Augmented Black-Box Simulation and Zero Knowledge Argument for NP

Li Hongda, Pan Dongxue, Ni Peifang

The Data Assurance and Communication Security Research Center, School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100093, China

Abstract. The standard zero knowledge notion is formalized by requiring that for any probabilistic polynomial-time (PPT) verifier V^*, there is a PPT algorithm (simulator) S_{V^*}, such that the outputs of S_{V^*} is indistinguishable from real protocol views. The simulator is not permitted to access the verifier V^*'s private state. So the power of S_{V^*} is, in fact, inferior to that of V^*.

In this paper, a new simulation method, called augmented black-box simulation, is presented by permitting the simulator to have access to the verifier's current private state in a special manner. The augmented black-box simulator only has the same computing power as the verifier although it is given access to the verifier's current private state. Therefore, augmented black-box simulation is a reasonable method to prove zero knowledge property, and brings results that hard to obtain with previous simulation techniques. Zero knowledge property, proved by means of augmented black-box simulation, is called augmented black-box zero-knowledge.

We present a 5-round statistical augmented black-box zero-knowledge argument for Exact Cover Problem under the Decision Multilinear No-Exact-Cover Assumption. In addition, we show a 2-round computational augmented black-box zero-knowledge argument protocol for Exact Cover problem under the Decision Multilinear No-Exact-Cover Assumption and the assumption of the existence of hash functions. It is well known that 2-round zero knowledge protocols does not exist under general zero knowledge notion. Besides, following [18], we consider leakage-resilient property of augmented black-box zero knowledge, and prove that the presented statistical zero-knowledge protocol has optimal leakage-resilient property.

Key word: zero-knowledge proofs (arguments), black-box simulation, constant-round, Exact-Cover problem, leakage-resilient.

1 Introduction

Zero-knowledge proofs were first introduced by Goldwasser, Micali, and Rackoff in [20]. They are interactive proof systems executed by two parties, one is the prover P, and the other is the verifier V. Zero-knowledge property requires that P can convince V of the correctness of the statement $x \in L$ without providing V with any additional information beyond the fact that the statement is true. Formally, zero-knowledge property requires that for every probabilistic polynomial-time (PPT) verifier V^* there exists

a PPT algorithm (simulator) S, such that the two distribution, the output of $S(x)$ and the view of V^* in the real protocol, are indistinguishable. According to the relation between these two distributions, zero-knowledge property is divided into computational zero knowledge and statistical zero knowledge, responding to computationally or statistically indistinguishable condition. Besides, according to the limit on the power of the prover, zero-knowledge protocols are divided into two types, zero-knowledge proofs (ZKP) and zero-knowledge arguments (ZKA).

In fact, the original notion of zero knowledge in [20] is computational zero knowledge and only considers the stand-alone setting, where the protocol runs in complete isolation. Such zero knowledge property cannot be maintained when many protocols are run concurrently. In order to construct more practical zero-knowledge protocols, Dolev, Dwork and Naor [13] introduced non-malleable zero knowledge and [15] studied concurrent zero-knowledge. Following these two works, a lot of works, such as [19, 27, 31, 11, 34, 30], devoted to studying protocols with concurrent zero-knowledge property and non-malleable zero knowledge property.

Statistical zero-knowledge proofs have statistical zero-knowledge property and statistical soundness. Unfortunately, [16, 2] showed that no language beyond the languages in $AM \cap coAM$ has statistical zero-knowledge proofs. [27] provided concurrent statistical zero-knowledge proofs for a variety of non-trivial problems without any complexity assumption.

