SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2014, Vol. 57 012112:1–012112:14
doi: 10.1007/s11432-013-5044-x

Constant-round zero-knowledge proofs of knowledge with strict polynomial-time extractors for NP

LI HongDa[1]* & FENG DengGuo[2]

[1]*State Key Lab of Information Security, Institute of Information Engineering of Chinese Academy of Sciences, Beijing* 100093*, China;*
[2]*Institute of software of Chinese Academy of Sciences, Beijing* 100080*, China*

Received July 7, 2013; accepted October 5, 2013

Abstract Barak and Lindell showed that there exist constant-round zero-knowledge arguments of knowledge with strict polynomial-time extractors. This leaves the open problem of whether it is possible to obtain an analogous result regarding constant-round zero-knowledge proofs of knowledge for NP. This paper focuses on this problem and gives a positive answer by presenting a construction of constant-round zero-knowledge proofs of knowledge with strict polynomial-time extractors for NP.

Keywords zero-knowledge proofs, proofs of knowledge, constant-round, strict polynomial-time extractors

Citation Li H D, Feng D G. Constant-round zero-knowledge proofs of knowledge with strict polynomial-time extractors for NP. Sci China Inf Sci, 2014, 57: 012112(14), doi: 10.1007/s11432-013-5044-x

1 Introduction

Goldwasser et al. [1] first introduced Zero-knowledge proofs. An interactive proof for a language L is called a zero-knowledge proof system if a non probabilistic polynomial-time verifier obtains information additional to the validity of the proven statement. More formally, the zero-knowledge property requires that for any probabilistic polynomial-time verifier there exists an efficient algorithm Sim_{V^*} (known as a simulator) which, only accessing the common input, can simulate everything that V^* obtains in the interaction with the prover. The soundness property of ZKP is required to protect an honest verifier from an all-powerful prover. Zero-knowledge arguments (ZKA) are a relaxation of ZKP, in which the soundness property is required to hold only with respect to a computationally bounded prover

Proofs of knowledge, defined by Goldwasser et al. [1], are proofs that allow the prover to convince the verifier of holding a secret witness w about a given common input x. There have been several attempts to present an adequate formalization for this [2–4]. What does it mean to say an interactive proof system is a proof of knowledge? Loosely speaking, a knowledge extractor $\mathcal{K}$ exists that, when given access to the prover's strategy, can output the secret within a reasonable time related to the probability that the prover convinces the verifier. There are two equivalent definitions: (1) there exists an expected polynomial-time extractor $\mathcal{K}$ that can obtain a witness with probability sufficiently close to the probability that the prover

---

*Corresponding author (email: lihongda@iie.ac.cn) convinces the verifier; and (2) an extractor $\mathcal{K}$ exists that must always output a witness w within an expected time inversely proportional to the probability that the prover convinces the verifier. Naturally, if a proof (or argument) of knowledge system for relation R is a zero-knowledge protocol for the language L_R induced by R, it is known as a zero-knowledge proof (or argument) of knowledge for R. ZKP (or ZKA) of knowledge have since played a crucial role in the design of cryptographic schemes and protocols.

Goldreich et al. [5] first presented a 5-round black-box zero-knowledge proof system for NP under the existence of claw-free functions. Subsequently, Rosen constructed an even simpler 7-round black-box zero-knowledge proof system for HC assuming the existence of two-round perfect-hiding commitment schemes [6]. Until Barak's result in [7], all known constant-round zero-knowledge protocols (proofs or arguments) allow simulators to run in expected polynomial-time (that is, the expectation of running time is polynomial). On the other hand, Goldreich et al. [8] proved that 3-round black-box ZKP do not exist for any language outside of bounded-error probabilistic polynomial time (BPP). Recently, Katz proved that NP-complete languages do not have 4-round black-box ZKP assuming the polynomial hierarchy does not collapse [9]. This result indicates that the round complexity of the construction in [5] for black-box simulation is optimal unless coNP $\subseteq$ MA.

Regarding proofs of knowledge, constant-round ZKA of knowledge for NP are known to exist [10]. In [11], the authors extends impossibility results from [8] to zero-knowledge proof of knowledge, and prove that the existence of 3-round black-box ZKP of knowledge for L implies there exists a PPT algorithm which, taking as input $x \in L$, can output a witness for $x \in L$ with overwhelming probability. Some known constant-round construction of ZKP for NP, such as ones in [5,6], are not proofs of knowledge. Moreover, Barak et al. [12,13] proved that no constant-round zero-knowledge strong proofs or arguments of knowledge exist for a non-trivial language. By using a special (non-black-box) simulation techniques, Ref. [14] recently presented a constant-round construction of ZKP of knowledge for NP. Subsequently, Ref. [15] presented 5-round black-box ZKP of knowledge for NP1).

Before the work of [16], all known constant-round ZKA of knowledge allowed the knowledge extractors to run in expected polynomial-time. In [16], this was shown to be necessary for a black-box knowledge extractor, and a constant-round construction of zero-knowledge argument of knowledge with a strict polynomial-time non-black-box knowledge extractor was presented. However, the recent constant-round constructions of ZKP of knowledge for NP allow the knowledge extractors to run in expected polynomial-time [15]. An interesting problem left by this work is whether it is possible to obtain constant-round ZKP of knowledge with strict polynomial-time extractors. We will focus on this problem and give such a construction.

1.1 Related works

Barak [7] presented the first construction of constant-round ZKA for NP with a strict polynomial-time simulator. However, Barak's construction is non-black-box zero-knowledge, as the simulator for the construction in [7] uses the description of the code of the verifier's strategy. Subsequently, Barak et al. [16] showed that such non-black-box simulation is necessary for obtaining constant-round ZKP or ZKA for any language outside of BPP. Regarding zero-knowledge protocols of knowledge, Ref. [16] proved that, under the existence of one-way functions, it is impossible to achieve constant-round zero-knowledge arguments of knowledge with a strict polynomial-time black-box knowledge extractor for any language outside of BPP. Thus, non-black-box extraction of knowledge defined in [17] is necessary in order to obtain constant-round ZKA of knowledge with a strict polynomial-time knowledge extractor. Using such a non-black-box knowledge extractor, Ref. [16] presented a construction of constant-round ZKA of knowledge with a PPT knowledge extractor under the assumption of the existence of trapdoor permutations and collision-resistant hash functions. The recent constant-round constructions of ZKP of knowledge for NP allow the knowledge extractors to run in expected polynomial-time [15].

---

1) In 2010, Lindell presents an analogous construction: see http://eprint.iacr.org/2010/656.

1.2 Our main results

This paper focuses on the constant-round ZKP of knowledge with PPT knowledge extractors for NP. The main contribution of this paper is to show a construction of constant-round ZKP of knowledge for NP with polynomial-time extractors for NP.

Main Theorem: Every NP problem has a constant-round ZKP of knowledge system with a strict PPT non-black-box knowledge extractor if collision-resistant hash functions, claw-free functions and enhanced trapdoor permutations exist.

Our construction of constant-round ZKP of knowledge relies mainly on special commit-with-extract schemes. To construct such a commit-with-extract scheme, we use Barak's non-black-box simulation techniques. Instead of directly using constant-round zero-knowledge arguments (with a PPT simulator) to prove that the committed is correctly opened as in [16], the receiver in our construction uses a special statistical WI argument of knowledge to prove that it executes the protocol honestly or knows the next-message function of the sender. The other ingredients to construct such a commit-with-extract scheme include Blum's non-interactive commitment scheme, a general (2-round) statistically hiding commitment scheme and an augmented coin-tossing protocol. Roughly speaking, by incorporating the non-black-box techniques with an augmented coin-tossing protocol, the presented scheme only uses a special statistical WI argument in the augmented coin-tossing protocol to prove that the committed is correctly opened or the transcript of the coin-tossing protocol satisfies a specified condition that holds with a negligible probability in real execution. This special statistical WI argument is based on public-coin universal arguments [18] and general statistical WI arguments of knowledge for NP as in [19].

2 Preliminaries

Some standard notations is used in this paper.

If A is a PPT machine or algorithm, $A(x, y, r)$ is the output of A upon input x, auxiliary input y and random input r. For a finite set S, we mean by $y \in_R S$ that y is uniformly selected from S. For two probability ensembles $\{X_\sigma\}_{\sigma \in I}$ and $\{Y_\sigma\}_{\sigma \in I}$, we denote by $\{X_\sigma\} \overset{c}{=} \{Y_\sigma\}$ (for simplicity, $X_\sigma \overset{c}{=} Y_\sigma$) that the ensembles $\{X_\sigma\}_{\sigma \in I}$ and $\{Y_\sigma\}_{\sigma \in I}$ are computationally indistinguishable

A non-negative function $\mu(\cdot) : N \to R$ is called negligible if for every positive polynomial p, there exists $N_p > 0$ such that for all $n > N_p$, $\mu(n) < 1/p(n)$. Throughout this paper, $\mu(\cdot)$ and poly$(\cdot)$ will respectively denote an unspecified negligible function and an unspecified polynomial.

