Individual Simulations

Yi Deng[1,2,3(✉)]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
deng@iie.ac.cn
[2] State Key Laboratory of Cryptology, Beijing, China
[3] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing, China

Abstract. We develop an *individual* simulation technique that explicitly makes use of particular properties/structures of a given adversary's functionality. Using this simulation technique, we obtain the following results.

1. We construct the *first* protocols that *break previous black-box barriers* under the standard hardness of factoring, both of which are *polynomial time simulatable* against *all a-priori bounded polynomial size* distinguishers:
   - Two-round selective opening secure commitment scheme.
   - Three-round concurrent zero knowledge and concurrent witness hiding argument for NP in the bare public-key model.
2. We present a simpler two-round weak zero knowledge and witness hiding argument for NP in the plain model under the sub-exponential hardness of factoring. Our technique also yields a significantly simpler proof that existing distinguisher-dependent simulatable zero knowledge protocols are also polynomial time simulatable against all distinguishers of a-priori bounded polynomial size.

The core conceptual idea underlying our individual simulation technique is an observation of the *existence* of *nearly optimal extractors* for all hard distributions: For any NP-instance(s) sampling algorithm, there exists a polynomial-size witness extractor (depending on the sampler's functionality) that almost outperforms any circuit of a-priori bounded polynomial size in terms of the success probability.

1 Introduction

1.1 Background

The simulation paradigm [GMR89] plays a pivotal role in complexity-based cryptography, which takes the reductionist approach to prove the security of a given cryptosystem. In a typical security proof, we devise a reduction algorithm, which invokes as a subroutine the adversary that claims to break the target cryptosystem, to crack the underlying hard problem. In this process, the reduction algorithm needs to simulate the honest parties for the adversary in order to exploit its

S. Moriai and H. Wang (Eds.): ASIACRYPT 2020, LNCS 12493, pp. 805–836, 2020.
https://doi.org/10.1007/978-3-030-64840-4_27

power. For most interactive cryptographic protocols, simulating the adversary's view is actually the essential part of the reduction.

The most commonly used simulation strategy is black-box simulation, which appears very restrictive since the black-box simulator ignores the internal workings of the adversary completely. Indeed, starting from the seminal work of Impagliazzo and Rudich [IR89], a lot of impossibility results regarding black-box simulation were proved in a variety of settings. In the last two decades, several new simulation techniques, notably the PCP-based non-black-box simulation [Bar01] and the recently distinguisher-dependent simulation [JKKR17,BKP19] techniques, were developed to get around certain black-box barriers on the round-complexity of cryptographic protocols. However, for many basic protocols, it still remains unclear whether the known black-box impossibility results on their round-complexity might be overcome using new (non-black-box) reduction/simulation techniques. In this paper, we consider the round-complexity of several related fundamental protocols: selective opening secure commitments and zero knowledge protocols.

Commitment Scheme Secure Under the Selective Opening Attacks. In a selective opening attack against a commitment scheme, the receiver observes many commitments and is allowed to ask the committer to open some of them. Dwork et al. [DNRS03] put foward the notion of selective opening security and asked if we can construct such a commitment that the unopened commitments in the selective opening attack still stay hiding. As showed in [DNRS03], this problem has a deep connection with the existence of 3-round zero knowledge and the soundness of the Fiat-Shamir heuristics.

Bellare et al. [BHY09] constructed the first selective opening secure commitment. The high-level idea of their construction (and the follow-up from [ORSV13] by Ostrovsky et al.) is as follows. The receiver generates a trapdoor for an equivocal trapdoor commitment scheme, and proves of knowledge of the trapdoor via a cut-and-choose type protocol; the committer then uses this trapdoor commitment scheme to commit to a value. In simulation, the simulator first extracts the trapdoor by rewinding the receiver, and then can open a commitment to any value it wishes. So far, the best known construction of (simulation-based notion of) selective opening secure commitment requires three rounds [ORSV13].

There is an obstacle to further reduce the round-complexity of selective opening secure commitment. Note that in a two-round scheme[1] the receiver sends only one message and the standard black-box simulator/extractor that treats the (possibly malicious) receiver as a black-box would fail. Indeed, Xiao [Xia11,Xia13] proved that it is impossible to achieve selective opening security in 2 rounds with a black-box simulator.

[1] The round-complexity of a commitment scheme refers to the one of its committing phase. In this paper we focus on commitment schemes with a non-interactive opening phase.

Zero Knowledge Protocols in Two and Three Rounds. Early constructions of zero knowledge proofs (with statistical soundness) [GMR89] and arguments (with computational soundness) [BCC88] are quite simple and round-efficient: only three messages are exchanged in a session. However, this round efficiency is achieved at the cost of huge soundness error. The work [FLS99] provides a very popular method–the so-called FLS-paradigm– to construct four round zero knowledge argument with negligible soundness error. In the FLS-paradigm, a zero knowledge protocol for proving some NP statement $x \in L$ proceeds in two phases. In the first phase, the verifier generates two puzzles and proves to the prover that he knows a solution to one of these puzzles; In the second phase, the prover proves to the verifier that either the statement being proven is true or he knows a solution to one of puzzles. Both proofs are carried out using a witness indistinguishable proof of knowledge. In simulation, an efficient simulator is able to extract a solution to one of these puzzles from a malicious verifier and then carry out the second phase using the solution just extracted as a witness.

Whether there are 3-round zero knowledge protocols with negligible soundness error based on standard assumptions for non-trivial languages is still a widely open problem. On the negative side, the work [GK96] showed that it is impossible to achieve 3-round zero knowledge argument or proof via black-box simulation. Similar impossibility result [Pas11] hold even for a relaxed notion of zero knowledge–witness hiding protocol [FS90]. Recently, Fleischhacker et al. [FGJ18] and Canetti et al. [CCH+19] extended this impossibility result to non-black-box simulation technique, and gave very strong negative evidence against the existence of 3-round zero knowledge *proofs* for non-trivial languages.

In their recently work [JKKR17], Jain et al. observed that a good distinguisher may leak some useful secrets of the verifier in certain settings, which will enable a successful simulation of the verifier's view. They developed a distinguisher-dependent simulation technique and constructed three-round delayed-input *weak* ϵ-distributional zero knowledge [DNRS03] from standard assumptions in a model where the simulator is allowed to depend on the distinguisher. Very recently, Bitansky et al. [BKP19] introduced a homomorphic trapdoor paradigm and presented a three-round *weak* ϵ-zero knowledge argument in the same model, but their simulator works for any individual statement (rather than in the distributional setting). Both constructions of [JKKR17,BKP19] can be made into two rounds assuming certain *sub-exponential* hardness.

Concurrent Zero Knowledge Protocols and the Bare Public Key (BPK) Model. Dwork et al. [DNS98] formalized the notion of concurrent zero knowledge in a setting where multiple sessions of the same protocol take place, and a malicious verifier is allowed to fully control the message scheduling. A protocol is called concurrent zero knowledge if it preserves zero knowledge even in this concurrent setting. Prabhakaran et al. [PRS02] refined the analysis of the simulators of [KP01,RK99] and proved (almost) logarithmic ($\widetilde{O}(\log n)$) round-complexity is sufficient for concurrent zero knowledge protocol, which almost matches the black-box lower bound of [CKPR01]. In his breakthrough

work [Bar01], Barak introduced a non-black-box simulation technique that makes use of the malicious verifier's code in simulation, and generated a long-line follow-up works (e.g., [DGS09, CLP13, BP15], just to name a few) to reduce the round-complexity of concurrent zero knowledge. However, despite decades of intensive research, the known constant-round constructions [CLP15a, FKP19] of concurrent zero knowledge still require non-standard assumptions.

Canetti et al. [CGGM00] introduced a very attracting model–the BPK model–to further reduce the round-complexity of stronger notions of zero knowledge, such as concurrent zero knowledge and resettable zero knowledge (which allows a verifier to reset the prover). In this model, each verifier deposits a public key in a public file and stores the associated secret key before any interaction with the prover begins. A huge advantage of this model is that, the trapdoors/secret keys useful for the simulator are fixed in advance, and if a simulator obtained all these trapdoors, it can simulate any session in a straight-line manner. Many constructions [YZ07, DFG+11, SV12] of concurrent/resettable zero knowledge in this model follows the FLS paradigm in which the verifier proves knowledge of his secret key in the first phase, and thus they require at least four rounds.

The question of whether we can achieve concurrent zero knowledge in fewer rounds in the BPK model is also subject to black-box limitations: As showed in [MR01, APV05], it is impossible to achieve concurrent *black-box* zero knowledge with concurrent (even sequential) soundness in three rounds in this model.

1.2 Motivation

In black-box simulations mentioned above, a simulator is usually to extract a piece of secret information from the adversary and then use it to mimic the honest parties (without knowing their private inputs). For such an extraction to go through, we usually design protocols so that the adversary is required to provide a proof of knowledge of such a piece of secret information. This incurs several additional rounds of interaction given the state-of-the-art constructions of proof of knowledge.

Indeed, Barak showed the adversary's code and internal workings allow us to break black-box barriers in certain settings. His non-black-box simulation technique relies on the PCP mechanism and often gives rise to complicated and (relatively) round-inefficient constructions. So far, for almost all known simulation techniques (including Barak's non-black-box simulation), the simulator is *universal* and is able to work for any adversary. This is in sharp contrast to the *individual* simulators, as required in most of security definitions, which switches the order of qualifiers $\exists\ \mathsf{Sim}\ \forall\ \mathcal{A}dv$:

- Universal Simulation: $\exists\ \mathsf{Sim}\ \forall\ \mathcal{A}dv$, Sim fools all efficient distinguishers.
- Individual Simulation: $\forall\ \mathcal{A}dv\ \exists\ \mathsf{Sim}$, Sim fools all efficient distinguishers.

Literally, an individual simulator is only required to work for a given *individual* adversary, thus we can assume that the simulator "knows/hardwires" any useful properties/structures (if exists) of this adversary's *functionality*, not just

its code. This makes individual simulators more powerful than universal/black-box ones. Under the widely believed hardness of reverse engineering[2], we cannot expect an efficient universal simulator to be able to figure out some useful property/structure about the adversary's functionality from its code. A natural question arises:

Can we develop individual simulations to break the known black-box barriers?

A motivating example is the black-box lower bound on round-complexity of concurrent zero knowledge [CKPR01], in which Canetti et al. constructed an explicit concurrent verifier strategy (for an arbitrary almost logarithmic round proof system) whose view cannot be simulated by any efficient black-box simulator (unless the statement being proven is trivial). However, as already showed in [Den17], an individual simulator can simulate this adversary's view in a straightforward way when given as input a certain crucial subfunctionality of the adversary. This demonstrates the potential power of individual simulations, but does not give a proof of the concurrent zero knowledge of the underlying protocol, which requires us to show for *any* efficient verifier we can build a successful individual simulator.

1.3 Summary of Our Results

In this paper we develop an individual simulation technique that explicitly makes use of particular properties/structures of the adversary's functionality, and achieve several constructions for selective opening secure commitment and zero knowledge arguments that break the known black-box lower bounds on their round-complexity.

As our main conceptual contribution, we show that for any NP-instance(s) sampling algorithm, there exists a nearly optimal *individual* witness extractor (depending on the sampler's functionality) that almost outperforms any circuit of a-priori bounded size. Combining this extraction strategy with an algebraic technique for Blum's encryption scheme, we obtain the following results.