A weaker notion is statistical zero-knowledge arguments, first presented by Brassard, Chaum, and Crepeau in [4]. G. Brassard gave the first perfect zero-knowledge protocol for any statement in NP with constant rounds, which is a 6-round perfect zero-knowledge argument for NP [5], where the perfect zero-knowledge property means that the output of the simulator is identical to the real protocol views. Naor et al. [29] showed a construction of a statistically hiding bit-commitment scheme and then obtained an efficient perfect zero-knowledge argument, with non-constant round, for all of NP from any one-way permutation. Their result gave the first general reduction that zero-knowledge arguments for NP can be constructed given any one-way permutation. Nguyen et al [28] constructed a relaxed variant of statistically hiding commitments, called 1-out-of-2-binding commitments, and then obtained a statistical zero-knowledge argument for NP with a polynomial number of rounds under the (minimal) complexity assumption that one-way functions exist. There are also works [19, 30, 24] devoted to concurrent statistical zero-knowledge arguments and non-malleable statistical zero-knowledge arguments. In this paper, we also consider statistical zero-knowledge arguments for NP and present a 5-round protocol. (reducing the round complexity)

Traditionally, zero-knowledge property is formalized by simulation, and two different kinds of simulator, black-box simulator and non-black-box simulator, are used. The black-box simulator is only given oracle access to the verifier's strategy program but can query it starting from any point. To simulate the real interaction between a prover and a verifier, one of advantage of the black-box simulator is that it can rewind the verifier's strategy. By rewinding the simulator can extract some verifier's private information needed for the simulation. Unfortunately, there are some fundamental tasks that cannot be achieved by black-box simulation, such as constant-round concurrent zero knowledge etc.. The non-black-box simulator has more methods to make use of the

verifier's strategy, such as to utilize its strategy description as Barak does in [3]. Until Barak presented his non-black-box simulation technique, almost all zero knowledge protocols are black-box zero knowledge. Under the existence of collision-resistant hash functions,. Barak first gave a bounded concurrent constant-round zero knowledge argument for NP which was proved not to exist in black-box setting [3]. Subsequently, a lot of works followed the Barak's work [6, 33, 11, 26]. Specifically, [12] showed a constant-round resettably-sound zero-knowledge argument based only on the existence of one-way functions by non-black-box simulation.

Recently, Garg et al. [18] introduced leakage-resilient zero-knowledge (LRZK), which guarantees the zero-knowledge property in stronger adversarial models where an malicious verifier has the ability to obtain leakage of the secret states of the honest prover by launching a side-channel attack. LRZK requires that for any cheating verifier V^* that can obtain l bits of leakage information about the prover's state via a series of leakage queries $f_1, f_2, ...$, there exists a simulator S with the leakage oracle L_w^n such that the simulator can output an indistinguishable view of the malicious verifier and simulate the leakage as well.

In this paper, we consider zero-knowledge property by a new simulation paradigm, called augmented black-box simulation. The formal definition of zero knowledge under this simulation paradigm is first presented in our another work. Roughly speaking, the simulator in the augmented black-box setting is given not only an oracle access to the verifier's strategy program but also an access to the verifier's current private information in a special manner. Let $\langle P, V\rangle$ be an interactive proof system for language L, the augmented black-box simulation is formalized as follows.

Consider an extended verifier $\widehat{V}$: $\widehat{V}$ has the same strategy and interacts with the prover P as V, and simultaneously outputs V's current private state secretly on its private output tape. The black-box simulator of $\langle P, \widehat{V}\rangle$ with oracle access to the extended verifier $\widehat{V}$ and its private output tape is called the augmented black-box simulator of the interactive proof system $\langle P, V\rangle$. $\langle P, V\rangle$ is known as augmented black-box zero-knowledge if for any malicious V^*, there exists an augmented black-box simulator $S^{\widehat{V^*}}$, such that the output of $S^{\widehat{V^*}}$ is indistinguishable from the view of V^* in the execution with P. The simulator is permitted to see all that the verifier currently has used, that is, when the simulator receives a message m from V^*, it is simultaneously permitted to get V^*'s private information used by V^* to computes m. It is obvious that the augmented black-box simulator only has the same computing power as the verifier although it given access to the verifier's current private state. Therefore, augmented black-box simulation is a reasonable method to prove zero knowledge property. In addition, using augmented black-box simulation can simplify proof of zero knowledge property, and help to bring simpler and more efficient zero knowledge protocol that may be impossible to achieve under the standard (non-)black-box simulation. It is obvious that any standard black-box zero-knowledge proof for a promise problem Π is also augmented black-box zero knowledge. Conversely, under a proper conditions, such as existence of knowledge of zero-knowledge proof for NP, any augmented black-box zero-knowledge protocol can also be translated into a general black-box zero-knowledge protocol.

