful relaxations of zero-knowledge are achievable non-interactively and without setup?*

Existing Relaxations of Zero-Knowledge. Towards addressing this question, several relaxations of zero-knowledge have been studied over the years.

- **Weak Zero-Knowledge [28]** relaxes zero-knowledge by switching the order of quantifiers. Specficially, weak zero-knowledge requires that for every verifier and every *distinguisher*, there exists a *distinguisher-dependent simulator* that fools this specific pair[1].
  Weak zero-knowledge is known to require at least two messages [37].
- **Witness Hiding [30]** loosely guarantees that a malicious verifier cannot recover a witness from a proof unless the witness can be efficiently computed from the statement alone.
- **Strong Witness Indistinguishability (Strong WI) [35]** requires that for two indistinguishable statement distributions $\mathcal{D}_0, \mathcal{D}_1$, a proof (or argument) for statement $d_0 \leftarrow \mathcal{D}_0$ must be indistinguishable from a proof (or argument) for statement $d_1 \leftarrow \mathcal{D}_1$.
- **Witness indistinguishability (WI) [30]** ensures that proofs of the *same statement* generated using different witnesses are indistinguishable. WI does not hold for statements sampled from different distributions, or statements that have a unique witness associated with them.

Two-message variants of weak zero-knowledge, witness hiding and strong WI have been obtained by [57, 47, 5, 25, 12]. But so far, the only relaxation known to be achievable *non-interactively* from well-studied assumptions, is witness indistinguishability. Non-interactive witness indistinguishable proofs (NIWIs) have been obtained by [8, 44, 15] under various assumptions. While NIWIs are quite natural and are useful as a building blocks in some applications, they are often quite limited. In (common) scenarios like committing to a secret message and proving a predicate about it – where statements being proven often have unique witnesses – the witness indistinguishability guarantee is meaningless.

[1] There are several variants of this definition strengthening/weakening different aspects [28, 22].

Commit-and-Prove. In a "commit-and-prove" protocol, a *prover* commits to (or encrypts) one or more messages, and would like to prove that the secret message(s) satisfy a predicate.

A simplification of the most basic privacy guarantee required in these applications is the following: for every pair of messages (m_0, m_1) that satisfy a (polynomial-time computable) predicate ϕ (i.e. $\phi(m_0) = \phi(m_1) = 1$), the following two distributions must be computationally indistinguishable:

$$\big(c_0 = \mathsf{Com}(m_0; r), \mathit{\Pi}_{c_0 \in \mathcal{L}_\phi}\big) \text{ and } \big(c_1 = \mathsf{Com}(m_1; r), \mathit{\Pi}_{c_1 \in \mathcal{L}_\phi}\big)$$

where Com denotes a perfectly binding commitment (or encryption), and $\mathit{\Pi}_{c \in \mathcal{L}_\phi}$ denotes a proof of the statement $c \in \mathcal{L}_\phi$ where

$$\mathcal{L}_\phi = \{c : \exists (m, r) \text{ such that } (c = \mathsf{Com}(m; r)) \wedge (\phi(m) = 1)\}.$$

In other words, any distributions $c_0 = \mathsf{Com}(m_0; r)$ and $c_1 = \mathsf{Com}(m_1; r)$ that are computationally indistinguishable, must remain indistinguishable even given proofs of membership in $\mathcal{L}_\phi$. Here ϕ is any efficiently computable predicate of the message, eg., $\phi(m) = 1$ if and only if $m > 10$.

The Insufficiency Of NIWIs. Because the statements in question clearly have unique witnesses, using NIWIs to generate the proof $\mathit{\Pi}_{c \in \mathcal{L}_\phi}$ does not guarantee that the secret message remains hidden. We note that the notion of *strong witness indistinguishability* would suffice, but whether strong WI can be achieved non-interactively remains an important open problem.

All known constructions [57, 47, 5, 25, 12] of *two-message* strong WI arguments follow variants of the common FLS [29] paradigm. Here, the prover provides a WI proof that:

"Either $x \in \mathcal{L}$ or the prover knows some trapdoor".

The trapdoor is designed to be hard for a (cheating) prover to compute, but easy for a simulator. Security is argued by having the simulator extract the secret trapdoor in polynomial or superpolynomial time, and use this trapdoor to generate the proof, instead of relying on a witness for x.

In settings where the verifier can send (at least) one message to the prover, the verifier's message can be used to set up a trapdoor, eg., by sampling $f(z)$ for a one-way permutation f and random trapdoor z [57]. The trapdoor z can be obtained by a simulator non-uniformly or in superpolynomial time (or even in polynomial time via specialized recent techniques [47, 25, 12]).

Establishing Trapdoors in the Non-Interactive Setting. In the non-interactive setting, since the verifier does not send any message to the prover, it becomes much more challenging to establish a trapdoor of the form described above, that is easy for a simulator to compute but not for a cheating prover.

Nevertheless, there have been exciting prior attempts. In particular, Barak and Pass [9] obtain variants of one-message zero-knowledge with nonuniform simulation and soundness against uniform provers. They rely on problems that are hard for uniform algorithms (eg., keyless collision-resistant hash functions) to set up a trapdoor that no *uniform prover* can obtain. Bitansky and Lin [13] propose a clever extension of this to the non-uniform setting by relying on problems that are hard for algorithms with a polynomial amount of non-uniformity. Assuming keyless collision-resistant hash functions with security against non-uniform adversaries, they obtain one-message zero-knowledge with superpolynomial simulation and *weak soundness against non-uniform provers.* They guarantee that the number of false statements a polynomial-time non-uniform prover can convince the verifier to accept is not much larger than its non-uniform advice.

In summary, known constructions of meaningful non-interactive secrecy-preserving arguments either (1) are not adequately sound and rely on non-standard hardness assumptions, or (2) do not provide meaningful secrecy, especially when considering statements with unique witnesses.

Bottlenecked Applications. The lack of non-interactive secrecy-preserving proofs for statements with unique witnesses has led to the need for non-standard assumptions in additional applications besides the example commit-and-prove scenario described above.

A prominent example are *non-interactive non-malleable commitments*: for which the only known constructions [56, 54, 13, 48, 31] either achieve non-standard forms of security or rely on relatively less standard assumptions like keyless collision resistant hashing with security against non-uniform adversaries. Eliminating non-standard assumptions appears to require appropriate non-interactive secrecy-preserving arguments, which were so far not known under well-founded assumptions. In the following section, we outline our contributions that aim to remedy this situation.

1.1 Our Results

We introduce and construct non-interactive distributional indistinguishable (NIDI) arguments without trusted setup from well-founded assumptions. These help overcome some of the drawbacks of existing non-interactive arguments, and enable applications like non-interactive commit-and-prove without trusted setup.

Non-Interactive Distributionally Indistinguishable (NIDI) Arguments. NIDI arguments enable a prover $\mathcal{P}$ with input a secret *efficiently sampleable* distribution $\mathcal{D}$ to send a single message (a "sampler") to verifier $\mathcal{V}$. Given this sampler, $\mathcal{V}$ can obtain a sample d from the (secret) distribution $\mathcal{D}$ *together with a proof of membership of the sampled instance d in a (public) NP language $\mathcal{L}$.* Specifically, after checking such a proof, the verifier either outputs $\perp$ or a sample d.[2]

In more detail, the prover algorithm $\mathcal{P}$ obtains input a security parameter, the description of a (secret) distribution $\mathcal{D}$, and a public NP language $\mathcal{L}$, and

[2] Jumping ahead, in our construction, a prover message will take the form of a program, to which the verifier will make a (randomized) query. In response, the program will output a sample d and a proof of membership of $d \in \mathcal{L}$.

generates $\mathcal{P}(1^\kappa, \mathcal{D}, \mathcal{L}) \rightarrow \pi$. The verifier $\mathcal{V}$ on input sampler π and the language $\mathcal{L}$ computes $\mathcal{V}(1^\kappa, \pi, \mathcal{L}) \rightarrow d$ or $\perp$.

- The **soundness** guarantee is that $\mathcal{V}$ does not output $d \notin \mathcal{L}$ (except with negligible probability). In other words, if the sample d obtained by $\mathcal{V}$ is not in $\mathcal{L}$, then the proof allows the verifier to detect this fact, and $\mathcal{V}$ outputs $\perp$ (except with negligible probability over the randomness of $\mathcal{V}$).
- The **secrecy** guarantee is that secrets in the distribution remain hidden from a malicious verifier: i.e., for every pair of (sufficiently) hard-to-distinguish distributions $\mathcal{D}_0 \approx \mathcal{D}_1$ where $\mathsf{Supp}(\mathcal{D}_0) \cup \mathsf{Supp}(\mathcal{D}_1) \in \mathcal{L}$,

$$\mathcal{P}(1^\kappa, \mathcal{D}_0, \mathcal{L}) \approx \mathcal{P}(1^\kappa, \mathcal{D}_1, \mathcal{L})$$

  Equivalently, a NIDI that outputs samples from $\mathcal{D}_0$ with proofs of membership in $\mathcal{L}$ is indistinguishable from one that outputs samples from $\mathcal{D}_1$ with proofs of membership in $\mathcal{L}$.

NIDI arguments bear a peripheral resemblance to, and are implied by (non-interactive) strong witness indistinguishable arguments, by simply having the prover on input $\mathcal{D}$ sample $d \leftarrow \mathcal{D}$ and attach a strong WI proof of membership of $d \in \mathcal{L}$. In particular, the secrecy guarantee of NIDI is similar in spirit to that of strong witness indistinguishable arguments. However, we do not know if non-interactive strong WI arguments exist under standard assumptions.

We note that the syntax/completeness properties of NIDI are different from strong WI: in the case of a strong WI proof system, the prover samples $d \leftarrow \mathcal{D}$ and attaches a proof that $d \in \mathcal{L}$. On the other hand, in the case of NIDI, the prover sends a "sampler" to $\mathcal{V}$, and the sample d (together with a proof) are obtained by $\mathcal{V}$ from this sampler. Therefore, while an honest prover knows the distribution $\mathcal{D}$, it may not know the exact value d that was sampled by a (randomized) $\mathcal{V}$.

Non-Interactive Distributionally Indistinguishable (NIDI) Arguments from Sub-exponential Indistinguishability Obfuscation. We rely on sub-exponential indistinguishability obfuscation and other standard assumptions to obtain NIDI arguments that satisfy the secrecy guarantee described above as long as the pair of distributions $(\mathcal{D}_0, \mathcal{D}_1)$ are *superpolynomially indistinguishable.*

Theorem 1. *(Informal) For every $p(\kappa) = \omega(\log \kappa)$ and every pair of distributions $\mathcal{D}_0, \mathcal{D}_1$ that cannot be distinguished with advantage better than $2^{-p(\kappa)}$ by any polynomial-sized adversary, NIDI arguments exist assuming sub-exponentially secure indistinguishability obfuscation and other standard assumptions.*

Application 1: Non-interactive Commit-and-Prove. A commit-and-prove argument is a protocol between a committer $\mathcal{C}$ and receiver $\mathcal{R}$. In the commit phase, the committer sends to the verifier a message that allows it to commit to a value $m \in \{0,1\}^\kappa$. It also proves that the committed value m satisfies a (public) efficiently computable predicate ϕ. Given the prover's message, the receiver outputs

$\perp$, or a string c. Later, $\mathcal{C}$ and $\mathcal{R}$ possibly engage in another decommit phase, at the end of which $\mathcal{R}$ outputs $\perp$ or $m \in \{0,1\}^{\kappa}$.

