Precise Time and Space Simulatable Zero-Knowledge

Ning Ding, Dawu Gu
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, 200240, China
cs.dingning@gmail.com, dwgu@sjtu.edu.cn

Abstract

Traditionally, the definition of zero-knowledge states that an interactive proof of $x \in L$ provides zero (additional) knowledge if the view of any *polynomial-time* verifier can be reconstructed by a *polynomial-time* simulator. Since this definition only requires that the worst-case running-time of the verifier and simulator are polynomials, zero-knowledge becomes a worst-case notion.

In STOC'06, Micali and Pass proposed a new notion of precise zero-knowledge, which captures the idea that the view of any verifier in every interaction can be reconstructed in (almost) the same time (i.e., the view can be "indistinguishably reconstructed"). This is the strongest notion among the known works towards precislization of the definition of zero-knowledge.

However, as we know, there are two kinds of computational resources (i.e. time and space) that every algorithm consumes in computation. Although the view of a verifier in the interaction of a precise zero-knowledge protocol can be reconstructed in almost the same time, the simulator may run in very large space while at the same time the verifier only runs in very small space. In this case it is still doubtful to take indifference for the verifier to take part in the interaction or to run the simulator. Thus the notion of precise zero-knowledge may be still insufficient. This shows that precislization of the definition of zero-knowledge needs further investigation.

In this paper, we propose a new notion of precise time and space simulatable zero-knowledge (PTSSZK), which captures the idea that the view of any verifier in each interaction can be reconstructed *not only* in the same time, *but also* in the same space. We construct the first PTSSZK proofs and arguments with simultaneous linear time and linear space precisions for all languages in **NP**. Our protocols do not use noticeably more rounds than the known precise zero-knowledge protocols, and the probability analysis of the successful extraction of the new simulation strategy may be of independent interests.

Keywords: Zero-Knowledge, Precise Zero-Knowledge, Proofs of Knowledge, Interactive Proofs and Arguments

Contents

List of Tables

1 Introduction

Zero-knowledge proofs were introduced by Goldwasser, Micali and Rackoff [14]. Their definition essentially states that an interactive proof of $x \in L$ provides zero (additional) knowledge if, for any efficient verifier V^*, the view of V^* in the interaction can be "indistinguishably reconstructed" by an efficient simulator S -interacting with no one- on just input x. Since efficiency is formalized as polynomial-time, a worst-case notion, zero-knowledge too automatically becomes a worst-case notion. The refinement of [11] (Sec. 4.4.4.2) calls for a tighter coupling between the expected running-time of V^* and that of S: a proof is zero-knowledge with tightness $t(\cdot)$ if there exists a fixed polynomial $p(\cdot)$ such that the expected running-time of $S(x)$ is upper-bounded by $t(|x|)$ times the expected running-time of $V^*(x)$ plus $p(|x|)$.

Micali and Pass [15] argued, however, that such coupling may still be insufficient, even when the tightness function is a constant and the polynomial $p(\cdot)$ is identically 0. Consider a family of malicious verifiers $\{V_i^*\}_{i\in\mathbb{N}}$, in which on input an instance $x \in \{0,1\}^n$, V_i^* takes n^{10i} computational steps with probability $\frac{1}{n}$, and n steps in the rest of the time. Since the expected running-time of V_i^* is $\Theta(n^{10i-1})$, zero-knowledge with optimal tightness only requires that V_i^*'s view be simulated in expected time $\Theta(n^{10i-1})$. Assume S always takes $\Theta(n^{10i-1})$ time to reconstruct V_i^*'s view. Then in the viewpoint of [11], it is indistinguishable for V_i^* to get out and interact with the prover or to stay home and run S for granted. However, by interacting with P, V_i^* will almost always execute n steps of computation, while (in absence of extra guarantees) running the simulator might always cause him to invest n^{10i-1} steps of computation, which tends to infinite as i tends to infinite. Is the view of V_i^* "reconstructed indistinguishably"? This discussion shows that we need a stronger notion of zero-knowledge.

Hence [15] put forward a notion of precise zero-knowledge. This notion captures the idea that prover provides a zero-knowledge proof of $x \in L$ if the view v of any verifier in an interaction with the prover about x can be reconstructed in (almost) the same time. Informally, by [15] a proof system is zero-knowledge with $p(n, y)$ (time) precision if for every verifier V^*, S' running-time in outputting a view is bounded by $p(n, \mathsf{T})$ whenever V^*'s running-time on this view is T. Following this notion, [15] constructed $\omega(\log n)$-round (resp. $\omega(1)$-round) (non-black-box) zero-knowledge proofs and arguments with linear (resp. polynomial) precision for all languages in **NP**.

Further, [15] showed there do not exist black-box zero-knowledge protocols with polynomial precision for languages outside **BPP**, and [19] showed the simulators of Barak's non-black-box zero-knowledge arguments [1] cannot provide polynomial precision. To achieve precise time simulation, [15][19] developed a method, called the "cut-off" technique. That is, the simulator S still needs to rewind a verifier V^* to extract secret information, but in the first run it records V^*'s running-time (steps) and then in the second run it uses this time to bound V^*'s computing, i.e., S emulates V^* for at most such time (steps). If V^* needs more time then S terminates its computing. It can be seen this simulation strategy uses verifiers in a non-black-box way.

However, as we know, there are two kinds of computational resources (i.e., time and space) that every algorithm consumes in computation. Although the notion of precise zero-knowledge is quite strong, it investigates precise simulation only with respect to the *running-time* of the simulator and verifier, regardless of their *running-space*. Consider a family of malicious verifiers $\{V_i^*\}_{i\in\mathbb{N}}$, in which on input an instance $x \in \{0,1\}^n$ (for sufficiently large n's), V_i^* takes n^{10i} computational steps and n^c space for some $c > 0$ (note that an n^c-space machine may take at most $n2^{\Theta(n^c)}$ time, which is greater than n^{10i} for arbitrary i when n is large enough). Zero-knowledge with linear (time) precision only requires that V_i^*'s view be simulated in time $\Theta(n^{10i})$. Assume S always takes $\Theta(n^{10i})$ time, but n^i space, to reconstruct V_i^*'s view. In this case in the viewpoint of [15], it is indistinguishable for V_i^* to take part in the interaction or run the simulator. But it can be seen

that (for sufficiently large n's) V_i^*'s running-space in the interaction is n^c, while the simulator needs n^i space to reconstruct V_i^*'s view, which is greater than any predetermined polynomial in n^c as i tends to infinite. Thus, *is it really "indistinguishable" for V_i^* to interact with the prover or to run the simulator?*

The above discussion shows the notion of precise zero-knowledge may be still insufficient and precislization of the definition of zero-knowledge needs further investigation. A natural (and maybe the strongest) avenue for extending precise zero-knowledge is to require that views be reconstructed *not only* in the same time, *but also* in the same space. In this paper we will try to pursue this avenue and to construct zero-knowledge proof systems which are precisely simulatable in this sense.

1.1 Our Results

We put forward a new notion of precise time and space simulatable zero-knowledge (PTSSZK), which strengthens the notion of precise zero-knowledge [15] by additionally requiring the space used in reconstructing the view of a verifier is also almost the same as that of the verifier in the interaction. Informally, we say a proof system is zero-knowledge with time precision $p_t(n, y)$ and space precision $p_s(n, y)$ if for every verifier V^* the simulator S' running-time and running-space in reconstructing a view are always respectively bounded by $p_t(n, \mathsf{T})$ and $p_s(n, \mathsf{S})$ whenever V^*'s running-time and running-space on this view are respectively T and S.

Since V^* and S are usually required to run in polynomial-time, it is less meaningful if $p_t(n, y)$ or $p_s(n, y)$ is super-polynomial in n or y, or else it is meaningful. To the best of our knowledge, all the known zero-knowledge protocols for languages outside **BPP** cannot obtain simultaneous meaningful time and space precisions. We construct the first PTSSZK proofs and arguments with simultaneous meaningful time and space precisions for all languages in **NP**. The formal results are shown as follows.

Theorem 1.1. *Assume the existence of constant-round perfectly-hiding commitments. Then for every language $L \in$ **NP***
1. there exist $\omega(\log^3 n)$-round zero-knowledge proofs with time precision $p_t(n, y) = poly(n) + O(y)$ and space precision $p_s(n, y) = poly(n) + O(y)$ for L.
2. there exist $\omega(\log n)$-round zero-knowledge proofs with time precision $p_t(n, y) = poly(n)+poly(n)O(y)$ and space precision $p_s(n, y) = poly(n) + O(y)$ for L.

Theorem 1.2. *Assume the existence of one-way functions. Then for every language $L \in$ **NP**,*
1. there exist $\omega(\log^3 n)$-round zero-knowledge arguments with time precision $p_t(n, y) = poly(n) + O(y)$ and space precision $p_s(n, y) = poly(n) + O(y)$ for L.
2. there exist $\omega(\log n)$-round zero-knowledge arguments with time precision $p_t(n, y) = poly(n) + poly(n)O(y)$ and space precision $p_s(n, y) = poly(n) + O(y)$ for L.

Comments. We give some comments on our results.

1. We stress that our results are model-independent. (Of course, the precise quantities of the poly(n)s depend on the underlying model.) Although this work concerns the precision of simulation, we are oblivious of the details how machines work and of definitions of the complexity measures of time and space. That is, we don't need to explicitly refer to any concrete machine model and definitions of the complexity measures of the two types of resource in that model. Instead, to handle issues on the measures we simply use variables, say T and S, to denote the quantities of time and space some machine consumes in computation, measured via some unspecified measures in some unspecified model. This abstract handling makes our results hold in general models.

2. Our precisions are pure in the sense that the simulator S' running-time (resp. running-space) are only related to V^*'s running-time T (resp. V^*'s running-space S), regardless of S (resp. T).

3. In Theorem 1.1 (or Theorem 1.2), the summand $\text{poly}(n)$ in p_t depends on both L and the primitives underlying the constructions (e.g. commitments and one-way functions). The coefficient $\text{poly}(n)$ in p_t (using $\omega(\log n)$ rounds) is independent of L, but to depend on the primitives. The $\text{poly}(n)$ in p_s depends on both L and the primitives. We did not try to minimize the $\text{poly}(n)$ in p_s by letting our simulators employ subtle strategies for space recycling, as our primary interest is just to obtain a fixed polynomial. The constant coefficients in O-notation of p_t, p_s are independent of both L and the primitives.

