Round-Optimal Zero-Knowledge Proofs of Knowledge for NP

Li HongDa[†1], Feng DengGuo[2], Li Bao[1], Xu HaiXia[1]

[1] State Key Lab of Information Security, Graduate University of Chinese Academy of Sciences, Beijing 100049, China;

[2] State Key Lab of Information Security, Institute of software of Chinese Academy of Sciences, Beijing 100080, China

It is well known that all the known black-box zero-knowledge proofs of knowledge for NP are non-constant-round. Whether there exit constant-round black-box zero-knowledge proofs of knowledge for all NP languages under certain standard assumptions is a open problem. This paper focuses on the problem and give a positive answer by presenting two constructions of constant-round (black-box) zero-knowledge proofs of knowledge for the HC (Hamiltonian Cycle) problem. By the recent result of Katz, our second construction which relies on the existence of claw-free functions has optimal round complexity (5-round) assuming the polynomial hierarchy does not collapse.

zero-knowledge proofs, proofs of knowledge, black-box simulation, constant-round.

1 Introduction

Zero-knowledge proofs (ZKP), first introduced by Goldwasser, Micali, and Rackoff [16], are protocols that allow the prover to convince the verifier that an assertion is true without providing the verifier with any additional information about the assertion being proved. What does it mean to say an interactive proof system is zero-knowledge? Loosely speaking, the zero-knowledge property requires that whatever the verifier might have learned from interacting with the prover, the verifier could actually have obtained itself. Now, the concept of ZKP has become one of the fundamental notions in cryptography and is widely used in the design and realization of many cryptography tasks. This is due in large part to the result that any language in NP has a zero-knowledge proof system [17]. ZKP is required to protect the honest verifier from an all powerful prover trying to convince it of the validity of a false assertion. zero-knowledge arguments (ZKA) are a relaxation of the Zero-knowledge proofs, in which the soundness property is required to hold only with respect to a computationally bounded prover.

Proofs of knowledge, first defined by Goldwasser, Micali, and Rackoff [16], are proofs that allow the prover to convince the verifier that it knows a secret witness w about a given common input x. There have been several attempts to present an adequate formalization for this [8,21,20,9]. A simple scheme to achieve this without any security requirement would be to reveal the secret witness in question. When one requires the proof is zero-knowledge, the notion of proofs of knowledge, known as zero-knowledge proofs of knowledge, becomes very useful and complex. Analogous to regular interactive proofs, proofs of knowledge protocols should thus satisfy certain constraints: completeness (if the prover knows w then the verifier should accept) and soundness (for any prover that does not know w, the verifier should almost always reject). In addition, it should be zero-knowledge: no polynomial-time verifier (no matter what possibly dishonest strategy is followed during the proof) can learn any information about w. If a zero-knowledge proof (resp. argument) system for L is also a proof of knowledge system, it is known as a zero-knowledge proof (resp. argument) of knowledge for L. Now, zero-knowledge proofs or arguments of knowledge have since played a crucial role in the design of cryptographic schemes and protocols.

It is known that there exist constant-round zero-knowledge arguments of knowledge for NP problems [10,13,6]. However, to our knowledge, all the known zero-knowledge proofs of knowledge for NP problem

[†]Corresponding author (email: hdli@gucas.ac.cn)

This work was partially supported by the National Basic Research Program (973 program) of China (Grant No. 2007CB311202,2007CB311201) and the National High-Tech Research and Development Plan of China (Grant No. 2006AA01Z427)

are non-constant-round. Therefore, It is left as an open question whether or not there exit constant-round zero-knowledge proofs of knowledge for NP problems under reasonable assumptions.

1.1 Related Works

There are some impossibility results about constant-round zero-knowledge proofs. Goldreich and Oren [18] first proved that two-round auxiliary-input zero-knowledge proof systems do not exist for languages outside of BPP. Analogously, Barak et al. recently proved that 2-round zero-knowledge proof system with perfect completeness do not exist for any NP-complete language[5]. Goldreich and Krawczyk [15] proved that 3-round black-box zero-knowledge proofs do not exist for the language outside of BPP. In [24], the authors extends impossibility results from [15] to zero knowledge proof of knowledge, and prove that the existence of 3-round black-box zero knowledge proofs of knowledge for L implies there exists a probabilistic polynomial time algorithm which, on input $x \in L$, can output a witness for $x \in L$ with overwhelming probability. Recently, Katz proved that NP-complete languages do not have 4-round black-box zero-knowledge proofs assuming the polynomial hierarchy does not collapse[25].

On the other hand, Goldreich and Kahan [19] first presented a 5-round black-box zero-knowledge proof system for Graph 3-Colorability under the existence of claw-free functions. The recent result from [25] indicates that the round complexity of the construction in [19] for black-box simulation is optimal. Subsequently, Rosen [29] constructed an even simpler 7-round black-box zero-knowledge proof system for HC assuming the existence of two-round perfectly-hiding commitment schemes. Unfortunately, the constructions in [19,29] are not proofs of knowledge. Barak et al [4,5] recently proved there do not exist constant-round zero-knowledge strong proof or argument of knowledge for a non-trivial language.

1.2 Our main results

Note that all the known constructions of constant-round black-box zero-knowledge proof are not proofs of knowledge. This paper focuses on the existence of constant-round (black-box) zero-knowledge proofs of knowledge for NP under general cryptographic assumptions. The main contribution of this paper is to show the existence of constant-round (black-box) zero-knowledge proofs of knowledge for HC under standard complexity assumptions.

Main Theorem *Every NP problem has a constant-round zero-knowledge proof of knowledge system, provided that 1-1 one way functions and two-round perfectly hiding commitments exist. Specifically, every NP problem has 5-round zero-knowledge proof of knowledge systems, assuming that claw-free trapdoor permutations exist.*

The known approaches to constructing a constant-round zero-knowledge proof for NP force the verifier to commit its challenge in advance of the prover sending its commitments to the proved statement. This will indeed ensure the protocol is zero-knowledge (that is, there exists a black-box simulator which will work efficiently), but also results in the fact that the knowledge extraction strategy by rewinding the prover no longer works efficiently. In fact, it is seemingly the fact that the verifier is forced to commit its challenge in advance that destroys the proof of knowledge property. Therefore, to obtain a constant-round zero-knowledge proof of knowledge protocol for NP, we need a new approach that enables the black-box simulator (asking for the verifier to commit its challenge in advance) and knowledge extractor (asking for the prover to send its commitments first) to work efficiently.

Our approach to solving the problem is to let the challenge the that prover must answer be jointly determined by both the prover and verifier, and then no one can learns or controls it in advance. The advantage that this approach holds is that the challenge is independent of the prover's first commitment, and need not have been committed in advance. Thus, on the one hand, the classical knowledge extraction strategy can work efficiently. On the other hand, instead of modifying the commitment to fit the challenges as usual, the simulator can try to modify the random challenges by itself in order to give a simulated proof. This is possible because the random challenge are jointly determined by both the prover and verifier. In other words, this interactive proof can admit both a (black-box) simulator and a knowledge extractor.

1.3 Organization

In Section 2, the standard definitions and cryptographic tools used in our protocols are presented. We give a 7-round zero-knowledge proof of knowledge for HC in Section 3. In Section 4, we construct an optimal-round (5-round) zero-knowledge proof of knowledge for HC under

the existence of claw-free trapdoor permutations.

2 Preliminaries

In this paper, we use some standard notations. If $A(\cdot)$ is a probabilistic algorithm, $A(x)$ is the result of running A on input x and $y = A(x)$ denote that y is set to $A(x)$. For a finite set S, we denote by $y \in_R S$ that y is uniformly selected from S. We write U_n to denote a uniform distribution over $\{0,1\}^n$ and $poly(\cdot)$ to denote an unspecified polynomial. As usual, R_L is the corresponding relation of language $L \in NP$.