The soundness and secrecy guarantees are as expected:

- **Soundness** requires that if the verifier outputs a string c that is not $\perp$, then there does not exist an opening m' of c such that m' does not satisfy ϕ.
- **Secrecy** guarantees that the message m is hidden, i.e. for all pairs of (equal-sized) messages (m_0, m_1) that satisfy the predicate ϕ, $\mathcal{C}(1^k, m_0, \phi) \approx \mathcal{C}(1^k, m_1, \phi)$.

Theorem 2. *(Informal) Assuming sub-exponentially secure indistinguishability obfuscation and other standard assumptions, there exist commit-and-prove arguments in the plain model that satisfy a relaxed notion of non-interactivity.*

In our construction, the commitment phase consists of a committer sending the receiver a string (representing a program), but the actual commitment transcript is finalized only after the receiver produces an output (based on a randomized query to this program). While the commitment transcript is a deterministic function of the committer's message and the receiver's randomness, the receiver randomness/receiver query *may or may not* have to be known to the committer before or during the decommitment phase. If this randomness needs to be made explicit, then the commitment needs an extra message from the receiver. If it is not necessary to make the receiver randomness explicit, it becomes possible to achieve a truly non-interactive protocol.

For example, in two-party settings where one player establishes a secret trapdoor for use in a larger protocol, the extra message from the receiver may either be unnecessary (since it is not needed for decommitment) or could be clubbed together with other receiver messages. At the same time, there could be multi-party settings where the committer and receiver must agree to an entire commitment transcript before the protocol can proceed. For example, on a blockchain, one may want to commit to the value of a transaction and prove that the committed value is positive. Applying our non-interactive commit-and-prove naıvely to such a setting, without an explicit receiver message, could allow a malicious committer to trick different verifiers into recording different transactions (although each to a positive value).

Application 2: Non-interactive Non-malleable (CCA) Commitments. Very roughly, non-malleability prevents an adversary from modifying a commitment $\mathsf{com}(m)$ to generate a commitment $\mathsf{com}(m')$ to a value m' that is related to the original m. This is equivalent (assuming the existence of signatures/one-way functions) to a tag-based notion where the commit algorithm obtains an additional input, a $\mathsf{tag} \in \{0,1\}^{\kappa}$, and where the adversary is restricted to using a tag, or identity, that is different from the tag used to generate the honest commitment.

We consider a strong form of *non-malleability* for non-interactive commitments: CCA security [21]. Namely, we build commitments that hide the committed value even from an adversary which has access to an oracle that computes decommitments of arbitrary commitment strings that the adversary sends to this oracle, as long as they are different from the challenge string.

Theorem 3. *(Informal) CCA commitments for 2^κ tags satisfying a relaxed notion of non-interactivity exist assuming sub-exponentially secure indistinguishability obfuscation, CCA commitments for* $\log\log\log\kappa$ *tags and other standard assumptions.*

We note that CCA commitments for $\log\log\log\kappa$ tags can be based on either (1) sub-exponential time-lock puzzles (which can be based on sub-exponential indistinguishability obfuscation and the existence of sub-exponentially hard non-parallelizable languages [11]), or (2) sub-exponential hardness of discrete log and sub-exponential quantum hardness of LWE.

Just like the setting of commit-and-prove, the underlying "committed value" is defined as a function of the (non-interactive) message from the committer, and the receiver's randomness. However, again like the case of commit-and-prove, the receiver can remain *silent* throughout, thereby leading to a truly non-interactive protocol. In this setting, the CCA commitment guarantees that the value underlying a mauled commitment is independent of the honestly committed message, with overwhelming over the randomness of an honest receiver. Therefore this appears to achieve the conceptual objective of completely non-interactive commitments.

In addition, this notion would suffice for classic applications of non-malleable commitments like coin-flipping and auctions, with a non-interactive committer message and without the need for any additional messages from the receiver. An auction would be implemented by having all parties commit to their inputs using the CCA commitment, with just a single (broadcast) message from the committer. In the next round, all committers reveal *all the input and randomness* they used to generate their entire obfuscated program. These openings are accepted only if the honest committer strategy applied to the opened input and randomness results in the same obfuscated program that the committer sent; otherwise the protocol aborts. If the protocol does not abort, then the result of the protocol is computed on these opened values.

Finally, we remark that recent exciting progress [1, 45, 3, 2, 33, 34, 17] has led to constructions of indistinguishability obfuscation from simpler assumptions, including most recently [34, 18, 65] that obtain sub-exponentially secure iO from simple-to-state (circular security) assumptions on LWE-based cryptosystems and [46] that obtains iO from the following sub-exponential well-founded assumptions: SXDH, LWE, (a variant of) LPN and boolean PRGs in $\mathsf{NC0}$.

1.2 Additional Related Work

Relaxations of Zero-Knowledge. Subsequent to the introduction of weak zero-knowledge [28], three-message weak ZK and witness hiding were constructed by [14] from what are now considered implausible assumptions (due to [19, 10]). The work of [22] proved equivalence between different variants of weak zero-knowledge. Next, [47] constructed *distributional* weak-zero-knowledge and witness-hiding protocols for a restricted class of *non-adaptive verifiers* who choose their messages obliviously of the proven statement. They obtain protocols in three messages under standard assumptions, and in two messages under standard, but super-polynomial, assumptions. More recently, [12] obtained two-message weak-zero knowledge (which implies witness hiding and strong WI) in the standard setting via a new simulation technique, and concurrently [25] obtained two-message witness hiding from new assumptions. Even more recently, [51] gave best-possible/universal and non-uniform witness hiding arguments, as well as witness hiding proofs under assumptions on the non-existence of weak forms of witness encryption for certain languages. We note that witness hiding arguments provide a weaker one-wayness guarantee, and are insufficient to achieve, e.g., commit-and-prove with message hiding as discussed in the example in the introduction.

Zero knowledge with simulators that run in super-polynomial time is known in two messages from standard, but super-polynomial, assumptions [57, 5]. One-message ZK with super-polynomial simulation can be obtained against uniform provers, assuming uniform collision-resistant keyless hash functions [9], or against non-uniform verifiers, but with *weak soundness*, assuming multi-collision-resistant keyless hash functions [13]. As discussed earlier, these proofs satisfy weak notions of soundness against non-uniform provers (allowing non-uniform provers to cheat on certain instances). This is undesirable in many settings.

Non-Malleable Commitments. Minimizing the round complexity of non-malleable commitments has been an important research goal in cryptography. Prior work, namely [27, 6, 58, 59, 55, 56, 53, 64, 60, 52, 39, 40, 43, 41, 24, 23] culminated in three round non-malleable commitments from standard polynomial-time assumptions [42, 49] and two round commitments from sub-exponential assumptions like time-lock puzzles [54] and sub-exponential DDH/LWE/QR/NR [50].

However, achieving non-interactive non-malleable commitments from well-found assumptions has been particularly challenging. In the non-interactive setting, Pandey, Pass and Vaikuntanathan [56] first gave constructions of non-malleable commitments based on a strong non-falsifiable assumption ("adaptive" one-way functions). Recently Bitansky and Lin [13] obtained constructions of non-interactive non-malleable commitments from sub-exponential time-lock puzzles and keyless hash functions with (variants of) collision resistance against non-uniform adversaries. Additionally Kalai and Khurana [48] obtained constructions satisfying a weaker notion of non-malleability w.r.t. 'replacement' (essentially allowing selective-abort attacks) from well-studied assumptions including sub-exponential NIWIs, discrete log and the *quantum* hardness of LWE. Very recently Garg et. al. [31] improved upon [13], eliminating the need for NIWIs and making black-box use of cryptography. Despite this substantial progress, prior to this work, there were no known constructions of non-interactive (or relaxed non-interactive) non-malleable commitments from well-founded assumptions.

2 Technical Overview

We now walk the reader through our construction and offer additional insight into the notion of a NIDI. Our aim will be to find a meaningful privacy guarantee that *is achievable non-interactively*, and applicable widely. A "commit-and-prove" protocol as described in the introduction will serve as a canonical example of the type of applications that we would like to enable.

2.1 Commit-and-Prove Arguments

Outline: Compressing Interactive Commit-and-Prove via Obfuscation. Our first stab at constructing non-interactive commit-and-prove with meaningful secrecy is as follows: let us try to *compress* an interactive commit-and-prove protocol to a non-interactive one, as follows.

Let $(\mathsf{ICP}.\mathcal{P}, \mathsf{ICP}.\mathcal{V})$ denote the (honest) prover and verifier circuits for an appropriate *interactive* n-round commit-and-prove protocol ICP. The prover in the non-interactive system simply outputs obfuscations of the next-message functions of $\mathsf{ICP}.\mathcal{P}$, one obfuscation for each round. The prover's next-message function $\mathsf{ICP}.\mathcal{P}_j$ for round $j \in [n]$ of ICP depends on its inputs m, ϕ (i.e. the secret message and predicate), and randomness r – all of which are hardwired in the obfuscated circuits. This function on input the transcript through round $(j-1)$, produces as output the next message. The prover must output, for every round $j \in [n]$, the obfuscated circuit

$$\mathcal{C}_j = \mathsf{Obf}\left(\mathsf{ICP}.\mathcal{P}_j(m, \phi, r, \cdot)\right).$$

Given $(\mathcal{C}_1, \ldots, \mathcal{C}_n)$, $\mathcal{V}$ queries these circuits as if it were interacting with $\mathsf{ICP}.\mathcal{P}$, feeding them the current transcript and obtaining the next message. Finally, it accepts if $\mathsf{ICP}.\mathcal{V}$ would have accepted.

But obfuscating the next message function in this manner leads to new vulnerabilities that do not necessarily arise in the interactive setting. Unlike queries to an actual prover, an adversarial verifier can query obfuscated programs $(\mathcal{C}_1, \ldots, \mathcal{C}_n)$ out of order, and may even query them many times, amounting to "resetting" attacks [20]. Thus one would generally need to rely on *resettably zero-knowledge* protocols that satisfy security in the presence of resetting attacks [20].

Second, we note that general-purpose obfuscators satisfying the most natural notion of security (virtual-black-box) cannot exist [7]. We would therefore like to base security of the compressed protocol on the weaker notion of *indistinguishability obfuscation*, for which we know constructions under plausible assumptions (most recently due to [34, 18, 65, 46]).

Basing Security on Indistinguishability Obfuscation. Recall that we would like the compressed commit-and-prove argument to hide the committed m. This means that for every pair of values m_0, m_1 that satisfy a predicate ϕ, obfuscated next-message circuits that commit to m_0 and generate a proof of m_0 satisfying ϕ, should be indistinguishable from obfuscated circuits that generate a similar commit-and-prove argument for m_1.