2.1 Zero-knowledge

We recall the definitions of zero-knowledge. These formal definitions are taken from [3].

Let P and V be a pair of interactive Turing machines, $\langle P, V\rangle(x)$ be a random variable representing the local output of Turing machine V interacting with machine P on common input x, when the random input to each machine is uniformly and independently chosen. Customarily, machine P is called the prover and machine V the verifier. By $\langle P, V\rangle(x) = 1$ ($\langle P, V\rangle(x) = 0$), we mean that V accepts (rejects) the proofs given by P.

Definition 1. A pair of interactive Turing machines $\langle P, V\rangle$ is called an interactive proof system for a language L if V is polynomial-time and the following two conditions hold:

1) Completeness: there exists a negligible function c such that for every $x \in L$, $\Pr[\langle P, V\rangle(x) = 1] > 1 - c(|x|)$.

2) Soundness: there exists a negligible function s such that for every $x \notin L$ and every interactive machine B, $\Pr[\langle B, V\rangle(x) = 1] < s(|x|)$.

$c(\cdot)$ is called the completeness error, and $s(\cdot)$ the soundness error. If the soundness condition is required to hold only with respect to a PPT prover, $\langle P, V\rangle$ is called an interactive arguments system for L.

An interactive proof is said to be auxiliary-input zero-knowledge if the interaction between the prover (with some auxiliary input y) and verifier (with any auxiliary input z) reveals nothing beyond the validity of the assertion to be proved to the verifier. This is formalized by requiring that for any polynomial-time verifier V^* with any auxiliary input z there exists a polynomial-time algorithm $\mathcal{S}_{V^*}$ (a.k.a the simulator) such that the view of V^* can be simulated by $\mathcal{S}_{V^*}(x,z)$. The idea behind this definition is that whatever V^* might have learned from interacting with P, it could have actually been obtained by itself. For convenience, we denote by $\text{View}_V\langle P(y), V(z)\rangle(x)$ a random variable describing the view of V, including the content of the random tape of V^* and the messages that V receives from P. Similarly, $\text{View}_P\langle P(y), V(z)\rangle(x)$ is a random variable describing the view of P.

Definition 2 (Zero-Knowledge with respect to auxiliary input)**.** Let $\langle P, V\rangle$ be an interactive proof system for a language L. Denote by $P_L(x)$ the set of string y satisfying the completeness condition with respect to $x \in L$. $\langle P, V\rangle$ is called an auxiliary-input zero-knowledge proof system if for every PPT machine V^*, there exists a PPT algorithm $\mathcal{S}_{V^*}$ such that, for arbitrary $y_x \in P_L(x)$, $\{\text{View}_{V^*}\langle P(y_x), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$ and $\{\mathcal{S}_{V^*}(x,z)\}_{x\in L, z\in\{0,1\}^*}$ are computationally indistinguishable.

- **Black-box zero-knowledge versus non-black-box zero-knowledge.** In general, the simulator $\mathcal{S}_{V^*}$ in the above definition is dependent on the strategy of verifier V^*. A zero-knowledge proof system is termed black-box if the simulator $\mathcal{S}_{V^*}$ only uses the algorithm V^* in a "black-box" manner. That is, there exists a "universal" simulator $\mathcal{S}$ such that for every $x \in L$ and every PPT verifier V^*, the oracle algorithm $\mathcal{S}^{V^*(x,z)}(x)$ can simulate $\text{View}_{V^*}\langle P(y_x), V^*(z)\rangle(x)$. A zero-knowledge proof system is termed a non-black-box if the simulator $\mathcal{S}_{V^*}$ not only uses the algorithm V^* as a black-box subroutine, but also uses the description of the algorithm V^*, denoted by desc(V^*), as its input. That is, there exists a PPT algorithm $\mathcal{S}$ such that for arbitrary $y_x \in P_L(x)$, $\{\text{View}_{V^*}\langle P(y_x), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$ and $\{\mathcal{S}_{V^*}(\text{desc}(V^*), 1^t, x, z)\}_{x\in L, z\in\{0,1\}^*}$ are computationally indistinguishable, where t is a bound on the running time of V^* upon inputs x, z.

2.2 Witness indistinguishable proof

The notion of witness indistinguishable proof (argument) requires that no information about which witness is being used by provers be revealed to verifiers. For any $L \in NP$, denote by $R_L(x)$ the corresponding NP-relation induced by L.

Definition 3. An interactive proof (argument) system $\langle P, V\rangle$ for a NP language L is (statistical) witness indistinguishable (WI) if for every pair of witness sequences $\{w_x^1\}_{x\in L}, \{w_x^2\}_{x\in L}$ such that $(x, w_x^1), (x, w_x^1) \in R_L$ and any polynomial-time machine V^*, two distributions, $\{\text{View}_{V^*}\langle P(w_x^1), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$ and $\{\text{View}_{V^*}\langle P(w_x^2), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$ are computationally (statistically) indistinguishable.

If a witness indistinguishable system $\langle P, V\rangle$ is also a proof of knowledge, $\langle P, V\rangle$ is said to be witness indistinguishable proof (argument) of knowledge. Constant-round statistical WI arguments for any $L \in$ NP are well known to exist.

2.3 Proof of knowledge

In a proof of knowledge for a relationship R, the prover holding a witness w such that $R(x, w) = 1$ interacts with the verifier on a common input x. The goal of the protocol is to convince the verifier that the prover indeed knows w. This is in contrast to a regular interactive proof, where the verifier is just convinced of the validity of the proved statement.

The concept of "knowledge" for machines is formalized by stating that if a prover can convince the verifier, then an efficient procedure (known as knowledge extractor) exists that can "extract" a witness from this prover (thus the prover knows a witness because it could run the extraction procedure on itself).

The standard definition of proofs of knowledge in [3] requires the existence of a universal knowledge extractor that use the algorithm of the prover only as a black-box. Our definition of proofs of knowledge slightly differs from the standard definition. Concretely speaking, the knowledge extractor of our definition is given the description of the prover's circuit (or algorithm), and so is called non-black-box extraction. Requiring that the knowledge extractor is a PPT algorithm, we need to assume that the description of the prover's circuit is polynomially bounded.

This non-black-box extraction was first used in [17] to obtain resettable ZKA of knowledge for NP. Subsequently, Ref. [16] also used the non-black-box-extraction definition of arguments of knowledge to present the first constant-round zero-knowledge argument of knowledge with strict polynomial-time extractors for NP, and proved that the use of non-black box extraction is essential for obtaining a constant-round argument (or proof) of knowledge with strict polynomial-time extractor. Our following definition is on proofs of knowledge.

Definition 4. An interactive protocol $\langle P, V\rangle$ is a system of proofs of knowledge for a (poly-balanced) relation R with knowledge error κ if the following conditions hold:

1) Efficiency: $\langle P, V\rangle$ is polynomially bounded, and V is computable in PPT.

2) Non-triviality: There exists an interactive machine P such that for every $(x, w) \in R$ all possible interactions of V with P on common input x and auxiliary y are accepting.

3) Validity: There exists a polynomial-time machine $\mathcal{K}$, such that for any interactive machine P^*, and every $x, y, r \in \{0,1\}^*$, the machine $\mathcal{K}$, upon input $(\mathrm{desc}(P^*), x, r)$, outputs a solution w satisfying $R(x, w) = 1$ with probability at least $p(x, y, r) - \kappa(|x|)$, where $p(x, y, r)$ is the probability that the interactive machine V on input x accepts the proof when interacting with the prover specified by $P^*_{x,y,r}$ (the prover's strategy when fixing common x, auxiliary input y and random tape r).

2.4 Universal arguments

The notion of universal arguments, introduced by Barak et al. [18], is used to present proof system for language $L_{\mathcal{U}} = \{(M, x, t) :$ non-deterministic machine M accepts x within t steps$\}$. The corresponding relation is denoted by $\mathcal{R}_{\mathcal{U}}$, that is, $\mathcal{R}_{\mathcal{U}}((M, x, t), w) = 1$ if and only if M (viewed here as a two-input deterministic machine) accepts (x, w) within t steps. We can handle all NP language by a universal argument for $L_{\mathcal{U}}$, because every language in NP is linear time reducible to $L_{\mathcal{U}}$.

