Resettably-Sound Resettable Zero Knowledge in Constant Rounds

Wutichai Chongchitmate
UCLA, USA
wutichai@cs.ucla.edu

Rafail Ostrovsky
UCLA, USA
rafail@cs.ucla.edu

Ivan Visconti
Università di Salerno, ITALY
visconti@unisa.it

Abstract

In FOCS 2001 Barak et al. conjectured the existence of zero-knowledge arguments that remain secure against resetting provers and resetting verifiers. The conjecture was proven true by Deng et al. in FOCS 2009 under various complexity assumptions and requiring a polynomial number of rounds. Later on in FOCS 2013 Chung et al. improved the assumptions requiring one-way functions only but still with a polynomial number of rounds.

In this work we show a *constant-round* resettably-sound resettable zero-knowledge argument system, therefore improving the round complexity from polynomial to constant. We obtain this result through the following steps.

1. We show an explicit transform from any ℓ-round concurrent zero-knowledge argument system into an $O(\ell)$-round resettable zero-knowledge argument system. The transform is based on techniques proposed by Barak et al. in FOCS 2001 and by Deng et al. in FOCS 2009. Then, we make use of a recent breakthrough presented by Chung et al. in CRYPTO 2015 that solved the longstanding open question of constructing a constant-round concurrent zero-knowledge argument system from plausible polynomial-time hardness assumptions. Starting with their construction Γ we obtain a constant-round resettable zero-knowledge argument system Λ.
2. We then show that by carefully embedding Λ inside Γ (i.e., essentially by playing a modification of the construction of Chung et al. against the construction of Chung et al.) we obtain the first constant-round resettably-sound concurrent zero-knowledge argument system Δ.
3. Finally, we apply a transformation due to Deng et al. to Δ obtaining a resettably-sound resettable zero-knowledge argument system Π, the main result of this work.

While our round-preserving transform for resettable zero knowledge requires one-way functions only, both Λ, Δ and Π extend the work of Chung et al. and as such they rely on the same assumptions (i.e., families of collision-resistant hash functions, one-way permutations and indistinguishability obfuscation for $\mathcal{P}/\texttt{poly}$, with slightly super-polynomial security).

1 Introduction

Private randomness is essential for many cryptographic tasks, including zero-knowledge (ZK) proofs [GMR85]. A natural question regards the possibility of having ZK proofs in applications where the computing machine is stateless and not equipped with a continuous source of randomness.

Resettable zero knowledge. The above question was put forth by Canetti, Goldreich, Goldwasser and Micali [CGGM00]. In particular, they considered a cheating verifier that mounts a *reset attack*, where provers are forced to execute the protocol multiple times possibly on the same inputs and random tapes, and without the ability to maintain states between executions. These attacks include the case of stateless provers, as well as provers implemented by devices that can physically be restored to their original states (e.g., through cloning, battery replacement).

More specifically, in [CGGM00], Canetti et al. introduced the notion of *resettable zero knowledge (rZK)*, in which the zero-knowledge property is required to hold even against cheating verifiers that can reset the provers to the initial states therefore forcing them to play again with the same randomnesses. This notion is closely related to *concurrent zero knowledge (cZK)* proposed earlier by Dwork, Naor and Sahai [DNS98] where a cheating verifier can engage in multiple possibly interleaving concurrent executions (called *sessions*) of the protocol. rZK is at least as hard to achieve as cZK since a resetting cheating verifier through specific reset strategies can emulate interleaving concurrent executions. In [GOVW12] Garg et al. showed that resettable *statistical* zero knowledge is possible for several interesting languages.

Round complexity of cZK and rZK. Constant-round cZK under plausible hardness assumptions has been a long-standing challenging open question that received a positive answer in the work of Chung et al. [CLP15] by means of indistinguishability obfuscation ($i\mathcal{O}$) [CLP15]. Instead the situation for rZK is worse. Canetti et al. in [CGGM00] constructed rZK proofs in the *standard model* relying on standard cryptographic assumptions but with polynomial round complexity[1].

The round complexity was then improved to poly-logarithmic in [KP01]. The state of affair leaves the following open problem.

Open Problem 1: *is there a construction for rZK with sub-logarithmic rounds?*

Resettably-sound zero knowledge. Barak, Goldreich, Goldwasser and Lindell [BGGL01] considered the natural opposite setting, called *resettably-sound zero knowledge (rsZK)* arguments, where soundness is required to hold even against cheating provers that can reset the verifiers forcing them to re-use the same random tapes. The standard zero-knowledge property remains untouched. They showed a constant-round construction assuming collision-resistant hash functions. The recent work of [COP+14] reached optimal round complexity and assumptions (i.e., 4 rounds and one-way functions).

The simultaneous resettability conjecture. Barak et al. in [BGGL01] conjectured the existence of a zero-knowledge argument that is secure simultaneously against resetting verifiers and against resetting provers: a resettably-sound resettable zero-knowledge argument system. The conjecture was proven true by Deng, Goyal and Sahai [DGS09] that presented a construction with a polynomial number of rounds and assuming collision-resistant hash functions and

[1] In addition they proposed a mild setup assumption based on bare public keys showing that it is sufficient for constant-round resettable zero knowledge. Follow up work optimized round complexity and complexity assumptions for rZK with bare public keys [MR01, DCPV04, DL07, YZ07, SV12].

trapdoor permutations. The computational assumptions have been improved to one-way functions [OV12, COPV13, CPS13, BP13, BP15a], while the barrier of the polynomial round complexity has remained untouched so far.

Open Problem 2: *is there a construction for resettably-sound rZK with sub-polynomial rounds?*

We stress that by relaxing the security against resetting verifiers from zero knowledge to witness indistinguishability, then constant-round simultaneous resettability is possible. Indeed just 1 or 2 rounds (i.e., ZAPs) are needed to obtain proofs, and a larger constant number of rounds is sufficient to obtain arguments of knowledge [COSV12].

1.1 Our Results

In this paper, we answer the above questions positively. In the main result we construct a *constant-round* simultaneous resettable zero-knowledge argument for $\mathcal{NP}$. Our result requires the existence of families of collision-resistant hash functions, one-way permutations and indistinguishability obfuscation ($i\mathcal{O}$) for $\mathcal{P}/\texttt{poly}$ (with slightly super-polynomial security). These assumptions are the same as the ones in [CLP15] that showed a constant-round concurrent zero-knowledge argument for $\mathcal{NP}$. Our result makes uses of the protocol of [CLP15] twice in some nested way. More precisely, the first time we use the protocol of [CLP15] Γ is to obtain a constant-round rZK argument Λ. Then we start again with Γ and we modify it by using Λ (that is a modification of Γ) as subprotocol in the opposite direction (i.e., the verifier will prove something to the prover). Therefore we roughly use the protocol of [CLP15] against the protocol of [CLP15] which is somehow intriguing. This nested use of the protocol of [CLP15] allows us to obtain a constant-round resettably-sound concurrent zero-knowledge argument Δ. We can then apply a compiler due to [DGS09] to Δ therefore obtaining our main argument system Π that is secure simultaneously against resetting provers and resetting verifiers needing only a constant number of rounds.

We now give our formal theorems that specify the precise complexity assumptions.

Theorem 1.1. *Assuming the existence of one-way functions, than any ℓ-round concurrent zero-knowledge argument system can be transformed in a $\mathcal{O}(\ell)$-round resettable zero-knowledge argument system.*

Theorem 1.2. *Assuming the existence of collision-resistant hash functions, one-way permutations and indistinguishability obfuscation for $\mathcal{P}/\texttt{poly}$ (with slightly super-polynomial security), there exists a constant-round resettably-sound resettable zero-knowledge argument system for $\mathcal{NP}$.*

1.2 Main Tools and Our New Techniques

Our constructions rely on new ideas as well as a combined use of several techniques used in previous results on concurrent, resettable and resettably-sound zero knowledge. We start by briefly describing the important tools that we use along with our new techniques for our constructions.

Barak's non-black-block protocol. The starting point is Barak's non-black-box zero-knowledge argument for $\mathcal{NP}$ [Bar01] that works as follows. The prover P sends a commitment $c \in \{0,1\}^n$ of 0 to the verifier V. The verifier V then sends a uniformly generated random string $r \in \{0,1\}^{2n}$. Finally, the prover gives a witness-indistinguishable universal argument (WIUA) that $x \in L$ or there exists $\sigma \in \{0,1\}^n$ such that c is a commitment of a program M such that $M(\sigma) = r^2$. The soundness follows from the binding of the commitment scheme and the soundness of the WIUA as any program M committed by the cheating prover does not have r in its support with overwhelming probability. For the zero-knowledge property, the simulator uses the code of the adversary. Indeed it commits to a program M corresponding to the code of V^*, the cheating verifier. Let σ be the commitment. We have that $M(\sigma) = r$ and σ is short compared to r.

Chung et al.'s constant-round cZK argument. In [CLP15], Chung et al. construct a constant-round cZK argument by using unique $\mathcal{P}$-certificate systems [CLP13] with delegatable CRS generation and $i\mathcal{O}$. Informally, a $\mathcal{P}$-certificate system allows an efficient prover to convince a verifier of the validity of any deterministic polynomial-time computation $M(x) = y$ using a certificate of fixed (polynomial) length, independent of the size and the running time of M. The verifier can also verify the certificate in fixed (polynomial) time, independent of the running time of M. In a $\mathcal{P}$-certificate system with delegatable CRS generation, the certificate is generated using a common reference string (CRS) that can be computed by using resources delegated by the verifier. More specifically, in this $\mathcal{P}$-certificate system, the $\mathcal{P}$-certificate verifier generates public and private parameters, PP and κ, and sends PP to the $\mathcal{P}$-certificate prover. The $\mathcal{P}$-certificate prover uses the public parameter PP and the statement $q = (M, x, y)$ to deterministically compute a short digest d, whose length is independent of the length of q, and sends it to the $\mathcal{P}$-certificate verifier. The $\mathcal{P}$-certificate verifier then computes the CRS from d and κ. Finally, the $\mathcal{P}$-certificate prover computes the certificate from the CRS and q. The $\mathcal{P}$-certificate system is unique if there exists at most one accepted certificate for any statement and CRS.

The argument of [CLP15] proceeds similarly to Barak's argument with the following modifications. In the last step, instead of requiring the prover P to prove that $x \in L$ or there exists σ such that c is a commitment to a program M such that $M(\sigma) = r$, the prover provides a special-sound witness-indistinguishability proof that $x \in L$ or there exists a $\mathcal{P}$-certificate π which certifies that $M(\sigma) = r$ for some short string σ. Additionally, P also commits and gives a WIUA proving that either $x \in L$ or there exists a $\mathcal{P}$-certificate for the statement $q = (M, \sigma, r)$ before receiving the public parameter PP from V. Note that since the honest prover of the protocol in [CLP15] has a witness for $x \in L$, it can just ignore CRS, d and q, and simply commit to zeroes. In order to allow the zero-knowledge simulator (note that an honest prover will just use the witness for $x \in L$) to compute the CRS from d and κ, the verifier sends an obfuscated program with κ embedded inside, that allows the simulator to compute CRS from d committed earlier. Finally, V also provides a zero-knowledge argument that the obfuscated program is computed correctly.