1.1 Our results

In this paper, we present a 5-round statistical zero-knowledge argument and a 2-round computational zero knowledge argument for Exact Cover problem $\Pi = (EC_Y, EC_N)$, a promise problem which is NP-complete, under the notion of augmented black-box zero-knowledge. This notion of zero-knowledge is formalized by augmented black-box simulation technique, where the simulator has oracle access to an extended verifier $\widehat{V}$. $\widehat{V}$ has one more private output tape than the verifier V. And $\widehat{V}$ outputs the message V outputs together with V's current private state, from which the message is computed.

Our construction of the 5-round statistical zero-knowledge argument includes two 3-round interactive arguments for two Exact Cover problems. The construction consists of three stages. In the first stage, the prover P and the verifier V execute the first two round of the 3-round interactive argument protocol for $x \in EC_Y$. In the second stage, P and V execute the 3-round interactive argument, in which V proves to P $y \in EC_Y$, where y is obtained by reducing part of the messages in the first stage to an Exact Cover problem. And in the last stage, P proceeds the last round of the 3-round interactive argument for $x \in EC_Y$, P completes the proof by sending a value (which can also be computed by V) that is computed from the value specified by V in the first stage. V accepts if and only if the value sent by P is correct. The soundness is guaranteed by the Decision Multilinear No-Exact-Cover Assumption and the simulation is completed by a PPT simulator with oracle access to $\widehat{V}$. Hence then, S can obtain V's private state and thus obtain the value specified by V in the first stage and use it to complete the proof.

Our 2-round computational zero-knowledge argument for Exact Cover Problem is very simple. Except the Decision Multilinear No-Exact-Cover Assumption, the protocol has used collision-resistant hash functions. Since in our protocol, a cheating verifier cannot verify the correctness of the proof from an honest prover, the construction of BPP simulator in [21] does not applicable for our 2-round zero-knowledge protocol. Therefore, we claim that our result does not contradict with the well known lower-bound of Goldreich and Oren [21], which says that 2-round auxiliary input zero knowledge argument exists only for trivial languages.

In addition, we consider leakage-resilient property of augmented black-box zero knowledge, and prove that the given statistical zero-knowledge argument is leakage-resilient. The length of leakages obtained by the leakage-resilient simulator from the leakage oracle is the same as that of the leakages obtained by the verifier.

1.2 Related work

Zero Knowledge Relevant to our work are the works on simulation techniques for zero knowledge. Besides the black-box simulation and Barak's non-black-box simulation technique, there are some other non-black-box simulation techniques.

One another non-black-box simulation technique is based on the knowledge assumption[23, 7, 1, 22, 34]. In essence, the KEAs require that if an adversary can complete some task, it must have "knowledge" of the value specified by the task. In non-black-box simulation based on KEAs, the simulator can extract the value from the verifier by an efficient algorithm and then complete the simulation.

Recently, based on the impossibility of program obfuscation, [9] gave a new non-black-box simulation technique . Assuming that the malicious verifier chooses an unobfuscatable function, of which the properties consist of black-box unlearnablility and non-black-box learnability. The prover interacting with the verifier has oracle access to this function and thus the prover cannot learn this function from the query-answer behavior. The simulator with the code of V^*'s algorithm, in essence, is given an obfuscation of V^*'s algorithm. By the non-black-box learnability of the unobfuscatable function, the simulator can learn this function and then simulate the verifier's view. [10] constructed robust unobfuscatable functions, by which they reduced the assumptions required for resettably-sound zero-knowledge to one-way functions.

[14] promoted the simulator's ability by using the "distinguisher-dependent" simulator, which permits the simulator to know the possibly cheating verifier's program V^*, the distinguisher T, and the distribution D on inputs. That is, for any V^* and PPT distinguisher T, there exists a simulator S_T, such that for any distribution D, the output of S_T is indistinguishable from the real view for T. By the distinguisher-dependent simulator, C. Dwork et al. in [14] weakened the definition of zero-knowledge and proved the existence of 3-round weak zero-knowledge argument for NP, which was recently constructed from the obfuscation of point functions by Bitansky and Paneth [8].