Before going into further detail, we point out that the general paradigm of using obfuscation to compress interactive protocols has been explored in prior work, (eg., MPC protocols were compressed via obfuscating the next-message function in [32, 26, 4]). However in these works, the set of allowable or meaningful inputs to the program are small in number and are fixed apriori. This makes it possible to hardwire a few meaningful paths in the obfuscated programs and use such paths to argue security.

In our setting, the obfuscated next-message function must remain functional for (nearly) *all* verifier inputs. Because of this, our strategy to prove indistinguishability will iterate over *all possible* verifier inputs. To make this easier, we will begin by fixing a specific two-message interactive protocol, that will then be compressed to a non-interactive protocol via obfuscation.

Fixing an Interactive Protocol. To begin with, the interactive protocol that we rely on will be the following two-message protocols due to Pass [57].

- The interactive verifier $\mathsf{ICP}.\mathcal{V}$ samples a random α and outputs $f(\alpha)$, where f denotes a one-way function with "efficiently recognizable range" : where it is easy to efficiently check given y if there exists α such that $f(\alpha) = y$ (eg., this is true whenever f is a one-way permutation).
- Next, the prover $\mathsf{ICP}.\mathcal{P}$ generates a commitment c to m by means of any perfectly binding non-interactive commitment, and also a non-interactive commitment c' to 0. In addition, it sends a NIWI asserting that:

$$\text{“}\big(c \text{ is a commitment to } m \text{ such that } \phi(m) = 1\big) \\ \mathsf{OR}\ \big(c' \text{ is a commitment to } \alpha \text{ such that } f(\alpha) = y\big).\text{”}$$

To argue that this interactive protocol *hides* the value m, one can rely on a simulator that extracts α given y in superpolynomial time, and uses the second *trapdoor* statement to generate the NIWI. This makes it possible to rely on the hiding property of the non-interactive commitment and replace c with a commitment to a different message.

Arguing Security of the Compressed Commit-and-Prove System. Plugging this two-message argument into the template described above yields the following commit-and-prove protocol:

The non-interactive prover simply obfuscates a circuit that on input an arbitrary string y computes c, c' as commitments to m and 0 respectively, and as described above a NIWI asserting that:

$$\text{“}\big(c \text{ is a commitment to } m \text{ such that } \phi(m) = 1\big) \\ \mathsf{OR}\ \big(c' \text{ is a commitment to } \alpha \text{ such that } f(\alpha) = y\big).\text{”}$$

Arguing secrecy of the non-interactive protocol is somewhat more involved as one cannot hope to directly emulate the proof of secrecy of the interactive protocol. In particular, ideally one would like to replace the obfuscated circuit with a different one that has the superpolynomial simulator's code hardwired into it. In the next hybrid step one could hope to switch the commitment string c to commit to a different value. But this does not immediately work because of the inefficiency introduced by the simulator. In fact, even if we started out with a resettably-secure protocol with a polynomial simulator, it is completely unclear how to replace the next-message circuit with one that generates simulated proofs, unless the simulator is straight-line and black-box. Unfortunately straight-line black-box simulators cannot exist in the plain model without trusted setup, so we explore a different route as described below. In what follows, we will outline a concrete construction by building on the ideas and pitfalls discussed above.

Towards a Concrete Construction. The commit-and-prove algorithm $\mathcal{C}(1^k, m, \phi)$ samples a random key K for a *puncturable* PRF, and then outputs an indistinguishability obfuscation $\widetilde{P}$ of the program P described in Figure 1.

```
Hardwired: Puncturable PRF Key $K$, Message $m$, Predicate $\phi$.

Input: Query $y \in \{0,1\}^\kappa$.

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $(r_1, r_2, r_3) = \mathsf{PRF}(K, y)$.
3. Set $c = \mathsf{com}(m; r_1)$ and $c' = \mathsf{com}(0^\kappa; r_2)$.
4. Let $e$ be a NIWI, computed with randomness $r_3$, asserting that

          "$\big(c$ is a commitment to $m$ such that $\phi(m) = 1\big)$
          $\mathsf{OR}$ $\big(c'$ is a commitment to $\alpha$ such that $f(\alpha) = y\big)$."

5. Output $(c, c', e)$.
```

Fig. 1: **Program P.**

The receiver on input the obfuscated program $\widetilde{P}$ samples random α, sets $y = f(\alpha)$ and queries the program on y to obtain output some (c, c', e). It parses e as a NIWI and outputs $\perp$ if the NIWI does not verify, otherwise outputs c.

Message Hiding. Recall that we would like to establish that for all pairs of (equal-sized) messages (m_0, m_1) such that $\phi(m_0) = \phi(m_1) = 1$, $\mathcal{C}(1^\kappa, m_0, \phi) \approx \mathcal{C}(1^\kappa, m_1, \phi)$.

We will prove this by iterating over exponentially many hybrids, corresponding to all possible inputs to the obfuscated program. The j^{th} intermediate hybrid Hybrid_j for $j \in [0, 2^\kappa]$ will obfuscate a program $P^{(j)}$ that is identical to P except the following. On all inputs y such that $y < j$, $P^{(j)}$ sets $c = \mathsf{com}(m_1)$, and on all inputs y such that $y \geq j$, sets $c = \mathsf{com}(m_0)$. When defined this way, note that $\mathsf{Hybrid}_0 \equiv \mathcal{C}(1^\kappa, m_0, \phi)$ and $\mathsf{Hybrid}_1 \equiv \mathcal{C}(1^\kappa, m_1, \phi)$.

Let us now argue that for all $j \in [1, 2^\kappa]$, $\mathsf{Hybrid}_{j-1} \approx \mathsf{Hybrid}_j$. Note that the only difference between the two hybrids is the difference in behavior of programs $P^{(j-1)}$ and $P^{(j)}$ on input $y = j$. While $P^{(j-1)}$ on input $y = j$ outputs $\mathsf{com}(m_0)$, $P^{(j-1)}$ on input $y = j$ outputs $\mathsf{com}(m_1)$.

We rely on standard iO techniques to show that Hybrid_{j-1} and Hybrid_j are indistinguishable. This is done by first puncturing the key K on input $y = j$, then hardwiring uniform randomness corresponding to input j, and then relying on the hiding of the commitments c and c', as well as the witness indistinguishability of NIWI.

Since there are $\sim 2^{\kappa}$ hybrids, denoting (an upper bound on) the adversary's distinguishing advantage between any consecutive pair Hybrid_{j-1} and Hybrid_j by μ, the overall advantage between $\mathcal{C}(1^{\kappa}, m_0, \phi)$ and $\mathcal{C}(1^{\kappa}, m_1, \phi)$ can grow to $2^{\kappa} \cdot \mu$, which is not negligible unless $\mu = \frac{\mathsf{negl}(\kappa)}{2^{\kappa}}$.

Therefore, we ensure that μ is small enough by relying on subexponential assumptions. Specifically, we will assume the PRF, non-interactive commitment, and iO allow adversarial advantage to be at most $2^{-k^{\epsilon}}$ for some arbitrary small $0 < \epsilon < 1$ when executed with security parameter k. By setting $k = \kappa^{1/\epsilon}$, we will achieve the desired small μ.

Proving Soundness: A Subtle Malleability Problem. Recall also that we would like to ensure soundness, meaning that a malicious prover, by sending an arbitrary obfuscated program $\widetilde{P}$ to a verifier, should not be able to convince such a verifier to output a string c for which the underlying value m does not satisfy predicate ϕ.

Note that this is only possible if the verifier's query to $\widetilde{P}$ results in output (c, c') and a NIWI e for which verification accepts, and which asserts that:

$$\text{“}\big(c \text{ is a commitment to } m \text{ such that } \phi(m) = 1\big)$$
$$\mathsf{OR}\ \big(c' \text{ is a commitment to } \alpha \text{ such that } f(\alpha) = y\big).\text{”}$$

By soundness of the NIWI, if the verifier outputs c such that the underlying value m does not satisfy $\phi(m) = 1$, then (w.h.p.) it must be the case that

$$c' \text{ is a commitment to } \alpha \text{ such that } f(\alpha) = y.$$

To rule out this possibility, we would like to argue that it is impossible for a committer to efficiently compute $\mathsf{com}(\alpha)$ given $y = f(\alpha)$. A natural way to achieve this is via complexity leveraging: we could try setting the parameter of the commitment to be relatively small so that it is easy to extract the value α from commitment string c' in time T. At the same time, we could require f to be uninvertible in time T. This would ensure that any committer that efficiently computes $\mathsf{com}(\alpha)$ given $y = f(\alpha)$, would necessarily be contradicting uninvertibility of f against adversaries running in time T.

But this leads to a circularity: recall that we set the size of y to be κ bits, and for our hybrid argument to go through, we needed com to use a security parameter $k = \kappa^{1/\epsilon}$ for the commitment scheme com, such that the commitment scheme can be broken in time $T = 2^k$. But because the size of y is κ bits, f cannot be more than $2^{\kappa} \ll T$-secure. Therefore, our setting of parameters for the proof of secrecy directly contradicts the parameters needed for the proof of soundness described above.

To get around this issue, we replace the commitment scheme used to generate the commitment c' in our construction, with a perfectly correct *public-key encryption scheme*.

Specifically, the commit-and-prove protcol outputs a public key pk in addition to the obfuscated program. And instead of generating c' as a commitment to 0, c' is generated as an encryption of 0, with respect to pk. This enables a non-uniform proof of soundness.

Specifically, given $(pk, \widetilde{P})$ if the verifier outputs c such that the underlying value m does not satisfy $\phi(m) = 1$, then (w.h.p.) it must be the case that

$$c' \text{ is an encryption (w.r.t. } pk\text{) of } \alpha \text{ such that } f(\alpha) = y.$$

Now given pk, our reduction/proof of soundness will non-uniformly obtain the corresponding sk. Next, given any prover that on input y outputs c' as an encryption of $f^{-1}(y)$, this reduction will be able to use sk to decrypt c' and recover α. This will yield a contradiction to the uninvertibility of f, and therefore help us obtain a proof of soundness. We note that a similar technique was used in [16] to achieve soundness in the context of post-quantum interactive ZK arguments.

2.2 Non-Interactive Distributional Indistinguishability

A reader may have already observed that the technique discussed so far is more general: it need not be limited to commit-and-prove, and may be used to prove arbitrary statements about (indistinguishable) distributions.

We distill out a general formulation of this technique into what we call a NIDI argument. The construction of our NIDI argument follows an outline identical to that of our commit-and-prove system. Namely, the prover algorithm $\mathcal{P}(1^\kappa, \mathcal{D}, \mathcal{L})$ is given a secret efficiently sampleable distribution $\mathcal{D}$ and public language $\mathcal{L}$ with corresponding relation $\mathcal{R}_\mathcal{L}$. It outputs a public key pk and an indistinguishability obfuscation of a program P' that is very similar to the program P discussed above. The key difference is that the commitment c to value m in the functionality of the program P is replaced by a general sample d from distribution $\mathcal{D}$. This program is described in Figure 2. Secrecy and soundness of this program follow identically to the commit-and-prove argument.

2.3 Application: CCA Commitments

These techniques also yield (relaxed) non-interactive non-malleable commmitments: in fact, we achieve a strong form of non-malleability, i.e. CCA security.