The simulator does not know a witness for $x \in L$ but is instead able to commit to the code of the adversary. More formally, the simulator is divided in two parts: S_1, which takes a $\mathcal{P}$-certificates π_i in the i-th round as an input, and interacts with the verifier V^*, and S_2 which, in the i-th round provides $\mathcal{P}$-certificates certifying that S_1 on input $(1, \pi_1), \ldots, (i-1, \pi_{i-1})$ outputs m_i. Instead of committing to a program M, using the verifier V^*'s code, such that $M(\sigma) = r$ for some short string σ, the simulator $S = (S_1, S_2)$ commits to a program $\widetilde{S}_1$. The program, on input $(1^n, j, s)$, runs an interaction between S_1 and V^* for j rounds using s as a seed to generate pseudorandom coins while having an access to the oracle $\mathcal{O}_{\mathsf{V_{cert}}}$ which provides $\mathcal{P}$-certificates. This prevents the nesting of concurrent sessions which may result in the blow-up in the running time as the expensive part of S consists in generating the $\mathcal{P}$-certificates. The simulator of the protocol in [CLP15] can therefore

[2]Since the size of M may not be known in advance, the commitment is to the hash of the program M using a hash function h sampled from a family of collision-resistant hash functions chosen in the beginning of the protocol by the verifier. The soundness is also based on the collision resistance of h.

succeed in the special-sound witness-indistinguishability proof for the statement $x \in L$ or there exists a $\mathcal{P}$-certificate π which certifies that $\widetilde{S}_1^{\mathcal{O}_{\mathsf{V_{cert}}}}(1^n, j, s) = r$ for some short string $(1^n, j, s)$ using the output from the oracle as a witness.

Deng, Goyal and Sahai's transformation. In [GS08, DGS09], Deng, Goyal and Sahai construct a hybrid resettably-sound and relaxed concurrent zero-knowledge argument Π_{DGS}. Then they apply a series of transformations to achieve simultaneous resettability.

Relaxed concurrent zero knowledge allows verifiers to interact in multiple sessions with independent provers. However, the zero-knowledge property only guarantees for "relaxed" concurrent verifiers whose random coins are fixed in the beginning of each session, independently of sessions that start after that session. Note that any concurrent zero-knowledge argument/proof is also relaxed concurrent zero-knowledge as any relaxed concurrent verifier is also a concurrent verifier.

Hybrid resettable soundness means that the verifier can be separated into two parts, V_1 and V_2. V_1 directly interacts with P, may relay some messages between P and V_2, and can be reset by a cheating prover. V_2 only interacts with V_1, cannot be reset by a cheating prover, and is responsible to decide whether to "accept" or "reject" the argument. Moreover, for each *determining message* (the first message V_2 receives in the protocol), P cannot find two different messages that P can convince V_1 to pass to V_2 in each round. We refer to [GS08] for a precise definition. Note that any resettably-sound argument is also hybrid resettably sound by letting V_1 behave as V except that instead of accepting the argument, it sends a message to V_2, and V_2 always accepts the argument when it receives a message from V_1.

The transformation of Deng et al. uses ZAPs and one-way functions to achieve simultaneous resettability and only increases the round complexity by a constant factor. However, the round complexity of Π_{DGS} is polynomial [DGS09]. Thus, their simultaneously resettable argument system also requires polynomial rounds.

Inapplicability of the transformation of [DGS09] to the construction of Chung et al. [CLP15]. Intuitively, one may try to apply the transformation of [DGS09] to the constant-round concurrent zero-knowledge argument in [CLP15] to get simultaneous resettability. However, in order for the result of the transformation to be simultaneously resettable, it is required that the starting protocol be relaxed concurrent zero-knowledge and hybrid resettably sound. While the protocol in [CLP15] is concurrent zero-knowledge, which implies that it is relaxed concurrent zero-knowledge, we argue that if the (non-resettable) ZK argument (proving that the obfuscated program is computed correctly) is not zero-knowledge against *resetting* verifiers, then the protocol can not be proved hybrid resettably sound. Two reasons follow below.

1. Suppose in the extreme case that there exists an adversarial resetting prover for the argument of [CLP15] that runs a resetting adversary $\mathcal{A}_{ZK}$ in the (non-resetting) zero-knowledge subprotocol in which the honest verifier proves that the obfuscated program is computed correctly. Remember that the zero-knowledge subprotocol could also be an argument of knowledge admitting a black-box (rewinding) extractor. By managing to run $\mathcal{A}_{ZK}$, the adversarial resetting prover could succeed in extracting some relevant information (e.g., the secret parameter for $\mathcal{P}$-certificate CRS generation, that is used in the (non-resettable) ZK argument proven by the verifier to prover to guarantee the correctness of the obfuscated program). However, according to the definition of hybrid resettable soundness, we need to consider two separate parts of the verifier $V = (V_1, V_2)$. One out of V_1 and V_2 will run as prover of the ZK argument proving that the obfuscated program is generated correctly. If the (non-resettable) ZK argument is played by V_1 (as a prover), which can be reset, the malicious prover of the protocol in [CLP15]

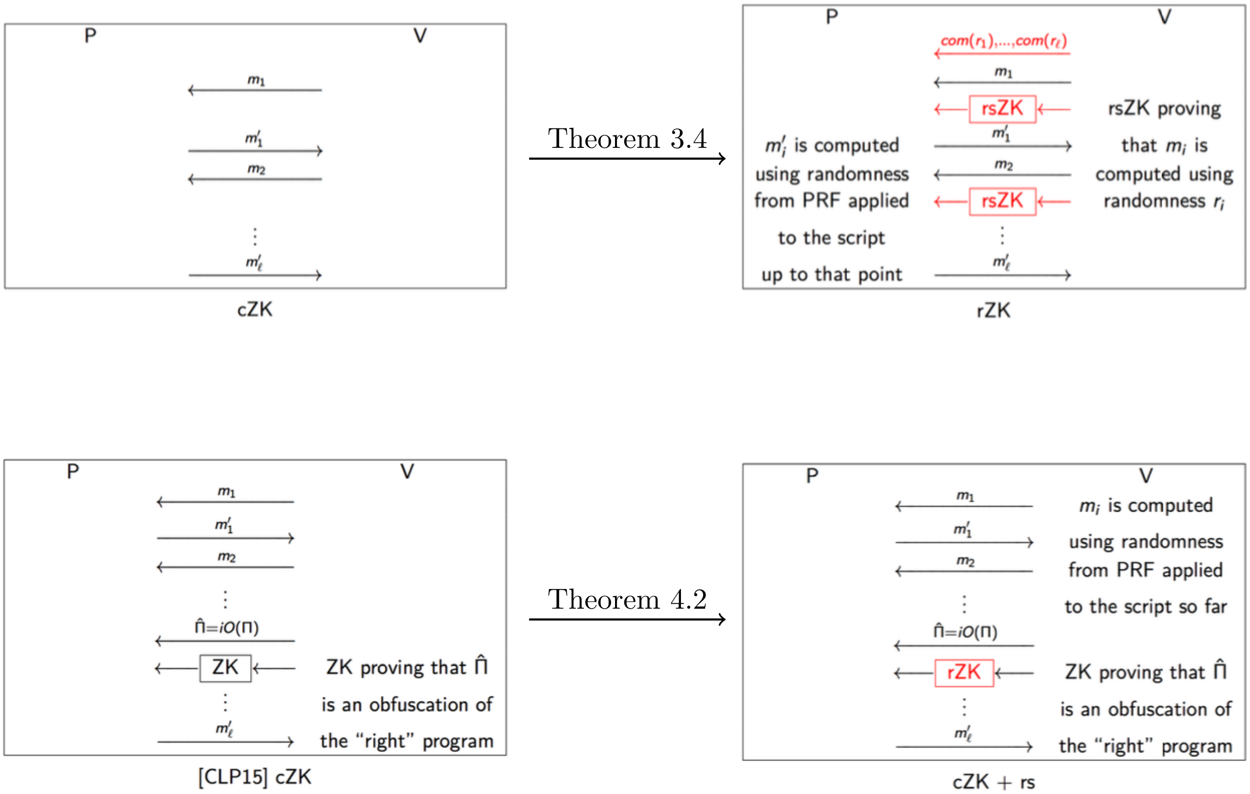

Figure 1: Our Transformations of Zero-Knowledge Argument Systems

can run $\mathcal{A}_{ZK}$ to learn some relevant information (.e.g, the secret parameter), and this can potentially be used to generate a certificate for a false statement. On the other hand, if the (non-resettable) ZK argument is played by V_2 (as a prover) then since the messages of the verifier of this argument are not fixed by a determining message in the protocol of [CLP15], we have that V_2 can receive two different messages for the same determining message, and thus, even in this case, the protocol is not hybrid resettably sound.

2. The $\mathcal{P}$-certificate generation in the protocol of [CLP15] cannot be transformed into a resettably-sound protocol using the techniques of [BGGL01]. This is because the $\mathcal{P}$-certificate system is not public coin. Recall that the proof of resettable soundness in [BGGL01] uses the reduction to the non-resettable case by starting (by contradiction) with a (successful) resetting prover. If we repeat here the same reduction, we have that the non-resetting prover runs all but one session by simulating the verifier itself. Of course this requires to generate legit verifier messages under reset attacks. When trying to send the legit verifier messages, the non-resetting prover may send the obfuscated program of the real verifier of the reduction to the resetting prover, and the resetting prover may reset to the step after which it receives the public parameter for the $\mathcal{P}$-certificate. In that case, the non-resetting prover will not be able to generate a new obfuscated program as specified in the protocol without knowing the secret parameter.

1.3 Our Approach

In order to get a constant-round resettably-sound concurrent zero-knowledge argument system, we consider the protocol from [CLP15] which is constant round and concurrent zero knowledge, but not resettably sound. As discussed above, there are two main problems that separate the protocol of [CLP15] from resettable soundness: the non-resettable ZK argument for $i\mathcal{O}$ and the delegatable CRS generation of the $\mathcal{P}$-certificate system, which cannot be generated without knowing the secret parameter generated in the earlier step.

Solving the first problem. We resolve the first problem by constructing a constant-round resettable ZK argument from the concurrent ZK argument of [CLP15]. This transformation is implicit in some previous works on the topic [BGGL01, DGS09]. We explicitly present it here for completion.

Unlike the concurrent verifier, the resetting verifier can exploit the reuse of the random tape during the resetting attack by sending different messages in order to extract additional information from the prover. We prevent such behavior by requiring 1) the verifier to commit to its random tape using a statistically binding commitment scheme and 2) to provide a zero-knowledge argument that it actually uses the random tape it has committed to. Note that since the verifier can reset the prover, a zero-knowledge argument without resettable soundness cannot be used by the verifier to prove that the verifier uses the committed random bits. Thus, the argument system needs to be resettably sound. In order to preserve the round complexity, this subprotocol must be constant round. This can be done using the 4-round resettably-sound zero-knowledge argument by Chung et al [COP$^+$14]. A similar technique has been used in [GS09] for resettably-secure computation.

We note that the constant-round rsZK argument and the commitment scheme can be constructed from one-way functions, which is assumed for the constant-round concurrent zero-knowledge argument in [CLP15]. Thus, applying this transformation on the protocol does not require any extra assumption. It turns out that the technique we use can be generalized to a compiler that works with *any* concurrent ZK protocol. The round complexity of the resulting protocol only increases by a constant factor.

