Simultaneous Amplification: The Case of Non-Interactive Zero-Knowledge

Vipul Goyal[1], Aayush Jain[2], and Amit Sahai[2]

[1] CMU
vipul@cmu.edu
[2] UCLA
{aayushjain,sahai}@cs.ucla.edu

Abstract. In this work, we explore the question of simultaneous privacy and soundness amplification for non-interactive zero-knowledge *argument* systems (NIZK). We show that any δ_s-sound and δ_z-zero-knowledge NIZK candidate satisfying $\delta_s + \delta_z = 1 - \epsilon$, for any constant $\epsilon > 0$, can be turned into a computationally sound and zero-knowledge candidate with the only extra assumption of a subexponentially secure public-key encryption.

We develop novel techniques to leverage the use of leakage simulation lemma (Jetchev-Peitzrak TCC 2014) to argue amplification. A crucial component of our result is a new notion for secret sharing NP instances. We believe that this may be of independent interest.

To achieve this result we analyze following two transformations:

- **Parallel Repetition:** We show that using parallel repetition any δ_s-sound and δ_z-zero-knowledge NIZK candidate can be turned into (roughly) δ_s^n-sound and $1-(1-\delta_z)^n-$zero-knowledge candidate. Here n is the repetition parameter.
- **MPC based Repetition:** We propose a new transformation that amplifies zero-knowledge in the same way that parallel repetition amplifies soundness. We show that using this any δ_s-sound and δ_z-zero-knowledge NIZK candidate can be turned into (roughly) $1-(1-\delta_s)^n-$sound and $2 \cdot \delta_z^n-$zero-knowledge candidate.

Then we show that using these transformations in a zig-zag fashion we can obtain our result. Finally, we also present a simple transformation which directly turns any NIZK candidate satisfying $\delta_s, \delta_z < 1/3 - 1/\mathsf{poly}(\lambda)$ to a secure one.

1 Introduction

Amplification techniques are central to cryptography and complexity theory. The basic approach is to first obtain a construction which achieves the desired property but "with some error". In the next step, the initial construction is compiled into a final one which achieves a much smaller error parameter. This is often done by having the final construction invoke the initial construction several times. Thus, we say that the compiler is used to "amplify" the desired security property by reducing or eliminating the error.

Amplification techniques have served as a gateway towards significant progress in cryptography (as well as complexity theory). As an example, all the initial constructions of zero-knowledge proofs were obtained via soundness amplification. First a zero-knowledge proof with a significant soundness error was obtained, and then, either sequential or parallel repetition was used to reduce the soundness error to negligible. Within the area of complexity theory, soundness amplification of interactive protocols has played a central role in various important advances such as in probabilistically checkable proofs (PCPs) and hardness of approximation. Another rich line of research studies hardness amplifications and its various connections to coding theory [23]. Not only does amplification help us develop our understanding of the assumptions that the primitives can be based upon, it is an invaluable tool to construct complex primitives. A notable recent success is that of [1], where, a security amplification theorem for functional encryption was pivotal to constructing first obfuscation scheme from succinctly stated and instance-independent assumptions.

Simultaneous Amplification. The problem of amplification is known to be especially challenging if one tries to amplify multiple properties simultaneously. A well-known example of this concerns oblivious transfer (OT). Weak oblivious transfer considers a situation where the security of both the sender and the receiver is prone to failure: a malicious sender might have advantage ϵ_1 in guessing the choice bit of the receiver, while, a malicious receiver might have advantage ϵ_2 in guessing the input bit of the sender which it did not select. A rich body of literature has studied amplification techniques to obtain a full-fledged OT given a weak OT [11, 36, 37]. These amplification techniques have proven to be useful in a variety of problem including cryptography from noisy channels [10, 24], and, multi-party differentially private protocols [16].

Our Focus: Amplification for Non-Interactive Zero-Knowledge Arguments. In this work, we study simultaneous amplification of soundness and zero-knowledge. As discussed before, a number of works have studied amplifying soundness for interactive proofs (and arguments), and as such, some of these results apply even to zero-knowledge protocols. However what if the zero-knowledge property had an error to start with?[3] More concretely, we are interested in the following question.

Suppose one is given a non-interactive zero-knowledge (NIZK) argument with soundness error δ_s and zero-knowledge error δ_z, is it possible to compile it into a full-fledged secure non-interactive zero-knowledge argument system?

In more detail, consider $(\delta_s, \delta_z) - \mathsf{NIZK}$, where δ_s is the probability with which any efficient adversary can win in the soundness experiment. Similarly δ_z is the advantage with which any efficient adversary can distin-

[3] The most important reason to consider this is that it may be easier to construct NIZK with relaxed soundness and zero knowledge requirements. Indeed, in the past, even slight relaxations of zero knowledge, such as ϵ-zero knowledge [12], have led to simpler protocols.

guish between simulated and honest proofs. (Please see next Section for more formal definitions.)

We first observe that it is trivial to construct a NIZK candidate where $\delta_s + \delta_z = 1$. This can be constructed by sampling a crs as a bit which is set to 1 with probability δ_s and 0 otherwise. If $\mathsf{crs} = 1$, the verification algorithm is supposed to verify the string $\perp$ as a valid proof, and otherwise it should only verify a valid NP witness of the given instance as a valid proof. This system is trivially δ_s-sound, and a simulator that just outputs $\perp$ achieves $(1 - \delta_s) = \delta_z$-zero knowledge. Of course, we cannot expect this trivial system to be amplifiable.

Thus the above question can be rewritten as:

Is it possible to amplify $(\delta_s, \delta_z)-$non-interactive zero-knowledge, where $\delta_s + \delta_z = 1 - \epsilon$ for any constant $\epsilon > 0$, to full-fledged non-interactive zero-knowledge under standard cryptographic assumptions?

To our knowledge, this question has not been studied before. We believe the question of amplifying soundness and zero-knowledge simultaneously is a basic one which is interesting in its own right.

We answer this in the affirmative, by giving such a transformation assuming that subexponentially secure public-key encryption exists. Formally, we prove the following theorem:

Theorem 1. *Assume a subexponentially secure* PKE *scheme, and a* NIZK *candidate Π with δ_s-soundness and δ_z-zero-knowledge where δ_z, δ_s are in $(0, 1)$ with $\delta_s + \delta_z < 1$ for all polynomial time adversaries, then there exists a fully secure* NIZK *candidate against all polynomial time adversaries.*

NIZK is a basic primitive in cryptography which is widely used to obtain the constructions of other basic and advanced primitives. Yet, despite much effort, NIZK is unfortunately known from very few assumptions: [19, 17, 18, 35, 6, 5, 13, 8, 32]. E.g., we do not yet know a NIZK system that is proven secure under the assumption of (even subexponentially secure) DDH or LWE. Given this state of the art our work gives an alternative easier path to construct NIZKs since now one only needs to obtain constructions satisfying $\delta_s + \delta_z < 1$ (as opposed to constructions achieving the standard notion where δ_s and δ_z are negligible).

We develop several novel techniques to prove our result. An interesting primitive we introduce is the notion of *secret sharing NP instances.*

Secret Sharing of NP Instances. Towards constructing a NIZK amplification theorem, our main technical tool is what we call *secret sharing of NP Instances*. Very roughly, this allows breaking a (statement, witness) pair into n different (statement, witness) pairs such that each pair can then be verified individually while no single pair (or upto a threshold t of pairs) reveals any information about the original witness. Additionally it allows that if more than some other threshold t' of instances are satisfiable then x itself should be satisfiable. We believe secret sharing of NP instances to be a novel conceptual tool which is of independent interest. Please see Section 2 for more details and a technical overview.

Inspired by [1],to prove our result, we use and build upon the ideas used to prove the dense model theorem [33].

Related Works. We are not aware of any prior works on on amplifying zero-knowledge and soundness simultaneously. However there have been a number of prior works on amplification in general. Soundness amplification of interactive proofs has been studied in a rich line of works [3, 21, 31, 7]. As mentioned before, another line of research studies amplification and combiners for oblivious transfer [20, 30]. Another related result concerns "polarization" (which is a type of simultaneous amplification) of complete problems for SZK [34].

2 Technical Overview

Suppose we have been given a NIZK candidate where $\delta_s + \delta_z = 1 - \epsilon$ for any constant $\epsilon > 0$, how do we construct one where $\delta_s + \delta_z < \mathsf{negl}$?
We study three basic transformations and analyze their effects on the parameters (δ_s, δ_z).

- **Parallel Repetition:** We show that this transformation converts a NIZK candidate with parameters (δ_s, δ_z) to roughly $(\delta_s^n, 1-(1-\delta_z)^n)$, where n is some parameter which can be set to be any polynomial in λ. Thus, this transformation boosts soundness but worsens zero-knowledge property.
- **MPC-based Repetition:** We show that this transformation converts a NIZK candidate with parameters (δ_s, δ_z) to roughly $(1-(1-\delta_s)^n, 2\cdot\delta_z^n)$, where n is some parameter which can be set to be any polynomial in λ. Thus, this transformation boosts zero-knowledge but worsens soundness property.
- **MPC-based Amplification:** This transformation converts a NIZK candidate with parameters (δ_s, δ_z) satisfying $\delta_s, \delta_z < 1/3 - 1/\mathsf{poly}(\lambda)$ to a fully secure NIZK candidate.

Then, we show using these three transformation how to take any (δ_S, δ_z) NIZK satisfying $\delta_s + \delta_z = 1 - \epsilon$ for any constant $\epsilon > 0$, and output a fully secure NIZK candidate.

2.1 Parallel Repetition

As a warm up that is useful to introduce some of the ideas we will use, let us first consider the standard parallel repetition transformation. The construction is as follows. Let Π be the underlying candidate. The setup algorithm of the transformed candidate $\Pi_{||}$ does the following. It computes $\Pi.\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{crs}_i$ for $i \in [n]$, where n is some repetition parameter. It sets $\mathsf{crs} = (\mathsf{crs}_1, ..., \mathsf{crs}_n)$. The prover then proves $x \in L$ using the given witness w, employing each crs_i independently to form n proofs $\pi = (\pi_1, ..., \pi_n)$. Finally, the verification succeeds if each π_i verifies with respect to crs_i. We discuss at a high level various properties associated with this scheme.

$(\delta_s^n + \mathsf{negl})$-*Soundness:* This is already known from many of the previous works (such as [7]) that soundness is amplified this way for any non-interactive argument system upon parallel repetition. The (overly simplified) intuition is the following. If the soundness error is δ_s, then there exists a hardcore set S of size $(1-\delta_s)\cdot|\mathcal{R}|$ where $\mathcal{R}$ is the space of randomness for the coins of $\Pi.\mathsf{Setup}$. This hardcore set has the property that if crs is generated using randomness from this set, then any adversary $\mathcal{A}$ of some large bounded size, will only break soundness with a small probability ϵ_s. Then, if we have n parallel systems, the probability that no crs_i is sampled using randomness from this set S falls as δ_s^n. In order to prove this formally, in spirit of [7], we prove the following lemma. The details can be found in the full version.

Lemma 1. *Let $F:\{0,1\}^\lambda \to \{0,1\}^l$ be a function where $l=\mathsf{poly}(\lambda)$ and $E:\{0,1\}^{\lambda+l+r(\lambda)} \to \{0,1\}$ be a circuit of size e. Let $\delta \geq \epsilon \in (0,1)$ and $s,s'>0$ be functions of λ. If for all circuits $C:\{0,1\}^{l(\lambda)} \to \{0,1\}^{r(\lambda)}$ of size s we have*

$$\Pr_{u \xleftarrow{\$} \{0,1\}^\lambda}[E(u,F(u),C(F(u)))=1] \leq \delta$$

Then there exists a set S of size $|S|=(1-\delta)2^\lambda$ and a polynomial $s_{overhead}(\lambda)$ (independent of s,s' and e) such that: For all circuits $C':\{0,1\}^{l(\lambda)} \to \{0,1\}^{r(\lambda)}$ of size less than $s'=\frac{s\epsilon(1-\delta)}{\delta}-e-s_{overhead}$

$$\Pr_{u \xleftarrow{\$} S}[E(u,F(u),C'(F(u)))=1] \leq \epsilon$$

Roughly F is the algorithm $\Pi.\mathsf{Setup}$, C is the adversary and E is the algorithm that tests if soundness is broken. Since the size of E is a factor that determines the size of the adversary that can be handled, we want to keep it small. Thus, we work with a NIZK argument of knowledge candidate instead of a NIZK candidate. This is done by using a public key of a public key encryption scheme (generated at setup) to encrypt the witness, and the NIZK system is used to prove that this encrypted witness is valid. Then, it becomes possible to check if the soundness of Π was broken by simply decrypting the witness and testing its validity for the instance x. This ensures that size of E is polynomially bounded.