Definition 5 (Universal arguments, [18])**.** A universal-argument system is a pair of strategies, denoted $\langle P, V\rangle$, that satisfies the following properties:

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

2) Completeness by a relatively-efficient prover: for every $((M, x, t), w)$ in $\mathcal{R}_{\mathcal{U}}$, $\Pr[\langle P(w), V\rangle(M, x, t) = 1] = 1$. Furthermore, there exists a polynomial p such that the total time spent by $P(w)$, on common input (M, x, t), is at most $p(T_M(x, w)) \leqslant p(t)$, where $T_M(x, w)$ denote the number of steps made by M on input (x, w).

3) Computational soundness: for every polynomial-size circuit family $\{\widetilde{P}_n\}_{n\in N}$, and every $(M, x, t) \in \{0,1\}^n - L_{\mathcal{U}}$, $\Pr[\langle \widetilde{P}_n, V\rangle(M, x, t) = 1] < \mathrm{neg}(n)$.

4) A weak proof of knowledge property: for every positive polynomial p there exists a positive polynomial p' and a PPT oracle machine E such that the following holds: For every polynomial-size circuit family $\{\widetilde{P}_n\}_{n\in N}$, and every sufficiently long $y = (M, x, t) \in \{0,1\}^*$ if $\Pr[\langle \widetilde{P}_n, V\rangle(y) = 1] > 1/p(|y|)$ then $\Pr[\exists w = w_1 \cdots w_t \in \mathcal{R}_{\mathcal{U}}(y)\ s.\ t.\ \forall i \in [t], E^{\widetilde{P}_n}(y, i) = w_i] > 1/p'(|y|)$. The oracle machine E is called a (knowledge) extractor.

2.5 Commitment schemes

Commitment schemes are used to enable a party (the sender) to commit itself to a value while keeping it secret from another party (the receiver). This property is called hiding. Furthermore, the commitment is binding, and thus, at a later stage when the commitment is opened, it is guaranteed that the opening can yield only a single value determined in the committing phase. Given below is a sketch of the two properties. General definitions can be found in [3].

1) Statistically binding commitments: the binding property holds against unbounded senders, while the hiding property only holds against computationally bounded receivers.

2) Statistically hiding commitments: the hiding property holds against unbounded receivers, while the binding property only holds against computationally bounded senders.

In this paper, we will use non-interactive statistically binding commitments and constant-round statistically hiding commitments. It is well known that noninteractive statistically binding commitment schemes can be constructed using any one-to-one one-way functions (see Subsection 4.4.1 of [3]). Statistically hiding commitments need stronger assumptions, for example, some number-theoretic complexity assumptions. In [20], a construction of statistically hiding commitments is presented under a general assumption of the existence of any one-way permutation. Haitner et al. [21] improved upon this result by showing a construction based on the assumption of the existence of one-way function. However, these known constructions are not constant-round protocols although constant-round statistically hiding commitment schemes exist under stronger assumptions. In fact, Haitner et al. [22] gave a lower bound of round complexity for a fully-black-box construction of a statistically hiding commitment scheme from trapdoor permutations showing a constant-round statistically hiding scheme cannot be constructed from one-way permutations. On the other hand, Ref. [5] presented a two-round construction of statistically hiding commitments by means of claw-free functions.

Here and throughout this paper, we denote by $\mathrm{Comm}(\cdot;\cdot)$ a non-interactive computationally hiding commitment scheme, and assume that a two-round construction of statistically hiding commitments is as follows: the receiver first sends a random index $\theta \in \Theta$, and then the sender's commitment to u is computed by $\mathrm{Comm}_\theta(u;s)$, where $s \in_R \{0,1\}^{\mathrm{poly}(|u|)}$. The decommitment of such a two-round scheme consists of the committed value and a random input of $\mathrm{Comm}_\theta(\cdot;\cdot)$.

- **Commit-with-extract schemes.** Commit-with-extract schemes were first introduced to construct constant-round ZKA of knowledge with a strict polynomial-time extractor in [16], and were used to construct the constant-round non-malleable commitment scheme in [23]. Loosely speaking, a commitment scheme is a commit-with-extract scheme if an extractor exists that can simulate the interaction with the sender, and extract a committed value that is determined by the transcript of the simulated interaction. To obtain the committed value in strict PPT, the extractor uses the strategy of the sender in a non-black-box manner.

Definition 6 (Commit-with-extract, [16])**.** A statistically binding commitment scheme Comm-Ext with sender S and receiver R is a commit-with-extract scheme if the following holds: there exists a probabilistic polynomial-time commitment extractor $\mathcal{K}_c$ such that for every probabilistic polynomial-time committing party S^* and for every $x, y, r \in \{0,1\}^*$, upon input $(\mathrm{desc}(S^*), 1^t, x, r)$, where t is a bound on the running time of $S^*(x,y,r)$, machine $\mathcal{K}_c$ outputs a pair, denoted by $(\mathcal{K}_c^1, \mathcal{K}_c^2) = (\mathcal{K}_c^1(\mathrm{desc}(S^*), 1^t, x, y, r);\ \mathcal{K}_c^2(\mathrm{desc}(S^*); 1^t, x, y, r))$, satisfying the following conditions:

1) $\{\mathcal{K}_c^1(\mathrm{desc}(S^*), 1^t, x, y, r)\}_{x,y,r\in\{0,1\}^*} \equiv \{\mathrm{View}_{S^*}\langle S^*(x,y,r); B\rangle\}_{x,y,r\in\{0,1\}^*}$.

2) $\Pr[\mathcal{K}_c^2(\mathrm{desc}(S^*); 1^t, x, y, r) = \text{commit-value}(K_c^1)] > 1 - (|x|)$, where commit-value$(K_c^1)$ denotes the committed value determined by K_c^1 , and commit-value$(K_c^1) = \perp$ if no committed value exists.

3 A new commit-with-extract scheme

The construction of ZKA of knowledge for an NP-relation R in [16] consists of two phases. In phase 1, the prover and verifier run a special (statistically binding) commitment scheme, known as a commit-with-extract scheme, in which the prover commits to the witness w satisfying $R(x,w) = 1$. In phase 2, they run a constant-round zero-knowledge argument for NP in which the prover proves that the value being committed to during the commitment in the previous phase is a valid witness of $x \in L_R$. The central tool in their construction is the constant-round (statistically binding) commit-with-extract scheme. Compared with general commitment schemes, commit-with-extract schemes have the following additional property: an extractor exists that can extract the value being committed to during the commitment stage. The fact that the construction is zero-knowledge results from the hiding property of the commitment scheme and the zero-knowledge property of the proof system used in phase 2. The knowledge extractor of the construction can be derived from the extraction property of the commit-with-extract scheme.

Their construction of a commit-with-extract scheme is derived from following Blums non-interactive commitment scheme: to commit to bit σ, the sender randomly picks r and sends $(f(r), b(r) \oplus \sigma)$ to the receiver, where f is a one-way permutation and $b(\cdot)$ is its hard-core predicate. The details of the commitment stage of the construction are as follows:

1) The sender chooses a permutation f.

2) Both the parties run a coin-tossing protocol to generate a uniformly random string y: a) The receiver randomly selects two strings y_1, s_1 and sends the sender $c = \mathrm{Comm}(y_1; s_1)$ (the commitments to y_1). b) The sender sends a random string y_2. c) The receiver opens the commitment c: only revealing y_1 and proving, by a constant-round zero-knowledge argument with a strict polynomial-time simulator, that the commitment c defined in the previous step is the commitment to y_1. d) Both the parties set $y = y_1 \oplus y_2$.

3) The sender computes $r = f^{-1}(y)$ and sends $b(r) \oplus \sigma$ to the receiver.

In the coin-tossing protocol, instead of revealing the commitment c, the receiver only reveals y_1 and invokes a (constant-round) zero-knowledge argument to prove it is consistent with c. Note that the commitment-value extractor works in the simulated setting, and it extracts the committed value by revealing a new string y'_1 such that it knows $f^{-1}(y'_1 \oplus y_2)$, and then invoking the simulator of this zero-knowledge argument system to prove a false statement: c is a correct commitment to y'_1. From the assumption that the sender is a computational bounded machine, the sender will always accept the proof given by the simulator. However, to extract the committed value in strict polynomial-time this simulator must be a strict polynomial-time algorithm. Here, Barak's first constant-round (non-black-box) zero-knowledge argument with a strict polynomial-time simulator can be used, and as a result in the knowledge extractor must work in a non-black-box manner.

To construct a zero-knowledge proof of knowledge for an NP-relation R with a strict polynomial-time (non-black-box) knowledge extractor, we need a constant-round statistically binding commit-with-extract scheme, the commitment-value extractor of which must succeed even if the sender is a computationally unbounded machine. It is unfortunate that such statistically-binding commit-with-extract schemes cannot be directly derived from the above commit-with-extract scheme by means of replacing $\mathrm{Comm}(\cdot;\cdot)$ with a statistically hiding commitment scheme.

