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 $\mathcal{V}$, given 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 distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ of instance-witness pairs in $\mathcal{L}$ such that instances sampled according to $\mathcal{D}_0$ or $\mathcal{D}_1$ are (sufficiently) hard-to-distinguish, a NIDI that outputs instances according to $\mathcal{D}_0$ with proofs of membership in $\mathcal{L}$ is indistinguishable from one that outputs instances according to $\mathcal{D}_1$ with proofs of membership in $\mathcal{L}$.

- We build NIDI arguments for sufficiently hard-to-distinguish distributions assuming sub-exponential indistinguishability obfuscation and sub-exponential 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 necessarily be returned to the committer).

*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.

Contents

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?

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 [GMR89], and it was subsequently shown that all languages in NP admit *interactive* ZK proofs [GMW91]. 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 [GO94]. 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 [DNRS03]** 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 [GO94].

- **Witness Hiding [FS90]** 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) [Gol01]** 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) [FS90]** 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 [Pas03, JKKR17, BGI+17, DK18, BKP19]. 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 [BOV07, GOS12, BP15] 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 [DNRS03, CLP15].

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:

$$\left(c_0 = \mathsf{Com}(m_0; r), \Pi_{c_0 \in \mathcal{L}_\phi}\right) \text{ and } \left(c_1 = \mathsf{Com}(m_1; r), \Pi_{c_1 \in \mathcal{L}_\phi}\right)$$

where Com denotes a perfectly binding commitment (or encryption), and $\Pi_{c \in \mathcal{L}_\phi}$ 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 $\Pi_{c \in \mathcal{L}_\phi}$ 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 [Pas03, JKKR17, BGI+17, DK18, BKP19] of *two-message* strong WI arguments follow variants of the common FLS [FLS99] 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 [Pas03]. The trapdoor z can be obtained by a simulator non-uniformly or in superpolynomial time (or even in polynomial time via specialized recent techniques [JKKR17, DK18, BKP19]).

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 [BP04] 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 [BL18] 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 [PPV08, LPS17, BL18, KK19, GKLW20] 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 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 $\perp$.

- 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 $\mathcal{V}$).

[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 $d \in \mathcal{L}$.

- 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 1.1. *(Informal) Assuming sub-exponentially secure indistinguishability obfuscation and one-way functions, there exists a constant $c \geq 1$ such that for every pair of distributions $\mathcal{D}_0, \mathcal{D}_1$ that cannot be distinguished with advantage better than $2^{-\omega(\log^c \kappa)}$ by any polynomial-sized adversary, NIDI arguments exist.*

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, $\mathcal{C}(1^k, m_0, \phi) \approx \mathcal{C}(1^k, m_1, \phi)$.

We obtain commit-and-prove satisfying the properties described above, that satisfies a relaxed notion of non-interactivity – that we refer to as relaxed non-interactive commit-and-prove.

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).

Theorem 1.2. *(Informal) Assuming sub-exponentially secure indistinguishability obfuscation and one-way functions, there exist relaxed non-interactive commit-and-prove arguments in the plain model.*

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 [CLP10]. 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 1.3. *(Informal) Relaxed non-interactive CCA commitments for 2^κ tags exist assuming sub-exponentially secure indistinguishability obfuscation and one-way functions, and quasi-polynomially secure "base" CCA commitments for $(\log\log\log\kappa)$ tags.*

In fact, we prove a stronger theorem where we only need the base commitments to satisfy a weaker property, namely same-tag CCA security w.r.t. extraction. We note that commitments for $\log\log\log\kappa$ tags satisfying this weaker property can be based on either (1) sub-exponential time-lock puzzles [LPS17] (which can be based on sub-exponential indistinguishability obfuscation and the existence of sub-exponentially hard non-parallelizable languages [BGJ+16]), or (2) sub-exponential hardness of discrete log and sub-exponential quantum hardness of LWE [KK19].

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 probability over the randomness of an honest receiver. Therefore this appears to achieve the conceptual objective of completely non-interactive CCA commitments.

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 (opening), 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 [Agr19, JLMS19, AJL+19, AP20, GJLS20, GP20, BDGM20a] has led to constructions of indistinguishability obfuscation from simpler assumptions, including in [GP20, BDGM20b, WW20] that obtain sub-exponentially secure iO from simple-to-state (circular security) assumptions on LWE-based cryptosystems. Notably, the breakthrough work of [JLS20] obtains (sub-exponential) iO from the following sub-exponential well-founded assumptions: SXDH, LWE, (a variant of) LPN and boolean PRGs in NC0. This helps instantiate indistinguishability obfuscation used in our approach based on sub-exponential well-founded assumptions.

1.2 Additional Related Work

Relaxations of Zero-Knowledge. Subsequent to the introduction of weak zero-knowledge [DNRS03], three-message weak ZK and witness hiding were constructed in [BP12] from assumptions related to point obfuscation with auxiliary inputs, that are now considered implausible due to [BM14, BST16]. Chung et al. [CLP15] proved equivalence between different variants of weak zero-knowledge. Next, [JKKR17] 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 obtained protocols in three messages from standard assumptions, and in two messages from standard, but super-polynomial, assumptions. More recently, [BKP19] obtained two-message weak-zero knowledge (which implies witness hiding and strong WI) in the standard model via a new simulation technique, and concurrently [DK18] obtained two-message witness hiding from new assumptions. Even more recently, [KZ20] 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 arguments with simulators that run in super-polynomial time are known in two messages from standard, but super-polynomial, assumptions [Pas03, BGI+17]. One-message ZK with super-polynomial simulation can be obtained against uniform provers, assuming uniform collision-resistant keyless hash functions [BP04], or against non-uniform verifiers, but with *weak soundness*, assuming multi-collision-resistant keyless hash functions [BL18]. As discussed earlier, these proofs satisfy weak notions of soundness against non-uniform provers (allowing non-uniform provers to cheat on certain instances). While useful in some settings as illustrated in [BL18], this may be undesirable in others.

Non-Malleable Commitments. Minimizing the round complexity of non-malleable commitments has been an important research goal in cryptography. Prior work, namely [DDN91, Bar02, PR05, PR08, LPV, PPV08, LP09, Wee10, PW10, LP, Goy11, GLOV12, GRRV14, GPR16, COSV17, COSV16] culminated in three round non-malleable commitments from standard polynomial-time assumptions [GR19, Khu17] and two round commitments from sub-exponential assumptions like time-lock puzzles [LPS17] and sub-exponential DDH/LWE/QR/NR [KS17].

However, achieving non-interactive non-malleable commitments from well-found assumptions has been particularly challenging. In the non-interactive setting, Pandey, Pass and Vaikuntanathan [PPV08] first gave constructions of non-malleable commitments based on a strong non-falsifiable assumption ("adaptive" one-way functions). Recently Bitansky and Lin [BL18] 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 [KK19] 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. [GKLW20] improved upon [BL18], 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 by employing $\mathsf{ICP}.\mathcal{V}$'s strategy, 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 [CGGM00]. Thus one would generally need to rely on *resettably zero-knowledge* protocols that satisfy security in the presence of resetting attacks [CGGM00].

Second, we note that general-purpose obfuscators satisfying the most natural notion of security (virtual-black-box) cannot exist [BGI+12]. 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 [GP20, BDGM20b, WW20, JLS20]).

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 [GGHR14, DHRW16, AJN+16]). 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 [Pas03].

- 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 ϕ.

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

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

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

Figure 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 $\mathsf{Hybrid}_{2^\kappa} \equiv \mathcal{C}(1^\kappa, m_1, \phi)$.

Let us now argue that for all $j \in [0, 2^\kappa - 1]$, $\mathsf{Hybrid}_j \approx \mathsf{Hybrid}_{j+1}$. Note that the only difference between the two hybrids is the difference in behavior of programs $P^{(j)}$ and $P^{(j+1)}$ on input $y = j$. While $P^{(j)}$ 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 and Hybrid_{j+1} 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 $O(2^\kappa)$ hybrids, denoting (an upper bound on) the adversary's distinguishing advantage between any consecutive pair Hybrid_j and Hybrid_{j+1} by μ, the overall advantage between $\mathcal{C}(1^\kappa, m_0, \phi)$ and $\mathcal{C}(1^\kappa, m_1, \phi)$ can grow to $O(2^\kappa) \cdot \mu$, which is not negligible unless $\mu = \frac{\mathsf{negl}(\kappa)}{O(2^\kappa)}$.

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 $\mathsf{negl}(2^{k^\epsilon})$ 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 [BS20] to achieve soundness in the context of post-quantum interactive ZK arguments.

2.2 Non-Interactive Distributional Indistinguishability

The 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 tags, and then (2) recursively apply *tag amplification*, discussed below, several times until a scheme supporting (2^λ)

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 $\mathcal{R}_{\mathcal{L}}(d, \mathcal{D}, r) = 1\big)$
   OR $\big(c'$ is an encryption w.r.t. pk, of α such that $f(\alpha) = y\big)$."

5. Output (d, c', e).

Figure 2: **Program P'.**

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 [DDN91].

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 '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

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

Π 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 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

$\widetilde{t}_1, \ldots \widetilde{t}_n$ 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 $\tilde{t}_{i_j} \notin \{t_1^1, t_2^1, \ldots t_n^1\}$. 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 becomes to a 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 like keyless collision resistance against non-uniform adversaries [BL18], or achieving only weak forms of security [LPS17, KK19, GKLW20]. In [LPS17, BL18], 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 [LPS17], 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 [BL18]. In addition [KK19] use NIWIs without trapdoors, but only achieve weaker forms of non-malleability (that is, w.r.t. replacement). Even more recently, [GKLW20] 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 schemes with slightly super-constant tags and their underlying assumptions for completeness, before going back to discussing the tag amplification bottleneck.

Base Schemes. Three recent works [LPS17, BL18, KK19] 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 [LPS17] assume a sub-exponential variant of the hardness of time-lock puzzles. Bitansky and Lin [BL18] 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 [KK19] 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.

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

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

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $(r_1, r_2, \ldots, r_{n+2}) = \mathsf{PRF}(K, y)$.
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. $\forall i \in [n], c_i = \mathsf{ComSmall}(m; r_i)$)
   OR $\big(c'$ is an encryption w.r.t. pk, of α such that $f(\alpha) = y\big)$."

6. Output $(\{c_i\}_{i\in[n]}, c', e)$.

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

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 simulation-based security 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

[3]In the main technical body, we use a somewhat more optimal encoding scheme due to [KS17], but we ignore this optimization for the purposes of this overview.

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 [LPS17, BL18, KK19] (for $O(\log \log \kappa)$ 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[4]. Before performing our tag amplification process, we will need to remove this "same-tag" restriction.

We rely on a technique proposed by [GKLW20] 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,

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

where Π 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 CCACom-same-tag w.r.t. tag t^*, and on the other hand, all of the adversary's oracle decommitment queries *will contain* CCACom-same-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.

[4]In fact, the scheme in [LPS17] only satisfies a weaker notion of CCA security called same-tag CCA security w.r.t. extraction. In Section 6, we show how our compiler also applies to this weaker notion of security.

- We model 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, we write $X \approx_{T(\kappa)} Y$ if for all $\mathsf{poly}(\kappa)$-sized circuits $\mathcal{A}$, there exists a negligible function μ such that for all κ,

$$\big|\Pr[\mathcal{A}(X) = 1] - \Pr[\mathcal{A}(Y) = 1]\big| \leq \mu(T(\kappa)).$$

- When we say that a primitive is $T(\kappa)$-secure, unless otherwise stated, we mean that $\mathsf{poly}(T(\kappa))$-sized adversaries do not have advantage better than $\mathsf{negl}(T(\kappa))$ in breaking the primitive's security. When we say that a primitive is sub-exponentially secure, we mean that there exists a constant $\epsilon > 0$ such that the primitive is $T(\kappa) = 2^{\lambda^\epsilon}$ secure.
- 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 $r \leftarrow \{0,1\}^*$.

3.1 One-Way Puzzles

A one-way puzzle is a generalization of a one-way function. While inverting a one-way function requires finding x such that $f(x) = y$, "inverting" the puzzle consists of finding a hard-to-compute solution/secret that need not necessarily correspond to the entirety of the inverse x. In more detail, we say that a puzzle consists of two algorithms with the following syntax,

- $\mathsf{Puzzle.Gen}(1^\kappa) \to (x, y)$ is a randomized algorithm that outputs a puzzle $y \in \{0,1\}^\kappa$, together with an implicit description of a set X of solutions to y.
- $\mathsf{Puzzle.Check}(1^\kappa, z, y) \to \{0,1\}$ returns 0 or 1. Informally, this algorithm returns 1 if and only if z is in the solution set X corresponding to puzzle y.

Definition 3.1 (One-Way Puzzles)**.** We will say that a puzzle $(\mathsf{Puzzle.Gen}, \mathsf{Puzzle.Check})$ is correct if for every $\kappa \in \mathbb{N}$, every $(X, y) \in \mathsf{Supp}(\mathsf{Puzzle.Gen}(1^\kappa))$, every $z \in X$, $\mathsf{Puzzle.Check}(1^\kappa, z, y) = 1$.

We say that a puzzle is sub-exponentially one-way if there exists a constant $\epsilon > 0$ such that for every (non-uniform) $\mathsf{poly}(2^{\kappa^\epsilon})$-sized adversary $\mathcal{A}$,

$$\Pr_{\substack{(x,y)\leftarrow\mathsf{Puzzle.Gen}(1^\kappa)\\ z\leftarrow\mathcal{A}(y)}}[\mathsf{Puzzle.Check}(1^\kappa, z, y) = 1] = \mathsf{negl}(2^{\kappa^\epsilon})$$

A one-way puzzle is a puzzle that satisfies the correctness and one-wayness properties described above.

In what follows, we define a property called verifiability that allows one to efficiently check whether a puzzle y has at least one solution.

Definition 3.2 (Verifiable One-Way Puzzles)**.** We say that a one-way puzzle is verifiable if there exists an algorithm $\mathsf{Puzzle.Ver}(1^\kappa, y)$ that on input any $y \in \{0,1\}^*$ outputs 1 if and only if there exist $z \in \{0,1\}^*$ such that $\mathsf{Puzzle.Check}(1^\kappa, z, y) = 1$.

Any one-way permutation is also a one-way puzzle, with a trivial verification algorithm. In what follows, following [BP15, BPW16] we sketch how a (subexponentially secure) verifiable one-way puzzle can be obtained based on any (subexponentially secure) one-way function family and any (subexponentially secure) obfuscation scheme.

Imported Theorem 3.1. *[BP15, BPW16] Assuming the existence of any (subexponentially secure) one-way function family $\{f_\kappa\}_{\kappa\in\mathbb{N}} : \{0,1\}^{n(\kappa)} \to \{0,1\}^\kappa$ and any (subexponentially secure) indistinguishability obfuscation scheme, there exist (subexponentially secure) verifiable one-way puzzles satisfying Definition 3.2.*

Proof. (Sketch) Following [BP15] we note that a *keyed* family of one-way permutations with a key generation algorithm $\mathsf{KeyGen}(1^\kappa) \to K \in \{0,1\}^\kappa$ and a function $F_K : \mathcal{D}_K \to \mathcal{D}_K$, where each key K defines a domain $\mathcal{D}_K \subseteq \{0,1\}^\kappa$, where it is possible to sample (pseudo)random elements in $\mathcal{D}_K$, and where membership in $\mathcal{D}_K$ can be tested efficiently, suffice. Such (sub-exponentially hard to invert) one-way permutations can be obtained from (sub-exponential) indistinguishability obfuscation and one-way functions following [BPW16].