2.1 Zero-knowledge proof

We recall the definitions of zero-knowledge. These formal definitions are taken from [16,20].

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 when 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 is called the verifier. We denote by $\langle P, V\rangle(x) = 1$ ($\langle P, V\rangle(x) = 0$) that machine V accepts (rejects) the proofs given by machine P.

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

- Completeness: there exists a negligible function c such that for every $x \in L$,
$$\mathbf{Pr}[\langle P, V\rangle(x) = 1] > 1 - c(|x|)$$
- Soundness: there exists a negligible function s such that for every $x \notin L$ and every interactive machine B,
$$\mathbf{Pr}[\langle B, V\rangle(x) = 1] < s(|x|)$$

$c(\cdot)$ is called the completeness error, and $s(\cdot)$ the soundness error. In the case that the soundness condition is required to hold only with respect to a probabilistic polynomial-time prover, $\langle P, V\rangle$ is called an interactive arguments system for L.

An interactive proof is said to be zero-knowledge if the interaction between the prover and verifier 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^* there exists a polynomial-time algorithm $\mathcal{S}_{V^*}$ (a.k.a the simulator) such that the view of V^* can be simulated by $\mathcal{S}_{V^*}$. The idea behind this definition is that whatever V^* might have learned from interacting with P, could actually have been obtained by itself. We denote by $\mathrm{View}^P_{V^*}(x)$ a random variable describing the content of the random tape of V^* and the messages V^* receives during the interaction with P on common input x.

Definition 2 Let $\langle P, V\rangle$ be an interactive proof system for a language L. $\langle P, V\rangle$ is called a zero-knowledge proof system if for every probabilistic polynomial-time machine V^* there exists a probabilistic polynomial-time algorithm $\mathcal{S}_{V^*}$ such that $\{\mathrm{View}^P_{V^*}(x)\}_{x\in L}$ and $\{\mathcal{S}_{V^*}(x)\}_{x\in L}$ are computationally indistinguishable.

Black-box zero-knowledge requires that there exists a "universal" simulator $\mathcal{S}$ such that for every $x \in L$ and every probabilistic polynomial-time verifier V^*, the simulator $\mathcal{S}$ can simulate $View^P_{V^*}(x)$ while using V^* in a "black-box" manner.

Definition 3 (Black-Box Zero-Knowledge Proof) Let $\langle P, V\rangle$ be an interactive proof system for a language L. $\langle P, V\rangle$ is called a black-box zero-knowledge proof if there exists a probabilistic polynomial-time algorithm $\mathcal{S}$ such that for every probabilistic polynomial-time machine V^* $\{View^P_{V^*}(x)\}_{x\in L}$ and $\{\mathcal{S}^{V^*}(x)\}_{x\in L}$ are computationally indistinguishable.

2.2 Proof of knowledge

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

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

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:

- (efficiency): $\langle P, V \rangle$ is polynomially bounded, and V is computable in probabilistic polynomial time.

- (nontriviality): 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.

- (validity with knowledge error κ): Denote by $p(x, y, r)$ the probability that the interactive machine V accepts, on input x, 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). If there exists an expected polynomial-time oracle machine $\mathcal{K}$ and a polynomial q such that on input x and access to oracle $P_{x,y,r}$, $\mathcal{K}^{P_{x,y,r}}(x)$ outputs w, such that $(x, w) \in R$, with probability of at least $(p(x, y, r) - \kappa(|x|))/q(|x|)$

2.3 Claw-free trapdoor permutations

In this section we define the notion of claw-free trapdoor permutations. The reader is referred to [19,30] for an extended background related to these definitions.

Definition 4 A collection of pairs of functions $\mathcal{F} = \{f^0_e, f^1_e : D_e \to D_e\}_{e \in I}$ over some index set $I \subseteq \{0,1\}^*$ is called a family of claw-free trapdoor permutations if the following hold:

- There exists a probabilistic polynomial-time generating algorithm, $Gen(\cdot)$, that on input 1^n outputs a random index $e \in I \cap \{0,1\}^n$ and trapdoor information t_e

- There are efficient sampling algorithms D which, on input e, output a random $x \in D_e$.

- Two functions f^0_e, f^1_e specified by e are efficiently computable given index e and input $x \in D_e$.

- f^0_e and f^1_e are permutations over D_e and easy to invert given index e and its trapdoor t_e. That is, given t_e, $(f^0_e)^{-1}(y), (f^1_e)^{-1}(y)$ are efficiently computable for any $y \in D_e$.

- It is hard to find a claw for index e. Formally, there exists a negligible function $\mu(\cdot)$, such that for every probabilistic polynomial-time algorithm $\mathcal{A}$ and sufficiently large n, we have
$\mathbf{Pr}[(e \leftarrow Gen(1^n), (x_0, x_1) \leftarrow \mathcal{A}(e) : f^0_e(x_0) = f^1_e(x_1)] < \mu(n)$

We remark that claw-free trapdoor permutations can be constructed from certain general assumptions, and more examples are given in [20,30].

2.4 Commitment Schemes

Commitment schemes are used to enable a party, known as the sender, to commit itself to a value while keeping it secret from another party, called 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. We sketch the two properties. General definitions can be found in [20].

- **Perfectly binding commitments:** the binding property holds against unbounded senders, while the hiding property only holds against computationally bounded receivers.

- **Perfectly hiding commitments:** the hiding property holds against unbounded receivers, while the binding property only holds against computationally bounded senders.

In this paper, we use two types of commitment schemes. One is a non-interactive perfectly-binding commitment scheme which can be constructed using any 1-1 one-way function (see Section 4.4.1 of [20]). The other is a two-round perfectly-hiding commitment scheme. In particular, this scheme can be constructed based on claw-free collections [19].

3 Constant-round zero-knowledge proofs of knowledge for HC

Suppose that the language HC consists of all directed graphs that contain a Hamiltonian cycle. Our goal in this section is to construct a constant-round zero-knowledge proof of knowledge for HC.

Recall the classical form of 3-round interactive proofs by reviewing the basic protocol for HC presented by Blum in [1]. Here and throughout this paper, $Comm(\cdot;\cdot)$ is a non-interactive perfectly binding commitment scheme, that can be constructed using any 1-1

one-way function (see Section 4.4.1 of [20]). For convenience, we denote by $Comm(G;r)$ the commitment to the adjacency matrix of G.

- The prover first sends its commitment c: The prover selects randomly a permutation π over V and $r \in_R \{0,1\}^{poly(|V|)}$, computes $c = Comm(\pi(G);r)$, and sends c to the verifier.
- The verifier sends a challenge σ: The verifier uniformly selects $\sigma \in_R \{0,1\}$ and sends it to the prover.
- The prover responds to the challenge with a_σ: If $\sigma = 0$, the prover reveals the partial commitments (of c) corresponding to the edges of the Hamiltonian cycle $\pi(H)$. If $\sigma = 1$ then the prover reveals all the commitments of c and π.
- Upon receiving the response a_σ from the prover, the verifier checks whether $Check_\sigma(c, a_\sigma) = 1$. (Here and throughout this paper, we define $Check_\sigma(c, a_\sigma) = 1$ if that all the values revealed by a_σ are 1 and the corresponding edges form a simple Hamiltonian cycle when $\sigma = 0$, or a_σ correctly reveals all the commitments of c and π such that the revealed graph is indeed isomorphic G via π when $\sigma = 1$)

The above construction is a proof of knowledge with knowledge error $1/2$ for HC. Furthermore, it is also a weak zero-knowledge proof system because its soundbess error is $1/2$ but not negligible.