Next, we will construct such a constant-round statistically binding commit-with- extract scheme. The presented commitment scheme is similar to the construction in [16], and it consists of an augmented coin-tossing protocol and Blums non-interactive commitment. Other than replacing the computationally hiding commitment scheme with a two-rounds statistically hiding commitment scheme, there are two differences between the presented scheme and that from [16]: (1) a new augmented coin-tossing is used; and so (2) instead of using a constant-round zero-knowledge argument (with PPT simulator) for the verifier to prove that the previous committed value is correctly opened, the proposed scheme only uses a special statistical WI argument in the augmented coin-tossing protocol for the verifier to prove that the committed value is correctly opened or the transcript of the augmented coin-tossing protocol satisfies a specified condition.

The construction uses any family of enhanced trapdoor permutations, that consists of four PPT algorithms: a function-sampling algorithm I, a domain-sampling algorithm D_f for the specified permutation by I, a permutation-computing algorithm f and permutation-inverting algorithm f^{-1}. For convenience we use $D_f(r)$ to denote the deterministic algorithm derived from D_f when treating the coins, denoted by r, as an auxiliary input. Let $b(\cdot)$ be a hard-core predictor of the enhanced trapdoor permutations.

Define $T(n) = n^{\omega(1)}$ and let $\{\mathcal{H}_n\}$ be a family of collision-resistant hash functions against $T(n)$-size circuits, where $h \in \mathcal{H}_n : \{0,1\}^* \rightarrow \{0,1\}^n$. Given an enhanced trapdoor permutation f, h and $\mathrm{Comm}_\theta(\cdot;\cdot)$, for convenience, define two languages:

$$\Lambda_0 = \left\{ \begin{array}{l} (c, y, w) : 1)\ c \in \{0,1\}^{\mathrm{poly}(n)}, y \in \{0,1\}^n; \\ \qquad\qquad 2)\ \exists z \in \{0,1\}^n \text{ and } s_1 \in \{0,1\}^{\mathrm{poly}(n)}, \text{ such that} \\ \qquad\qquad\quad a)\ c = \mathrm{Comm}_\theta(z; s_1);\ b)\ w = D_f(y \oplus z). \end{array} \right\}$$

$$\Lambda_1 = \left\{ \begin{array}{l} (h,c,y): 1)\ (h,c,y) \in \mathcal{H}_n \times \{0,1\}^{\mathrm{poly}(n)} \times \{0,1\}^n; \\ \qquad\qquad 2)\ \exists d \in \{0,1\}^n,\ \text{and TM } \pi, s_2 \in \{0,1\}^{\mathrm{poly}(n)} \text{ such that} \\ \qquad\qquad\quad a)\ d = h(\pi);\ b)\ c = \mathrm{Comm}_\theta(d; s_2);\ c)\ \pi(c) \text{ output } y \text{ within } T(n) \text{ steps.} \end{array} \right\}$$

Obviously, $(c, y, w) \in \Lambda_0$ means that the random input used to select w is determined by y and the committed value of c. Whereas $(h, c, y) \in \Lambda_1$ means there exists a Turing machine π that, upon inputting the commitment to $h(\pi)$, will output y.

The details of the proposed construction are as follows:

Construction 1 (Commit-with-extract scheme)**.** The sender commits to $\sigma \in \{0,1\}$; n is a security parameter.

Commitment stage:

1) The sender randomly picks an enhanced trapdoor permutation f by the function-sampling algorithm I, a hash function $h \in \mathcal{H}_n$, and $\theta \in \Theta$ (the index for the two-round statistically hiding commitment schemes).

2) The receiver randomly selects $z \in \{0,1\}^n$ and $s_1 \in \{0,1\}^{\mathrm{poly}(n)}$, computes $c = \mathrm{Comm}_\theta(z; s_1)$ and sends c to the prover P.

3) The sender randomly picks $y \in \{0,1\}^n$ and sends y to the receiver.

4) The receiver computes $w = D_f(y \oplus z)$ and sends w to the sender.

5) By running a special WI argument (given in construction 2) with a common input $\lambda = (h, c, y, w)$, the receiver proves that $(c, y, w) \in \Lambda_0$ or $(h, c, y) \in \Lambda_1$.

6) If the above proof fails, the sender aborts; Otherwise, the sender computes $r = f^{-1}(w)$ and sends $\rho = b(r) \oplus \sigma$ to the receiver.

Revealment stage: To reveal the commitment, the sender sends σ and r to the receiver. The receiver verifies $f(r) = w$ and $\sigma = \rho \oplus b(r)$.

The special WI argument used in the protocol deals with language $\Lambda_0 \vee \Lambda_1$. Note that Λ_1 does not lie in NP under the assumption that $T(n) = n^{\omega(1)}$. So general WI arguments for NP do not work and a special WI universal argument is needed.

Additionally, it follows from the statistically hiding property of $\mathrm{Comm}_\theta(\cdot;\cdot)$ that $(c, y, w) \in \Lambda_0$ and $(h, c, y) \in \Lambda_1$ hold almost certainly. Therefore, to resist a cheating receiver this special WI universal argument used in step 5 should be a "proof-of-knowledge", that is, it should be a statistical WI universal argument of knowledge. In such a way, the probability that the proof is accepted by the sender is almost same as the probability that the receiver knows the witness for $(c, y, w) \in \Lambda_0$ or $(h, c, y) \in \Lambda_1$. Thus, if the sender accepts the proof with a non-negligible probability then the receiver must obtain, with a non-negligible probability, the witness for $(c, y, w) \in \Lambda_0$ or $(h, c, y) \in \Lambda_1$. The former means, by the definitions of Λ_0, that the receiver selects w according to the protocol (otherwise the receiver is able to attack the binding property of $\mathrm{Comm}_\theta(\cdot;\cdot)$), and then the receiver has no information of $f^{-1}(w)$ as f is an enhanced trapdoor permutation. The later shows by the definition of Λ_1 that the receiver can guess y (the output of the sender) with non-negligible probability in advance. This is impossible in real setting. Therefore, the sender accepts the proof with a non-negligible probability must mean that the receiver honestly executes the protocol and then has no information of $f^{-1}(w)$. The computationally hiding property of Construction 1 follows.

Our construction of the special WI universal arguments of knowledge for $\Lambda_0 \vee \Lambda_1$, denoted by $\langle P_{\mathrm{swi}}, V_{\mathrm{swi}} \rangle$, is similar to the construction in [18] and two ingredients, Barak's universal argument and a statistical WI argument of knowledge for NP, are used. Let $\langle P_{\mathrm{ua}}, V_{\mathrm{ua}} \rangle$ be Barak's 4-round, public-coin universal argument for Λ_1 (for simplicity, denote the interaction by (1)$P_{\mathrm{ua}} \overset{\alpha}{\leftarrow} V_{\mathrm{ua}}$; (2) $P_{\mathrm{ua}} \overset{\beta}{\rightarrow} V_{\mathrm{ua}}$; (3) $P_{\mathrm{ua}} \overset{\gamma}{\leftarrow} V_{\mathrm{ua}}$; (4) $P_{\mathrm{ua}} \overset{\delta}{\rightarrow} V_{\mathrm{ua}}$), and let $\langle P_{\mathrm{wi}}, V_{\mathrm{wi}} \rangle$ be a statistical WI argument of knowledge for NP. Without loss of generality, the length of the messages in $\langle P_{\mathrm{ua}}, V_{\mathrm{ua}} \rangle$ has upper bound from above by the length of the common input. The details of $\langle P_{\mathrm{swi}}, V_{\mathrm{swi}} \rangle$ are as follows.

Construction 2 (Special WI argument for $\Lambda_0 \vee \Lambda_1$: $\langle P_{\mathrm{swi}}, V_{\mathrm{swi}} \rangle$)**.** Common inputs: (h, c, y, w) and index θ.

Phase 1: encrypted $\langle P_{\mathrm{ua}}, V_{\mathrm{ua}} \rangle$ for Λ_1. Let $|(h, c, y)| = n$.

1) The verifier randomly selects $\alpha \in \{0,1\}^n$ and sends α to the prover.

2) The prover sends $\widetilde{\beta} = \mathrm{Comm}_\theta(0^n; \tau_1)$ to the verifier, where $\tau_1 \in_R \{0,1\}^{\mathrm{poly}(n)}$.

3) The verifier randomly selects $\gamma \in \{0,1\}^n$ and sends γ to the prover.

4) The prover sends $\widetilde{\delta} = \mathrm{Comm}_\theta(0^n; \tau_2)$ to the verifier, where $\tau_2 \in_R \{0,1\}^{\mathrm{poly}(n)}$.