We model CCA commitments as being associated with identities or tags, where the CCA adversary gets access to a decommitment oracle for all tags/identities different from its own. All non-malleable commitment schemes assign "tags" (or identities) to parties, and require non-malleability to hold whenever the adversary is trying to generate a commitment $\mathsf{CCACom}_{\widetilde{T}}$ w.r.t. a tag $\widetilde{T}$ that is different from the honest tag T. Existing constructions of non-interactive non-malleable commitments (1) develop a scheme for a small (constant) number of

```
**Hardwired:** Puncturable PRF Key $K$, Distribution $\mathcal{D}$, Language $\mathcal{L}$, Public key $pk$.

**Input:** Query $y \in \{0,1\}^{\kappa}$.

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $(r_1, r_2, r_3) = \mathsf{PRF}(K, y)$.
3. Set $d = \mathcal{D}(r_1)$ and $c' = \mathsf{Enc}_{pk}(0^{\kappa}; r_2)$.
4. Let $e$ be a NIWI, computed with randomness $r_3$, asserting that
       "$\big(d = \mathcal{D}(r)$ for some $\mathcal{D}$ and $r$ such that $\mathcal{R}_{\mathcal{L}}(d, \mathcal{D}, r) = 1\big)$
       $\mathsf{OR}$ $\big(c'$ is an encryption w.r.t. $pk$, of $\alpha$ such that $f(\alpha) = y\big)$."
5. Output $(d, c', e)$.
```

Fig. 2: **Program P'.**

tags, and then (2) recursively apply *tag amplification*, discussed below, several times until a scheme supporting (2^{λ}) tags is achieved – which corresponds to supporting every possible λ-bit identity that a participant can assume.

Outline of Existing Tag Amplification Techniques. Non-interactive CCA commitments that support a small space of tags can be bootstrapped into commitments for a larger space of tags by executing (a round optimized variant of) a tag encoding scheme first suggested by [27].

Given a large tag T (in $[2^n]$) where $n \leq \mathsf{poly}(\lambda)$, first encode T into n small tags $t_1, t_2, \ldots t_n$ each in $[2n]$, by setting each $t_i = (i||T_i)$ where T_i denotes the i^{th} bit of T. This encoding ensures that for any different large tags $T \neq \widetilde{T}$, there exists at least one index i such that $\widetilde{t}_i \notin \{t_1, t_2, \ldots t_n\}$, where $(\widetilde{t}_1, \widetilde{t}_2, \ldots \widetilde{t}_n)$ is an encoding of $\widetilde{T}$. Note that when $T \in [2^n]$, each of the small tags t will only be as large as $2n$. Now starting with a CCA commitment '$\mathsf{ComSmall}$' for tags in $[2n]$, a scheme CCACom for tags in $[2^n]$ can be obtained as follows:

To commit to a message m w.r.t. a tag T, set

$$\mathsf{CCACom}_T(m) = \big(\{c_i = \mathsf{ComSmall}_{t_i}(m)\}_{i \in [n]}, \mathit{\Pi}\big), \text{ where}$$

$\mathit{\Pi}$ is (an appropriate variant of a) zero-knowledge argument certifying that:

"All n commitments c_i are to the same message."

Analysis. Suppose the adversary used large tag $\widetilde{T} = (\widetilde{t}_1, \ldots, \widetilde{t}_n)$ and the honest party used tag $T = (t_1, \ldots, t_n)$. By the property of the encoding, for any two large tags $T \neq \widetilde{T}$, there exists at least one index i such that $\widetilde{t}_i \notin \{t_1, t_2, \ldots t_n\}$, where $(t_1, t_2, \ldots t_n)$ and $(\widetilde{t}_1, \widetilde{t}_2, \ldots \widetilde{t}_n)$ refer to encodings of T and $\widetilde{T}$ respectively. This means (due to non-malleability of $\mathsf{ComSmall}$) that the message committed by the adversary using tag $\widetilde{t}_i$ must be independent of the honest committer's input. By the soundness of ZK, the message committed by the adversary using each (small) tag $\widetilde{t}_1, \dots \widetilde{t}_n$ is identical, so independence of the one committed using $\widetilde{t}_i$ implies independence of them all! Loosely, it then suffices to argue that a message corresponding to *any* tag $\widetilde{t}_i$ is generated independently of the honest committer's message.

In some more detail, for the CCA attacker's j^{th} oracle decommitment query, we will focus on the index i_j such that the tag $\tilde{t}_{i_j} \not\in \{t^1_1, t^1_2, \dots t^1_n\}$. In the real interaction, by soundness of the ZK argument, the value committed by the attacker is identical to the value committed using $\tilde{t}_{i_j}$. This makes it possible to rely on CCA security of the value committed using $\tilde{t}_{i_j}$. We note that this method will need rely on a ZK argument that is secure against adversaries running in time T, where T is the time required to brute-force break the CCA commitment with $\tilde{t}_{i,j}$. This is because we will want to argue that the value committed using tag $\tilde{t}_{i_j}$ remains unchanged even when the challenge commitment is generated by simulating the underlying ZK argument.

Once the ZK argument in the challenge commitment is simulated, it becomes possible to switch all components of the challenge commitment one by one, while arguing CCA security w.r.t. the value committed by the adversary via tag $\tilde{t}_{i_j}$. This follows because of CCA security of the underlying commitment scheme for small tags.

The Zero-Knowledge Bottleneck. Unfortunately, this process makes cricital use of the zero-knowledge argument. Recall that ZK requires more than 2 rounds of interaction, which leads to a clear problem in the non-interactive setting. Existing methods to overcome this problem without interaction rely on special (weak) types of ZK – thus requiring non-standard assumptions [13], or achieving only weak forms of security [54, 48, 31]. In [54, 13], NIWIs are combined with a trapdoor statement to enable weak forms of NIZKs without setup: against uniform provers assuming keyless collision-resistant hash functions in [54], and a weak form of soundness against non-uniform provers under the non-standard assumption of keyless collision-resistant hash against *non-uniform* adversaries in [13]. In addition [48] use NIWIs without trapdoors, but only achieve weaker forms of non-malleability (that is, w.r.t. replacement). Even more recently, [31] replace NIWIs with hinting PRGs and remove the need for non-black-box use of cryptography. However, they also rely on keyless hash functions to set up "trapdoors" for equivocal commitments, thereby achieving only uniform security. In summary, due to the need for (variants of) non-interactive ZK, all known constructions achieving the standard notion of non-malleability w.r.t. commitment (or the stronger notion of CCA security) without trusted setup and against non-uniform adversaries end up having to rely on non-standard assumptions.

In fact by now, CCA commitments – *only* for constant (and slightly super-constant) tags – *are known* based on relatively mild assumptions, whereas tag amplification requires stronger assumptions. We now briefly describe the milder assumptions for schemes with slightly super-constant tags for completeness, before going back to discussing the tag amplification bottleneck.

Base Schemes. Three recent works [54, 13, 48] build non-interactive "base" schemes: i.e. non-malleable commitments for a tag/identity space of size $c \log \log \kappa$ for a specific constant $c > 0$, based on various hardness assumptions. This is achieved by relying on families of assumptions, each of which is harder than the other along some axis of hardness.

Lin, Pass and Soni [54] assume a sub-exponential variant of the hardness of time-lock puzzles. Bitansky and Lin [13] show that base commitments can also rely on sub-exponentially hard one-way functions that admit a strong form of hardness amplification (the assumption is stronger than what is currently known to be provable by known results on hardness amplification). Subsequently, Kalai and Khurana [48] showed that one can assume classically sub-exponentially hard but quantum easy one-way functions (which can be based, e.g., on sub-exponential hardness of DDH), and sub-exponentially quantum hard one-way functions (which can be based, e.g., on sub-exponential quantum hardness of LWE). As discussed above, we would like to enable an alternative tag amplification process.

Commit-and-Prove. Going back to the tag amplification process outlined above, one may observe that the type of statement being proved via ZK fits well into the "non-interactive commit-and-prove" paradigm. In particular, one may hope that it would suffice to replace the ZK argument Π with (an appropriate) commit-and-prove – which allows a committer to generate n commitments w.r.t. n different small tags, and give a (privacy-preserving) proof that all n strings commit to the same message. As such, by carefully relying on our non-interactive commit-and-prove discussed in Section 2.1, it seems like one should be able to achieve generic tag amplification.

In fact, our construction is roughly as expected at this point. The committer $\mathcal{C}$ on input a message m and tag T encoded as $\{t_1, \ldots, t_n\}$[3], outputs a public key pk, together with an obfuscation of the program P_{CCA} described in Figure 3.

The proof of security of the resulting CCA commitment for large tags relies on a delicate interplay of parameters between the CCA commitment and the zero-knowledge argument. Specifically, recall that the tag amplification method sketched out earlier requires the "strength" of zero-knowledge to be higher than the time needed to brute-force extract the committed value from the underlying CCA commitment for small tags. In our setting, this translates to carefully fine-tuning parameters so that the NIWI, PRF and public key encryption scheme are all secure against T-size adversaries, where T is the time needed to break (via brute-force) the underlying CCA commitment for small tags. This requirement for fine-tuned parameters requires us to "open the black-box" and give a monolithic proof of security. By contrast, our (regular) commit-and-prove system makes black-box use of the NIDI abstraction.

A Final Subtle Issue. We now point out one additional subtlety that we glossed over the in the overview so far. Existing base schemes [54, 13, 48] (for $O(\log \log \kappa)$

[3] In the main technical body, we use a somewhat more optimal encoding scheme due to [50], but we ignore this optimization for the purposes of this overview.

```
Hardwired: Puncturable PRF Key $K$, Message $m$, Tags $t_1, \dots, t_n$, Public key $pk$.

Input: Query $y \in \{0,1\}^{\kappa}$.

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $(r_1, r_2, \dots, r_{n+2}) = \mathsf{PRF}(K, y)$.
3. Set $c_i = \mathsf{ComSmall}(m; r_i)$ for all $i \in [n]$.
4. Set $c' = \mathsf{enc}_{pk}(0^{\kappa}; r_{n+1})$.
5. Let $e$ be a NIWI, computed with randomness $r_{n+2}$, asserting that
       "(There exist $m$ and $\{r_i\}_{i \in [n]}$ s.t. $\forall i \in [n], c_i = \mathsf{ComSmall}(m; r_i)$)
        $\mathsf{OR}$ $\big(c'$ is an encryption w.r.t. $pk$, of $\alpha$ such that $f(\alpha) = y\big)$."
6. Output $(\{c_i\}_{i \in [n]}, c', e)$.
```

Fig. 3: **Program P_{CCA}.**

tags) are only secure in a setting where the adversary is restricted to using the same tag in all its queries to the CCA decommitment oracle. Before performing our tag amplification process, we will need to remove this "same-tag" restriction.

We build on a technique proposed by [31] to eliminate this restriction. A CCA commitment scheme without the same-tag restriction, for tags in $[n]$ where $n \leq \mathsf{poly}(\kappa)$, can be obtained from a CCA commitment with the same tag restriction, via the following process: To commit w.r.t. tag $t \in [n]$, send commitments w.r.t. all tags in $[n]$ that are *not equal to* t. In more detail,

$$\mathsf{CCACom}_t(m) = (\{\mathsf{CCACom\text{-}same\text{-}tag}_i(m)\}_{i \in [n] \setminus \{t\}}, \varPi),$$

where $\varPi$ is (an appropriate variant of a) ZK argument certifying that

"All $n-1$ commitments c_i are to the same message."