To reduce the knowledge error, a straightforward method can be used, that is, sequentially running the basic protocol many times. Clearly, sequentially repeating the basic protocol sufficient many times yields a zero-knowledge proof of knowledge for HC (with negligible knowledge error), but this is not a constant-round proof system. To obtain constant-round proofs of knowledge, one can run the basic protocol many times in parallel. Of course, the parallel protocol is also a proof of knowledge with negligible knowledge error too. In fact, the classical knowledge extraction strategy is given below.

- Obtain the first message from P^*.
- Pick a random challenge $\sigma = \sigma_1 \cdots \sigma_n \in \{0,1\}^n$, and feed P^* with this challenge. If P^* does not answer correctly, stop.
- Otherwise, repeatedly choose a random challenge $\sigma' \in \{0,1\}^n$, and then rewind P^* by feeding σ' until P^* answers correctly a second time.
- If $\sigma' \neq \sigma$, compute H from the answer and outputs it.

However, the protocol yielded from paralleling n basic protocols cannot be proved to be zero-knowledge.

By forcing the verifier to commit (using a perfectly-hiding commitment scheme) to its challenges $\sigma = \sigma_1 \cdots \sigma_n$ in advance, we can recover the zero-knowledge property of n parallel repetitions of the basic protocol, such as in [19,29]. However, the resulting protocol is no longer a proof of knowledge, because the fact that the verifier is forced to commit to σ in advance will result in the above classical strategy no longer being valid. In fact, to invoke P^* with a new different challenge, knowledge extractor $\mathcal{K}$ must rewind P^* back to the point where P^* receives the verifier's commitment, and consequentially, P^* might change its commitments.

The zero-knowledge property requires that there exists a simulator that rewinds a possible cheating verifier V^* to simulate a proof without knowing a witness, while proofs of knowledge require that there exists a knowledge extractor that accesses the prover's strategy P^* to extract a witness from P^*. We know that the approach forcing the verifier to commit to its challenges in advance ensures the zero-knowledge property, but then seemingly destroys the proof of knowledge property. To obtain a constant-round protocol that admits both a (black-box) simulator and a knowledge extractor, we will consider a new type of interactive proof, in which the challenge is jointly determined by both the prover and verifier, and neither has outright control over it. The advantage that this approach holds is that the challenge is independent of the prover's first commitment, and need not have been committed in advance. Thus, on the one hand, the classical knowledge extraction strategy can work efficiently. On the other hand, the simulator can simulate the proof by modifying the challenge all by itself. In other words, this interactive proof can admit both a (black-box) simulator and a knowledge extractor.

The details of our construction are given below. Let $Comm_{ph}(\cdot;\cdot)$ be a two-round perfectly hiding commitment scheme.

Construction 1

Common input: $G = (V, E) \in HC, |V| = n$.
Auxiliary input to the prover: A Hamilton cycle H in G such that $(G, H) \in R_{HC}$.

P1 The prover first proceeds as follows:

- Randomly select n permutations $\pi_1, \cdots, \pi_n$ over V and commit to the adjacency matrix of $\pi_j(G)$ for $j = 1, \cdots, n$. That is, compute $C_k = Comm(\pi_j(G); R_j)$ for $R_j \in_R \{0,1\}^{n^2 \cdot poly(n)}$, $j = 1, \cdots, n$.
- Send $(C_1, \cdots, C_n), m$ to the verifier, where m is the first message of the commitment scheme $Comm_{ph}(\cdot;\cdot)$.

V1 After receiving the first messages from the prover, the verifier proceeds as follows:

- Select $\sigma = \sigma_1 \cdots \sigma_n \in_R \{0,1\}^n$, $\theta_j^0, \theta_j^1 \in_R \{0,1\}^n$, such that $\sigma = \theta_j^0 \oplus \theta_j^1$, $j = 1, \cdots, n$.
- Commit to σ: Concretely, compute $y = Comm_{ph}(\sigma; r)$ for $r \in_R \{0,1\}^{poly(n)}$.
- Commit to $\theta_1^0, \theta_1^1, \cdots, \theta_n^0, \theta_n^1$: That is, compute $y_j^0 = Comm_{ph}(\theta_j^0; r_j^0)$, $y_j^1 = Comm_{ph}(\theta_j^1; r_j^1)$, where $r_j^0, r_j^1 \in_R \{0,1\}^{poly(n)}$, $j = 1, \cdots, n$.
- Send $y, (y_1^0, y_1^1), \cdots, (y_n^0, y_n^1)$ to the prover.

P2 The prover sends $\delta = \delta_1 \cdots \delta_n \in_R \{0,1\}^n$ to the verifier.

V2 The verifier reveals $(\theta_1^{\delta_1}, r_1^{\delta_1}), \cdots, (\theta_n^{\delta_n}, r_n^{\delta_n})$.

P3 The prover verifies the received revealment. If the verification fails, the prover aborts. Otherwise, the prover picks $\varepsilon = \varepsilon_1 \cdots \varepsilon_n \in_R \{0,1\}^n$ and sends ε to the verifier.

V3 The verifier reveals σ and $\theta_1^{1-\delta_1}, \cdots, \theta_n^{1-\delta_n}$.

P4 The prover first verifies the revealment. If the verification fails, the prover aborts. Otherwise, the prover answers the challenge $\alpha_j = \varepsilon_j \oplus \sigma_j$ by revealing the partial commitments or all the commitments (of C_j), $j = 1, \cdots, n$. That is, the prover sends $A_1, \cdots, A_n$ to the verifier, where A_j reveals the partial commitments (of C_j) corresponding to the edges of the Hamiltonian cycle $\pi_j(H)$ when $\alpha_j = 0$, or reveals all the commitments of $\pi(G)$ and π when $\alpha_j = 1$, $j = 1, \cdots, n$.

V4 After receiving $(A_1, \cdots, A_n)$, the verifier checks

$$\bigwedge_{j=1}^{n} \left(Check_{\alpha_j}(C_k, A_j) = 1 \right)$$

where $\alpha_j = \varepsilon_j \oplus \sigma_j$. The verifier accepts if and only if the verification succeeds.

Theorem 1 Construction 1 constitutes a zero-knowledge proof of knowledge (with negligible knowledge error) for HC, assuming $Comm(\cdot;\cdot)$ is a non-interactive perfectly binding commitment scheme and $Comm_{ph}(\cdot;\cdot)$ is a two-round perfectly hiding commitment scheme.