Phase 2: Both parties run statistical (black-box) WI arguments of knowledge $\langle P_{\mathrm{wi}}, V_{\mathrm{wi}} \rangle$, with common input $((h, c, y, w), (\alpha, \widetilde{\beta}, \gamma, \widetilde{\delta}))$, in which the prover proves the OR of the following statements:

1) $(c, y, w) \in \Lambda_0$.

2) There exists $(\beta, \gamma, \tau_1, \tau_2)$ such that a) $\widetilde{\beta} = \mathrm{Comm}_\theta(\beta; \tau_1)$; b) $\widetilde{\gamma} = \mathrm{Comm}_\theta(\gamma; \tau_2)$; c) $(\alpha, \beta, \gamma, \delta)$ is an accepting transcript for $\langle P_{\mathrm{ua}}, V_{\mathrm{ua}} \rangle$ proving that $(h, c, y) \in \Lambda_1$.

Next, we will show that construction 1 is a commit-with extract scheme. It is easy to see that an honest sender can commit to any bit. The computationally hiding property of the scheme, roughly speaking, follows from the following two facts: (1) Blum's scheme is computationally hiding; (2) Λ_1 is a hard language, that is, no matter what the verifier does in the real interaction, $(h, c, y) \in \Lambda_1$ with at most a negligible probability. On the other hand, when given the description of the sender's strategy the extractor is able to make $(h, c, y) \in \Lambda_1$ hold, and can then honestly complete step 5 using a witness for $(h, c, y) \in \Lambda_1$. Thus, the extractor can extract the committed value by choosing r and setting $w = f(r)$.

Theorem 1. Let $\langle P_{\mathrm{ua}}, V_{\mathrm{ua}} \rangle$, $\langle P_{\mathrm{wi}}, V_{\mathrm{wi}} \rangle$ and $\mathrm{Comm}_\theta(\cdot;\cdot)$ are as above. Construction 1, using the special WI argument given by construction 2 in step 5, is a constant-round statistically binding commit-with-extract scheme.

Proof. We will first prove that construction 1 is a statistically binding scheme, and then prove that it satisfies definition 6.

• **Computationally hiding property.** From the property of Blum's commitment scheme, it is easy to see that the computationally hiding property holds if the augmented coin-tossing protocol is secure; i.e., w is generated by the receiver according to the protocol specification. As the sender cannot directly verify whether the receiver selects w honestly, the receiver is asked to prove that at least one of $(c, y, w) \in \Lambda_0$ (i.e. w is generated according to the protocol specification) and $(h, c, y) \in \Lambda_1$ (i.e. the transcript meets some special conditions) holds by means of the special WI universal arguments of knowledge for $\Lambda_0 \vee \Lambda_1$. Note that $(c, y, w) \in \Lambda_0$ and $(h, c, y) \in \Lambda_1$ all are hold almost certainly (because $\mathrm{Comm}_\theta(\cdot;\cdot)$ is statistically-hiding). Therefore, we need to prove the following: if the sender accepts the proof of step 5 with a noticeable probability then the receiver knows a witness for $(c, y, w) \in \Lambda_0$ with a noticeable probability, which means that the receiver selects w according to the protocol specification (otherwise, the receiver with a noticeable probability knows a witness for $(c, y, w) \in \Lambda_0$ while he does not follow the protocol specification to generate w. This means that the receiver is able to attack the binding property of $\mathrm{Comm}_\theta(\cdot;\cdot)$).

To this end, we need to prove that, if the sender accepts the proof in step 5 with a noticeable probability, there exists an extractor $\mathcal{K}$ that can output a witness for $(c, y, w) \in \Lambda_0$ with a noticeable probability when given access to the strategy of the (possibly cheating) receiver R^*. In the following we will show the existence of $\mathcal{K}$.

Roughly speaking, $\mathcal{K}$ only needs to simulate the interaction with R^* by playing an honest sender. As the sender only sends random messages before reaching the phase 2, R^* cannot distinguish this simulated interaction from the real one. $\mathcal{K}$ proceeds as follows:

Extractor $\mathcal{K}$: 1) Emulate an execution from step 1 to 4 of construction 1 by playing the role of the sender. If the emulation fails at any time, output $\perp$ and halt. 2) Emulate an execution of encrypted $\langle P_{\mathrm{ua}}, V_{\mathrm{ua}} \rangle$ by playing the role of the sender (that is, a verifier of the special WI argument). If the emulation fails at any time, output $\perp$ and halt. 3) Construct a prover (denoted by P^*_{wi}) and a verifier (denoted by V_{wi}) of the WI argument of knowledge system $\langle P_{\mathrm{wi}}, V_{\mathrm{wi}} \rangle$, respectively from the residual receiver R^* and the residual sender S with a fixed view so far. 4) Invoke the extractor $\mathcal{E}_{\mathrm{wi}}$ for $\langle P^*_{\mathrm{wi}}, V_{\mathrm{wi}} \rangle$. Finally, output the output of $\mathcal{E}_{\mathrm{wi}}$.

Let p_c be the probability that the sender S, interacting with the (possible) cheating receiver R^*, accepts the proof given by R^* in step 5 of construction 1, and $p_{\rm wi}$ be the probability that $V_{\rm wi}$ (the residual sender of S) accepts the proof given by $P_{\rm wi}$ (the residual receiver of R^*) in phase 2 of the special WI argument. Obviously, $p_c \leqslant p_{\rm wi}$. It follows that if p_c is noticeable then so is $p_{\rm wi}$.

Let Γ denote the event that $\mathcal{K}$ completes steps 1 and 2 without aborting. If Γ takes place, $P^*_{\rm wi}$ (a prover of $\langle P_{\rm wi}, V_{\rm wi}\rangle$ in the simulation setting) is well defined from the residual receiver R^* with a fixed view until now. Assume that $p^*_{\rm wi}$ is the probability that $P^*_{\rm wi}$ succeeds in convincing $V_{\rm wi}$ in the emulation process. As the sender only needs to send random messages to the receiver, the probability that Γ take place is the same as the probability that the protocol $\langle P_{\rm wi}, V_{\rm wi}\rangle$ is reached in the real execution. This shows that $|p_{\rm wi} - p^*_{\rm wi}|$ must be negligible. Hence, $p^*_{\rm wi}$ is noticeable if p_c is a noticeable.

From the proof-of-knowledge property of $\langle P^*_{\rm wi}, V_{\rm wi}\rangle$ (with common input $((h, c, y, w), (\alpha, \widetilde{\beta}, \gamma, \widetilde{\delta})))$, there exists an expected polynomial-time extractor $\mathcal{E}_{\rm wi}$ such that, with a noticeable probability, $\mathcal{E}_{\rm wi}$ (when given access to $P^*_{\rm wi}$) can output a witness (z, s_1) (for $(c, y, w) \in \Lambda_0$) satisfying $c = {\rm Comm}_\theta(z; s_1)$ and $w = D_f(y \oplus z)$, or outputs a witness $(\beta, \tau_1, \delta, \tau_2)$ such that the following conditions hold: a) $\widetilde{\beta} = {\rm Comm}_\theta(\beta; \tau_1)$; b) $\widetilde{\delta} = {\rm Comm}_\theta(\delta; \tau_2)$; c) $(\alpha, \beta, \gamma, \delta)$ is an accepting transcript for $\langle P_{\rm ua}, V_{\rm ua}\rangle$ proving that $(h, c, y) \in \Lambda_1$.

Next, we will prove the probability, denoted by p_e, that $\mathcal{E}_{\rm wi}$ outputs a accepting transcript $(\beta, \tau_1, \delta, \tau_2)$ is negligible. $\mathcal{K}$ then outputs a witness for $(c, y, w) \in \Lambda_0$ with a noticeable probability.