Our compiler turning any concurrent ZK argument into a resettable ZK argument works as follows. First, we replace the random coin used by the prover to generate his messages with outputs of a PRF. This step allows a prover with fixed random tape to send different messages when the resetting verifier changes its messages after resetting similarly to the technique used in [BGGL01] against resetting provers. Additionally, the verifier commits to its random coins used in each round at the beginning of the protocol. After sending each message, the verifier gives a constant-round resettably-sound ZK argument that it uses the random coins committed in the first round. This modification ensures that the verifier follows the protocol in every session.

Solving the second problem. In order to solve the second problem, we observe that while the protocol of [CLP15] is not public-coin, it is "almost public-coin". By almost public-coin, we mean that, beside the ZK argument which is replaced by rZK argument above, there is only one message from the verifier that cannot be generated independently as public-coin, but depends on a hidden randomness. Thus, we modify the technique in [BGGL01] to resolve the problem in two steps as follows.

First, we consider a modified version of the protocol of [CLP15], in which we can prove its (non-resettable) soundness. In this protocol, the round in which the message from V cannot be generated with uniformly random coins is repeated m times, where $m = \texttt{poly}(n)$ is the upper bound on the running time of a cheating prover P^*. More specifically, after receiving the public parameter for $\mathcal{P}$-certificate, the prover for the modified protocol P_S repeatedly commits to and proves the validity of the digest d of his statement while the verifier V_S repeatedly replies with the obfuscated program verifying the committed value and output the CRS for the $\mathcal{P}$-certificate. P_S then chooses which commitment and obfuscated program pair P_S will use to complete the protocol. Because of the security of the $i\mathcal{O}$, P_S does not learn the secret parameter for the $\mathcal{P}$-certificate even after m repetitions. Thus, the resulting protocol is still sound.

Then we reduce the resettable soundness of the final protocol to the non-resettable soundness of the above protocol with polynomial reduction in success probability as follows. Given a resetting prover P^*, we construct a non-resetting prover P^*_S by internally simulating P^* interaction with a verifier V, and randomly choosing which of the m repetitions will lead to accepting transcript. For other repetitions, P^*_S will generate the parameters for $\mathcal{P}$-certificate itself to get around the non-public-coin situation. In the case that P^*_S guesses the accepting transcript correctly , which occurs with probability $1/m$, it will convince the verifier V_S with the accepting transcript from the simulation.

1.4 Open Questions

Unlike the above compiler from concurrent ZK to resettable ZK, our construction for resettably sound resettable zero knowledge uses in a non-black-box way the protocol of [CLP15].

Our work leaves open the natural questions of producing a generic round-preserving transform from cZK to rZK, and of obtaining constant-round resettably sound resettable zero knowledge under more standard complexity assumptions.

2 Definitions

A polynomial-time relation R is a relation for which it is possible to verify in time polynomial in $|x|$ whether $R(x, w) = 1$. Let us consider an $\mathcal{NP}$-language L and denote by R_L the corresponding polynomial-time relation such that $x \in L$ if and only if there exists w such that $R_L(x, w) = 1$. We will call such a w a *valid witness for* $x \in L$. Let λ denote the security parameter. A *negligible* function $\nu(\lambda)$ is a non-negative function such that for any constant $c < 0$ and for all sufficiently large λ, $\nu(\lambda) < \lambda^c$. We will denote by $\Pr_r[X]$ the probability of an event X over coins r, and $\Pr[X]$ when r is not specified. The abbreviation "PPT" stands for probabilistic polynomial time. For a randomized algorithm A, let $A(x; r)$ denote running A on an input x with random coins r. If r is chosen uniformly at random with an output y, we denote $y \leftarrow A(x)$. For a pair of interactive Turing machines (P, V), let $\langle P, V\rangle(x)$ denotes V's output after interacting with P upon common input x. We say V accepts if $\langle P, V\rangle(x) = 1$ and rejects if $\langle P, V\rangle(x) = 0$. We denote by $\mathtt{view}^{P(w)}_{V(x,z)}$ the view (i.e., its private coins and the received messages) of V during an interaction with $P(w)$ on common input x and auxiliary input z. We will use the standard notion of computational indistinguishability [GM84].

We now give definitions for interactive proof/argument systems with all variants that are useful in this work.

Definition 2.1 (interactive proofs [GMR85]). *An* interactive proof system *for the language L, is a pair of interactive Turing machines (P, V) running on common input x such that:*

- *Efficiency: P and V are* PPT.
- *Completeness: For every $\lambda \in \mathbb{N}$ and for every pair (x, w) such that $R_L(x, w) = 1$,*

$$\Pr[\langle P(w), V\rangle(1^\lambda, x) = 1] = 1.$$

- *Soundness*[3]*: There exists a negligible function $\nu(\cdot)$ such that for every pair of interactive Turing machines (P_1^*, P_2^*)*

$$\Pr[(x, z) \leftarrow P_1^*(1^\lambda) : x \notin L \wedge \langle P_2^*, V\rangle(1^\lambda, x) = 1] < \nu(\lambda).$$

In the above definition we can relax the soundness requirement by considering P^* as PPT. In this case, we say that (P, V) is an *interactive argument system* [BCC88].

Definition 2.2 (zero-knowledge arguments [GMR85])**.** *Let (P, V) be an interactive argument system for a language L. We say that (P, V) is zero knowledge (ZK) if, for any probabilistic polynomial-time adversary V^*, there exists a probabilistic polynomial-time algorithm S_{V^*} such for all auxiliary inputs z and all pairs $(x, w) \in R_L$ the ensembles $\{\mathtt{view}_{V^*(x,z)}^{P(w)}\}$ and $\{S_{V^*}(x, z)\}$ are computationally indistinguishable.*

Suppose (P, V) is used as a sub-protocol of another interactive protocol (A^1, A^2) where A^1 runs P and A^2 runs V. We call a Turing machine A^1_α a *residual prover* if A^1_α runs A^1 on inputs $\alpha = (\alpha_1, \ldots, \alpha_\ell)$ from A^2 up to and including the ℓth round when A^1 invokes P. A *residual verifier* A^2_α is defined similarly by switching A^1 and A^2. Note that the residual prover is invoked when simulating V (for soundness) while the residual verifier is invoked when simulating P (for zero-knowledge).

Definition 2.3 (resetting adversary [CGGM00])**.** *Let (P, V) be an interactive proof or argument system for a language L, $t = \mathtt{poly}(\lambda)$, $\bar{x} = x_1, \ldots, x_t$ be a sequence of common inputs and $\bar{w} = w_1, \ldots, w_t$ the corresponding witnesses (i.e., $(x_i, w_i) \in R_L$) for $i = 1, \ldots, t$. Let $r_1, \ldots, r_t$ be independent random tapes. We say that a PPT V^* is a* resetting *verifier if it concurrently interacts with an unbounded number of independent copies of P by choosing for each interaction the value i so that the common input will be $x_i \in \bar{x}$, and the prover will use witness w_i, and choosing j so that the prover will use r_j as randomness, with $i, j \in \{1, \ldots, t\}$. The scheduling or the messages to be sent in the different interactions with P are freely decided by V^*. Moreover we say that the transcript of such interactions consists of the common inputs $\bar{x}$ and the sequence of prover and verifier messages exchanged during the interactions. We refer to $\mathtt{view}_{V^*(\bar{x},z)}^{P(\bar{w})}$ as the random variable describing the content of the random tape of V^* and the transcript of the interactions between P and V^*, where z is an auxiliary input received by V^*.*

Definition 2.4 (resettable zero knowledge [CGGM00])**.** *Let (P, V) be an interactive argument system for a language L. We say that (P, V) is resettable zero knowledge (rZK) if, for any PPT resetting verifier V^* there exists a expected probabilistic polynomial-time algorithm S_{V^*} such that the for all pairs $(\bar{x}, \bar{w}) \in R_L$ the ensembles $\{\mathtt{view}_{V^*(\bar{x},z)}^{P(\bar{w})}\}$ and $\{S_{V^*}(\bar{x}, z)\}$ are computationally indistinguishable.*

The definition of concurrent zero knowledge can be seen as a relaxation of the one of resettable zero knowledge. The adversarial concurrent verifier has the same power of the resetting verifier except it can not ask the prover to run multiple sessions with the same randomness.

Definition 2.5 (concurrent adversary)**.** *Let (P, V) be an interactive proof or argument system for a language L, $t = \mathtt{poly}(\lambda)$, $\bar{x} = x_1, \ldots, x_t$ be a sequence of common inputs and $\bar{w} = w_1, \ldots, w_t$ the corresponding witnesses (i.e., $(x_i, w_i) \in R_L$) for $i = 1, \ldots, t$. We say that a PPT V^* is a*

[3]This version of soundness given by [CLP15] is slightly different from standard version with one Turing machine P^*. Separating them makes the proof cleaner while it is still equivalent to the standard version.

concurrent *verifier if it concurrently interacts with an unbounded number of independent copies of P by choosing for each interaction the value i so that the common input will be $x_i \in \bar{x}$, and the prover will use witness w_i. Each copy of P runs with independent randomness. The scheduling or the messages to be sent in the different interactions with P are freely decided by V^*. Moreover we say that the transcript of such interactions consist of the common inputs $\bar{x}$ and the sequence of prover and verifier messages exchanged during the interactions. We refer to $\mathtt{view}^{P(\bar{w})}_{V^*(\bar{x},z)}$ as the random variable describing the content of the random tape of V^* and the transcript of the interactions between P and V^*, where z is an auxiliary input received by V^*.*

Definition 2.6 (concurrent zero knowledge [DNS98]). *Let (P, V) be an interactive argument system for a language L. We say that (P, V) is concurrent zero knowledge (cZK) if, for any PPT concurrent verifier V^* there exists a probabilistic polynomial-time algorithm S_{V^*} such that the for all pairs $(\bar{x}, \bar{w}) \in R_L$ the ensembles $\{\mathtt{view}^{P(\bar{w})}_{V^*(\bar{x},z)}\}$ and $\{S_{V^*}(\bar{x}, z)\}$ are computationally indistinguishable.*

Definition 2.7 (witness indistinguishability [FS90]). *Let L be a language in $\mathcal{NP}$ and R_L be the corresponding relation. An interactive argument (P, V) for L is* witness indistinguishable *(WI) if for every verifier V^*, every pair (w_0, w_1) such that $(x, w_0) \in R_L$ and $(x, w_1) \in R_L$ and every auxiliary input z, the following ensembles are computationally indistinguishable:*

$$\{\mathtt{view}^{P(w_0)}_{V^*(x,z)}\} \quad \text{and} \quad \{\mathtt{view}^{P(w_1)}_{V^*(x,z)}\}.$$

Definition 2.8 (resettable WI [CGGM00]). *Let L be a language in $\mathcal{NP}$ and R_L be the corresponding relation. An interactive argument (P, V) for L is* resettable *witness indistinguishable (rWI) if for every PPT resetting verifier V^* every $t = \mathtt{poly}(\lambda)$, and every pair $(\bar{w}^0 = (w^0_1, \ldots, w^0_t), \bar{w}^1 = (w^1_1, \ldots, w^1_t))$ such that $(x_i, w^0_i) \in R_L$ and $(x_i, w^1_i) \in R_L$ for $i = 1, \ldots, t$, and any auxiliary input z, the following ensembles are computationally indistinguishable:*

$$\{\mathtt{view}^{P(\bar{w}^0)}_{V^*(\bar{x},z)}\} \quad \text{and} \quad \{\mathtt{view}^{P(\bar{w}^1)}_{V^*(\bar{x},z)}\}.$$

In [DN00], a construction of 2-round resettable witness-indistinguishable proof based on NIZK proofs has been shown, and then in [GOS06], a non-interactive resettable witness-indistinguishable proof has been shown by relying on specific number-theoretic assumptions, and from $i\mathcal{O}$ [BP15b].