Proof: Completeness is obvious. Soundness follows directly from the perfectly hiding property of $Comm_{ph}(\cdot;\cdot)$. The (black-box) simulator $\mathcal{S}^{V^*}$ showing its zero-knowledge property operates as follows:

1. Uniformly select a string $r \in \{0,1\}^{poly(n)}$ to be used as the content of the local random tape of V^*.
2. From $j = 1$ to n,
   - Randomly chooses a simple Hamiltonian cycle H_j, a permutation π_j over V and $R_j \in_R \{0,1\}^{n^2 \cdot poly(n)}$.
   - Randomly pick $\alpha_j \in_R \{0,1\}$. If $\alpha_j = 0$, then compute $C_j = Comm(H_j; R_j)$. Otherwise, compute $C_j = Comm(\pi_j(G); R_j)$.
3. Feed V^* with $\overline{C} = (C_1, \cdots, C_n)$ and m (the first message of $Comm_{ph}(\cdot;\cdot)$), that respond to $y, (y_1^0, y_1^1), \cdots, (y_n^0, y_n^1)$.
4. Randomly selects $\delta = \delta_1 \cdots \delta_n \in \{0,1\}^n$, and feed δ to V^*, which answers by revealing $(\theta_1^{\delta_1}, r_1^{\delta_1}), \cdots, (\theta_n^{\delta_n}, r_n^{\delta_n})$.
5. If some of the decommitments are not correct, output $(r, \overline{C}, m, \delta)$ and stop. Otherwise, proceed to the next step.
6. Repeatedly choose $\delta' = \delta_1' \cdots \delta_n' \in \{0,1\}^n$, rewind the V^*, and feed V^* with δ' until V^* decommits $y_1^{\delta_1'}, \cdots, y_1^{\delta_1'}$ correctly.
7. If $\delta = \delta'$, output $\perp$ and stop.
8. Otherwise, there exists $1 \leqslant i \leqslant n$ such that $\delta_i \neq \delta_i'$. Let $\sigma = \sigma_1 \cdots \sigma_n = \theta_i^{\delta_i} \oplus \theta_i^{\delta_i'}$.

9. Feed V^* with $\varepsilon = \alpha \oplus \sigma$, and receive V^*'s response σ and $\theta_1^{1-\delta_1}, \cdots, \theta_n^{1-\delta_n}$.

10. Verify the revealed values. If this verification fails, output $(r, \overline{C}, m, \delta, \varepsilon)$ and stop.

11. Otherwise, let A_j be the revealment of the partial commitments (of C_j) corresponding to the edges of the Hamiltonian cycle H_j when $\alpha_j = 0$, or be the revealment of all the commitments of C_j and π_j when $\alpha_j = 1$, $j = 1, \cdots, n$. Then output $(r, \overline{C}, m, \delta, \varepsilon, (A_1, \cdots, A_n))$ and stop.

It is easy to see that the expected running time of $\mathcal{S}^{V^*}(G)$ is strictly polynomial-time. Next, we move on to show that $\{\mathcal{S}^{V^*}(G)\}_{G \in HC}$ is computationally indistinguishable from $\{\mathrm{View}_{V^*}^P(G)\}_{G \in HC}$.

Note that the two differences between $\mathcal{S}^{V^*}(G)$ and $\mathrm{View}_{V^*}^P(G)$ are: (1) $\mathcal{S}^{V^*}(G) = \perp$ when $\delta' = \delta$, and (2) some of the committed values of $C_1, \cdots, C_n$ are different. Note that $Pr[\mathcal{S}^{V^*}(G) = \perp]$ is negligible [1]. Thus, it follows directly from the hiding property of $Comm(\cdot;\cdot)$ that $\{\mathcal{S}^{V^*}(G)\}_{G \in HC}$ is computationally indistinguishable from $\{\mathrm{View}_{V^*}^P(G)\}_{G \in HC}$.

Next, we show the validity by defining the knowledge extractor $\mathcal{K}$, which has access to the prover-strategy oracle P^*. On input x, $\mathcal{K}$ proceeds as follows:

1. On input G, invoke P^* and obtains its responses: $\overline{C} = (C_1, \cdots, C_n)$ and m.

2. Select $\sigma, \theta_1^0, \theta_1^1, \cdots, \theta_n^0, \theta_n^1 \in_R \{0,1\}^n$, compute commitments $y, (y_1^0, y_1^1), \cdots, (y_n^0, y_n^1)$ as **V1**. Feed P^* with $(y, (y_1^0, y_1^1), \cdots, (y_n^0, y_n^1))$, and then obtain the response $\delta = \delta_1 \cdots \delta_n$.

3. Query the oracle P^* with $(\theta_1^{\delta_1}, r_1^{\delta_1}) \cdots, (\theta_n^{\delta_n}, r_n^{\delta_n})$, (the decommitment of $(y_1^{\delta_1}, \cdots, y_n^{\delta_n})$), and then obtains ε.

4. Query P^* with (σ, r) (the decommitment of y) and $(\theta_1^{1-\delta_1}, r_1^{1-\delta_1}) \cdots, (\theta_n^{1-\delta_n}, r_n^{1-\delta_n})$ (the commitments of $(y_1^{1-\delta_1}, \cdots, y_n^{1-\delta_n})$), and then obtain its responses $(A_1, \cdots, A_n)$.

5. After receiving $(A_1, \cdots, A_n)$, verify
$$\bigwedge_{j=1}^{n} Check_{\alpha_j}(C_k, A_j) = 1$$
where $\alpha = \alpha_1 \cdots \alpha_n = \varepsilon \oplus \sigma$. If the verification fails, stop without any output.

6. Otherwise, repeat the following.

   - Select $\sigma', \theta_1^0, \theta_1^1, \cdots, \theta_n^0, \theta_n^1 \in_R \{0,1\}^n$, such that $\theta_1^0 \oplus \theta_1^1 = \cdots = \theta_n^0 \oplus \theta_n^1 = \sigma'$, compute the commitments $y, (y_1^0, y_1^1), \cdots, (y_n^0, y_n^1)$ as **V1**. Feed P^* with $y, (y_1^0, y_1^1), \cdots, (y_n^0, y_n^1)$, then get the response $\delta = \delta_1 \cdots \delta_n$.
   - Invoke P^* with $(\theta_1^{\delta_1}, r_1^{\delta_1}), \cdots, (\theta_n^{\delta_n}, r_n^{\delta_n})$, and then obtains ε'.
   - Query P^* with (σ', r) and $(\theta_1^{1-\delta_1}, r_1^{1-\delta_1})$, $\cdots$, $(\theta_n^{1-\delta_n}, r_n^{1-\delta_n})$, and then obtain its responses $(A'_1, \cdots, A'_n)$.
   - After receiving $(A'_1, \cdots, A'_n)$, verify
   $$\bigwedge_{j=1}^{n} Check_{\alpha'_j}(C_k, A'_j) = 1$$
   where $\alpha' = \alpha'_1 \cdots \alpha'_n = \varepsilon' \oplus \sigma'$, If the verification succeeds, terminate the repeat and proceed to the next step.

7. If $\alpha = \alpha'$, stop without any output. Otherwise, there exists at least k, such that $\alpha_k \neq \alpha'_k$. Compute a cycle H from A_k and A'_k. Output H and stop.

Let p be the probability that P^* convinces the verifier of the proved statement. Note that $\mathcal{K}$ always stops at step 5 and then runs in strict polynomial time when $p = 0$. So consider $p > 0$ in what follows. Let E denote the event that the knowledge extractor $\mathcal{K}$ terminates the repeat successfully, $p' = Pr[E]$. Note that $p' \geqslant p$ (in fact, $p' = p$), and then the expected number of times of running the repeat is $\frac{1}{p'}$. Therefore, the expected running time of $\mathcal{K}$ given by
$(1-p) \cdot poly(n) + p \cdot (\frac{1}{p'} \cdot poly(n) + poly(n)) = poly(n)$
is polynomial-time.