4. We obtain simultaneous *linear* time precision (using $\omega(\log^3 n)$ rounds) and *linear* space precision. Remark 4.7 further shows if we employ the model of (interactive) Turing machines, we can obtain fully linear time precision (using $\omega(\log^3 n)$ rounds), i.e. $p_t(n, y) = O(y)$.

Our Technique. In the high level, our constructions follow the paradigm used in [19] (based on [15]). To construct precise zero-knowledge protocols, [19] first constructed some proofs (arguments) of knowledge with time precisions of extraction. Then using these proofs of knowledge as tools, [19] showed how to construct the desired precise zero-knowledge protocols with corresponding time precisions of simulation. Thus the task of constructing precise zero-knowledge protocols is reduced to the constructions of the required proofs of knowledge. Following this paradigm, we show if such proofs of knowledge possess simultaneous time and space precisions (of extraction), the zero-knowledge protocols possess corresponding time and space precisions (of simulation). Thus the main technical task in this work is to construct the proofs of knowledge with simultaneous time and space precisions.

Technically, [19] constructed the proofs of knowledge with time precisions by using the "cut-off" technique. As aforementioned, the simulators in [19] (as extractor in the proofs of knowledge) in the first run record V^*'s (as prover in the proofs of knowledge) running-time (steps) and then in the second run use this time to bound V^*'s computing, i.e., they emulate V^* for at most such time (steps) and if V^* needs more time then they terminate its computing. Since this (original) "cut-off" technique only concerns V^*'s running-time, the simulators in [19] cannot automatically provide simultaneous meaningful time and space precisions. Remark 5.2 shows an example that the two types of precision indeed cannot be obtained by using the original "cut-off" technique.

We extend the "cut-off" technique to present an improved extraction strategy and thus construct the proofs of knowledge with simultaneous meaningful time and space precisions. (This shows our simulators use verifiers in the non-black-box way.) To extract secret information our simulators not only record V^*'s running-time (steps), but also record its running-space in the first run. In the second run our simulators emulate V^* for at most such steps, times a factor, and keep track of V^*'s running-space. If V^* needs more time, or more space than the recorded space (in size) in the rewind, our simulators terminate its computing. Remark 5.3 shows the usage of the factor for the time in the rewind is necessary, or else our simulator cannot succeed in extraction for any V^*. Since there is one more necessary condition with respect to space for our simulators to succeed in extraction, we shall present more subtle probability analysis of successful extraction, which is the main technical novelty of this work.

1.2 Outline of This Paper

The rest of the paper is arranged as follows. In Section 2 we present the preliminaries throughout this paper, which contains probabilistic notations and definitions and known constructions of basic cryptographic primitives. In Section 3, we present the new notion of PTSSZK. In Section 4 we

present the high-level proofs of our results assuming the existence of the proofs of knowledge with simultaneous time and space precisions. In Section 5, we present the constructions of the required proofs of knowledge and complete the entire proofs.

2 Preliminaries

2.1 Basic Notations

We use $[1, m]$ to denote all integers in $\{1, 2, \cdots, m\}$, and use symbol "$\circ$" to denote concatenation operation of two strings and use $|s|$ to denote the bit-length of string s.

Set Notations. Let A and B be two sets. We use A, B (or AB) to denote the intersection of A and B, $A+B$ to denote the union of A and B if they are disjoint, $A-B$ to denote the difference of A and B, i.e., the set of points that belong to A but not to B, $\bar{A}$ to denote the complement of A.

Probabilistic Notations. We follow the standard notations in probability theory. Let $\Pr[A, B]$ (resp. $\Pr[A + B], \Pr[A - B], \Pr[\bar{A}]$) denote the probability of the intersection of A and B (resp. $A + B$, $A - B$, $\bar{A}$), $\Pr[A|B]$ denote the conditional probability of A on the occurrence of B. A decomposition of the sample space (resp. A) is a set of $\{D_1, \cdots, D_d\}$ where d is a natural number, satisfying that $D_i \cap D_j = \phi$ for any $1 \leq i \neq j \leq d$ and $\sum_{i=1}^{d} D_i$ equals the sample space (resp. $\sum_{i=1}^{d} D_i = A$).

Definition 2.1. A function $\mu(\cdot)$, where $\mu : \mathbb{N} \to [0, 1]$, is called *negligible* if $\mu(n) = n^{-\omega(1)}$ (i.e., $\mu(n) < \frac{1}{p(n)}$ for all polynomial $p(\cdot)$ and large enough n's). We say a random event happens with *overwhelming* probability if it happens with probability $1 - \mu(n)$ for some negligible function $\mu(\cdot)$. We will sometimes use neg to denote an unspecified negligible function.

Definition 2.2. (Computational Indistinguishability) We say two probability ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable, if for every family of polynomial-sized circuits $\{C_n\}_{n\in\mathbb{N}}$ $|\Pr[C_n(X_n) = 1] - \Pr[C_n(Y_n) = 1]| = \mathsf{neg}(n)$. We will sometime abuse the notation and say that X_n and Y_n are computationally indistinguishable when each of them is a part of $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ and $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable. We will drop the index n if it can be inferred from the context. In most cases, n is the security parameter.

2.2 Commitment Schemes

Definition 2.3. (Perfectly-Binding Commitment) A (non-interactive perfectly binding computationally hiding) commitment scheme is a uniform polynomial-time computable sequence of functions $\{C_n\}_{n\in N}$ where $C_n : \{0,1\}^n \times \{0,1\}^{p(n)} \to \{0,1\}^{q(n)}$, and $p(\cdot), q(\cdot)$ are some polynomials, that satisfies:

Perfect Binding For every $x \neq x' \in \{0,1\}^n$, $C(x, \{0,1\}^{p(n)}) \cap \to C(x, \{0,1\}^{p(n)}) = \phi$.

Computational Hiding For every $x, x' \in \{0,1\}^n$, the random variables $C(x, U_n)$ and $C(x', U_n)$ are computationally indistinguishable.

A perfectly-binding commitment scheme can be constructed under the assumption that one-way permutations exist [4] (using the generic hard-core bit of [12]). Another construction, under incomparable assumptions, was given by [2]. We can also use instead the two-round scheme of Naor [16], which can be based on any one-way function.

Perfectly-Hiding Commitment Schemes. In a perfectly-hiding commitment scheme, the binding property is guaranteed to hold only with respect to a probabilistic polynomial-time sender. On the other hand, the hiding property is information-theoretic. That is, the distributions of commitments different strings are identical, and thus even an all powerful receiver cannot know the value committed to by the sender. We stress that the binding property guarantees that a cheating probabilistic polynomial-time sender can find only one decommitment, even though decommitments to different strings exist. See [11] (Sec. 4.8.2) for a full definition. Perfectly-hiding commitment schemes can be constructed from any one-way permutation [17]. Constant-round schemes are only known to exist under stronger assumptions of the existence of collision-resistant hash functions [18][7] or the existence of a collection of certified claw-free functions [11] (Sec. 4.8.2.3).

2.3 Interactive Proofs and Arguments

An interactive proof [14] is a two-party protocol, where one party is called the prover and the other party is called the verifier. We use the following definition:

Definition 2.4. An interactive protocol (P, V) is called an interactive proof system for a language L if the following conditions hold:
Efficiency: The number and total length of messages exchanged between P and V are polynomially bounded and V is a probabilistic polynomial-time machine.
Perfect completeness: If $x \in L$, then V will always accept x.
Soundness: If $x \notin L$, then for any cheating prover P^* the probability that P^* convinces V of x is $\mathsf{neg}(n)$.

Let L be a **NP** language. An interactive argument for L [6] is the following variation on the definition of an interactive proof:
1. The soundness requirement is relaxed to quantify only over prover strategies P^* that can be implemented by a polynomial-sized circuit.
2. The system is required to have an efficient prover strategy.

2.4 Zero-Knowledge

Informally, a proof or argument system for L is zero-knowledge [14] if after seeing a proof that $x \in L$, the verifier does not learn anything about x that it didn't know before. Moreover this holds even if the verifier does not follow its prescribed strategy for the proof system, as long as its strategy can be implemented by an efficient algorithm. The formal definition is below:

Definition 2.5. Let $L = L(R)$ be some language and let (P, V) be an interactive proof (argument) for L. We say (P, V) is (perfect) zero-knowledge if there exists a probabilistic polynomial-time algorithm, called simulator, such that for every polynomial-sized circuit V^* and every $(x, w) \in R$, the following two probability variables are (identically distributed) computationally indistinguishable:
1. The view of V^* in the real execution of $(P(w), V^*)(x)$.
2. The output of the simulator on input (x, V^*).

There are two classical constructions of 3-round zero-knowledge proofs for **NP** (without requiring negligible soundness error probability) which are Blum's proof for Hamilton Circuits (HC) [5] and Goldreich, Micali and Wigderson's proof for Graph 3-Coloring [13]. These two constructions use a common paradigm. That is, the proofs consist of three steps: Prover firstly sends a commitment (using a perfectly-binding commitment scheme) to verifier and then verifier responses a random challenge (For Blum's protocol, the challenge is one bit). Lastly, prover answers the challenge by sending the decommitment. Verifier accepts the proof if the decommitment is valid.

2.5 Witness Indistinguishability

Witness indistinguishability (WI) is a weaker property than zero-knowledge, introduced by [9]. In a witness indistinguishable proof system if both w_1 and w_2 are witnesses that $x \in L$, then it is infeasible for the verifier to distinguish whether the prover used w_1 or w_2 as the auxiliary input. The formal definition is below:

Definition 2.6. Let $L = L(R)$ be some language and (P, V) be a proof (argument) for L. We say that (P, V) is witness indistinguishable if for any polynomial-sized circuit family $\{V_n^*\}_{n\in\mathbb{N}}$, any x, w_1, w_2 where $(x, w_1) \in R$ and $(x, w_2) \in R$ such that the view of V^* in the interacting with $P(x, w_1)$ is computationally indistinguishable from the view of V^* in the interacting with $P(x, w_2)$.

[9] showed that WI property can be preserved in concurrent setting. Hence n parallel composition of Blum's proof for **NP** is a construction of WI proofs for **NP** with negligible soundness error probability.