Let us recall the definition of resettable soundness due to [BGGL01].

Definition 2.9 (resettably-sound arguments [BGGL01]). *A resetting attack of a cheating prover P^* on a resettable verifier V is defined by the following two-step random process, indexed by a security parameter λ.*

1. *Uniformly select and fix $t = \mathtt{poly}(\lambda)$ random-tapes, denoted $r_1, \ldots, r_t$, for V, resulting in deterministic strategies $V^{(j)}(x) = V_{x,r_j}$ defined by $V_{x,r_j}(\alpha) = V(x, r_j, \alpha)$,[4] where $x \in \{0,1\}^\lambda$ and $j \in [t]$. Each $V^{(j)}(x)$ is called an incarnation of V .*

2. *On input 1^λ, machine P^* is allowed to initiate $\mathtt{poly}(\lambda)$-many interactions with the $V^{(j)}(x)$'s. The activity of P^* proceeds in rounds. In each round P^* chooses $x \in \{0,1\}^\lambda$ and $j \in [t]$, thus defining $V^{(j)}(x)$, and conducts a complete session with it.*

Let (P, V) be an interactive argument for a language L. We say that (P, V) is a resettably-sound argument *for L if the following condition holds:*

[4]Here, $V(x, r, \alpha)$ denotes the message sent by the strategy V on common input x, random-tape r, after seeing the message-sequence α.

- Resettable-soundness: *For every polynomial-size resetting attack, the probability that in some session the corresponding* $V^{(j)}(x)$ *has accepted and* $x \notin L$ *is negligible.*

Definition 2.10 (commitment scheme)**.** *Given a security parameter* 1^λ*, a commitment scheme* com *is a two-phase protocol between two* PPT *interactive algorithms, a sender* S *and a receiver* R*. In the commitment phase* S *on input a message* m *interacts with* R *to produce a commitment* $c = \mathsf{com}(m)$*. In the decommitment phase,* S *sends to* R *a decommitment information* d *such that* R *accepts* m *as the decommitment of* c*.*

Formally, we say that com *is a perfectly binding commitment scheme if the following properties hold:*

Correctness:

- *Commitment phase. Let* $c = \mathsf{com}(m)$ *be the commitment of the message* m *given as output of an execution of* com *where* S *runs on input a message* m*. Let* d *be the private output of* S *in this phase.*
- *Decommitment phase*[5]*.* R *on input* m *and* d *accepts* m *as decommitment of* c*.*

Statistical (resp. Computational) Hiding ([Lin10]): *for any adversary (resp.* PPT *adversary)* $\mathcal{A}$ *and a randomly chosen bit* $b \in \{0,1\}$*, consider the following hiding experiment* $\mathsf{ExpHiding}^b_{\mathcal{A},\mathsf{com}}(\lambda)$*:*

- *Upon input* 1^λ*, the adversary* $\mathcal{A}$ *outputs a pair of messages* m_0, m_1 *that are of the same length.*
- S *on input the message* m_b *interacts with* $\mathcal{A}$ *to produce a commitment of* m_b*.*
- $\mathcal{A}$ *outputs a bit* b' *and this is the output of the experiment.*

For any adversary (resp. PPT *adversary)* $\mathcal{A}$*, there exist a negligible function* ν*, s.t.:*

$$\left| \Pr[\mathsf{ExpHiding}^0_{\mathcal{A},\mathsf{com}}(\lambda) = 1] - \Pr[\mathsf{ExpHiding}^1_{\mathcal{A},\mathsf{com}}(\lambda) = 1] \right| < \nu(\lambda).$$

Statistical (resp. Computational) Binding: *for every commitment* $\mathtt{com}$ *generated during the commitment phase by a possibly malicious unbounded (resp. malicious* PPT*) sender* S^* *there exists a negligible function* ν *such that* S^**, with probability at most* $\nu(\lambda)$*, outputs two decommitments* (m_0, d_0) *and* (m_1, d_1)*, with* $m_0 \neq m_1$*, such that* R *accepts both decommitments.*

We also say that a commitment scheme is perfectly binding *iff* $\nu(\lambda) = 0$*.*

In this paper, we consider non-interactive perfectly binding computationally hiding commitment schemes, which can be constructed from one-to-one one-way functions [Gol01]. Two-message statistically binding commitment schemes can be obtained from one-way functions [Nao91, HILL99].

Definition 2.11 (pseudorandom function (PRF))**.** *A family of functions* $\{f_s\}_{s \in \{0,1\}^*}$ *is called* pseudorandom *if for all adversarial PPT machines* $\mathcal{A}$*, for every positive polynomial* $p()$*, and sufficiently large* $\lambda \in \mathbb{N}$*, it holds that*

$$|\Pr[\mathcal{A}^{f_s}(1^\lambda) = 1] - \Pr[\mathcal{A}^F(1^\lambda) = 1]| \leq \frac{1}{p(\lambda)}.$$

where $|s| = n$ *and* F *denotes a truly random function.*

[5] In this paper we consider a non-interactive decommitment phase only.

Definition 2.12 (indistinguishability obfuscation)**.** *A uniform machine $i\mathcal{O}$ is an* indistinguishability obfuscator *for a class of deterministic circuits $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if it satisfies the following:*

- *Correctness: For all security parameter $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all input x,*

$$\Pr[\Lambda \leftarrow i\mathcal{O}(1^\lambda, C) : \Lambda(x) = C(x)] = 1.$$

- *Security: For every non-uniform PPT sampleable distribution $\mathcal{D}$ and adversary $\mathcal{A}$, there exists a negligible function ν such that for sufficiently large $\lambda \in \mathbb{N}$, if*

$$\Pr[(C_1, C_2, z) \leftarrow \mathcal{D} : \forall x, C_1(x) = C_2(x)] > 1 - \nu(\lambda),$$

then

$$\begin{array}{l}\Pr[(C_1, C_2, z) \leftarrow \mathcal{D} : \mathcal{A}(i\mathcal{O}(1^\lambda, C_1), z) = 1] \\ -\Pr[(C_1, C_2, z) \leftarrow \mathcal{D} : \mathcal{A}(i\mathcal{O}(1^\lambda, C_2), z) = 1]\end{array} \leq \nu(\lambda).$$

We say an iO is super-polynomially secure *if there is a super-polynomial function T such that the above condition holds for all adversary $\mathcal{A}$ running in time at most $T(\lambda)$.*

Let $R_U = \{((M, x, t), w) : M$ accepts (x, w) in t steps$\}$, $S_U = \{(M, x, t) : \exists w, ((M, x, t), w) \in R_U\}$ and $R_U(M, x, t) = \{w : ((M, x, t), w) \in R_U\}$. Let $T_M(x, w)$ denote the number of steps made by M on input (x, w).

Definition 2.13 (universal argument [BG08])**.** *A pair of interactive Turing machines (P, V) is called a* universal argument system *if it satisfies the following properties:*

- *Efficient verification: There exists a polynomial p such that for any $y = (M, x, t)$, the total time spent by the (probabilistic) verifier V, on common input y, is at most $p(|y|)$. In particular, all messages exchanged in the protocol have length smaller than $p(|y|)$.*

- *Completeness via a relatively efficient prover: For every $((M, x, t), w) \in R_U$,*

$$\Pr[\langle P(w), V\rangle(M, x, t) = 1] = 1.$$

Furthermore, there exists a polynomial q such that for every $((M, x, t), w) \in R_U$, the total time spent by $P(w)$, on common input (M, x, t), is at most $q(|M| + T_M(x, w)) \leq q(|M| + t)$.

- *Computational soundness: For every polynomial-size circuit family $\{\widetilde{P}_n\}_{n\in\mathbb{N}}$, and every $(M, x, t) \in \{0,1\}^n \setminus S_U$, there exists a negligible function ν such that*

$$\Pr[\langle \widetilde{P}_n, V\rangle(M, x, t) = 1] < \nu(n).$$

- *Weak proof-of-knowledge property: For every positive polynomial p there exists a positive polynomial p' and a probabilistic polynomial-time oracle machine E such that the following holds: for every polynomial-size circuit family $\{\widetilde{P}_n\}_{n\in\mathbb{N}}$, and every sufficiently long $y = (M, x, t) \in \{0,1\}^*$, if $\Pr[\langle \widetilde{P}_n, V\rangle(y) = 1] > 1/p(|y|)$, then*

$$\Pr_r[\exists w = w_1 \dots w_t \in R_U(y), \forall i \in [t], E_r^{\widetilde{P}_n}(y, i) = w_i] > 1/p'(|y|)$$

where $E_r^{\widetilde{P}_n}$ denotes the function defined by fixing the random-tape of E to r and providing it with oracle access to $\widetilde{P}_n$.

By abusing the notation, we let E be the oracle machine, running in time $\mathtt{poly}(n) \cdot t$, that extracts the whole witness. We call E a *global proof-of-knowledge extractor*. Note that E is not necessarily polynomial time.

Definition 2.14 (witness-indistinguishable universal argument [BG08]). *A universal argument system, (P, V), is called* witness-indistinguishable (WIUA) *if, for every polynomial p, every polynomial-size circuit family $\{V_n^*\}_{n\in\mathbb{N}}$, and every three sequences $\langle y_n = (M_n, x_n, t_n)\rangle_{n\in\mathbb{N}}$, $\langle w_n^1\rangle_{n\in\mathbb{N}}$ and $\langle w_n^2\rangle_{n\in\mathbb{N}}$ such that $|y_n| = n$, $t_n \leq p(|x_n|)$ and $(y_n, w_n^1), (y_n, w_n^2) \in R_U$, the probability ensembles $\{\langle P(w_n^1), V_n^*\rangle(y_n)\}_{n\in\mathbb{N}}$ and $\{\langle P(w_n^2), V_n^*\rangle(y_n)\}_{n\in\mathbb{N}}$ are computationally indistinguishable.*

Theorem 2.15 ([BG08]). *Assuming the existence of families of collision-resistant hash functions, there exists a 4-round public-coin WIUA.*

Definition 2.16 (special-sound witness-indistinguishable proof [CLP15]). *A 4-round public-coin interactive proof for the language $L \in \mathcal{NP}$ with witness relation R_L is* special-sound *with respect to R_L, if for any two transcripts $(\delta, \alpha, \beta, \gamma)$ and $(\delta', \alpha', \beta', \gamma')$ such that the initial two messages, (δ, α) and (δ', α'), are the same but the challenges β and β' are different, there is a deterministic procedure to extract the witness from the two transcripts and runs in polynomial time. Special-sound proofs with witness-indistinguishability (WISSP) for languages in $\mathcal{NP}$ can be based on one-way functions.*

Definition 2.17 (ZAP [GS08]). *ZAPs are two round public coin witness indistinguishable proofs introduced by Dwork and Naor [DN00]. ZAPs further have the special property that the first message (sent by the prover) can be reused for multiple proofs. As noted in [BGGL01], any ZAP system already has the property of resettable soundness. Furthermore, resettable witness indistinguishability property can be obtained by applying the transformation in [CGGM00]. We refer to the resulting system as an rZAP system having the property of resettable soundness as well as resettable witness indistinguishability.*

2.1 $\mathcal{P}$-certificate with Delegatable CRS Generation

For $c \in \mathbb{N}$, let $L_c = \{(M, x, y) : M(x) = y \text{ within } |x|^c \text{ steps}\}$. Let $T_M(x)$ denote the number of steps made by M on input x.