- $\mathsf{Puzzle.Gen}(1^\kappa)$ samples $K \leftarrow \mathsf{KeyGen}(1^\kappa)$, x (pseudo)randomly from $\mathcal{D}_K$ and outputs $y = (K, F_K(x)), x$.
- $\mathsf{Puzzle.Check}(1^\kappa, z, y)$ parses $y = (K, y')$ and outputs 1 if either
  - z was an inverse of y, that is $y' = F_K(z)$, or
  - y was not in the domain D_K, or
  - $F_k(z) \notin \mathcal{D}_K$ or $z = (x_0, x_1)$ such that $F_k(x_0) = F_k(x_1)$.
- $\mathsf{Puzzle.Ver}(1^\kappa)$ is a trivial algorithm that outputs 1 on any input of size κ.

Correctness and verifiability follow by noting that for any $y = (K, y')$,

- Either K is a valid key for a permutation and $y' \in \mathcal{D}_K$, in which case there always exists an inverse z of y' under F_K, or
- K is a valid key for a permutation over domain $\mathcal{D}_K$ and $y' \notin \mathcal{D}_K$, which can be efficiently checked, or
- K corresponds to a function F_K that is not a permutation over its domain $\mathcal{D}_K$, in which case there either exists z such that $F_K(z) \notin \mathcal{D}_K$ or there exists $z = (x_0, x_1)$ such that $F_K(x_0) = F_K(x_1)$.

This means that for every $y = (K, y')$ there exists z such that $\mathsf{Puzzle.Check}(1^\kappa, z, y) = 1$. □

3.2 Indistinguishability Obfuscation

The notion of indistinguishability obfuscation (iO) [BGI+12] guarantees that the obfuscation of two circuits are computationally indistinguishable as long as they both are equivalent circuits, i.e., the output of both the circuits are the same on every input. Formally,

Definition 3.3 (Indistinguishability Obfuscator (iO) for Circuits)**.** A uniform PPT algorithm iO is called an indistinguishability obfuscator for a circuit family $\{\mathcal{C}_\kappa\}_{\kappa\in\mathbb{N}}$, where $\mathcal{C}_\kappa$ consists of circuits C of the form $C : \{0,1\}^n \to \{0,1\}$ with $n = n(\kappa)$, if the following holds:

- **Perfect Correctness:** For every $\kappa \in \mathbb{N}$, every $C \in \mathcal{C}_\kappa$, every input $x \in \{0,1\}^n$, we have that

$$\Pr\left[C'(x) = C(x) \ : \ C' \leftarrow \mathsf{iO}(1^\kappa, C)\right] = 1$$

- **Indistinguishability:** For all pairs of circuits $C_0, C_1 \in \mathcal{C}_\kappa$ such that $C_0(x) = C_1(x)$ for all inputs $x \in \{0,1\}^n$ and $|C_0| = |C_1|$, we have:

$$\mathsf{iO}(\kappa, C_0) \approx_\kappa \mathsf{iO}(\kappa, C_1)$$

- **Polynomial Slowdown**: For every $\kappa \in \mathbb{N}$, every $C \in \mathcal{C}_\kappa$, we have that $|\mathsf{iO}(1^\kappa, C)| = \mathrm{poly}(\kappa, C)$.

Definition 3.4 (Sub-exponentially Secure Indistinguishability Obfuscator (iO) for Circuits)**.** This is defined identically to the previous definition except the indistinguishability condition is modified to the following: There exists a constant $\epsilon > 0$ such that for all pairs of circuits $C_0, C_1 \in \mathcal{C}_\kappa$ such that $C_0(x) = C_1(x)$ for all inputs $x \in \{0,1\}^n$ and $|C_0| = |C_1|$, we have that for every $\mathsf{poly}(2^{\kappa^\epsilon})$-sized adversary $\mathcal{A}$,

$$|\Pr[\mathcal{A}(\mathsf{iO}(\kappa, C_0)) = 1] - \Pr[\mathsf{iO}(\kappa, C_1) = 1]| \leq \mathsf{negl}(2^{\kappa^\epsilon})$$

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

Let $\mathcal{D}$ denote a sampling circuit that on input uniform randomness outputs samples from a distribution $(\mathcal{X}, \mathcal{W})$ over instance-witness pairs,

In a NIDI, the prover algorithm $\mathcal{P}$ obtains input a security parameter, a sampling circuit $\mathcal{D}$, 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. We begin with the most natural definition, which we unfortunately do not achieve in this work, and leave as an interesting open question for future work. Next, we provide a more relaxed definition which we do achieve.

Definition 4.1 (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 $\mathcal{P}$ and $\mathcal{V}$[5] satisfy:

- **Completeness:** For every distribution $(\mathcal{X}, \mathcal{W})$ over instance-witness pairs in $R_{\mathcal{L}}$ that can be sampled by a $\mathsf{poly}(\kappa)$-size circuit $\mathcal{D}$,

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

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

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

- **Distributional Indistinguishability:** For every pair of distributions $(\mathcal{X}_0, \mathcal{W}_0)$ and $(\mathcal{X}_1, \mathcal{W}_1)$ over instance-witness pairs in $R_{\mathcal{L}}$ that can be sampled by $\mathsf{poly}(\kappa)$-size circuits $\mathcal{D}_0$ and $\mathcal{D}_1$ respectively, if $\mathcal{X}_0 \approx_\kappa \mathcal{X}_1$, then

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

Definition 4.2 (NIDI Arguments for $T(\kappa)$-Hard Distributions)**.** Let $T(\kappa)$ denote a time bound. A pair of PPT algorithms $(\mathcal{P}, \mathcal{V})$ is a non-interactive distributionally-indistinguishable (NIDI) argument for $T(\kappa)$-hard distributions and NP language $\mathcal{L}$ with associated relation $\mathcal{R}_{\mathcal{L}}$ if the non-interactive algorithms $\mathcal{P}$ and $\mathcal{V}$ satisfy the completeness and soundness properties from Definition 4.1, and additionally satisfy:

- **Distributional Indistinguishability for $T(\kappa)$-Hard Distributions:** For every pair of distributions $(\mathcal{X}_0, \mathcal{W}_0)$ and $(\mathcal{X}_1, \mathcal{W}_1)$ over instance-witness pairs in $R_{\mathcal{L}}$ that can be sampled by $\mathsf{poly}(\kappa)$-size circuits $\mathcal{D}_0$ and $\mathcal{D}_1$ respectively, if $\mathcal{X}_0 \approx_{T(\kappa)} \mathcal{X}_1$, then

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

4.2 Construction and Analysis

In this section, we construct NIDIs and prove the following theorem.

Theorem 4.1. *Assuming the existence of sub-exponentially secure indistinguishability obfuscation and sub-exponentially secure one-way functions, 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 4.2.*

To prove Theorem 4.1, 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 described below. Looking ahead, we point out that the constant c will be related to the exact sub-exponential security (i.e. related to the constant ϵ where we assume 2^{k^ϵ} security of the primitive with security parameter k) of the cryptographic primitives used in the construction.

[5]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.

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

- There exists a *sub-exponentially secure* one-way puzzle ensemble $\{f_{k'} : \{0,1\}^{\mathsf{poly}(k')} \to \{0,1\}^{k'}\}_{k' \in \mathbb{N}}$ These can be based on sub-exponential indistinguishability obfuscation and one-way functions (Section 3). We require that for large enough security parameter k', the corresponding puzzle can be solved with probability at most $\mathsf{negl}(2^{k'^{\epsilon}})$ by circuits of size $\mathsf{poly}(2^{k'^{\epsilon}})$.
- There exists a *perfectly correct, sub-exponentially secure* public-key encryption scheme $\mathsf{PKE} = \mathsf{PKE.KeyGen}(1^k), \mathsf{PKE.Enc}(1^k, pk, x; r), \mathsf{PKE.Dec}(1^k, sk, c)$. We require that for large enough security parameter k, the advantage of every $\mathsf{poly}(2^{k^{\epsilon}})$-size adversary in the IND-CPA game is $\mathsf{negl}(2^{k^{\epsilon}})$.
- There exists a *sub-exponentially secure* indistinguishability obfuscation scheme $(\mathsf{iO.Obf}, \mathsf{iO.Eval})$. We require that for large enough security parameter k, the advantage of every $\mathsf{poly}(2^{k^{\epsilon}})$-size adversary in the obfuscation security game is $\mathsf{negl}(2^{k^{\epsilon}})$.
- There exists a *sub-exponentially secure* puncturable PRF. We require that for large enough security parameter k, the advantage of every $\mathsf{poly}(2^{k^{\epsilon}})$-size adversary in the puncturable PRF indistinguishability game is $\mathsf{negl}(2^{k^{\epsilon}})$.
- There exist *sub-exponentially secure* $\mathsf{NIWI} = (\mathsf{NIWI.Prove}(1^k, x, w), \mathsf{NIWI.Verify}(1^k, x, \pi))$ proofs with statistical sub-exponential soundness. We require that for large enough security parameter k, the advantage of every $\mathsf{poly}(2^{k^{\epsilon}})$-size adversary in the NIWI indistinguishability game is $\mathsf{negl}(2^{k^{\epsilon}})$.

We note that the required public-key encryption scheme can be based on sub-exponential indistinguishability obfuscation and sub-exponential one-way functions following [SW14]. The required puncturable PRF is known to be implied by sub-exponential one-way functions (see eg., [SW14]) and the required NIWIs can also be based on sub-exponential iO and sub-exponential verifiable one-way puzzles [BP15].

Now, 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

$$\mathcal{L}_{\mathsf{NIWI}} = \Big\{(pk, d_x, cm, y) : \exists (d_w, s, sk) \text{ s.t. } \big((d_x, d_w) \in \mathcal{R}_{\mathcal{L}}\big) \bigvee \big((pk, sk) \leftarrow \mathsf{KeyGen}(1^k, s) \wedge \mathsf{Puzzle.Check}(y, \mathsf{Dec}_{sk}(cm)) = 1\big)\Big\}$$

- The prove algorithm $\mathcal{P}(1^{\kappa}, \mathcal{D}, \mathcal{L})$ does the following:
  - Set $k = (\log \kappa)^{c^2}, k' = (\log \kappa)^c$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k, s)$.
  - 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 $(pk, \widetilde{P})$.
- The verify algorithm $\mathcal{V}(1^{\kappa}, \pi, \mathcal{L})$ on input a proof $\pi = (pk, \widetilde{P})$ does the following:

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

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

1. If $\mathsf{Puzzle.Ver}(y) \neq 1$, 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 $cm = \mathsf{Enc}_{pk}(0^{k'}; r_2)$.
5. Set $x = (pk, d_x, cm, y)$, $w = (d_w, 0^{2k})$.
   Then compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_{\mathsf{NIWI}}; r_3)$.
6. Output (x, e).

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

- Sample $(x, y) = \mathsf{Puzzle.Gen}(1^{k'})$.
- Compute $\mathsf{out} = \mathsf{iO.Eval}(\widetilde{P}, y)$. Parse $\mathsf{out} = (x, e)$ and parse $x = (pk, d, cm, y)$.
- If $\mathsf{NIWI}.\mathcal{V}(1^k, x, e, \mathcal{L}_{\mathsf{NIWI}})$ rejects, output $\perp$ and stop.
- Else output d.

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

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 4.2. *Under the assumptions in Theorem 4.1, construction 4.1 satisfies soundness according to Definition 4.1.*

Proof. Suppose there exists a $\mathsf{poly}(\kappa)$-sized (non-uniform) prover $\mathcal{P}^*$ that outputs string $\pi^* \in \{0,1\}^*$, and suppose there exists a polynomial $p(\cdot)$ such that:

$$\Pr\left[\big(\mathcal{V}(1^\kappa, \pi, \mathcal{L}) \neq \perp\big) \wedge \big(\mathcal{V}(1^\kappa, \pi, \mathcal{L}) \notin \mathcal{L}\big) \Big| \pi \leftarrow \mathcal{P}^*\right] \geq \frac{1}{p(\kappa)}$$

Then, we construct a non-uniform adversary $\mathcal{A}^*$ that contradicts one-wayness of the puzzle f. $\mathcal{A}^*$ obtains non-uniformly a purported NIDI proof π^* for language $\mathcal{L}$, where $\pi^* = (pk^*, \widetilde{P})$, and also non-uniformly obtains the secret key sk^* for pk^* (if one exists). Otherwise, it sets sk^* to 0^k.

Next, $\mathcal{A}^*$ obtains a string y and does the following:

- Compute $\mathsf{out} = \mathsf{iO.Eval}(\widetilde{P}, y)$. Parse $\mathsf{out} = (x^*, e^*)$ and parse $x^* = (pk^*, d^*, c^*, y)$.
- Output $z = \mathsf{Dec}_{sk^*}(c^*)$.

Now by assumption,

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

which by our construction implies that

$$\Pr\left[\left(\mathsf{NIWI}.\mathcal{V}(x^*,e^*,\mathcal{L}_{\mathsf{NIWI}})\text{ accepts }\right)\wedge\left(\mathcal{V}(1^\kappa,\pi,\mathcal{L})\notin\mathcal{L}\right)\middle|\pi\leftarrow\mathcal{P}^*\right]\geq\frac{1}{p(\kappa)}\tag{1}$$

By (statistical) soundness of the NIWI, with probability $1-\mathsf{negl}(\kappa)$ over the NIWI verifier's randomness, $\mathsf{NIWI}.\mathcal{V}(x^*,e^*,\mathcal{L}_{\mathsf{NIWI}})$ accepts iff $x^*\in\mathcal{L}_{\mathsf{NIWI}}$, or equivalently for $x^*=(pk^*,d^*,c^*,y^*)$,

$$\exists(r^*,s^*,sk^*)\text{ s.t. }\Big((d^*,r^*)\in\mathcal{R}_{\mathcal{L}}\Big)\bigvee\Big((pk^*,sk^*)\leftarrow\mathsf{KeyGen}(s^*)\wedge\mathsf{Puzzle.Check}(y,(\mathsf{Dec}_{sk^*}(c^*)))=1\Big)\tag{2}$$

Combining equations (1) and (2),

$$\Pr\Bigg[\Bigg(\exists(r^*,s^*,sk^*)\text{ s.t. }\big((d^*,r^*)\in\mathcal{R}_{\mathcal{L}}\big)\bigvee\big((pk^*,sk^*)\leftarrow\mathsf{KeyGen}(s^*)\wedge$$
$$\mathsf{Puzzle.Check}(y,(\mathsf{Dec}_{sk^*}(c^*)))=1\big)\Bigg)\bigwedge\Big(\mathcal{V}(1^\kappa,\pi,\mathcal{L})\notin\mathcal{L}\Big)\Bigg]\geq\frac{1}{p(\kappa)}-\mathsf{negl}(\kappa)=\frac{1}{2p(\kappa)}$$

By noting that $d^*\notin\mathcal{L}$ implies that there does not exist r^* such that $\big((d^*,r^*)\in\mathcal{R}_{\mathcal{L}}\big)$, we have:

$$\Pr\left[\Big(\exists(r^*,s^*,sk^*)\text{ s. t. }(pk^*,sk^*)\leftarrow\mathsf{KeyGen}(s^*)\wedge\mathsf{Puzzle.Check}(y,(\mathsf{Dec}_{sk^*}(c^*)))=1\Big)\bigwedge\Big(d^*\notin\mathcal{L}\Big)\right]\geq\frac{1}{2p(\kappa)}$$

which in particular implies that for $(pk^*,\widetilde{P})$ and corresponding sk^* set non-uniformly by $\mathcal{A}$,

$$\Pr\left[\Big(\exists(r^*,s^*,sk^*)\text{ s. t. }(pk^*,sk^*)\leftarrow\mathsf{KeyGen}(s^*)\wedge\mathsf{Puzzle.Check}(y,(\mathsf{Dec}_{sk^*}(c^*)))=1\Big)\right]\geq\frac{1}{p(\kappa)}=\frac{1}{2p(2^{k'^\epsilon})}$$

which implies that with probability at least $\frac{1}{2p(2^{k'^\epsilon})}$, the output of $\mathcal{A}^*(y)$, which is $z=\mathsf{Dec}_{sk^*}(c^*)$ as discussed above, is such that $\mathsf{Puzzle.Check}(y,(\mathsf{Dec}_{sk^*}(c^*))=1$.