2.6 Proofs of Knowledge

In a proof/argument system, the prover convinces the verifier that some string x is a member of a language L. In a proof/argument of knowledge (POK/AOK) [8][3][14][20] the prover should convince the verifier that it also knows a witness to the fact that $x \in L$. This is formalized by requiring that if the verifier is convinced with some probability p' by some (possibly cheating) prover strategy, then by applying an efficient algorithm, called knowledge extractor, to the cheating prover's strategy and private inputs, it is possible to obtain a witness to the fact that $x \in L$, with probability (almost equal to) p'. The formal definition is below:

Definition 2.7. Let $L = L(R)$ and let (P, V) be a proof (argument) system for L. We say that (P, V) is a proof (argument) of knowledge for L if there exists a probabilistic polynomial-time algorithm E (called the knowledge extractor) such that for every polynomial-sized prover P^* and for every $x \in \{0,1\}^n$, if we let p' denote the probability that V accepts x when interacting with P^*, then $\Pr[E(P^*, x) \in R(x)] \geq p' - \mathsf{neg}(n)$.

Instantiated with a perfectly-binding (resp. perfectly-hiding) commitment scheme, the n parallel executions of Blum's protocol is a (resp. perfectly) WI POK (resp. AOK), ensured by the special soundness property.

3 The New Notion

Counting Time and Space. If M is a probabilistic (non-interactive) machine, denote by M_r the deterministic one obtained by fixing M's random coins to r, by $\mathsf{T}_{M_r(x)}$ and $\mathsf{S}_{M_r(x)}$ the running-time and running-space of M_r on input x. (As aforementioned, we herein don't explicitly refer to any machine model and definitions of the complexity measures of running-time and running-space. Each of general models and the appropriate definitions of the complexity measures in it are suitable for this paper.)

Assume (P, V) uses κ-round prover's messages. For any interactive machine V^* with auxiliary input z (w.l.o.g. assume V^* is deterministic), denote by $v = (x, z, (m_1, m_2, ..., m_\kappa))$ the view of V^*. Then denote by $\mathsf{T}_{V^*}(v)$ and by $\mathsf{S}_{V^*}(v)$ the running-time and running-space of V^* on input x and letting the j^{th} message received be m_j, $1 \leq j \leq \kappa$. For convenience of statement, we will always consider V^* to have x, z hardwired and refer the view of V^* to $(m_1, m_2, ..., m_\kappa)$ in the rest of this paper (similarly for any P^*).

Definition 3.1. (PTSSZK) Let (P, V) be an interactive proof (argument) for a language $L = L(R)$, $p_t : N \times N \to N$ and $p_s : N \times N \to N$ be two monotonically increasing 2-variate functions. We say that (P, V) is computational zero-knowledge, or just zero-knowledge, with time precision p_t and space precision p_s if there exists a probabilistic polynomial-time algorithm S, called *precise simulator*, such that for every polynomial-sized V^* and every $\{(x, w)\} \in R$ the following conditions hold:

1. The output of $S(x, V^*)$ is computationally indistinguishable from the view of V^* in an execution of $(P(w), V^*)(x)$.
2. For every sufficiently long $r \in \{0,1\}^*$, let v be the view generated by $S_r(x, V^*)$. Then $\mathsf{T}_{S_r(x,V^*)} \leq p_t(n, \mathsf{T}_{V^*}(v))$ and $\mathsf{S}_{S_r(x,V^*)} \leq p_s(n, \mathsf{S}_{V^*}(v))$.

Remark 3.2. If we only require the running-time of S and V^* satisfy the constraint $p_t(n, y)$, regardless of the constraint $p_s(n, y)$ for their running-space, this is essentially the definition of precise zero-knowledge given in [15].

It can be seen a main task of constructing a PTSSZK protocol is to construct a precise simulator S. To output a simulated view, S usually needs to invoke an interaction in which it acts as prover on one hand and emulates V^* to output verifier's messages on the other hand. To make the analysis of S' running-space accurately, we make the following specifications on the decomposition of running-space of S when emulating V^* (which are quite general).

Specifications. For our simulator S, $\mathsf{S}_{S_r(x,V^*)}$ consists of three parts: (1) the space needed to emulate V^*, (2) the predetermined space (decided by the protocol) needed to store the communicated messages (notice that V^*'s computation is emulated by S and thus if S detects V^* tries to send a message longer than what the protocol specifies it aborts) and (3) the space needed to perform the prover's strategy and extraction etc.

Time and Space Cost in Emulation. For universal machine S, we assume there are two universal constants l_1, l_2 satisfying that the time taken by S in emulating V^* on view v is $l_1\mathsf{T}_{V^*}(v)$ [15], and the space taken by S in emulating V^* on v (i.e. part 1 described in the previous paragraph) is $l_2\mathsf{S}_{V^*}(v)$.

4 High-Level Proofs of Our Results

In this and next section we prove Theorem 1.1 and Theorem 1.2. This section only presents the high-level constructions of the desired protocols. To do this we present a new primitive in Section 4.1, called precise time and space emulatable proofs/arguments of knowledge (PTSEPOK/PTSEAOK). In Section 4.2, we present the PTSSZK proofs and arguments under the assumption of the existence of the PTSEPOKs and PTSEAOKs. In Section 5 we will present the detailed constructions of the required PTSEPOKs and PTSEAOKs and thus complete the entire proofs.

4.1 PTSEPOKs and PTSEAOKs

The notion of PTSEPOKs and PTSEAOKs captures the idea that for any prover P^* and any x, (1) the joint view of P^* and the honest verifier V and (2) the witness for $x \in L$ whenever V's view is accepting, can be simultaneously reconstructed (by an algorithm, called emulator-extractor) in the time and space almost identical to those taken by P^* on the reconstructed view. The formal description is shown as follows.

Definition 4.1. (PTSEPOK/PTSEAOK) Let $L = L(R)$, (P, V) be an interactive proof (argument) for L, $p_t : N \times N \to N$ and $p_s : N \times N \to N$ be two monotonically increasing 2-variate functions. We

Public Input: $x \in \{0,1\}^n$ (statement to be proved is "$x \in L$"). **Auxiliary Input to Prover:** w, a witness for $x \in L$.
Stage 1: $V \to P$: V chooses $\sigma \in_R \{0,1\}^n$, $s \in_R \{0,1\}^{\text{poly}(n)}$ and sends $\mathsf{c} = \mathsf{Com}(\sigma; s)$. $V \to P$: V proves to P via the perfectly WI PTSEAOK of the statement: there exist values σ and s such that $\mathsf{c} = \mathsf{Com}(\sigma; s)$.
Stage 2: $P \to V$: P proves to V in the (slightly modified) n parallel repetitions of Blum's proof [5] that $x \in L$, in which V opens c and uses the string σ as its challenge.

Protocol 4.4. The PTSSZK proof for L.

say (P, V) is a proof (argument) of knowledge with time precision p_t and space precision p_s if there exists a probabilistic polynomial-time algorithm E, called *precise emulator-extractor*, satisfying that for every polynomial-sized prover P^* and for every $x \in \{0,1\}^n$ the following conditions hold:

1. Let $(\mathsf{view}_{P^*}, \mathsf{view}_V, w)$ denote the output of $E(x, P^*)$. Then $(\mathsf{view}_{P^*}, \mathsf{view}_V)$ is identically distributed to the joint view in the interaction of $(P^*, V)(x)$. Further, if with non-negligible probability p' view_V is a convincing view, then w is a witness for $x \in L$ with probability $p' - \mathsf{neg}(n)$.
2. For every sufficiently long $r \in \{0,1\}^*$ let $(\mathsf{v}_{P^*}, \mathsf{v}_V, w) \leftarrow E_r(x, P^*)$. Then $\mathsf{T}_{E_r(x,P^*)} \leq p_t(n, \mathsf{T}_{P^*}(\mathsf{v}_{P^*}))$ and $\mathsf{S}_{E_r(x,P^*)} \leq p_s(n, \mathsf{S}_{P^*}(\mathsf{v}_{P^*}))$.

Remark 4.2. To be a building block of our PTSSZK protocols, a PTSEPOK (or PTSEAOK) (P, V) should have one more property, i.e. (perfect) WI. That is, (P, V) should additionally satisfy that for every polynomial-sized V^* and for every x, w_1, w_2 where $(x, w_1) \in R$ and $(x, w_2) \in R$ V^*'s view in the interacting with $P(x, w_1)$ is (identical to) computationally indistinguishable from its view in the interacting with $P(x, w_2)$.

4.2 The Constructions

Claim 4.3. *Assume there exist constant-round perfectly-hiding commitments and m'-round perfectly WI PTSEAOKs with time precision poly$(n) + O(y)$ (resp. poly$(n) + O(ny)$) and space precision poly$(n) + O(y)$ for each* **NP** *relation. Then for every language $L \in$* **NP**, *there exists an $m' + O(1)$-round zero-knowledge proof with time precision $p_t(n, y) =$ poly$(n) + O(y)$ (resp. poly$(n) +$ poly$(n)O(y)$) and space precision $p_s(n, y) =$ poly$(n) + O(y)$ for L.*

Proof. Our protocol is depicted in Protocol 4.4 where Com is a constant-round perfectly-hiding commitment. We show the protocol has all the required properties.

Completeness and Soundness. First, it is obvious completeness holds. Second, by perfect WI of the PTSEAOK, it holds that the whole stage 1 is still a perfectly-hiding commitment. Thus we deduce that when reaching stage 2 the prover has no idea about the value of challenge σ. In other words, even though the cheating prover reaches the second stage after seeing all messages in the first stage, the messages in the second stage are independent of the verifier's messages in stage 1. So a cheating prover violating the soundness of our protocol can be transformed to a full power cheating prover violating the soundness of the n parallel executions of Blum's proof. Thus soundness holds.

Simulator S. For every polynomial-sized V^* and $x \in L$, S works as follows: It first emulates V^* to the commitment c. Then S uses the strategy of the emulator-extractor E of the PTSEAOK

to generate $(\mathsf{view}_{V^*1}, \mathsf{view}_{P1}, (\sigma, s))$. (Strictly speaking, the input to E should be c and V^*'s code as Definition 4.1 requires, not the residual V^* after sending c. But since the essential way that E uses (the residual) V^* is to emulate it, as next section shows, our statement is still correct.) If view_{P1} is not a convincing view, S terminates the interaction. Otherwise, the extracted σ is that one V^* chose. Then to generate the view in stage 2, S adopts the honest prover strategy using the knowledge of the verifier challenge σ to interact with V^*. Denote by view_{V^*2} the simulated view of V^* in stage 2. Let T (resp. S) denote V^*'s running-time (resp. running-space) on $\mathsf{view}_{V^*1} \circ \mathsf{view}_{V^*2}$, where "$\circ$" denotes concatenation.