The First Protocols That Break Previous Black-Box Barriers. We construct the *first* protocols that *break black-box barriers* mentioned above under the standard hardness of factoring, both of which are *polynomial time simulatable* against *all a-priori bounded polynomial size* distinguishers:

- Two-round selective opening secure commitment scheme.
- Three-round concurrent zero knowledge and concurrent witness hiding argument for NP in the bare public-key model.

All these protocols are quasi-polynomial time simulatable against all polynomial-size distinguishers with a *negligible* distinguishing gap.

Simpler Construction and Analysis of Zero Knowledge Protocols. We present a construction of two-round weak zero knowledge and witness hiding

[2] Under this assumption, the work [DGL+16] showed a limitation of universal simulation in a particular setting.

argument for NP in the plain model under the sub-exponential hardness of factoring, which is much simpler than the constructions in [JKKR17,BKP19,DK18, BGI+17]. Our technique also yields a significantly simpler proof of the equivalence theorem of [CLP15b]) for existing distinguisher-dependent simulatable zero knowledge protocols in [JKKR17,BKP19], showing that these protocols are also polynomial time simulatable against all distinguishers of a-priori bounded polynomial size.

1.4 Individual Extractions and Simulations: An Overview

Recall that the standard simulation-based security definitions only require that for every adversary, there *exists* a simulator that can fool all efficient distinguishers. This means such an existential simulator, like distinguishers, can depend on any properties/structures of the *functionality* of a given specific verifier.

Imagine that we have a two-round FLS-type protocol (A, B) in which B sends an NP instance y in the first round, with these properties:

1. A solution to the instance y generated by a adversary $\mathcal{B}$ enables the simulator to efficiently generate $\mathcal{B}$'s view that is indistinguishable from the real interaction;
2. Distinguishing the honest A's message from even a dummy message is equivalent to extracting a solution to y from $\mathcal{B}$.

In this scenario, for a given adversary $\mathcal{B}$, there are only two cases in which an efficient simulator will win[3]: a) the simulator succeeds to extract a solution to y from $\mathcal{B}$, or, b) no efficient algorithm can extract a solution to y except for negligible probability. In the former case, by the first property of (A, B), regardless of whether the distinguisher knows the solution, the simulator can reconstruct $\mathcal{B}$'s view successfully; in the latter case, the distinguisher does not know the solution either, and thus by the second property of (A, B), a simulator can easily fool the distinguisher.

Nearly Optimal Extractors for Single-instance Samplers. Note that the above solution extraction algorithm– the key subroutine of the simulator–can also be *individual*: It can depend on any property/structure of the individual adversary $\mathcal{B}$, besides being given the same input as $\mathcal{B}$.

To simulate $\mathcal{B}$'s view, one naive approach is to apply the best possible extractor (in terms of success probability) to extract a solution then simulate. An issue with this approach is that the success probability of an extractor may increase with its size. This makes it hard to control the size of the extractor (and the simulator). In this paper, we consider a weak simulation security–(T, ϵ)-simulatability: The simulation is required only against distinguishers of size T with distinguishing gap less than ϵ. Note that this notion is stronger than the

[3] Here we are aiming to construct a normal simulator, not a distinguisher-dependent simulator like the ones in [JKKR17,BKP19].

distinguisher-dependent simulatability defined in [CLP15b,JKKR17], where the simulator depends on the specific distinguishing algorithm, not just its size.

We view $\mathcal{B}$ as a single-instance sampler, and show that for any $\mathcal{B}$ there exists of a good extractor that outperforms all circuits of size T (given the *same* input as the extractor) with at most gap ϵ. The basic proof strategy is to keep iterating to include new powerful circuits into the extractor until we have a desired one.

Subtleties. One should be careful when carrying out this proof strategy. First, the number of iterations in this process may depend on the security parameter n, and this may cause some difficulties in controlling the size of the final circuit family Ext; second, in the asymptotic setting, when we add a new circuit family to the extractor, this family may work only when the security parameter n is greater than a specific n_0. Thus, it is possible that the iterative procedure keeps increasing the number n_0, and therefore we are not able to specify any n_0' so that the final circuit family Ext works for all $n > n_0'$.

To get around these difficulties, we use the a-priori fixed T and ϵ as a *global* guideline, and do *local* iterations at each parameter n[4]: In each iteration of this process, we have an extractor Ext at the beginning and ask: Does there *exist* another instance solver C of size T, given the same input as Ext, such that

$$\Pr\left[y \leftarrow \mathcal{B} : C \text{ extractsasolutionto } y \textit{ but } \mathsf{Ext} \text{ fails}\right] > \epsilon?$$

If so, then we have a new extractor: On input y, it runs the Ext first, and if Ext fails then runs C to extract a solution to y. This will increase the success probability of the extractor by at least ϵ; otherwise, we return the current extractor Ext.

It is not hard to verify that, after at most $\frac{1}{\epsilon}$ steps, we will have an extractor Ext of size at most $O(T\frac{1}{\epsilon})$ such that, the event that Ext fails to extract a solution to y but some other circuit of size T succeeds happens with probability at most ϵ.

The Dependence on the Functionality of the Sampler. We give two examples to illustrate how the nearly optimal extractor Ext *intrinsically* depends on the functionality of the sampler. Consider the following two image-sampling algorithms for some one-way permutation g: (a) use randomness y and then generate an image $x = g(y)$, and (b) sample a random string x from the co-domain of g. Then, for the former sampler, there is a nearly optimal extractor (taking the sampler's randomness y) that can simply output the pre-image y of the given sampled image x with probability 1; for the latter, a dummy algorithm (with success probability 0) is also an optimal extractor (this is almost best possible since g is one-way).

With this nearly optimal extractor, we now have an *individual* simulator for $\mathcal{B}$: it first applies this nearly optimal individual extractor Ext to extract a solution to y generated by $\mathcal{B}$ and then simulates in a somewhat straightforward manner (see below). Note that this simulator inherently depends on the functionality of

[4] We would like to stress that one cannot expect this process to be constructive.

the adversary (instance sampler) since the nearly optimal extractor does, and that it will fool all distinguishers[5] of size T except for probability at most ϵ.

Now, if the protocol (A, B) satisfies the above two properties, we have a good individual simulator against all distinguishers of size T. Our remaining task is to construct protocols with such properties.

A suitable building block for such protocols is the well-known encryption scheme based on the hardness of factoring. The public key of the encryption scheme is a Blum integer N, and the secret key is a prime factor of N. A ciphertext of a bit b is given by $c = (f_N(s), h(s) \oplus b)$, where $f_N : QR_N \to QR_N$ defined by $f_N(s) = s^2 \mod N$ and h is the hardcore of f_N. A key property (implied by [TW87]) of this encryption scheme we will make use of is the equivalence between distinguishing ciphertexts and extracting a secret key, even if the public key N is not a Blum integer[6].

Constructions. With these extraction and construction ideas in mind, we construct selective opening secure commitment and zero knowledge arguments as follows.

Two-Round Selective Opening Secure Commitment: In the committing phase, we have the receiver generate a Blum integer N for the committer; upon receiving N, the committer uses the trapdoor commitment scheme (a prime factor of N serves as a trapdoor) [FS89] to compute a commitment c, encrypts it bit-wise under the public-key N and sends these encryptions to the receiver; In the opening phase, the committer sends the opening of c to the receiver, and the latter decrypts the encryptions received in the first phase and accepts if the plaintext is c and the opening received is a valid opening of c. This construction relies on *polynomial hardness* of factoring.

Three-Round Weak Concurrent Zero Knowledge in the BPK Model: In the key registration phase, each verifier generates two Blum integers (N_0, N_1) as its public-key, and stores *two* prime factors (q_0, q_1), $q_i | N_i$ for $i \in \{0, 1\}$. In the proof phase, the prover and the verifier execute the three round parallel version of Blum's protocol (Let a session be of the form (a, e, z)) in which the prover proves "the statement to be proven is true or I know a prime factor of one of the two integers", and in addition, the prover encrypts the last message z bit-wise under each of verifier's public key. The verifier decrypts all these ciphertexts and obtains $\hat{z}$ and $\tilde{z}$, and accepts if $\hat{z} = \tilde{z}$ and the underlying transcript is accepting. This construction relies on *polynomial hardness* of factoring.

Two-Round Weak Zero Knowledge in the Plain Model: The verifier sends a Blum integer N (and stores one prime factor) to the prover, and the prover computes a commitment c to n zeros, sends back c together with ciphertexts (encrypted bit-wise under N) of a NIWI proof for "the statement to be proven is true or I know a prime factor of N". The verifier decrypts the ciphertexts, and accepts

[5] One can think of a distinguisher as a solution extractor since they are essentially equivalent because of the property 2. of (A, B).

[6] In this case, we view any prime factor of N as a secret key.

if the plaintexts forms an accepting NIWI proof. This construction relies on sub-exponential hardness of factoring.

A Difficulty in the Individual Simulations for Composable Protocols. At a high level, our simulation strategy for these protocols are quite simple: The simulator first applies the nearly optimal extractor to obtain the corresponding witness for each session, and if the extractor succeeds, then it can simulate this session in a straightforward manner; otherwise, it sends a dummy message in the last round of the protocol.

The Simulator for the Commitment Scheme. Suppose that a malicious receiver R^* initiates k sessions in parallel. In the committing phase, for each $i \in [k]$, the simulator first runs the nearly optimal extractor and tries to obtain a prime factor of N_i sent by R^*, and commits to 0 via the trapdoor commitment scheme and obtains a commitment c_i, then sends encryptions of c_i; In the opening phase, upon receiving $\{b_i\}_{i \in I}$ and the index set I, then the simulator opens c_i in the following way: If $b_i = 0$, open it in an honest way; if $b_i = 1$ and the extractor succeeds to extract a prime factor of N_i, then use it as trapdoor and open c_i to value 1; else send $(b_i = 1, dec')$ to R^*, where the decommiment $(b_i = 1, dec')$ is a valid opening of some commitment c'_i. (In other words, in the third case, the simulator pretends that the ciphertexts it sent in the committing phase is bit-wise encryptions of c'_i).

The Simulators for zero knowledge protocols are much simpler. For concurrent zero knowledge protocol in the BPK model, after the key registration phase, for each pair (N_0, N_1) registered by a malicious V^*, the simulator first tries to extract a prime factor of one of (N_0, N_1) using the nearly optimal extractor; if this extraction is successful, then the simulator can simulate any session under (N_0, N_1) successfully; otherwise, the simulator simply computes encryptions of all zeros under both public keys in the last round. The same simulation strategy works also for the protocol in the plain model.

One must be careful in proving that these simulations are indistinguishable from the real interaction against any distinguisher of a-priori bounded size T except for small probability ϵ. A technical difficulty arises in such proofs due to the *composition* of the first two protocols. Let us take the example of the simulator for the commitment scheme. As usual, the proof of (T, ϵ)-simulatability is done by a hybrid argument. We construct a sequence of hybrid non-uniform simulators, gradually switching from the simulation to the real interaction, so that a consecutive pair of simulators, say the i-th and the $(i+1)$-th simulators, behave differently only in the i-th session in the case that the extractor fails to factor N_i, and then prove that any two consecutive simulations are indistinguishable except for a very small probability by contradiction: For any D_n of size T that distinguishes the i-th and the $(i+1)$-th simulations with a large distinguishing gap, we use D_n to construct a circuit A_n that contradicts the optimality of the nearly optimal extractor. However, to exploit the power of D_n, A_n needs also to simulate other sessions for D_n, which in turn requires A_n to know prime factors for some other N_j's ($j \neq i$) obtained by the extractor. (otherwise A_n needs to

run the extractor on its own, which results in the circuit A_n of size larger than the extractor and thus makes no sense).