Therefore $\mathcal{A}^*(y)$ runs in time $\mathsf{poly}(\kappa)=\mathsf{poly}(2^{k'^\epsilon})$ and contradicts one-wayness of the puzzle, as desired. This completes the proof. □

Lemma 4.3. *Under the assumptions in Theorem 4.1, Construction 4.1 satisfies distributional indistinguishability for $T(\kappa)$-hard distributions per Definition 4.2.*

Proof. We prove the following (stronger) statement: there is a large enough constant $c>1$ such that for every language $\mathcal{L}$, every $\mathsf{poly}(\kappa)$-sampleable pair of distributions $(\mathcal{D}_0,\mathcal{D}_1)$ such that $\mathsf{Supp}(\mathcal{D}_0)\cup\mathsf{Supp}(\mathcal{D}_1)\subseteq\mathcal{L}$ and $\mathcal{D}_0\approx_{2^{\log\kappa^c}}\mathcal{D}_1$,

$$\mathcal{P}(1^\kappa,\mathcal{D}_0,\mathcal{L})\approx_{2^{\log\kappa^c}}\mathcal{P}(1^\kappa,\mathcal{D}_1,\mathcal{L})$$

We will in fact set $c=\frac{1}{\epsilon}$ as above, where we recall that $0<\epsilon<1$ is a constant such that the underlying cryptographic primitives are assumed to satisfy sub-exponential 2^{k^ϵ} security.

To prove this lemma, we define a sequence of $(2^{k'}+1)$ hybrids, $(\mathsf{Hybrid}_0, \mathsf{Hybrid}_1, \ldots, \mathsf{Hybrid}_{2^{k'}})$ such that $\mathsf{Hybrid}_0 \equiv \mathcal{P}(1^\kappa, \mathcal{D}_0, \mathcal{L})$ and $\mathsf{Hybrid}_{2^{k'}} \equiv \mathcal{P}(1^\kappa, \mathcal{D}_1, \mathcal{L})$.

For each $i \in [0, 2^{k'}]$, the distribution Hybrid_i depends on $(1^{k'}, \mathcal{D}_0, \mathcal{D}_1, \mathcal{L}, \epsilon)$ and is defined as follows:

- 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 $P^i_{pk,K,\mathcal{D},\mathcal{L}}$ defined in Figure 5.
- Compute $\widetilde{P} = \mathsf{iO.Obf}(P^i_{pk,K,\mathcal{D},\mathcal{L}}; R)$.
- Output $(pk, \widetilde{P})$.

Hardwired: Public key pk, Index $i \in [2^\kappa]$, Puncturable PRF Key K, Distributions $(\mathcal{D}_0, \mathcal{D}_1)$, Language $\mathcal{L}$.

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

1. If $\mathsf{Puzzle.Ver}(y) \neq 1$, output $\perp$. Otherwise, continue.
2. Set $(r_1, r_2, r_3) = \mathsf{PRF}(K, y)$.
3. If $y < i$, set $d = \mathcal{D}_1(r_1)$.
4. If $y \geq i$, set $d = \mathcal{D}_0(r_1)$.
5. Set $c = \mathsf{Enc}_{pk}(0^{k'}; r_2)$.
6. Set $x = (pk, d, c, y)$, $w = (r_1, 0^{2k})$. Then compute $e = \mathsf{NIWI}.\mathcal{P}(x, w, \mathcal{L}_{\mathsf{NIWI}}; r_3)$.
7. Output (x, e).

Figure 5: **Program** $P^i_{pk,K,\mathcal{D},\mathcal{L}}$.

We now prove that for every $i \in [1, 2^{k'}]$ and every (non-uniform) $\mathsf{poly}(2^{k^\epsilon})$-sized distinguisher D there exists a negligible function $\mu(\cdot)$ such that:

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_i) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i+1}) = 1] \right| = \mu(2^{k^\epsilon})$$

We also note that when $\mathsf{Puzzle.Ver}(i) \neq 1$, the programs $P^i_{pk,K,\mathcal{D},\mathcal{L}}$ and $P^{i+1}_{pk,K,\mathcal{D},\mathcal{L}}$ are functionally equivalent. Therefore by security of iO with security parameter k, we have that for every $\mathsf{poly}(2^{k^\epsilon})$-sized distinguisher D there exists a negligible function $\mu(\cdot)$ such that:

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_i) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i+1}) = 1] \right| = \mu(2^{k^\epsilon})$$

When $\mathsf{Puzzle.Ver}(i) = 1$, we prove the claim by considering a sequence of additional sub-hybrids between Hybrid_i and Hybrid_{i+1} called $(\mathsf{Hybrid}_{i,1}, \ldots, \mathsf{Hybrid}_{i,10})$ where:

$$\mathsf{Hybrid}_{i,1} \equiv \mathsf{Hybrid}_i \text{ and } \mathsf{Hybrid}_{i,10} \equiv \mathsf{Hybrid}_{i+1}$$

- $\mathsf{Hybrid}_{i,1} \equiv \mathsf{Hybrid}_i$.
- $\mathsf{Hybrid}_{i,2}$ depends on $(1^\kappa, \mathcal{D}_0, \mathcal{D}_1, \mathcal{L}, \epsilon)$ and is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{i,1}$ and $\mathsf{Hybrid}_{i,2}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k, R \leftarrow \{0,1\}^k$.
  - <u>Set $(r_1^*, r_2^*, r_3^*) = \mathsf{PRF}(K, i)$.</u>
  - <u>Set $d^* = \mathcal{D}_0(r_1^*)$ and $c^* = \mathsf{Enc}_{pk}(0^\kappa; r_2^*)$.</u>
  - <u>Set $x^* = (pk, d^*, c^*, i)$, $w^* = (r_1^*, 0^{2k})$. Then compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_3^*)$.</u>
  - Generate program $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ defined in Figure 6.
  - Compute $\widetilde{P} = \mathsf{iO}.\mathsf{Obf}(\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}; R)$.
  - Output $(pk, \widetilde{P})$.

Hardwired: Public key pk, Index $i \in [2^\kappa]$, Punctured PRF Key $K\{i\}$, <u>(x^*, e^*)</u>, Distributions $(\mathcal{D}_0, \mathcal{D}_1)$, Language $\mathcal{L}$.

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

1. If $\mathsf{Puzzle.Ver}(y) \neq 1$, output $\perp$. Otherwise, continue.
2. <u>If $y = i$, output (x^*, e^*) and stop.</u>
3. Compute $(r_1, r_2, r_3) = \mathsf{PRF}(K, y)$.
4. If $y < i$, set $d = \mathcal{D}_1(r_1)$.
5. If $y \geq i$, set $d = \mathcal{D}_0(r_1)$.
6. Set $c = \mathsf{Enc}_{pk}(0^{k'}; r_2)$.
7. Set $x = (pk, d, c, y)$, $w = (r_1, 0^{2k})$. Then compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_{\mathsf{NIWI}}; r_3)$.
8. Output (x, e).

Figure 6: **Program** $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$.

Claim 4.1. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_1(\cdot)$ *such that:*

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,1}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,2}) = 1] \right| = \mu_1(2^{k^\epsilon})$$

Proof. Note that the programs $P^i_{pk,K,\mathcal{D},\mathcal{L}}$ and $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ have identical functionality. Therefore, by sub-exponential security of indistinguishability obfuscation (with security parameter k), for every $i \in [1, 2^{k'}]$ and every (non-uniform) $\mathsf{poly}(2^{k^\epsilon})$-sized distinguisher D there exists a negligible function $\mu_1(\cdot)$ such that:

$$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,1}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,2}) = 1]\right| = \mu_1(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{i,3}$ is identical to $\mathsf{Hybrid}_{i,2}$ except that $(r^*_1, r^*_2, r^*_3) \leftarrow \{0,1\}^{\kappa+2k}$.

  Claim 4.2. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_2(\cdot)$ *such that:*

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,2}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,3}) = 1]\right| = \mu_2(2^{k^\epsilon})$$

  Proof. By sub-exponential security of the puncturable PRF (with security parameter k), we have directly that for every $i \in [1, 2^{k'}]$ and every (non-uniform) $\mathsf{poly}(2^{k^\epsilon})$-sized distinguisher D there exists a negligible function $\mu_2(\cdot)$ such that:

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,2}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,3}) = 1]\right| = \mu_2(2^{k^\epsilon})$$

  □

- $\mathsf{Hybrid}_{i,4}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{i,3}$ and $\mathsf{Hybrid}_{i,4}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k, R \leftarrow \{0,1\}^k$.
  - Set $(r^*_1, r^*_2, r^*_3) \leftarrow \{0,1\}^{\kappa+2k}$.
  - Compute (in time $\mathsf{poly}(2^{k'}) = \mathsf{poly}(2^{k^\epsilon})$) a solution s^*_1 such that $\mathsf{Puzzle.Check}(s^*_1, i) = 1$.
  - Set $d^* = \mathcal{D}_0(r^*_1)$ and $c^* = \mathsf{Enc}_{pk}(s^*_1; r^*_2)$.
  - Set $x^* = (pk, d^*, c^*, i)$, $w^* = (r^*_1, 0^{2k})$. Then compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r^*_3)$.
  - Generate program $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ defined in Figure 6.
  - Compute $\widetilde{P} = \mathsf{iO}.\mathsf{Obf}(\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}; R)$.
  - Output $(pk, \widetilde{P})$.

  Claim 4.3. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_3(\cdot)$ *such that:*

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,3}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,4}) = 1]\right| = \mu_3(2^{k^\epsilon})$$

  Proof. We will prove this claim based on the sub-exponential IND-CPA security of the encryption scheme against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher D and a polynomial $p(\cdot)$ such that

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,3}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,4}) = 1]\right| \geq \frac{1}{p(2^{k^\epsilon})}$$

Then there exists an adversary D' that computes $s_1^* = f^{-1}(i)$ by brute-force listing all possible solutions to the one-way puzzle in time $\mathsf{poly}(2^{k'}) = \mathsf{poly}(2^{k^\epsilon})$, then begins the experiment. It obtains c^* from the CPA challenger as either $\mathsf{Enc}_{pk}(0^{k'}; r_2^*)$ or $\mathsf{Enc}_{pk}(s_1^*; r_2^*)$ It completes the rest of the experiment according to $\mathsf{Hybrid}_{i,3}$ except setting c^* according to the ciphertext obtained from the external challenger. It then mirrors the output of D given the resulting distribution, which implies that

$$\left| \Pr[\mathsf{D}'(\mathsf{Enc}_{pk}(0^\kappa; r_2^*)) = 1] - \Pr[\mathsf{D}'(\mathsf{Enc}_{pk}(s_1^*; r_2^*)) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

which gives a contradiction, as desired. Therefore, by sub-exponential IND-CPA security of the encryption scheme (with security parameter k), for every $i \in [1, 2^\kappa]$ and every (non-uniform) $\mathsf{poly}(k)$-sized distinguisher D there exists a negligible function $\mu_3(\cdot)$ such that:

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,3}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,4}) = 1] \right| = \mu_3(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{i,5}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{i,4}$ and $\mathsf{Hybrid}_{i,5}$)
  - Set $k = \kappa^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k, R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*, r_2^*, r_3^*) \leftarrow \{0,1\}^{\kappa+2k}$.
  - Compute (in time $\mathsf{poly}(2^{k^\epsilon})$) a solution s_1^* such that $\mathsf{Puzzle.Check}(s_1^*, i) = 1$.
  - Set $d^* = \mathcal{D}_0(r_1^*)$ and $c^* = \mathsf{Enc}_{pk}(s_1^*; r_2^*)$.
  - Set $x^* = (pk, d^*, c^*, i)$, $\underline{w^* = (0^\kappa, s, sk).}$ Then compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_3^*)$.
  - Generate program $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ defined in Figure 6.
  - Compute $\widetilde{P} = \mathsf{iO}.\mathsf{Obf}(\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}; R)$.
  - Output $(pk, \widetilde{P})$.

Claim 4.4. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_4(\cdot)$ *such that:*

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,4}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,5}) = 1] \right| = \mu_4(2^{k^\epsilon})$$

Proof. We will prove this claim based on the sub-exponential witness indistinguishability of the NIWI against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher D and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,4}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,5}) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

Then there exists an adversary D' that computes $s_1^* = f^{-1}(i)$ by brute-force listing all possible solutions to the one-way puzzle in time $\mathsf{poly}(2^{k'}) = \mathsf{poly}(2^{k^\epsilon})$, then begins the experiment

It obtains e^* from the NIWI challenger either using witness $w^* = (r_1^*, 0^{2k})$ as in $\mathsf{Hybrid}_{i,4}$, or using witness $w^* = (0^\kappa, s, sk)$ as in $\mathsf{Hybrid}_{i,5}$. It completes the rest of the experiment according to $\mathsf{Hybrid}_{i,4}$ except setting e^* according to the NIWI obtained from the external challenger. It then mirrors the output of D given the resulting distribution, which implies that

$$\left| \Pr[\mathsf{D}'(w^* = (r_1^*, 0^{2k})) = 1] - \Pr[\mathsf{D}'(w^* = (0^\kappa, s, sk)) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

which gives a contradiction, as desired. Therefore, by sub-exponential security of the NIWI (with security parameter k), for every $i \in [1, 2^\kappa]$ and every (non-uniform) $\mathsf{poly}(2^{k^\epsilon})$-sized distinguisher D there exists a negligible function $\mu_4(\cdot)$ such that:

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,4}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,5}) = 1] \right| \leq \mu_4(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{i,6}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{i,5}$ and $\mathsf{Hybrid}_{i,6}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k, R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*, r_2^*, r_3^*) \leftarrow \{0,1\}^{\kappa+2k}$.
  - Compute (in time $\mathsf{poly}(2^{k^\epsilon})$) a solution s_1^* such that $\mathsf{Puzzle.Check}(s_1^*, i) = 1$.
  - Set $\underline{d^* = \mathcal{D}_1(r_1^*)}$ and $c^* = \mathsf{Enc}_{pk}(s_1^*; r_2^*)$.
  - Set $x^* = (pk, d^*, c^*, i)$, $w^* = (0^\kappa, s, sk)$. Then compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_3^*)$.
  - Generate program $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ defined in Figure 6.
  - Compute $\widetilde{P} = \mathsf{iO.Obf}(\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}; R)$.
  - Output $(pk, \widetilde{P})$.