$1-(1-\delta_z)^n$*−Zero-Knowledge:* Since parameters are very crucial to achieve our result, we also have to show that zero-knowledge is not completely destroyed by parallel repetition. This is so that we can tolerate some amount of degradation. To achieve this theorem, we prove and rely on the following lemma:

Theorem 2. *Fix $1^\lambda, x \in \mathsf{SAT}$ with $|x|=\mathsf{poly}(\lambda)$ and corresponding witness u. Define two functions E_b for $b\in\{0,1\}$, that takes as input $\{0,1\}^{\ell_b}$. Here ℓ_b is the length of randomness required to compute the following.*
Consider the following process:
1. *Sample $r_1, r_2 \leftarrow \{0,1\}^{\ell_0}$.*
2. *Run $\Pi.\mathsf{Setup}(1^\lambda; r_1) \to \mathsf{crs}$.*
3. *Run $\Pi.\mathsf{Prove}(\mathsf{crs}, x, u) \to \pi$.*
4. *Sample $\tilde{r} \leftarrow \{0,1\}^{\ell_1}$*

5. Compute $(\widetilde{crs}, \widetilde{\pi}) \leftarrow \mathit{\Pi}.\mathsf{Sim}(1^\lambda, x; \tilde{r})$*.*
6. E_0 *on input* $(r_1, r_2) \in \{0,1\}^{\ell_0}$ *outputs* (crs, π)*.*
7. E_1 *on input* $\tilde{r} \in \{0,1\}^{\ell_1}$ *outputs* $(\widetilde{\mathsf{crs}}, \widetilde{\pi})$*.*

If $\mathit{\Pi}$ *satisfies* $\delta-$*zero knowledge for all adversaries of size* s*, then, there exists two computable (not necessarily efficient) measures* $\mathcal{M}_0$ *and* $\mathcal{M}_1$ *(*$\mathcal{M}_b$ *defined over* $\{0,1\}^{\ell_b}$ *for* $b \in \{0,1\}$*) of density exactly* $1-\delta$ *such that, for all circuits* $\mathcal{A}$ *of size* $s' < s\epsilon^2/128(\ell_0+\ell_1+1)$*,*

$$\left| \Pr_{(r_1,r_2)\leftarrow \mathcal{D}_{\mathcal{M}_0}} [\mathcal{A}(E_0(r_1,r_2)) = 1] - \Pr_{\tilde{r}\leftarrow \mathcal{D}_{\mathcal{M}_1}} [\mathcal{A}(E_1(\tilde{r})) = 1] \right| < \epsilon$$

Here both measures may depend on (x, u)

This theorem roughly says that there exists two measures $\mathcal{S}_0$ and $\mathcal{S}_1$ of density exactly $1-\delta_z$ such that the when the proof and setup is done using randomness from $\mathcal{S}_0$ then for a bounded adversary it is computationally indistinguishable from the case when the crs and the proof is simulated using randomness from $\mathcal{S}_1$. Thus, using this one can show that if randomness from for all n parallel systems is generated from this measure $\mathcal{S}_0$, then it is computationally close to the case when the proofs for all n systems are simulated using randomness from $\mathcal{S}_1$. Since the densities of $\mathcal{S}_0$ and $\mathcal{S}_1$ is exactly equal to $1-\delta_z$, this allows to (informally) argue that the zero-knowledge parameter of the resulting candidate is (very roughly) bounded by $1-(1-\delta_z)^n + \mathsf{negl}$. Here is the formal theorem statement:

Theorem 3. *Assuming* $\mathit{\Pi}$ *is* δ_z-*zero-knowledge against adversaries of size* s*,* $\mathit{\Pi}_{\|}$ *is* $(1-(1-\delta_z)^n)+O(n\cdot\epsilon)-$*zero-knowledge against adversaries of size* $s' = s\cdot\epsilon^2/\mathsf{poly}(\lambda)$ *for some fixed polynomial* poly*.*

The details can be found in the full version. Given that we have a way to reduce soundness error while not letting zero-knowledge degrade too much, we turn to the next question:

Is there a natural transformation that amplifies zero-knowledge, while not degrading soundness too much?

We consider this question and propose a very natural transformation to achieve this. We call it MPC-based repetition because it achieves parameters similar to parallel repetition where the roles of zero-knowledge error and soundness error are switched, but it is based on secure multi-party computation (MPC) protocols instead of simple parallel invocation of the NIZK candidate. In another words, it is a natural dual of the construction above.

2.2 MPC-based Repetition:

A First Idea: Before we describe our approach, we first describe a seemingly more natural approach that we do not know how to analyze: Specifically, consider the new candidate which runs $\mathit{\Pi}.\mathsf{Setup} \rightarrow \mathsf{crs}_i$ for $i \in [2]$.

The prover first computes π_1 with respect to crs_1 for the given NP relation. The prover then considers the NP relation that is satisfied with a "witness" that is any valid proof with respect to the verification procedure of the NIZK candidate. Then, the prover can use π_1 as a witness to to satisfy this new relation, and thus compute π_2 with respect to crs_2. The output is then set as π_2. For this construction it may seem reasonable to expect that soundness should fall as $1-(1-\delta_s)^2$, because if both $\mathsf{crs}_1, \mathsf{crs}_2$ are sampled using randomness from the hardcore set, then the soundness should hold. It may also seem reasonable to expect that zero-knowledge should be amplified as δ_z^2 as it appears that zero-knowledge should be retained as long as either π_1 or π_2 is computed from the hardcore set. We do not know how to formally convert this intuition into a proof. In fact, as far as we know, this intuition may be false, and we leave it as an interesting open problem to analyse this construction. We now summarize the difficulties in turning the intuition above into a proof:

- Arguing soundness is hard because the NIZK candidate is only required to have computational soundness. Therefore, with respect to crs_2 there may always exist a valid witness π_1 for an instance x, crs_1 even when $x \notin L$. As a result, we do not know how to analyze how soundness is affected by this construction.
- Arguing zero-knowledge is also hard for important technical reasons related to hard core sets, that we also have to keep in mind when we try to repair this state of affairs. When randomness is sampled from the hardcore measure to prove instance x, crs_1, it may already leak information about the witness for x, as the hardcore measure now can depend on w.

For the reasons above we consider a completely different approach. Crucial to our approach is the following primitive, which we call verifiable sharing scheme for NP statements (denoted by NPSS). We believe this notion may be of independent interest to other interesting applications.

Secret Sharing NP Instances: Informally speaking[4], an NPSS scheme consists of three algorithms Share, Verify, and Sim. Given any instance $x \in \mathsf{SAT}$ and its witness w, we have that $\mathsf{Share}(n, x, w)$ outputs n instances along with witnesses $\{x_i, w_i\}_{i\in[n]}$ such that the following guarantees are met. The scheme is parameterized by two thresholds t_1, t_2.

1. If $x \in \mathsf{SAT}$ with w being a valid witness, then the output of $\mathsf{Share}(n, x, w)$ will have the property that w_i is a valid witness of the statement $x_i \in \mathsf{SAT}$ for all $i \in [n]$.
2. **Robustness for threshold t_1.** There exists a verification algorithm Verify such that if $\mathsf{Verify}(n, x, x_1, ..., x_n) = 1$, then if there is a set $S \subset [n]$ of size greater than or equal to t_1 such that $x_i \in \mathsf{SAT}$ for $i \in S$, then we have that $x \in \mathsf{SAT}$. Furthermore there is an efficient algorithm that recovers the witness to x given witnesses for the statements x_i where $i \in S$.
3. **Simulatability for threshold t_2.** Consider any set $Z \subset [n]$ of size less than or equal to t_2. Then, informally, we want that the instances $x_1, .., x_n$ and witnesses $\{w_i\}_{i\in Z}$ should not "reveal any knowledge"

[4] Formal details can be found in Section 5.

about membership of x in SAT. That is, the output of $\mathsf{Sim}(n, x, Z)$ is computationally indistinguishable from the output of $\mathsf{Share}(n, x, w)$ restricted to all instances $x_1, \ldots, x_n$ and witnesses $\{w_i\}_{i \in Z}$.

Our actual notion of NPSS also includes a setup algorithm $\mathsf{Setup}(1^\lambda)$ that outputs public parameters pp, that is also input to constituent NPSS algorithms. We will describe how to construct such a sharing scheme for various choices of t_1, t_2 later. Assuming we have such a notion, we now describe how to achieve our goal. The following is our construction of $\Pi_\perp$ with repetition parameter n. Here is our construction. In the following set $t_1 = n$ and $t_2 = n - 1$ for the NPSS scheme.

- $\Pi_\perp.\mathsf{Setup}(1^\lambda)$:
  - Run $\Pi.\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{crs}_i$ for $i \in [n]$.
  - Run $\mathsf{NPSS}.\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{pp}$.
  - Output $\mathsf{crs} = (\mathsf{pp}, \mathsf{crs}_1, ..., \mathsf{crs}_n)$.
- $\Pi_\perp.\mathsf{Prove}(\mathsf{crs}, x, w)$:
  - Run $\mathsf{NPSS}.\mathsf{Share}(\mathsf{pp}, n, x, w) \rightarrow (x_1, ..., x_n, w_1, ..., w_n)$
  - Run $\Pi.\mathsf{Prove}(\mathsf{crs}_i, x_i, w_i) \rightarrow \pi_i$ for $i \in [n]$.
  - Output $\pi = (x_1, ..., x_n, \pi_1, ..., \pi_n)$.
- $\Pi_\perp.\mathsf{Verify}(\mathsf{crs}, x, \pi)$:
  - Parse $\pi = (x_1, ..., x_n, \pi_1, ..., \pi_n)$.
  - Run $\mathsf{NPSS}.\mathsf{Verify}(\mathsf{pp}, n, x, x_1, ..., x_n)$.
  - Run $\Pi.\mathsf{Verify}(x_i, w_i)$ for $i \in [n]$.
  - Output 1 if all these steps pass. Output 0 otherwise.

We now revisit both the soundness and zero-knowledge property to observe the change in the parameters.

$(1 - (1 - \delta_s))^n -$*Soundness:* The idea here is that since the size of the hardcore measure is $(1 - \delta_s)|\mathcal{R}|$, where $\mathcal{R}$ is the set from which the randomness for $\Pi.\mathsf{Setup}$ is chosen, with probability $(1 - \delta_s)^n$ all crs_i for $i \in [n]$ will behave nicely. In such a case, if crs_i is used to prove $x_i \in \mathsf{SAT}$, then any efficient adversary can produce a false proof only with some tiny probability ϵ_s. Thus, by the robustness property and the lemmas described above we can argue soundness. A PKE scheme plays an important role because the associated secret key is used by our reduction to verify in polynomial time if the adversary has indeed succeeded in breaking soundness. Note that this is a highly simplified description and the proof requires a very careful analysis of the the structure of the adversary. This proof can be found in Section 8. Here is the formal theorem:

Theorem 4. *Assuming* PKE *is perfectly correct and* Π *is* δ_s-*sound against adversaries of size* s*, then for every* $1 > \epsilon > 0$*,* $\Pi_\perp$ *is* $(1 - (1 - \delta_s)^n) + O(\epsilon)-$*sound against adversaries of size* $s' = O(s \cdot \epsilon \cdot \delta_s / (1 - \delta_s)) - \mathsf{poly}(\lambda)$ *for a fixed polynomial* poly.