Let E_α denote the event that $\mathcal{K}$ obtains $\alpha = \alpha_1 \cdots \alpha_n$ and $(A_1, \cdots, A_n)$ such that
$$\bigwedge_{j=1}^{n} Check_{\alpha_j}(C_k, A_j) = 1$$

[1] In fact, let E denote the event that V^* answers the $\mathcal{S}^{V^*}$'s query δ correctly, and note that $q = Pr[\delta' = \delta | E] Pr[E]$. Therefore, $q \leqslant 2^{-n}$ when $p = Pr[E] \leqslant 2^{-n}$, whereas $q = p \sum_{k=0} (1 - \frac{m-1}{2^n})^k \cdot 2^{-n} = 2^{-n} p \cdot \frac{2^n}{m-1} = 2^{-n} \cdot \frac{m}{m-1}$ when $p = Pr[E] = \frac{m}{2^n}$ ($m \geqslant 2$). Overall, $q \leqslant 2^{-n+1}$.

and T denote the event that $\mathcal{K}$ terminates the repeats. Obviously, $p = \sum_\alpha Pr[E_\alpha]$. Assume $p = \frac{m}{2^n} > 2^{-n}$, we have

$$\begin{aligned} & Pr[(G,H) \in R_{HC} : H \leftarrow \mathcal{K}(G)] \\ = & \sum_\alpha Pr[E_\alpha \wedge T \wedge (\alpha \neq \alpha')] \\ = & \sum_\alpha Pr[E_\alpha] \cdot Pr[T \wedge (\alpha \neq \alpha') | E_\alpha] \end{aligned}$$

where α' is determined when T occurs. Since

$$\begin{aligned} & Pr[T \wedge (\alpha \neq \alpha') | E_\alpha] \\ = & (\tfrac{m-1}{2^n} + \cdots + (1 - \tfrac{m}{2^n})^k \cdot \tfrac{m-1}{2^n} + \cdots) \\ = & \tfrac{m-1}{m} \end{aligned}$$

we obtain

$$\begin{aligned} & Pr[(G,H) \in R_{HC} : H \leftarrow \mathcal{K}(G)] \\ = & p\tfrac{m-1}{m} = p - 2^{-n} \end{aligned}$$

That is, $\mathcal{K}$ succeeds in computing a Hamiltonian cycle in G with probability of at least $p - 2^{-n}$ when $p > 2^{-n}$. ■

4 Round-optimal zero-knowledge proofs of knowledge for HC

In [25], Katz shows that 4-round black-box zero-knowledge proofs do not exist for any NP-complete language assuming the polynomial hierarchy does not collapse. In this section, our aim is to construct a round-optimal zero-knowledge proof of knowledge for HC. In other word, we will present a 5-round zero-knowledge proof of knowledge for the NP-complete language HC by modifying the above construction.

The details of our construction are given below. Note that the pseudorandom generator $G^{l(n)}(\cdot)$ used in the construction is derived from one trapdoor permutation of the claw-free permutations generated by the prover, where $l(n)$ is any polynomial that specifies the output length. Note the verifier can obtain the claw-free permutation from the prover, so the verifier can hold $G^{l(n)}(\cdot)$.

Construction 2

Common input: $G = (V, E) \in HC, |V| = n$.
Auxiliary input to the prover: A Hamilton cycle H in G such that $(G, H) \in R_{HC}$.

P1 The prover first proceeds as follows:

- Randomly selects n permutations $\pi_1, \cdots, \pi_n$ over V and n^3 strings $R_1 = (r^{(1)}_{1,1}, \cdots, r^{(1)}_{n,n}), \cdots, R_n = (r^{(n)}_{1,1}, \cdots, r^{(n)}_{n,n})$, where $r^{(k)}_{i,j} \in_R \{0,1\}^{poly(n)}$, and $poly(\cdot)$ is determined by the commitment scheme. Then computes

$$C_k = Comm(\pi_k(G); R_k), k = 1, \cdots, n$$

- Set $A^0_k = (\pi_k(H), R_{\pi_k(H)}), A^1_k = (\pi_k, R_k)$, $k = 1, \cdots, n$, where $R_{\pi_k(H)} = \{r^{(k)}_{i,j} : r^{(k)}_{i,j} \in R_k, h^{(k)}_{i,j} = 1\}$, and $(h^{(k)}_{j,k})_{n \times n}$ is the adjacency matrix of $\pi_k(H)$. Assume $p_0(n) = |A^0_1| = \cdots = |A^0_n|$, $p_1(n) = |A^1_1| = \cdots = |A^1_n|$.
- $e \leftarrow Gen(1^n)$.
- Send $\overline{C} = (C_1, \cdots, C_n), e$ to the verifier.

V1 After receiving the first messages from the prover, the verifier proceeds as follows:

- Select $\sigma = \sigma_1 \cdots \sigma_n \in_R \{0,1\}^n$, $\theta^0_j = \theta^0_{j,1} \cdots \theta^0_{j,n}, \theta^1_j = \theta^1_{j,1} \cdots \theta^1_{j,n} \in_R \{0,1\}^n$, such that $\sigma = \theta^0_j \oplus \theta^1_j, j = 1, \cdots, n$.
- Select $x_1, \cdots, x_n \in_R D_e$, and compute $y_i = f^{\sigma_i}_e(x_i), i = 1, \cdots, n$. Let $y = (y_1, \cdots, y_n)$.
- Pick $x^0_{j,k}, x^1_{j,k} \in_R D_e$ and compute $y^0_{j,k} = f^{\theta^0_{j,k}}_e(x^0_{j,k}), y^1_{j,k} = f^{\theta^1_{j,k}}_e(x^1_{j,k})$, for $j, k = 1, \cdots, n$. Let $y^0_j = (y^0_{j,1}, \cdots, y^0_{j,n}), y^1_j = (y^1_{j,1}, \cdots, y^1_{j,n}), j = 1, \cdots, n$.
- Send $y, (y^0_1, y^1_1), \cdots, (y^0_n, y^1_n)$ to the prover.

P2 The prover sends $\delta = \delta_1 \cdots \delta_n \in_R \{0,1\}^n$ to the verifier.

V2 The verifier reveals $\theta^{\delta_1}_1, \cdots, \theta^{\delta_n}_n$. That is, the verifier sends $(\theta^{\delta_1}_1, \cdots, \theta^{\delta_n}_n)$, and $(x^{\delta_1}_{1,1}, \cdots, x^{\delta_1}_{1,n}), \cdots, (x^{\delta_n}_{n,1}, \cdots, x^{\delta_n}_{n,n})$ to the prover.

P3 The prover verifies the revealment. If the verification fails, the prover aborts. Otherwise, the prover acts as follows:

- Randomly pick $\varepsilon = \varepsilon_1 \cdots \varepsilon_n \in_R \{0,1\}^n$.
- Compute $z^0_j = (f^{\varepsilon_j}_e)^{-1}(y_j)$, $z^1_j = (f^{1 \oplus \varepsilon_j}_e)^{-1}(y_j), j = 1, \cdots, n$.
- Randomly pick $w^0_1, w^1_1, \cdots, w^0_n, w^1_n \in_R D_e$ and compute $d^0_1 = z^0_1 \oplus w^0_1, d^1_1 = z^1_1 \oplus w^1_1$, $\cdots$, $d^0_n = z^0_n \oplus w^0_n, d^1_n = z^1_n \oplus w^1_n$

– Compute $s_1^0 = A_1^0 \oplus G^{p_0(n)}(w_1^0), s_1^1 = A_1^1 \oplus G^{p_0(n)}(w_1^1), \cdots, s_n^0 = A_n^0 \oplus G^{p_0(n)}(w_n^0), s_n^1 = A_n^1 \oplus G^{p_0(n)}(w_n^1)$.

– Set $T_j^0 = (d_j^0, s_j^0)$, $T_j^1 = (d_j^1, s_j^1)$, $T_j = (T_j^0, T_j^1)$, $j = 1, \cdots, n$. Send $\overline{T} = (T_1, \cdots, T_n)$ and ε to the verifier.