Claim 4.5. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_5(\cdot)$ *such that:*

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,5}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,6}) = 1] \right| = \mu_5(2^{k^\epsilon})$$

Proof. We will prove this claim based on $2^{\log \kappa^c} = 2^{k^\epsilon}$-hardness of $(\mathcal{D}_0, \mathcal{D}_1)$ against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher D and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,5}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,6}) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

Then there exists an adversary D' that computes $s_1^* = f^{-1}(i)$ by brute-force listing all possible solutions to the one-way puzzle in time $\mathsf{poly}(2^{k'}) = \mathsf{poly}(2^{k^\epsilon})$, then begins the experiment. It obtains d^* from an external challenger sampled either as $\mathcal{D}_0(r^*)$ as in $\mathsf{Hybrid}_{i,5}$, or as $\mathcal{D}_1(r^*)$ as in $\mathsf{Hybrid}_{i,6}$. It completes the rest of the experiment according to $\mathsf{Hybrid}_{i,5}$ except setting d^*

according to the sample obtained from the external challenger. It then mirrors the output of D given the resulting distribution, which implies that

$$\left|\Pr[\mathsf{D}'(d^* = \mathcal{D}_0(r^*)) = 1] - \Pr[\mathsf{D}'(d^* = \mathcal{D}_1(r^*)) = 1]\right| \geq \frac{1}{p(2^{k^\epsilon})} = \frac{1}{p(2^{(\log \kappa)^c})}$$

which gives a contradiction to the $2^{(\log \kappa)^c}$-hardness of $(\mathcal{D}_0, \mathcal{D}_1)$, as desired. □

- $\mathsf{Hybrid}_{i,7}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{i,6}$ and $\mathsf{Hybrid}_{i,7}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k, R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*, r_2^*, r_3^*) \leftarrow \{0,1\}^{\kappa+2k}$.
  - Compute (in time $\mathsf{poly}(2^{k^\epsilon})$) a solution s_1^* such that $\mathsf{Puzzle.Check}(s_1^*, i) = 1$.
  - Set $d^* = \mathcal{D}_1(r_1^*)$ and $c^* = \mathsf{Enc}_{pk}(s_1^*; r_2^*)$.
  - Set $x^* = (pk, d^*, c^*, i)$, $\underline{w^* = (r_1^*, 0^{2k})}$. Then compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_3^*)$.
  - Generate program $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ defined in Figure 6.
  - Compute $\widetilde{P} = \mathsf{iO.Obf}(\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}; R)$.
  - Output $(pk, \widetilde{P})$.

  Claim 4.6. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_6(\cdot)$ *such that:*

$$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,6}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,7}) = 1]\right| = \mu_6(2^{k^\epsilon})$$

  Proof. By sub-exponential witness indistinguishability of the NIWI (and following an identical argument to that of the indistinguishability between $\mathsf{Hybrid}_{i,4}$ and $\mathsf{Hybrid}_{i,5}$), for every $\mathsf{poly}(2^{k^\epsilon})$-sized distinguisher D there exists a negligible function $\mu_6(\cdot)$ such that:

$$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,6}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,7}) = 1]\right| \leq \mu_6(2^{k^\epsilon})$$

  □

- $\mathsf{Hybrid}_{i,8}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{i,7}$ and $\mathsf{Hybrid}_{i,8}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k, R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*, r_2^*, r_3^*) \leftarrow \{0,1\}^{\kappa+2k}$.
  - ~~Compute (in time $\mathsf{poly}(2^{k^\epsilon})$) a solution s_1^* such that $\mathsf{Puzzle.Check}(s_1^*, i) = 1$.~~
  - Set $d^* = \mathcal{D}_1(r_1^*)$ and $\underline{c^* = \mathsf{Enc}_{pk}(0^{k'}; r_2^*)}$.
  - Set $x^* = (pk, d^*, c^*, i)$, $w^* = (r_1^*, 0^{2k})$. Then compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_3^*)$.
  - Generate program $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ defined in Figure 6.

– Compute $\widetilde{P} = \mathsf{iO.Obf}(\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}; R)$.

– Output $(pk, \widetilde{P})$.

Claim 4.7. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_7(\cdot)$ *such that:*

$$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,7}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,8}) = 1]\right| = \mu_7(2^{k^\epsilon})$$

Proof. By sub-exponential IND-CPA security of the encryption scheme (and following an identical argument to that of the indistinguishability between $\mathsf{Hybrid}_{i,3}$ and $\mathsf{Hybrid}_{i,4}$), for every $\mathsf{poly}(k)$-sized distinguisher D there exists a negligible function $\mu_7(\cdot)$ such that:

$$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,7}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,8}) = 1]\right| \leq \mu_7(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{i,9}$ is identical to $\mathsf{Hybrid}_{i,8}$ except that $(r^*_1, r^*_2, r^*_3) = \mathsf{PRF}(K, i)$.

  Claim 4.8. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_8(\cdot)$ *such that:*

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,8}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,9}) = 1]\right| = \mu_8(2^{k^\epsilon})$$

  Proof. By sub-exponential security of the puncturable PRF (with security parameter k), we have that for every $\mathsf{poly}(k)$-sized distinguisher D there exists a negligible function $\mu_8(\cdot)$ such that:

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,8}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,9}) = 1]\right| \leq \mu_8(2^{k^\epsilon})$$

  □

- $\mathsf{Hybrid}_{i,10} \equiv \mathsf{Hybrid}_i$.

  Claim 4.9. *For every* $\mathsf{poly}(2^{k^\epsilon})$*-sized distinguisher* D *there exists a negligible function* $\mu_9(\cdot)$ *such that:*

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,9}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,10}) = 1]\right| = \mu_9(2^{k^\epsilon})$$

  Proof. Note that now the programs $P^i_{pk,K,\mathcal{D},\mathcal{L}}$ and $\widetilde{P}^i_{pk,K,\mathcal{D},\mathcal{L},(x^*,e^*)}$ have identical functionality. Therefore, by sub-exponential security of indistinguishability obfuscation (with security parameter k), for every $\mathsf{poly}(k)$-sized distinguisher D there exists a negligible function $\mu_9(\cdot)$ such that:

  $$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_{i,9}) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i,10}) = 1]\right| \leq \mu_9(2^{k^\epsilon})$$

  □

By combining all the above claims, we have that for every $i \in [1, 2^{k'}]$ and every $\mathsf{poly}(2^{k^\epsilon})$-sized distinguisher D there exists a negligible function $\mu(\cdot)$ such that:

$$\left|\Pr[\mathsf{D}(\mathsf{Hybrid}_i) = 1] - \Pr[\mathsf{D}(\mathsf{Hybrid}_{i+1}) = 1]\right| \leq \mu(2^{k^\epsilon})$$

Moreover, recall that $\log \kappa = k^{\epsilon^2}$. We have that for every (non-uniform) $\mathsf{poly}(\kappa)$-sized adversary $\mathcal{A}$,

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_0) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{2^\kappa}) = 1] \right| \leq 2^{k'} \cdot \mu(2^{k^\epsilon}) = 2^{k^\epsilon} \cdot \mu(2^{k^\epsilon}) = \mathsf{negl}(2^{k^\epsilon}) = \mathsf{negl}(2^{\log \kappa^c})$$

which proves the lemma, as desired. □

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 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 $\phi(m) = 1$. After receiving the committer's message, $\mathcal{R}$ 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 5.1 (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 $\phi(m) = 1$,

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

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

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

- **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 4.1.

We define

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

- 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 5.1. *Construction 5.1 satisfies completeness according to Definition 5.1.*

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

Lemma 5.2. *Construction 5.1 satisfies soundness according to Definition 5.1.*

Proof. We prove that this lemma follows by the soundness of the NIDI according to Definition 4.2 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 $\kappa \in \mathbb{N}$,

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

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

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

which contradicts the soundness of NIDI, as desired. □

Lemma 5.3. *Construction 5.1 satisfies computational hiding according to Definition 4.2.*

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 $\mathcal{D}_{m_b}$ consists of instance-witness pairs (x, w) for $x = (\mathsf{Com}(m_b; r)), w = ((m_b, r))$.

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 4.2 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

6.1 Definitions

We now define CCA secure commitments, parts of some definitions are taken from [GKLW20]. This is a scheme where a commitment to message m under tag tag and randomness r is created as $\mathsf{CCACom}(\mathsf{tag}, m; r) \to \mathsf{com}$. For all $\mathsf{tag}_0, \mathsf{tag}_1, r_0, r_1$ and $m_0 \neq m_1$ we have that $\mathsf{CCACom}(\mathsf{tag}_0, m_0; r_0) \neq \mathsf{CCACom}(\mathsf{tag}_1, m_1; r_1)$.

Chosen commitment security, first defined in [CLP10], allows an adversary to give a challenge tag tag^* along with messages m_0, m_1 and receive a challenge commitment com^* to either m_0 or m_1 from the experiment. The adversary must then guess the message that was committed to with the aid of oracle access to an (inefficient) value function CCAVal where $\mathsf{CCAVal}(\mathsf{com})$ will return m if $\mathsf{CCACom}(\mathsf{tag}, m; r) \to \mathsf{com}$ for some r. The adversary is allowed oracle access to $\mathsf{CCAVal}(\cdot)$ for any $\mathsf{tag} \neq \mathsf{tag}^*$.

A CCA secure commitment is parameterized by a tag space of size $N = N(\kappa)$ where tags are in $[1, N]$. It consists of three algorithms:

$\mathsf{CCACom}(1^\kappa, \mathsf{tag}, m; r_C, r_R) \to \mathsf{com}$ is a randomized process between committer $\mathcal{C}$ and receiver $\mathcal{R}$ that takes as input the security parameter κ, $\mathsf{tag} \in [N]$, a message $m \in \{0,1\}^*$ and outputs a commitment com, including the tag tag. We denote the random coins of the committer and receiver explicitly as r_C and r_R respectively.

$\mathsf{CCAVal}(\mathsf{com}) \to m \cup \perp$ is an algorithm that takes in a commitment com and outputs either a message $m \in \{0,1\}^*$ or a reject symbol $\perp$.

Definition 6.1. A non-interactive CCA commitment is a tagged commitment as described above that involves a single message from the committer to the receiver, and that satisfies the following correctness, efficiency and security properties.

- **Correctness/Binding.** For all $m \in \{0,1\}^*$, $\mathsf{tag} \in [N]$ and r_C, r_R we have that

$$\mathsf{CCAVal}(\mathsf{CCACom}(1^\kappa, \mathsf{tag}, m; r_C, r_R)) = m.$$

  Moreover,

$$\Pr_{c \leftarrow \langle \mathcal{C}^*, \mathcal{R}(r_R) \rangle}[\nexists (m, r_C, \mathsf{tag}) \text{ s.t. } c = \mathsf{CCACom}(1^\kappa, \mathsf{tag}, m; r_C, r_R) \wedge \mathsf{CCAVal}(c) \neq \perp] = \mathsf{negl}(\kappa)$$

  where the probability is over the randomness of the receiver.

- **Efficiency.** CCACom runs in time $\mathsf{poly}(|m|, \kappa)$, while CCAVal runs in time $\mathsf{poly}(|m|, 2^\kappa)$.
- **Security.** We now define a CCA game between a challenger and an adversary. The game is parameterized by a security parameter κ. First, we require that for any string c for which there do not exist $m \in \{0,1\}^*, r, \mathsf{tag} \in [N]$ such that $c = \mathsf{CCACom}(1^\kappa, \mathsf{tag}, m; r)$, $\mathsf{CCAVal}(c)$ outputs $\perp$.

  1. The adversary sends a "challenge tag" $\mathsf{tag}^* \in [N]$.
  2. The adversary makes repeated commitment queries com. If $\mathsf{com.tag} = \mathsf{tag}^*$ the challenger responds with $\perp$. Otherwise it responds as
  $$\mathsf{CCAVal}(\mathsf{com}).$$
  3. For some p, the adversary sends two messages $m_0, m_1 \in \{0,1\}^p$.
  4. The challenger flips a coin $b \in \{0,1\}$ and sends $\mathsf{com}^* = \mathsf{CCACom}(\mathsf{tag}^*, m_b; r)$ for randomly chosen r.
  5. The adversary makes repeated commitment queries again, denoted by com. If $\mathsf{com.tag} = \mathsf{tag}^*$ the challenger responds with $\perp$. Otherwise it responds as
  $$\mathsf{CCAVal}(\mathsf{com}).$$
  6. The adversary finally outputs a guess b'.

  We require that for any polynomial-sized adversary $\mathcal{A}$ there exists a negligible function $\mu(\cdot)$ such that the adversary's advantage in the game, defined to be $\Pr[b' = b] - \frac{1}{2}$, is $\mu(\kappa)$.

We will assume that the message underlying any commitment generated with security parameter 1^κ can be retrieved with probability 1 by an algorithm running in time $q(2^\kappa)$ for some polynomial function q. We will refer to this as a trivial implementation of the CCAVal function.

Definition 6.2. A commitment scheme $(\mathsf{CCACom}, \mathsf{CCAVal})$ is said to be "same tag" CCA secure if for any poly-time non-uniform adversary $\mathcal{A}$ such that all commitment queries submitted by $\mathcal{A}$ are on the same tag, there exists a negligible function $\mathsf{negl}(\cdot)$ such that $\mathcal{A}$'s advantage in the above game with Step 1 removed is $\mathsf{negl}(\kappa)$.

Definition 6.3. A commitment scheme $(\mathsf{CCACom}, \mathsf{CCAVal})$ is said to be "same tag" CCA secure w.r.t. extraction if for any poly-time non-uniform adversary $\mathcal{A}$ such that all commitment queries submitted by $\mathcal{A}$ are on the same tag, there exists a negligible function $\mathsf{negl}(\cdot)$ such that $\mathcal{A}$'s advantage in a modified version of the above game with Step 1 removed is $\mathsf{negl}(\kappa)$.

The only modification is that the correctness/binding property is relaxed to *remove* the second requirement, and is defined as: for all $m \in \{0,1\}^*$, $\mathsf{tag} \in [N]$ and r_C, r_R we have that

$$\mathsf{CCAVal}(\mathsf{CCACom}(1^\kappa, \mathsf{tag}, m; r_C, r_R)) = m.$$

6.2 Tag Amplification: Construction and Analysis

In this section, we prove the following theorem.

Theorem 6.1. *Assume the existence of sub-exponentially secure indistinguishability obfuscation, sub-exponentially secure one-way functions and quasi-polynomially (i.e. $2^{\log \kappa^c}$-secure for some large enough $c > 1$) secure same-tag CCA commitments w.r.t. extraction (Definition 6.3) for tags in $[\log\log\log \kappa]$. Then there exist CCA commitments (Definition 6.1) for tags in 2^κ.*

We prove this theorem in two parts. In the first, we build a compiler that amplifies same-tag CCA commitments w.r.t. extraction for tags in $[t]$ to CCA commitments for tags in $[t]$. In the second we build a tag amplification compiler that amplifies CCA commitments for tags in $[t/2]$ for $t \leq \mathsf{poly}(\kappa)$ to CCA commitments for 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.

From same-tag CCA commitments w.r.t. extraction for tags in $[t]$ for $t \leq \mathsf{poly}(\kappa)$ to CCA commitments for tags in $[t]$. We prove the following theorem.