Nearly Optimal Extractors for Multi-instance Samplers. We prove a stronger result of the existence of nearly optimal extractors for all multiple-instance sampling algorithms to address the above issue. Specifically, for any polynomial t and any t-instance sampler, we show there exists a nearly optimal extractor such that, for every $i \in [t]$, for any circuit C of a-prior bound size that is given *the output of the extractor*, the probability that C solves the i-th instance but the extractor fails is small. This result is proved by a similar argument as above, but a more delicate iterative procedure is requried.

Binding/Soundness: Trust the Adversary. At first glance, the binding and soundness properties of the first two protocols seem to be problematic. For the binding of our commitment scheme, a usual proof-by-contradiction approach is to construct a reduction with oracle access to the cheating committer to factor the public key N. A problem with this approach is that the reduction itself does not know the corresponding secret key (i.e., a prime factor of N), and as a consequence, it cannot decrypt the message from the committer to obtain the commitment c and determine whether the opening sent by the cheating committer is a valid decommitment of c. Here we use a "trust the adversary" trick to save the proof: Since the cheating committer can make the *honest* receiver (who knows the secret key) accept two different decommitments, these decommitments should be valid for the same commitment c. Hence, in reduction, the reduction algorithm can trust the committer and simply assume that the two decommitments are both valid for some unknown c.

A similar but more subtle problem occurs in the proof of soundness of the zero knowledge protocol in the BPK model. In this case, a usual reduction algorithm keeps one secret key of N_i (for a random $i \in \{0, 1\}$) in the public key pair (N_0, N_1), and wants to use the power of the cheating prover to factor N_{1-i}. However, such a reduction seems to fail for the following cheating P^*: At the begining P^* somehow magically factors *both* N_0 and N_1 and obtains q_0 and q_1; in its last step, it compute z_0 and z_1 using witnesses q_0 and q_1 respectively, and sends to the verifier encryptions of z_0 and z_1 under the public keys N_0 and N_1 respectively. Note that the reduction can decrypt only the encryptions under public key N_i, and hence it can only obtain a prime factor of N_i by rewinding P^* (using the special soundness of Blum's protocol). However, this issue is taken care by the verification step in which the honest verifier decrypts *all* encryptions and check if the two last round messages z_0 and z_1 are equal and both acceptable. Thus, such a cheating P^* cannot make an honest verifier accept at all, and therefore is not a successful cheating prover. In other words, for a successful cheating prover, the reduction algorithm can trust that the two last round messages of Blum's protocol encrypted under both public keys are equal. This is the key to the proof of soundness.

1.5 Related Work and Discussion

On Upgrading the Distinguisher-Dependent Simulatable Zero Knowledge. As mentioned earlier, it is proved in [CLP15b] that, in the *plain* model, distinguisher-dependent simulatable zero knowledge protocols (such as [JKKR17,BKP19]) satisfy the stronger notion of (T, ϵ)-simulatability. However, this "distinguisher-dependent simulation then upgrade" approach to (T, ϵ)-simulatability seems to work only for *standalone* zero knowledge protocols in the *plain* model. Note that the equivalence theorem of [CLP15b] says nothing about zero knowledge in other models/settings, or other cryptographic primitives, like the commitment schemes under *parallel composition* and *concurrent* zero knowledge in the BPK model considered in this paper.

The equivalence theorem of [CLP15b] was proved via the minimax theorem, which leads to a complicated proof[7]. Our proof of existence of a nearly optimal extractor is quite simple and easy to understand, and it can also be used to upgrade existing constructions of [JKKR17,BKP19]. However, it is unclear if our technique could be used to prove the full version of the equivalence theorem of [CLP15b].

Other Notions of Selective Opening Security for Commitments. The work of [BHY09] also introduced the notion of selective opening security under *concurrent* composition, where a malicious receiver is allowed to interact with the committers concurrently. This notion is stronger than the selective opening security under *parallel* composition considered in this paper. However, as proved in [ORSV13], we cannot achieve such a security in the *full-fledged* concurrent setting if the simulator does not know the distribution of the message committed to by the honest committer. Another related notion is the indistinguishability-based selective opening security, which can be achieved by any statistical hiding (standalone) commitment scheme [BHY09].

Conditional Disclosure Schemes. A conditional disclosure scheme can be thought of as *interactive* version of witness encryption [AIR01,BP12,PA17]. It is a useful tool for constructing protocols of low round-complexity, such as the three round zero knowledge protocol of [BKP19], but the usage of such a scheme often requires an additional sub-protocol to make sure a (malicious) party indeed knows a relevant witness. The protocols in this paper do not need such an extra sub-protocol, and therefore is significantly simpler than previous constructions.

(T, ϵ)-Security in Practice. A silent feature of the notion of (T, ϵ)-simulatability is that the we need not embed the parameters T and ϵ into the protocol instructions. That is, we can have a *single* construction that achieves (T, ϵ)-simulatability for *any* polynomial T and *any* inverse polynomial ϵ, which stands in sharp contrast to Barak's n-bounded concurrent zero knowledge argument, whose construction depends on the a-priori upper-bound n on the number of total sessions allowed. From a practical point of view, we think the weak notion

[7] See https://eprint.iacr.org/2013/260.pdf for the detailed proof.

of (T, ϵ)-simulatability is good enough in practice: For any fixed security parameter λ, any constants κ and ε, it already achieves a *concrete* (κ, ε)-simulatability, since there always exist T and ϵ satisfying $T(\lambda) > \kappa$ and $\epsilon(\lambda) < \varepsilon$.

1.6 Organization

We present relevant definitions in Sect. 2. In Sect. 3, we prove the existence of nearly optimal extractors for all hard distributions. In Sect. 4, we give a formal proof of the equivalence between distinguishing ciphertexts and extracting a secret key for the factoring-based encryption scheme. In the last three sections, we give our main results on selective opening secure commitment, weak concurrent zero knowledge in the BPK model and the two-round weak zero knowledge respectively.

2 Preliminaries

Throughout the paper, we let n be the security parameter. We write the set $\{1, 2, ..., m\}$ as $[m]$, and the set $\{i, i+1, ..., j\}$ as $[i, j]$. We denote by $\bar{x} = \{x_i\}_{i\in[k]} \leftarrow \bar{D}^k$ the process of sampling k times x from D independently. A function $negl(n)$ is called negligible if it vanishes faster than any inverse polynomial. We write $\{X_n\}_{n\in\mathbb{N}} \overset{c}{\approx} \{Y_n\}_{n\in\mathbb{N}}$ to indicate that the two distribution ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable. A Blum integer N is a product of two primes p, q satisfying $p, q \equiv 3 \bmod 4$. We denote by $\mathsf{Blum}(1^n)$ the algorithm that on input a security parameter n outputs a Blum integer N and one of its prime factors q, where the corresponding two prime factors are of length n. Due to space limitations, we refer readers to [Gol01] for definitions of witness indistinguishability, witness hiding.

Commitment and Trapdoor Commitment Schemes. Commitment schemes are "digital" safes. Formally, a commitment scheme (C, R) is a two-phase protocol between a committer C and a receiver R. To commit to a bit $b \in \{0, 1\}$, $C(b)$ and R execute the committing phase of (C, R) (denoted by $(C, R)_{\mathsf{Com}}$) and generate a commitment transcript $\mathsf{Com}(b)$; To decommit $\mathsf{Com}(b)$, C and R execute the opening phase of (C, R) (denoted by $(C, R)_{\mathsf{Open}}$) and reveal a decommitment (b, dec), and R accepts if the decommitment is valid.

Definition 1 (Commitment Scheme). *A two-phase protocol (C, R) is called a commitment scheme if it satisfies the following two properties:*

- *Binding: For every committer C^* of polynomial-size, the probability of the following event is negligible: C^* interacts with R and generates a commitment $\mathsf{Com}(b)$ in the committing phase, and then produces two decommitments (b, dec) and (b', dec') with $b \neq b'$ in two executions of the opening phase.*
- *Hiding: For every receiver R^* of polynomial size, the commitments $\mathsf{Com}(0)$ and $\mathsf{Com}(1)$ are computational indistinguishable.*

A trapdoor commitment scheme is a commitment scheme with an additional property: Given a trapdoor, C can later open a commitment to different values. In [FS89], Feige and Shamir showed how to transform Blum's 3-round interactive proof into a trapdoor commitment scheme. In our construction of selective opening secure commitment, we need a version of Feige-Shamir trapdoor commitment based on factoring. Using a standard commitment (built from the factoring assumption) Com as a building block, our trapdoor commitment scheme $(\mathsf{TDGen}, \mathsf{TDCom}, \mathsf{Open}, \mathsf{Fakeopen})$ proceeds as follows.

- TDGen: On input the security parameter n, TDGen generates $(N, q) \leftarrow \mathsf{Blum}(1^n)$. Define an NP relation $\{(N, q) : q|N\}$, and transform (N, q) into a graph G and an associated Hamiltonian cycle $H \subseteq G$. Output $((N, G), q)$.
- TDCom: On input G, a bit b and randomness r, if $b = 0$, pick a random permutation π and commit to the adjacency matrix of $\pi(G)$; if $b = 1$, pick a random cycle H' and commit to the adjacency matrix of H'. In both cases, we use commitment scheme Com when committing to the adjacency matrix.
- Open: On input $(G, \mathsf{TDCom}(G, b, r), b, r)$, if $b = 0$, send π and open the entire adjacency matrix of $\pi(G)$; if $b = 1$, open the non-zero entries in the adjacency matrix of H' (i.e., open the cycle H'). We denote by (b, dec) the decommitment of the commiment $\mathsf{TDCom}(G, b, r)$.
- $\mathsf{Fakeopen}$: On input $(G, H, \mathsf{TDCom}(G, 0, r), b, r)$, open to b in the same way as Open by setting $H' = \pi(H)$. Note that only when TDCom commits to 0, the commitment can be opened to both 0 and 1.

A Crucial Property. Our construction of a selective opening secure commitment scheme relies on the following property of the above trapdoor commitment scheme, which can be easily proved by applying standard hybrid argument to the underlying commitment scheme Com:
$\{(c, (1, dec)) : c \leftarrow \mathsf{TDCom}(G, 1, r); (1, dec) \leftarrow \mathsf{Open}(G, \mathsf{TDCom}(G, 1, r), 1, r)\}$ and
$\{(c, (1, dec)) : c \leftarrow \mathsf{TDCom}(G, 0, r); (1, dec) \leftarrow \mathsf{Fakeopen}(G, H, \mathsf{TDCom}(G, 0, r), 1, r)\}$
are indistinguishable.

(T, ϵ)-Secure Under Selective Opening Attacks. Consider a k-parallel composition of a commitment scheme (C, R). A committer C^k and a receiver R^* execute the committing phase k times in parallel and generate k commitments $\{\zeta^i\}_{i\in[k]}$ to $\bar{b} = b_1||b_2||\cdots||b_k$, each ζ^i is a commitment to b_i. In a selective opening attack, R^* chooses a set $I \in \mathcal{I}$ (possibly depending the commitments received) and asks the committer C^k to open the commitments $\{\zeta^i\}_{i\in I}$, where $\mathcal{I}$ is the family of subset of $[k]$. Informally, the commitment scheme (C, R) is said to be secure under selective opening attacks if the remaining unopened commitments still stay secret.