Leakage-resilient Zero Knowledge In [18], Garg et al. showed a $(1+\epsilon)$-LRZK proof system under a standard general assumption (the existence of statistically hiding commitment scheme that is public-coin w.r.t. the receiver) for any $\epsilon > 0$. That is, for any $\epsilon > 0$, there exists a proof system such that for any PPT verifier which can obtain l bits leakage there exists a simulator, obtaining at most $(1+\epsilon)\cdot l$ bits of leakage from the leakage oracle L_w^n, can simulate the verifier's view. The round complexity of the protocol in [18] is at least $\omega(\log n)/\epsilon$, where n is the security parameter. Pandey [32] constructed the first constant-round LRZK argument with $\epsilon = 0$ under the assumption of DDH and collision-resistant hash functions. Very Recently, Kiyoshima [25] constructed the first LRZK argument system only under the existence of the collision-resilient hash function family.

1.3 Organization

Section 2 contains the standard definitions, cryptographic tools, and the relaxed notion of zero knowledge used in our protocols. In section 3, we present a 5-round statistical zero-knowledge argument for Exact-Cover problem. In section 4, we first recall the model of leakage-resilient zero knowledge, and prove the construction in section 3 is a leakage-resilient zero-knowledge argument. In section 5, we show a 2-round computational zero-knowledge argument for Exact-Cover problem.

2 Preliminaries

In this paper, we use some standard notations. Let $A(\cdot)$ be a probabilistic algorithm and let $A(x)$ be the result of running algorithm A on input x, then we use $y = A(x)$ (or $y \leftarrow A(x)$) to denote that y is set as $A(x)$. For a finite set $\mathcal{S}$, we use $y \in_R S$ to denote that y is uniformly selected from $\mathcal{S}$. We use $[l]$ to denote the set $\{1, 2, \cdots, l\}$. We write

$neg(\cdot)$ to denote an unspecified negligible function, $poly(\cdot)$ an unspecified polynomial. We use "$X \stackrel{c}{=} Y$" ("$X \stackrel{s}{=} Y$") to denote that probabilistic distributions X and Y are computationally (statistically) indistinguishable. Unless otherwise stated, we use λ to denote the security parameter.

We recall the following definitions.

2.1 Exact Cover problem

Recall the well-known NP-complete problem, Exact Cover problem: Given a finite set X, and a collection of subsets of X, $\mathcal{T} = \{T_1, \cdots, T_l : T_i \subseteq X\}$, decide whether there is a subset $\mathcal{T}' \subseteq \mathcal{T}$ such that every element of X lies in exactly one element of $\mathcal{T}'$, which equals that whether there exists a subset $I \subseteq \{1, \cdots, l\}$ such that $\cup_{i \in I} T_i = X; \forall i \neq j \in I, T_i \cap T_j = \emptyset$.

Let EC_Y and EC_N be all "Yes" and "No" instances of the Exact Cover problem respectively. A witness of an "Yes " instance $(X, \mathcal{T} = \{T_1, \cdots, T_l\})$ is the subset $I \subseteq [l]$ such that $\{T_i : i \in I\}$ is a partition of X. The corresponding NP-relation is denoted by R_{EC}.

2.2 Multi-linear map

Let $G_1, \cdots, G_n$ be a sequence of groups of the same order p, and $g_1, \cdots, g_n$ be the corresponding generators. Define a set of maps

$$e_{i,j} : G_i \times G_j \to G_{i+j}, i + j \leq n$$

satisfying $e_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab}$ for any $a, b \in Z_p$. Then,

$$\{e_{i,j} : G_i \times G_j \to G_{i+j} : i, j \geq 1, i + j \leq n\}$$

is called multilinear map. For convenience, we write $e_{i,j}(g_i^a, g_j^b)$ as $e(g_i^a, g_j^b) = g_{i+j}^{ab}$.