Let n' be the length of the common input of $\langle P^*_{\rm wi}, V_{\rm wi}\rangle$ ($n' = {\rm poly}(n)$). Suppose to the contrary that for infinitely many values of n' (denoted by N_1), p_e is noticeable, i.e. $p_e > 1/{\rm poly}(n')$ for $n' \in N_1$. Then, we can construct $P^*_{\rm ua}$, a strict polynomial-time prover of the universal argument $\langle P_{\rm ua}, V_{\rm ua}\rangle$, from $\langle P_{\rm swi}, V_{\rm swi}\rangle$ and $\mathcal{E}_{\rm wi}$ (the extractor of knowledge of $\langle P_{\rm wi}, V_{\rm wi}\rangle$), such that the probability with which $P^*_{\rm ua}$ convince $V_{\rm ua}$ of $(h, c, y) \in \Lambda_1$ is noticeable for infinitely many value of n'' (here, n'' is the length of the common input for $\langle P_{\rm ua}, V_{\rm ua}\rangle$ and $n'' = {\rm poly}(n)$), that is, the following holds $p_{\rm ua} = \Pr[\langle P^*_{\rm ua}, V_{\rm ua}\rangle(h, c, y) = 1] = {\rm poly}(p_e) \geqslant 1/{\rm poly}(n'')$, $n'' \in N_2$ We first construct $P^{*'}_{\rm ua}$ as follows:

1) $P^{*'}_{\rm ua}$ first receives message α from verifier $V_{\rm ua}$. To generate response β, $P^{*'}_{\rm ua}$ starts to emulate the execution of $\langle P_{\rm swi}, V_{\rm swi}\rangle$ by playing the role of an honest verifier $V_{\rm swi}$. $P^*_{\rm ua}$ proceeds as follows: a) Sends α to $P_{\rm swi}$ and obtains $\widetilde{\beta}$ from $P_{\rm swi}$. b) Randomly selects γ and then sends it to $P_{\rm swi}$. $P_{\rm swi}$ responds with $\widetilde{\delta}$. c) Executes protocol $\langle P_{\rm wi}, V_{\rm wi}\rangle$ with prover $P_{\rm wi}$, and abort if $P_{\rm wi}$ fails in convincing of $P^{*'}_{\rm ua}$; otherwise, invokes extractor $\mathcal{E}_{\rm wi}$. d) If $\mathcal{E}_{\rm wi}$ outputs a witness $(\beta, \delta, \tau_1, \tau_2)$ such that $(\alpha, \beta, \gamma, \delta)$ is an accepting transcript for $\langle P_{\rm ua}, V_{\rm ua}\rangle$ proving that $(h, c, y) \in \Lambda_1$, proceeds to the next step; otherwise, aborts.

2) $P^{*'}_{\rm ua}$ sends β to $V_{\rm ua}$ and receives γ from $V_{\rm ua}$.

3) To generate response δ, $P^*_{\rm ua}$ rewinds $P_{\rm swi}$ to the point where it waits for the response message γ and emulates the execution of $\langle P_{\rm swi}, V_{\rm swi}\rangle$ again. $P^*_{\rm ua}$ proceed as follows: a) sends γ to $P_{\rm swi}$ and receives $\widetilde{\delta}$ from $P_{\rm swi}$; b) executes protocol $\langle P_{\rm wi}, V_{\rm wi}\rangle$ with prover $P_{\rm wi}$, and aborts if $P_{\rm wi}$ fails in convincing of $P^{*'}_{\rm ua}$; otherwise, invokes the extractor $\mathcal{E}_{\rm wi}$; c) if $\mathcal{E}_{\rm wi}$ again outputs a witness $(\beta', \delta', \tau'_1, \tau'_2)$ such that $(\alpha, \beta', \gamma, \delta')$ is an accepting transcript proving that $(h, c, y) \in \Lambda_1$, proceed to the next step; otherwise, aborts.

4) If $\beta \neq \beta'$, $P^{*'}_{\rm ua}$ aborts; otherwise, $P^{*'}_{\rm ua}$ sends δ to $V_{\rm ua}$.

Note that $P_{\rm swi}$ succeeds in convincing $V_{\rm swi}$ with a larger probability than p_e. It is obvious that $P^{*'}_{\rm ua}$ reaches step 4 without aborting with probability $p = {\rm poly}(p_e) \geqslant 1/{\rm poly}(n') \geqslant 1/{\rm poly}(n'')$. On the other hand, because $\beta \neq \beta'$ means that $P^{*'}_{\rm ua}$ is able to open the commitment $\widetilde{\beta}$ as two different values, then from the computationally binding property of ${\rm comm}_\theta(\cdot;\cdot)$, $\Pr[\beta \neq \beta']$ must be negligible. Therefore, $P^{*'}_{\rm ua}$ can convince $V_{\rm ua}$ with probability $p - \mu(n') \geqslant 1/{\rm poly}(n') \geqslant 1/{\rm poly}(n'')$ (because $(\alpha, \beta, \gamma, \delta')$ is an accepting transcript). Clearly, $P^{*'}_{\rm ua}$ is only expected polynomial-time because the extractor $\mathcal{E}_{\rm wi}$ is expected polynomial-time. Nevertheless, we can obtain a strict polynomial-time $P^*_{\rm ua}$, which can convince $V_{\rm ua}$ with a non-negligible probability, by truncating the execution of $P^{*'}_{\rm ua}$.

By the weak proof of knowledge property of $\langle P_{\rm ua}, V_{\rm ua}\rangle$, there exists a PPT oracle machine $E^{P^*_{\rm ua}}$ who can extract any bit of a witness for $(h, c, y) \in \Lambda_1$ with a probability $q \geqslant 1/{\rm poly}(n'') \geqslant 1/{\rm poly}(n)$. Furthermore, because the length of d and s_2 is bounded above by a fixed polynomial, d and s_2 can be exactly extracted by $E^{P^*_{\rm ua}}$ in polynomial-time.

Because the probability that the receiver correctly guesses y in advance is negligible, there must exist many y's (at least a noticeable fraction $1/{\rm poly}(n)$) such that $E^{P^*_{\rm ua}}$ can give an implicit representation of a witness with probability $q \geqslant 1/\mathrm{poly}(n'') \geqslant 1/\mathrm{poly}(n)$. Thus, $E^{P^*_{\mathrm{ua}}}$ can obtain the implicit presentations of the witness for $(h,c,y) \in \Lambda_1$ (denoted by (d,π,s_2)) and the witness for $(h,c,y') \in \Lambda_1$ (denoted by (d',π',s'_2)) for uniformly distributed y and y' with a noticeable probability. By definition, we have $d = h(\pi),\ c = \mathrm{Comm}_\theta(d,s_1),\ y = \pi(c)$ and $d' = h(\pi'),\ c = \mathrm{Comm}_\theta(d',s'_1),\ y' = \pi(c')$. As $\Pr[y = y'] = 2^{-n}$, $\pi \neq \pi'$ holds with probability $1-\mu(n)$. On the one hand, $d = h(\pi) \neq h(\pi') = d'$ implies that the receiver is able to open c in two different ways. This contradicts the binding property of $\mathrm{Comm}_\theta(\cdot;\cdot)$. On the other hand, $h(\pi) = h(\pi')$ means that $E^{P^*_{\mathrm{ua}}}$ can find a collision of h with a noticeable probability and contradicts the assumption. Therefore, p_{ua} must be negligible.

More formally, we construct a non-uniform strategy $\mathcal{A}$ from $E^{P^*_{\mathrm{ua}}}$ to attack the binding property of $\mathrm{Comm}_\theta(\cdot;\cdot)$ or to find collisions for $\{\mathcal{H}_n\}$. The details of $\mathcal{A}$ are as follows:

1) Incorporate the receiver's strategy R^* and emulate the honest sender S until the special WI argument protocol $\langle P_{\mathrm{wi}}, V_{\mathrm{wi}}\rangle$ is reached.

2) Invoke the knowledge extractor $E^{P^*_{\mathrm{ua}}}$, fail and halt if $E^{P^*_{\mathrm{ua}}}$ fails.

3) Otherwise, obtain d, s_2 and an implicit representation of the Turing machine π.

4) Rewind R^* until the point where y is expected.

5) After receiving w, invoke the knowledge extractor $E^{P^*_{\mathrm{ua}}}$, fail and halt if $E^{P^*_{\mathrm{ua}}}$ fails.

6) Otherwise, obtain d', s'_2 and an implicit representation of the Turing Machine π'.

7) If $d \neq d'$, output $(d,s_2),(d',s'_2)$ and halt.

8) Otherwise, obtain π and π' from the respective implicit representation in time $T(n)^{O(1)}$.

From the above we know that $E^{P^*_{\mathrm{ua}}}$ succeeds with a noticeable property if p_e is noticeable. Indeed, $\mathcal{A}$ with a non-negligible property obtains either (d,s_2) and (d',s'_2) satisfying $d \neq d'$ and $\mathrm{comm}_\theta(d;s_2) = \mathrm{comm}_\theta(d';s'_2)$, or π and π' satisfying $\pi \neq \pi'$ and $h(\pi) = h(\pi')$. This contradicts the binding property of $\mathrm{Comm}_\theta(\cdot;\cdot)$ or the collision property of $\{\mathcal{H}_n\}$. Therefore, p_e is negligible.

• **Binding property.** The statistically binding property is derived from that of Blum's commitment scheme.

• **Extraction property.** To prove the extraction property, we show that there exists a polynomial-time extractor $\mathcal{K}_c$ which, given the polynomial bounded description of the sender's strategy , can extract the committed value from S^*.