Definition 2 ((T, ϵ)-secure under selective opening attacks). *Let k be an arbitrary polynomial in n, and $\mathcal{B}$ be a distribution on $\{0, 1\}^k$, and $\mathcal{I}$ be the family of subset of $[k]$. A commitment scheme (C, R) is (T, ϵ)-secure under selective opening attacks if for any polynomial T, any inverse polynomial ϵ, any polynomial size $\mathcal{B}$, and any polynomial size R^*, there exists a polynomial size Sim such*

that for any distinguisher D_n *of size* T, D_n *cannot tell apart the following two distributions*

- $(C^k(\bar{b}), R^*)$: $\bar{b} \leftarrow \mathcal{B}$; $\{\zeta^i\}_{i\in[k]} \leftarrow (C^k(\bar{b}), R^*)_{\mathsf{Com}}$; $I \leftarrow R^*(\{\zeta^i\}_{i\in[k]})$; $\{(b_i, dec_i)\}_{i\in I} \leftarrow (C^k(\bar{b}), R^*)_{\mathsf{Open}}$; $Out_{R^*} \leftarrow R^*(\{(b_i, dec_i)\}_{i\in I})$. *Output* $(\bar{b}, I, Out_{R^*})$;
- SIM: $\bar{b} \leftarrow \mathcal{B}$; $I \leftarrow \mathsf{Sim}$; $Out_{\mathsf{Sim}} \leftarrow \mathsf{Sim}(\{b_i\}_{i\in I})$. *Output* $(\bar{b}, I, Out_{\mathsf{Sim}})$,

with probability greater than ϵ, *i.e.,*

$$|\Pr[\mathsf{D}_n((C^k(\bar{b}), R^*)) = 1] - \Pr[\mathsf{D}_n(\mathrm{SIM}) = 1]| < \epsilon.$$

Delayed Input Argument and (T, ϵ)**-ZK.** Let L be an NP language and R_L be its associated relation. An interactive argument system (P, V) for L is a pair of parties of polynomial size, in which the prover P wants to convince the verifier V of some statement $x \in L$. We denote by $(P, V)(x)$ the output of V at the end of interaction on common input x, and by $\mathsf{View}_V^P(x)$ the view of the verifier in the real interaction. Without loss of generality, we have the verifier V outputs 1 (resp. 0) if V accepts (resp. rejects).

In this paper we consider *delayed-input* interactive arguments, in which the common input to both parties is the size of the statement x, and the verifier receives x only in the last round. Note that in a delayed-input interactive argument, a malicious prover may choose statement depending on the history, and thus such an argument needs to satisfy a stronger notion of adaptive soundness (cf. [JKKR17]).

A delayed-input argument system is zero knowledge if the view of the (even malicious) verifier in an interaction can be efficiently reconstructed. In this paper, we consider a weak version of zero knowledge–(T, ϵ)-zero knowledge [CLP15b], in which the indistinguishability gap between the real interaction and the simulation is at most ϵ against any T-size distinguisher.

Definition 3 (Delayed-input (T, ϵ)**-zero knowledge).** *We say that a delayed-input interactive argument* (P, V) *for language* L *is* (T, ϵ)*-zero-knowledge if for any polynomial* T, *any inverse polynomial* ϵ, *any polynomial-size* V^*, *there exists a circuit* Sim *of polynomial size such that for any* $x \in L$ *and any probabilistic* T*-size circuit* $\{D_n\}_{n\in\mathbb{N}}$ *and sufficiently large* n, *it holds that*

$$\left|\Pr[\mathsf{D}_n(\mathsf{View}_{V^*}^P(x)) = 1] - \Pr[\mathsf{D}_n(\mathsf{Sim}(x)) = 1]\right| < \epsilon.$$

Concurrent Zero Knowledge with Concurrent Soundness in the BPK Model. The bare public-key model (BPK model) simply works in two phases: the key-registration phase and the proof phase. In the key-registration phase, each verifier registers a public-key pk (the honest verifier is supposed to store the corresponding secret key sk) on a public-file F before the proof phase. In the proof phase, on a common input x, the prover and the verifier interact under the verifier's public key. The completeness of an interactive argument is normally defined.

Concurrent Soundness in the BPK Model. A malicious concurrent prover P^* is allowed to launch the following attack: In the proof phase, on input a pubic key pk, P^* initiates polynomially many sessions, in each of which it chooses a statement x adpatively (based on the history so far), and fully controls the message scheduling in the entire interaction with V.

Definition 4 (Concurrent Soundness in the BPK model). *An interactive argument (P, V) for a language L in the BPK model is called* concurrent sound *if for all malicious* concurrent *prover P^*, the probability that it makes V accept a false statement $x \notin L$ is negligible.*

Concurrent (T, ϵ)-Zero Knowledge in the BPK Model. A malicious concurrent verifier V^* is allowed to generate an arbitrary file F of polynomially many public keys in the key-registration phase. In the proof phase, it receives s (for some polynomial s) statements $\bar{x} = \{x_i\}_{i\in[s]}$, and initiates at most s sessions under public keys on F. During the entire interaction, V^* fully controls the message scheduling.

Definition 5 (Concurrent (T, ϵ)-zero knowledge In the BPK model). *An interactive argument (P, V) for language L is called concurrent (T, ϵ)-zero-knowledge if for any polynomial T, any inverse polynomial ϵ, any polynomial-size concurrent V^*, any polynomial s, there exists a circuit* Sim *of polynomial size such that for any Yes instances $\bar{x} = \{x_i\}_{i\in[s]}$, for any probabilistic T-size circuit $\{\mathsf{D}_n\}_{n\in\mathbb{N}}$ and sufficiently large n it holds that*

$$\left|\Pr[\mathsf{D}_n(\mathsf{View}_{V^*}^{P(F)}(\bar{x})) = 1] - \Pr[\mathsf{D}_n(\mathsf{Sim}(\bar{x})) = 1]\right| < \epsilon.$$

3 The Existence of Nearly Optimal Extractors for All Hard Distribution

In this section we prove the existence of nearly optimal extractors for all NP-instance(s) sampling algorithms. Essentially, we show that, for any NP-instance(s) sampler, any polynomial T, any inverse polynomial ϵ, and any circuit family C_n of size T, there exists an efficient extractor such that the probability that C_n extracts a witness for an instance generated by the sampler but the extractor fails is at most ϵ. Furthermore, if the extractor is allowed to be of quasi-polynomial size, then the same result holds with respect to *negligible* ϵ.

Let Samp be an arbitrarily sampling algorithm over an NP language L and $\{Y_n\}_{n\in\mathbb{N}}$ be its input distribution ensemble. Throughout this paper, we assume that the input $y \leftarrow Y_n$ to Samp includes its randomness. (Thus one can view Samp as a deterministic algorithm.)

Lemma 1 [nearly optimal (T, ϵ)-Extractor]**.** *Let* Samp *be as above. Let $f : \{0,1\}^* \to \{0,1\}^*$ be an arbitrary (not necessarily efficient-computable) function.*

1. *For every polynomial T, every inverse polynomial ϵ, there exists a probabilistic circuit family* $\mathsf{Ext} := \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ *of polynomial size such that for every probabilistic circuit family* $\{C_n\}_{n\in\mathbb{N}}$ *of size* T*,*

$$\Pr\left[\begin{array}{c} y \leftarrow Y_n; x \leftarrow \mathsf{Samp}(y); \\ w \leftarrow \mathsf{Ext}_n(x, y, f(y)); \\ w' \leftarrow C_n(x, y, f(y)) \end{array} : \begin{array}{c} (x,w) \notin R_L \wedge \\ (x,w') \in R_L \end{array}\right] < \epsilon(n) \tag{1}$$

   We call Ext *a* (T,ϵ)*-extractor.*
2. *There exists a probabilistic circuit family* $\mathsf{Ext} := \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ *of quasi-polynomial size such that for every probabilistic circuit family* $\{C_n\}_{n\in\mathbb{N}}$ *of polynomial size, the above probability is negligible.*

Remark 1. Jumping ahead, in our protocols the receiver/verifier will play the role of the hard instance sampler. For all our constructions, we need not take the function f into account since they just compute a hard instance based solely on their random tape. However, when our protocols are used as a sub-protocol in some big protocols or in the settings of [JKKR17,BKP19], the receiver/verifier may compute a hard instance based on some history y, and the simulator may need certain secret information $f(y)$ (e.g., an opening of a commitment in history y) to go through. In such cases, it is more flexible to allow the extractor to take as additional input $f(y)$.

As mentioned in the introduction, the basic idea underlying the proof is to keep iterating to include new powerful circuits into the extractor until we have a desired one. For applications, we need a stronger and robust version of Lemma 1 for samplers that output multiple instances, which we prove below.

Fix a polynomial t and consider a t-instance sampler Samp that is given y as input and outputs t instances of NP language L, $(x_1, x_2, \cdot, \cdot, \cdot, x_t) \leftarrow \mathsf{Samp}(y)$, where y is drawn from distribution Y_n.

Lemma 2 [nearly optimal (T,ϵ)-Extractor for t-Instance Sampler]**.** *Let L be an NP language and poly be the size of the circuits for deciding the NP-relation R_L. Let* Samp *be an arbitrarily t-instance sampling algorithm over L with input distribution ensemble $\{Y_n\}_{n\in\mathbb{N}}$. Let $f : \{0,1\}^* \to \{0,1\}^*$ be an arbitrary (not necessarily efficient-computable) function.*

1. *For every polynomial T, every inverse polynomial ϵ, there exists a probabilistic circuit family* $\mathsf{Ext} := \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ *of size $O(\frac{t}{\epsilon}(T + poly))$, such that for every $j \in [t]$, every probabilistic circuit family $\{C_n\}_{n\in\mathbb{N}}$ of size T,*

$$\Pr\left[\begin{array}{c} y \leftarrow Y_n; \{x_k\}_{k\in[t]} \leftarrow \mathsf{Samp}(y); \\ \{w_k\}_{k\in[t]} \leftarrow \mathsf{Ext}_n(\{x_k\}_{k\in[t]}, y, f(y)); \\ w'_j \leftarrow C_n(\{x_k\}_{k\in[t]}, \{w_k\}_{k\in[t]}, y, f(y)) \end{array} : \begin{array}{c} (x_j,w_j) \notin R_L \wedge \\ (x_j,w'_j) \in R_L \end{array}\right] < \epsilon(n), \tag{2}$$

   where the probability takes over the randomness choice of y, and the random tapes for that for Ext_n *and* C_n*.*

2. *There exists a probabilistic circuit family* $\mathsf{Ext} := \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ *of quasi-polynomial size such that for every* $j \in [t]$ *and every probabilistic circuit family* $\{C_n\}_{n\in\mathbb{N}}$ *of polynomial size, the above probability is negligible.*

Remark 2. Notice that in the above lemma we allow the circuit C_n to take the output of Ext_n as input. This does not matter for a single-instance sampler. However, as we shall see in Sect. 5 and 6, this property is critical for hybrid arguments to go through in the composable settings.

Lemma 2 says there is an extractor for the multi-instance sampler that is nearly optimal for solving instances in every coordinate $j \in [t]$. We argue the existence of such a nearly optimal extractor via the following delicate iterative procedure. In each *outer* iteration $i \in [\frac{t}{\epsilon}]$, for every $j \in [t]$ we ask if there is circuit $C_{n,j}^{(i)}$ that, *taking as input the output of the current* Ext_n, can be used to increase the success probability of solving the j-th instance x_j by (at least) ϵ, and if so, then we add $C_{n,j}$ to Ext_n.