$2 \cdot \delta_z^n -$*zero-knowledge:* Proving zero-knowledge for this construction turns out to be highly nontrivial. Let us understand why is this the case. Consider an honest sharing of instance x and witness y, denoted by $x_1, ..., x_n$ with corresponding witnesses $w_1, ..., w_n$. As noted above, Theorem 2 says that there exist two hardcore measures $\mathcal{S}_{0,i}$ and $\mathcal{S}_{1,i}$

of density $1-\delta_z$ such that the distribution of honestly generated pair (crs_i, π_i) for x_i generated using randomness from $\mathcal{S}_{0,i}$ is computationally close to the simulated distribution generated by choosing randomness from $\mathcal{S}_{1,i}$. Thus it seems that with probability at least $1-\delta_z^n$, we should have at least one index $i \in [n]$, where we can shift to simulating proofs for one index i^*. Then, it seems plausible that we can use the security of NPSS scheme to simulate sharing $x_1, ..., x_n, \{w_i\}_{i \neq i^*}$. But unfortunately, this intuition fails to materialize as these measures $\mathcal{S}_{0,i}, \mathcal{S}_{1,i}$ are inefficient and may depend on w_i itself. In fact, this has been a major hurdle in various amplification scenarios, and that is why amplifying security for complex cryptographic primitives is considered a hard problem.
In order to fix this issue, we rely on the techniques building the dense model theorem. We overcome this issue by using the following idea, which can be made formal via the work on simulating auxiliary input [27, 9]. Because the hardcore measure has reasonable probability mass $1-\delta_z$, it cannot *verifiably* contain useful information to the adversary. For example, even if the hardcore distribution revealed the first few bits of the w_i, the adversary could not *know* for sure that these bits were in fact the correct bits. Indeed, we use the works of [27, 9] to make this idea precise, and show that the hardcore measures can be simulated in a way that fools all efficient adversaries, with a simulation that runs in subexponential time. This allows us to argue witness indistinguishability. Finally, as witness indistinguishability is enough to get zero-knowledge the result holds. Similar techniques were also used in [1], to give an amplification theorem for any functional encryption scheme. Let us now go over the steps of the argument carefully. We will prove witness indistinguishability first. Consider an instance x and two witnesses (y_0, y_1). For all indices $i \in [n]$ let us output $\mathsf{crs} = (\mathsf{pp}, \mathsf{crs}_1, ..., \mathsf{crs}_n)$, instance $x_1, ..., x_n$ and proofs $\pi_1, ..., \pi_n$. We construct a series of hybrids from $\mathbf{Hybrid}_0$ to $\mathbf{Hybrid}_m$ where $\mathbf{Hybrid}_0$ is the hybrid where witness y_b for a random $b \in \{0,1\}$ is used to prove honestly and $\mathbf{Hybrid}_m$ is independent of the witness. We prove that $|\Pr[\mathcal{A}(\mathbf{Hybrid}_0) = 1] - \Pr[\mathcal{A}(\mathbf{Hybrid}_m) = 1]| \leq \delta_z^n + \mathsf{negl}$ for any efficient adversary $\mathcal{A}$. Thus, this gives us the required result. Before delving slightly in the details, we recall the following two theorems. First theorem describes how sampling an element from measures of high density is computationally indistinguishable to sampling an element uniformly from a large set constructed using the measure.

Theorem 5 (Imported Theorem [22]). *Let $\mathcal{M}$ be any measure on $\{0,1\}^n$ of density $\mu(\mathcal{M}) \geq 1-\rho(n)$ Let $\gamma(n) \in (0, 1/2)$ be any function. Then, for a random set $\mathcal{S}$ chosen according to the measure $\mathcal{M}$ the following two holds with probability at least $1 - 2(2^{-2^n \gamma^2 (1-\rho)^4/64})$:*

- $(1 - \frac{\gamma(1-\rho)}{4})(1-\rho)2^n \leq |\mathcal{S}| \leq (1 + \frac{\gamma(1-\rho)}{4})(1-\rho)2^n$
- *For such a random set $\mathcal{S}$, for any distinguisher $\mathcal{A}$ with size $|\mathcal{A}| \leq 2^n(\frac{\gamma^2(1-\rho)^4}{64n})$ satisfying*

$$|\Pr_{x \leftarrow \mathcal{S}}[\mathcal{A}(x) = 1] - \Pr_{x \leftarrow \mathcal{D}_{\mathcal{M}}}[\mathcal{A}(x) = 1]| \leq \gamma$$

The following theorem from [9] says that for every distribution X and every potentially inefficient function $g : X \rightarrow \{0,1\}^{\ell_X}$, there exists a relatively efficient function h such that $(X, g(X))$ is computationally close to

$(X, h(X))$. The complexity of h is roughly $O(s\epsilon^{-2}2^{\ell_X})$. Here s is the size of adversaries that h wants to fool and ϵ is the maximum distinguishing advantage against adversaries of size s.
We also import a theorem from [9] that will be used by our security proofs.

Theorem 6 (Imported Theorem [9]).
Let $n, \ell \in \mathbb{N}$, $\epsilon > 0$ and $\mathcal{C}_{leak}$ be a family of distinguisher circuits from $\{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}$ of size $s(n)$. Then, for every distribution (X, Z) over $\{0,1\}^n \times \{0,1\}^\ell$, there exists a simulator $h : \{0,1\}^n \to \{0,1\}^\ell$ such that:

- *h has size bounded by $s' = O(s2^\ell\epsilon^{-2})$.*
- *(X, Z) and $(X, h(X))$ are indistinguishable by $\mathcal{C}_{leak}$. That is for every $C \in \mathcal{C}_{leak}$,*

$$| \Pr_{(x,z)\leftarrow(X,Z)}[C(x, z) = 1] - \Pr_{x\leftarrow X,h}[C(x, h(x)) = 1]| \leq \epsilon$$

Now we define our hybrids:

1. We define the first hybrid as the hybrid where each index $i \in [n]$ uses hardcore measure $\mathcal{S}_{0,i}$ to generate π_i with probability $1 - \delta_z$, and its complement $1 - \mathcal{S}_{0,i}$ otherwise. This is done by maintaining a string $z \in \{0,1\}^n$ which sets $z_i = 1$ with probability $1 - \delta_z$ and $z_i = 0$ otherwise. This string describes how randomness for various indices are chosen. Note that this hybrid is identical to $\mathbf{Hybrid}_0$.
2. Next we define $\mathbf{Hybrid}_2$ where we abort if $z = 0^n$. This occurs with probability bounded by δ_z^n. Thus $|\Pr[\mathcal{A}(\mathbf{Hybrid}_1) = 1] - \Pr[\mathcal{A}(\mathbf{Hybrid}_2) = 1]| \leq \delta_z^n$
3. Next, for all indices where $z_i = 1$, generate π_i using $\Pi.\mathsf{Sim}$ algorithm where the randomness is sampled from $\mathcal{S}_{1,i}$ whose density is also equal to $1 - \delta_z$. This hybrid is computationally close for an efficient adversary due to theorem 2.
4. Now we consider the following inefficient machine $\mathsf{Machine}$ that takes as input $(z, x, x_1, ..., x_n, \{w_i\}_{i|z_i=0})$ and outputs $(R_1, .., R_n)$ where R_i is the randomness sampled to generate proof for the index i. This may involve the machine to potentially brute force break $x_1, .., x_n$ and sample from various measures involved. This hybrid is identical to the previous hybrid as its just a representation change. At this point, ideally we would like to use theorem 6 from [27, 9], recalled above. We would like to "fake" the output of $\mathsf{Machine}$ using an efficient simulator h constructed using theorem 6. However since the size of h grows exponentially with the length of the randomness used to prove, there is no hope to argue any security.
5. To fix this, we observe that the density of hardcore measure as well as its complement is quite large. In other words, suppose, $\delta_z, 1-\delta_z > 2^{-\lambda/10}$. Thus we can rely on theorem 5 and have $\mathsf{Machine}$ to sample a large enough sets Set_i for $i \in [n]$ from the measures and use that set to generate the proofs. This hybrid is indistinguishable because of theorem 5. By large enough, we mean that they will at least have about $2^{-\lambda/10} \cdot |R|$ elements.

6. Now for each index $i \in [n]$, sample uniformly a set SetR_i from the space of randomness of Π by choosing $q = \lambda 2^{\lambda/10}$ inputs. Thus the probability of $\mathsf{SetR}_i \cap \mathsf{Set}_i = \phi$ is bounded by $e^{-\lambda}$. Then, change $\mathsf{Machine}$ to take as input $z, \mathsf{SetR}_1, .., \mathsf{SetR}_n, x_1, ..., x_n$ and output indices $(j_1, ..., j_n)$. Each index j_i denotes the index of the randomness in SetR_i used for generating (crs_i, π_i) pair for system i. This is picked by sampling randomness uniformly from $\mathsf{Set}_i \cap \mathsf{SetR}_i$. These hybrids are statistically close with the the statistical distance being bounded by the probability that the intersection of SetR_i and Set_i is empty.
7. Now since $\mathsf{Machine}$ always outputs indices of length bounded by λ^2, we can use theorem 6 to simulate it. This ensure that size of h grows as $s' \cdot 2^{n\lambda^2} \cdot \epsilon'^{-2}$. Here ϵ' is the advantage with which we want to fool the adversary of size s'.
8. Finally we use complexity leveraging and a super-strong PKE to instantiate NPSS to argue that even for adversaries of the same size as that of h, cannot distinguish the case when $x_1, ..., x_n, \{w_i\}_{i|z_i=0}$ are generated using y_b, or they are simulated. This makes the hybrid independent of b.

This leaves us with the following question:

How to Construct NPSS*?* Our constructions of NPSS are inspired by the MPC-in-the-head paradigm [25]. The idea is to visualise n parties $P_1, .., P_n$ in an MPC protocol where each party P_i has an additive secret sharing y_i of the witness w. What they do is, they run MPC protocol to compute the relation function $R(x, \Sigma_i y_i)$. In an honest behavior this should output 1. Thus when the MPC protocol, such as [4], is run each party P_i receives an output out_i and it has its view view_i (which contains its randomness, input y_i and messages sent and received by it). Then there is also a transcript T which is the collection of messages sent and received by each party. We define x_i to be the circuit that has a PKE encryption of the commitment of T, inputs y_i and party's randomness hardwired and it takes as input a set of corresponding commitment openings and checks:

1. view_i is a valid view for this MPC protocol corresponding to the transcript T. That is each message in the view is computed correctly using incoming messages and a fixed valid input and randomness. This step only takes as input the openings corresponding to commitments of view_i.
2. The output in the view_i is 1.

This allows us to secret share instances. We can prove security we rely on properties of underlying MPC protocol. For example, [4] has two properties (other than correctness):

1. Upto $n/3$ semi-honest views are simulatable.
2. Even if at most $n/3$ parties behave arbitrarily, they can't force honest parties to receive an incorrect output. This property is called perfect robustness.

This allows us to give an instantiation for $t_1 = \lfloor n/3 \rfloor$ and $t_2 = \lfloor n/3 \rfloor$. We rely on the protocol of [14] in the OT-hybrid model [26] to get an instantiation for $t_1 = n$ and $t_2 = n = 1$. The details can be found in Section 6.3 and the full version.