In this instance, $\mathcal{K}_c$ should play the role of receiver and simulate the interaction with S^*. From the hiding property of $\mathrm{Comm}_\theta(\cdot;\cdot)$, S^* cannot distinguish this simulated interaction from the real one. Thus, the behavior of S^* is almost the same in the real setting as in the simulated setting.

This extractor $\mathcal{K}_c$ works if the following two conditions hold: (1) it knows the pre-image of w under f and (2) it can convince the sender of the statement that $(c,y,w) \in \Lambda_0$ or $(h,c,y) \in \Lambda_1$. Obviously, this is impossible in a real setting (the computationally binding property). This however can be achieved in this simulation because taking as input the description of the sender's strategy (for simplicity, denoted by $\mathrm{desc}(\pi)$), $\mathcal{K}_c$ has an advantage over the real receiver. To make condition (1) hold, $\mathcal{K}_c$ needs to choose a string in r advance and then to uses $w = f(r)$ to respond to y. To make condition (2) hold, the extractor $\mathcal{K}_c$ only needs to make $(h,c,y) \in \Lambda_1$ hold by computing $c = \mathrm{Comm}_\theta(h(\mathrm{desc}(\pi));s_1)$ for $s_1 \in_R \{0,1\}^{\mathrm{poly}(n)}$ (here, assuming $\mathrm{desc}(\pi)$) is polynomially bounded).

In more details, $\mathcal{K}_c$ taking as input π proceeds as follows:

1) $\mathcal{K}_c$ receives a trapdoor permutation f, a hash function h, and an index θ.

2) $\mathcal{K}_c$ randomly picks s_2, computes $c = \mathrm{Comm}_\theta(h(\pi);s_1)$, and sends c to the sender.

3) $\mathcal{K}_c$ receives y.

4) $\mathcal{K}_c$ randomly chooses r and sets $w = f(r)$.

5) Playing the role of a prover, $\mathcal{K}_c$ honestly runs the special WI argument (given in construction 2) with $(\pi, d = h(\pi), s_2)$, a witness for $(h,c,y) \in \Lambda_1$, to prove $((c,y,w) \in \Lambda_0) \vee ((h,c,y) \in \Lambda_1)$. If the proof succeeds, $\mathcal{K}_c$ will receive the sender's commitment ρ.

6) Upon receiving ρ, $\mathcal{K}_c$ computes and outputs $\sigma = \rho \oplus b(r)$.

Note that $c = \mathrm{Comm}_\theta(h(\mathrm{desc}(\pi));s_1)$ is enough to make $(h,c,y) \in \Lambda_1$ hold. Thus, although $\mathcal{K}_c$ deviate from the protocol specification by computing $w = f(r)$, $\mathcal{K}_c$ has a witness for $((c,y,w) \in \Lambda_0) \vee (h,c,y) \in \Lambda_1$, by which $\mathcal{K}_c$ can convince the sender of $((c,y,w)\in\Lambda_0)\vee(h,c,y)\in\Lambda_1$ in step 5. Therefore, $\mathcal{K}_c$ can receive ρ and compute $\sigma=\rho\oplus b(r)$.

Remark 1. The above analysis is based on the assumption that $\{\mathcal{H}_n\}$ is collision resistant against $T(n)^{O(1)}$-size circuits. By the methods of [18], this assumption can be weakened to the assumption that is collision resistant against polynomial-size circuits.

Remark 2. Construction 1 is only a bit commitment scheme, but we can obtain a string commitment scheme by running it several times in parallel. Concretely speaking, to commit $w=w_1\cdots w_k\in\{0,1\}^k$, the sender and the receiver run k instances of the bit commitment protocol in parallel, where instance i is used to commit w_i. Nevertheless, there exist two exceptions: (1) the sender use the same h, f, and θ in all instance; (2) as Barak's non-black-box technique is used, a single special WI argument is necessary in step 5. To this end, we define a language:

$$\Lambda_1'=\left\{\begin{array}{r}(h,\overline{c},\overline{y}):\ 1)\ h\in\mathcal{H}_n,\overline{c}=(c_1,\ldots,c_k)\in\{0,1\}^{k\cdot\mathrm{poly}(n)},\ \text{and}\ \overline{y}=(y_1,\ldots,y_k)\in\{0,1\}^{k\cdot n}.\\ 2)\ \exists\ \text{an TM}\ \pi\ \text{and}\ (s_2^1,\ldots,s_2^k)\in\{0,1\}^{k\cdot\mathrm{poly}(n)},\ \text{such that}\ a)\ d=h(\pi);\\ b)\ c_j=\mathrm{Comm}_\theta(d;s_2^j),j=1,\ldots,k;\ c)\ \pi(\overline{c})\ \text{outputs}\ \overline{y}\ \text{within}\ T(n)\ \text{steps.}\end{array}\right\}$$

The details of the string commitment scheme are as follows:

Construction 3 (String commit-with-extract scheme)**.** The sender commits to $\sigma=\sigma_1\cdots\sigma_k\in\{0,1\}^k$. n is a security parameter.

Commitment stage:

1) The sender randomly picks an enhanced trapdoor permutation f by I, a hash function $h\in\mathcal{H}_n$, and an index θ (the index of two-round statistically hiding commitment scheme).

2) The receiver randomly selects $z_j\in\{0,1\}^n$ and $s_1^j\in\{0,1\}^{\mathrm{poly}(n)}$, computes $c_j=\mathrm{Comm}_\theta(z_j;s_1^j)$, $j=1,\ldots,k$. Then, it sends $c=(c_1,\ldots,c_k)$ to the sender S.

3) The sender randomly picks $\overline{y}=(y_1,\ldots,y_k)\in\{0,1\}^{k\cdot n}$ and sends $\overline{y}$ to the receiver.

4) The receiver computes $w_j=D_f(y_j\oplus z_j)$ and sends $\overline{w}=(w_1,\cdots,w_k)$ to the sender.

5) By running a special WI argument (given in construction 2) with a common input $\lambda=(h,\overline{c},\overline{y},\overline{w})$, the receiver proves that (1) $(c_j,y_j,w_j)\in\lambda_0$ for $j=1,\ldots,k$, or (2) $(h,\overline{c},\overline{y})\in\Lambda_1'$.

6) If the above proof fails, the sender aborts; otherwise, the sender computes $r_j=f^{-1}(w_j)$ and $\rho_j=b(r_j)\oplus\sigma_j$. Finally, the sender sends $\rho=\rho_1\cdots\rho_k$ to the receiver.

Revealment stage: To reveal the commitment, the sender sends σ and $r=(r_1,\ldots,r_k)$ to the receiver. The receiver verifies $f(r_j)=w_j$ and $\sigma_j=\rho_j\oplus b(r_j)$, $j=1,\ldots,k$.

The proof of construction 3 is analogous to that of construction 1. For convenience, the commitment to a string w is denoted by $c=\text{Commit-Extract}(w;r)$, where r is the decommitment of c.

4 Constant-round zero-knowledge proofs of knowledge with PPT extractor

In this section, we will use the proposed commit-with-extract scheme to construct a constant-round zero-knowledge proof-of-knowledge system with PPT extractor for NP.

Construction 4. Zero-knowledge proof of knowledge.

Common input: $x\in L$.

Auxiliary input to the prover: a witness w such that $R_L(x,w)=1$.

Part 1: Both parties run the above commit-with-extract protocol, in which the prover commits to the witness w, denoted by $\rho=\text{Commit-Extract}(w;r)$. Let v be all messages received by the receiver in this stage.

Part 2: a) The prover and the verifier run a general zero-knowledge proof Π, in which the prover (with common input (x,ρ) and auxiliary input (w,r)) proves to the verifier (with auxiliary v) that there exist w and r such that the following two conditions hold: (1) ρ is a commitment to w; (2) $R_L(x,w)=1$, where R_L is the poly-balanced relation derived by L. b) The verifier accepts if and only if it accepts the above proof.

Theorem 2. Construction 4 is a Zero-knowledge proof of knowledge for NP satisfying the definition 3 under the assumed conditions.

Proof. Completeness: This follows immediately from the definitions.

Validity: We need to construct a knowledge extractor $\mathcal{K}_z$ satisfying definition 3. In fact, all $\mathcal{K}_z$ does is to invoke the extractor of the commit-with-extract scheme. Concretely, the knowledge extractor $\mathcal{K}_z(\text{desc}(P^*), x, r)$ proceeds as follows:

1) Invoke the extractor $\mathcal{K}_z$ of the commit-with-extract protocol and obtain the committed string w. 2) If $R_L(x, w) = 1$, output w; otherwise, abort.

Assume that the prover P^* succeeds in convincing the honest verifier of $x \in L$ with probability p. We show that $\mathcal{K}_z$ outputs a valid witness for $x \in L$ with probability at lest $p - \mu(|x|)$.