Output Distribution. First, the view output by E is identical to that of stage 1 in the real interaction. Second, knowing the challenge in advance makes S' output in stage 2 computationally indistinguishable from a real view of V^* in stage 2, ensured by Blum's simulation strategy. Using a hybrid argument, we infer S' output is computationally indistinguishable from the real view of V^*.

Running-Time. By the time precision of the PTSEAOK, it takes S at most $\mathrm{poly}(n) + O(\mathsf{T})$ time (resp. $\mathrm{poly}(n) + \mathrm{poly}(n)O(\mathsf{T})$, where the latter $\mathrm{poly}(n)$ is $|\mathsf{c}|$) to finish the simulation of stage 1. In stage 2 it takes S a $\mathrm{poly}(n)$ time to perform the honest prover's strategy and $O(\mathsf{T})$ time to emulate V^*. Consequently, there is a $p_t(n, y)$ as required satisfying S' running-time is bounded by $p_t(n, \mathsf{T})$.

Running-Space. By the space precision of the PTSEAOK, S' running-space in stage 1 is less than $\mathrm{poly}(n) + O(\mathsf{S})$ (this $\mathrm{poly}(n)$ is within $|(\mathsf{view}_{V^*1}, \mathsf{view}_{P1}, (\sigma, s)| + \mathrm{poly}(|\mathsf{c}|)$, where the latter poly is that one in the space precision of the PTSEAOK). Second, S' running-space in stage 2 is less than $\mathrm{poly}(n) + O(\mathsf{S})$ (this $\mathrm{poly}(n)$ denotes the space needed to perform the honest prover's strategy and store the view of stage 2). Consequently, there is a $p_s(n, y)$ as required satisfying S' running-space is bounded by $p_s(n, \mathsf{S})$. The claim follows. □

Claim 4.5. *Assume there exist one-way functions and m'-round WI PTSEPOKs with time precision $\mathrm{poly}(n) + O(y)$ (resp. $\mathrm{poly}(n) + O(ny)$) and space precision $\mathrm{poly}(n) + O(y)$ for each* **NP** *relation. Then for every language $L \in$ **NP**, there exists an $m' + O(1)$-round zero-knowledge argument with time precision $p_t(n, y) = \mathrm{poly}(n) + O(y)$ (resp. $\mathrm{poly}(n) + \mathrm{poly}(n)O(y)$) and space precision $p_s(n, y) = \mathrm{poly}(n) + O(y)$ for L.*

Proof. Our protocol is shown in Protocol 4.6, where f is a one-way function. Since it is an instantiation of Feige and Shamir's protocol [10], completeness and computational soundness hold. The simulator S works in almost the same way described in the previous proof. The difference is that what S obtains by running E in stage 1 is r_1 or r_2 and then S uses r_1 or r_2 as a witness for the combined statement to finish the interaction of stage 2. The desired time and space precisions are also satisfied by using the same analysis in the previous proof. Lastly, we know the view output by E is identical to that of stage 1 in the real interaction, and the protocol in stage 2 is WI. Then S' output is computationally indistinguishable from the real view of V^*. The claim follows. □

Remark 4.7. If we use (interactive) Turing machines as the underlying model, better time precisions can be obtained, i.e. $p_t(n, y) = O(y)$ (resp. $p_t(n, y) = \mathrm{poly}(n)O(y)$), only with a slightly modification introduced by [15] on the above constructions. That is, we add a step in the beginning of the protocols, i.e., V sends $1^{W(n)}$ to P, where $W(n)$ is a polynomial bound on the summand $\mathrm{poly}(n)$ in p_t; then P verifies that V sent a string of length $W(n)$ and if not it aborts. In the model of interactive Turing machines (refer to [11], Sec 4.2.), writing/reading a string of length $W(n)$ to/from the communication tapes needs time at least $W(n)$, which results in $\mathsf{T} \geq W(n)$. Verifying the validity of the 1-string costs $O(W(n)) = O(\mathsf{T})$ time. Thus the $p_t(n, y)$ can be expressed as $O(y)$ (resp. $p_t(n, y) = \mathrm{poly}(n)O(y)$).

Public Input: $x \in \{0,1\}^n$ (statement to be proved is "$x \in L$"). **Auxiliary Input to Prover:** w, a witness for $x \in L$.
Stage 1: $V \to P$: V chooses $r_1, r_2 \in_{\mathrm{R}} \{0,1\}^n$ and sends $v_1 = f(r_1)$, $v_2 = f(r_2)$. $V \to P$: V proves to P via the WI PTSEPOK of the statement: either there exists a r_1 s.t. $v_1 = f(r_1)$ or there exists a r_2 s.t. $v_2 = f(r_2)$.
Stage 2: $P \to V$: P proves to V in the n parallel repetitions of Blum's proof that $x \in L$ or there exists a r_1 s.t. $v_1 = f(r_1)$ or there exists a r_2 s.t. $v_2 = f(r_2)$.

Protocol 4.6. The PTSSZK argument for L.

In the high-level proofs, many details of S are hidden in E. In next section we will construct the required PTSEPOKs and PTSEAOKs, in which the key task is to present all the details of E.

5 Constructions of the PTSEPOKs and PTSEAOKs

This section is devoted to the constructions of the required PTSEPOKs and PTSEAOKs. The emphasis is the probability analysis of successful extraction of E. (Since it is actually a partial strategy of S, E also follows the specifications on running-space shown in Section 3). We illustrate the constructions of the PTSEPOKs in detail and then show the PTSEAOKs can be obtained similarly. We also show two remarks on E's extraction strategy and its comparison with the original "cut-off" technique in [19].

Claim 5.1. *Assume there exist one-way functions. Then for each* **NP** *relation R there exists an $\omega(\log^3 n)$-round WI PTSEPOK with time precision $p_t(n,y) = poly(n) + O(y)$ and space precision $p_s(n,y) = poly(n) + O(y)$ for R.*

Proof. Instantiate the commitment scheme in Blum's proof for R with a constant-round perfectly-binding commitment scheme. This proof is special-sound in the sense that there is a polynomial-time algorithm, denoted Ex_R, which given the public input x and two different valid transcripts w.r.t. a common commitment can compute a witness w for x. We call the n parallel repetitions of this proof the atomic protocol. Let (P,V) denote the $m = \omega(\log^3 n)$ sequential repetitions of the atomic protocol, in each of which P proves to V the knowledge of a witness for x. We will show (P,V) is a WI PTSEPOK with the required properties. It can be seen that completeness and soundness hold and WI is preserved. Thus we only need to construct a precise emulator-extractor E for (P,V), as the following shows.

The Construction of E. For any polynomial-sized (deterministic) P^* (we only need to consider polynomial-sized provers) and x, and for each $i \in \{1, \cdots, m\}$:

Step i.1. E adopts the honest verifier's strategy to interact with P^* in the i^{th} atomic protocol to output the joint view (this is the first run, used to gather the joint view), and at the same time records P^*'s running-time and running-space. If P^* sends an invalid message in this atomic protocol, E halts outputting $\perp$. Otherwise let r_i denote E's challenge in this atomic protocol, v_i denote the current prover's view prior to r_i (exclusive). Let t_i denote P^*'s running-time in computing the response on receiving r_i, s_i denote (*the size* of) P^*'s running-space on view $\mathsf{v}_i \circ r_i$.

Step i.2. If the witness has not been extracted, E performs the extraction (this is the second run only for extraction). It rewinds P^* to the point where r_i is supposed to be sent, and chooses a

new challenge $r'_i \in_R \{0,1\}^n$ and sends r'_i to P^*, and recycles the space used in emulating P^* of the previous rewinds. It emulates P^* for at most $2t_i$ steps, and at the same time keeps track of P^*'s running-space on view $\mathsf{v}_i \circ r'_i$ and checks whether it is within s_i. If P^* can output a response to r'_i within $2t_i$ time and its running-space on $\mathsf{v}_i \circ r'_i$ is within s_i, E computes a witness w by running Ex_R (in the case that the response is valid and $r'_i \neq r_i$ or else E proceeds to next atomic protocol). Otherwise, E cuts off P^*'s computing and proceeds to next atomic protocol. (If $i = m$, E performs the extraction and halts outputting w if it has been extracted.)

Properties of E. Let $(\mathsf{view}_{P^*}, \mathsf{view}_V)$ be the joint view output by E. By E's strategy in the first run, we have $(\mathsf{view}_{P^*}, \mathsf{view}_V)$ is identical to the joint view in a real interaction of $(P^*, V)(x)$. Let T and S denote the running-time and running-space of P^* on view_{P^*} respectively. Hence we only need to show E has the following properties.

Running-Time. Since P^*'s running-time in the rewind of the i^{th} atomic protocol is at most $2t_i$, $1 \leq i \leq m$, P^*'s running-time in all rewind runs is no more than $2 \cdot \sum_{i \in [1,m]} t_i < 2\mathsf{T}$. Then it takes E $O(\mathsf{T})$ time to emulate P^* in the entire simulation. Further, sending prover's messages, checking if a transcript is accepting, and extracting a witness for two accepting transcripts, can be done in a fixed polynomial time. Thus E's running-time is bounded by $\mathrm{poly}(n) + O(\mathsf{T})$.

Running-Space. First, it takes E $O(\mathsf{S})$ space to emulate P^*'s computing on view_{P^*}. Second, by the recycling strategy, it also takes E $O(\mathsf{S})$ space to emulate all P^*'s computing on $\mathsf{v}_i \circ r'_i$ for $1 \leq i \leq m$. Third, besides emulating P^*, E needs a more fixed polynomial space to carry out other computing and store the joint view. Thus E's running-space is bounded by $\mathrm{poly}(n) + O(\mathsf{S})$.