Let us assume that the adversary's challenge commitment has tag t^*. This means that the challenge commitment *does not contain* the underlying commitment $\mathsf{CCACom\text{-}same\text{-}tag}$ w.r.t. tag t^*, and on the other hand, all the adversaries oracle decommitment queries *will contain* $\mathsf{CCACom\text{-}same\text{-}tag}$ w.r.t. tag t^*. This means that all decommitment queries that the adversary makes contain a commitment w.r.t. tag t^* that does not appear in the challenge commitment. This leads to an identical situation as the setting of tag amplification, and a very similar construction (and proof) helps bootstrap same-tag schemes for $n \leq \mathsf{poly}(\kappa)$ tags to those that do not have such a requirement.

In summary, our final CCA commitment is obtained by first bootstrapping "base" same-tag commitment schemes for small tags to remove the same-tag requirement, and then bootstrapping the resulting small tag commitment via the tag amplification process outlined above.

Organization. The rest of this paper is organized as follows. In Section 3 we set up notation and define building blocks. In Section 4 we define and construct NIDIs, in Section 5, we use NIDIs in a black-box way to obtain commit-and-prove, and finally in Section 6 we build CCA commitments.

3 Preliminaries

We rely on the standard notions of Turing machines and Boolean circuits.

- A polynomial-size circuit family $\mathcal{C}$ is a sequence of circuits $\mathcal{C} = \{C_\kappa\}_{\kappa\in\mathbb{N}}$, such that each circuit C_κ is of polynomial size $\kappa^{O(1)}$ and has $\kappa^{O(1)}$ input and output bits. We also consider probabilistic circuits that may toss random coins.
- We follow the standard habit of modeling any efficient adversary as a family of polynomial-size circuits. For an adversary $\mathcal{A}$ corresponding to a family of polynomial-size circuits $\{\mathcal{A}_\kappa\}_{\kappa\in\mathbb{N}}$, we omit the subscript κ, when it is clear from the context.
- A function $f : \mathbb{N} \to \mathbb{R}$ is $\mathsf{negl}(n)$ if $f(n) = n^{-\omega(1)}$.
- For random variables X, Y, and $0 < \mu < 1$, we write $X \approx_{T(\kappa)} Y$ if for all polynomial-sized circuits $\mathcal{A}$, there exists a negligible function μ such that for all κ,
$$\big|\Pr[\mathcal{A}(X) = 1] - \Pr[\mathcal{A}(Y) = 1]\big| \le \mu(T(\kappa)).$$
- We will use $d \leftarrow \mathcal{D}$ to denote a random sample from distribution $\mathcal{D}$. This will sometimes be denoted equivalently as $d = \mathcal{D}(r)$ for $r \leftarrow \{0,1\}^*$. Similarly, we will consider randomized algorithms that obtain inputs, and toss coins. We will use notation $t \leftarrow \mathcal{T}(m)$ to denote the output of randomized algorithm $\mathcal{T}$ on input m. Sometimes we will make the randomness of $\mathcal{T}$ explicit, in which case we will use notation $t = \mathcal{T}(m; r)$ for $r \leftarrow \{0,1\}^*$.

4 Non-Interactive Distributionally Indistinguishable (NIDI) Arguments

In this section, we define and construct NIDI arguments. As discussed earlier, NIDI arguments enable a prover $\mathcal{P}$ with input a secret *efficiently sampleable* distribution $\mathcal{D}$ to send a single message (a "sampler") to verifier $\mathcal{V}$. Given this sampler, $\mathcal{V}$ can obtain a sample d from the (secret) distribution $\mathcal{D}$ *together with a proof of membership of the sampled instance* d *in a (public) NP language* $\mathcal{L}$. Specifically, after checking such a proof, the verifier either outputs $\perp$ or a sample d from the distribution.

4.1 Definitions

In a NIDI, the prover algorithm $\mathcal{P}$ obtains input a security parameter, the description of a (secret) distribution $\mathcal{D}$, and a public NP language $\mathcal{L}$, and generates $\mathcal{P}(1^\kappa, \mathcal{D}, \mathcal{L}) \to \pi$. The verifier $\mathcal{V}$ on input sampler π and the language $\mathcal{L}$ computes $\mathcal{V}(1^\kappa, \pi, \mathcal{L}) \to d$ or $\perp$. We formally define this primitive below.

Definition 1 (Non-Interactive Distributionally-Indistinguishable (NIDI) Arguments). *A pair of PPT algorithms is $(\mathcal{P}, \mathcal{V})$ is a non-interactive distributionally-indistinguishable (NIDI) argument for* NP *language $\mathcal{L}$ with associated relation $R_\mathcal{L}$ if the non-interactive algorithms $\mathcal{P}(1^\kappa, \mathcal{D}, \mathcal{L})$ and $\mathcal{V}(1^\kappa, \pi, \mathcal{L})$*[4] *satisfy:*

- **Completeness:** *For every* $\mathsf{poly}(\lambda)$*-sampleable distribution*[5] $\mathcal{D} = (\mathcal{X}, \mathcal{W})$ *over instance-witness pairs in $R_\mathcal{L}$ such that* $\mathsf{Supp}(\mathcal{X}) \subseteq \mathcal{L}$,

$$\Big(\mathcal{V}(1^\kappa, \pi, \mathcal{L}) : \pi \in \mathsf{Supp}\left(\mathcal{P}(1^\kappa, \mathcal{D}, \mathcal{L})\right)\Big) \in \mathsf{Supp}(\mathcal{X}).$$

- **Soundness:** *For every ensemble of polynomial-length strings $\{\pi_\kappa\}_{\kappa \in \mathbb{N}}$ there exists a negligible function $\mu(\cdot)$ such that:*

$$\Pr_{x \leftarrow \mathcal{V}(1^\kappa, \pi, \mathcal{L})} \Big[\big(x \neq \perp\big) \wedge \big(x \notin \mathcal{L}\big)\Big] \leq \mu(\kappa)$$

- **Distributional Indistinguishability:** *For every* $\mathsf{poly}(\kappa)$*-sampleable pair of distributions $\mathcal{D}_0 = (\mathcal{X}_0, \mathcal{W}_0)$ and $\mathcal{D}_1 = (\mathcal{X}_1, \mathcal{W}_1)$ over instance-witness pairs in $R_\mathcal{L}$ where* $\mathsf{Supp}(\mathcal{X}_0) \cup \mathsf{Supp}(\mathcal{X}_1) \subseteq \mathcal{L}$*, and $\mathcal{X}_0 \approx_\kappa \mathcal{X}_1$,*

$$\mathcal{P}(1^\kappa, \mathcal{D}_0, \mathcal{L}) \approx_\kappa \mathcal{P}(1^\kappa, \mathcal{D}_1, \mathcal{L})$$

Definition 2 (NIDI Arguments for $T(\kappa)$-Hard Distributions). *A pair of PPT algorithms is $(\mathcal{P}, \mathcal{V})$ is a non-interactive distributionally-indistinguishable (NIDI) argument for $T(\kappa)$-hard distributions and* NP *language $\mathcal{L}$ with assocaited relation $\mathcal{R}_\mathcal{L}$ if the non-interactive algorithms $\mathcal{P}(1^\kappa, \mathcal{D}, \mathcal{L})$ and $\mathcal{V}(1^\kappa, \pi, \mathcal{L})$ satisfy the completeness and soundness properties from Definition 1, and additionally satisfy:*

- **Distributional Indistinguishability for $T(\kappa)$-Hard Distributions:** *For every* $\mathsf{poly}(\kappa)$*-sampleable pair of distributions $\mathcal{D}_0 = (\mathcal{X}_0, \mathcal{W}_0)$ and $\mathcal{D}_1 = (\mathcal{X}_1, \mathcal{W}_1)$ over instance-witness pairs in $R_\mathcal{L}$ where* $\mathsf{Supp}(\mathcal{X}_0) \cup \mathsf{Supp}(\mathcal{X}_1) \subseteq \mathcal{L}$*, and $\mathcal{X}_0 \approx_{T(\kappa)} \mathcal{X}_1$,*

$$\mathcal{P}(1^\kappa, \mathcal{D}_0, \mathcal{L}) \approx_\kappa \mathcal{P}(1^\kappa, \mathcal{D}_1, \mathcal{L})$$

[4] Since we define a NIDI for $\mathcal{L}$, it is not necessary to explicitly send $\mathcal{L}$ as input to $\mathcal{P}$ and $\mathcal{V}$ but we nevertheless write it this way for clarity.

[5] Here, we slightly abuse notation and use $\mathcal{D}$ to also denote a circuit that on input uniform randomness, outputs a sample from the distribution $\mathcal{D}$.

4.2 Construction and Analysis

We prove the following theorem.

Theorem 4. *Assuming the existence of sub-exponentially secure one-way functions with efficiently recognizable range and sub-exponentially secure indistinguishability obfuscation, there exists a constant $c > 1$ s.t. for $T(\kappa) = 2^{(\log \kappa)^c}$ there exist* NIDI *arguments for $T(\kappa)$-Hard Distributions satisfying Definition 2.*

To prove Theorem 4, we show that there exist NIDI arguments for $T(\kappa)$-hard distributions, where $\log T = (\log \kappa)^c$, and $c > 1$ is some constant. Our construction depends on T, and is as follows.

Construction 4.1. Let $\epsilon > 0$ be an arbitrarily small constant such that:

- There exists a sub-exponentially secure one-way function $f : \{0,1\}^{\mathsf{poly}(k)} \to \{0,1\}^{\mathsf{poly}(k)}$ with an *efficiently recognizable range*, i.e., given y there is an efficient algorithm to check whether there exists a value x such that $f(x) = y$. Note that permutations have this property, because every y is in the range of the permutation. We require that for security parameter k', this function is invertible with probability at most $\frac{1}{2^{(k')^\epsilon}}$ by machines of size $2^{(k')^\epsilon}$.
- There exists a *perfectly correct*, *sub-exponentially secure* public-key encryption scheme with key generation, encryption and decryption algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ that for security parameter 1^k satisfies 2^{k^ϵ}- $\mathsf{IND\text{-}CPA}$ security against (non-uniform) adversaries[6].
- There exists a *sub-exponentially secure* indistinguishability obfuscation scheme $(\mathsf{iO.Obf}, \mathsf{iO.Eval})$ that for security parameter 1^k satisfies 2^{k^ϵ}- security against (non-uniform) adversaries.
- There exists a *sub-exponentially secure* puncturable PRF that for security parameter 1^k satisfies 2^{k^ϵ}- security against (non-uniform) adversaries.
- There exist *sub-exponentially secure* NIWIs that for security parameter 1^k satisfy 2^{k^ϵ}- security against (non-uniform) adversaries.