Lemma 6.1. *Assume the existence of 2^{k^ϵ}-secure indistinguishability obfuscation and one-way functions, and $2^{(\log k^c)}$ secure same-tag CCA commitments w.r.t. extraction for tags in $[t]$ where $t \leq \mathsf{poly}(\kappa)$, satisfying Definition 6.2 and where $c\epsilon > 1$. Then there exist $2^{(\log k^{c\epsilon})}$-secure CCA commitments for tags in $[t]$ satisfying Definition 6.1.*

Combining this lemma with (quasi-polynomially secure) same-tag base schemes, eg., those due to [KK19], yields CCA commitments for small tags from sub-exponential indistinguishability obfuscation, sub-exponential quantum-hard non-interactive commitments, and sub-exponential non-interactive commitments in BQP. Alternatively, on combining with the base scheme of [LPS17], we obtain CCA commitments for small tags from sub-exponential indistinguishability obfuscation and the [LPS17] variant of the (subexponential) time-lock puzzle assumption.

In what follows, let $\epsilon > 0$ and $c > 1$ be constants such that $\epsilon c \geq 1$ and:

- The same-tag CCA commitment w.r.t. extraction for small tags and security parameter κ is $2^{(\log \kappa)^c}$ secure and has a value algorithm CCAVal that runs in time at most $\mathsf{poly}(2^\kappa)$ on *valid* commitments.
- There exists a *subexponentially secure verifiable one-way puzzle* f such that for large enough security parameter k, the corresponding puzzle can be solved with probability at most $\mathsf{negl}(2^{k^\epsilon})$ by machines of size 2^{k^ϵ}.
- 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* statistically sound and *sub-exponentially* computationally witness indistinguishable 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

$$\mathcal{L}_{\mathsf{NIWI}} = \Big\{ \{(c_i, s_i)\}_{i\in[t-1]}, (pk, \mathsf{enc}, y) : \exists (M, r_1, \ldots, r_t, s, sk) \text{ s.t. } \Big(\forall i \in [t-1], c_i = \mathsf{ComSame}_{s_i}(M; r_i)\Big) \bigvee \Big((pk, sk) \leftarrow \mathsf{KeyGen}(s) \wedge \mathsf{Puzzle.Check}(y, f(\mathsf{Dec}_{sk}(c)))\Big)\Big\}$$

where s_i denotes a tag in $[t]$, and ComSame denotes the commit algorithm for an underlying CCA commitment with tags in $[t]$.

Construction 6.1. We now describe the CCACom and CCAVal algorithms for the scheme without the same tag restriction. We note that just like our commit-and-prove system described in the previous section, the commit phase ends *after* the receiver has queries the committer's program on a random input. The output of the commit phase is the output of such a receiver.

On input security parameter κ, we will set parameters of our building blocks as follows. Our verifiable one-way puzzle will have security parameter k_{f} set to $(\log \kappa)^c$. The CCA commitment for same 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 $(s_1, \ldots s_{t-1}) = [t] \setminus \{\mathsf{tag}\}$.
- The committer $\mathcal{C}(1^\kappa, M, \mathsf{tag})$ does the following:
  - Set $k = \kappa^{\frac{1}{\epsilon}}$ and $k_{\mathsf{f}} = (\log \kappa)^c$.
  - 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 7.
  - 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 $c = (\mathsf{tag}, pk, \widetilde{P})$ runs a randomized algorithm that does the following.
  - Sample $v \leftarrow \{0,1\}^\kappa$ and set $y = f(v)$.
  - Compute $\mathsf{out} = \mathsf{iO.Eval}(\widetilde{P}, y)$. Parse $\mathsf{out} = (x, e)$, $x = (d, pk, \mathsf{enc}, y)$ and $d = \{c_i\}_{i\in[t-1]}$.
  - Set $x' = \{(c_i, s_i)\}_{i\in[t-1]}, (pk, \mathsf{enc}, y)$. If $\mathsf{NIWI}.\mathcal{V}(1^k, x', e, \mathcal{L}_{\mathsf{NIWI}})$ rejects, output $\perp$ and stop.
  - Else output v, and for each $i \in [t-1]$, execute the receiver algorithm $\mathsf{ComSame}.\mathcal{R}(c_i)$. If any of these (t) algorithms output $\perp$, then output $\perp$ and stop.

– At the end of this process, the receiver either outputs $\perp$ or $(y, \tau_1, \ldots, \tau_t)$ where τ_i denotes the (non-$\perp$) outcome of $\mathsf{ComSame}.\mathcal{R}(c_i)$[6].

The CCAVal **Algorithm:** The CCAVal algorithm obtains as input a commitment string parsed as $\perp$ or $(\tau_1, \ldots, \tau_{t-1})$, generated as the output of the commit phase above, and does the following.

- On input a commitment string, if $\perp$, output $\perp$. Otherwise parse the string as $(\tau_1, \ldots, \tau_{t-1})$ and execute $\mathsf{ComSame.CCAVal}(\tau_1)$.

Hardwired: Public key pk, Puncturable PRF Key K, message $M \in \{0,1\}^p$, small tags $(s_1, \ldots s_t)$ corresponding to tag.

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

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

Figure 7: **Program** $P_{K,M,\mathsf{tag}}$

It is easy to see that the running time of CCAVal remains $\mathsf{poly}(|m|, 2^\kappa)$. The correctness/binding property follows based on soundness of the NIWI and the weak correctness/binding properties of the same-tag commitment w.r.t. extraction. Any adversary that (with non-negligible probability) produces a commitment c for which the NIWI verifies but where the underlying commitments are not well-formed, can be used to break the one-way puzzle identically to Section 5.

We now prove the following lemma, which implies Theorem 6.1.

Lemma 6.2. *Construction 6.1 satisfies* $2^{(\log \kappa)^{c\epsilon}}$*-CCA security according to Definition 6.1 for tags in* $[t]$*.*

Proof. To prove the lemma, we define a sequence of $(2^{k_\mathsf{f}} + 1)$ hybrid distributions, $(\mathsf{Hybrid}_0, \mathsf{Hybrid}_1, \ldots, \mathsf{Hybrid}_{2^{k_\mathsf{f}}})$ and prove 2^{k^ϵ}-indistinguishability between them all.
For each $j \in [0, 2^\kappa]$, the distribution Hybrid_j is defined as follows:

1. The adversary sends a "challenge tag" $\mathsf{tag}^* \in [N]$. Compute $(s_1^*, \ldots s_{t-1}^*)$ for tag^*.
2. The adversary makes repeated commitment queries where it sends an obfuscated progam com. If $\mathsf{com.tag} = \mathsf{tag}^*$ the challenger responds with $\perp$.

   Otherwise compute $(y, \tau_1, \ldots, \tau_t) \leftarrow \mathcal{R}(\mathsf{com})$ and $(s_1, \ldots s_t)$ for tag as described above. Compute $\iota \in [t]$ such that $s_\iota \notin \{s_1^*, \ldots, s_{t-1}^*\}$ and output

$$y, \mathsf{ComSame.CCAVal}(\tau_\iota).$$

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

3. The adversary sends two messages $m_0, m_1 \in \{0,1\}^p$.

4. The challenger sends $\mathsf{com}^* = \mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ for randomly chosen r, for the $\mathsf{CCACom}^{(j)}$ algorithm described below.

5. The adversary makes repeated commitment queries again, denoted by com. If $\mathsf{com.tag} = \mathsf{tag}^*$ the challenger responds with $\perp$.

   Otherwise compute $(y, \tau_1, \ldots, \tau_t) \leftarrow \mathcal{R}(\mathsf{com})$ and $(s_1, \ldots s_{t-1})$ for tag as described above. Compute $\iota \in [t]$ such that $s_\iota \notin \{s_1^*, \ldots, s_{t-1}^*\}$ and output

   $$y, \mathsf{ComSame.CCAVal}(\tau_\iota).$$

   (Note that s_ι this will equal the tag tag^* by construction.)

6. The adversary finally outputs its view.

For every $j \in [0, 2^{k_\mathsf{f}}]$, $\mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ is identical to $\mathsf{CCACom}(\mathsf{tag}^*, m_b; r)$ except that the challenger generates $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}; R)$, for the program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}$ described in Figure 8.

Hardwired: Public key pk, Puncturable PRF Key K, messages $m_0, m_1 \in \{0,1\}^p$, small tags $(s_1^*, \ldots s_{t-1}^*)$ corresponding to tag^*.

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

1. If $\mathsf{Puzzle.Ver}(y) \neq 1$, output $\perp$. Otherwise, continue.
2. Set $r = (r_1 || r_2 || \ldots || r_{t+1}) = \mathsf{PRF}(K, y)$.
3. If $y < j$, then for $i \in [t-1]$, set $c_i = \mathsf{ComSame}_{s_i^*}(m_1; r_i)$. Set $d = \{c_i\}_{i \in [t-1]}$.
4. If $y \geq j$, then for $i \in [t-1]$, set $c_i = \mathsf{ComSame}_{s_i^*}(m_0; r_i)$. Set $d = \{c_i\}_{i \in [t-1]}$.
5. Set $\mathsf{enc} = \mathsf{Enc}_{pk}(0^\kappa; r_t)$.
6. Set $x = d, (pk, \mathsf{enc}, y), w = (M, r_1, \ldots, r_{t-1})$.
7. Compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_\mathsf{NIWI}; r_{t+1})$ and output (x, e).

Figure 8: **Program** $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}$

First, we note that Hybrid_0 is identical to the CCA game where the challenge commitment commits to m_0, except that the adversary's oracle queries in Step 2 are decrypted by running $\mathsf{ComSame.CCAVal}(\tau_\iota)$, as opposed to running $\mathsf{ComSame.CCAVal}(\tau_1)$. We argue that the adversary's view is indistinguishable with error $\mathsf{negl}(2^{k_\mathsf{f}^\epsilon}) = \mathsf{negl}(2^{(\log \kappa)^{c\epsilon}})$, between Hybrid_0 and the CCA game with challenge commitment to m_0, by relying on the soundness of the NIWI. Specifically, by the soundness of NIWI, any adversary that distinguishes the two hybrids makes at least one decommitment query for which, on verifier randomness y, the commitment program outputs enc as an

encryption of s^* s.t. $\mathsf{Puzzle.Ver}(y, s^*) \neq 1$. By a non-uniform argument identical to that in Lemma 4.2, such an adversary can be used to build a $\mathsf{poly}(\kappa) = \mathsf{poly}(2^{k_\mathsf{f}^{1/c}}) < \mathsf{poly}(2^{k_\mathsf{f}^\epsilon})$-adversary that contradicts one-wayness of f with probability $1/\mathsf{poly}(2^{k_\mathsf{f}^\epsilon})$.

By a similar argument, the adversary's view is indistinguishable with error $\mathsf{negl}(2^{k_f^\epsilon}) = \mathsf{negl}(2^{(\log \kappa)^{c\epsilon}})$, between $\mathsf{Hybrid}_{2^{k_\mathsf{f}}}$ and the CCA game with challenge commitment to m_1. We therefore conclude that indistinguishability between the CCA games with challenge commitments to m_0 versus m_1 respectively is implied by indistinguishability between Hybrid_0 and $\mathsf{Hybrid}_{2^{k_\mathsf{f}}}$. Towards establishing the latter, we begin by proving the following claim.

Claim 6.1. *For every $j \in [1, 2^{k_\mathsf{f}}]$ and every (non-uniform) $\mathsf{poly}(\kappa)$-sized adversary $\mathcal{A}$ there exists a negligible function $\mu(\cdot)$ such that:*

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_j) = 1] \right| = \mu(2^{(\log \kappa)^{c\epsilon}})$$

To prove this claim, we fix arbitrary $j \in [1, 2^{k_\mathsf{f}}]$ and consider two cases. In the first, when $\mathsf{Puzzle.Ver}(j) \neq 1$, we have that by (subexponential) security of iO,

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_j) = 1] \right| = \mu(2^{k^\epsilon}) = \mu(2^\kappa)$$

When $\mathsf{Puzzle.Ver}(j) = 1$, we consider a sequence of sub-hybrids between Hybrid_{j-1} and Hybrid_j called $(\mathsf{Hybrid}_{j-1,1}, \ldots, \mathsf{Hybrid}_{j-1,10})$ where:

$$\mathsf{Hybrid}_{j-1,1} \equiv \mathsf{Hybrid}_{j-1} \text{ and } \mathsf{Hybrid}_{j-1,10} \equiv \mathsf{Hybrid}_j$$

- $\mathsf{Hybrid}_{j-1,1} \equiv \mathsf{Hybrid}_{j-1}$.
- $\mathsf{Hybrid}_{j-1,2}$ is identical to $\mathsf{Hybrid}_{j-1,1}$ except the challenge commitment $\mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ is generated as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,1}$ and $\mathsf{Hybrid}_{j-1,2}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^* || \ldots || r_{t+1}^*) = \mathsf{PRF}(K, i)$.
  - For $i \in [t-1]$ set $c_i^* = \mathsf{ComSame}_{s_i^*}(m_0; r_i^*)$. Set $d^* = \{c_i^*\}_{i \in [t-1]}$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(0^\kappa; r_{t-1}^*)$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (m_0, r_1^*, \ldots, r_{t-1}^*, 0^{2k})$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_\mathsf{NIWI}; r_{t+1}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 9.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

  Indistinguishability from the previous hybrid follows by (subexponential) security of iO.

  Claim 6.2.

$$\mathsf{Hybrid}_{j-1,1} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,2}$$

Hardwired: Public key pk, Puncturable PRF Key K, messages $m_0, m_1 \in \{0,1\}^p$, small tags $(s_1^*, \ldots s_{t-1}^*)$ corresponding to tag^*.

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

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $r = (r_1 || r_2 || \ldots || r_t || r_{t+1}) = \mathsf{PRF}(K, y)$.
3. If $y = j$, output (x^*, e^*).
4. If $y < j$, then for $i \in [t-1]$, set $c_i = \mathsf{ComSame}_{s_i^*}(m_1; r_i)$. Set $d = \{c_i\}_{i \in [t-1]}$.
5. If $y > j$, then for $i \in [t-1]$, set $c_i = \mathsf{ComSame}_{s_i^*}(m_0; r_i)$. Set $d = \{c_i\}_{i \in [t-1]}$.
6. Set $\mathsf{enc} = \mathsf{Enc}_{pk}(0^\kappa; r_t)$.
7. Set $x = d, (pk, \mathsf{enc}, y), w = (M, r_1, \ldots, r_{t-1}, 0^{2k})$.
8. Compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_{\mathsf{NIWI}}; r_{t+1})$ and output (x, e).

Figure 9: **Program** $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}$

Proof. Note that the programs have identical functionality. Furthermore, note that answers to all CCAVal queries on valid commitments are computable in time $2^\kappa = 2^{k^\epsilon}$. Therefore, by sub-exponential security of indistinguishability obfuscation (with security parameter k), for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_1(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,2}) = 1] \right| \leq \mu_1(2^{k^\epsilon}) = \mu_1(2^\kappa)$$

□

- $\mathsf{Hybrid}_{j-1,3}$ is identical to $\mathsf{Hybrid}_{j-1,2}$ except that $(r_1^* || \ldots || r_{t+1}^*) \leftarrow \{0,1\}^{2\kappa+k}$.

  Indistinguishability from the previous hybrid follows by (subexponential) security of the puncturable PRF.