Extractable Probability. Let Accept (resp. Reject) denote the event that view_V is an accepting (resp. rejective) view. Let Suc denote the event E succeeds in extraction. Then what we need to prove is $\Pr[\mathsf{Suc}|\mathsf{Accept}] = 1 - \mathsf{neg}(n)$ if $\Pr[\mathsf{Accept}]$ is non-negligible. On the occurrence of Reject, we can regard the extraction as successful with probability 1, i.e. $\Pr[\mathsf{Suc}|\mathsf{Reject}] = 1$ (note since E is a partial strategy of S, if E's view is rejective then S terminates the simulation and the view output by S is still identical to the real view). Thus if we are able of proving $\Pr[\mathsf{Suc}] = 1 - \mathsf{neg}(n)$, then $\Pr[\mathsf{Accept}]\Pr[\mathsf{Suc}|\mathsf{Accept}] + \Pr[\mathsf{Reject}]\Pr[\mathsf{Suc}|\mathsf{Reject}] = 1 - \mathsf{neg}(n)$. By some calculation, we have $\Pr[\mathsf{Accept}](1 - \Pr[\mathsf{Suc}|\mathsf{Accept}]) = \mathsf{neg}(n)$. Since $\Pr[\mathsf{Accept}]$ is non-negligible, $\Pr[\mathsf{Suc}|\mathsf{Accept}] = 1 - \frac{\mathsf{neg}(n)}{\Pr[\mathsf{Accept}]} = 1 - \mathsf{neg}(n)$. Consequently, to prove $\Pr[\mathsf{Suc}|\mathsf{Accept}] = 1 - \mathsf{neg}(n)$, we only need to show $\Pr[\mathsf{Suc}] = 1 - \mathsf{neg}(n)$. Actually, this is true, ensured by Claim 5.4. The claim follows. □

Before proceeding to Claim 5.4, we show two remarks on E's extraction strategy and its comparison with the original "cut-off" technique in [19]. If we modify E by only requiring it cuts off P^*'s computing in the second run of each atomic protocol iff P^* cannot output a response to r'_i within t_i time, then this is the original "cut-off" technique. Notice that there are two differences between it and the extraction strategy herein. The main difference is that E herein uses the "cut-off" technique by simultaneously considering P^*'s running-time and running-space instead of P^*'s running-time alone. Another difference is that E herein allows P^* to run $2t_i$ time at most in the second run instead of t_i time. In Remark 5.2, we show the original "cut-off" technique cannot obtain simultaneous meaningful time and space precisions. In Remark 5.3, we show the time for P^* to run in the second run cannot be reduced to t_i (if we require that the space for it to run is at most s_i). That is, it is necessary for E to allow P^* to run t_i times a factor (e.g. 2) steps in the second run.

Remark 5.2. We now show if E adopts the original "cut-off" technique, although it can of course obtain linear time precision it cannot obtain any meaningful space precision. To this end we need

to show for any meaningful $p_s(n, y)$ there exists a P^* satisfying E's running-space is more than $p_s(n, \mathsf{S})$ with non-negligible probability (even if m can be any arbitrary polynomial).

First consider a prover P' defined below. P' proceeds just as the honest prover P, except that P' receives a random 2-wise independent hash function $h : \{0,1\}^n \to \{0,1\}^n$ as the auxiliary input. Let $Q(n)$ be a polynomial bound on the running-space of P. Then in the 1^{st} atomic protocol on receiving a challenge (r_1 or r'_1) P' applies h to this challenge and generates a random string a. If the first $\log n$ bits of a are not zeros, P' executes some instructions to consume $p_s(n, Q(n))^2$ more space. Otherwise, it also executes some dummy instructions to consume the same time but no more space. Then, it proceeds to do what P would on a challenge and also executes some instructions (which don't consume any more space), if needed, to adjust its running-time in this step to be same in answering different challenges. In all the residual atomic protocols, P' behaves identically to P.

In the 1^{st} atomic protocol, since P''s running-time on different challenges are always identical, there is no possibility for E to cut off P''s computing in the second run and thus the extraction always succeeds in case $r_1 \neq r'_1$. However, it can be seen the event that P''s running-space on view $\mathsf{v}_1 \circ r_1$ is less than $Q(n)$ and its running-space on view $\mathsf{v}_1 \circ r'_1$ is at least $p_s(n, Q(n))^2$ occurs with probability $\frac{1}{n}(1 - \frac{1}{n})$. Thus E's running-space is more than $p_s(n, \mathsf{S})$ with probability at least $\frac{1}{n}(1 - \frac{1}{n})$. By an averaging argument, there exists at least one auxiliary input $z = h$ resulting in the event occurs with that probability. Let P^* be P' having z hardwired and then P^* is the desired one.

Remark 5.3. Now we show the factor (i.e. 2) for t_i in E's strategy in the second run cannot be eliminated. That is, we show if we bound the time and space for a prover to run in the second run are respectively t_i and s_i, then there exists a P^* such that E fails in extraction with non-negligible probability even if m can be any arbitrary polynomial.

Consider a prover P^* defined below. P^* proceeds just as P, except that the following differences. According to P^*'s running-time and running-space, all the $N = 2^n$ challenges in $\{0,1\}^n$ can be divided into $n^{c_0} > m$ disjoint classes, ordered in an arbitrary way, each of which has $\frac{N}{n^{c_0}}$ challenges, such that for $1 \leq i \leq m$ when executing the i^{th} atomic protocol (the view prior to this atomic protocol is then fixed): 1. besides doing what P would, P^* also executes some instructions to ensure its running-time (resp. running-space) on any of the challenges (resp. the generated view concatenated with any of the challenges) from the same class are identical; 2. if denote by $t_i^{(j)}$ (resp. $s_i^{(j)}$) P^*'s running-time (resp. running-space) on any challenge (resp. the generated view concatenated with any challenge) from the j^{th} class, then $t_i^{(1)} < t_i^{(2)} < \cdots < t_i^{(n^{c_0})}$ but $s_i^{(1)} > s_i^{(2)} > \cdots > s_i^{(n^{c_0})}$.

Hence for each i a sufficient condition for the extraction to fail in the i^{th} atomic protocol is that r_i and r'_i are chosen from two different classes, which occurs with probability $n^{c_0} \cdot \frac{1}{n^{c_0}} \cdot (1 - \frac{1}{n^{c_0}}) = 1 - \frac{1}{n^{c_0}}$. Thus the extraction fails in all the m atomic protocols with probability at least $(1 - \frac{1}{n^{c_0}})^m = O((\frac{1}{e})^{m/n^{c_0}}) > c'$ for a constant $0 < c' < 1$.

Claim 5.4. *The E described in the proof of Claim 5.1 succeeds in extraction with probability $1 - \mathsf{neg}(n)$.*

Proof. If there exists an i satisfying that the extraction always succeeds in the first i atomic protocols for all outcomes of E's coins used in the first and second runs of these protocols, the claim of course holds. Thus in the following we prove this claim by assuming for all i's there are some outcomes of the E's coins resulting in the extraction fails in the first i atomic protocols.

Fix any outcome of such E's coins that result in the extraction fails in the first $i - 1$ atomic protocols. Let us analyze the probability of successful extraction in the i^{th} atomic protocol. We

will present a uniform lower bound for this probability when further fixing any outcome of E's coins used in this atomic protocol before sending the challenge (r_i). (Thus view v_i is determined.) In the case P^* outputs an invalid message on view v_i, the probability of successful extraction is 1. Thus we only need to consider the case the extraction can reach the step E sends the challenge as follows.

We first (re)highlight some crucial random variables. Let r_i (resp. r'_i) denote the random challenge in $\{0,1\}^n$ sent to P^* in the first (resp. second) run. Let t_i (resp. t'_i) denote P^*'s running-time in answering r_i (resp. r'_i) in the first (resp. second) run. Let s_i (resp. s'_i) denote (the size of) P^*'s running-space on view $\mathsf{v}_i \circ r_i$ (resp. $\mathsf{v}_i \circ r'_i$). Since coins E uses are independent, we have that r_i and r'_i are independently identically distributed (*i.i.d.*), t_i and t'_i are *i.i.d.*, s_i and s'_i are *i.i.d.*.

If P^* outputs a valid response to r_i in the first run, a necessary and sufficient condition for the extraction to succeed is that $s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i$ and P^* can output a valid response to r'_i in the second run. Otherwise, the extraction is regarded as successful. If we set t_i, s_i (resp. t'_i, s'_i) the values that are respectively greater than 2 times P^*'s maximal running-time and P^*'s maximal running-space when P^* outputs an invalid response in the first (resp. second) run, then in the former case the necessary and sufficient condition can be reduced to $s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i$ and in the latter case $s'_i \leq s_i, t'_i \leq 2t_i$ can be satisfied too. This means $s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i$ can be used as a uniform sufficient condition for the extraction to succeed.

Thus assuming P^*'s running-time and running-space are bounded by $\frac{1}{4}n^c - 1$ for some non-predetermined constant $c > 0$ and for sufficiently large n's (the coefficients $\frac{1}{4}$ and -1 were chosen only for notational convenience), we set $t_i = \frac{1}{2}n^c - 1$ and $s_i = \frac{1}{4}n^c$ (resp. $t'_i = \frac{1}{2}n^c - 1$ and $s'_i = \frac{1}{4}n^c$) if P^* outputs an invalid response in the first (resp. second) run. Hence, by the analysis in the above paragraph, the probability the extraction succeeds is at least $\Pr[s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i]$. Thus, our main task in this proof is to present a lower bound for this probability.

Since $\Pr[s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i] \geq 1 - \Pr[s'_i > s_i] - \Pr[t'_i > 2t_i] - \Pr[r_i = r'_i]$, and $\Pr[r_i = r'_i] = 2^{-n}$, we only need to evaluate $\Pr[s'_i > s_i]$ and $\Pr[t'_i > 2t_i]$.

It follows from the symmetry property that $\Pr[s'_i < s_i] = \Pr[s_i < s'_i]$. Since $\Pr[s'_i < s_i] + \Pr[s_i < s'_i] + \Pr[s_i = s'_i] = 1$, we have $2 \cdot \Pr[s'_i > s_i] < 1$. Thus, $\Pr[s'_i > s_i] < \frac{1}{2}$. For the same reason, $\Pr[t'_i < t_i] = \frac{1}{2} - \frac{1}{2}\Pr[t_i = t'_i]$, $\Pr[t'_i \leq t_i] = \frac{1}{2} + \frac{1}{2}\Pr[t_i = t'_i]$.

Let $\delta(n) = (\frac{1}{m})^{1/3}$. ($m(n) = \omega(\log^3 n)$, shown in the proof of Claim 5.1.) Then $\delta(n) = \frac{1}{\omega(\log n)}$. In the following we divide the evaluation of $\Pr[t'_i \leq 2t_i]$ and $\Pr[s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i]$ into two cases. Before proceeding we first define some notations and variables.