Thus, this is the formal theorem:

Theorem 7. *Assume that there exists a subexponentially secure public key encryption and a NIZK candidate Π satisfying δ_z-zero-knowledge against adversaries of size Size_Π where $\delta_z, 1-\delta_z > 2^{-\lambda/5}$. If $\mathsf{Size}_\Pi > \mathsf{Size}_1 \epsilon^{-2} \mathsf{poly}(\lambda)$ for any $1 > \epsilon > 0$ and $0 < \mathsf{Size}_1 < 2^{\lambda/5}$ then the construction $\Pi_\perp$ satisfies $2\delta_z^n + O(n\epsilon + 2^{-\lambda^c})-$witness indistinguishability against adversaries of size Size_1. Here poly is some fixed polynomial. $c > 0$ is a fixed constant.*

2.3 The general case: $\boldsymbol{\delta_s + \delta_z < 1}$

This is perhaps best understood using an example. Consider $\delta_s = 0.3$ and $\delta_z = 0.60$. Consider the following steps:

- Run parallel repetition using the repetition parameter $n_1 = \log_2 \lambda$. Thus the new parameters are (upto negligible additive factors) $\delta'_s = 0.3^{n_1}$ and $\delta'_z = 1 - 0.4^{n_1}$. Observe that $\delta'_s = \lambda^{-\log_2 10/3}$ and $\delta'_z = 1 - \lambda^{-\log_2 10/4}$.
- On the resulting candidate, perform sequential repetition with parameter $n_2 = \lambda^{\log_2 10/3}$. Thus, we observe that the soundness parameter changes as $\delta''_s = 1-(1-\delta'_s)^{n_2}$. Note that this is roughly $1-e^{-1}$ (e is the base of natural logarithm). As for the zero-knowledge, $\delta''_z = 2 \cdot (1 - \lambda^{=\log_2 10/4})^{n_2}$. As $\log_2 10/3 > \log_2 10/4 > 0$, we have that $\delta''_z = \mathsf{negl}$ for some negligible. Thus, finally we made progress.
- Apply parallel repetition with parameter λ to get a fully secure NIZK!

The idea above can be used to handle any parameters satisfying $\delta_s+\delta_z = 1-\epsilon$ for any constant $\epsilon > 0$. Details can be found in Section 9.

Simultaneous Amplification: We observe that the transformation described above is highly inefficient as we have to compose one transformation on top of other. When $\delta_s, \delta_z \leq 1/3 - 1/\mathsf{poly}$ then one can provide a single transformation which yields a fully secure NIZK. The details can be found in the full version.

2.4 Reader's Guide.

In Section 3, we recall some preliminaries useful for the rest of the paper. In Section 4 we define the notion of a NIZK candidate. In Section 5 we define the notion of NPSS. In full version [15], we construct the notion of NPSS. In Section 7 we prove a lemma useful for arguing soundness amplification. In Section 8, we analyse our MPC based repetition transformation. We analyse the parallel repetition construction in the full version. In Section 9 we show how to convert a candidate satisfying $\delta_s+\delta_z = 1-\epsilon$ for any constant $\epsilon > 0$ to a fully secure candidate. Finally, in full version, we present our direct transformation that transforms any candidate with $\delta_s, \delta_z < 1/3 - 1/\mathsf{poly}(\lambda)$ to a fully secure one.

3 Preliminaries

We denote by λ the security parameter. We say that a function $\epsilon(\lambda)$ is negligible in λ if $\epsilon(\lambda) = o(1/\lambda^c)$ for every $c \in \mathbb{N}$, and we write $\mathsf{negl}(\lambda)$

to denote a negligible function in λ. For a distribution X, we denote by $x \leftarrow X$ the process of sampling a value of x from the distribution X. For a set S, we denote by $s \overset{\$}{\leftarrow} S$ the process of sampling uniformly from S. For two sequence of random variable $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$, we say that X and Y are computationally indistinguishable if for any probabilistic polynomial time distinguisher D,

$$\left| \Pr[D(1^\lambda, x \leftarrow X_\lambda) = 1] - \Pr[D(1^\lambda, y \leftarrow Y_\lambda) = 1] \right| \leq \mathsf{negl}(\lambda)$$

for any sufficiently large $\lambda \in \mathbb{N}$. We say that the distributions are subexponentially indistinguishable if this negl is $2^{-\lambda^\varepsilon}$ for some constant $\varepsilon > 0$. We now define the notion of statistical distance.

Definition 1 (Statistical Distance). *Let E be a finite set, Ω a probability space, and $X, Y : \Omega \to E$ random variables. We define the* statistical distance *between X and Y to be the function* Dist *defined by* $\mathsf{Dist}(X, Y) = \frac{1}{2} \Sigma_{e \in E} |\Pr_X(X = e) - \Pr_Y(Y = e)|$

3.1 Amplification Preliminaries

Now we recall some notions and theorems that will be useful for the rest of the paper.

Definition 2 (Distinguishing Gap). *For any adversary $\mathcal{A}$ and two distributions $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$, define $\mathcal{A}$'s distinguishing gap in distinguishing these distributions to be* $|\Pr_{x \leftarrow \mathcal{X}_\lambda}[\mathcal{A}(1^\lambda, x) = 1] - \Pr_{y \leftarrow \mathcal{Y}_\lambda}[\mathcal{A}(1^\lambda, y) = 1]|$

Now we recall the definition of a measure.

Definition 3. *A measure is a function $\mathcal{M} : \{0,1\}^k \to [0,1]$. The size of a measure is $|\mathcal{M}| = \Sigma_{x \in \{0,1\}^k} \mathcal{M}(x)$. The density of a measure, $\mu(\mathcal{M}) = |\mathcal{M}| 2^{-k}$*

Each measure $\mathcal{M}$ induces a probability distribution $\mathcal{D}_\mathcal{M}$.

Definition 4. *Let $\mathcal{M} : \{0,1\}^k \to [0,1]$ be a measure. The distribution defined by measure $\mathcal{M}$ (denoted by $\mathcal{D}_\mathcal{M}$) is a distribution over $\{0,1\}^k$, where for every $x \in \{0,1\}^k$, $\Pr_{X \leftarrow \mathcal{D}_\mathcal{M}}[X = x] = \mathcal{M}(x)/|\mathcal{M}|$.*

We will consider a scaled version $\mathcal{M}_c$ of a measure $\mathcal{M}$ for a constant $0 < c < 1$ defined as $\mathcal{M}_c = c\mathcal{M}$. Note that $\mathcal{M}_c$ induces the same distribution as $\mathcal{M}$.

3.2 Useful Lemmas

We first import the following theorem from [29].

Theorem 8 (Imported Theorem [29]). *Let $E : \{0,1\}^n \to \mathcal{X}$ and $F : \{0,1\}^m \to \mathcal{X}$ be two functions, and let $\epsilon, \gamma \in (0,1)$ and $s > 0$ be given. If for all distinguishers $\mathcal{A}$ with size s we have*

$$|\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(E(x)) = 1] - \Pr_{y \leftarrow \{0,1\}^m}[\mathcal{A}(F(y)) = 1]| \leq \epsilon$$

Then there exist two measures $\mathcal{M}_0$ (on $\{0,1\}^n$) and $\mathcal{M}_1$ (on $\{0,1\}^n$) that depend on γ, s such that:
- *$\mu(\mathcal{M}_b) \geq 1 - \epsilon$ for $b \in \{0,1\}$*
- *For all distinguishers $\mathcal{A}'$ of size $s' = \frac{s\gamma^2}{128(m+n+1)}$*

$$|\Pr_{x \leftarrow \mathcal{D}_{\mathcal{M}_0}}[\mathcal{A}(E(x)) = 1] - \Pr_{y \leftarrow \mathcal{D}_{\mathcal{M}_1}}[\mathcal{A}(F(y)) = 1]| \leq \gamma$$

Now we describe a lemma from [22], that shows that if we sample a set $\mathcal{S}$ from any measure $\mathcal{M}$ by choosing each element i in the support with probability $\mathcal{M}(i)$, then no circuit of (some) bounded size can distinguish a sample x chosen randomly from the set $\mathcal{S}$ from an element sampled from distribution given by $\mathcal{M}$. Formally,

Theorem 9 (Imported Theorem [22].). *Let $\mathcal{M}$ be any measure on $\{0,1\}^n$ of density $\mu(\mathcal{M}) \geq 1 - \rho(n)$ Let $\gamma(n) \in (0, 1/2)$ be any function. Then, for a random set $\mathcal{S}$ chosen according to the measure $\mathcal{M}$ the following two holds with probability at least $1 - 2(2^{-2^n \gamma^2 (1-\rho)^4/64})$:*
- *$(1 - \frac{\gamma(1-\rho)}{4})(1-\rho)2^n \leq |\mathcal{S}| \leq (1 + \frac{\gamma(1-\rho)}{4})(1-\rho)2^n$*
- *For such a random set $\mathcal{S}$, for any distinguisher $\mathcal{A}$ with size $|\mathcal{A}| \leq 2^n(\frac{\gamma^2(1-\rho)^4}{64n})$ satisfying*

$$|\Pr_{x \leftarrow \mathcal{S}}[\mathcal{A}(x) = 1] - \Pr_{x \leftarrow \mathcal{D}_{\mathcal{M}}}[\mathcal{A}(x) = 1]| \leq \gamma$$

We also import a theorem from [9] that will be used by our security proofs. This lemma would be useful to simulate the randomness used to encrypt in an inefficient hybrid.

Theorem 10 (Imported Theorem [9].).
Let $n, \ell \in \mathbb{N}$, $\epsilon > 0$ and $\mathcal{C}_{leak}$ be a family of distinguisher circuits from $\{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}$ of size $s(n)$. Then, for every distribution (X, Z) over $\{0,1\}^n \times \{0,1\}^\ell$, there exists a simulator $h : \{0,1\}^n \to \{0,1\}^\ell$ such that:
- *h has size bounded by $s' = O(s2^\ell \epsilon^{-2})$.*
- *(X, Z) and $(X, h(X))$ are indistinguishable by $\mathcal{C}_{leak}$. That is for every $C \in \mathcal{C}_{leak}$,*

$$|\Pr_{(x,z) \leftarrow (X,Z)}[C(x,z) = 1] - \Pr_{x \leftarrow X, h}[C(x, h(x)) = 1]| \leq \epsilon$$

4 Definitions

Let SAT denote the language of satisfiable circuits. Let R denote the corresponding relation for SAT. For any instance x in SAT such that w is a witness of x, we write $R(x, w) = 1$ and $x(w) = 1$ to mean the same thing. Any candidate for an NP-complete language can be used to build a candidate for SAT (via NP reductions) and that is why we focus on that.

4.1 Non-Interactive Zero-Knowledge Candidates

A NIZK candidate $\Pi = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim})$ is composed of the following p.p.t. algorithms:

- $\underline{\mathsf{Setup}(1^\lambda) \to \mathsf{crs}:}$ The setup algorithm is a randomized algorithm that takes as input the security parameter and outputs a common reference string crs.
- $\underline{\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi:}$ The proving algorithm is a randomized algorithm that takes as input a common reference string crs, an instance x in the language SAT and a witness w such that $R(x, w) = 1$. The algorithm outputs a proof string π.
- $\underline{\mathsf{Verify}(\mathsf{crs}, x, \pi) \to \{0,1\}:}$ The deterministic verification algorithm takes as input a common reference string crs, an instance x and a string π and it outputs from the set $\{0, 1\}$
- $\underline{\mathsf{Sim}(1^\lambda, x) \to (\widetilde{\mathsf{crs}}, \widetilde{\pi}):}$ The randomized Sim algorithm (short for simulator) takes as an input an instance x and outputs a common reference string $\widetilde{\mathsf{crs}}$ along with a simulated proof string $\widetilde{\pi}$.

Remark 1. Wherever unspecified, the strings such as crs, π e.t.c. lie in $\{0, 1\}^*$.

Completeness We say that a NIZK candidate Π is complete if the following property is satisfied. For any instance x in SAT and its witness w such that $R(x, w) = 1$ it holds that:

$$Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w), \mathsf{Verify}(\mathsf{crs}, x, \pi) = 1] \geq 1 - \mathsf{negl}(\lambda)$$

Here the probability is taken over coins of the algorithms of Π

$\boldsymbol{\delta_s}$−Soundness We define two notion of soundness:

Adaptive Soundness: For any non-uniform p.p.t adversary $\mathcal{A}$ consider the following experiment:

1. Run $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$.
2. Adversary outputs $(x, \pi) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs})$.
3. Output 1 if $x \notin \mathsf{SAT}$ and $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$.

We say that the candidate Π is (adaptive) δ_s−sound if the probability that the above experiment (over coins of all algorithms of the candidate and the adversary) outputs 1 is at most δ_s.