Definition 2.18 ($\mathcal{P}$-certificate system [CLP15]). *A tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{P}_{\mathsf{cert}}, \mathsf{V}_{\mathsf{cert}})$ *is a* $\mathcal{P}$-certificate system in the CRS model *if there exist polynomials l_{CRS} and l_π such that for $c, \lambda \in \mathbb{N}$ and $q = (M, x, y) \in L_c$*

- *CRS Generation: $CRS \leftarrow \mathsf{Gen}(1^\lambda, c)$, where Gen runs in time $\mathtt{poly}(\lambda)$. The length of CRS is bounded by $l_{CRS}(\lambda)$.*
- *Proof Generation: $\pi \leftarrow \mathsf{P}_{\mathsf{cert}}(1^\lambda, c, CRS, q)$, where $\mathsf{P}_{\mathsf{cert}}$ runs in time $\mathtt{poly}(\lambda, |x|, T_M(x))$ with $T_M(x) \leq |x|^c$. The length of π is bounded by $l_\pi(\lambda)$.*
- *Proof Verification: $b = \mathsf{V}_{\mathsf{cert}}(1^\lambda, c, CRS, q, \pi)$, where $\mathsf{V}_{\mathsf{cert}}$ runs in time $\mathtt{poly}(\lambda, |q|)$.*

Completeness: *For every $c, d, \lambda \in \mathbb{N}$ and $q = (M, x, y) \in L_c$ such that $|q| \leq \lambda^d$,*

$$\Pr[CRS \leftarrow \mathsf{Gen}(1^\lambda, c), \pi \leftarrow \mathsf{P}_{\mathsf{cert}}(1^\lambda, c, CRS, q) : \mathsf{V}_{\mathsf{cert}}(1^\lambda, c, CRS, q, \pi) = 1] = 1.$$

Strong soundness: *There exists a super-polynomial function* $T(\lambda) = \lambda^{\omega(1)}$ *and a super-constant function* $C(\lambda) = \omega(1)$ *such that for every probabilistic algorithm* P^* *with running time bounded by* $T(\lambda)$*, there exists a negligible function* ν *such that for every* $\lambda \in \mathbb{N}$ *and* $c \leq C(\lambda)$*,*

$$\Pr\left[\begin{array}{l}(q, st) \leftarrow P^*(1^\lambda, c),\\ CRS \leftarrow \mathsf{Gen}(1^\lambda, c),\\ \pi \leftarrow P^*(st, CRS)\end{array} : \mathsf{V}_{\mathsf{cert}}(1^\lambda, c, CRS, q, \pi) = 1 \wedge q \notin L_c\right] \leq \nu(\lambda).$$

A $\mathcal{P}$*-certificate system is* two-message *if the generation of the CRS* Gen *also depends on the statement* q*, i.e.* $CRS \leftarrow \mathsf{Gen}(1^\lambda, c, q)$*. The two-message* $\mathcal{P}$*-certificate system can be considered an interactive protocol as follows: the prover sends* q *to the verifier; the verifier replies with* $CRS \leftarrow \mathsf{Gen}(1^\lambda, c, q)$*; the prover sends* $\pi \leftarrow \mathsf{P}_{\mathsf{cert}}(1^\lambda, c, CRS, q)$*; the verifier accepts if* $\mathsf{V}_{\mathsf{cert}}(1^\lambda, c, CRS, q, \pi) = 1$*.*

A two-message $\mathcal{P}$*-certificate system has a* simple verification procedure *if the verification algorithm* $\mathsf{V}_{\mathsf{cert}}$ *only depends on the security parameter* 1^λ*, the CRS and the proof* π*, i.e. it is independent of the statement* q *and the language index* c*. In this case, we denote the verification by* $\mathsf{V}_{\mathsf{cert}}(1^\lambda, CRS, \pi)$*.*

A $\mathcal{P}$*-certificate system is* unique *if for every* $\lambda, c \in \mathbb{N}$*,* $CRS, q \in \{0,1\}^*$*, there exists at most one* $\pi \in \{0,1\}^*$ *such that* $\mathsf{V}_{\mathsf{cert}}(1^\lambda, c, CRS, q, \pi) = 1$*.*

Note that the uniqueness of a $\mathcal{P}$-certificate holds even against invalid CRS.

Definition 2.19 (delegatable CRS generation [CLP15])**.** *A two-message* $\mathcal{P}$*-certificate* $(\mathsf{Gen}, \mathsf{P}_{\mathsf{cert}}, \mathsf{V}_{\mathsf{cert}})$ *has* delegatable CRS generation *if* Gen *consists of three subroutines:* SetUp*,* PreGen *and* CRSGen*, and there exist polynomials* l_d *and* l_{CRS} *satisfying the following properties:*

- *Parameters Generation:* $(PP, K) \leftarrow \mathsf{SetUp}(1^\lambda, c)$*, where* SetUp *is probabilistic and runs in time* $\mathtt{poly}(\lambda)$*.* PP *is a public parameter and* K *is a secret parameter.*
- *Statement Processing:* $d = \mathsf{PreGen}(PP, q)$*, where* PreGen *is deterministic and runs in time* $\mathtt{poly}(\lambda, |q|)$ *and the length of* d *is bounded by* $l_d(\lambda)$ *independent of* $|q|$*.*
- *CRS Generation:* $\kappa \leftarrow \mathsf{CRSGen}(PP, K, d)$*, where* CRSGen *is probabilistic and runs in time* $\mathtt{poly}(\lambda)$ *and the length of* κ *is bounded by* $l_{CRS}(\lambda)$*.*

Gen *outputs* $CRS = (PP, \kappa)$*.*

Theorem 2.20 ([CLP15])**.** *Assuming the existence of an indistinguishability obfuscation for* $\mathcal{P}/poly$ *and an injective one-way function (that are super-polynomially secure), there exists a (super-polynomially secure) two-message* $\mathcal{P}$*-certificate system with (strong) soundness, uniqueness and delegatable CRS generation.*

3 Constant-Round Resettable Zero Knowledge

In [CLP15], Chung et al. construct a constant-round concurrent ZK argument assuming the existence of families of collision-resistant hash functions, one-way permutations, and indistinguishability obfuscators for $\mathcal{P}/\mathtt{poly}$ (with slightly super-polynomial security). We present it here as follows:

Let com be a non-interactive perfectly binding computationally hiding commitment scheme. As mentioned in [CLP15], the protocol can be modified to work with a 2-message statistically binding commitment scheme based on one-way functions [Nao91, HILL99]. Let $\{\mathcal{H}_n\}_{n\in\mathbb{N}}$ be a family of collision-resistant hash functions. Let $(\mathsf{Gen}, \mathsf{P}_{\mathsf{cert}}, \mathsf{V}_{\mathsf{cert}})$ be a two-message $\mathcal{P}$-certificate system with strong soundness, uniqueness and delegatable CRS generation where Gen consists of subroutines $(\mathsf{SetUp}, \mathsf{PreGen}, \mathsf{CRSGen})$. Let $D = D(n)$ be a super-constant function such that $D(n) \leq C(n)$ for $C(\cdot)$ in Definition 2.18. Let (P_{UA}, V_{UA}) be a constant-round public-coin WIUA. Let (P_{SS}, V_{SS}) be a constant-round public-coin WISSP. Let (P_{ZK}, V_{ZK}) be a constant-round ZK argument.

Let $\Pi_{n,c_3,PP,K,\rho_{\mathsf{CRSGen}}}$ and $\Pi'_{n,c_3,\kappa}$ be programs defined as follows:

$\Pi_{n,c_3,PP,K,\rho_{\mathsf{CRSGen}}}$: on input (d, ρ)

1. If $c_3 \neq \mathsf{com}(d; \rho)$, output $\perp$.
2. Output $\mathsf{CRSGen}(PP, K, d; \rho_{\mathsf{CRSGen}})$.

$\Pi'_{n,c_3,\kappa}$: on input (d, ρ)

1. If $c_3 \neq \mathsf{com}(d; \rho)$, output $\perp$.
2. Output κ.

Let $\mathcal{O}^n_{\mathsf{V_{cert}}}$ be a (deterministic) $\mathcal{P}$-certificate oracle which, on input CRS, outputs a (unique) π such that $\mathsf{V_{cert}}(1^n, CRS, \pi) = 1$.

Let Emu_n be a deterministic polynomial-time machine which, on input (S, y, σ), emulates the execution of the deterministic oracle machine S on input y with access to the oracle $\mathcal{O}^n_{\mathsf{V_{cert}}}$. Emu_n simulates $\mathcal{O}^n_{\mathsf{V_{cert}}}$ by, on input CRS_i in the ith call from S, checking if π_i in $\sigma = (\pi_1, \pi_2, \ldots)$ satisfies $\mathsf{V_{cert}}(1^n, CRS_i, \pi) = 1$. If so, it returns π_i to S, and halts otherwise.

Constant-Round Concurrent Zero-Knowledge Argument Γ [CLP15]

The prover P and the verifier V on common input 1^n and x, and private input w for P:

1. V sends $h \leftarrow \mathcal{H}_n$ to P.
2. P sends $c_1 = \mathsf{com}(0; \rho_1)$ to V.
3. V sends $r \leftarrow \{0,1\}^{4n}$ to P.
4. P sends $c_2 = \mathsf{com}(0; \rho_2)$ to V.
5. P and V run (P_{UA}, V_{UA}) for the following statement: either $x \in L$ or there exists S, $j \in [m]$, $s \in \{0,1\}^n$, σ, ρ_1, ρ_2 such that
   - $c_1 = \mathsf{com}(h(S); \rho_1)$ and
   - $c_2 = \mathsf{com}(h(q); \rho_2)$ where $q = (\mathsf{Emu}_n, (S, (1^n, j, s), \sigma), r)$.