Set $c = \frac{1}{\epsilon}$. We construct our non-interactive distributionally-indistinguishable (NIDI) argument below, where letting $\mathcal{R}_\mathcal{L}$ denote the relation corresponding to NP language $\mathcal{L}$ we define

$$\mathcal{L}_{\mathsf{NIWI}} = \Big\{(pk, d_x, c, y) : \exists (d_w, s, sk) \text{ s.t. } \big((d_x, d_w) \in \mathcal{R}_\mathcal{L}\big) \bigvee \big((pk, sk) \leftarrow \mathsf{KeyGen}(s) \wedge y = f(\mathsf{Dec}_{sk}(c))\big)\Big\}$$

- The prove algorithm $\mathcal{P}(1^\kappa, \mathcal{D}, \mathcal{L})$ does the following:
  - Set $k = 2^{(\log \kappa)^{c^2}}, k' = 2^{(\log \kappa)^c}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k, R \leftarrow \{0,1\}^k$.
  - Generate program $P_{pk,K,\mathcal{D},\mathcal{L}}$ defined in Figure 4.
  - Compute $\widetilde{P} = \mathsf{iO.Obf}(P_{pk,K,\mathcal{D},\mathcal{L}}; R)$.
  - Output $(pk, \widetilde{P})$.

```
**Hardwired:** Public key $pk$, Puncturable PRF Key $K$, Distribution $\mathcal{D}$,
Language $\mathcal{L}$.

**Input:** Query $y \in \{0,1\}^{k'}$.

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $(r_1, r_2, r_3) = \mathsf{PRF}(K, y)$.
3. Set $(d_x, d_w) = \mathcal{D}(r_1)$.
4. Set $c = \mathsf{Enc}_{pk}(0^{k'}; r_2)$.
5. Set $x = (pk, d_x, c, y)$, $w = (d_w, 0^{k'+k})$.
   Then compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_{\mathsf{NIWI}}; r_3)$.
6. Output $(x, e)$.
```

Fig. 4: **Program** $P_{pk,K,\mathcal{D},\mathcal{L}}$.

- The verify algorithm $\mathcal{V}(1^\kappa, \pi, \mathcal{L})$ on input a proof $\pi = (pk, \widetilde{P})$ does the following:
  - Sample $v \leftarrow \{0,1\}^{k'}$ and set $y = f(v)$.
  - Compute $\mathsf{out} = \mathsf{iO}.\mathsf{Eval}(\widetilde{P}, y)$. Parse $\mathsf{out} = (x, e)$ and parse $x = (pk, d, c, y)$.
  - If $\mathsf{NIWI}.\mathcal{V}(1^k, x, e, \mathcal{L}_{\mathsf{NIWI}})$ rejects, output $\perp$ and stop.
  - Else output d.

Lemma 1. *Construction 4.1 satisfies completeness according to Definition 1.*

Proof. The proof follows by observing that due to perfect correctness of iO, $\mathcal{V}(\pi, \mathcal{L})$ for $\pi = (pk, \widetilde{P})$ obtains (x, e) from $\widetilde{P}$, where $x = (pk, d, c, y)$. By perfect correctness of NIWI, $\mathcal{V}$ will output d with probability 1. Recall that $(d, \cdot) = \mathcal{D}(r_1)$ by construction, and therefore $d \in \mathsf{Supp}(\mathcal{X})$.

Lemma 2. *Under the assumptions in Theorem 4, construction 4.1 satisfies soundness according to Definition 1.*

Lemma 3. *Under the assumptions in Theorem 4, construction 4.1 satisfies distributional indistinguishability for $T(\kappa)$-hard distributions per Definition 2.*

The proofs of these lemmas appear in the full version but are omitted from this version due to lack of space.

5 Commit-and-Prove

A (relaxed) non-interactive commit-and-prove argument is a protocol between a committer $\mathcal{C}$ and receiver $\mathcal{R}$. In the commit phase, $\mathcal{C}$ sends $\mathcal{R}$ a single message to

[6] This can be based on sub-exponential indistinguishability obfuscation and sub-exponential one-way functions following [62].

commit to a value $m \in \{0,1\}^\kappa$. The transcript of the commitment is finalized as a function of the receiver's randomness and the committer's message, although the receiver does not need to return this randomness to the committer. It also proves that m satisfies some public predicate ϕ, in other words it proves that $\phi(m) = 1$. At the end of this phase, $\mathcal{R}$ either outputs $\perp$ (denoting that the commitment phase was rejected) or outputs a commitment string c.

Later, the parties $\mathcal{C}$ and $\mathcal{R}$ possibly engage in another decommit phase, at the end of which $\mathcal{R}$ outputs $\perp$ or $m \in \{0,1\}^\kappa$.

Definition 3 (Non-Interactive Commit-and-Prove). *A pair of PPT algorithms $(\mathcal{C}, \mathcal{R})$ where $\mathcal{R} = (\mathcal{R}_1, \mathcal{R}_2)$ is a non-interactive commit-and-prove argument if it satisfies the following.*

- **Completeness:** *For every ϕ and every $m \in \{0,1\}^\kappa$ such that $\phi(m) = 1$,*

$$\Pr\left[\begin{array}{l} m \leftarrow \mathcal{R}_2(1^\kappa, c, \mathsf{cert}, \mathsf{st}) \quad \wedge \\ \phi(m) = 1 \end{array}\middle| \begin{array}{l} (\pi, \mathsf{st}) \leftarrow \mathcal{C}(1^\kappa, m, \phi) \\ (c, \mathsf{cert}) \leftarrow \mathcal{R}_1(1^\kappa, \pi, \phi) \end{array}\right] = 1.$$

- **Soundness:** *For every $\mathsf{poly}(\kappa)$-sized (non-uniform) committer $\mathcal{C}^*$ there exists a negligible function $\mu(\cdot)$ such that for large enough $\kappa \in \mathbb{N}$,*

$$\Pr\left[\begin{array}{l} \exists(m^*, \mathsf{st}^*) \ s.t. \ (m^* \neq \perp) \quad \wedge \\ m^* \leftarrow \mathcal{R}_2(1^\kappa, c, \mathsf{cert}, \mathsf{st}^*) \quad \wedge \\ \phi(m^*) \neq 1 \end{array}\middle| \begin{array}{l} \pi \leftarrow \mathcal{C}^* \\ (c, \mathsf{cert}) \leftarrow \mathcal{R}_1(1^\kappa, \pi, \phi) \end{array}\right] \leq \mu(\kappa).$$

- **Computational Hiding:** *For every language $\mathcal{L}$, every pair of messages (m_0, m_1) such that $\phi(m_0) = \phi(m_1) = 1$,*

$$\mathcal{C}(1^\kappa, m_0, \phi) \approx_\kappa \mathcal{C}(1^\kappa, m_1, \phi)$$

Construction 5.1. Let $\epsilon > 0$ be a constant such that:

- There exists a non-interactive perfectly binding commitment Com that satisfies hiding against 2^{κ^ϵ}-time (non-uniform) adversaries, and
- There exists a NIDI argument for 2^{κ^ϵ}-hard distributions that satisfies Definition 1.

We define

$$\mathcal{L}_\phi = \big\{c : \exists(m, r) \text{ s.t. } c = \mathsf{Com}(m; r) \wedge \phi(m) = 1\big\}$$

- The commit algorithm $\mathcal{C}(1^\kappa, m, \phi)$ does the following:
  - Define distribution $\mathcal{D}_m(r) = \mathsf{Com}(m; r)$.
  - Output $\pi = \mathcal{P}(1^\kappa, \mathcal{D}_m, \mathcal{L}_\phi)$ computed using uniform randomness $r_\mathcal{C}$.
  - Set $\mathsf{st} = (m, r_\mathcal{C})$.
- The receiver algorithm $\mathcal{R}_1(1^\kappa, \pi, \phi)$ does the following.
  - Sample randomness $r_\mathcal{R}$.
  - Obtain $y \leftarrow \mathcal{V}(1^\kappa, \pi, \mathcal{L}_\phi; r_\mathcal{R})$.

- Output $(y, r_{\mathcal{R}})$.

– The receiver algorithm $\mathcal{R}_2(1^\kappa, c, \mathsf{cert}, \mathsf{st}^*)$ does the following:
  - Parse $\mathsf{st}^* = (m^*, r^*_{\mathcal{C}})$ and $\mathsf{cert} = r_{\mathcal{R}}$.
  - Compute $\pi^* = \mathcal{P}(1^\kappa, \mathcal{D}_{m^*}, \mathcal{L}_\phi; r^*_{\mathcal{C}})$.
  - If $\mathcal{V}(1^\kappa, \pi^*, \mathcal{L}_\phi; r_{\mathcal{R}}) = (c, \cdot)$, output m^*.
  - Otherwise, output $\perp$.

Lemma 4. *Construction 5.1 satisfies completeness according to Definition 3.*

Proof. The proof follows by the perfect correctness of NIDI.

Lemma 5. *Construction 5.1 satisfies soundness according to Definition 3.*

Proof. We prove that this lemma follows by the soundness of the NIDI according to Definition 2 and the perfect binding property of Com. Towards a contradiction, suppose there exists a $\mathsf{poly}(\kappa)$-sized (non-uniform) committer $\mathcal{C}^*$ for which there exists a polynomial $p(\cdot)$ such that for infinitely many $\kappa \in \mathbb{N}$,

$$\Pr\left[\begin{array}{l} \exists (m^*, \mathsf{st}^*) \text{ s.t. } (m^* \neq \perp) \quad \wedge \\ m^* \leftarrow \mathcal{R}_2(1^\kappa, c, \mathsf{cert}, \mathsf{st}^*) \quad \wedge \\ \phi(m^*) \neq 1 \end{array} \middle| \begin{array}{l} \pi \leftarrow \mathcal{C}^* \\ (c, \mathsf{cert}) \leftarrow \mathcal{R}_1(1^\kappa, \pi, \phi) \end{array}\right] \geq \frac{1}{p(\kappa)}.$$

Fix any string π, and let $(c, \mathsf{cert}) \leftarrow \mathcal{R}_1(1^\kappa, \pi, \phi)$.

– By construction, for any st^* parsed as $(m^*, r^*_{\mathcal{C}})$, $\mathcal{R}_2(1^\kappa, c, \mathsf{cert}, \mathsf{st}^*)$ outputs $m^* \neq \perp$ if and only if for $\pi^* = \mathcal{P}(1^\kappa, \mathcal{D}_{m^*}, \mathcal{L}_\phi; r^*_{\mathcal{C}})$, $\mathcal{V}(1^\kappa, \pi^*, \mathcal{L}_\phi; \mathsf{cert}) = (c, \cdot)$. By perfect completeness of NIDI, this implies that $\mathcal{R}_2(1^\kappa, c, \mathsf{cert}, \mathsf{st}^*)$ outputs some $m^* \neq \perp$ if and only if there exists $r^*_{\mathcal{C}}$ such that $c = \mathsf{Com}(m^*; r^*_{\mathcal{C}})$.

– Next by the perfect binding of Com, for every string c, there exists at most one message m^* and randomness $r^*_{\mathcal{C}}$ such that $c = \mathsf{Com}(m^*; r^*_{\mathcal{C}})$. Then $\phi(m^*) \neq 1 \iff c \notin \mathcal{L}_\phi$.