  Claim 6.3.

  $$\mathsf{Hybrid}_{j-1,2} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,3}$$

  Proof. Note that answers to all CCAVal queries are computable in time $2^\kappa \leq 2^{k^\epsilon}$. Therefore, by sub-exponential security of the puncturable PRF (with security parameter k), we have directly that for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_2(\cdot)$ such that:

  $$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,2}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,3}) = 1] \right| \leq \mu_2(2^{k^\epsilon}) = \mu_2(2^\kappa)$$

  □

- $\mathsf{Hybrid}_{j-1,4}$ is is identical to $\mathsf{Hybrid}_{j-1,1}$ except $\mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,3}$ and $\mathsf{Hybrid}_{j-1,4}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^* || \ldots || r_{t+1}^*) = \mathsf{PRF}(K, i)$.
  - Compute (in time $\mathsf{poly}(2^\kappa)$) a solution s^* such that $\mathsf{Puzzle.Check}(s^*, j) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r_t^*)$.
  - For $i \in [t-1]$ set $c_i^* = \mathsf{ComSame}_{s_i^*}(m_0; r_i^*)$. Set $d^* = \{c_i^*\}_{i \in [t]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (m_0, r_1^*, \ldots, r_{t-1}^*, 0^{2k})$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t+1}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 9.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

  Indistinguishability from the previous hybrid follows by (subexponential) CPA security of the encryption scheme.

  Claim 6.4.

$$\mathsf{Hybrid}_{j-1,3} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,4}$$

  Proof. We will prove this claim based on the sub-exponential IND-CPA security of the encryption scheme against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,3}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

  Then there exists a non-uniform adversary $\mathcal{A}'$ that non-uniformly fixes $s^* = f^{-1}(j)$, then begins the experiment. It obtains c^* from the CPA challenger as either $\mathsf{Enc}_{pk}(0^\kappa; r_{t+1}^*)$ or $\mathsf{Enc}_{pk}(s^*; r_{t+1}^*)$. Note that answers to all CCAVal queries are computable in time $2^\kappa = 2^{k^\epsilon}$. It completes the rest of the experiment according to $\mathsf{Hybrid}_{j-1,3}$ except setting c^* according to the ciphertext obtained from the external challenger. It then mirrors the output of $\mathcal{A}$ given the resulting distribution, which implies that

$$\left| \Pr[\mathcal{A}'(\mathsf{Enc}_{pk}(0^\kappa; r_{t+1}^*)) = 1] - \Pr[\mathcal{A}'(\mathsf{Enc}_{pk}(s^*; r_{t+1}^*)) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

  which gives a contradiction, as desired. Therefore, by sub-exponential IND-CPA security of the encryption scheme (with security parameter k), for every $i \in [1, 2^\kappa]$ there exists a negligible function $\mu_3(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,3}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] \right| \leq \mu_3(2^{k^\epsilon})$$

  □

- $\mathsf{Hybrid}_{j-1,5}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,4}$ and $\mathsf{Hybrid}_{j-1,5}$)
  - Set $k = \kappa^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^* || \ldots || r_{t+1}^*) = \mathsf{PRF}(K, i)$.
  - Compute (in time $\mathsf{poly}(2^\kappa)$) a solution s^* such that $\mathsf{Puzzle.Check}(s^*, j) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r_{t+1}^*)$.
  - For $i \in [t-1]$ set $c_i^* = \mathsf{ComSame}_{s_i^*}(m_0; r_i^*)$. Set $d^* = \{c_i^*\}_{i \in [t]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $\underline{w^* = (0^{p(\kappa)+(t)\cdot\kappa}, s, sk).}$
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t+2}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 9.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

Indistinguishability from the previous hybrid follows by (subexponential) security of the NIWI.

Claim 6.5.

$$\mathsf{Hybrid}_{j-1,4} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,5}$$

Proof. We will prove this claim based on the sub-exponential witness indistinguishability of the NIWI against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,5}) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

Then there exists a non-uniform adversary $\mathcal{A}'$ that non-uniformly fixes $s_1^* = f^{-1}(j)$, then begins the experiment. It obtains e^* from the CPA challenger either using witness w^* corresponding to $\mathsf{Hybrid}_{j-1,4}$, or using witness w^* corresponding to $\mathsf{Hybrid}_{j-1,5}$. It completes the rest of the experiment according to $\mathsf{Hybrid}_{j-1,4}$ except setting e^* according to the NIWI obtained from the external challenger. Here note that answers to all CCAVal queries are computable in time $2^\kappa = 2^{k^\epsilon}$.

$\mathcal{A}'$ finally mirrors the output of $\mathcal{A}$ given the resulting distribution, which implies that

$$\left| \Pr[\mathcal{A}'(w^* = (m_1, r_1^*, \ldots, r_t^*, 0^{\kappa+k})) = 1] - \Pr[\mathcal{A}'(w^* = (0^{p(\kappa)+(t)\cdot\kappa}, s^*, r_{t+1}^*)) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

which gives a contradiction, as desired. Therefore, by sub-exponential security of the NIWI (with security parameter k), for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_4(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,5}) = 1] \right| \leq \mu_4(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{j-1,6}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,5}$ and $\mathsf{Hybrid}_{j-1,6}$)
  - Set $k = \kappa^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^* || \ldots || r_{t+1}^*) = \mathsf{PRF}(K, i)$.
  - Compute (in time $\mathsf{poly}(2^\kappa)$) a solution s^* such that $\mathsf{Puzzle.Check}(s^*, j) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r_t^*)$.
  - For $i \in [t-1]$ set $c_i^* = \mathsf{ComSame}_{s_i^*}(m_1; r_i^*)$. Set $d^* = \{c_i^*\}_{i \in [t-1]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (0^{p(\kappa)+(t)\cdot\kappa}, s, sk)$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t+1}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 9.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

Indistinguishability from the previous hybrid follows by CCA security of ComSame.

Claim 6.6.

$$\mathsf{Hybrid}_{j-1,5} \approx_{(2^{(\log \kappa)^c})} \mathsf{Hybrid}_{j-1,6}$$

Proof. We will prove this claim based on $(2^{(\log \kappa)^c})$-non-malleability of ComSame against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,5}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,6}) = 1] \right| \geq \frac{1}{p(2^{(\log \kappa)^c})}$$

Then there exists a (non-uniform) adversary $\mathcal{A}'$ that non-uniformly fixes a pre-challenge transcript as well as $s_1^* \in f^{-1}(j)$, then begins the experiment. It obtains d^* from an external challenger sampled either as $\{c_i^* = \mathsf{ComSame}_{s_i^*}(m_0; r_i^*)\}_{i \in [t]}$ as in $\mathsf{Hybrid}_{j-1,5}$, or as $\{c_i^* = \mathsf{ComSame}_{s_i^*}(m_1; r_i^*)\}_{i \in [t]}$ as in $\mathsf{Hybrid}_{j-1,6}$[7]. It completes the rest of the experiment according to $\mathsf{Hybrid}_{j-1,5}$ except setting d^* according to the sample obtained from the external challenger.

It relies on the oracle of ComSame to answer CCAVal queries (this is possible since the CCAVal algorithm simply returns the output of $\mathsf{ComSame.CCAVal}(\tau_\iota)$), which by construction of the hybrid always uses a different tag than all challenge tags.

It then mirrors the output of $\mathcal{A}$ given the resulting distribution, which implies that

$$\left| \Pr[\mathcal{A}'(\{c_i^* = \mathsf{ComSame}_{s_i^*}(m_0; r_i^*)\}_{i \in [t]}) = 1] - \Pr[\mathcal{A}'(\{c_i^* = \mathsf{ComSame}_{s_i^*}(m_1; r_i^*)\}_{i \in [t]}) = 1] \right| \geq \frac{1}{p(2^{(\log \kappa)^c})}$$

which gives a contradiction to the non-malleability of ComSame, as desired. □

[7] One can also imagine a more fine-grained sequence of hybrids where these commitments are replaced one by one.

- $\mathsf{Hybrid}_{j-1,7}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,6}$ and $\mathsf{Hybrid}_{j-1,7}$)
  - Set $k = \kappa^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*||\ldots||r_t^*) = \mathsf{PRF}(K, i)$.
  - Compute (in time $\mathsf{poly}(2^\kappa)$) a solution s^* such that $\mathsf{Puzzle.Check}(s^*, j) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r_{t+1}^*)$.
  - For $i \in [t-1]$ set $c_i^* = \mathsf{ComSame}_{s_i^*}(m_1; r_i^*)$. Set $d^* = \{c_i^*\}_{i\in[t-1]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $\underline{w^* = (m_1, r_1^*, \ldots, r_{t-1}^*, 0^{2k}).}$
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t+1}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 9.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

  Indistinguishability (with error $\mathsf{negl}(2^{k^\epsilon}) = \mathsf{negl}(2^\kappa)$) from the previous hybrid follows by (subexponential) security of the NIWI, similar to the proof of indistinguishability between $\mathsf{Hybrid}_{j-1,4}$ and $\mathsf{Hybrid}_{j-1,5}$.

- $\mathsf{Hybrid}_{j-1,8}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,7}$ and $\mathsf{Hybrid}_{j-1,8}$)
  - Set $k = \kappa^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*||\ldots||r_{t+1}^*) = \mathsf{PRF}(K, i)$.
  - ~~Compute (in time $\mathsf{poly}(2^\kappa)$) a solution s^* such that $\mathsf{Puzzle.Check}(s^*, j) = 1$.~~
  - $\underline{\text{Set } \mathsf{enc}^* = \mathsf{Enc}_{pk}(0; r_t^*).}$
  - For $i \in [t-1]$ set $c_i^* = \mathsf{ComSame}_{s_i^*}(m_1; r_i^*)$. Set $d^* = \{c_i^*\}_{i\in[t-1]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (m_1, r_1^*, \ldots, r_{t-1}^*, 0^{2k})$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t+1}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 9.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

  Indistinguishability (with error $\mathsf{negl}(2^{k^\epsilon}) = \mathsf{negl}(2^\kappa)$) from the previous hybrid follows by (subexponential) CPA security of the encryption scheme, similar to the proof of indistinguishability between $\mathsf{Hybrid}_{j-1,3}$ and $\mathsf{Hybrid}_{j-1,4}$.

- $\mathsf{Hybrid}_{j-1,9}$ is identical to $\mathsf{Hybrid}_{j-1,8}$ except that $(r_1^*||\ldots||r_{t+1}^*) = \mathsf{PRF}(K, j)$.

  Indistinguishability (with error $\mathsf{negl}(2^{k^\epsilon}) = \mathsf{negl}(2^\kappa)$) from the previous hybrid follows by (subexponential) security of the puncturable PRF, similar to the proof of indistinguishability between $\mathsf{Hybrid}_{j-1,2}$ and $\mathsf{Hybrid}_{j-1,3}$.

- $\mathsf{Hybrid}_{j-1,10} \equiv \mathsf{Hybrid}_j$.

  Indistinguishability (with error $\mathsf{negl}(2^{k^\epsilon}) = \mathsf{negl}(2^\kappa)$) from the previous hybrid follows by (subexponential) security of iO, similar to the proof of indistinguishability between $\mathsf{Hybrid}_{j-1,1}$ and $\mathsf{Hybrid}_{j-1,2}$.

We thus have that for every $j \in [1, 2^{k_\mathsf{f}}]$ there exists a negligible function $\mu(\cdot)$ such that:

$$\left|\Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_j) = 1]\right| \leq \mu(2^{(\log\kappa)^c})$$

For any (non-uniform) $\mathsf{poly}(\kappa)$-sized distinguisher $\mathcal{A}$, let μ_{max} denote the maximum value of the distinguishing advantage $\mu(\cdot)$ between any two consecutive hybrids Hybrid_j and Hybrid_{j+1}, which is again a negligible function. Then we have that

$$\left|\Pr[\mathcal{A}(\mathsf{Hybrid}_0) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{2^{k_\mathsf{f}}}) = 1]\right| \leq 2^{k_\mathsf{f}} \cdot \mu_{\mathsf{max}}(2^{(\log\kappa)^c})$$

$$= 2^{(\log\kappa)^c} \cdot \mathsf{negl}(2^{(\log\kappa)^c}) = \mathsf{negl}(2^{(\log\kappa)^c})$$

This combined with the analysis above implies that the adversary's advantage in the CCA hiding game is at most $\mathsf{negl}(2^{(\log\kappa)^{c\epsilon}})$ which completes the proof of the lemma, as desired. □

From CCA commitments for tags in $[t/2]$ for $t \leq \mathsf{poly}(\kappa)$ to CCA commitments for tags in $[T]$ where $T = \binom{t}{t/2}$. We prove the following theorem.

Lemma 6.3. *Assume the existence of 2^{k^ϵ}-secure indistinguishability obfuscation and one-way functions, and $2^{(\log k^c)}$ secure CCA commitments for tags in $[t/2]$ for $t \leq \mathsf{poly}(\kappa)$, satisfying Definition 6.1 and where $c\epsilon > 1$. Then there exist $2^{(\log k^{c\epsilon})}$-secure CCA commitments for tags in $\binom{t}{t/2}$ satisfying Definition 6.1.*

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

- The CCA commitment for small tags and security parameter κ is $2^{(\log\kappa)^c}$ secure and has a "brute-force" recovery algorithm CCAVal that recovers the value underlying any commitment, and runs in time at most $\mathsf{poly}(2^\kappa)$.
- There exists a *subexponentially secure verifiable one-way puzzle* that with security parameter k is 2^{k^ϵ} one-way.
- 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.

As before, note that sub-exponential public-key encryption and NIWIs can be based on sub-exponential iO and sub-exponential one-way functions/permutations [SW14, BP15], and sub-exponential puncturable PRFs can be based on sub-exponential one-way functions.

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

$$\begin{aligned}\mathcal{L}_{\mathsf{NIWI}} = \Big\{ & \{(c_i, s_i)\}_{i\in[t/2]}, (pk, \mathsf{enc}, y) : \exists (M, r_1, \ldots, r_{t/2}, s, sk) \text{ s.t.} \\ & \Big(\forall i \in [t/2], c_i = \mathsf{ComSmall}_{s_i}(M; r_i)\Big) \\ & \bigvee \Big((pk, sk) \leftarrow \mathsf{KeyGen}(s) \wedge \mathsf{Puzzle.Check}(y, f(\mathsf{Dec}_{sk}(c))) = 1\Big)\Big\}\end{aligned}$$

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 6.2. We now describe the CCACom and CCAVal algorithms for our 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 $(\log \kappa)^c$. 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 $i = \mathsf{tag}$.[8] 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 $k_{\mathsf{f}} = (\log \kappa)^c$.

[8]Here, we use a different tag encoding scheme due to [KS17] that offers a slightly more optimized way to the same effect as the DDN encoding [DDN91] 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.

- 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 10.
- 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 $\mathsf{out} = \mathsf{iO.Eval}(\widetilde{P}, y)$. 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 $\mathsf{ComSmall}.\mathcal{R}(c_i)$[9].

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.

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

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

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

1. If $\mathsf{Puzzle.Ver}(y) \neq 1$, output $\perp$. Otherwise, continue.
2. Set $r = (r_1 || r_2 || \ldots || r_{t/2} || r_{t/2+2}) = \mathsf{PRF}(K, y)$.
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 $x = d, (pk, \mathsf{enc}, y), w = (M, r_1, \ldots, r_{t/2}, 0^{2k})$.
6. Compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+2})$ and output (x, e).

Figure 10: **Program** $P_{K,M,\mathsf{tag}}$

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

Proof of Security The perfect correctness, efficiency and binding properties follow identically to the same-tag setting. We now prove the following lemma. We will need to apply the lemma recursively 4 times to obtain Theorem 6.1.