Non-Adaptive Soundness: For any non uniform p.p.t adversary $\mathcal{A}$ and any instance $x \notin \mathsf{SAT}$ with $|x| = \mathsf{poly}(\lambda)$, consider the following experiment:

1. Run $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$.
2. Adversary outputs $\pi \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs})$.
3. Output 1 if $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$.

We say that the candidate Π is (non-adaptive)-δ_s sound if the probability that the above experiment (over coins of all algorithms of the candidate and the adversary) outputs 1 is at most δ_s.

Remark 2. Wherever unspecified we will refer to the adaptive soundness of any candidate.

δ_z−Zero Knowledge We say that a NIZK candidate Π is δ_z−zero knowledge if the following property is satisfied. For any instance x and a witness w such that $R(x, w) = 1$, and all p.p.t adversaries $\mathcal{A}$

$$|Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), \mathcal{A}(\mathsf{crs}, x, \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)) = 1] -$$

$$Pr[(\mathsf{crs}, \pi) \leftarrow \mathsf{Sim}(1^\lambda, x), \mathcal{A}(\mathsf{crs}, x, \pi) = 1]| \leq \delta_z(\lambda)$$

Here the probability is taken over coins of the algorithms of Π and the adversary $\mathcal{A}$.

Remark 3. In general, a NIZK candidate is not required to satisfy soundness or zero knowledge. So, for example a candidate that outputs the witness in the clear is also a valid candidate. We will specify soundness and zero-knowledge properties when referring to them.

Remark 4. We say that a NIZK candidate is secure if it is $\mathsf{negl}(\lambda)$−sound and $\mathsf{negl}(\lambda)$−zero knowledge for some negligible function negl.

Remark 5. (Length of Instance.) We could also consider a definition where length of instance is given as input to the Setup algorithm so that the the argument system can only be used for statements of that fixed length. In particular, it can also be set as the security parameter. Our analysis can be easily extended for such a definition. We omit introducing this parameter for simplicity

NIWI *Candidate* A non-interactive witness indistinguishable argument (NIWI) candidate Π consists of three algorithms Setup, Prove and Verify with the same syntax as for a NIZK candidate. It has same completeness and δ_s−soundness property. Instead of δ_z−zero-knowledge property it has δ_w−witness indistinguishability requirement which is defined below.

δ_w−Witness Indistinguishability We say that a NIWI candidate Π is δ_w−witness indistinguishability if the following property is satisfied. For any instance x and any valid witness w_0, w_1 such that $R(x, w_b) = 1$ for $b \in \{0, 1\}$, and all (non-uniform) p.p.t adversaries $\mathcal{A}$

$$|Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), \mathcal{A}(\mathsf{crs}, x, \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w_0), w_0, w_1) = 1] -$$

$$Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), \mathcal{A}(\mathsf{crs}, x, \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w_1), w_0, w_1) = 1]| \leq \delta_w(\lambda)$$

Here the probability is taken over coins of the algorithms of Π and the adversary $\mathcal{A}$.

5 Verifiable Sharing for Statements

In this section, we define a new notion of sharing for SAT statements. We will denote it with NPSS. A verifiable sharing scheme for statements NPSS consists of the following p.p.t. algorithms:

- $\underline{\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{pp}}$: The setup algorithm takes as input the security parameter and outputs public parameters pp.

- $\underline{\mathsf{Share}(\mathsf{pp}, n, x, w) \to (x_1, ..., x_n, w_1, .., w_n)}$: The sharing algorithm takes as input an instance x and a witness w such that $R(x, w) = 1$ along with number of parties n and the public parameter pp. It outputs n instances $(x_1, .., x_n)$ along with valid corresponding witnesses $\{w_i\}_{i\in[n]}$.
- $\underline{\mathsf{Verify}(\mathsf{pp}, n, x, x_1, .., x_n) :\to \{1, 0\}}$: The Verify algorithm is a deterministic algorithm that takes as input public parameter pp, any instance x, a number n and a set of n instances x_i for $i \in [n]$. It outputs from $\{0, 1\}$.

We require that a NPSS satisfy the following properties.

Correctness: We say a verifiable sharing scheme for statements in SAT is correct if it happens for any satisfiable instance x in SAT having a witness w and $n \in \mathbb{N}$,

$$\Pr\begin{bmatrix} \mathsf{Setup}(1^\lambda) \to \mathsf{pp} \\ \mathsf{Share}(\mathsf{pp}, n, x, w) \to (x_1, ..., x_n, w_1, .., w_n) \\ R(x_i, w_i) = 1 \forall i \in [n] \\ \mathsf{Verify}(\mathsf{pp}, n, x, x_1, ..., x_n) = 1 \end{bmatrix} \geq 1 - 2^{-\lambda} \qquad (1)$$

Here the probability is only over the coins of Setup.

Next important property is of robustness for a threshold $t_{\mathsf{NPSS},r}$. This property says that if $(x_1, .., x_n)$ be shared instances associated with x. Then if there exists any set T of size $t_{\mathsf{NPSS},r}$ such that x_i is in SAT for all $i \in T$, this implies that x itself is satisfiable.

Robustness: This property says that for any instance x, number $n \in \mathbb{N}$, any sharing $(x_1, .., x_n)$ and any $T \subseteq [n]$ of size at least $t_{\mathsf{NPSS},r}$: If $\exists\{w_i\}_{i\in T}$ such that $\mathsf{Verify}(\mathsf{pp}, x, n, x_1, .., x_n) = 1$ and $R(x_i, w_i) = 1$ for all $i \in T$, then there exists w such that w is a witness of x. Formally, for any (even unbounded adversary $\mathcal{A}$), the following holds:

$$\Pr\begin{bmatrix} \mathsf{Setup}(1^\lambda) \to \mathsf{pp} \\ \mathcal{A}(\mathsf{pp}) \to (x, x_1, ..., x_n) \\ \mathsf{Verify}(\mathsf{pp}, n, x, x_1, ..., x_n) = 1 \\ \exists w_i, R(x_i, w_i) = 1 \forall i \in [T] \\ \nexists w, R(x, w) = 1 \end{bmatrix} \leq 2^{-\lambda} \qquad (2)$$

Here the probability is over the coins of the Setup.

Finally, last property is that of simulatability for a threshold $t_{\mathsf{NPSS},sim}$. In layman terms it says that for any instance x, a set of $t_{\mathsf{NPSS},sim}$ witnesses do not reveal anything about membership of x in the language SAT.

Simulatability: This property says that there exists a polynomial time simulator Sim that takes as input any $x \in \mathsf{SAT}$, n, and a set $T \subseteq [n]$ of size less than or equal to $t_{\mathsf{NPSS},sim}$. It outputs simulated instance-shares $\mathsf{Sim}(\mathsf{pp}, n, x, T) \to (x_1, .., x_n, \{w_i\}_{i\in T})$. Then consider the following distributions:

Distribution 1.

- Run $\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{pp}$
- Compute $\mathsf{Share}(\mathsf{pp}, n, x, w) \rightarrow (x_1, ..., x_n, w_1, ..., x_n)$
- Output $\{\mathsf{pp}, x, x_1, ..., x_n, \{w_i\}_{i\in[T]}\}$

Distribution 2.

- Run $\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{pp}$
- Compute $\mathsf{Sim}(\mathsf{pp}, n, x, T) \rightarrow (x_1, ..., x_n, \{w_i\}_{i\in T})$
- Output $\{\mathsf{pp}, x, x_1, ..., x_n, \{w_i\}_{i\in[T]}\}$

Then it holds that for any polynomial time adversary $\mathcal{A}$, the distinguishing gap between these two distributions is $\mathsf{negl}(\lambda)$ for some negligible function negl

Remark: In general, we ask Robustness and Simulatability property to hold with respect to thresholds $t_{\mathsf{NPSS},r}$ and $t_{\mathsf{NPSS},sim}$. Whenever required, we will instantiate these values once and omit explicitly mentioning them for simplicity.

6 Instantiating Verifiable Sharing of Statements

This section is organized as follows. In Section 6.1 we describe an MPC Framework that will be used to construct verifiable sharing scheme. In Section 6.2 we describe how to instantiate the framework. Then in Section 6.3 we describe the construction.

6.1 Σ-Pre-processing MPC

In this section we define an MPC framework associated with a protocol Σ, which we call $\Sigma-$pre-processing MPC. This framework will be used to instantiate verifiable sharing of statements. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a class of polynomial sized circuits. Here the security parameter λ is the length of inputs to this family. Our MPC framework consists of the following algorithms

- $\underline{\mathsf{Preproc}(y, n, 1^\ell) \rightarrow (y_1, r_1, ..., y_n, r_n)}$: This randomized algorithm takes as input the number of parties n, size of the function ℓ and the input y. It outputs pre-processed inputs and randomness $y_1, r_1, ..., y_n, r_n$. Here y_i, r_i is viewed as input and randomness of the party P_i participating in the protocol Σ.
- $\underline{\mathsf{Eval}(f, y_1, r_1, ..., y_n, r_n) \rightarrow T}$: The deterministic Eval algorithm takes as input the function $f \in \mathcal{F}_{|y|}$ and n input-randomness pairs (y_i, r_i) for $i \in [n]$. It outputs the entire emulated transcript of the protocol Σ (run by n parties $P_1, .., P_n$) to compute f using the inputs $(y_1, .., y_n)$ and the randomness $(r_1, .., r_n)$. Let us represent the transcript as $T = \{(i, j, k, T_{i,j,k}), \mathsf{Out}_i\}_{i\in[n],j\in[n],k\in\phi_{|f|}},$. Here $T_{i,j,k}$ represents the message sent by party P_i to P_j in round k. Here $\phi_{|f|}$ denotes the number of rounds in Σ and Out_i denotes the output of P_i.

Notation: We now give some notation. Let T denote the transcript of the protocol Σ run between n parties to compute any function f on the inputs $\{y_i\}_{i\in[n]}$ using randomness $\{r_i\}_{i\in[n]}$. We define by view_i the set containing the input y_i, randomness r_i and messages sent and received by the party i along with its output Out_i. More formally, we let $\mathsf{view}_i = \{y_i, r_i, \{T_{i,j,k}\}_{j \in [n], k \in \phi_{|f|}}, \mathsf{Out}_i, \{T_{j,i,k}\}_{j \in [n], k \in \phi_{|f|}}\}$. Further, we say that for any party i, view_i is consistent with the transcript T if the messages sent and received by party i are exactly equal to ones described in the transcript T and Out_i is also the output that occurs in the transcript.

Second, by $V_{\Sigma,f,n}()$ we denote a circuit that takes as input (i, view_i) for $i \in [n]$ and checks if the view_i is consistent with the protocol Σ computing f. If the check passes it outputs 1 and 0 otherwise. That is, it internally emulates the next message function and checks if all the outgoing messages of P_i are correctly computed using the input and previous messages. We say that view_i is consistent if $V_{\Sigma,f,n}(i, \mathsf{view}_i) = 1$. Now we require the following properties from this framework.