Notations and Variables. We define the following notations and variables.

(1) We use $t(u)$ to denote the value of t_i when the challenge to P^* is $u \in \{0,1\}^n$ in the first run. (Equivalently, $t(u)$ is the value of t'_i when the challenge to P^* is u in the second run.) We order the $N = 2^n$ challenges in $\{0,1\}^n$ as $u_1, u_2, \cdots, u_N$ in an arbitrary way only if this order satisfies $t(u_1) \leq t(u_2) \leq \cdots \leq t(u_N)$. We also write $u_1 < u_2 < \cdots < u_N$ by this order, and say u_1 is less than u_2, u_2 is less than u_3 and so on. Then when we say "choose the minimal challenge from set $U \subset \{0,1\}^n$, denoted u" we mean the chosen $u \in U$ is less than any one in $U - \{u\}$. For simplicity, let a_k denote $t(u_k)$, $1 \leq k \leq N$. (a_k's may not be mutually different and $1 \leq a_k < \frac{1}{2}n^c$ for each k.)

(2) For each u_k, we define a set A_k corresponding to u_k as $A_k \stackrel{\text{def}}{=} \{u \in \{0,1\}^n : a_k \leq t(u) \leq 2a_k\}$.

(3) For each u_k, we define N_k as the cardinal number of the set $\{u \in \{0,1\}^n : t(u) = a_k\}$. It is clear that $|A_k| \geq N_k$ for each k.

Case 1. Assume there is a k_0 satisfying $\frac{N_{k_0}}{N} \geq \delta(n)$. Let $B \stackrel{\text{def}}{=} \{k \in [1, N] : t(u_k) = a_{k_0}\}$. Then $N_{k_0} = |B| = N_k$ for each $k \in B$. Since r_i is independent of t'_i, $\Pr[t_i = t'_i] =$

$\sum_{k\in[1,\,N]} \Pr[r_i = u_k, t'_i = a_k] \geq \sum_{k\in B} \Pr[r_i = u_k, t'_i = a_k] = \sum_{k\in B} \frac{1}{N} \cdot \frac{N_{k_0}}{N} = \frac{N_{k_0}}{N}\frac{N_{k_0}}{N} \geq \delta^2$.

Thus, $\Pr[t'_i \leq 2t_i] \geq \Pr[t'_i \leq t_i] = \frac{1}{2} + \frac{1}{2}\Pr[t_i = t'_i] \geq \frac{1}{2} + \frac{1}{2}\delta^2$. Then $\Pr[t'_i > 2t_i] \leq \frac{1}{2} - \frac{1}{2}\delta^2$. Hence, $\Pr[s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i] > 1 - \frac{1}{2} - (\frac{1}{2} - \frac{1}{2}\delta^2) - 2^{-n} = \frac{1}{2}\delta^2 - 2^{-n}$.

Case 2. Assume $\frac{N_k}{N} < \delta(n)$ for all k's. Since $\Pr[t'_i \leq 2t_i] = \Pr[t'_i < t_i] + \Pr[t_i \leq t'_i \leq 2t_i] = \frac{1}{2}(1 - \Pr[t_i = t'_i]) + \Pr[t_i \leq t'_i \leq 2t_i]$, to evaluate $\Pr[t'_i \leq 2t_i]$ we first compute $\Pr[t_i \leq t'_i \leq 2t_i]$, which is actually the main part of this case.

Our fundamental approach to computing $\Pr[t_i \leq t'_i \leq 2t_i]$ is that we construct a decomposition $\{D_1, \cdots, D_H\}$ of the sample space for some integer $H > 0$, and compute $\Pr[t_i \leq t'_i \leq 2t_i | D_h]$ for $1 \leq h \leq H$ and then apply the formula of total probability to evaluate $\Pr[t_i \leq t'_i \leq 2t_i]$. Thus, in the following we will show how to construct the desired decomposition and compute the conditional probabilities.

We first show what D_1 is. Choose the minimal challenge from $\{0,1\}^n$, denoted $u_{k_{11}}$, $1 \leq k_{11} \leq N$. ($k_{11} = 1$ actually.) If $A_{k_{11}}$ doesn't contain $\{0,1\}^n$, then choose the minimal challenge from $\{0,1\}^n - A_{k_{11}}$, denoted $u_{k_{12}}$, $1 \leq k_{12} \leq N$. If $A_{k_{11}} + A_{k_{12}}$ (note that $A_{k_{11}} \cap A_{k_{12}} = \phi$) still doesn't contain $\{0,1\}^n$, continue to choose the minimal challenge from $\{0,1\}^n - A_{k_{11}} - A_{k_{12}}$, denoted $u_{k_{13}}$, $1 \leq k_{13} \leq N$, and proceed. Since $\{0,1\}^n$ is finite, using this selection method we can finally obtain a sequence of challenges $\{u_{k_{11}}, \cdots, u_{k_{1d_1}}\}$ for some $d_1 > 0$, satisfying $\sum_{j\in[1,d_1]} A_{k_{1j}} = \{0,1\}^n$ (note that $A_{k_{1j}}$'s are disjoint). Let D_1 denote the event of $r_i \in \{u_{k_{11}}, \cdots, u_{k_{1d_1}}\}$, D_1^* denote $\{u_{k_{11}}, \cdots, u_{k_{1d_1}}\}$.

Let us evaluate the value of d_1. Since $a_1, \cdots, a_N$ are bounded by $\frac{1}{2}n^c$, and $a_{k_{11}} < 2a_{k_{11}} < a_{k_{12}} < 2a_{k_{12}} < \cdots < a_{k_{1d_1}} < 2a_{k_{1d_1}}$, we have $a_{k_{11}} \cdot 2^{d_1 - 1} < a_{k_{1d_1}} < \frac{1}{2}n^c$. Hence $d_1 < c\log n$. Further, as $\sum_{j\in[1,d_1]} A_{k_{1j}} = \{0,1\}^n$, there is a K_1, $1 \leq K_1 \leq d_1$, satisfying $|A_{k_{1K_1}}| \geq \max[\frac{N}{d_1}, N_{k_{1K_1}}]$. Then,

$$\Pr[t_i \leq t'_i \leq 2t_i | D_1] = \frac{1}{\Pr[D_1]}\Pr[t_i \leq t'_i \leq 2t_i, D_1] = \frac{1}{\Pr[D_1]}\sum_{j\in[1,\,d_1]} \Pr[r_i = u_{k_{1j}}, a_{k_{1j}} \leq t'_i \leq 2a_{k_{1j}}]$$

$$= \frac{1}{\Pr[D_1]}\sum_{j\in[1,\,d_1]} \Pr[r_i = u_{k_{1j}}] \cdot \Pr[a_{k_{1j}} \leq t'_i \leq 2a_{k_{1j}}] = \sum_{j\in[1,\,d_1]} \Pr[r_i = u_{k_{1j}} | D_1] \cdot \Pr[a_{k_{1j}} \leq t'_i \leq 2a_{k_{1j}}]$$

$$\geq \sum_{j\in[1,\,d_1]-\{K_1\}} \frac{1}{d_1}\frac{N_{k_{1j}}}{N} + \frac{1}{d_1 N}\max[\frac{N}{d_1}, N_{k_{1K_1}}] = \sum_{j\in[1,\,d_1]} \frac{1}{d_1}\frac{N_{k_{1j}}}{N} + \frac{1}{d_1 N}\max[\frac{N}{d_1} - N_{k_{1K_1}}, 0]$$

If D_1 equals the sample space, the decomposition has only one element D_1. Otherwise, we need to construct D_2 by using the similar way in constructing D_1. Choose the minimal challenge from $\{0,1\}^n - D_1^*$, denoted $u_{k_{21}}$, $1 \leq k_{21} \leq N$. If $A_{k_{21}}$ doesn't contain $\{0,1\}^n - D_1^*$, then choose the minimal challenge from $\{0,1\}^n - D_1^* - A_{k_{21}}$, denoted $u_{k_{22}}$, $1 \leq k_{22} \leq N$. If $A_{k_{21}} + A_{k_{22}}$ still doesn't contain $\{0,1\}^n - D_1^*$, continue this selection. Finally, using this selection method we obtain a sequence of challenges $\{u_{k_{21}}, \cdots, u_{k_{2d_2}}\}$ for some $d_2 > 0$ satisfying $\sum_{j\in[1,d_2]} A_{k_{2j}}$ contains $\{0,1\}^n - D_1^*$. Let D_2 denote the event of $r_i \in \{u_{k_{21}}, \cdots, u_{k_{2d_2}}\}$, D_2^* denote $\{u_{k_{21}}, \cdots, u_{k_{2d_2}}\}$. Similarly, we have $d_2 < c\log n$.

As $\sum_{j\in[1,d_2]} A_{k_{2j}} \supset \{0,1\}^n - D_1^*$ and $|D_1^*| = d_1$, there is a K_2, $1 \leq K_2 \leq d_2$, satisfying $|A_{k_{2K_2}}| \geq \max[\frac{N-d_1}{d_2}, N_{k_{2K_2}}]$. Then in the same way,

$$\Pr[t_i \le t_i' \le 2t_i | D_2] = \sum_{j\in[1,\, d_2]} \Pr[r_i = u_{k_{2j}} | D_2] \cdot \Pr[a_{k_{2j}} \le t_i' \le 2a_{k_{2j}}]$$
$$\ge \sum_{j\in[1,\, d_2]} \frac{1}{d_2}\frac{N_{k_{2j}}}{N} + \frac{1}{d_2 N}\max[\frac{N-d_1}{d_2} - N_{k_{2K_2}}, 0]$$

We now turn to describe the general case. Assume we have constructed $D_1 \cdots, D_{h-1}$ for some $h > 0$. If $D_1 + \cdots + D_{h-1}$ equals the sample space, the decomposition is $\{D_1 \cdots, D_{h-1}\}$. Otherwise, we need to construct D_h. Using the same selection method, we can select a sequence of challenges $\{u_{k_{h1}}, \cdots, u_{k_{hd_h}}\}$ for some $d_h > 0$, $1 \le k_{h1}, \cdots, k_{hd_h} \le N$, satisfying $\sum_{j\in[1,d_h]} A_{k_{hj}}$ contains $\{0,1\}^n - D_1^* - \cdots - D_{h-1}^*$. Let D_h denote the event of $r_i \in \{u_{k_{h1}}, \cdots, u_{k_{hd_h}}\}$, D_h^* denote $\{u_{k_{h1}}, \cdots, u_{k_{hd_h}}\}$. Similarly, we have $d_h < c \log n$.