   V rejects if V_{UA} rejects.
6. V runs $(PP, K) \leftarrow \mathsf{SetUp}(1^n, D)$ and sends PP to P.
7. P sends $c_3 = \mathsf{com}(0; \rho_3)$ to V.
8. P and V run (P_{UA}, V_{UA}) so that P proves to V that either $x \in L$ or there exists q, ρ_2, ρ_3 such that $c_2 = \mathsf{com}(h(q); \rho_2)$ and $c_3 = \mathsf{com}(d; \rho_3)$ where $d = \mathsf{PreGen}(PP, q)$. V rejects if V_{UA} rejects.
9. V computes $\widehat{\Pi} \leftarrow i\mathcal{O}(\Pi_{n,c_3,PP,K,\rho_{\mathsf{CRSGen}}})$ and sends $\widehat{\Pi}$ to P.
10. V and P run (P_{ZK}, V_{ZK}) so that V proves to P that there exist K, ρ_{SetUp}, ρ_{CRSGen}, $\rho_{i\mathcal{O}}$ such that
    - $(PP, K) = \mathsf{SetUp}(1^n, D; \rho_{\mathsf{SetUp}})$ and
    - $\widehat{\Pi} = i\mathcal{O}(\Pi_{n,c_3,PP,K,\rho_{\mathsf{CRSGen}}}; \rho_{i\mathcal{O}})$.

    P aborts if V_{ZK} rejects.
11. P sends $c_4 = com(0; \rho_4)$ to V.
12. P and V run (P_{SS}, V_{SS}) so that P proves to V that either $x \in L$ or there exists d, ρ_3, ρ_4 such that $c_3 = com(d; \rho_3)$ and $c_4 = \mathsf{com}(CRS; \rho_4)$ where $CRS = (PP, \widehat{\Pi}(d, \rho_3))$. V rejects if V_{SS} rejects.
13. P and V run (P_{SS}, V_{SS}) so that P proves to V that either $x \in L$ or there exists CRS, ρ_4 and P-certificate π such that $c_4 = \mathsf{com}(CRS; \rho_4)$ and $\mathsf{V_{cert}}(CRS, \pi) = accept$. V accepts if V_{SS} accepts. Otherwise, V rejects.

Theorem 3.1 ([CLP15]). *Assuming the existence of families of collision-resistant hash functions, one-way permutations, and indistinguishability obfuscators for* P/*poly that are super-polynomially* *secure, there exists a constant-round concurrent zero-knowledge argument for $\mathcal{NP}$.*

3.1 From Concurrent ZK to Resettable ZK

Let $\Gamma = (P_\Gamma, V_\Gamma)$ be an ℓ-round concurrent ZK argument. We construct a $\mathcal{O}(\ell)$-round resettable ZK argument Λ as follows:

Let com be a non-interactive perfectly binding computationally hiding commitment scheme. Let (P_{rsZK}, V_{rsZK}) be a constant-round resettably-sound ZK argument with the simulator Sim_{rsZK}.

Constant-Round Resettable Zero-Knowledge Argument Λ

The prover P and the verifier V on common input 1^n and x, and private input w for P:

1. V sending $m_0 = (\mathsf{com}(r_1), \ldots, \mathsf{com}(r_\ell))$ to P.
2. P chooses a random seed s for a pseudorandom function $f_s : \{0,1\}^* \to \{0,1\}^{l(n)}$ where $l(n)$ is the upper bound on the size of random bits P_Γ needs in each round of Γ.
3. P and V run Γ with the following modifications:
   - For each message m_i that V_Γ sends in the ith round of Γ, V and P run (P_{rsZK}, V_{rsZK}) so that V proves to P that m_i is computed using random bits r_i committed in m_0 in the first round.
   - For each message m'_i that P_Γ sends in the ith round of Γ, P applies f_s to the transcript so far and uses the output as random bits to compute m'_i.

3.2 Proofs

Lemma 3.2. Λ *is a resettable ZK argument system.*

Proof. First, we consider the protocol Λ_F where we replace a pseudorandom function f_s by a truly random function $F : \{0,1\}^* \to \{0,1\}^{l(n)}$. We argue that Λ_F is indistinguishable from Λ by the reduction to the security of pseudorandom function as follows. We construct an adversary $\mathcal{A}_{PRF}$ having access to an oracle computing either f_s or F such that $\mathcal{A}_{PRF}$ runs Λ (or Λ_F) with the following modification: for each message m'_i sent by an honest P, $\mathcal{A}_{PRF}$ asks the oracle using the transcript of the protocol up to that point as input; it then uses the oracle output as the random bits to compute m'_i. Finally, $\mathcal{A}_{PRF}$ runs and outputs the output of the distinguisher on the view of the protocol. Since $\mathcal{A}_{PRF}$ runs the honest P from the beginning to the end, it has access to private parameters of P, and thus is able to finish the protocol. Thus, any non-uniform polynomial-size verifiers must behave in the same way except with negligible probability.

Let V^*_{RES} be a resetting verifier in Λ_F. We construct a concurrent verifier V^*_{CONC} such that for any P_{CONC} there exists P_{RES} such that $\{\mathtt{view}^{P_{RES}}_{V^*_{RES}}\}$ and $\{\mathtt{view}^{P_{CONC}}_{V^*_{CONC}}\}$ are computationally indistinguishable as follows: V^*_{CONC} runs V^*_{RES} internally and delivers messages between V^*_{RES} and P_{CONC} while recording the first message (commitments) of V^*_{RES} and every message of P_{CONC}. Whenever V^*_{RES} resets P_{RES} and sends the first message, V^*_{CONC} checks if it has been sent before. If so, V^*_{CONC} resends the appropriate responses or continues the session if necessary. Otherwise, V^*_{CONC} starts a new session of P_{CONC}. The randomness used in this new session is indistinguishable from the randomness P_{RES} used by applying F to the new transcript (as m_0 is different).

Claim. *For a fixed seed s and m_0, for each $i \in [\ell]$, V_r^* cannot find two different messages m_i, m_i' in the ith round such that it can make P_{RES} accepting the ith resettably-sound ZK argument except with negligible probability.*

Proof. Let the first round message $m_0 = (c_1, \ldots, c_\ell)$. Assume for contradiction that there exists $i \in [\ell]$ such that V_r^* can find $m_i \neq m_i'$ and the corresponding resettably-sound ZK argument that P_{RES} accepts with non-negligible probability. In such case, by the resettable soundness of the ZK argument, m_i and m_i' are both computed correctly with respect to the protocol Λ_F using the randomness committed in c_i. In other words, there exists a deterministic polynomial-time function μ_i such that m_i and m_i' have the form $m_i = \mu_i(r_i)$ with $c_i = \mathsf{com}(r_i)$ and $m_i' = \mu_i(r_i')$ with $c_i = \mathsf{com}(r_i')$, for some $r_i \neq r_i'$. However, this implies $\mathsf{com}(r_i) = \mathsf{com}(r_i')$, which contradicts the perfectly binding of com. □

Thus, the transcript of the whole session depends only on s and m_0. Therefore, $\{\mathsf{view}_{V^*_{RES}}^{P_{RES}}\}$ and $\{\mathsf{view}_{V^*_{CONC}}^{P_{CONC}}\}$ are computationally indistinguishable. □

Lemma 3.3. Λ *is sound.*

Proof. Suppose there exists a cheating prover P^*_{RES} that can prove a false theorem $x \notin L$ with non-negligible probability. Consider the following hybrid experiments:

Exp_0: Run $\langle P^*_{RES}, V_{RES}\rangle(1^n, x)$.

Let $\text{Exp}_{1,0}$ be the same as Exp_0, and for $i = 1, \ldots, \ell$,

$\text{Exp}_{1,i}$: Similar to $\text{Exp}_{1,i-1}$ except that the execution of $P_{rsZK}(r_i)$ following the message m_i is replaced by the execution of $\mathsf{Sim}_{rsZK}^{P^*_{RES,i}}$ where $P^*_{RES,i}$ is the residual rsZK verifier (note that P^*_{RES} runs V_{rsZK}) who has received $m_0, \ldots, m_i$ as inputs. Assume for contradiction that there exists a distinguisher D for $\text{Exp}_{1,i}$ and $\text{Exp}_{1,i-1}$. We construct a distinguisher D' for the (standard) zero-knowledge property of (P_{rsZK}, V_{rsZK}) as follows. First, we generate $r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_\ell$ uniformly and let $\tilde{c}_i = \mathsf{com}(0)$. Then we produce the transcript for P^*_{RES} as in Λ except that we use $\tilde{c}_i$ instead of $c_i = \mathsf{com}(r_i)$. By the computational hiding of com, P^*_{RES} cannot distinguish $\tilde{c}_i$ from c_i. Given either $\{\mathtt{view}_{V_{rsZK}}^{P_{rsZK}}\}$ where V_{rsZK} is run by $P^*_{RES,i}$ or $\mathsf{Sim}_{rsZK}^{P^*_{RES,i}}$, we generate the rest of the transcript for protocol Λ using r_j generated earlier. Finally, D' runs D on the entire transcript. In either case, the transcript is computationally indistinguishable to either $\text{Exp}_{1,i}$ or $\text{Exp}_{1,i-1}$. Thus, D' can break the zero-knowledge property of (P_{rsZK}, V_{rsZK}), which is a contradiction. Hence, $\text{Exp}_{1,i}$ and $\text{Exp}_{1,i-1}$ are indistinguishable.

Let $\text{Exp}_{2,0}$ be the same as $\text{Exp}_{1,\ell}$, and for $i = 1, \ldots, \ell$,

$\text{Exp}_{2,i}$: Similar to $\text{Exp}_{2,i-1}$ except that $\mathsf{com}(r_i)$ in the first message m_0 is replaced by $\mathsf{com}(0)$. Consider the following reduction to the computational hiding property of com: $\mathcal{A}_{\mathsf{com}}$ sends r_i and 0 to S_{com}; it passes the commitment from S_{com} as the ith commitment in m_0 of $\text{Exp}_{2,i-1}$ (or $\text{Exp}_{2,i}$); $\mathcal{A}_{\mathsf{com}}$ can complete the experiment as it does not need to know which message it commits using Sim_{rsZK}; $\mathcal{A}_{\mathsf{com}}$ outputs the output of the experiment. The computational hiding property implies that $\text{Exp}_{2,i}$ and $\text{Exp}_{2,i-1}$ are indistinguishable.

Now we construct a cheating prover P^*_{CONC} for Γ by running $\text{Exp}_{2,\ell}$ internally as follows: P^*_{CONC} sends $\mathsf{com}(0)$ to P^*_{RES}; P^*_c passes every messages from P^*_{RES} to V_{CONC}; P^*_{CONC} passes every message from V_{CONC} to P^*_{RES} then runs Sim_{rsZK} while P^*_{RES} runs V_{rsZK}. Thus, P^*_{CONC} can prove a false theorem $x \notin L$ with non-negligible probability, which contradicts the soundness of Γ. □

Theorem 3.4. *Assuming one-way functions, there exists a compiler transforming an ℓ-round concurrent zero-knowledge argument to a $\mathcal{O}(\ell)$-round resettable zero-knowledge argument.*

Proof. The resettable zero knowledge and soundness are proved in Lemma 3.2 and Lemma 3.3, respectively. The completeness follows from the completeness of Γ by inspection. For each round of Γ, P and V has to run additional $\mathcal{O}(1)$ rounds for resettably-sound ZK protocol that V uses the committed random bits, and 1 extra round in the beginning. Thus, the round complexity is $\mathcal{O}(\ell)$. □

Corollary 3.5. *Assuming the existence of families of collision-resistant hash functions, one-way permutations, and indistinguishability obfuscators for $\mathcal{P}/\mathtt{poly}$ that are super-polynomially secure, there exists a constant-round resettable zero-knowledge argument for $\mathcal{NP}$.*

Proof. We instantiate Λ by letting Γ be the constant-round concurrent zero-knowledge argument system of [CLP15]. Perfectly binding com can be constructed from one-way permutations. A constant-round resettably-sound ZK argument can be constructed from one-way functions [COP+14]. □

4 Concurrent ZK with Resettable Soundness

In this section, we construct a constant-round resettably-sound concurrent ZK argument based on the constant-round cZK argument in [CLP15]. We make use of our constant-round rZK argument from the previous section (Corollary 3.5), the technique used in [BGGL01] to add resettable soundness to a public-coin protocol, and our new techniques to deal with non-public coin nature of the cZK protocol in [CLP15].