Lemma 6.4. *Construction 6.1 satisfies $2^{(\log \kappa)^{c\epsilon}}$-CCA security according to Definition 6.1 for tags in $[T]$ where $T = \binom{t}{t/2}$.*

Proof. To prove this lemma, we define a sequence of $(2^{k_{\mathsf{f}}} + 1)$ hybrid distributions, $(\mathsf{Hybrid}_0, \mathsf{Hybrid}_1, \ldots, \mathsf{Hybrid}_{2^{k_{\mathsf{f}}}})$ and prove indistinguishability between them all.
For each $j \in [0, 2^{k_{\mathsf{f}}}]$, the distribution Hybrid_j is defined as follows:

1. The adversary sends a "challenge tag" $\mathsf{tag}^* \in [N]$.

   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 $i = \mathsf{tag}^*$. Let this element be denoted by $(s_1^*, \ldots s_{t/2}^*)$.

2. The adversary makes repeated commitment queries com. If $\mathsf{com.tag} = \mathsf{tag}^*$ the challenger responds with $\perp$.

   Otherwise 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 $i = \mathsf{tag}$. Let this element be denoted by $(s_1, \ldots s_{t/2})$.

   Compute $\iota \in [T]$ such that $s_\iota \notin \{s_1^*, \ldots, s_{t/2}^*\}$ and output

   $$\mathsf{ComSmall.CCAVal}(\tau_\iota).$$

3. The adversary sends two messages $m_0, m_1 \in \{0,1\}^p$.

4. The challenger sends $\mathsf{com}^* = \mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ for randomly chosen r, for the $\mathsf{CCACom}^{(j)}$ algorithm described below.

5. The adversary makes repeated commitment queries again, denoted by com. If $\mathsf{com.tag} = \mathsf{tag}^*$ the challenger responds with $\perp$. If $\mathsf{com.tag} = \mathsf{tag}^*$ the challenger responds with $\perp$.

   Otherwise 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 $i = \mathsf{tag}$. Let this element be denoted by $(s_1, \ldots s_{t/2})$.

   Compute $\iota \in [T]$ such that $s_\iota \notin \{s_1^*, \ldots, s_{t/2}^*\}$ and output

   $$\mathsf{ComSmall.CCAVal}(\tau_\iota).$$

6. The adversary finally outputs its view.

For every $j \in [0, 2^{k_{\mathsf{f}}}]$, $\mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ is identical to $\mathsf{CCACom}(\mathsf{tag}^*, m_b; r)$ except that the challenger generates $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}; R)$, for the program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}$ described in Figure 11.

First, we note that Hybrid_0 is identical to the CCA game where the challenge commitment commits to m_0, except that the adversary's oracle queries in Step 2 are decrypted by running $\mathsf{ComSmall.CCAVal}(\tau_\iota)$, as opposed to running $\mathsf{ComSmall.CCAVal}(\tau_1)$. We argue that the adversary's

Hardwired: Public key pk, Puncturable PRF Key K, messages $m_0, m_1 \in \{0,1\}^p$, small tags $(s_1^*, \ldots s_{t/2}^*)$ corresponding to tag^*.

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

1. If $y \notin \mathsf{Range}(f)$, output $\perp$. Otherwise, continue.
2. Set $r = (r_1 || r_2 || \ldots || r_{t/2} || r_{t/2+2}) = \mathsf{PRF}(K, y)$.
3. If $y < j$, then for $i \in [t/2]$, set $c_i = \mathsf{ComSmall}_{s_i^*}(m_1; r_i)$. Set $d = \{c_i\}_{i \in [t/2]}$.
4. If $y \geq j$, then for $i \in [t/2]$, set $c_i = \mathsf{ComSmall}_{s_i^*}(m_0; r_i)$. Set $d = \{c_i\}_{i \in [t/2]}$.
5. Set $\mathsf{enc} = \mathsf{Enc}_{pk}(0^\kappa; r_{t/2+1})$.
6. Set $x = d, (pk, \mathsf{enc}, y), w = (M, r_1, \ldots, r_{t/2})$.
7. Compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_\mathsf{NIWI}; r_{t/2+2})$ and output (x, e).

Figure 11: **Program** $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}$

view is indistinguishable (with error $\mathsf{negl}(2^{\log \kappa^{c\epsilon}})$) between Hybrid_0 and the CCA game with challenge commitment to m_0, by relying on the soundness of the NIWI. Specifically, by the soundness of NIWI, any adversary that distinguishes the two hybrids makes at least one decommitment query for which, on verifier randomness y, the commitment program outputs enc as an encryption of s^* s.t. $f(s^*) = y$. By a non-uniform argument identical to that in Lemma 4.2, such an adversary can be used to build a $\mathsf{poly}(\kappa) = \mathsf{poly}(2^{k_\mathsf{f}^{1/c}}) < \mathsf{poly}(2^{k_\mathsf{f}^\epsilon})$-adversary that contradicts one-wayness of f with probability $1/\mathsf{poly}(2^{k_\mathsf{f}^\epsilon})$.

By a similar argument, the adversary's view is indistinguishable (with error $\mathsf{negl}(2^{(\log \kappa)^{c\epsilon}})$ between $\mathsf{Hybrid}_{2^{k_\mathsf{f}}}$ and the CCA game with challenge commitment to m_1. We therefore conclude that indistinguishability between the CCA games with challenge commitments to m_0 versus m_1 respectively is implied by indistinguishability between Hybrid_0 and $\mathsf{Hybrid}_{2^{k_\mathsf{f}}}$. Towards establishing the latter, we begin by proving the following claim.

Claim 6.7. *For every $j \in [1, 2^{k_\mathsf{f}}]$ and every (non-uniform) $\mathsf{poly}(\kappa)$-sized adversary $\mathcal{A}$ there exists a negligible function $\mu(\cdot)$ such that:*

$$\Big| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_j) = 1] \Big| = \mu(2^{k^\epsilon})$$

To prove this claim, we fix arbitrary $j \in [1, 2^{k_\mathsf{f}}]$ and consider two cases. In the first, when $\mathsf{Puzzle.Ver}(j) \neq 1$, we have that by (subexponential) security of iO,

$$\Big| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_j) = 1] \Big| = \mu(2^{k^\epsilon}) = \mu(2^\kappa)$$

When $\mathsf{Puzzle.Ver}(j) = 1$, we consider a sequence of sub-hybrids between Hybrid_{j-1} and Hybrid_j called $(\mathsf{Hybrid}_{j-1,1}, \ldots, \mathsf{Hybrid}_{j-1,10})$ where:

$$\mathsf{Hybrid}_{j-1,1} \equiv \mathsf{Hybrid}_{j-1} \text{ and } \mathsf{Hybrid}_{j-1,10} \equiv \mathsf{Hybrid}_j$$

- $\mathsf{Hybrid}_{j-1,1} \equiv \mathsf{Hybrid}_{j-1}$.
- $\mathsf{Hybrid}_{j-1,2}$ is identical to $\mathsf{Hybrid}_{j-1,1}$ except the challenge commitment $\mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ is generated as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,1}$ and $\mathsf{Hybrid}_{j-1,2}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*||\ldots||r_{t/2+1}^*) = \mathsf{PRF}(K, i)$.
  - For $i \in [t/2]$ set $c_i^* = \mathsf{ComSmall}_{s_i^*}(m_0; r_i^*)$. Set $d^* = \{c_i^*\}_{i\in[t/2]}$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(0^\kappa; r_{t/2+1}^*)$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (m_0, r_1^*, \ldots, r_{t/2}^*, 0^{2k})$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+2}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 12.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

Hardwired: Public key pk, Puncturable PRF Key K, messages $m_0, m_1 \in \{0,1\}^p$, small tags $(s_1^*, \ldots s_{t/2}^*)$ corresponding to tag^*.

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

1. If $\mathsf{Puzzle.Ver}(y) \neq 1$, output $\perp$. Otherwise, continue.
2. Set $r = (r_1||r_2||\ldots||r_{t/2}||r_{t/2+1}) = \mathsf{PRF}(K, y)$.
3. If $y = j$, output (x^*, e^*).
4. If $y < j$, then for $i \in [t/2]$, set $c_i = \mathsf{ComSmall}_{s_i^*}(m_1; r_i)$. Set $d = \{c_i\}_{i\in[t/2]}$.
5. If $y > j$, then for $i \in [t/2]$, set $c_i = \mathsf{ComSmall}_{s_i^*}(m_0; r_i)$. Set $d = \{c_i\}_{i\in[t/2]}$.
6. Set $\mathsf{enc} = \mathsf{Enc}_{pk}(0^\kappa; r_2)$.
7. Set $x = d, (pk, \mathsf{enc}, y), w = (M, r_1, \ldots, r_{t/2}, 0^{2k})$.
8. Compute $e = \mathsf{NIWI}.\mathcal{P}(1^k, x, w, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+1})$ and output (x, e).

Figure 12: **Program** $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*}$

Claim 6.8.

$$\mathsf{Hybrid}_{j-1,1} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,2}$$

Proof. Note that the programs have identical functionality. Furthermore, note that answers to all CCAVal queries are computable in time $2^\kappa \leq 2^{\kappa^\epsilon}$. Therefore, by sub-exponential security of indistinguishability obfuscation (with security parameter k), for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_1(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,2}) = 1] \right| \leq \mu_1(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{j-1,3}$ is identical to $\mathsf{Hybrid}_{j-1,2}$ except that $(r_1^*||\ldots||r_{t/2+2}^*) \leftarrow \{0,1\}^{2\kappa+k}$.

  Claim 6.9.

  $$\mathsf{Hybrid}_{j-1,2} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,3}$$

  Proof. Note that answers to all CCAVal queries are computable in time $2^\kappa \leq 2^{\kappa^\epsilon}$. Therefore, by sub-exponential security of the puncturable PRF (with security parameter k), we have directly that for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_2(\cdot)$ such that:

  $$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,2}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,3}) = 1] \right| \leq \mu_2(2^{k^\epsilon})$$

  □

- $\mathsf{Hybrid}_{j-1,4}$ is is identical to $\mathsf{Hybrid}_{j-1,1}$ except $\mathsf{CCACom}^{(j)}(\mathsf{tag}^*, m_0, m_1; r)$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,3}$ and $\mathsf{Hybrid}_{j-1,4}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^*||\ldots||r_{t/2+1}^*) = \mathsf{PRF}(K, i)$.
  - Compute (in time upto 2^κ) value s^* such that $\mathsf{Puzzle.Check}(j, s^*) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r_{t/2+1}^*)$.
  - For $i \in [t/2]$ set $c_i^* = \mathsf{ComSmall}_{s_i^*}(m_0; r_i^*)$. Set $d^* = \{c_i^*\}_{i \in [t/2]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (m_0, r_1^*, \ldots, r_{t/2}^*, 0^{2k})$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+2}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 12.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

  Claim 6.10.

  $$\mathsf{Hybrid}_{j-1,3} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,4}$$

Proof. We will prove this claim based on the sub-exponential IND-CPA security of the encryption scheme against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,3}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

Then there exists a non-uniform adversary $\mathcal{A}'$ that non-uniformly fixes $s^* = f^{-1}(j)$, then begins the experiment. It obtains c^* from the CPA challenger as either $\mathsf{Enc}_{pk}(0^\kappa; r^*_{t/2+1})$ or $\mathsf{Enc}_{pk}(s^*; r^*_{t/2+1})$. Note that answers to all CCAVal queries are computable in time $2^\kappa \leq 2^{\kappa^\epsilon}$. It completes the rest of the experiment according to $\mathsf{Hybrid}_{j-1,3}$ except setting c^* according to the ciphertext obtained from the external challenger. It then mirrors the output of $\mathcal{A}$ given the resulting distribution, which implies that

$$\left| \Pr[\mathcal{A}'(\mathsf{Enc}_{pk}(0^\kappa; r^*_{t/2+1})) = 1] - \Pr[\mathcal{A}'(\mathsf{Enc}_{pk}(s^*; r^*_{t/2+1})) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

which gives a contradiction, as desired. Therefore, by sub-exponential IND-CPA security of the encryption scheme (with security parameter k), for every $i \in [1, 2^\kappa]$ there exists a negligible function $\mu_3(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,3}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] \right| \leq \mu_3(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{j-1,5}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,4}$ and $\mathsf{Hybrid}_{j-1,5}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r^*_1 || \ldots || r^*_{t/2+1}) = \mathsf{PRF}(K, i)$.
  - Compute (in time upto 2^κ) value s^* such that $\mathsf{Puzzle.Check}(j, s^*) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r^*_{t/2+1})$.
  - For $i \in [t/2]$ set $c^*_i = \mathsf{ComSmall}_{s^*_i}(m_0; r^*_i)$. Set $d^* = \{c^*_i\}_{i \in [t/2]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $\underline{w^* = (0^{p(\kappa)+(t/2)\cdot\kappa}, s, sk).}$
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r^*_{t/2+2})$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 12.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

Claim 6.11.

$$\mathsf{Hybrid}_{j-1,4} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,5}$$

Proof. We will prove this claim based on the sub-exponential witness indistinguishability of the NIWI against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,5}) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

Then there exists a non-uniform adversary $\mathcal{A}'$ that non-uniformly fixes $s_1^* = f^{-1}(j)$, then begins the experiment. It obtains e^* from the CPA challenger either using witness w^* corresponding to $\mathsf{Hybrid}_{j-1,4}$, or using witness w^* corresponding to $\mathsf{Hybrid}_{j-1,5}$. It completes the rest of the experiment according to $\mathsf{Hybrid}_{j-1,4}$ except setting e^* according to the NIWI obtained from the external challenger. Here note that answers to all CCAVal queries are computable in time $2^\kappa \leq 2^{\kappa^\epsilon}$.

$\mathcal{A}'$ finally mirrors the output of $\mathcal{A}$ given the resulting distribution, which implies that

$$\left| \Pr[\mathcal{A}'(w^* = (m_1, r_1^*, \ldots, r_{t/2}^*, 0^{\kappa+k})) = 1] - \Pr[\mathcal{A}'(w^* = (0^{p(\kappa)+(t/2)\cdot\kappa}, s^*, r_{t/2+1}^*)) = 1] \right| \geq \frac{1}{p(2^{k^\epsilon})}$$

which gives a contradiction, as desired. Therefore, by sub-exponential security of the NIWI (with security parameter k), for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_4(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,5}) = 1] \right| \leq \mu_4(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{j-1,6}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,5}$ and $\mathsf{Hybrid}_{j-1,6}$)
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^* || \ldots || r_{t/2+1}^*) = \mathsf{PRF}(K, i)$.
  - Compute (in time upto 2^κ) value s^* such that $\mathsf{Puzzle.Check}(j, s^*) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r_{t/2+1}^*)$.
  - <u>For $i \in [t/2]$ set $c_i^* = \mathsf{ComSmall}_{s_i^*}(m_1; r_i^*)$.</u> Set $d^* = \{c_i^*\}_{i \in [t/2]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (0^{p(\kappa)+(t/2)\cdot\kappa}, s, sk)$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+2}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 12.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

Claim 6.12.

$$\mathsf{Hybrid}_{j-1,5} \approx_{(2^{(\log \kappa)^c})} \mathsf{Hybrid}_{j-1,6}$$

Proof. We will prove this claim based on $(2^{(\log \kappa)^c})$-non-malleability of $\mathsf{ComSmall}$ against non-uniform adversaries. Towards a contradiction, suppose there exists a distinguisher $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,5}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,6}) = 1] \right| \geq \frac{1}{p(2^{(\log \kappa)^c})}$$

Then there exists a (non-uniform) adversary $\mathcal{A}'$ that non-uniformly fixes a pre-challenge transcript as well as $s_1^* \in f^{-1}(j)$, then begins the experiment. It obtains d^* from an external challenger sampled either as $\{c_i^* = \mathsf{ComSmall}_{s_i^*}(m_0; r_i^*)\}_{i\in[t/2]}$ as in $\mathsf{Hybrid}_{j-1,5}$, or as $\{c_i^* = \mathsf{ComSmall}_{s_i^*}(m_1; r_i^*)\}_{i\in[t/2]}$ as in $\mathsf{Hybrid}_{j-1,6}$[10]. It completes the rest of the experiment according to $\mathsf{Hybrid}_{j-1,5}$ except setting d^* according to the sample obtained from the external challenger.