Proof (of Lemma 2). For every $j \in [t]$, we define $\uplus_j$ composition of two circuits Ext_n and $C_{n,j}$ in the following way:

$\mathsf{Ext}_n \uplus_j C_{n,j}(\{x_k\}_{k\in[t]}, y, f(y))$:
1. Sampling a random tape for Ext_n, obtain $\{w_k\}_{k\in[t]} \leftarrow \mathsf{Ext}_n(\{x_k\}_{k\in[t]}, y, f(y))$;
2. If $(x_j, w_j) \in R_L$, return $\{w_k\}_{k\in[t]}$;
3. Sampling a random tape for $C_{n,j}$, obtain $w'_j \leftarrow C_{n,j}(\{x_k\}_{k\in[t]}, \{w_k\}_{k\in[t]}, y, f(y))$;
4. If $(x_j, w'_j) \in R_L$, then $w_j \leftarrow w'_j$ and return $\{w_k\}_{k\in[t]}$; otherwise, return $\{w_k\}_{k\in[t]}$.

Note that the order of executions of these two circuits matters here since we have the second circuit take as input the output of the first circuit. This applies to each iteration of the following construction, and the final circuit Ext_n will execute all these $C_{n,j}^i$ in the order of their appearance. Let $\mathsf{Ext}_n^{(0)}$ be a dummy circuit that outputs t zeros. For an arbitrary t-instance Samp, we construct a nearly optimal extractor Ext_n as follows[8].

Constructing circuit Ext_n **for the** t**-instance** Samp:
1. $\mathsf{Ext}_n \leftarrow \mathsf{Ext}_n^{(0)}$;
2. For $i = 1$ to $\frac{t}{\epsilon}$, do:
   2.1 For $j = 1$ to t, do:

[8] We would like to stress that in this construction the number of outer iterations may reach $\frac{t}{\epsilon}$. Notice that in each iteration, the quality of the current extractor may have impact on the answer to the question of whether or not there exists a new satisfactory circuit $C_{n,j}^{(i)}$ since the new target circuit is given the output of the current extractor. Thus, even if there does not exists a satisfactory $C_{n,j}^{(i)}$ in the i-th outer iteration, we cannot rule out the possibility that we will find a satisfactory $C_{n,j}^{(i+1)}$ in the $(i+1)$-th outer iteration, because the extractor would become more powerful as iterations proceed.

If $\exists$ a circuit $C_{n,j}^{(i)}$ of size T s.t.

$$\Pr\left[\begin{array}{c} y \leftarrow Y_n; \{x_k\}_{k\in[t]} \leftarrow \mathsf{Samp}(y); \\ \{w_k\}_{k\in[t]} \leftarrow \mathsf{Ext}_n(\{x_k\}_{k\in[t]}, y, f(y)); \\ w'_j \leftarrow C_{n,j}^{(i)}(\{x_k\}_{k\in[t]}, \{w_k\}_{k\in[t]}, y, f(y)) \end{array} : \begin{array}{l} (x_j, w_j) \notin R_L \wedge \\ (x_j, w'_j) \in R_L \end{array}\right] \geq \epsilon(n), \tag{3}$$

then $\mathsf{Ext}_n \leftarrow \mathsf{Ext}_n \uplus_j C_{n,j}^{(i)}$;

2.2 If for any $j \in [t]$, $\nexists\, C_{n,j}^{(i)}$ satisfying (3), then break and return Ext_n.

3. Return Ext_n

We now show that the Ext_n constructed above satisfies Lemma 2. We first make the following two observations:

1. For any $j' \neq j$, the circuit $\mathsf{Ext}_n \uplus_{j'} C_{n,j'}^{(i)}$ solves the j-th instance x_j with exactly the same probability of Ext_n. This is because in the above composition $C_{n,j}$ is only invoked to correct the witness w_j obtained by Ext_n.
2. For each new $C_{n,j}^{(i)}$, the circuit $\mathsf{Ext}_n \uplus_j C_{n,j}^{(i)}$ increases the success probability of solving the j-th instance x_j by (at least) ϵ.

Note that if in some outer iteration $i \leq \frac{t}{\epsilon}$, no new circuit is added to Ext_n in any inner iteration $j \in [t]$, then the iterative process will return a desirable circuit Ext_n as required in Lemma 2; otherwise, the following two events must happen during the entire iterative process: (a) There are (at least) $\frac{t}{\epsilon}$ circuits $C_{n,j}^{(i)}$ of size T that are added to Ext_n, and (b) For each $j \in [t]$ the number of circuits $C_{n,j}^{(i_m)}$ ($i_m \in [\frac{t}{\epsilon}]$) added to Ext_n is at most $\frac{1}{\epsilon}$. The latter event (b) holds because of the two observations mentioned above, which imply that adding more than $\frac{1}{\epsilon}$ circuits $C_{n,j}^{(i_m)}$ would yield an extractor with success probability of solving the j-th instance greater than 1.

Putting (a) and (b) together, we have that, for every j, exactly $\frac{1}{\epsilon}$ circuits $C_{n,j}^{(i_m)}$ are added to Ext_n, and the final circuit Ext_n returned solves the j-th instance with probability 1. It is easy to verify that the size of the final Ext_n is of at most $O(\frac{t}{\epsilon}(T + \mathrm{poly}))$. This concludes Lemma 2.

For the second part of this lemma, one can set T and ϵ to be $n^{\omega(1)}$ and $\frac{1}{n^{\omega(1)}}$ respectively, construct the circuit family $\mathsf{Ext} = \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ of size $n^{\omega(1)}$ in a similar way. □

4 Extracting the Secret Key of a Variant of Rabin's Encryption Scheme

We are now going to apply Lemma 2 to a variant of a factoring-based encryption scheme, and show the existence of a nearly optimal secret-key extractor, such that the probability that an arbitrary bounded-size circuit family succeeds in distinguishing ciphertexts but the extractor fails to extract a secret key is very small.

We consider an encryption scheme based on Rabin's trapdoor one-way permutations. Let N be a Blum integer of length n, and QR_N be the set of quadratic residues (mod N). Rabin's trapdoor one way permutation $f_N : QR_N \to QR_N$ (with a prime factor of N as its trapdoor) is defined as $f_N(s) = s^2 \bmod N$. The one-wayness of f_N is based on the fact that different square roots lead to factor N. Specifically, given a circuit A of size T that inverts $f_N(s)$ with probability ε, by Lemma 10 in [TW87], we have a circuit of size $O(T\frac{1}{\varepsilon})$ that can factor N with probability negligibly close to 1.

Let $h(\cdot)$ be an arbitrary hard-core function of $f_N(\cdot)$[9]. We follow the classic approach and obtain the following semantically secure bit encryption scheme $(\mathsf{Gen} = \mathsf{Blum}, \mathsf{Enc}, \mathsf{Dec})$. The public key is a randomly generated Blum integer N, and the secret key is a prime factor of N:

- Enc_N: To encrypt a bit b, the encryption algorithm Enc selects a random $s \in QR_N$ (which can be done by selecting a random $t \in Z_N$ and then set s to be $t^2 \bmod N$), and computes $f_N(s)$ and $h(s) \oplus b$. Enc outputs the ciphertext $c = (f_N(s), h(s) \oplus b)$;
- Dec_N: To decrypt a ciphertext c, the decryption algorithm Dec uses the secret key to invert the first part of c, and then computes $h(s)$ and outputs b.

The semantic security follows from the hardness of factoring assumption: A good ciphertext distinguisher will give rise to an efficient algorithm that finds square roots modulo N, which can be used to factor N.

In our constructions of commitment and zero knowledge protocols, we will have one party generate one (or two) public key(s) of the above encryption scheme and use one secret key to decrypt the messages from the other party. We would like to stress that, in case that a malicious party generates a non-Blum integer as its public key, the function f_N in the encryption may no longer be a permutation. Fortunately, such a malicious behavior only causes difficulty for the malicious party to decrypt the ciphertext computed by the honest party, and does not affect the property –the equivalence between distinguishing ciphertexts and factoring– that is required to establish simulatability of our protocols.

We now give a formal statement of this property with respect to the encryption scheme above. Here we slightly abuse these notations, and define $f_N : QR_N \to QR_N$ and the "encryption" function $\mathsf{Enc}_N(b) := (f_N(s), h(s) \oplus b)$ over an arbitrary (positive) integer N.

Lemma 3 [Implied by [GL89, ACGS88, TW87]]**.** *For any positive integer N of length n and any inverse polynomial $\delta(n)$, if there exists a probabilistic circuit family $\{A_n\}_{n\in\mathbb{N}}$ of size T such that for any auxiliary input $\alpha \in \{0,1\}^*$,*

[9] The constructions of the hardcore of $f_N(\cdot)$ appeared in [ACGS88, GL89]. Note that, when using the Goldreich-Levin hardcore function [GL89], we need to change the description of our encryption scheme a little bit, since the Goldreich-Levin hardcore function is actually constructed for the permutation $f'_N(s, r) = (f_N(s), r)$ (where $|r| = |s|$). We ignore such changes in the description of our encryption scheme for the sake of simplifying the presentation.

$$\Pr[b \leftarrow \{0,1\}; c \leftarrow \mathsf{Enc}_N(b); A_n(c, N, \alpha) = b' : b = b'] \geq \frac{1}{2} + \delta(n)$$

then there exists a probabilistic circuit family $\{B_n\}_{n\in\mathbb{N}}$ *of size* $O(\frac{1}{\delta^5}n^3T)$ *that can factor* N *with probability*

$$\Pr[q \leftarrow B_n(N, \alpha) : q|N] \geq 1 - negl(n).$$

Proof Sketch. The hardcore theorems [GL89, ACGS88] state that, given a successful distinguisher A_n of size T for the "encryption" function Enc_N with advantage δ, we can construct a new circuit of size $O(\frac{1}{\delta^4}n^3T)$ that computes the square roots modulo N with roughly the same successful probability. If δ is an inverse polynomial, then by [TW87] such a square root circuit can be used to factor the integer N in size $O(\frac{1}{\delta^5}n^3T)$ with probability negligibly close to 1. □

Applying Lemma 2 to a t-integer sampler $\{N_i\}_{i\in[t]} \leftarrow \mathsf{Samp}$, we can show that there exists a nearly optimal extractor Ext for Samp such that for every j if Ext fails to extract a prime factor of N_j, then no circuit of a-prior bounded size can distinguish a ciphertext (except for small advantage). Formally, we obtain the following result (and defer the proof of this lemma to the full version).