V3 After receiving $\overline{T} = (T_1, \cdots, T_n)$, the verifier sets $\alpha = \alpha_1 \cdots \alpha_n = \sigma \oplus \varepsilon$ and computes: $B_k^{\alpha_k} = s_k^{\alpha_k} \oplus G^{p_{\alpha_k}(n)}(x_k \oplus d_k^{\alpha_k}), k = 1, \cdots, n$. The verifier accepts if and only if the following condition holds

$$\bigwedge_{k=1}^{n} (Check_{\alpha_k}(C_k, B_k^{\alpha_k}) = 1)$$

Theorem 2 Construction 2 constitutes a zero-knowledge proof of knowledge (with negligible knowledge error) for HC, assuming $Comm(\cdot;\cdot)$ is a non-interactive perfectly binding commitment scheme and $\mathcal{F} = \{f_e^0, f_e^1\}_{e \in I}$ is a claw-free collection.

Proof: Completeness and Soundness are obvious.

In the following, we first prove the zero-knowledge property. To achieve this, we define a (black-box) simulator $\mathcal{S}^{V^*}$ for any V^* on input G as follows:

1. Uniformly select a string $r \in \{0,1\}^{poly(n)}$ to be used as the content of the local random tape of V^*.
2. From $j = 1$ to n,
   - Randomly choose a simple Hamilton cycle H_j, a permutations π_j over V and $R_j \in_R \{0,1\}^{n^2 \cdot poly(n)}$.
   - Randomly pick $\alpha_j \in_R \{0,1\}$. If $\alpha_j = 0$, then compute $C_j = Comm(H_j; R_j)$, set $A_j^0 = (H_j, R_{H_j})$ (here, R_{H_j} is analogous to $R_{\pi_k(H)}$ in **P1**) and randomly pick A_j^1. Otherwise, compute $C_j = Comm(\pi_i(G); R_i)$, set $A_j^1 = (\pi_j, R_j)$ and randomly picks A_j^0.
3. $e \leftarrow Gen(1^n)$.
4. Feed V^* with $\overline{C} = (C_1, \cdots, C_n)$ and e, which responds with $y, (y_1^0, y_1^1), \cdots, (y_n^0, y_n^1)$.
5. Randomly selects $\delta = \delta_1 \cdots \delta_n \in \{0,1\}^n$, and feed V^* with δ, which responds by decommitting $y_1^{\delta_1}, \cdots, y_1^{\delta_1}$. That is, V^* output $(\theta_1^{\delta_1}, \cdots, \theta_n^{\delta_n})$, and $(x_{1,1}^{\delta_1}, \cdots, x_{1,n}^{\delta_1}), \cdots, (x_{n,1}^{\delta_n}, \cdots, x_{n,n}^{\delta_n})$.
6. If some of the decommitments of $y_1^{\delta_1}, \cdots, y_1^{\delta_1}$ are not correct, outputs $(\overline{C}, e, \delta)$ and stop. Otherwise, proceed to the next step.
7. Repeatedly choose $\delta' = \delta_1' \cdots \delta_n' \in \{0,1\}^n$, rewind the V^*, and feed V^* with δ' until V^* decommits $y_1^{\delta_1'}, \cdots, y_1^{\delta_1'}$ correctly.
8. If $\delta = \delta'$, output $\perp$ and stop.
9. Otherwise, there exists $1 \leqslant i \leqslant n$ so that $\delta_i \neq \delta_i'$. Let $\sigma = \sigma_1 \cdots \sigma_n = \theta_i^{\delta_i} \oplus \theta_i^{\delta_i'}$
10. Proceed as follows:
    - Set $\varepsilon = \alpha \oplus \sigma$, where $\alpha = \alpha_1 \cdots \alpha_n$ is selected in step 2.
    - Compute, for $j = 1, \cdots, n$, $z_j^0 = (f_e^{\varepsilon_j})^{-1}(y_j), z_j^1 = (f_e^{1 \oplus \varepsilon_j})^{-1}(y_j)$
    - Randomly pick $w_1^0, w_1^1, \cdots, w_n^0, w_n^1 \in_R D_e$ and compute $d_1^0 = z_1^0 \oplus w_1^0, d_1^1 = z_1^1 \oplus w_1^1, \cdots, d_n^0 = z_n^0 \oplus w_n^0, d_n^1 = z_n^1 \oplus w_n^1$
    - Compute, for $j = 1, \cdots, n$, $s_j^0 = A_j^0 \oplus G^{p_0(n)}(w_j^0)$ $s_j^1 = A_j^1 \oplus G^{p_1(n)}(w_j^1)$
    - Finally, set $T_j^0 = (d_j^0, s_j^0)$, $T_j^1 = (d_j^1, s_j^1)$, $T_j = (T_j^0, T_j^1)$, $j = 1, \cdots, n$.
11. Output $(r, (C_1, \cdots, C_n), e, \delta, (T_1, \cdots, T_n), \varepsilon)$ and stop.

Let p be the probability that V^* correctly reveals $(\theta_1^{\delta_1}, \cdots, \theta_n^{\delta_n})$. It is easy to see that $\mathcal{S}^{V^*}(G)$ runs in strict polynomial time when $p = 0$. Consider the case where $p > 0$. The expected running time of the simulator is bounded by

$$poly(n) + p(\frac{1}{p} poly(n) + poly(n)) \leqslant poly(n))$$

Overall, $\mathcal{S}$ runs in the expected polynomial time.

Let $\mathcal{S}^{V^*}(G)$ be the distribution induced by $\mathcal{S}^{V^*}$ on input G. We now prove that $\{\mathcal{S}^{V^*}(G)\}_{G \in HC}$ is computationally indistinguishable from $\{\text{View}_{V^*}^P(G)\}_{G \in HC}$.

Note that $q = Pr[\mathcal{S}^{V^*}(G) = \perp] = Pr[\delta' = \delta]$ is negligible [2]. So consider the case where $\mathcal{S}^{V^*}(G)$ never outputs $\perp$ in what follows. That is, we only need to show that $\{\mathcal{S}(G)\}_{G \in HC}$ conditional on it not being $\perp$ is computationally indistinguishable from $\{\text{View}_{V^*}^P(G)\}_{G \in HC}$.

[2] Analogous to [1].

In addition, because the probability that the simulator outputs $(\overline{C}, e, \delta)$ is equal to the probability that the prover aborts in **P**3, we only need to show that $\{\text{View}^P_{V^*}(G)\}_{G\in HC}$ is computationally indistinguishable from $\{\mathcal{S}(G)\}_{G\in HC}$ conditional on it being in form of $(r, (C_1, \cdots, C_n), e, \delta, (T_1, \cdots, T_n), \varepsilon)$.

We define a hybrid simulator $\mathcal{S}'$, which follows the same strategy as the simulator $\mathcal{S}$ on input G, except that, instead of performing step 2, it is given the Hamiltonian cycle H in G and then does as the following:

2' From $j = 1$ to n,

- Randomly choose a permutation π_j, $R_j \in \{0,1\}^{poly(n)}$ and $\rho_j \in_R \{0,1\}$.
- Compute $C'_j = Comm(\pi_i(G); R_i)$. Set $A^0_j = (\pi_j(H), R_{H_j})$ and randomly pick A^1_j when $\rho_j = 0$, set $A^1_j = (\pi_j, R_j)$ and randomly pick A^0_j when $\rho_j = 1$.