Taken together, this implies that

$$\Pr\left[\left(\mathcal{R}(1^\kappa, \pi, \mathcal{L}_\phi) \neq \perp\right) \wedge \left(\mathcal{R}(1^\kappa, \pi, \mathcal{L}_\phi) \notin \mathcal{L}\right) \middle| \pi \leftarrow \mathcal{C}^*\right] \geq \frac{1}{p(\kappa)},$$

which contradicts the soundness of NIDI, as desired.

Lemma 6. *Construction 5.1 satisfies computational hiding according to Definition 2.*

Proof. This lemma follows almost immediately from the distributional indistinguishability of NIDI.

Specifically, for language $\mathcal{L} = \mathcal{L}_\phi$, for any pair of messages m_0, m_1 such that $\phi(m_0) = \phi(m_1) = 1$, define $\mathsf{poly}(\kappa)$-sampleable distributions $(\mathcal{D}_{m_0}, \mathcal{D}_{m_1})$ where $\mathcal{D}_{m_b} = (\mathsf{Com}(m_b; r), (m_b, r))$.

By definition of $\mathcal{L}_\phi$, $\mathsf{Supp}(\mathcal{D}_0) \cup \mathsf{Supp}(\mathcal{D}_0) \subseteq \mathcal{L}_\phi$. Moreover by 2^{κ^ϵ}-hardness of Com, we have that $\mathsf{Com}(m_0; r) \approx_{2^{\kappa^\epsilon}} \mathsf{Com}(m_1; r)$, Therefore, distributional indistinguishability of NIDI according to Definition 2 implies that: $\mathcal{P}(1^\kappa, \mathcal{D}_{m_0}, \mathcal{L}_\phi) \approx_\kappa \mathcal{P}(1^\kappa, \mathcal{D}_{m_1}, \mathcal{L}_\phi)$ or equivalently, $\mathcal{C}(1^\kappa, m_0, \phi) \approx_\kappa \mathcal{C}(1^\kappa, m_1, \phi)$, as desired.

6 CCA Commitments from Indistinguishability Obfuscation

In this section, we prove the following theorem.

Theorem 5. *Assume the existence of sub-exponentially secure indistinguishability obfuscation, sub-exponentially secure one-way functions with efficiently recognizable range and sub-exponentially secure CCA commitments for tags in* $[\log\log\log\kappa]$*. Then there exist CCA commitments for tags in* 2^κ*.*

We prove this theorem by building a tag amplification compiler that amplifies CCA commitments for tags in $[t/2]$ for $t \leq \mathsf{poly}(\kappa)$ to tags in $[T]$ where $T = \binom{t}{t/2}$. Applying this compiler 4 times to a CCA commitments for tags in $[\log\log\log\kappa]$ yields the statement of the theorem.

In what follows, let $\epsilon > 0$ be an arbitrarily small constant such that:

- The CCA commitment for small tags and security parameter κ is $2^{(\log\kappa)^{1/\epsilon}}$ secure and has a "brute-force" value algorithm CCAVal that recovers the value underlying any commitment, and runs in time at most $\mathsf{poly}(2^\kappa)$.
- There exists a *subexponentially secure one-way function* f that with security parameter k is 2^{k^ϵ} one-way. Furthermore, f has an *efficiently recognizable range*, i.e., given y there is an efficient algorithm to check whether there exists a value x such that $f(x) = y$. Note that permutations have this property, because every y is in the range of a permutation.
- There exists a *perfectly correct*, *sub-exponentially secure* public-key encryption scheme with key generation, encryption and decryption algorithms ($\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$) that for security parameter 1^k satisfies 2^{k^ϵ}- $\mathsf{IND\text{-}CPA}$ security against (non-uniform) adversaries.
- There exists a *sub-exponentially secure* indistinguishability obfuscation scheme ($\mathsf{iO.Obf}, \mathsf{iO.Eval}$) that for security parameter 1^k satisfies 2^{k^ϵ}- security against (non-uniform) adversaries.
- There exists a *sub-exponentially secure* puncturable PRF that for security parameter 1^k satisfies 2^{k^ϵ}- security against (non-uniform) adversaries.
- There exist *sub-exponentially secure* NIWIs that for security parameter 1^k satisfy 2^{k^ϵ}- security against (non-uniform) adversaries.

Our compiler is described formally below, where letting $\mathcal{R}_\mathcal{L}$ denote the relation corresponding to NP language $\mathcal{L}$ we define language

$$\begin{aligned}\mathcal{L}_{\mathsf{NIWI}} = \Big\{ &\{(c_i, s_i)\}_{i\in[t/2]}, (pk, \mathsf{enc}, y) : \exists (M, r_1, \ldots, r_{t/2}, s, sk) \text{ s.t.} \\ &\Big(\forall i \in [t/2], c_i = \mathsf{ComSmall}_{s_i}(M; r_i)\Big) \\ &\bigvee \Big((pk, sk) \leftarrow \mathsf{KeyGen}(s) \wedge y = f(\mathsf{Dec}_{sk}(c))\Big)\Big\}\end{aligned}$$

where s_i denotes a tag in $[t/2]$, and $\mathsf{ComSmall}$ denotes the commit algorithm for an underlying CCA commitment with tags in $[t/2]$.

Construction 6.1. We now describe the CCACom and CCAVal algorithms for the scheme with large tags. We note that just like our commit-and-prove system described in the previous section, the commit phase ends *after* the receiver has queried the committer's program on a random input. The output of the commit phase is the output of such a receiver (and depending on the application, the receiver may or may not need to send its input back to the committer).

On input security parameter κ, we will set parameters of our building blocks as follows. Our one-way function with efficiently recognizable range and sub-exponential security will have security parameter k_{f} set to $(\log \kappa)^{1/\epsilon}$. The CCA commitment for small tags will have security parameter set to κ. Note that this implies (by assumption) that CCAVal runs in time $\mathsf{poly}(2^{\kappa})$. Finally, all other primitives including iO, the puncturable PRF and the PKE scheme will have security parameter set to $k = \kappa^{\frac{1}{\epsilon}}$.

The CCACom Algorithm: $\mathsf{CCACom}(1^{\kappa}, m, \mathsf{tag})$ does the following.

- Let $\mathbb{T}$ denote the ordered set of all possible subsets of $[t]$, of size $t/2$. Pick the i^{th} element in set $\mathbb{T}$, for $i = \mathsf{tag}$.[7] Let this element be denoted by $(s_1, \ldots s_{t/2})$.
- The committer $\mathcal{C}(1^{\kappa}, M, \mathsf{tag})$ does the following:
  - Set $k = \kappa^{\frac{1}{\epsilon}}$, and $k_{\mathsf{f}} = (\log \kappa)^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Generate program $P_{pk,K,M,\mathsf{tag}}$ defined in Figure 5.
  - Compute $\widetilde{P} = \mathsf{iO}(P_{pk,K,M,\mathsf{tag}}; R)$.
  - Output $c = (\mathsf{tag}, pk, \widetilde{P})$.
- The receiver $\mathcal{R}$ on input a commitment $c = (\mathsf{tag}, pk, \widetilde{P})$ does the following.
  - Sample $v \leftarrow \{0,1\}^{\kappa}$ and set $y = f(v)$.
  - Compute $\mathsf{out} = \mathsf{iO}.\mathsf{Eval}(\widetilde{P}, y)$. Parse $\mathsf{out} = (x, e)$, $x = (d, pk\mathsf{enc}, y)$ and $d = \{c_i\}_{i \in [t/2]}$.
  - Set $x' = \{(c_i, s_i)\}_{i \in [t/2]}, (pk, \mathsf{enc}, y)$. If $\mathsf{NIWI}.\mathcal{V}(1^k, x', e, \mathcal{L}_{\mathsf{NIWI}})$ rejects, output $\perp$ and stop.
  - Else output v, and for each $i \in [t/2]$, execute the receiver algorithm $\mathsf{ComSmall}.\mathcal{R}(c_i)$.

    If any of these $(t/2)$ algorithms output $\perp$, then output $\perp$ and stop.
  - At the end of this process, the receiver either outputs $\perp$ or $(\tau_1, \ldots, \tau_{t/2})$ where τ_i denotes the (non-$\perp$) outcome of $\mathsf{ComSmall}.\mathcal{R}(c_i)$[8].

The CCAVal Algorithm: The CCAVal algorithm obtains as input a commitment string parsed as $\perp$ or $(\tau_1, \ldots, \tau_{t/2})$, generated as the output of the commit phase above, and does the following.

[7] Here, we use a different tag encoding scheme due to [50] that offers a slightly more optimized way to the same effect as the DDN encoding [27] discussed in the overview. That is, for every pair of unequal large tags T and T', there is at least one member in the set corresponding to T that is not present in the set corresponding to T', and vice-versa.

[8] Note that for the base scheme, $\mathcal{R}$ simply outputs the string it obtained from the committer.

– On input a commitment string, if $\perp$, output $\perp$. Otherwise parse the string as $(\tau_1, \dots, \tau_{t/2})$ and execute $\mathsf{ComSmall.CCAVal}(\tau_1)$.

```
Hardwired: Public key $pk$, Puncturable PRF Key $K$, message $M \in \{0,1\}^p$, small tags $(s_1, \dots s_{t/2})$ corresponding to $\mathsf{tag}$.

Input: Query $y \in \{0,1\}^{k_\mathsf{f}}$.

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $r = (r_1||r_2|| \dots ||r_{t/2}||r_{t/2+2}) = \mathsf{PRF}(K, y)$.
3. For $i \in [t/2]$, set $c_i = \mathsf{ComSmall}_{s_i}(M; r_i)$. Set $d = \{c_i\}_{i \in [t/2]}$.
4. Set $\mathsf{enc} = \mathsf{Enc}_{pk}(0^\kappa; r_{t/2+1})$.
5. Set $x = d, (pk, \mathsf{enc}, y), w = (M, r_1, \dots, r_{t/2}, 0^{2k})$.
6. Compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+2})$ and output $(x, e)$.
```

Fig. 5: **Program** $P_{K,M,\mathsf{tag}}$

We prove the security of this construction, and discuss how to eliminate the same-tag restriction in the full version of the paper.

Acknowledgments

We thank the anonymous Eurocrypt reviewers for their insightful suggestions. We are also grateful to Ran Canetti, Suvradip Chakraborty, Oxana Poburinnaya and Manoj Prabhakaran for useful discussions.