4.1 Construction

Let Γ be the constant-round concurrent ZK argument from [CLP15] described in Section 3. We construct a constant-round concurrent ZK argument with resettable soundness Δ as follows:

Let (P_{rZK}, V_{rZK}) be a constant-round resettable ZK argument with the simulator Sim_{rZK}. The verifier V chooses a random seed s for a pseudorandom function $f_s : \{0,1\}^* \to \{0,1\}^{l(n)}$, where $l(n)$ is the upper bound on the size of random bits V need in each round of Γ. Then P and V run Γ with the following modifications. In Step 10, instead of running a ZK argument (P_{ZK}, V_{ZK}), V and P run the resettable ZK argument (P_{rZK}, V_{rZK}). Additionally, for each message m that V sends in Γ, V uses the output of f_s applying to the transcript from the protocol up to this point as random bits to compute m.

4.2 Proofs

Before we prove that the protocol above is a concurrent ZK argument with resettable soundness, we consider another modification, Γ', of the protocol Γ in [CLP15]. First, P and V repeat Step 7-9 for t times with V using the same ρ_{CRSGen} for some $t = \mathtt{poly}(n)$. Let Step $7j - 9j$ denoted jth repeat of Step 7-9. Secondly, we remove the zero-knowledge proof in Step 10, and replace it with "P chooses $i \in [t]$ and sends i to V", and then P and V follows the rest of the protocol ignoring Step $7j - 9j$ for $j \neq i$.

Lemma 4.1. *Γ' is a sound interactive argument.*

Proof. We strictly follow the proof of soundness of Γ in [CLP15] with a modification necessary for the repetition of Step 7-9. Assume for contradiction that there is a non-uniform deterministic polynomial-time prover P^* and a positive polynomial p such that for infinitely many $n \in \mathbb{N}$, P^*

can convince V to accept $x \notin L$ with non-negligible probability $1/p(n)$. Let E be the global proof-of-knowledge extractor of the WIUA (P_{UA}, V_{UA}), and E' be the knowledge extractor of the WISSP (P_{SS}, V_{SS}). We define the experiment Exp which runs $\langle P^*, V\rangle(1^n, x)$ with the following addition:

- In Step 5, let $P^*_{\mathsf{prefix}_1}$ be the residual WIUA prover who has received $\mathsf{prefix}_1 = (h, r)$ in Step 1 and 3. Run $w_1 \leftarrow E_{s_1}^{P^*_{\mathsf{prefix}_1}}$, where s_1 is uniform randomness. If E fails, halt and output $\perp$.
- In Step 7j, for $j = 1, \ldots, t$, let $P^*_{\mathsf{prefix}_{2,j}}$ be the residual WIUA prover who has received $\mathsf{prefix}_{2,j}$ consisting of h, r, WIUA messages, PP and $\widehat{\Pi}_k$ in Step 1, 3, 5, 6, $8k$ and $9k$ for $k = 1, \ldots, j-1$. Run $w_{2,j} \leftarrow E_{s_{2,j}}^{P^*_{\mathsf{prefix}_{2,j}}}$, where $s_{2,j}$ is uniform randomness. If E fails, halt and output $\perp$.
- In Step 12, let $P^*_{\mathsf{prefix}_3}$ be the residual WISSP prover who has received prefix_3 consisting of h, r, WIUA messages, PP and $\widehat{\Pi}_j$ in Step 1, 3, 5, 6, $8j$ and $9j$ for $j = 1, \ldots, t$. Run $w_3 \leftarrow E'^{P^*_{\mathsf{prefix}_3}}_{s_3}$, where s_3 is uniform randomness. If E' fails, halt and output $\perp$.
- In Step 13, let $P^*_{\mathsf{prefix}_4}$ be the residual WISSP prover who has received prefix_4 consisting of prefix_3 and WISSP messages in Step 12. Run $w_4 \leftarrow E'^{P^*_{\mathsf{prefix}_4}}_{s_4}$, where s_4 is uniform randomness. If E' fails, halt and output $\perp$.
- If V rejects, output $\perp$. Otherwise,
  - Parse $w_1 = (S, j, s, \sigma, \rho_1, \rho_2)$. If w_1 does not have this form, output $\perp$.
  - Let $q = (\mathsf{Emu}_n, (S, (1^n, j, s), \sigma), r)$. For $j = 1, \ldots, t$, if $w_{2,j} \neq (q, \rho_{2,j}, \rho_{3,j})$ for some $\rho_{2,j}, \rho_{3,j}$, output $\perp$.
  - Let $d = \mathsf{PreGen}(PP, q)$. If $w_3 \neq (d, \rho_{3,i}, \rho_4)$ for some ρ_4 where $i \in [t]$ is chosen by P^* in Step 10, output $\perp$.
  - Let $CRS = (PP, \widehat{\Pi}(d, \rho_{3,i}))$. If $w_4 \neq (CRS, \rho_4, \pi)$ for some π, output $\perp$.
- output (S, q, r).

By the weak proof-of-knowledge property of WIUA and special soundness of WISSP, when P^* convinces V to accept $x \notin L$, the extractors E and E' succeed in extracting the witnesses described above (instead of the actual witness of the theorem) with non-negligible probability $1/p'(n)$. By perfectly binding property of com and collision-resistance of $\mathcal{H}$, the consistency check in the last step will pass except with negligible probability $\nu(n)$. In this case, except with negligible probability, $c_{3,j}$ sent in Step $7j$ is $\mathsf{com}(d; \rho_{3,j})$ for the same $d = \mathsf{PreGen}(PP, q)$ for all $j = 1, \ldots, t$. Otherwise, we can construct a cheating WIUA prover that commits to $c' = \mathsf{com}(d'; \rho')$ with $d' \neq \mathsf{PreGen}(PP, q)$ with non-negligible probability by randomly pick $j \in [t]$ and commit to $c' = c_{3,j}$. This breaks the soundness of WIUA. So, the only output of $\widehat{\Pi}_j$ is $\mathsf{CRSGen}(PP, K, d, \rho_{\mathsf{CRSGen}}) = \kappa$ for all $j = 1, \ldots, m$ except with negligible probability $\nu'(n)$. Thus, the probability that Exp does not output $\perp$ and every $\widehat{\Pi}_j$ output the same κ is $1/p'(n) - \nu(n) - \nu'(n)$ which is non-negligible. We call this event Good.

Now consider a series of experiments Exp'_j for $j \in [t]$ defined as follows: $\mathsf{Exp}'_0 = \mathsf{Exp}$, and Exp'_j differs from Exp'_{j-1} in Step $9j$ where we replace $\widehat{\Pi}_j \leftarrow i\mathcal{O}(\Pi_{n,c_{3,j},PP,K,\rho_{\mathsf{CRSGen}}})$ with $\widehat{\Pi}'_j \leftarrow i\mathcal{O}(\Pi'_{n,c_{3,j},\kappa})$ where $\kappa = \mathsf{CRSGen}(PP, K, d; \rho_{\mathsf{CRSGen}})$. When Good occurs, by perfectly binding property of com, $\Pi'_{n,c_{3,j},\kappa}$ and $\Pi_{n,c_{3,j},PP,K,\rho_{\mathsf{CRSGen}}}$ are functionally equivalent except with negligible probability. In this case, Exp'_{j-1} and Exp'_j are indistinguishable by the reduction to $i\mathcal{O}$ as follows: $\mathcal{D}_{i\mathcal{O}}$ runs Exp'_{j-1}

(or Exp'_j) up to Step $8j$ and outputs $\Pi'_{n,c_{3,j},\kappa}$ and $\Pi_{n,c_{3,j},PP,K,\rho_{\mathsf{CRSGen}}}$ and the state of the experiment z; up to receiving obfuscated program $\widehat{\Pi}$ and z, $\mathcal{A}_{i\mathcal{O}}$ sends $\widehat{\Pi}$ to P^*, continues the experiment until the end, and outputs the output of the experiment. Thus, Exp'_{j-1} and Exp'_j are indistinguishable by the security of $i\mathcal{O}$. Hence, by hybrid argument, the probability of Good event is non-negligible in Exp'_j for $j = 1, \ldots, t$. Let $\mathsf{Exp}' = \mathsf{Exp}'_t$.

Now suppose that Good and q is false occurs with non-negligible probability. Then we construct $P^*_{\mathcal{P}\mathsf{cert}}$ that breaks the strong soundness of the $\mathcal{P}$-certificate system as follows: $P^*_{\mathcal{P}\mathsf{cert}}$ runs Exp' up to Step 5 where it extracts q from w_1. Up on receiving $CRS = (PP, \kappa)$ where $(PP, K) \leftarrow \mathsf{SetUp}(1^n, D)$ and $\kappa \leftarrow \mathsf{CRSGen}(PP, K, \mathsf{PreGen}(PP, q))$, it continues Exp' using PP and κ and output π extracted from w_4. If Good occurs, by the soundness of WISSP, $P^*_{\mathcal{P}\mathsf{cert}}$ succeeds and $\mathsf{V}_{\mathsf{cert}}(CRS, \pi) = 1$ except with negligible probability. Thus, $P^*_{\mathcal{P}\mathsf{cert}}$ contradicts the strong soundness of the $\mathcal{P}$-certificate system. Hence, Good and q is true occurs with non-negligible probability. We call this event Good'. By averaging argument, there exists h such that $\mathsf{Good}'|h$ occurs with non-negligible probability.

Finally, consider Exp'' where Exp' is run twice with this h but with the second execution replacing r in Step 3 by an independent random string r'. With non-negligible probability, both executions succeed and output (S, q, r) and (S', q', r'). Since c_1 must be the same in both executions, $S = S'$ except with negligible probability by perfectly binding property of com and collision-resistance of $\mathcal{H}$. Since $q = (\mathsf{Emu}_n, (S, (1^n, j, s), \sigma), r)$ and $q' = (\mathsf{Emu}_n, (S, (1^n, j', s'), \sigma'), r')$ are true, we have $S^{\mathcal{O}^n_{\mathsf{V}_{\mathsf{cert}}}}(1^n, j, s) = r$ and $S^{\mathcal{O}^n_{\mathsf{V}_{\mathsf{cert}}}}(1^n, j', s') = r'$. We have that $|(1^n, j, s)| < 3n < 4n = |r|$ and $|(1^n, j', s')| < |r'|$. However, the deterministic machine $S^{\mathcal{O}^n_{\mathsf{V}_{\mathsf{cert}}}}$ predicts independent r and r' with non-negligible probability. This is information theoretically impossible as there are at most 2^{3n} possible outputs for $S^{\mathcal{O}^n_{\mathsf{V}_{\mathsf{cert}}}}$. Thus, we reach a contradiction.