Perfect Correctness:

Definition 5. *(Perfect Correctness) For any input $y \in \{0,1\}^*$, $n \in \mathbb{N}$ and function $f \in \mathcal{F}_{|y|}$, consider the following experiment :*

- *Run* $\mathsf{Preproc}(y, n, 1^{|f|}) \to (y_1, r_1, ..., y_n, r_n)$.
- *Run* $\mathsf{Eval}(f, y_1, r_1, ..., y_n, r_n) \to T$.
- *Output 1 if* $\mathsf{Out}_i = f(y)$ *for all* $i \in [n]$ *and 0 otherwise.*

We say that a $\Sigma-$preprocessing MPC is perfectly correct if $Pr[\mathsf{Expt}(y, n, f) = 1] = 1$. Here the probability is taken over the coins of all the algorithms

Perfect Privacy:

Definition 6. *(Privacy for a threshold $t_{\Sigma,sim}$) We say that the a $\Sigma-$preprocessing MPC satisfies perfect privacy for a threshold $t_{\Sigma,sim}$ if there exists a simulator Sim such that for any $y \in \{0,1\}^*$, any $f \in \mathcal{F}_{|y|}$ and any set S of size less than or equal to $t_{\Sigma,sim}$ the following two experiments are computationally close.*

Expt_1

- *Run* $\mathsf{Preproc}(y, n, 1^{|y|}) \to (y_1, r_1, ..., y_n, r_n)$
- *Run* $\mathsf{Eval}(f, y_1, r_1, .., y_n, r_n) \to T$
- *Output* $\{\mathsf{view}_i\}_{i \in S}$

Expt_2

- *Output* $\mathsf{Sim}(1^{|y|}, f(y), n, S) \to \{\mathsf{view}_i\}_{i \in S}$

Robustness:

Definition 7. *(Robustness for a threshold $t_{\Sigma,r}$) We say that a $\Sigma-$preprocessing MPC is robust if the following happens: Let $f \in \mathcal{F}_\lambda$ for any $\lambda \in \mathbb{N}$ be a function such that $f(y) \neq 1$ for all $y \in \{0,1\}^\lambda$. Then, given any number of parties n, candidate transcript T and its consistent views $\{\mathsf{view}_i\}_{i \in S}$ corresponding to some set $S \subseteq [n]$ of size $t_{\Sigma,r}$, it holds that if $V_{\Sigma,f,n}(i, \mathsf{view}_i) = 1$ for all $i \in S$ then,*

$$\mathsf{Out}_i \neq 1$$

for some $i \in S$

This intuitively means that a collusion of at most $n - t_{\Sigma,r}$ parties can't force an incorrect output onto honest parties.

6.2 Instantiating MPC Framework for $t_{\Sigma,sim} = \lfloor n/3 \rfloor$ and $t_{\Sigma,r} = \lceil 2n/3 \rceil$

We cite [4] as the protocol. This protocol satisfies these three properties [2]:

1. Perfect correctness for 0 corruptions.
2. Perfect security for up to $n/3$ semi-honest corruptions.
3. Perfect robustness for up to $n/3$ corruptions.

The framework then works as follows. The $\mathsf{Preproc}$ algorithm takes as input the witness w and secret shares it using additive secret sharing scheme to get shares $y_1, .., y_n$. It also samples randomness for the parties $(r_1, .., r_n)$ to participate in a protocol computing $f(\Sigma_i y_i)$. The Eval algorithm emulates the protocol and outputs the transcript.
Thus using [4] we can achieve robustness and perfect privacy properties.

6.3 Construction of Verifiable Sharing Scheme for Statements

In this section we construct Verifiable Sharing Scheme for Statements from a $\Sigma-$pre-processing MPC Δ_Σ with thresholds $t_{\Sigma,r}, t_{\Sigma,sim}$ and a statistically binding non-interactive commitment scheme Com. We describe the construction below.

- $\underline{\mathsf{Setup}(1^\lambda) :}$ Run the setup of the commitment scheme $\mathsf{Com.Setup}(1^\lambda) \rightarrow \mathsf{pp}$.
- $\underline{\mathsf{Share}(\mathsf{pp}, n, x, w) :}$ The algorithm takes as input the number of parties n, instance x and witness w along with commitment parameters pp. It runs the algorithm described in Figure 1 to output $(x_1, ..., x_n, w_1, .., w_n)$.
- $\underline{\mathsf{Verify}(\mathsf{pp}, n, x, x_1, ..., x_n) :}$ The Verify algorithm takes as input the instance x and shares $x_1, .., x_n$ and does the following:
  - Let $f = R(x, \cdot)$ be the relation function hardwired with x. Check that there exists strings $Z_T = \{Z_i, Z_{\mathsf{Out},i}, Z_{i,j,k}, Z_{j,i,k}\}_{i\in[n],j\in[n],k\in[\phi_{|f|}]}$ in the circuit descriptions.
  - Check that these commitments to the views $Z_{\mathsf{view},i}$ for $i \in [n]$ are consistent to a single commitment to a transcript $Z_T = \{Z_i, Z_{\mathsf{Out},i}, Z_{i,j,k}, Z_{j,i,k}\}_{i\in[n],j\in[n],k\in[\phi_{|f|}]}$. Here each $Z_{\mathsf{view},i} = \{Z_i, Z_{\mathsf{Out},i}, Z_{i,j,k}, Z_{j,i,k}\}_{j\in[n],k\in[\phi_{|f|}]}$ for all $i \in [n]$.
  - If all the above checks pass output 1 otherwise output 0.

We prove the associated properties in the full version.

7 Technical Lemmas

Now we prove some technical lemmas useful for the rest of the paper.
We now present a hardcore set lemma that represents the soundness experiment.

Inputs: Commitment parameter pp, Number of parties n, instance $x \in \mathsf{SAT}$ and a valid witness w.
Notation: Let $f = R(x, \cdot)$ be the circuit denoting the corresponding NP relation R hardwired with the instance x.

1. Run the pre-process algorithm of the MPC framework on input w. That is $\mathit{\Delta}.\mathsf{Preproc}(w, n, 1^{|f|}) \to (y_1, r_1, ..., y_n, r_n)$
2. Run the evaluation algorithm of the framework to get a transcript. $\mathit{\Delta}_{\Sigma}.\mathsf{Eval}(f, y_1, r_1, ..., y_n, r_n) \to T$
3. Parse the transcript as $\{i, j, k, T_{i,j,k}, \mathsf{Out}_i\}_{i \in [n], j \in [n], k \in \phi_{|f|}}$. Here $T_{i,j,k}$ refers to the message sent by party P_i to party P_j in round k, and Out_i refers to the output of party P_i.
4. Let view_i denote the set $\{y_i, r_i, T_{i,j,k}, T_{j,i,k}, \mathsf{Out}_i\}_{j \in [n], k \in \phi_{|f|}}$.
5. Compute commitments $Z_{i,j,k} = \mathsf{Com}(\mathsf{pp}, T_{i,j,k}; s_{i,j,k})$ for $i, j \in [n]$ and $k \in \phi_{|f|}$.
6. Compute commitments $Z_{\mathsf{Out},i} = \mathsf{Com}(\mathsf{pp}, \mathsf{Out}_i, s_{i,\mathsf{Out}})$ and $Z_i = \mathsf{Com}(\mathsf{pp}, y_i, r_i; s_i)$ for $i \in [n]$.
7. Define by $Z_{\mathsf{view},i} = \{Z_i, Z_{\mathsf{Out},i}, Z_{i,j,k}, Z_{j,i,k}\}_{j \in [n], k \in \phi_{|f|}}$. Output the circuit $x_i[Z_{\mathsf{view},i}, f]$ (defined in Figure 2) as the i^{th} share.
8. For every $i \in [n]$ witness are given out as the set of associated commitment openings and the views,. That is output $w_i = \{\{s_i, s_{\mathsf{Out},i}, s_{i,j,k}, s_{j,i,k}\}_{j \in [n], k \in \phi_{|f|}}, \mathsf{view}_i\}$.

Fig. 1: Description of Share algorithm

Inputs: Commitment openings $\{s_i, s_{\mathsf{Out},i}, s_{i,j,k}, s_{j,i,k}\}_{j \in [n], k \in \phi_{|f|}}$, view of the party P_i $\mathsf{view}_i = \{y_i, r_i, T_{i,j,k}, T_{j,i,k}, \mathsf{Out}_i\}$
Hardwired: pp, $Z_{\mathsf{view},i} = \{Z_i, Z_{\mathsf{Out},i}, Z_{i,j,k}, Z_{j,i,k}\}_{j \in [n], k \in [\phi_{|f|}]}$ and function f.

1. Check that the commitment openings are valid.
   - $Z_i = \mathsf{Com}(\mathsf{pp}, y_i, r_i; s_i)$.
   - $Z_{\mathsf{Out},i} = \mathsf{Com}(\mathsf{pp}, \mathsf{Out}_i, s_{\mathsf{Out},i})$.
   - $Z_{i,j,k} = \mathsf{Com}(\mathsf{pp}, T_{i,j,k}; s_{i,j,k})$ for all $j \in [n]$ and $k \in [\phi_{|f|}]$
   - $Z_{j,i,k} = \mathsf{Com}(\mathsf{pp}, T_{j,i,k}; s_{j,i,k})$ for all $j \in [n]$ and $k \in [\phi_{|f|}]$
2. Check that $V_{\Sigma,f,n}(i, \mathsf{view}_i) = 1$.
3. If all the above checks output 1, output 1 otherwise output 0.

Fig. 2: Description of circuit x_i

Lemma 2. *Let $F : \{0,1\}^{\lambda} \to \{0,1\}^{l}$ be a function where $l = \mathsf{poly}(\lambda)$ and $E : \{0,1\}^{\lambda+l+r(\lambda)} \to \{0,1\}$ be a circuit of size e. Let $\delta \geq \epsilon \in (0,1)$ and $s, s' > 0$ be functions of λ. If for all circuits $C : \{0,1\}^{l(\lambda)} \to \{0,1\}^{r(\lambda)}$ of* *size s we have*

$$\Pr_{u \xleftarrow{\$} \{0,1\}^{\lambda}} [E(u, F(u), C(F(u))) = 1] \leq \delta$$

Then there exists a set S of size $|S| = (1-\delta)2^{\lambda}$ and a polynomial $s_{overhead}(\lambda)$ (independent of s, s' and e) such that: For all circuits $C' : \{0,1\}^{l(\lambda)} \to \{0,1\}^{r(\lambda)}$ of size less than $s' = \frac{s\epsilon(1-\delta)}{\delta} - e - s_{overhead}$

$$\Pr_{u \xleftarrow{\$} S} [E(u, F(u), C'(F(u))) = 1] \leq \epsilon$$

Proof. The proof strategy can be described as follows: we assume that there does not exist a hardcore set of size $(1-\delta)2^{\lambda}$ for circuits of size less than s'. We use this fact to construct a circuit of size s which contradicts the assumption made in the theorem statement.

Formally, let us assume that the following happens: for every set $S \subset \{0,1\}^{\lambda}$ such that $|S| = (1-\delta)2^{\lambda}$ there exists a circuit C_S of size s',

$$\Pr_{u \xleftarrow{\$} S} [E(u, F(u), C_S(F(u))) = 1] \geq \epsilon$$

We now define two collections:

1. Collection of inputs $\mathbb{X} \subseteq \{0,1\}^{\lambda}$. This collection is initialised to be empty and stores the list of "solved inputs". Here, we say that $x \in \{0,1\}^{\lambda}$ is solved by C, if $E(F(x), C(F(x))) = 1$.
2. Collection of circuits $\mathbb{C}$ which stores circuits of size s'. This collection is also initialised to be empty and stores circuits that "solve" at least $\delta(1-\epsilon)$ fraction of input points.

This collection $\mathbb{C}$ will later be used to build a circuit $C[\mathbb{C}]$ of size s such that it will solve at least $\mathbb{X}$. Contradiction will come from the fact $|\mathbb{X}|$ is greater than $2^{\lambda}\delta$.
Both $\mathbb{X}$ and $\mathbb{C}$ are build iteratively as follows. Pick any set S_1 of size $(1-\delta)\cdot 2^{\lambda}$. There exists a circuit C_1 of size s' such that $\Pr_{u \xleftarrow{\$} S_1}[E(u, F(u), C_1(F(u))) = 1] \geq \epsilon$ as per the hypothesis. Let X_1 be the maximal subset of S_1 of size at least $(1-\delta)\epsilon 2^{\lambda}$ such that $\Pr_{u \xleftarrow{\$} X_1}[E(u, F(u), C_1(F(u))) = 1] = 1$. The size $|X_1| \geq \epsilon(1-\delta)2^{\lambda}$.

We now update $\mathbb{X} = \mathbb{X} \cup X_1$ and $\mathbb{C} = \mathbb{C} \cup C_1$.

This process is repeated t times (defined later) as follows.