References

1. Agrawal, S.: Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. In: EUROCRYPT (2019)
2. Agrawal, S., Pellet-Mary, A.: Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE. In: EUROCRYPT (2020)
3. Ananth, P., Jain, A., Lin, H., Matt, C., Sahai, A.: Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In: CRYPTO (2019)
4. Ananth, P., Jain, A., Naor, M., Sahai, A., Yogev, E.: Universal constructions and robust combiners for indistinguishability obfuscation and witness encryption. In: Advances in Cryptology - CRYPTO 2016, Proceedings, Part II. pp. 491–520 (2016). https://doi.org/10.1007/978-3-662-53008-5_17
5. Badrinarayanan, S., Garg, S., Ishai, Y., Sahai, A., Wadia, A.: Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and

Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III. pp. 275–303 (2017). https://doi.org/10.1007/978-3-319-70700-6_10, https://doi.org/10.1007/978-3-319-70700-6_10

6. Barak, B.: Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. In: FOCS 2002. pp. 345–355 (2002)
7. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. J. ACM **59**(2), 6:1–6:48 (2012). https://doi.org/10.1145/2160158.2160159, https://doi.org/10.1145/2160158.2160159
8. Barak, B., Ong, S.J., Vadhan, S.P.: Derandomization in cryptography. SIAM J. Comput. **37**(2), 380–400 (2007). https://doi.org/10.1137/050641958, https://doi.org/10.1137/050641958
9. Barak, B., Pass, R.: On the possibility of one-message weak zero-knowledge. In: Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings. pp. 121–132 (2004). https://doi.org/10.1007/978-3-540-24638-1_7, https://doi.org/10.1007/978-3-540-24638-1_7
10. Bellare, M., Stepanovs, I., Tessaro, S.: Contention in cryptoland: Obfuscation, leakage and UCE. In: Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II. pp. 542–564 (2016). https://doi.org/10.1007/978-3-662-49099-0_20, https://doi.org/10.1007/978-3-662-49099-0_20
11. Bitansky, N., Goldwasser, S., Jain, A., Paneth, O., Vaikuntanathan, V., Waters, B.: Time-lock puzzles from randomized encodings. In: Sudan, M. (ed.) Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016. pp. 345–356. ACM (2016). https://doi.org/10.1145/2840728.2840745, https://doi.org/10.1145/2840728.2840745
12. Bitansky, N., Khurana, D., Paneth, O.: Weak zero-knowledge beyond the black-box barrier. In: Charikar, M., Cohen, E. (eds.) STOC 2019. pp. 1091–1102. ACM (2019). https://doi.org/10.1145/3313276.3316382, https://doi.org/10.1145/3313276.3316382
13. Bitansky, N., Lin, H.: One-message zero knowledge and non-malleable commitments. In: Theory of Cryptography Conference, TCC 2018, Goa, India, November 11-14, 2018, Proceedings (2018)
14. Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge. In: TCC 2012. pp. 190–208 (2012). https://doi.org/10.1007/978-3-642-28914-9_11, https://doi.org/10.1007/978-3-642-28914-9_11
15. Bitansky, N., Paneth, O.: Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. Lecture Notes in Computer Science, vol. 9015, pp. 401–427. Springer (2015). https://doi.org/10.1007/978-3-662-46497-7_16, https://doi.org/10.1007/978-3-662-46497-7_16
16. Bitansky, N., Shmueli, O.: Post-quantum zero knowledge in constant rounds. In: Makarychev, K., Makarychev, Y., Tulsiani, M., Kamath, G., Chuzhoy, J. (eds.) STOC 2020. pp. 269–279. ACM (2020). https://doi.org/10.1145/3357713.3384324
17. Brakerski, Z., Döttling, N., Garg, S., Malavolta, G.: Candidate io from homomorphic encryption schemes. In: EUROCRYPT (2020)
18. Brakerski, Z., Döttling, N., Garg, S., Malavolta, G.: Factoring and pairings are not necessary for io: Circular-secure LWE suffices. IACR Cryptol. ePrint Arch. (2020), https://eprint.iacr.org/2020/1024

19. Brzuska, C., Mittelbach, A.: Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II. pp. 142–161 (2014). https://doi.org/10.1007/978-3-662-45608-8_8, https://doi.org/10.1007/978-3-662-45608-8_8
20. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: Yao, F.F., Luks, E.M. (eds.) Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA. pp. 235–244. ACM (2000). https://doi.org/10.1145/335305.335334, https://doi.org/10.1145/335305.335334
21. Canetti, R., Lin, H., Pass, R.: Adaptive Hardness and Composable Security in the Plain Model from Standard Assumptions. In: Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science. pp. 541–550. FOCS '10 (2010)
22. Chung, K., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I. pp. 66–92 (2015). https://doi.org/10.1007/978-3-662-46494-6_4, https://doi.org/10.1007/978-3-662-46494-6_4
23. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Concurrent non-malleable commitments (and more) in 3 rounds. In: Robshaw and Katz [61], pp. 270–299. https://doi.org/10.1007/978-3-662-53015-3, https://doi.org/10.1007/978-3-662-53015-3
24. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Four-round concurrent non-malleable commitments from one-way functions. In: Annual International Cryptology Conference. pp. 127–157. Springer (2017)
25. Deshpande, A., Kalai, Y.: Proofs of ignorance and applications to 2-message witness hiding. IACR Cryptology ePrint Archive **2018**, 896 (2018)
26. Dodis, Y., Halevi, S., Rothblum, R.D., Wichs, D.: Spooky encryption and its applications. In: Robshaw and Katz [61], pp. 93–122. https://doi.org/10.1007/978-3-662-53015-3, https://doi.org/10.1007/978-3-662-53015-3
27. Dolev, D., Dwork, C., Naor, M.: Non-Malleable Cryptography (Extended Abstract). In: STOC 1991 (1991)
28. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. J. ACM **50**(6), 852–921 (2003). https://doi.org/10.1145/950620.950623, http://doi.acm.org/10.1145/950620.950623
29. Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. SIAM J. Comput. **29**(1), 1–28 (1999). https://doi.org/10.1137/S0097539792230010, https://doi.org/10.1137/S0097539792230010
30. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA. pp. 416–426 (1990). https://doi.org/10.1145/100216.100272, http://doi.acm.org/10.1145/100216.100272
31. Garg, R., Khurana, D., Lu, G., Waters, B.: Black-box non-interactive non-malleable commitments. Cryptology ePrint Archive, Report 2020/1197 (2020), https://eprint.iacr.org/2020/1197

32. Garg, S., Gentry, C., Halevi, S., Raykova, M.: Two-round secure MPC from indistinguishability obfuscation. In: Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings. pp. 74–94 (2014). https://doi.org/10.1007/978-3-642-54242-8_4, https://doi.org/10.1007/978-3-642-54242-8_4
33. Gay, R., Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. IACR Cryptol. ePrint Arch. (2020), https://eprint.iacr.org/2020/764
34. Gay, R., Pass, R.: Indistinguishability obfuscation from circular security. IACR Cryptol. ePrint Arch. (2020), https://eprint.iacr.org/2020/1010
35. Goldreich, O.: The Foundations of Cryptography - Volume 1, Basic Techniques. Cambridge University Press (2001)
36. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J. ACM **38**(3), 691–729 (1991). https://doi.org/10.1145/116825.116852, http://doi.acm.org/10.1145/116825.116852
37. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptology **7**(1), 1–32 (1994). https://doi.org/10.1007/BF00195207, https://doi.org/10.1007/BF00195207
38. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989). https://doi.org/10.1137/0218012, https://doi.org/10.1137/0218012
39. Goyal, V.: Constant Round Non-malleable Protocols Using One-way Functions. In: STOC 2011. pp. 695–704. ACM (2011)
40. Goyal, V., Lee, C.K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: FOCS (2012)
41. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: STOC. pp. 1128–1141. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2897518.2897657, http://doi.acm.org/10.1145/2897518.2897657
42. Goyal, V., Richelson, S.: Non-malleable commitments using goldreich-levin list decoding. In: Zuckerman, D. (ed.) FOCS 2019. pp. 686–699. IEEE Computer Society (2019). https://doi.org/10.1109/FOCS.2019.00047, https://ieeexplore.ieee.org/xpl/conhome/8936052/proceeding
43. Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. In: FOCS 2014. pp. 41–50 (2014). https://doi.org/10.1109/FOCS.2014.13
44. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. J. ACM **59**(3), 11:1–11:35 (2012). https://doi.org/10.1145/2220357.2220358, http://doi.acm.org/10.1145/2220357.2220358
45. Jain, A., Lin, H., Matt, C., Sahai, A.: How to leverage hardness of constant-degree expanding polynomials over r to build io. In: EUROCRYPT (2019)
46. Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. Cryptology ePrint Archive, Report 2020/1003 (2020), https://eprint.iacr.org/2020/1003
47. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. Lecture Notes in Computer Science, vol. 10402, pp. 158–189. Springer (2017). https://doi.org/10.1007/978-3-319-63715-0, https://doi.org/10.1007/978-3-319-63715-0

48. Kalai, Y.T., Khurana, D.: Non-interactive non-malleability from quantum supremacy. In: CRYPTO 2019. pp. 552–582 (2019). https://doi.org/10.1007/978-3-030-26954-8_18, https://doi.org/10.1007/978-3-030-26954-8_18
49. Khurana, D.: Round optimal concurrent non-malleability from polynomial hardness. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. Lecture Notes in Computer Science, vol. 10678, pp. 139–171. Springer (2017). https://doi.org/10.1007/978-3-319-70503-3_5, https://doi.org/10.1007/978-3-319-70503-3_5
50. Khurana, D., Sahai, A.: How to achieve non-malleability in one or two rounds. In: Umans [63], pp. 564–575. https://doi.org/10.1109/FOCS.2017.58, https://doi.org/10.1109/FOCS.2017.58
51. Kuykendall, B., Zhandry, M.: Towards non-interactive witness hiding. Cryptology ePrint Archive, Report 2020/1205 (2020), https://eprint.iacr.org/2020/1205
52. Lin, H., Pass, R.: Constant-round Non-malleable Commitments from Any One-way Function. In: STOC 2011. pp. 705–714
53. Lin, H., Pass, R.: Non-malleability Amplification. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. pp. 189–198. STOC '09 (2009)
54. Lin, H., Pass, R., Soni, P.: Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In: Umans [63], pp. 576–587. https://doi.org/10.1109/FOCS.2017.59, https://doi.org/10.1109/FOCS.2017.59
55. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent Non-malleable Commitments from Any One-Way Function. In: TCC 2008. pp. 571–588
56. Pandey, O., Pass, R., Vaikuntanathan, V.: Adaptive One-Way Functions and Applications. In: Advances in Cryptology — CRYPTO '08. pp. 57–74 (2008)
57. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: EUROCRYPT 2003. pp. 160–176 (2003)
58. Pass, R., Rosen, A.: Concurrent Non-Malleable Commitments. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of ComputerScience. pp. 563–572. FOCS '05 (2005)
59. Pass, R., Rosen, A.: New and Improved Constructions of Nonmalleable Cryptographic Protocols. SIAM J. Comput. **38**(2), 702–752 (2008)
60. Pass, R., Wee, H.: Constant-round non-malleable commitments from sub-exponential one-way functions. In: EUROCRYPT 2010. pp. 638–655 (2010)
61. Robshaw, M., Katz, J. (eds.): Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III, Lecture Notes in Computer Science, vol. 9816. Springer (2016). https://doi.org/10.1007/978-3-662-53015-3, https://doi.org/10.1007/978-3-662-53015-3
62. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) STOC 2014. pp. 475–484. ACM (2014). https://doi.org/10.1145/2591796.2591825, https://doi.org/10.1145/2591796.2591825
63. Umans, C. (ed.): 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017. IEEE Computer Society (2017), https://ieeexplore.ieee.org/xpl/conhome/8100284/proceeding
64. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: FOCS 2010. pp. 531–540 (2010). https://doi.org/10.1109/FOCS.2010.87
65. Wee, H., Wichs, D.: Candidate obfuscation via oblivious LWE sampling. IACR Cryptol. ePrint Arch. (2020), https://eprint.iacr.org/2020/1042