As in the proof of soundness of Γ in [CLP15], the WIUA global proof-of-knowledge extractor E runs in super-polynomial time as a part of the witness q is of super-polynomial size. Thus, the collision-resistant hash functions $\mathcal{H}$, the commitment scheme com and indistinguishability obfuscators $i\mathcal{O}$ need to be super-polynomially secure. □

Now we can prove the main theorem of this section.

Theorem 4.2. Δ *is a concurrent ZK argument with resettable soundness.*

Proof. Since the rZK argument (P_{rZK}, V_{rZK}) is also a ZK argument and we only further modify an honest verifier V, the concurrent zero-knowledge of Δ follows directly from the concurrent zero-knowledge property of Γ. Now we consider the protocol Δ_F where we replace a pseudorandom function f_s by a truly random function $F : \{0,1\}^* \rightarrow \{0,1\}^{l(n)}$. We argue that Δ_F is indistinguishable from Δ by the reduction to the security of pseudorandom function as follows. Fix $x \notin L$ and P^*_{RES} that convinces a resettable verifier V_{RES} to accept $x \notin L$ with probability ϵ through protocol Δ_F. We construct an adversary $\mathcal{A}_{PRF}$ having access to an oracle computing either f_s or F such that $\mathcal{A}_{PRF}$ runs Δ (or Δ_F) with the following modification: for each message m sent by an honest V_{RES}, $\mathcal{A}_{PRF}$ asks the oracle using the transcript of the protocol up to that point as input; it then uses the oracle output as the random bits to compute m. $\mathcal{A}_{PRF}$ outputs the output of V. Since $\mathcal{A}_{PRF}$ runs the honest V_{RES} from the beginning to the end, it has access to private parameter K that V generates in Step 6, and thus is able to compute the obfuscated program and rZK messages in Step 9 and 10. Thus, any non-uniform polynomial-size provers must behave in the same way except with negligible probability. Hence, the completeness follows from the completeness of Γ.

We now show the resettable soundness of the protocol. Assume for contradiction that there is a non-uniform polynomial-time resetting prover P^*_{RES} that convinces a resettable verifier V_{RES} to accept $x \notin L$ with probability ϵ through protocol Δ_F. We construct a polynomial-time (standard) prover P^*_S, emulating P^*_{RES}, that convinces a (standard) verifier V_S to accept the same $x \notin L$

through protocol Γ' repeating Step 7-9 for t times, where $t = \texttt{poly}(n)$ is the total number of messages sent by P^*_{RES}. Let c be the number of (prover) rounds in Δ.

The cheating prover P^*_S proceeds as follows. First it uniformly selects $i_1, \ldots, i_c \in \{1, \ldots, t\}$. It invokes P^*_{RES} while emulating V_{RES}. In the jth round of Δ_F, P^*_S answers a message from P^*_{RES} according to the following cases:

- If the prefix of the current session transcript is identical to a corresponding prefix of a previous session, then P^*_{CONC} answers by using the same answer it has given in the previous session.
- Otherwise, P^*_S either forwards the message to V_S and then forwards the reply it receives, or generates the reply itself according to the following conditions:
  - If the message is c_3 or WIUA in Step $7j - 8j$, P^*_S repeats its decision whether to forward the message in Step 6. In other words, if P^*_S forwards the message in Step 6, it will forward this message. If it generates the reply in Step 6 itself, it will generate the reply for this message as well. This is because it can only generate an answer in Step $9i$ if it has generated the answer in Step 6 of the same transcript (instead of passing to V_S).
  - If the message is $i \in [t]$ in Step 10, P^*_S does not forward the message, but instead runs the simulator Sim_{rZK} with P^*_{RES} corresponding to obfuscated program in Step $9i$.
  - If the index of the current message from P^*_{RES} does not equal to i_j selected previously, P^*_S generates a reply message using a uniformly selected random bits.
  - Otherwise, P^*_S forwards the current message to V_S and sends P^*_{RES} a reply it receives from V_S.

  In each case, P^*_{CONC} records the messages from both sides for later use.

By the resettable zero-knowledge of (P_{rZK}, V_{rZK}), the probability of P^*_{RES} proving a false theorem $x \notin L$ only changes negligibly by running Sim_{rZK} instead of P_{rZK}. By the property of truly random function, the view of P^*_{RES} is identical to the distribution that P^*_{RES} sees when interacting with an honest V_{RES}. If the chosen $i_1, \ldots, i_c$ equal the indices of the messages that correspond to the c messages sent in a session in which P^*_{RES} convinces V_{RES} to accept $x \notin L$, then P^*_S will also convince V_S to accept $x \notin L$ by our construction of V_{RES}. Thus, the probability of V_S accepting $x \notin L$ is at least $\epsilon/t^c - \nu(n)$ for some negligible function ν. This probability is non-negligible. Therefore, it contradicts Lemma 4.1. □

Let $\Lambda = (P_{rZK}, V_{rZK})$ be the constant-round resettable ZK protocol obtained in Corollary 3.5, we get the following corollary.

Corollary 4.3. *Assuming the existence of families of collision-resistant hash functions, one-way permutations, and indistinguishability obfuscators for* $\mathsf{P}/poly$ *that are super-polynomially secure, there exists a constant-round resettably-sound concurrent zero-knowledge argument for* $\mathcal{NP}$.

5 Simultaneous Resettable ZK

To obtain our main theorem, we apply a combination of the transformations in Theorem 4 and 5 in Section 6, and Theorem 6 and 7 in Appendix C of [GS08] to our protocol in Section 4 to obtain simultaneous resettability.

More specifically, we combine three transformations in [GS08]:

- from resettably-sound (relaxed) concurrent zero-knowledge argument to hybrid-sound hybrid-resettable zero-knowledge argument;
- from hybrid-sound zero-knowledge argument to resettably-sound zero-knowledge argument while maintaining (hybrid) resettability;
- from hybrid-resettable zero-knowledge argument to resettable zero-knowledge argument while maintaining (hybrid) resettable soundness;

We refer to Section 1 for an informal discussion and [GS08] for formal definitions of relaxed concurrent zero-knowledge, hybrid resettability and hybrid soundness.

Theorem 5.1 (implied from [GS08]). *Assuming the existence of ZAPs (i.e., 2-round resettably-sound resettable witness-indistinguishable proof systems) and family of pseudorandom functions, there exists a transformation from an ℓ-round resettably-sound concurrent zero-knowledge argument to a $\mathcal{O}(\ell)$-round resettably-sound resettable zero-knowledge argument.*

Applying the transformations to the protocol Δ in Corollary 4.3 results in the following theorem. Note that ZAPs can be constructed from $i\mathcal{O}$ and one-way functions [BP15b], which can then be transformed to have resettable soundness and resettable witness indistinguishability. Furthermore, only the first transformation is based on ZAPs while all of them assume pseudorandom functions.

Theorem 5.2. *Assuming the existence of families of collision-resistant hash functions, one-way permutations, and indistinguishability obfuscators for $\mathcal{P}/\texttt{poly}$ that are super-polynomially secure, there exists a constant-round resettably-sound resettable zero-knowledge argument for $\mathcal{NP}$.*

6 Acknowledgments

Research supported in part by "GNCS - INdAM", EU COST Action IC1306, NSF grants 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported in part by DARPA Safeware program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work of the 3rd author has been done in part while visiting UCLA.

References

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS '01*, pages 106–115, 2001.

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

[BG08] Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM Journal on Computing*, 38(5):1661–1694, 2008.

[BGGL01] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. Resettably-sound zero-knowledge and its applications. In *FOCS '02*, pages 116–125, 2001.

[BP13] Nir Bitansky and Omer Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In *STOC '13*, 2013.

[BP15a] Nir Bitansky and Omer Paneth. On non-black-box simulation and the impossibility of approximate obfuscation. *SIAM J. Comput.*, 44(5):1325–1383, 2015.

[BP15b] Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *TCC '15*, pages 401–427. Springer, 2015.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC '00*, pages 235–244, 2000.

[CLP13] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero knowledge from p-certificates. In *FOCS '13*, pages 50–59. IEEE, 2013.

[CLP15] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *CRYPTO '15*, pages 287–307. Springer, 2015.

[COP+14] Kai-Min Chung, Rafail Ostrovsky, Rafael Pass, Muthuramakrishnan Venkitasubramaniam, and Ivan Visconti. 4-round resettably-sound zero knowledge. In *TCC '14*, pages 192–216. Springer, 2014.

[COPV13] Kai-Min Chung, Rafail Ostrovsky, Rafael Pass, and Ivan Visconti. Simultaneous resettability from one-way functions. In *FOCS '13*, pages 60–69. IEEE, 2013.

[COSV12] Chongwon Cho, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Simultaneously resettable arguments of knowledge. In *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

[CPS13] Kai-Min Chung, Rafael Pass, and Karn Seth. Non-black-box simulation from one-way functions and applications to resettable security. In *STOC '13*. ACM, 2013.

[DCPV04] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In *Annual International Cryptology Conference*, pages 237–253. Springer, 2004.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *FOCS '09*, pages 251–260. IEEE, 2009.

[DL07] Yi Deng and Dongdai Lin. Instance-dependent verifiable random functions and their application to simultaneous resettability. In *EUROCRYPT*, pages 148–168, 2007.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *FOCS '00*, pages 283–293. IEEE, 2000.

[DNS98] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *STOC '98*, pages 409–418. ACM, 1998.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC '90*, pages 416–426, 1990.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *STOC '85*, pages 291–304. ACM, 1985.

[Gol01] Oded Goldreich. *Foundations of Cryptography — Basic Tools*. Cambridge University Press, 2001.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for nizk. In *Advances in Cryptology – CRYPTO '06*, volume 4117 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2006.

[GOVW12] Sanjam Garg, Rafail Ostrovsky, Ivan Visconti, and Akshay Wadia. Resettable statistical zero knowledge. In *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 494–511. Springer, 2012.

[GS08] Vipul Goyal and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. *IACR Cryptology ePrint Archive*, 2008:545, 2008.

[GS09] Vipul Goyal and Amit Sahai. Resettably secure computation. In *Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '09, pages 54–71. Springer, 2009.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:12–24, 1999.

[KP01] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in polyloalgorithm rounds. In *STOC '01*, pages 560–569, 2001.

[Lin10] Yehuda Lindell. Foundations of cryptography 89-856. `http://u.cs.biu.ac.il/~lindell/89-856/complete-89-856.pdf`, 2010.

[MR01] Silvio Micali and Leonid Reyzin. Soundness in the public-key model. In *Advances in Cryptology – Crypto '01*, volume 2139 of *Lecture Notes in Computer Science*, pages 542–565. Springer-Verlag, 2001.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[OV12] Rafail Ostrovsky and Ivan Visconti. Simultaneous resettability from collision resistance. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:164, 2012.

[SV12] Alessandra Scafuro and Ivan Visconti. On round-optimal zero knowledge in the bare public-key model. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 153–171. Springer, 2012.

[YZ07] Moti Yung and Yunlei Zhao. Generic and practical resettable zero-knowledge in the bare public-key model. In *EUROCRYPT*, pages 129–147, 2007.