Lemma 4. *Let* t *be a polynomial, and* Samp *be an arbitrarily* t*-integer sampling algorithm with input distribution ensemble* $\{Y_n\}_{n\in\mathbb{N}}$*. Let* $f : \{0,1\}^* \to \{0,1\}^*$ *be an arbitrary (not necessarily efficiently computable) function.*

1. *For any polynomial* T*, any inverse polynomial* ϵ*, there exists a probabilistic circuit family* $\mathsf{Ext} := \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ *of polynomial-size such that for every probabilistic circuit family* $\{A_n\}_{n\in\mathbb{N}}$ *of size* T*, for every* $j \in [t]$*, we have*

$$\Pr\left[\begin{array}{rl} y \leftarrow Y_n; \{N_i\}_{i\in[t]} \leftarrow \mathsf{Samp}(y); & \\ \{q_i\}_{i\in[t]} \leftarrow \mathsf{Ext}_n(\{N_i\}_{i\in[t]}, y, f(y)); & : \ b = b' \wedge \\ b \leftarrow \{0,1\}; c \leftarrow \mathsf{Enc}_{N_j}(b); & \ q_j \nmid N_j \\ b' \leftarrow A_n(c, \{q_i\}_{i\in[t]}, \{N_i\}_{i\in[t]}, j, y, f(y)) & \end{array}\right]$$
$$< \frac{1}{2}\Pr\left[\begin{array}{rl} y \leftarrow Y_n; \{N_i\}_{i\in[t]} \leftarrow \mathsf{Samp}(y); & \\ \{q_i\}_{i\in[t]} \leftarrow \mathsf{Ext}_n(\{N_i\}_{i\in[t]}, y, f(y)) & : \ q_j \nmid N_j \end{array}\right] + \epsilon(n)$$

2. *There exists a probabilistic circuit family* Ext *of quasi-polynomial size such that for every probabilistic circuit family* $\{C_n\}_{n\in\mathbb{N}}$ *of polynomial size, the above holds with respect to a negligible function* ϵ*.*

5 Selective Opening (T, ϵ)-Secure Commitment Scheme

We use the following ingredients in our construction of a selective opening secure commitment scheme:

– The trapdoor commitment $(\mathsf{TDGen}, \mathsf{TDCom}, \mathsf{Open}, \mathsf{Fakeopen})$ described in Sect. 2;

– The variant of Rabin's encryption scheme presented in Sect. 4.

With these two building blocks, we construct a selective opening secure commitment scheme as follows. In the committing phase, we have the receiver run the trapdoor generator and produce (N, q) $(q|N)$ and transform (N, q) into (G, H), then send N and the graph G to the committer; upon receiving N, the committer invokes TDCom and generates a commitment c, encrypts c bit-by-bit under the public key N, and sends all these encryptions to the receiver. In the opending phase, the committer simply sends the opening of c to the receiver, who decrypts the ciphertexts received in the committing phase using secret keys q and obtains c, and checks whether the opening received from the committer is a valid decommitment of c.

Formally, our selective opening secure commitment scheme proceeds as follows.

$\mathsf{Protocol}_{soa}$:

Committing phase:

$R \longrightarrow C$: $((N, G), q) \leftarrow \mathsf{TDGen}(1^n)$. Send (N, G).

$C \longrightarrow R$: $c = c_1||c_2|| \cdots ||c_\ell \leftarrow \mathsf{TDCom}(G, b, r)$, $\{\zeta_i \leftarrow \mathsf{Enc}_N(c_i)\}_{i\in[\ell]}$. Send $\{\zeta_i\}_{i\in[\ell]}$.

Opening Phase:

$C \longrightarrow R$: Send $(b, dec) \leftarrow \mathsf{Open}(G, \mathsf{TDCom}(G, b, r), b, r)$.

R: $c \leftarrow \{\mathsf{Dec}_N(\zeta_i, q)\}_{i\in[\ell]}$. Accept iff (b, dec) is a valid opening of c.

Theorem 1. *Assuming the standard hardness of factoring,* $\mathsf{Protocol}_{soa}$ *is a commitment scheme that satisfies the following properties:*

1. *(T, ϵ)-security under selective opening attacks.*
2. *Full security under selective opening attacks with a quasi-polynomial simulator.*

Proof. Note that the second property follows directly from the first property and the second part of Lemma 4. Here we just prove the first property.

Computational Binding Property. Suppose that there is a malicious adversary C^* that can open a random commitment to two different values with noticeable probability δ. We construct an efficient algorithm Factor, which uses C^* as a subroutine, to break the factoring assumption.

Factor plays the role of the honest receiver R, except that it doesn't check if a decommitment is consistent with the plaintext c encrypted in the ciphertexts received in the committing phase. More specifically, given a Blum integer N as input, Factor transforms it into a graph G, and sends (N, G) to C^* as its first message; upon receiving C^*'s committing phase message and two different decommitments (b, dec) and (b', dec') (with $b \neq b'$), Factor applies the standard extractor to these decommitments, and if it extracts a prime factor q of N, outputs it.

Note that a successful opening in a real interaction implies at least that the decommitment received by R is a valid opening of the plaintext c encrypted by C^* in the committing phase. That means, in case C^* successfully opens a commitment to two different decommitments (b, dec) and (b', dec') in the real world, one can alway extract a prime factor of N from only the two decommitments (without the need for knowledge of the plaintext c). Thus, the above algorithm Factor will output a prime factor of N with probability δ, breaking the factoring assumption.

(T, ϵ)-Security Under Selective Opening. Our simulation strategy for a k-parallel selective opening attacker R^* is quite simple in spirit. When receiving the first k integers $N_1, N_2, ..., N_k$, the simulator applies the nearly optimal extractor against T-size circuits and tries to extract a prime factor for each N_i, if it succeeds for some N_i, then the i-th commitment becomes equivocal and can be opened to different values; if it fails for N_i, then, in the eye of a T-size distinguisher, the i-th commitment is also "equivocal", since it is unable to extract a secret key of N_i either, and hence unable to tell whether the commitment c determined by the decommitment (b', dec') received is the very plaintext encrypted in the ciphertexts.

To give a formal description of the simulator, we introduce the following notations. (In what follows, we ignore the function f considered in Sect. 3 and 4.)

- $\{Y_n\}_{n\in\mathbb{N}}$: the distribution ensemble of the randomnesses for the k-parallel selective opening receiver R^*.
- Algorithm Samp is defined to be the committing phase of R^*: $y \leftarrow Y_n$, $\{N_i, G_i\}_{i\in[k]} \leftarrow R^*(y)$, output $\{N_i\}_{i\in[k]}$.
- $(T', \delta) := ((kT_c + T), \frac{\epsilon}{k\ell})$. Here T_c and T denote the size of the committer C and the distinguisher D_n respectively. ϵ is the advantage of the distinguisher that we tolerate. Note that our goal is to show that an arbitrary circuit of size T cannot distinguish a simulation from the real interaction with advantage greater than ϵ.

For the above sampling algorithm Samp, Lemma 4 guarantees that there exists a nearly optimal $(T', \delta = \frac{\epsilon}{k\ell})$-extractor $\mathsf{Ext} := \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ against any plaintext-extractor of size T'. Let $\mathcal{B}$ be a k-bit message distribution.

Consider the following distribution SIM generated by Sim.

SIM:

1. $y \leftarrow Y_n$; $\{N_i, G_i\}_{i\in[k]} \leftarrow R^*$; $\bar{b} = b_1||b_2|| \cdots ||b_k \leftarrow \mathcal{B}$;
2. Sim runs $\mathsf{Ext}_n(\{N_i\}_{i\in[k]}, y)$ and obtains $\{q_i\}_{i\in[k]}$.
3. Sim computes k commitments to 0 independently, $c^i \leftarrow \mathsf{TDcom}(G_i, 0, r_i)$, $1 \le i \le k$, $\zeta^i \leftarrow \{\mathsf{Enc}_{N_i}(c^i_j)\}_{j\in[\ell]}$, and sends $\{\zeta^i\}_{i\in[k]}$ to R^*.

4. Upon receiving $I \leftarrow R^*(\{\zeta^i\}_{i\in[k]})$ and $\{b_i\}_{i\in I}$, Sim opens $\{\zeta^i\}_{i\in I}$ in the following way:

   (a) If $b_i = 0$, open ζ^i to $(b_i = 0, dec_i)$ in an honest way;
   (b) If $q_i | N_i$ and $b_i = 1$, run $\mathsf{Fakeopen}(G_i, H_i, c^i, 0, r_i)$ to open ζ^i to $(b_i = 1, dec_i)$, where H_i is a simple cycle of G_i, transformed from (N_i, q_i);
   (c) If $q_i \nmid N_i$ and $b_i = 1$, compute a commitment $\tilde{c}^i \leftarrow \mathsf{TDcom}(G_i, 1, \tilde{r}_i)$ to 1, and set the opening of ζ^i to be the decommitment $(1, dec_i)$ of $\tilde{c}^i$.
5. Run $Out_{\mathsf{Sim}} \leftarrow R^*(\{(b_i, dec_i))\}_{i\in I})$, and output $(\bar{b}, I, Out_{\mathsf{Sim}})$.

We use hybrid argument to prove that SIM is indistinguishable from the real interaction between R^* and C^k. Consider the following sequence of hybrid experiments, in each of which we allow Sim to take the message $\bar{b}$ as an auxiliary input.

Define SIM^0 be identical to SIM. For $1 \le m \le k$, SIM^m acts in the same way as SIM^{m-1} except that Sim in SIM^m computes the m-th commitment c^m to b_m in step 3 and opens it honestly in step 4 when $m \in I$.

Note that SIM^k is identical to the real interaction. To conclude the proof of Theorem 1, it remains to show that, for every distinguisher D_n of size T, for all $1 \le m \le k$,

$$|\Pr[\mathsf{D}_n(\mathrm{SIM}^{m-1}) = 1] - \Pr[\mathsf{D}_n(\mathrm{SIM}^m) = 1]| < \frac{\epsilon}{k}. \tag{4}$$

We now construct a sequence of sub-hybrids to establish the inequality (4). Fix an $m \in [k]$. For $0 \le t \le \ell$, consider the hybrid SIM^m_t:

SIM^m_t:

1. Run step 1 and 2 of SIM and obtain $\bar{b}$, $\{N_i, G_i\}_{i\in[k]}$ and $\{q_i\}_{i\in[k]}$.
2. On input $\bar{b}$, Sim runs TDcom and generates the first $m-1$ commitments to $b_1, b_2, ..., b_{m-1}$, and the last $k-m-1$ commitments to 0, and then encrypts these commitments bit-wise and obtains $\{\zeta^i\}_{i\in[k]\setminus m}$. Sim computes the m-th commitment in the following way:

   (a) If $q_m | N_m$ or $b_m = 0$, Sim computes a commitment c^m to 0 and generates ζ^m correspondingly.
   (b) If $q_m \nmid N_m$ and $b_m = 1$, it computes a commitment c^m to 0 and a commitment $\tilde{c}^m$ to 1, and the bit-wise encryptions ζ^m of $\hat{c}^m = c^m_1 || \cdots || c^m_t || \tilde{c}^m_{t+1} || \cdots || \tilde{c}^m_\ell$, where c^m_j and $\tilde{c}^m_j$ are the j-th bit of c^m and $\tilde{c}^m$ respectively.

   Sim sends $\{\zeta^i\}_{i\in[k]}$ to R^*.
3. Upon receiving $I \leftarrow R^*(\{\zeta^i\}_{i\in[k]})$, Sim does the following: for $i \in [m-1] \cap I$, open ζ^i in an honest way; for $i \in [m+1, k] \cap I$, open ζ^i according to the step 4 of SIM; for $i = m \in I$, Sim opens ζ^i according to the step 4 of SIM except that, in the case of $q_m \nmid N_m$ and $b_m = 1$, it sets the opening of ζ^m to be the decommitment of $\tilde{c}^m$ (already computed in the previous step).

4. Run $Out_{\mathsf{Sim}} \leftarrow R^*(\{(b_i, dec_i)\}_{i\in I})$, and output $(\bar{b}, I, Out_{\mathsf{Sim}})$.

Observe that when $t = 0$, SIM_0^m computes the commitment c^m to 0 in case $q_m \nmid N_m$ and $b_m = 1$, and sets its opening to be the decommitment of an independent commitment $\tilde{c}^m$ to 1. That is, SIM_0^m acts exactly in the same way as SIM^{m-1}. We conclude the inequality (4) (and the Theorem 1) by the following two lemmas.