It relies on the oracle of $\mathsf{ComSmall}$ to answer CCAVal queries (this is possible since the CCAVal algorithm simply returns the output of $\mathsf{ComSmall.CCAVal}(\tau_\iota)$, which by construction of the hybrid always uses a different tag than all challenge tags.

It then mirrors the output of $\mathcal{A}$ given the resulting distribution, which implies that

$$\left| \Pr[\mathcal{A}'(\{c_i^* = \mathsf{ComSmall}_{s_i^*}(m_0; r_i^*)\}_{i\in[t/2]}) = 1] - \Pr[\mathcal{A}'(\{c_i^* = \mathsf{ComSmall}_{s_i^*}(m_1; r_i^*)\}_{i\in[t/2]}) = 1] \right| \geq \frac{1}{p(2^{(\log \kappa)^c})}$$

which implies a contradiction to the $(2^{(\log \kappa)^c})$ non-malleability of $\mathsf{ComSmall}$, as desired. □

- $\mathsf{Hybrid}_{j-1,7}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,6}$ and $\mathsf{Hybrid}_{j-1,7}$)
  - Set $k = \kappa^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^* || \ldots || r_{t/2+1}^*) = \mathsf{PRF}(K, i)$.
  - Compute (in time upto 2^κ) value s^* such that $\mathsf{Puzzle.Check}(j, s^*) = 1$.
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(s^*; r_{t/2+1}^*)$.
  - For $i \in [t/2]$ set $c_i^* = \mathsf{ComSmall}_{s_i^*}(m_1; r_i^*)$. Set $d^* = \{c_i^*\}_{i\in[t/2]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $\underline{w^* = (m_1, r_1^*, \ldots, r_{t/2}^*, 0^{2k})}$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+2}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 12.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

[10] One can also imagine a more fine-grained sequence of hybrids where these commitments are replaced one by one.

Claim 6.13.

$$\mathsf{Hybrid}_{j-1,6} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,7}$$

Proof. By sub-exponential witness indistinguishability of the NIWI against non-uniform adversaries (and following an identical argument to that of the indistinguishability between $\mathsf{Hybrid}_{j-1,4}$ and $\mathsf{Hybrid}_{j-1,5}$), for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_6(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,6}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,7}) = 1] \right| \leq \mu_6(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{j-1,8}$ is defined as follows: (we underline the difference between $\mathsf{Hybrid}_{j-1,7}$ and $\mathsf{Hybrid}_{j-1,8}$)
  - Set $k = \kappa^{\frac{1}{\epsilon}}$.
  - Sample $s \leftarrow \{0,1\}^k$ and set $(pk, sk) \leftarrow \mathsf{KeyGen}(s)$.
  - Sample $K \leftarrow \{0,1\}^k$ and $R \leftarrow \{0,1\}^k$.
  - Set $(r_1^* || \ldots || r_{t/2+1}^*) = \mathsf{PRF}(K, i)$.
  - ~~Compute (in time upto 2^κ) value s^* such that Puzzle.Check$(j, s^*) = 1$.~~
  - Set $\mathsf{enc}^* = \mathsf{Enc}_{pk}(0; r_{t/2+1}^*)$.
  - For $i \in [t/2]$ set $c_i^* = \mathsf{ComSmall}_{s_i^*}(m_1; r_i^*)$. Set $d^* = \{c_i^*\}_{i \in [t/2]}$.
  - Set $x^* = (d^*, pk, \mathsf{enc}^*, j)$, $w^* = (m_1, r_1^*, \ldots, r_{t/2}^*, 0^{2k})$.
  - Compute $e^* = \mathsf{NIWI}.\mathcal{P}(1^k, x^*, w^*, \mathcal{L}_{\mathsf{NIWI}}; r_{t/2+2}^*)$.
  - Generate program $P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}$ defined in Figure 12.
  - Compute $\widetilde{P} = \mathsf{iO}(P^{(j)}_{pk,K,m_0,m_1,\mathsf{tag}^*,(x^*,e^*)}; R)$.
  - Output $c = (\mathsf{tag}^*, pk, \widetilde{P})$.

Claim 6.14.

$$\mathsf{Hybrid}_{j-1,7} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,8}$$

Proof. By sub-exponential IND-CPA security of the encryption scheme against non-uniform adversaries (and following an identical argument to that of the indistinguishability between $\mathsf{Hybrid}_{j-1,3}$ and $\mathsf{Hybrid}_{j-1,4}$), for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_7(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,3}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,4}) = 1] \right| \leq \mu_7(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{j-1,9}$ is identical to $\mathsf{Hybrid}_{j-1,8}$ except that $(r_1^* || \ldots || r_{t/2+1}^*) = \mathsf{PRF}(K, j)$.

Claim 6.15.

$$\mathsf{Hybrid}_{j-1,8} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,9}$$

Proof. By sub-exponential security of the puncturable PRF (with security parameter k), we have directly that for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_8(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,8}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,9}) = 1] \right| \leq \mu_8(2^{k^\epsilon})$$

□

- $\mathsf{Hybrid}_{j-1,10} \equiv \mathsf{Hybrid}_j$.

  Claim 6.16.

  $$\mathsf{Hybrid}_{j-1,9} \approx_{2^{k^\epsilon}} \mathsf{Hybrid}_{j-1,10}$$

  Proof. Note that the hybrids are indistinguishable except they obfuscate two different programs that have identical functionality. Therefore, by sub-exponential security of indistinguishability obfuscation (with security parameter k), for every $j \in [1, 2^\kappa]$ there exists a negligible function $\mu_9(\cdot)$ such that:

  $$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,9}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1,10}) = 1] \right| \leq \mu_9(2^{k^\epsilon})$$

  □

By combining all the above claims, we have that for every $j \in [1, 2^{k_\mathsf{f}}]$ there exists a negligible function $\mu(\cdot)$ such that:

$$\left| \Pr[\mathcal{A}(\mathsf{Hybrid}_{j-1}) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_j) = 1] \right| \leq \mu(2^{(\log \kappa)^c})$$

For any (non-uniform) $\mathsf{poly}(k)$-sized distinguisher $\mathcal{A}$, let μ_{max} denote the maximum value of the distinguishing advantage $\mu(\cdot)$ between any two consecutive hybrids Hybrid_j and Hybrid_{j+1}, which is again a negligible function. Then we have that

$$\begin{aligned} \left| \Pr[\mathcal{A}(\mathsf{Hybrid}_0) = 1] - \Pr[\mathcal{A}(\mathsf{Hybrid}_{2^{k_\mathsf{f}}}) = 1] \right| &\leq 2^{k_\mathsf{f}} \cdot \mu_{\mathsf{max}}(2^{(\log \kappa)^c}) \\ &= 2^{(\log \kappa)^c} \cdot \mathsf{negl}(2^{(\log \kappa)^c}) = \mathsf{negl}(2^{(\log \kappa)^c}) \end{aligned}$$

This combined with the analysis above implies that the adversary's advantage in the CCA hiding game is at most $\mathsf{negl}(2^{(\log \kappa)^{c\epsilon}})$ which completes the proof of the lemma, as desired. □

Lin, Pass and Soni [LPS17] note that by the work of Bitansky et. al. [BGJ+16], time-lock puzzles needded for their base scheme can be based on (sub-exponential) indistinguishability obfuscation and the existence of a parallel-time hard language. Combining this with our results implies CCA commitments according to our notion of non-interactivity based on (sub-exponential) one-way functions, indistinguishability obfuscation and the existence of a parallel-time hard language.

Acknowledgments

We thank the anonymous Eurocrypt reviewers for their suggestions. We are grateful to Ran Canetti, Suvradip Chakraborty, Oxana Poburinnaya and Manoj Prabhakaran for helpful discussions and to Nishant Kumar for detailed feedback on an initial draft of this work.

References

[Agr19] Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. In *EUROCRYPT*, 2019.

[AJL+19] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In *CRYPTO*, 2019.

[AJN+16] Prabhanjan Ananth, Aayush Jain, Moni Naor, Amit Sahai, and Eylon Yogev. Universal constructions and robust combiners for indistinguishability obfuscation and witness encryption. In *Advances in Cryptology - CRYPTO 2016, Proceedings, Part II*, pages 491–520, 2016.

[AP20] Shweta Agrawal and Alice Pellet-Mary. Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE. In *EUROCRYPT*, 2020.

[Bar02] Boaz Barak. Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. In *FOCS 2002*, pages 345–355, 2002.

[BDGM20a] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate io from homomorphic encryption schemes. In *EUROCRYPT*, 2020.

[BDGM20b] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure LWE suffices. *IACR Cryptol. ePrint Arch.*, 2020.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, 2012.

[BGI+17] Saikrishna Badrinarayanan, Sanjam Garg, Yuval Ishai, Amit Sahai, and Akshay Wadia. 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*, pages 275–303, 2017.

[BGJ+16] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 345–356. ACM, 2016.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In Moses Charikar and Edith Cohen, editors, *STOC 2019*, pages 1091–1102. ACM, 2019.

[BL18] Nir Bitansky and Huijia Lin. One-message zero knowledge and non-malleable commitments. In *Theory of Cryptography Conference, TCC 2018, Goa, India, November 11-14, 2018, Proceedings*, 2018.

[BM14] Christina Brzuska and Arno Mittelbach. 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*, pages 142–161, 2014.

[BOV07] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

[BP04] Boaz Barak and Rafael Pass. 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*, pages 121–132, 2004.

[BP12] Nir Bitansky and Omer Paneth. Point obfuscation and 3-round zero-knowledge. In *TCC 2012*, pages 190–208, 2012.

[BP15] Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015*, volume 9015 of *Lecture Notes in Computer Science*, pages 401–427. Springer, 2015.

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 474–502. Springer, 2016.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *STOC 2020*, pages 269–279. ACM, 2020.

[BST16] Mihir Bellare, Igors Stepanovs, and Stefano Tessaro. 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*, pages 542–564, 2016.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 235–244. ACM, 2000.

[CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. 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*, FOCS '10, pages 541–550, 2010.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. 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*, pages 66–92, 2015.

[COSV16] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Concurrent non-malleable commitments (and more) in 3 rounds. In Robshaw and Katz [RK16], pages 270–299.

[COSV17] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Four-round concurrent non-malleable commitments from one-way functions. In *Annual International Cryptology Conference*, pages 127–157. Springer, 2017.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-Malleable Cryptography (Extended Abstract). In *STOC 1991*, 1991.

[DHRW16] Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs. Spooky encryption and its applications. In Robshaw and Katz [RK16], pages 93–122.

[DK18] Apoorvaa Deshpande and Yael Kalai. Proofs of ignorance and applications to 2-message witness hiding. *IACR Cryptology ePrint Archive*, 2018:896, 2018.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999.

[FS90] Uriel Feige and Adi Shamir. 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*, pages 416–426, 1990.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. 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*, pages 74–94, 2014.

[GJLS20] Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. *IACR Cryptol. ePrint Arch.*, 2020.

[GKLW20] Rachit Garg, Dakshita Khurana, George Lu, and Brent Waters. Black-box non-interactive non-malleable commitments. Cryptology ePrint Archive, Report 2020/1197, 2020. `https://eprint.iacr.org/2020/1197`.

[GLOV12] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *FOCS*, 2012.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

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

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

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

[Goy11] Vipul Goyal. Constant Round Non-malleable Protocols Using One-way Functions. In *STOC 2011*, pages 695–704. ACM, 2011.

[GP20] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. *IACR Cryptol. ePrint Arch.*, 2020.

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *STOC*, pages 1128–1141, New York, NY, USA, 2016. ACM.

[GR19] Vipul Goyal and Silas Richelson. Non-malleable commitments using goldreich-levin list decoding. In David Zuckerman, editor, *FOCS 2019*, pages 686–699. IEEE Computer Society, 2019.

[GRRV14] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. In *FOCS 2014*, pages 41–50, 2014.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017*, volume 10402 of *Lecture Notes in Computer Science*, pages 158–189. Springer, 2017.

[JLMS19] Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. How to leverage hardness of constant-degree expanding polynomials over r to build io. In *EUROCRYPT*, 2019.

[JLS20] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. Cryptology ePrint Archive, Report 2020/1003, 2020. `https://eprint.iacr.org/2020/1003`.

[Khu17] Dakshita Khurana. Round optimal concurrent non-malleability from polynomial hardness. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017*, volume 10678 of *Lecture Notes in Computer Science*, pages 139–171. Springer, 2017.

[KK19] Yael Tauman Kalai and Dakshita Khurana. Non-interactive non-malleability from quantum supremacy. In *CRYPTO 2019*, pages 552–582, 2019.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In Umans [Uma17], pages 564–575.

[KZ20] Benjamin Kuykendall and Mark Zhandry. Towards non-interactive witness hiding. Cryptology ePrint Archive, Report 2020/1205, 2020. `https://eprint.iacr.org/2020/1205`.

[LP] Huijia Lin and Rafael Pass. Constant-round Non-malleable Commitments from Any One-way Function. In *STOC 2011*, pages 705–714.

[LP09] Huijia Lin and Rafael Pass. Non-malleability Amplification. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 189–198, 2009.

[LPS17] Huijia Lin, Rafael Pass, and Pratik Soni. Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In Umans [Uma17], pages 576–587.

[LPV] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent Non-malleable Commitments from Any One-Way Function. In *TCC 2008*, pages 571–588.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *EUROCRYPT 2003*, pages 160–176, 2003.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive One-Way Functions and Applications. In *Advances in Cryptology — CRYPTO '08*, pages 57–74, 2008.

[PR05] Rafael Pass and Alon Rosen. Concurrent Non-Malleable Commitments. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of ComputerScience*, FOCS '05, pages 563–572, 2005.

[PR08] Rafael Pass and Alon Rosen. New and Improved Constructions of Nonmalleable Cryptographic Protocols. *SIAM J. Comput.*, 38(2):702–752, 2008.

[PW10] Rafael Pass and Hoeteck Wee. Constant-round non-malleable commitments from sub-exponential one-way functions. In *EUROCRYPT 2010*, pages 638–655, 2010.

[RK16] Matthew Robshaw and Jonathan Katz, editors. *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*. Springer, 2016.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *STOC 2014*, pages 475–484. ACM, 2014.

[Uma17] Chris Umans, editor. *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. IEEE Computer Society, 2017.

[Wee10] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *FOCS 2010*, pages 531–540, 2010.

[WW20] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. *IACR Cryptol. ePrint Arch.*, 2020.