1. Select a set S_i of size at least $(1-\delta)2^{\lambda} \subseteq \{0,1\}^{\lambda} \setminus \mathbb{X}$.
2. Let C_i be a circuit of size s' such that $\Pr_{u \xleftarrow{\$} S_i}[E(u, F(u), C_i(F(u))) = 1] \geq \epsilon$.
3. Let X_i be a maximal set of cardinality at least $(1-\delta)\epsilon 2^{\lambda}$, $\Pr_{u \xleftarrow{\$} X_i}[E(u, F(u), C_i(F(u))) = 1] = 1$
4. Update $\mathbb{C} = \mathbb{C} \cup C_i$ and $\mathbb{X} = \mathbb{X} \cup X_i$

Define a circuit $C[\mathbb{C}]$ for $\mathbb{C} = (C_1, .., C_t)$. On any input $F(x) \in \{0,1\}^{l(\lambda)}$, it checks if there exist i such that $E(x, F(x), C_i(F(x))) = 1$. If this is the case it outputs $C_i(F(x))$, otherwise it outputs $C_1(F(x))$. We now claim this process cannot continue indefinitely. Observe the following:

1. $|\mathbb{X}| > t \cdot \epsilon \cdot (1-\delta) \cdot 2^{\lambda}$
2. $|C[\mathbb{C}]| \leq ts' + t \cdot e + t \cdot s_{overhead}(\lambda)$, for some fixed polynomial $s_{overhead}$ independent of s, s' and e.

Thus we can achieve a contradiction if the following holds simultaneously.

1. $|\mathbb{X}| \geq t \cdot \epsilon \cdot (1-\delta) \cdot 2^{\lambda} \geq \delta 2^{\lambda}$.
2. $|C[\mathbb{C}]| \leq t \cdot s' + t \cdot e + t \cdot s_{ovehead}(\lambda) \leq s$.

This is because these conditions ensure that $C[\mathbb{C}]$ is a required circuit that violates the hypothesis. For these conditions to happen we can set any s' and t satisfying, $t \geq \frac{1-\delta}{\delta \cdot \epsilon}$ and $s' \leq \frac{s-p(\lambda)}{t} - e - s_{overhead}$.

8 Sequential Repetition

In this section, we construct $\Pi_{\perp}$ which is an analogue of parallel repetition. It starts from δ_z-zero knowledge candidate, δ_s sound NIZK candidate and constructs (roughly) δ_z^n-zero knowledge and $1-(1-\delta_s)^n$ sound NIZK candidate $\Pi_{\perp}$. Note that these are the parameters for parallel repetition where soundness and zero knowledge errors (parameters) are interchanged and that is why we call it sequential repetition.

Ingredient: We require a verifiable sharing scheme NPSS with the following properties:

- Perfect Correctness.
- Robustness holds if $T = [n]$.
- Computational Simulatability as long as at most $n-1$ witnesses are revealed.

Such a scheme can be constructed by instantiating $\Sigma-$preprocessing MPC framework with perfectly correct, information theoretically secure GMW protocol [14] in the OT hybrid model [28]. This protocol satisfies information theoretic security for $n-1$ corruptions. More details can be found in the full version. We also assume that the commitment scheme used in constructing NPSS uses perfectly correct a public key encryption scheme PKE. We now describe our construction.

- $\Pi_{\perp}.\mathsf{Setup}(1^{\lambda})$:
  - Run $\Pi.\mathsf{Setup}(1^{\lambda}) \rightarrow \mathsf{crs}_i$ for $i \in [n]$.
  - Run $\mathsf{NPSS}.\mathsf{Setup}(1^{\lambda}) \rightarrow \mathsf{pp}$.
  - Output $\mathsf{crs} = (\mathsf{pp}, \mathsf{crs}_1, ..., \mathsf{crs}_n)$.
- $\Pi_{\perp}.\mathsf{Prove}(\mathsf{crs}, x, w)$:
  - Run $\mathsf{NPSS}.\mathsf{Share}(\mathsf{pp}, n, x, w) \rightarrow (x_1, ..., x_n, w_1, ..., w_n)$
  - Run $\Pi.\mathsf{Prove}(\mathsf{crs}_i, x_i, w_i) \rightarrow \pi_i$ for $i \in [n]$.
  - Output $\pi = (x_1, ..., x_n, \pi_1, ..., \pi_n)$.
- $\Pi_{\perp}.\mathsf{Verify}(\mathsf{crs}, x, \pi)$:
  - Parse $\pi = (x_1, ..., x_n, \pi_1, ..., \pi_n)$.
  - Run $\mathsf{NPSS}.\mathsf{Verify}(\mathsf{pp}, n, x, x_1, ..., x_n)$.
  - Run $\Pi.\mathsf{Verify}(x_i, w_i)$ for $i \in [n]$.
  - Output 1 if all these steps pass. Output 0 otherwise.

Completeness. Completeness follows immediately from the completeness of Π.

$(1-(1-\delta_s)^n)-$ soundness:

Theorem 11. *Assuming* PKE *is perfectly correct and* Π *is* δ_s-*sound against adversaries of size* s*, then for every* $1 > \epsilon > 0$*,* $\Pi_\perp$ *is* $(1-(1-\delta_s)^n)+O(\epsilon)-$*sound against adversaries of size* $s' = O(s\cdot\epsilon\cdot\delta_s/(1-\delta_s)) - \mathsf{poly}(\lambda)$ *for a fixed polynomial* poly*.*

Proof. Let $C = (C_1, ..., C_n)$ be the circuit attacking the soundness experiment.
First define a function:
$F(r)$

- Compute $\Pi.\mathsf{Setup}(1^\lambda; r) \rightarrow \mathsf{crs}$.
- Output crs.

Let $\mathsf{pp} \leftarrow \mathsf{NPSS.Setup}(1^\lambda)$. We fix pp, and we claim that soundness holds with overwhelming probability over the coins for generating pp. Now, $C(F(u_1), ..., F(u_n)) = x, x_1, ..., x_n, \pi_1, ..., \pi_n$. Define the output of C_i as x, x_i, π_i
Denote $c = \delta_s$. Let us recall the soundness experiment in detail.

- The challenger samples $\Pi.\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{crs}_i$ for $i \in [t]$. Then it hands over $\mathsf{crs}_\perp = (\mathsf{pp}, \mathsf{crs}_1, .., \mathsf{crs}_t)$
- The adversary on input $\mathsf{crs}_\perp$ comes up with a proof $\pi = (x_1, .., x_n, \pi_1, .., \pi_n)$ and an instance x such that $\mathsf{NPSS.Verify}(\mathsf{pp}, x, x_1, .., x_n) = 1$, $\Pi.\mathsf{Verify}(\mathsf{crs}_i, x_i, \pi_i) = 1$ for $i \in [n]$. The adversary wins if x is unsatisfiable.

We begin by setting some notation for the rest of the proof.

- Define $F(\cdot) = \Pi.\mathsf{Setup}(1^\lambda, \cdot) : \{0,1\}^{\ell_{rand}(\lambda)} \rightarrow \{0,1\}^{\ell_{crs}(\lambda)}$. Note that both ℓ_{crs}, ℓ_{rand} are some polynomials.
- Let $C = (C_1, .., C_n)$ be the polynomial sized-circuit attacking the soundness experiment. Each $C_i : \{0,1\}^{n\ell_{crs}} \rightarrow \{0,1\}^{\ell_\pi + 2\cdot\ell_x}$. Each C_i is thought to output x, x_i, π_i. They have pp hardwired.
- Let E denote the circuit that on input $(\mathsf{crs}_i, x_i, \pi_i) \in \{0,1\}^{\ell_{crs}+\ell_{pi}}$ does the following. It checks that $x_i = x_i[Z_{\mathsf{view},i}, R(x,\cdot)]$ (as in the construction of NPSS) and $\Pi.\mathsf{Verify}(\mathsf{crs}_i, x_i, \pi_i) = 1$. Then it opens the commitment $Z_{\mathsf{view},i}$ (using the secret-key corresponding to pp) and checks if the circuit $x_i \notin \mathsf{SAT}$. It outputs 1 if all these checks pass. Since the commitment can be opened in $\mathsf{poly}(\lambda)$ time using the decryption algorithm, size of E is $\mathsf{poly}(n, \lambda)$.

Since Π is $c-$sound against adversaries of size s, for all circuits D of size s,

$$\Pr_{u \xleftarrow{\$} \{0,1\}^{\ell_{rand}}} [E(u, F(u), D(F(u)) = 1] \leq c$$

Thus there exists a hardcore set by lemma 2 H of size $(1-c)2^{r(\lambda)}$ such that for any polynomial-sized circuit D' with size $s' \leq O(s\epsilon_{s'}(1-c)/c - s_{overhead} - \mathsf{poly}(\lambda))$,

$$\Pr_{u \xleftarrow{\$} H} [E(u, F(u), D'(F(u)) = 1] \leq \epsilon_{s'} \tag{3}$$

for any $0 < \epsilon_{s'} < 1$.

Define V to be the set $\{0,1\}^{r(\lambda)} \times ....\{0,1\}^{r(\lambda)}$ (i.e. the set of randomness used to sample all crs_i for $i \in [n]$). For every set $S \subseteq [n]$, define $V_S = A_1 \times A_2 ... \times A_t$, where $A_i = H$ if $i \in S$ and $A_i = \{0,1\}^r \setminus H$ otherwise. Note that V is a disjoint union of $\{V_S\}_{S \subseteq [n]}$.

For any set W, we define by Break_W the following event that is satisfied if the following conditions are satisfied.

1. $(u_1, .., u_n) \xleftarrow{\$} W$
2. $C_i(F(u_1), ..., F(u_n)) = (x, x_i, \pi_i)$ for all $i \in [n]$.
3. $\Pi.\mathsf{Verify}(F(u_i), x_i, \pi_i) = 1$ for all $i \in [n]$.
4. $\mathsf{NPSS.Verify}(\mathsf{pp}, t, x, x_1, .., x_n) = 1$
5. $x \notin \mathsf{SAT}$

Let,

$$\Pr[\mathsf{Break}_V] = q$$

Then, note that,

$$\Pr[\mathsf{Break}_V] = \Sigma_{S \subseteq [n]} \Pr[\mathsf{Break}_{V_S}] |V_S| / |V|$$

We make the following two claims now.

Claim. $\Sigma_{S \subseteq [n], |S| < n} |V_S| / |V| \leq (1 - (1-c)^n)$.

Proof. Consider n independent random variables y_i for $i \in [n]$ where $y_i = 0$ with probability c and 1 with probability $1 - c$. The probability that $y \neq 1^n$ is $= 1 - \Pr[y = 1^n]$. Since each bit is independently chosen, the claim follows as $\Pr[y = 1^n] = (1-c)^n$

Thus, $S^* = [n]$. $\Pr[\mathsf{Break}_V] \leq (1-c)^n \cdot \Pr[\mathsf{Break}_{V_{S^*}}] + (1 - (1-c)^n)$ Now we claim that $\Pr[\mathsf{Break}_{V_{S^*}}] \leq n\epsilon_{s'}$
Observe that $\Pr[\mathsf{Break}_V] = q \leq (1-c)^n \cdot \Pr[\mathsf{Break}_{V_{S^*}}] + (1 - (1-c)^n)$
Thus $\Pr[\mathsf{Break}_{V_{S^*}}] \geq q - (1 - (1-c)^n)$ We now define another event Sound_i.

1. $(u_1, .., u_n) \xleftarrow{\$} V_{S^*}$
2. $C_i(F(u_1), ..., F(u_n)) = (x, x_i, \pi_i)$ for all $i \in [n]$.
3. $\Pi.\mathsf{Verify}(F(u_i), x_i, \pi_i) = 1$ for all $i \in [n]$.
4. $\mathsf{NPSS.Verify}(\mathsf{pp}, n, x, x_1, .., x_n) = 1$
5. $x \notin \mathsf{SAT}$
6. $x_i \notin \mathsf{SAT}$

Note that $\Pr[\cup_{i \in S^*} \mathsf{Sound}_i] \geq \Pr[\mathsf{Break}_{V_{S^*}}]$ due to robustness of NPSS scheme. Thus by the union bound,

$$\Sigma_{i \in S^*} \Pr[\mathsf{Sound}_i] \geq q - (1 - (1-c)^n)$$

as $|S^*| = n$, there exist i^* such that,

$$\Pr[\mathsf{Sound}_{i^*}] \geq \frac{q - (1 - (1-c)^n)}{n}$$

Finally, define the event Final_{i^*}

1. $(u_1, .., u_n) \xleftarrow{\$} V_{S^*}$
2. $C_i^*(F(u_1), ..., F(u_n)) = (x, x^*, \pi^*)$.