Let $\mathcal{S}'(G, H)$ be the distribution output by $\mathcal{S}'$. It then follows directly from the hiding property of the commitment scheme $Comm(\cdot;\cdot)$ that $\{\mathcal{S}'(G, H)\}_{G\in HC}$ is computationally indistinguishable from $\{\mathcal{S}(G)\}_{G\in HC}$. Thus, all we need to show is that $\{\mathcal{S}'(G, H)\}_{G\in HC}$ is computationally indistinguishable from $\{\text{View}^P_{V^*}(G)\}_{G\in HC}$. Note that, by the definition,
$View^P_{V^*}(G) = (r, (C_1, \cdots, C_n), e, \delta, (T_1, \cdots, T_n), \varepsilon)$
We rewrite $T_j = (T_j^{\alpha_j}, T_j^{1\oplus\alpha_j})$, where $\alpha_j = \sigma_j \oplus \varepsilon_j, j = 1, \cdots, n$. For convenience, we write
$\mathcal{S}'(G, H) = (r', (C'_1, \cdots, C'_n), e', \delta', (T'_1, \cdots, T'_n), \varepsilon')$
where $T'_j = (T_j^{\alpha_j}, T_j^{1\oplus\alpha_j})$, $j = 1, \cdots, n$.

For every $k \in \{0, 1, \cdots, n\}$, define a new hybrid simulation procedure $\mathcal{S}'_k$: $\mathcal{S}'_k$ follows the same strategy as the simulator $\mathcal{S}'$ on input G, except for setting $A^0_1, A^1_1, \cdots, A^0_k, A^1_k$ as the prover does in step P1.

We define hybrid random variables $\mathcal{S}'_k(G, H)$ which are induced by the output of the hybrid simulation procedure $\mathcal{S}_k$. Clearly, $\mathcal{S}_0(G, H) = \mathcal{S}'(G, H)$, whereas $\mathcal{S}_n(G, H) = \text{View}^P_{V^*}(G, H)$.

Suppose that there is a PPT algorithm $\mathcal{D}$ that can distinguish $\{\mathcal{S}'(G, H)\}_{G\in HC}$ from $\{\text{View}^P_{V^*}(G, H)\}_{G\in HC}$. It follows that there exist $0 \leqslant k < n$ and an algorithm $\mathcal{D}'$, such that $\mathcal{D}'$ can distinguish a pair of neighboring hybrids $\mathcal{S}'_k(G, H)$ and $\mathcal{S}'_{k+1}(G, H)$. Note that the only difference between $\mathcal{S}'_k(G, H)$ and $\mathcal{S}'_{k+1}(G, H)$ is that T'_{k+1} is different from T_{k+1}. Therefore, using $\mathcal{D}'$ we can derive a non-uniform algorithm to distinguish T_{k+1} from T'_{k+1}. Note that the only difference between T_{k+1} and T'_{k+1} is $T^{1\oplus\alpha_{k+1}}_{k+1}$: $T^{1\oplus\alpha_{k+1}}_{k+1} = (A^{\varepsilon_{k+1}}_{k+1}, x^{\varepsilon_{k+1}}_{k+1}) \oplus (G(w^{\varepsilon_{k+1}}_{k+1}), w^{\varepsilon_{k+1}}_{k+1})$ in T_{k+1}, whereas $T^{1\oplus\alpha_{k+1}}_{k+1} = (U_{1\oplus\alpha_{k+1}}, U_e)$ in T'_{k+1} because $A^{1\oplus\alpha_{k+1}}_{k+1}$ and $w^{\varepsilon_{k+1}}_{k+1}$ are all uniformly distributed in T'_{k+1}. Thus, the fact that $\mathcal{D}'$ can distinguish T_{k+1} from T'_{k+1} will contradict the assumptions that $G^{l(n)}(\cdot)$ is pseudorandom and $\mathcal{F}$ is a claw-free collection. Overall, $\{\mathcal{S}(G)\}_{G\in HC}$ is computationally indistinguishable from $\{\text{View}^P_{W^*}(G)\}_{G\in HC}$.

Next, we turn to prove its validity by constructing a knowledge extractor $\mathcal{K}$. For convenience, we refer to $(B^0_1, B^1_1), \cdots, (B^0_n, B^1_n)$ as an accepted answer to $\zeta = \zeta_1 \cdots \zeta_n$ if $check_{\zeta_1}(C_1, B^{\zeta_1}_1) = 1, \cdots, check_{\zeta_n}(C_n, B^{\zeta_n}_n) = 1$.

On input x and access to the prover-strategy oracle P^*, $\mathcal{K}$ proceeds as follows:

1. On input x, invoke P^* and obtains its responses: $\overline{C} = (C_1, \cdots, C_n)$ and e.
2. Compute $y, (y^0_1, y^1_1), \cdots, (y^0_n, y^1_n)$ as in **V1**, and then feed P^* with $(y, (y^0_1, y^1_1), \cdots, (y^0_n, y^1_n))$. Next, obtain the response $\delta = \delta_1 \cdots \delta_n$.
3. Query P^* with $(\theta^{\delta_1}_1, \cdots, \theta^{\delta_n}_n)$, and $(x^{\delta_1}_{1,1}, \cdots, x^{\delta_1}_{1,n}), \cdots, (x^{\delta_n}_{n,1}, \cdots, x^{\delta_n}_{n,n})$, and then obtain $(T_1, \cdots, T_n)$ and ε, where $T^0_j = (d^0_j, s^0_j), T^1_j = (d^1_j, s^1_j)$.
4. Compute $B^0_k = s^0_k \oplus G^{p_0(n)}(z^0_k)$, $B^1_k = s^1_k \oplus G^{p_1(n)}(z^1_k)$, $k = 1, \cdots, n$, where $z^b_k = x_k + d^b_k$. If there exists $\zeta = \zeta_1 \cdots \zeta_n$, so that $(B^0_1, B^1_1), \cdots, (B^0_n, B^1_n)$ is an acceptable answer to ζ, then proceed to the next step. Otherwise, stop without any output.
5. If there exists k such that $check_{1-\zeta_k}(C_k, B^{1-\zeta_k}_k) = check_{\zeta_k}(C_k, B^{\zeta_k}_k) = 1$, compute a cycle H from $B^{\zeta_k}_k$ and $B^{1-\zeta_k}_k$. Output H and stop.
6. Otherwise, repeat the following.
   - Feed P^* with $(y, (y^0_1, y^1_1), \cdots, (y^0_n, y^1_n))$ computed as in **V1**, and then get the response $\delta' = \delta'_1 \cdots \delta'_n$.
   - Invoke P^* with $(\theta^{\delta'_1}_1, \cdots, \theta^{\delta'_n}_n)$ and $(x^{\delta_1}_{1,1}, \cdots, x^{\delta_1}_{1,n}), \cdots, (x^{\delta_n}_{n,1}, \cdots, x^{\delta_n}_{n,n})$, and then obtain $(T_1, \cdots, T_n)$ and ε.

- Compute $B'^0_j = s^0_j \oplus G^{p_0(n)}(z^0_j)$ and $B'^1_j = s^1_j \oplus G^{p_1(n)}(z^1_j)$ $(j = 1, \cdots, n)$ as in step 4. If there exists k, such that $check_{1-\zeta_k}(C_k, B'^{1-\zeta_k}_k) = 1$ or $check_0(C_k, B'^0_k) = check_1(C_k, B'^1_k) = 1$ then terminate the repeat and proceed to the next step.
- Else if $(B'^0_1, B'^1_1), \cdots, (B'^0_n, B'^1_n)$ is acceptable only for ζ, then terminate the repeat and stop without output.

7. Compute a cycle H from $B^{\zeta_l}_k$ and $B'^{1-\zeta_k}_k$ or from B'^0_k and B'^1_k. Output H and stop.

We now show two facts: (1) $\mathcal{K}$ runs in the expected polynomial time, and (2) if the probability (denoted by p) that P^* convinces the verifier of the proved statement is greater than $2^{-n/3}$, then $\mathcal{K}$ succeeds in computing a Hamiltonian cycle in G with a probability of at least $p - 2^{-2n/3}$.