Lemma 5. *$SIM_\ell^m \overset{c}{\approx} SIM^m$.*

Lemma 6. *For all $1 \le t \le \ell$, and for all distingshuier D_n of size T,*

$$|\Pr[\mathsf{D}_n(SIM_{t-1}^m) = 1] - \Pr[\mathsf{D}_n(SIM_t^m) = 1]| < \frac{\epsilon}{k\ell}.$$

Due to space limitations, the proof of these two lemmas are provided in the full verison of this paper. □

6 Concurrent (T,ϵ)-Zero Knowledge and Witness Hiding in the BPK Model

In this section we present a very simple three-round concurrent (T, ϵ)-zero knowledge and witness hiding argument for NP in the BPK model. The construction relies on the *polynomial hardness* of factoring, and makes use of only two simple building blocks: the factoring-based encryption and the three round parallel version of Blum's protocol $(P_{\mathrm{B}}, V_{\mathrm{B}})$. Let a transcript of $(P_{\mathrm{B}}, V_{\mathrm{B}})$ be of the form (a, e, z), and P_{B}^1 and P_{B}^2 be the first and the second prover steps respectively.

In the key registration phase, an honest verifier generates two Blum integers N_0 and N_1 of length n, and stores *two* prime factors q_0 and q_1, $q_i|N_i$ for each $i \in \{0, 1\}$. It registers (N_0, N_1) as his public-key. In the proof phase, on input the verifier's public key (N_0, N_1) and the statement $x \in L$, the prover and the verifier execute $(P_{\mathrm{B}}, V_{\mathrm{B}})$ in which P_{B} proves the statement "$x \in L$ OR $\exists q$ *s.t.* $q|N_0$ or $q|N_1$". Denote such a prover by $P_{\mathrm{B}}(x \vee N_0 \vee N_1))$.

The formal description of our protocol follows.

$\mathsf{Protocol}_{czk}$:

Common input: $x \in R_L$, (N_0, N_1).
Private input to P: w s.t. $(x, w) \in R_L$.

$P \longrightarrow V$: Send $a \leftarrow P_{\mathrm{B}}^1(x \vee N_0 \vee N_1))$.

$V \longrightarrow P$: Send $e \leftarrow V_{\mathrm{B}}$.

$P \longrightarrow V$: $z = z_1||z_2||\cdots||z_\ell \leftarrow P_{\mathrm{B}}^2(x \vee N_0 \vee N_1)$, $\{\zeta_{i,j} \leftarrow \mathsf{Enc}_{N_i}(z_j)\}_{i\in\{0,1\}j\in[\ell]}$. Send $\{\zeta_{0,j}\}_{j\in[\ell]}$ and $\{\zeta_{1,j}\}_{j\in[\ell]}$.

V: $\hat{z} \leftarrow \{\mathsf{Dec}_N(\zeta_{0,j}, q_0)\}_{j\in[\ell]}$, $\tilde{z} \leftarrow \{\mathsf{Dec}_N(\zeta_{1,j}, q_1)\}_{j\in[\ell]}$. Accept iff $\hat{z} = \tilde{z}$ and $(a, e, \hat{z})$ is accepting.

Theorem 2. *Under the standard hardness assumption of factoring, $\mathsf{Protocol}_{czk}$ is an argument that satisfies the following properties:*

1. *Concurrent (T, ϵ)-zero knowledge with concurrent soundness.*
2. *Concurrent witness hiding.*
3. *Concurrent zero knowledge with quasi-polynomial time simulator.*

Proof. **Completeness** is obvious.

Concurrent Soundness. Suppose, towards a contradiction, that a cheating concurrent prover P^* initiates k sessions and makes the verifier accept a false statement $x \notin L$ with noticeable probability δ in one session. We can then construct an efficient algorithm Factor using P^* as a subroutine to factor a randomly chosen Blum integer with noticeable probability. Factor takes a Blum integer N as input, chooses two primes p, q ($\equiv 3 \bmod 4$) and a random $i \in \{0, 1\}$, sets N_i to be pq, N_{1-i} to be N. In the key registration phase, Factor registers (N_0, N_1) as his public key and keeps q as its secret key. In the proof phase, Factor chooses a random session, and try to obtain two accepting accpeting transcripts (a, e, z) and (a, e', z') and compute a witness q'(i.e., a prime factor of N_0 or N_1) from them.

It is not hard to show that q' is a prime factor of N_{1-i} with high probabiity, and this contradicts the hardness of factoring. The actual proof can be done by combining the standard analysis with a crucial observation, as mentioned in the introduction, that a successful cheating on session s means it will pass an *honest* verifier's check, which in turn implies that at least the both collections of ciphertexts in the last message can be decrypted to the same accepting z.

Concurrent (T, ϵ)-Zero Knowledge. Consider an arbitrary concurrent adversary V^* of polynomial size. We show there *exists* a simulator of polynomial size to establish the weak zero knowledge property.

Suppose that V^* registers k public keys $\{(N_0^i, N_1^i)\}_{i\in[k]}$ and initiates at most s sessions. As before, the simulator applies the nearly optimal extractor to factor all integers registered by V^* in the key registration phase. Once the simulator extracts a prime factor of one of (N_0^i, N_1^i), it can complete any session under the public key (N_0^i, N_1^i) successfully; if it fails for a public key (N_0^i, N_1^i), the simulator computes encryptions of zeros as its last message in the sessions under the public key (N_0^i, N_1^i).

Let Y_n be the distribution of V^*'s randomness, and the sampling algorithm Samp to be the V^*'s registration step. Set (T', δ) to be $((s(2\ell T_{enc} + T_p) + T), \frac{\epsilon}{4s\ell})$, where T_{enc}, T_p and T are the size of Enc, the honest prover of the Blum protocol (P_B, V_B) and the distinguisher respectively, and ϵ is the advantage of the distinguisher that we tolerate. By Lemma 4 we have a polynomial-size $(T', \delta = \frac{\epsilon}{4\ell})$-extractor $\mathsf{Ext} := \{\mathsf{Ext}_n\}_{n\in\mathbb{N}}$ against any circuit family of size T'.

On input s Yes instances $\bar{x} = \{x_i\}_{i\in[s]}$, the simulator proceeds as follows.

$\mathsf{Sim}(\bar{x})$:

1. $y \leftarrow Y_n$, $\{(N_0^i, N_1^i)\}_{i\in[k]} \leftarrow V^*(y)$.
2. $\{(q_0^i, q_1^i)\}_{i\in[k]} \leftarrow \mathsf{Ext}_n(\{(N_0^i, N_1^i)\}_{i\in[k]}, y)$.

3. For a session under the public key (N_0^i, N_1^i), do the following:
   (a) If $q_0^i | N_0^i$ or $q_1^i | N_1^i$, complete this session using the extracted prime factor as witness.
   (b) Otherwise, produce an honest message a in its first step. Upon receiving a challenge e, set $z = 0^\ell$, and compute $\{\mathsf{Enc}_{N_0^i}(z_j)\}_{i\in[\ell]}$ and $\{\mathsf{Enc}_{N_1^i}(z_j)\}_{i\in[\ell]}$ as the last message of this session.
4. Output the entire history when V^* terminates.

We are ready to prove the first part of Theorem 2. Suppose, towards a contradiction, that there exists a distinguisher D_n of size T such that

$$|\Pr[\mathsf{D}_n(\mathsf{View}^P_{V^*})(\bar{x})) = 1] - \Pr[\mathsf{D}_n(\mathsf{Sim}(\bar{x})) = 1]| > \epsilon. \tag{5}$$

We order all s sessions according to its appearance, and construct the following hybrid simulators with all witnesses hardwired: Define $\mathsf{Sim}^0(\bar{x}, \bar{w})$ be the $\mathsf{Sim}(\bar{x}, \bar{w})$, and $\mathsf{Sim}^k(\bar{x}, \bar{w})$ as in the same way except that in each of the first k sessions it uses the real witness to complete a proof. Clearly, $\mathsf{Sim}^s(\bar{x}, \bar{w})$ is identical to the real interaction. From (5), there must exist a $m \in [s]$ such that

$$|\Pr[\mathsf{D}_n(\mathsf{Sim}^{m-1}(\bar{x}, \bar{w})) = 1] - \Pr[\mathsf{D}_n(\mathsf{Sim}^m(\bar{x}, \bar{w})) = 1]| > \frac{\epsilon}{s}. \tag{6}$$

Fix such a m, and for $t \in [2\ell]$, consider the sub-hybrid simulator $\mathsf{Sim}^m_t(\bar{x}, \bar{w})$:

$\mathsf{Sim}^m_t(\bar{x}, \bar{w})$:
1. Run step 1,2 of $\mathsf{Sim}^m(\bar{x}, \bar{w})$ and obtain $\{(N_0^i, N_1^i)\}_{i\in[k]}$ and $\{(q_0^i, q_1^i)\}_{i\in[k]}$.
2. For the session m under the public key (N_0^m, N_1^m), do the following:
   (a) If $q_0^i | N_0^i$ or $q_1^i | N_1^i$, act in the same way as $\mathsf{Sim}^m(\bar{x})$.
   (b) Otherwise, produce an honest message a in its first step. Upon receiving a challenge e, produce an accepting z using the real witness, set $z' = 0^t || z^{2\ell-t}$, where $z^{2\ell-t}$ is the suffix of $z||z$, and encrypt the first half bits of z' under N_0^i, their second half bits under N_1^i.

   For any other session, act in the same way as $\mathsf{Sim}^{m-1}(\bar{x}, \bar{w})$.
3. Output the entire history when V^* terminates.

Observe that $\mathsf{Sim}^m_{2\ell}(\bar{x}, \bar{w}) = \mathsf{Sim}^m(\bar{x}, \bar{w})$. It follows from the witness indistinguishability of the Blum protocol that $\mathsf{Sim}^m_0(\bar{x}, \bar{w}) \overset{c}{\approx} \mathsf{Sim}^{m-1}(\bar{x}, \bar{w})$ (with a negligible distinguishing gap). By (6), there must exist a $t \in [2\ell]$ such that

$$|\Pr[\mathsf{D}_n(\mathsf{Sim}^m_{t-1}(\bar{x})) = 1] - \Pr[\mathsf{D}_n(\mathsf{Sim}^m_t(\bar{x})) = 1]| > \frac{\epsilon}{4s\ell}. \tag{7}$$

Note that the only difference between $\mathsf{Sim}^m_{t-1}(\bar{x})$ and $\mathsf{Sim}^m_t(\bar{x})$ lies in the t-th ciphertext in case that the extractor fails to find any prime factors of the public key. Hence, if the inequality (7) holds, we can construct a size-T' circuit A_n with $(barx, \bar{w})$ hardwired, and show that it constradicts the (nearly) optimality of the extractor Ext_n. (The detailed proof can be found in the full version of this work.) This concludes the first part of Theorem 2.

The second part of Theorem 2 follows from the fact that (concurrent) (T, ϵ)-zero knowledge implies (concurrent) witness hiding (see [JKKR17] for the detailed proof). Here we just describe the underlying idea. For a given malicious verifier V^* of size T that can output a witness of a statement drawn from X_n at the end of a session with probability greater than some inverse polynomial ϵ, as we showed above, there exists a simulator of polynomial size such that V^* cannot distinguish the real interaction from simulation with probability greater than $\frac{\epsilon}{2}$. Combining the simulator and V^*, we will have a circuit family of polynomial size that breaks the hardness of X_n. Quasi-polynomial simulatability follows again from the second part of Lemma 4 directly. □

7 Simpler (T,ϵ)-Zero Knowledge and Analysis in the Plain Model

In this section we present a very simple delayed-input 2-round (T, ϵ)-zero knowledge argument for NP, and then sketch how to use our individual simulation technique to give a significantly simpler proof that the distinguisher-dependent simulatable zero knowledge protocols of [JKKR17, BKP19] also satisfy the stronger notion of (T, ϵ)-zero knowledge.