3. $\mathit{\Pi}.\mathsf{Verify}(F(u_{i^*}), x^*, \pi^*) = 1$.
4. Instance x^* is of the form $x^* = x^*[Z_{\mathsf{view},i^*}, f]$.
5. $x^* \notin \mathsf{SAT}$.

As Final_i is true whenever Sound_i is, $\Pr[\mathsf{Final}_i] \geq (q - (1 - (1 - c)^n))/n$. This translates to the following

$$\Pr_{u_1,..,u_n \xleftarrow{\$} V_{S^*}} [E(u_{i^*}, F(u_{i^*}), C_{i^*}(F(u_1), ..., F(u_n))) = 1] \geq \frac{q - (1 - (1 - c)^n)}{n}$$

This implies that there exists $\{u_i\}_{i \neq i^*}$ such that:

$$\Pr_{u_{i^*} \xleftarrow{\$} V_{S^*}} [E(u_i^*, F(u_{i^*}), C_{i^*}(F(u_1), ..., F(u_n))) = 1] \geq \frac{q - (1 - (1 - c)^n)}{n}$$

The circuit $C_{i^*}(u_1, ..., u_{i^*-1}, \cdot, u_{i^*+1}, .., u_n)$ violates equation 3 if $q > n\epsilon_{s'} + 1 - (1 - c)^n$.

$2 \cdot \delta_s^n$*−zero-knowledge.*

Theorem 12. *Assume that there exists a subexponentially secure public key encryption and a NIZK candidate $\mathit{\Pi}$ satisfying δ_z−zero-knowledge against adversaries of size $\mathsf{Size}_{\mathit{\Pi}}$ where $\delta_z, 1 - \delta_z > 2^{-\lambda/5}$. If $\mathsf{Size}_{\mathit{\Pi}} > \mathsf{Size}_1 \epsilon^{-2} \mathsf{poly}(\lambda)$ for any $1 > \epsilon > 0$ and $0 < \mathsf{Size}_1 < 2^{\lambda/5}$ then the construction $\mathit{\Pi}_\perp$ satisfies $2\delta_z^n + O(n\epsilon + 2^{-\lambda^c})$−witness indistinguishability against adversaries of size Size_1. Here* poly *is some fixed polynomial. $c > 0$ is a fixed constant.*

We present the proof in the full version.

9 Amplifying Security when $\boldsymbol{\delta_s + \delta_z < 1}$

Now we show the following theorem:

Theorem 13. *Assume a subexponentially secure* PKE *scheme, and a* NIZK *candidate $\mathit{\Pi}$ with δ_s−soundness and δ_z−zero-knowledge where δ_z, δ_s are any constants in $(0, 1)$ with $\delta_s + \delta_z < 1$ for all polynomial time adversaries, then there exists a fully secure* NIZK *candidate against all polynomial time adversaries.*

We prove this is as follows:

1. First we use parallel repetition with repetition parameter $n_1 = \log \lambda$. Note that in that case, we get $\delta_{s,1} = \delta_s^{n_1} + O(n_1\epsilon_1)$ soundness and $\delta_{z,1} = 1-(1-\delta_z)_1^n+O(n_1\epsilon_1)$ from the theorems on parallel repetition. This holds for all adversaries of size $\mathsf{Size}_1 = \mathsf{Size} \cdot \epsilon^2/\mathsf{poly}(\lambda)$ where Size is the size of the adversaries for which $\mathit{\Pi}$ is secure and ϵ is chosen and poly is fixed.
2. Then we apply sequential repetition on the new parameters. Let $a = \log_2(1/\delta_s)$ and $b = \log_2(1/(1 - \delta_z))$. Note that as $\delta_s + \delta_z < 1$, $b < a$. Then $\delta_{s,1} = 1/\lambda^a + O(\epsilon \log \lambda)$ and $\delta_{z,1} = 1 - 1/\lambda^b + O(\epsilon \log \lambda)$. We now apply sequential repetition with parameter $n_2 = \lambda^a$. Once we do this, following happens.

- $\delta_{s,2}$, which is the soundness of the resulting candidate, becomes $\delta_{s,2} = 1 - (1 - \delta_{s,1})^{n_2} + O(n_2\epsilon_2)$. It holds against all adversaries of size $\mathsf{Size}_2 = \mathsf{Size}_1 \cdot \epsilon_2^2/\mathsf{poly}(\lambda)$, where ϵ_2 is chosen. Thus this is $1 - e^{-1} + O(\mathsf{poly}(\lambda)\epsilon + \epsilon_2)$ if ϵ, ϵ_2 are sufficiently small. Here poly is some fixed polynomial.
- On the other hand zero-knowledge becomes $\delta_{z,2} = 2 \cdot \delta_{z,1}^{n_2} + O(n_2\epsilon_2)$. This is equal to $\delta_{z,1}^{n_2} = (1 - 1/\lambda^b + \log\lambda\epsilon)^{\lambda_b}$. This is equal to $e^{-\lambda^{a-b}} + \mathsf{poly}(\lambda)\epsilon$ if ϵ is sufficiently small. Thus this results in $\delta_{z,2} = 2 \cdot e^{-\lambda^{a-b}} + O(\mathsf{poly}(\lambda)\epsilon + \epsilon_3)$. Here poly is some fixed polynomial.

Finally, we apply parallel repetition once again with parameter $n_3 = \lambda$ to obtain the result.

- $\delta_{s,3}$, which is the soundness of the resulting candidate, becomes $\delta_{s,3} = \delta_{s,2}^{n_3} + O(n_3\epsilon_3)$. It holds against all adversaries of size $\mathsf{Size}_3 = \mathsf{Size}_2 \cdot \epsilon_3^2/\mathsf{poly}(\lambda)$, where ϵ_3 is chosen. This is $2^{-c\lambda} + O(\mathsf{poly}(\lambda)(\epsilon + \epsilon_2 + \epsilon_3))$ if ϵ_2, ϵ is chosen sufficiently small. Here poly is some fixed polynomial and $c > 0$ is some constant.
- On the other hand zero-knowledge becomes $\delta_{z,3} = 1 - (1 - \delta_{z,2})^{\lambda} + O(\lambda\epsilon_3)$. This is bounded by $\lambda \cdot \delta_{z,2} + O(\lambda\epsilon_3) = O(2^{-\lambda_1^c} + \mathsf{poly}(\lambda)\epsilon + \epsilon_2 + \epsilon_3)$ by union bound. Here poly is some fixed polynomial and $c_1 > 0$ is some constant.

This proves the result.

10 Acknowledgements

Aayush Jain would like to thank Ashutosh Kumar and Alain Passelègue for very insightful discussions about simultaneous amplification and in particular how independent zero-knowledge and soundness amplification theorems imply general simultaneous amplification.

Vipul Goyal supported in part by a gift from Ripple, a gift from DoS Networks, a grant from Northrop Grumman, a JP Morgan Faculty Fellowship, and, a Cylab seed funding award.

Aayush Jain and Amit Sahai were supported in part from a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, and NSF grant 1619348, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. Aayush Jain is also supported by a Google PhD fellowship award in Privacy and Security. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C- 0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, the U.S. Government or Google.

References

1. Ananth, P., Jain, A., Sahai, A.: Indistinguishability obfuscation without multilinear maps: io from lwe, bilinear maps, and weak pseudorandomness. IACR Cryptology ePrint Archive 2018, 615 (2018)

2. Asharov, G., Lindell, Y.: A full proof of the BGW protocol for perfectly secure multiparty computation. J. Cryptology 30(1), 58–151 (2017)
3. Bellare, M., Impagliazzo, R., Naor, M.: Does parallel repetition lower the error in computationally sound protocols? In: FOCS. pp. 374–383 (1997)
4. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: STOC. pp. 1–10 (1988)
5. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: ITCS. pp. 326–349 (2012)
6. Bitansky, N., Paneth, O.: Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In: TCC. pp. 401–427 (2015)
7. Canetti, R., Halevi, S., Steiner, M.: Hardness amplification of weakly verifiable puzzles. In: TCC. pp. 17–33 (2005)
8. Canetti, R., Lombardi, A., Wichs, D.: Non-interactive zero knowledge and correlation intractability from circular-secure FHE. IACR Cryptology ePrint Archive 2018, 1248 (2018)
9. Chen, Y., Chung, K., Liao, J.: On the complexity of simulating auxiliary input. IACR Cryptology ePrint Archive 2018, 171 (2018)
10. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions (extended abstract). In: FOCS. pp. 42–52 (1988)
11. Damgård, I., Kilian, J., Salvail, L.: On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In: EUROCRYPT. pp. 56–73 (1999)
12. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. In: STOC. pp. 409–418 (1998)
13. Goldreich, O.: Basing non-interactive zero-knowledge on (enhanced) trapdoor permutations: The state of the art. In: Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation, pp. 406–421 (2011)
14. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: STOC. pp. 218–229 (1987)
15. Goyal, V., Jain, A., Sahai, A.: Simultaneous amplification: The case of non-interactive zero-knowledge. IACR Cryptology ePrint Archive 2019
16. Goyal, V., Khurana, D., Mironov, I., Pandey, O., Sahai, A.: Do distributed differentially-private protocols require oblivious transfer? In: ICALP. pp. 29:1–29:15 (2016)
17. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: CRYPTO. pp. 97–111 (2006)
18. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: EUROCRYPT. pp. 339–358 (2006)
19. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: EUROCRYPT. pp. 415–432 (2008)
20. Harnik, D., Ishai, Y., Kushilevitz, E., Nielsen, J.B.: Ot-combiners via secure computation. In: TCC. pp. 393–411 (2008)

21. Håstad, J., Pass, R., Wikström, D., Pietrzak, K.: An efficient parallel repetition theorem. In: TCC. pp. 1–18 (2010)
22. Holenstein, T.: Strengthening key agreement using hard-core sets. Ph.D. thesis, ETH Zurich (2006)
23. Impagliazzo, R., Jaiswal, R., Kabanets, V.: Approximate list-decoding of direct product codes and uniform hardness amplification. SIAM J. Comput. 39(2), 564–605 (2009)
24. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Prabhakaran, M., Sahai, A., Wullschleger, J.: Constant-rate oblivious transfer from noisy channels. In: Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. pp. 667–684 (2011), https://doi.org/10.1007/978-3-642-22792-9_38
25. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: STOC. pp. 21–30 (2007)
26. Ishai, Y., Prabhakaran, M., Sahai, A.: Secure arithmetic computation with no honest majority. In: TCC Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings. pp. 294–314 (2009)
27. Jetchev, D., Pietrzak, K.: How to fake auxiliary input. In: TCC. pp. 566–590 (2014)
28. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC. pp. 20–31 (1988)
29. Maurer, U.M., Tessaro, S.: A hardcore lemma for computational indistinguishability: Security amplification for arbitrarily weak prgs with optimal stretch. In: TCC. pp. 237–254 (2010)
30. Meier, R., Przydatek, B., Wullschleger, J.: Robuster combiners for oblivious transfer. In: TCC. pp. 404–418 (2007)
31. Pass, R., Venkitasubramaniam, M.: An efficient parallel repetition theorem for arthur-merlin games. In: STOC. pp. 420–429 (2007)
32. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. IACR Cryptology ePrint Archive 2019, 158 (2019)
33. Reingold, O., Trevisan, L., Tulsiani, M., Vadhan, S.P.: Dense subsets of pseudorandom sets. In: FOCS. pp. 76–85 (2008)
34. Sahai, A., Vadhan, S.P.: A complete problem for statistical zero knowledge. J. ACM 50(2), 196–249 (2003), https://doi.org/10.1145/636865.636868
35. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014 (2014)
36. Wullschleger, J.: Oblivious-transfer amplification. In: EUROCRYPT. pp. 555–572 (2007)
37. Wullschleger, J.: Oblivious transfer from weak noisy channels. In: TCC. pp. 332–349 (2009)