As $\sum_{j\in[1,d_h]} A_{k_{hj}} \supset \{0,1\}^n - D_1^* - \cdots - D_{h-1}^*$, there is a K_h, $1 \le K_h \le d_h$, satisfying $|A_{k_{hK_h}}| \ge \max[\frac{N-d_1-\cdots-d_{h-1}}{d_h}, N_{k_{hK_h}}]$. Thus,

$$\Pr[t_i \le t_i' \le 2t_i | D_h] = \sum_{j\in[1,\, d_h]} \Pr[r_i = u_{k_{hj}} | D_h] \cdot \Pr[a_{k_{hj}} \le t_i' \le 2a_{k_{hj}}]$$
$$\ge \sum_{j\in[1,\, d_h]} \frac{1}{d_h}\frac{N_{k_{hj}}}{N} + \frac{1}{d_h N}\max[\frac{N-d_1-\cdots-d_{h-1}}{d_h} - N_{k_{hK_h}}, 0]$$

Since the sample space is finite, the construction of the decomposition will halt certainly. Namely, there is a number H satisfying $\sum_{h\in[1,H]} D_h$ contains the sample space. Thus the desired decomposition is $\{D_1, \cdots, D_H\}$. It can be seen $\{D_1^*, \cdots, D_H^*\}$ is actually a decomposition of $\{0,1\}^n$, which shows k_{hj}'s for all $1 \le h \le H, 1 \le j \le d_h$ are a permutation of $1, \cdots, N$. Consequently, by the formula of total probability,

$$\begin{aligned}
\Pr[t_i \le t_i' \le 2t_i] &= \sum_{h\in[1,\, H]} \Pr[D_h] \cdot \Pr[t_i \le t_i' \le 2t_i | D_h] \\
&\ge \sum_{h\in[1,\, H]} \frac{d_h}{N} \cdot \{ \sum_{j\in[1,\, d_h]} \frac{1}{d_h}\frac{N_{k_{hj}}}{N} + \frac{1}{d_h N}\max[\frac{N-d_1-\cdots-d_{h-1}}{d_h} - N_{k_{hk_h}}, 0]\} \\
&= \sum_{h\in[1,\, H]} \sum_{j\in[1,\, d_h]} \frac{N_{k_{hj}}}{N^2} + \sum_{h\in[1,\, H]} \frac{1}{N^2}\max[\frac{N-d_1-\cdots-d_{h-1}}{d_h} - N_{k_{hK_h}}, 0] \\
&= \sum_{k\in[1,\, N]} \frac{N_k}{N^2} + \sum_{h\in[1,\, H]} \frac{1}{N^2}\max[\frac{N-d_1-\cdots-d_{h-1}}{d_h} - N_{k_{hK_h}}, 0] \\
&= \Pr[t_i = t_i'] + \sum_{h\in[1,\, H]} \frac{1}{N^2 d_h}\max[N-d_1-\cdots-d_{h-1} - N_{k_{hK_h}} d_h, 0]
\end{aligned}$$

In the last step of the above formula, we use the fact $\sum_{k\in[1,\, N]} \frac{N_k}{N^2} = \Pr[t_i = t_i']$. This is because $\Pr[t_i = t_i'] = \sum_{k\in[1,\, N]} \Pr[r_i = u_k, t_i' = a_k] = \sum_{k\in[1,\, N]} \Pr[r_i = u_k] \cdot \Pr[t_i' = a_k] = \sum_{k\in[1,\, N]} \frac{N_k}{N^2}$. Now let

us evaluate the second item at the right hand of the last equal sign in the above formula. Since $N_{k_{hK_h}} < \delta N$ (by the assumption of Case 2), $N_{k_{hK_h}} d_h < \delta d_h N < c\delta \log n N$. Let γ denote $c\delta \log n$ and then $\gamma(n) = o(1)$. Then $N_{k_{hK_h}} d_h < \gamma N$ for $1 \leq h \leq H$. Thus (for sufficiently large n's),

$$\begin{aligned}\text{The second item} &= \sum_{h\in[1,\,H]} \frac{1}{N^2 d_h} \cdot \max[N - d_1 - \cdots - d_{h-1} - N_{k_{hK_h}} d_h, 0] \\ &> \frac{1}{N^2 c \log n} \cdot \sum_{h\in[1,\,H]} \max[N - d_1 - \cdots - d_{h-1} - \gamma N, 0] \\ &> \frac{1}{N^2 c \log n} \cdot \sum_{h\in[1,\,H]} \max[(1-\gamma)N - (h-1)c\log n, 0]\end{aligned}$$

Denote by H' the largest integer satisfying $(1-\gamma)N - (h-1)c\log n \geq 0$ for $1 \leq h \leq H'$. Then $H' = \lfloor \frac{1-\gamma}{c\log n} N \rfloor + 1$. Thus (for sufficiently large n's),

$$\begin{aligned}&\text{The second item} > \frac{1}{N^2 c\log n} \cdot \sum_{h\in[1,\,H']} \{(1-\gamma)N - (h-1)c\log n\} \\ &> \frac{1}{N^2 c\log n} \frac{(1-\gamma)^2}{c\log n} N^2 - \frac{1}{N^2 c\log n} \sum_{h\in[1,\,H']} (h-1)c\log n = \frac{(1-\gamma)^2}{(c\log n)^2} - \frac{1}{N^2} \frac{H'(H'-1)}{2} \\ &\geq \frac{(1-\gamma)^2}{(c\log n)^2} - \frac{1}{2N^2}\{\frac{(1-\gamma)^2 N^2}{(c\log n)^2} + \frac{1-\gamma}{c\log n N}\} = \frac{1}{2}\frac{(1-\gamma)^2}{(c\log n)^2} - \frac{1-\gamma}{2N^3 c\log n} > \frac{1}{4c^2\log^2 n} > \delta^2\end{aligned}$$

Thus, $\Pr[t_i \leq t'_i \leq 2t_i] > \Pr[t_i = t'_i] + \delta^2$. Consequently,

$$\Pr[t'_i \leq 2t_i] = \Pr[t'_i < t_i] + \Pr[t_i \leq t'_i \leq 2t_i] > \frac{1}{2}(1 - \Pr[t_i = t'_i]) + \Pr[t_i = t'_i] + \delta^2 = \frac{1}{2} + \frac{1}{2}\Pr[t_i = t'_i] + \delta^2 > \frac{1}{2} + \delta^2$$

Then $\Pr[t'_i > 2t_i] < \frac{1}{2} - \delta^2$. Hence in Case 2, $\Pr[s'_i \leq s_i, t'_i \leq 2t_i, r_i \neq r'_i] > 1 - \frac{1}{2} - (\frac{1}{2} - \delta^2) - 2^{-n} = \delta^2 - 2^{-n}$.

Combining the results of Cases 1 and 2 (and the case the extraction cannot reach the step E sends r_i), we conclude that for any outcome of E's coins used in the i^{th} atomic protocol prior to r_i the extraction succeeds with probability at least $\frac{1}{2}\delta^2 - 2^{-n}$. This means the extraction succeeds in this atomic protocol with probability at least $\frac{1}{2}\delta^2 - 2^{-n}$. Namely, the extraction fails with probability at most $1 - \frac{1}{2}\delta^2 + 2^{-n}$. It follows from $m = \frac{1}{\delta^3}$ that the extraction fails in all atomic protocols with probability at most $(1 - \frac{1}{2}\delta^2 + 2^{-n})^m = O(e^{-1/(2\delta)}) = \mathsf{neg}(n)$. □

Claim 5.5. *Assume there exist one-way functions. Then for each* **NP** *relation R there exists an $\omega(\log n)$-round WI PTSEPOK with time precision poly$(n) + O(ny)$ and space precision poly$(n) + O(y)$ for R.*

Proof. The desired protocol (P, V) and E are constructed in almost the same way as those in the proof of Claim 5.1, except for two differences. One is $m = \omega(\log n)$ herein. The other is E's extraction strategy in the second run. In the extraction of the i^{th} atomic protocol, $1 \leq i \leq m$, E chooses a new challenge $r'_i \in_{\mathrm{R}} \{0,1\}^n$ and sends r'_i to P^*, and recycles the space used in the previous rewinds. The difference is that in the second run E emulates P^* for at most nt_i steps. E does not cut off P^*'s computing iff P^* can finish the computing within nt_i time and its running-space

on $\mathsf{v}_i \circ r'_i$ is no more than s_i. Others remain unchanged. We can show that E has all the required properties using a similar analysis.

Let $(\mathsf{view}_{P^*}, \mathsf{view}_V)$ denote the joint view output by E. It is clear that $(\mathsf{view}_{P^*}, \mathsf{view}_V)$ is identical to the real view in the interaction of $(P^*, V)(x)$. Further, by applying the analysis in the proof of Claim 5.1, we have that E can provide time precision $\mathrm{poly}(n) + O(ny)$ and space precision $\mathrm{poly}(n) + O(y)$. Thus, all that is left to prove is that E succeeds in extraction with overwhelming probability, which can be shown by using the method in the proof of Claim 5.4.

Fix any outcome of E's coins used in the first and second runs of the first $i-1$ atomic protocols that results in E's view in the first run of these protocols is accepting and the extraction fails. Let us analyze the probability of successful extraction in the i^{th} atomic protocol. We will also present a uniform lower bound for this probability when further fixing any outcome of E's coins used in this atomic protocol prior to sending the challenge. (Thus view v_i is determined.) In the same way, we only need to consider the case the extraction can reach the step E sends r_i as follows.

Assuming P^*'s running-time and running-space are bounded by $n^{c-2}-1$ for some non-predetermined constant $c > 0$ (for sufficiently large n's), we set $t_i = n^{c-1} - 1$ and $s_i = n^{c-2}$ (resp. $t'_i = n^{c-1} - 1$ and $s'_i = n^{c-2}$) if P^* outputs an invalid response in the first (resp. second) run. Hence, by the similar analysis in the proof of Claim 5.4, a uniform sufficient condition for the extraction succeeds is $s'_i \leq s_i, t'_i \leq nt_i, r_i \neq r'_i$. Thus, our main task in this proof is to present a lower bound for $\Pr[s'_i \leq s_i, t'_i \leq nt_i, r_i \neq r'_i]$, which is greater than $\geq 1 - \Pr[s'_i > s_i] - \Pr[t'_i > nt_i] - \Pr[r_i = r'_i]$. Since $\Pr[r_i = r'_i] = 2^{-n}$ and $\Pr[s'_i > s_i] < \frac{1}{2}$, we only need to evaluate $\Pr[t'_i > nt_i]$.