In fact, the only difference between executing $\mathcal{K}_z$ and executing the commit-with-extract protocol is that the receiver completes the WI argument with two different witnesses. This means that the residual prover P^{**} generated by executing $\mathcal{K}_z$ can successfully convince the verifier with almost same probability as a real prover. That is, the probability that P^{**} convinces the verifier, denoted by p', is at least $p - \mu(|x|)$. If $\mathcal{K}_z(\text{desc}(P^*), x, r)$ outputs w with probability at most $p - 1/\text{poly}(|x|)$, the residual prover P^{**} can convince the verifier of a false statement with non-negligible probability. This contradicts the soundness of the zero-knowledge proof system used in part 2. Therefore, the probability that $\mathcal{K}_z$ outputs a valid witness for $x \in L$ is at least $p - \mu(|x|)$.

- **Zero-knowledge.** By the definition of the zero-knowledge property, we need to present a simulator $\mathcal{S}_{V^*}$ for any given polynomial-time verifier V^*. Because of the hiding property of Blum's commitment scheme, $\mathcal{S}_{V^*}$ can play an honest prover in part 1 to interact with V^*, although $\mathcal{S}_{V^*}$ does not know any witness. In fact, $\mathcal{S}_{V^*}$ only needs to commit to a dummy string, such as $0^{\text{poly}(|x|)}$. Thus, by the soundness of the zero-knowledge proof system used in part 2, $\mathcal{S}_{V^*}$ cannot complete the proof of part 2. Let v' denote all the messages sent (by $\mathcal{S}_{V^*}$) to V^* before reaching part 2 in the simulated setting. We know the strategy of the residual verifier $V^*(v')$ is the same as that of $V^*(v)$, where v denotes all the messages V^* received during executing part 1 in the real setting. Hence, $\mathcal{S}_{V^*}$ can invoke the simulator for $V^*(v)$ to simulate the proof of part 2.

Let $\mathcal{O}$ be the simulator for the zero-knowledge proof system used in part 2. For any (possibly cheating) verifier V^*, the simulator $\mathcal{S}_{V^*}$ proceeds as follows:

1) $\mathcal{S}_{V^*}$ plays the role of a prover to run part 1. Instead of committing to a witness, $\mathcal{S}_{V^*}$ commits to $0^{\text{poly}(|x|)}$ by $\rho' = \text{Commit-Extract}(0^{\text{poly}(|x|)}; r')$.

2) Let v' denote all the messages sent to V^* during the running of part 1. $\mathcal{S}_{V^*}$ invokes the simulator $\mathcal{O}$ for the zero-knowledge proof system executed in part 2 with (x, ρ'), denoted as $\mathcal{O}_{V^*(v')}(x, \rho')$, where $V^*(v')$ plays the role of the verifier.

3) $\mathcal{S}_{V^*}$ outputs v' and $\mathcal{O}_{V^*(v')}(x, \rho')$.

By the definition, we have to prove that $\{\mathcal{S}_{V^*}(x)\} \overset{c}{=} \{\text{View}_{V^*}\langle P, V^*\rangle(x)\}$. To this end, we define a hybrid simulator $\widetilde{\mathcal{S}}$ which completes part 1 using the strategy of an honest prover and complete part 2 as $\mathcal{S}_{V^*}$ does.

First, note that $\text{View}_{V^*}\langle P, V^*\rangle(x) = (v, \text{View}_{V^*(v)}\langle P(w, r), V^*(v)\rangle(x, \rho))$ and thus it follows from the zero-knowledge property of Π that $\{\text{View}_{V^*}\langle P, V^*\rangle(x) \overset{c}{=} \{\widetilde{\mathcal{S}}(x)\} = \{(v, \mathcal{O}_{V^*(v)}(x, \rho))\}$. In addition, the hiding property of the presented commit-with-extract scheme means that the strategy of residual $V^*(v')$ is same as the strategy of the residual verifier $V^*(v)$. Thus, we have $\{\mathcal{S}_{V^*}(x)\} = \{(v', \mathcal{O}_{V^*(v')}(x, \rho'))\} \overset{c}{=} \{(v, \mathcal{O}_{V^*(v)}(x, \rho))\} = \{\widetilde{\mathcal{S}}(x)\}$. Over all, we have $\{\mathcal{S}_{V^*}(x)\} \overset{c}{=} \{\text{View}_{V^*}\langle P, V^*\rangle(x)\}$.

It is well known that 4-round public-coin universal arguments exist if collision-resistant hash functions exist and that the existence of claw-free functions means the the existence of a 2-round statistically hiding commitment scheme and statistical WI arguments of knowledge for NP. Therefore, we obtain our main theorem from Theorems 1 and 2.

Acknowledgements

This work was partially supported by National Basic Research Program of China (973 program) (Grant No. 2007CB311202), National Natural Science Foundation of China (Grant No. 60970139), Strategic Priority Program of Chinese Academy of Sciences (Grant No. XDA06010702), and IIEs Cryptography Research Project.

References

1 Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems. SIAM J Comput, 1989, 18: 186–208
2 Bellare M, Goldreich O. On defining proofs of knowledge. Lect Notes Comput Sci, 1992, 740: 390–420
3 Goldreich O. Foundations of Cryptography—Basic Tools. Cambridge: Cambridge University Press, 2001
4 Bellare M, Goldreich O. On probabilistic versus deterministic provers in the definition of proofs of knowledge. Lect Notes Comput Sci, 2011, 6650: 114–123
5 Goldreich O, Kahan A. How to construct constant-round zero-knowledge proof system for NP. J Cryptol, 1996, 9: 167–189
6 Rosen A. A note on constant-round zero-knowledge proofs for NP. Lect Notes Comput Sci, 2004, 2951: 191–202
7 Barak B. How to go beyond the black-box simulation barrier. In: 42th Annual Syposium on Foundation of Computing Science, Las Vegas, 2001. 106–115
8 Goldreich O, Krawczyk H. On the composition of zero-knowledge proof systems. SIAM J Comput, 1996, 25: 169-192
9 Katz J. Which languages have 4-round zero-knowledge proofs. In: 5th Theory of Cryptography Conference, New York, 2008. 73–88
10 Feige U, Shamir A. Zero knowledge proofs of knowledge in two rounds. In: Proceedings of CRYPTO'89, Santa Barbara, 1989. 526–545
11 Toshiya I, Kouichi S. On the complexity of constant round ZKIP of possession of knowledge. IEICE Trans Fund, 1993, E76-A: 31–39
12 Barak B, Lindell Y, Vadhan S. Lower bounds for non-black-box zero knowledge. In: 44th Annual IEEE Symposium Foundations of Computer Science, Cambridge, 2003. 384–393
13 Barak B, Lindell Y, Vadhan S. Lower bounds for non-black-box zero knowledge. J Comput Syst Sci, 2006, 72: 321–391
14 Li H D, Xu H X, Li B, et al. On constant-round zero-knowledge proofs of knowledge for NP-relation. Sci China Inf Sci, 2010, 53: 788–899
15 Li H D, Feng D G, Li B, et al. Round-optimal zero-knowledge proofs of knowledge for NP. Sci China Inf Sci, 2012, 55: 2473–2484
16 Barak B, Lindell Y. Strict polynomial-time in simulation and extractor. In: 34th ACM Symposium on the Theory of Computing, Montreal, 2002. 484–493
17 Barak B, Goldreich O, Goldwasser S, et al. Resettably-sound zero-knowledge and its application. In: 42th Annual Syposium on Foundation of Computing Science, Las Vegas, 2001. 116–125
18 Barak B, Goldreich O. Universal arguments and their applications. In: Proceedings of 17th IEEE Annual Conference on Computational Complexity, Montreal, 2002. 162–171
19 Pass R, Rosen A. New and improved constructions of non-malleable cryptographic protocols. In: 37th ACM Symposium on the Theory of Computing, Baltimore, 2005. 533–542
20 Naor M, Ostrovsky R, Venkatesan R, et al. Perfect zero-knowledge arguments for NP using any one-way permutation. J Cryptol, 1996, 11: 87–108
21 Haitner I, Reingold O. Statistically-hiding commitment from any one-way function. In: 39th ACM Symposium on the Theory of Computing, San Diego, 2007. 1–10
22 Haitner I, Hoch J J, Reingold O, et al. Finding collisions in interactive protocols—a tight lower bound on the round complexity of statistically-hiding commitments. In: 48th Annual IEEE Symposium on Foundations of Computer Science, Providence Rhode Island, 2007. 669–679
23 Barak B. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In: Proceedings of 43th Annual IEEE Symposium on Foundations of Computer Science, Vancouver, 2002. 345–355