We build such an argument on a quasi-polynomial extractable perfectly binding commitment scheme Com [Pas03] (which can be based on sub-exponential hardness of factoring) and a NIWI proof system $(P_{\text{WI}}, V_{\text{WI}})$[10].

As usual, we denote by $P_{\text{WI}}(x \vee (N, c))$ the prover of the NIWI proof that proves to the verifier the statement "$x \in L$ OR $\exists\ q$ such that c is a commitment to q and $q|N$"

$\mathsf{Protocol}_{zk}$:

Private input to P: w s.t. $(x, w) \in R_L$.

$V \longrightarrow P$: $(N, q) \leftarrow \mathsf{Blum}(1^n)$. Send N to P.

$P \longrightarrow V$: $c \leftarrow \mathsf{Com}(0^n)$, $z = z_1 || z_2 || \cdots || z_\ell \leftarrow P_{\text{WI}}(x \vee (N,c)), \{\zeta_j \leftarrow \mathsf{Enc}_N(z_j)\}_{j\in[\ell]}$. Send x, c and $\{\zeta_j\}_{j\in[\ell]}$ to V.

V: $z \leftarrow \{\mathsf{Dec}_N(\zeta_j, q)\}_{j\in[\ell]}$. Accept iff (x, z) is accepting.

Theorem 3. *Under the sub-exponential hardness assumption of factoring, $\mathsf{Protocol}_{zk}$ is a delayed-input interactive argument that satisfies all the following properties:*

1. *Delayed-input (T, ϵ)-zero knowledge.*
2. *Delayed-input witness hiding.*
3. *Delayed-input zero knowledge with quasi-polynomial time simulator.*

[10] One can also use two-round WI (such as [DN00]) here. We use NIWI (such as [GOS06]) to simplify our construction.

The soundness of this protocol is also straightforward. Note that a cheating prover P^* on a false statement $x \notin L$ with noticeable success probability δ implies that the message c sent by P^* is a commitment to a prime factor of N. This leads to a simple quasi-polynomial factoring algorithm Factor with success probability at least δ that contradicts the sub-exponential hardness of factoring: On input an integer N, it plays the role of the verifier and sends it to P^*; upon receiving the message c, it extracts a prime factor of N from c in quasi-polynomial time.

The proof of (T, ϵ)-zero knowledge, witness hiding and quasi-polynomial simulatability are essentially the same as in the previous section, we omit it here.

Upgrade the Distinguisher-Dependent Simulations. The work of [CLP15b] implies that existing distinguisher-dependent simulatable weak zero knowledge protocols of [JKKR17,BKP19] are also (T, ϵ)-zero knowledge. We note that both constructions of [JKKR17,BKP19] enjoy the two properties of (A, B) listed in Sect. 1.4, hence our individual simulation technique can also be applied to prove that they satisfy the stronger notion of (T, ϵ)-zero knowledge. For their 3-round protocols, one can view the verifier step as an NP instance (to which a solution will enable a successful simulation) sampler that takes as input its randomness and the first prover message a and outputs an instance (verifier message). To show the (T, ϵ)-zero knowledge property, we can construct an individual simulator in a similar way. The simulator applies a nearly optimal extractor (which is also given certain secret information $f(a)$ about the message a as an additional input[11]) to the sampler/verifier and tries to extract the corresponding witness, and then follows the residual strategy of the distinguisher-dependent simulator in [JKKR17,BKP19] after their extraction from the distinguisher oracle.

Acknowledgments. We would like to thank Takahiro Matsuda, Xinxuan Zhang and anonymous reviewers from Asiacrypt'20 and Crypto'19 for pointing out two errors in earlier versions of this paper, and for their valuable suggestions. We are supported by PlatON, the National Natural Science Foundation of China (Grant No. 61932019, No. 61772521 and No. 61772522), Key Research Program of Frontier Sciences, CAS (Grant No. QYZDB-SSW-SYS035).

References

[ACGS88] Alexi, W., Chor, B., Goldreich, O., Schnorr, C.-P.: RSA and rabin functions: certain parts are as hard as the whole. SIAM J. Comput. **17**(2), 194–209 (1988)

[11] This is in contrast to our settings, where the hard instances generated by the adversary depend only on its randomness. When these instances depend also on the first prover message a, the nearly optimal extractor usually needs to take as input some secret information about a, since in a proof by contradiction like ours, those algorithms D_n and A_n need these information to go through.

[AIR01] Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_8

[APV05] Alwen, J., Persiano, G., Visconti, I.: Impossibility and feasibility results for zero knowledge with public keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 135–151. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_9

[Bar01] Barak, B.: How to go beyond the black-box simulation barrier. In: Proceedings of the 42th Annual IEEE Symposium on Foundations of Computer Science - FOCS 2001, pp. 106–115. IEEE Computer Society (2001)

[BCC88] Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. **37**(2), 156–189 (1988)

[BGI+17] Badrinarayanan, S., Garg, S., Ishai, Y., Sahai, A., Wadia, A.: Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10626, pp. 275–303. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70700-6_10

[BHY09] Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_1

[BKP19] Bitansky, N., Khurana, D., Paneth, O.: Weak zero-knowledge beyond the black-box barrier. In: Annual ACM Symposium on the Theory of Computing - STOC 2019, pp. 1091–1102. ACM Press (2019)

[BP15] Bitansky, N., Paneth, O.: On non-black-box simulation and the impossibility of approximate obfuscation. SIAM J. Comput. **44**(5), 1325–1383 (2015)

[BP12] Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 190–208. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_11

[CCH+19] Canetti, R., et al.: Fiat-Shamir: from practice to theory. In: Annual ACM Symposium on the Theory of Computing - STOC 2019, pp. 1082–1090. ACM Press (2019)

[CGGM00] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero knowledge. In: Proceedings of the 32rd Annual ACM Symposium Theory of Computing- STOC 2000, pp. 235–244. ACM press (2000)

[CKPR01] Canetti, R., Kilian, J., Petrank, E., Rosen, A.: Black-box concurrent zero-knowledge requires omega(log n) rounds. In: Proceedings of the 33rd Annual ACM Symposium Theory of Computing- STOC 2001, pp. 570–579. ACM press (2001)

[CLP13] Chung, K.-M., Lin, H., Pass, R.: Constant-round concurrent zero knowledge from p-certificates. In: Proceedings of the 54th Annual Symposium on Foundations of Computer Science - FOCS 2013, pp. 50–59. IEEE Computer Society (2013)

[CLP15a] Chung, K.-M., Lin, H., Pass, R.: Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 287–307. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_14

[CLP15b] Chung, K.-M., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 66–92. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_4

[Den17] Deng, Y.: Magic adversaries versus individual reduction: science wins either way. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10211, pp. 351–377. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_12

[DFG+11] Deng, Y., Feng, D., Goyal, V., Lin, D., Sahai, A., Yung, M.: Resettable cryptography in constant rounds – the case of zero knowledge. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 390–406. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_21

[DGL+16] Deng, Y., Garay, J., Ling, S., Wang, H., Yung, M.: On the implausibility of constant-round public-coin zero-knowledge proofs. In: Zikas, V., De Prisco, R. (eds.) SCN 2016. LNCS, vol. 9841, pp. 237–253. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44618-9_13

[DGS09] Deng, Y., Goyal, V., Sahai, A.: Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In: Proceedings of the 50th Annual Symposium on Foundations of Computer Science - FOCS 2009, pp. 251–260. IEEE Computer Society (2009)

[DK18] Deshpande, A., Kalai, Y.: Proofs of ignorance and applications to 2-message witness hiding. Cryptology ePrint Archive, Report 2018/896 (2018)

[DN00] Dwork, C., Naor, M.: Zaps and their applications. In: Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science - FOCS 2000, pp. 283–293. IEEE Computer Society (2000)

[DNRS03] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. J. ACM **50**(6), 852–921 (2003)

[DNS98] Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. In: Proceedings of the 30rd Annual ACM Symposium Theory of Computing - STOC 1998, pp. 409–418. ACM press (1998)

[FGJ18] Fleischhacker, N., Goyal, V., Jain, A.: On the existence of three round zero-knowledge proofs. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10822, pp. 3–33. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_1

[FKP19] Freitag, C., Komargodski, I., Pass, R.: Non-uniformly sound certificates with applications to concurrent zero-knowledge. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 98–127. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_4

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

[FS89] Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 526–544. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_46

[FS90] Feige, U., Shamir, A.: Witness indistinguishability and witness hiding protocols. In: Proceedings of the 22rd Annual ACM Symposium Theory of Computing- STOC 1990, pp. 416–426. ACM press (1990)

[GK96] Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. **25**(1), 169–192 (1996)

[GL89] Goldreich, O., Levin, L.: A hard-core predicate for all one-way functions. In: Proceedings of the 21th Annual ACM Symposium on the Theory of Computing - STOC 1989, pp. 25–32. ACM Press (1989)

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

[Gol01] Goldreich, O.: Foundations of Cryptography, Volume Basic Tools. Cambridge University Press, Cambridge (2001)

[GOS06] Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 97–111. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_6

[IR89] Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proceedings of the 21th Annual ACM Symposium on the Theory of Computing - STOC 1989, pp. 44–61. ACM Press (1989)

[JKKR17] Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 158–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_6

[KP01] Kilian, J., Petrank, E.: Concurrent and resettable zero-knowledge in poly-loalgorithm rounds. In: Proceedings of the 33rd Annual ACM Symposium Theory of Computing- STOC 2001, pp. 560–569. ACM press (2001)

[MR01] Micali, S., Reyzin, L.: Soundness in the public-key model. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 542–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_32

[ORSV13] Ostrovsky, R., Rao, V., Scafuro, A., Visconti, I.: Revisiting lower and upper bounds for selective decommitments. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 559–578. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_31

[PA17] Ananth, P., Jain, A.: On secure two-party computation in three rounds. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 612–644. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_21

[Pas03] Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_10

[Pas11] Pass, R.: Limits of provable security from standard assumptions. In: Proceedings of the 45rd Annual ACM Symposium Theory of Computing-STOC 2011, pp. 109–118. ACM press (2011)

[PRS02] Prabhakaran, M., Rosen, A., Sahai, A.: Concurrent zero knowledge with logarithmic round-complexity. In: Proceedings of the 43th Annual IEEE Symposium on Foundations of Computer Science - FOCS 2002, pp. 366–375. IEEE Computer Society (2002)

[RK99] Richardson, R., Kilian, J.: On the concurrent composition of zero-knowledge proofs. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 415–431. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_29

[SV12] Scafuro, A., Visconti, I.: On round-optimal zero knowledge in the bare public-key model. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 153–171. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_11

[TW87] Tompa, M., Woll, H.: Random self-reducibility and zero knowledge interactive proofs of possession of information. In: Proceedings of the 28th Annual Symposium on Foundations of Computer Science - FOCS 1987, pp. 472–482. IEEE Computer Society (1987)

[Xia11] Xiao, D.: (Nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 541–558. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_33

[Xia13] Xiao, D.: Errata to (nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 721–722. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_40

[YZ07] Yung, M., Zhao, Y.: Generic and practical resettable zero-knowledge in the bare public-key model. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 129–147. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_8