For $m = \omega(\log n)$, we can choose two appropriate functions α, δ satisfying $m = \frac{\alpha}{\delta^2}$, $\alpha(n) = \omega(\log n)$ and $\delta(n) = o(1)$. In the following we still divide the evaluation of $\Pr[t'_i \leq nt_i]$ and $\Pr[s'_i \leq s_i, t'_i \leq nt_i, r_i \neq r'_i]$ into two cases, where we still use the notations and variables presented in the proof of Claim 5.4, which admit the same definitions except for A_k. In this proof $A_k \stackrel{\text{def}}{=} \{u \in \{0,1\}^n : a_k \leq t(u) \leq na_k\}$. ($1 \leq a_k < n^{c-1}$ for each k.)

Case 1. Assume there is a k_0 satisfying $\frac{N_{k_0}}{N} \geq \delta(n)$. Using the analysis of Case 1 in the proof of Claim 5.4, we have $\Pr[s'_i \leq s_i, t'_i \leq nt_i, r_i \neq r'_i] > \frac{1}{2}\delta^2(n) - 2^{-n}$.

Case 2. Assume $\frac{N_k}{N} < \delta(n)$ for all k's. In this case $\Pr[t_i \leq t'_i \leq nt_i]$ can be evaluated as follows.

Adopting the same selection method as the proof of Claim 5.4 shows, we can obtain a decomposition $\{D_1, \cdots, D_H\}$ of the sample space and a decomposition $\{D^*_1, \cdots, D^*_H\}$ of $\{0,1\}^n$. It is noticeable that the value of d_h, $1 \leq h \leq H$, differs from it in the proof of Claim 5.4 as we modify the definition of A_k, $1 \leq k \leq N$. We illustrate this difference with respect to d_1. Assume $D^*_1 = \{u_{k_{11}}, \cdots, u_{k_{1d_1}}\}$. Then $a_{k_{11}} < na_{k_{11}} < a_{k_{12}} < na_{k_{12}} < \cdots < a_{k_{1d_1}} < na_{k_{1d_1}}$. It follows $a_{k_{11}} \cdot n^{d_1-1} < a_{k_{1d_1}} < n^{c-1}$. Hence $d_1 < c$. Similarly, we have $d_h < c$ for $1 \leq h \leq H$. According to the analysis in the proof of Claim 5.4, we infer that

$$\Pr[t_i \leq t'_i \leq nt_i] = \Pr[t_i = t'_i] + \sum_{h \in [1,\, H]} \frac{1}{N^2 d_h} \cdot \max[N - d_1 - \cdots - d_{h-1} - N_{i_{hk_h}} d_h, 0]$$

Now let us evaluate the second item at the right hand of the equal sign. Since $N_{i_{hk_h}} < \delta N$, $N_{i_{hk_h}} d_h < \delta d_h N < c\delta N$. Let $\gamma = c\delta$. This shows $N_{i_{hk_h}} d_h < \gamma N$. Thus (for sufficiently large n's),

$$\text{The second item} = \sum_{h\in[1,\,H]} \frac{1}{N^2 d_h} \cdot \max[N - d_1 - \cdots - d_{h-1} - N_{i_{hk_h}} d_h, 0]$$

$$> \frac{1}{N^2 c} \cdot \sum_{h\in[1,\,H]} \max[(N - d_1 - \cdots - d_{h-1} - \gamma N), 0] > \frac{1}{N^2 c} \cdot \sum_{h\in[1,\,H]} \max[(1-\gamma)N - (h-1)c, 0]$$

Denote by H' the largest value satisfying $(1-\gamma)N-(h-1)c \geq 0$ for $1 \leq h \leq H'$ (and sufficiently large n's). Then $H' = \lfloor \frac{1-\gamma}{c} N \rfloor + 1$. Thus (for sufficiently large n's),

$$\text{The second item} > \frac{1}{N^2 c} \cdot \sum_{h\in[1,\,H']} \{(1-\gamma)N - (h-1)c\} > \frac{1}{N^2 c} \frac{(1-\gamma)^2}{c} N^2 - \frac{1}{N^2 c} \sum_{h\in[1,\,H']} (h-1)c$$

$$= \frac{(1-\gamma)^2}{c^2} - \frac{1}{N^2} \frac{H'(H'-1)}{2} \geq \frac{(1-\gamma)^2}{c^2} - \frac{1}{2N^2} \{ \frac{(1-\gamma)^2 N^2}{c^2} + \frac{1-\gamma}{cN} \} = \frac{1}{2} \frac{(1-\gamma)^2}{c^2} - \frac{1-\gamma}{2N^3 c}$$

$$> \frac{1}{2c^2} - \frac{\gamma}{c^2} - \frac{1-\gamma}{2N^3 c} > \frac{1}{3c^2} > \delta$$

Then $\Pr[t_i \leq t'_i \leq nt_i] > \Pr[t_i = t'_i] + \delta$. Hence

$$\Pr[t'_i \leq nt_i] = \Pr[t'_i < t_i] + \Pr[t_i \leq t'_i \leq nt_i] > \frac{1}{2}(1 - \Pr[t_i = t'_i]) + \Pr[t_i = t'_i] + \delta \geq \frac{1}{2} + \delta$$

Thus, $\Pr[t'_i > nt_i] < \frac{1}{2} - \delta$. So in Case 2 $\Pr[s'_i \leq s_i, t'_i \leq nt_i, r_i \neq r'_i] > 1 - \frac{1}{2} - (\frac{1}{2} - \delta) - 2^{-n} = \delta - 2^{-n}$.

Combining the results from Cases 1 and 2 (and the case the extraction cannot reach the step E sends r_i) we conclude that for any outcome of E's coins used in the i^{th} atomic protocol prior to r_i E succeeds in extraction in the i^{th} atomic protocol with probability at least $\frac{1}{2}\delta^2 - 2^{-n}$. This means the extraction succeeds in this atomic protocol with probability at least $\frac{1}{2}\delta^2 - 2^{-n}$. Namely, E fails with probability at most $1 - \frac{1}{2}\delta^2 + \mathsf{neg}(n)$. It follows from $m = \frac{\alpha}{\delta^2}$, E fails in all atomic protocols with probability at most $(1 - \frac{1}{2}\delta^2 + 2^{-n})^m = O(e^{-\frac{\alpha}{2}}) = \mathsf{neg}(n)$. □

Claim 5.6. *Assume there exist constant-round perfectly-hiding commitments. Then for each* **NP** *relation R there exists an $\omega(\log^3 n)$-round (resp. $\omega(\log n)$-round) perfectly WI PTSEAOK with time precision poly$(n) + O(y)$ (resp. poly$(n) + O(ny)$) and space precision poly$(n) + O(y)$ for R.*

Proof. Instantiate the commitment scheme in Blum's proof for R with a constant-round perfectly-hiding commitment scheme. Then using the proofs of Claim 5.1, Claim 5.4 and Claim 5.5, we infer this claim holds. Details omitted. □

Combining Claim 5.6 with Claim 4.3, we have Theorem 1.1 follows. Combining Claim 5.1 and Claim 5.5 with Claim 4.5, we have Theorem 1.2 follows.

References

[1] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. In Proc. 42nd FOCS, IEEE, pages 106-115, 2001.

[2] B. Barak, S. J. Ong, and S. Vadhan. Derandomization in Cryptography, In Proc. Crypto'03, Springer-Verlag, LNCS 2729, pages 299-315, 2003.

[3] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. Lecture Notes in Computer Science 740, pages 390-420, 1993.

[4] M. Blum. Coin Flipping by Phone. In Proc. 24th Computer Conference, IEEE, pages 133-137, 1982.

[5] M. Blum. How to Prove a Theorem So No One Else Can Claim It. In Proc. the International Congress of Mathematicians, Berkeley, California, USA, pages 1444-1451, 1986.

[6] G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. J. Comput. Syst. Sci., 37(2): 156-189, 1988.

[7] I. Damgård, T. Pedersen and B. Pfitzmann. On the Existence of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures. In Proc. Crypto'93, Springer-Verlag, LNCS 773, pages 250-265, 1993.

[8] U. Feige, A. Fiat and A. Shamir. Zero-Knowledge Proofs of Identity. Journal of Cryptology, 1(2):77-94, 1988.

[9] U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. In Proc. 22nd STOC, ACM, pages 416-426, 1990.

[10] U. Feige, A. Shamir, Zero Knowledge Proofs of Knowledge in Two Rounds. In Proc. Crypto'89, Santa Barbara, California, USA, Springer-Verlag, LNCS 435, pages 526-544, 1989.

[11] O. Goldreich. Foundations of Cryptography - Basic Tools. Cambridge University Press, Cambridge, 2001.

[12] O. Goldreich and L. A. Levin. A Hard-Core Predicate for All One-Way Functions. In Proc. 21st STOC, ACM, pages 25-32, 1989.

[13] O. Goldreich, S. Micali and A. Wigderson. Proofs That Yields Nothing But Their Valid or All Languages in **NP** Have Zero-Knowledge Proof Systems. Journal of the ACM, 38(3): 169-192, 1991.

[14] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In Proc. 17th STOC, ACM, pages 291-304, 1985.

[15] S. Micali and R. Pass. Local Zero Knowledge. In Proc. 38th STOC, ACM, pages 306-315, 2006.

[16] M. Naor. Bit Commitment Using Pseudorandomness. Journal of Cryptology, 4(2):151-158, 1991.

[17] M. Naor, R. Ostrovsky, R. Venkatesan and M. Yung. Zero-Knowledge Arguments for **NP** Can Be Based on General Assumptions. J. Cryptology, 11, pages 87-108, 1998.

[18] M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. In Proc. 21st STOC, ACM, pages 33-43, 1989.

[19] R. Pass. A Precise Computational Approach to Knowledge, Dissertation for the Doctoral Degree, MIT, 2006.

[20] M. Tompa and H. Woll. Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information. In Proc. 28th FOCS, IEEE, pages 472-482, 1987.