Note that $\mathcal{K}$ clearly runs in strict polynomial time when $p = 0$. So we assume that $p > 0$ in what follows. Let E denote the event that the knowledge extractor $\mathcal{K}$ terminates the repeat successfully, $p' = Pr[E]$. Note that $p' \geqslant p$, and then the expected number of times of is $\frac{1}{p'}$. Therefore, the expected running time of $\mathcal{K}$ is given by

$$(1-p) \cdot poly(n) + p \cdot \tfrac{1}{p'} \cdot poly(n) = poly(n)$$

To show the second fact, we note that $\mathcal{K}$ terminates its execution without output when either (1) it does not obtain $(B^0_1, B^1_1), \cdots, (B^0_n, B^1_n)$ that is acceptable, or (2) assuming $(B^0_1, B^1_1), \cdots, (B^0_n, B^1_n)$ is acceptable for ζ, it terminates the repeat on finding $(B'^0_1, B'^1_1), \cdots, (B'^0_n, B'^1_n)$ which is acceptable only for $\zeta' = \zeta$.

If the prover does not know of any Hamilton cycle in G, the probability that P^* gives a successful proof is at most 2^{-n}, that is, $p \leqslant 2^{-n}$. In fact, it is the probability that the prover correctly guesses $\sigma = \sigma_1 \cdots \sigma_n$. If $p > 2^{-n/3}$, then there must exist $1 \leqslant k_0 \leqslant n$ such that

$$Check_0(C_{k_0}, A^0_{k_0}) = Check_1(C_{k_0}, A^1_{k_0}) = 1$$

Therefore, $p' = Pr[E] \geqslant p > 0$. Note ζ' is uniformly distributed and then $Pr[\zeta' = \zeta] = 2^{-n}$. Thus, the probability that case (2) takes place is given by

$$2^{-n} + (1-p') \cdot 2^{-n} + \cdots + (1-p')^k \cdot 2^{-n} + \cdots$$
$$= 2^{-n} \cdot \tfrac{1}{p'} \leqslant 2^{-2n/3}$$

It follows that $\mathcal{K}$ outputs a Hamilton cycle $H \subset G$ with probability $p - 2^{-2n/3}$ when $p > 2^{-n/3}$. ■

Since claw-free trapdoor permutations imply the existence of non-interactive perfectly binding commitment schemes, we get the following theorem.

Theorem 3 Every NP language has 5-round zero-knowledge proofs of knowledge (with negligible knowledge error), provided that claw-free trapdoor permutations exist.

By combining Theorems 1 and 3, we obtain our main theorem.

1 M. Blum. How to prove a theorem so no one else can claim it. Proceedings of the International Congress of Mathematicians, California, USA,1986:1444-1451.

2 B. Barak. How to go beyond the black-box simulation barrier. In 42th Annual Syposium on Foundation of Computing Science,IEEE Computer Society, 2001:106-115.

3 B. Barak. Non-black-box techniques in cryptography. Thesis for the Ph. D. Degree, Weizmann Institute of Science, 2004:53-102.(http://www.math.ias.edu/ boaz/index.html)

4 B. Barak, Y. Lindell, and S. Vadhan. Lower bounds for non-black-box zero knowledge. In 44th Annual IEEE Symposium Foundations of computer science, IEEE Computer Society 2003: 384-393.

5 B. Barak, Y. Lindell, and S. Vadhan. Lower bounds for non-black-box zero knowledge. Journal of Computer and System Sciences, 2006, 72(2): 321-391.

6 B. Barak, Y. lindell. Strict polynomial-time in simulation and extraction. In 34th STOC, 2002, pasges 484-493.

7 G. Brassard, C. Crepeau, and M. Yung. Constant-round perfect zero-knowledge computationally convincing protocols. Theoretical Computer Science, 1991, Vol.84:23-54.

8 M. Bellare, O. Goldreich. On defining proofs of knowledge. Advances in Cryptology-CRYPT'92, LNCS, vol.740, Springer-Verlag,1992, pages 390-420.

9 M. Bellare, O. Goldreich. On probabilistic versus deterministic provers in the definition of proofs of knowledge.

10 M. Bellare, M. Jakobsson, and M. Yung. Round-optimal zero-knowledge arguments base on any one-way function. In EURO-CRPT'97, LNCS, Vol.1233, Spring-Verlag,1997:280-305.

11 M. Bellare, A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocol. http://eprint.iacr.org/2003

12 M. Bellare, M. Yung. Certifying permutations: non-interactive zero-knowledge based on an trapdoor permutation. Journal of Cryptology, 1996, 9(1):149-166.

13 U. Feige, A. Shamir. Zero knowledge proofs of knowledge in two rounds. In proceedings of CRYPTO'89, Berlin:Springer-Verlag,1989:526-545.

14 O. Goldreich, L.Levin. A hard predicate for all one-way functions. In Proceeding of the 21st Annu. ACM Symp. on the theory of Computing,

1989:25-32.

15 O. Goldreich, H. Krawczyk. On the composition of zero-knowledge proof systems. SIAM Journal of Computing, 1996, 25(1):169-192.

16 S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. SIAM Journal on computing, 1989, 18(16):186-208.

17 O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. of the ACM, 1991, 38(3):691-729.

18 O. Goldreich, Y. Oren. Definitions and properties of zero-knowledge proof systems. Journal of Cryptology, 1994, 7(1):1-32.

19 O. Goldreich, A. Kahan. How to construct constant-round zero-knowledge proof system for NP. Journal of Cryptology, 1996,9(3):167-189.

20 O. Goldreich. Foundations of Cryptography - Basic Tools. Cambridge University Press, 2001.

21 S. Halevi, S. Micali. More on proofs of knowledge. http://eprint.iacr.org/1998/015.

22 S. Hada, T. Tanaka. On the existence of 3-round zero-knowledge protocol. Advances in Cryptology-CRYPT'98, LNCS, vol.1462, H.Krawczyk ed., Springer-Verlag,1998. (Preliminary version of [23])

23 S. Hada, T. Tanaka. On the existence of 3-round zero-knowledge protocol. http://eprint.iacr.org/1999/009. (Final version of [22])

24 Itoh Toshiya, Sakurai Kouichi. On the Complexity of Constant Round ZKIP of Possession of Knowledge. IEICE TRANS. FUNDAMENTALS, 1993, VOL. E76-A, NO, 1:31-39.

25 J. Katz. Which languages have 4-round zero-knowledge proofs. In Fifth Theory of Cryptography Conference, LNCS Vol. 4948, Spring-Verlag,2008:73-88.

26 M. Lepinski. On the existence of 3-round zero-knowledge proofs. Thesis for the degree of master, Massachusetts Institute of Technology,2002.(http://citeseer.ist.psu.edu/lepinski01existence.html)

27 Li Hongda, Li Bao. The existence of 3-round zero-knowledge proof systems for NP. Science in China Series F-Information Sciences, 2008, 51(3):273-282.

28 M. Naor. On Cryptographic Assumptions and Challenges. In In proceedings of Advances in Cryptology-CRYPT'2003, LNCS, vol 2729, 2003, pages 96-109.

29 A. Rosen. A note on constant-round zero-knowledge proofs for NP. In First Theory of Cryptography Conference(TCC), LNCS 2951, Spring-Verlag, 2004:191-202.

30 Y. Dodis, L. Reyzin. On the power of claw-free permutations. In Third Conference on Security in Communication Networks'02, LNCS, Vol. 2576, Spring-Verlag,2003:55-73.