Similarly, for any $i_1, \cdots, i_t \in [n]$ with $i_1 + \cdots + i_t \leq n$, we define multilinear map

$$e\left(g_{i_1}^{a_{i_1}}, \cdots g_{i_t}^{a_{i_t}}\right) = (g_{i_1 + \cdots + i_t})^{a_{i_1} \cdots a_{i_t}}.$$

Assume there is a PPT generation algorithm $\mathcal{G}$. The output of $\mathcal{G}(1^\lambda, n; r)$, where $r \in_R \{0,1\}^{poly(\lambda}$ includes the description of a sequence of groups $G_1, \cdots, G_n$ of the same prime order p, the corresponding generators $g_1, \cdots, g_n$, and a multilinear map e. For simplicity, the out of $\mathcal{G}(1^\lambda, n; r)$ is denoted by pp.

Definition 1. ***Decision Multilinear No-Exact-Cover Problem. [17]*** *Let $x = \{X; \mathcal{T}\}$ be a "No" instance of Exact Cover Problem ($x \in EC_N$), where $|X| = n$. Let*

$$((G_1, g_1), \cdots, (G_n, g_n), e, p) \leftarrow \mathcal{G}(1^\lambda, n; r)$$

where the prime order $p = p(\lambda)$ and λ is the security parameter. Let $\ell = |\mathcal{T}|$ and $a_1, \cdots, a_n, r$ be uniformly selected from Z_p. For every $i \in [\ell]$, let $h_i = (g_{|T_i|})^{\prod_{j \in T_i} a_j}$. Then, the Decision Multilinear No-Exact-Cover Problem is to distinguish between the two distributions: $(pp; h_1, \cdots h_\ell; g_n^{a_1 \cdots a_n})$ and $(pp; h_1, \cdots h_\ell; g_n^r)$.

Definition 2. ***Decision Multilinear No-Exact-Cover Assumption. [17]*** *The Decision Multilinear No- Exact-Cover Assumption is that for every instances $x \in EC_N$, all PPT algorithms A can solve the Decision Multilinear No-Exact-Cover Problem with negligible probability.*

2.3 Zero knowledge

Definition 3. ***Interactive Proof System.*** *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 $x \in L$,*

$$\Pr[\langle P, V\rangle(x) = 1] > 1 - c(|x|)$$

- *Soundness: There exists a negligible function s such that for every $x \notin L$ and every interactive machine B, it holds that*

$$\Pr[\langle P, V\rangle(x) = 1] < s(|x|)$$

$c(\cdot)$ is called the completeness error, and $s(\cdot)$ the soundness error.

An interactive proof $\langle P, V\rangle(x)$ for a promise problem Π is zero-knowledge if the possibly malicious verifier V learns nothing from the interaction with P except the fact that $x \in \Pi_Y$ being proven is true. It implies that whatever V could learn from the interaction could be obtained by V with an oracle $\mathcal{O}_\Pi$. When being queried with an instance $x \in \Pi$, $\mathcal{O}_\Pi$ returns 1 if $x \in \Pi_Y$ and returns 0 otherwise.

Let $View^P_{V(z)}(x)$ denote the view of V with auxiliary input z in the real execution of the protocol with P. Thus $View^P_{V(z)}(x)$ includes the random coins of V and the messages received from P. It is reasonable that whatever V learned from the interaction can be computed from $View^P_{V(z)}(x)$. Therefore, zero knowledge requires that for any PPT V^*, there is a simulator S with some advantage against P, such that the output of S is indistinguishable from $View^P_{V^*(z)}(x)$.

Definition 4. ***Zero-knowledge Proof.*** *Let $\langle P, V\rangle$ be an interactive proof system for a promise problem Π. $\langle P, V\rangle$ is said to be zero-knowledge if for every PPT malicious verifier V^* there exists a PPT algorithm $\mathcal{S}$ such that $\{View^P_{V^*(z)}(x)\}_{x\in\Pi_Y, z\in\{0,1\}^*}$ and $\{\mathcal{S}(x, z)\}_{x\in\Pi_Y, z\in\{0,1\}^*}$ are computationally indistinguishable.*

If $\{View^P_{V^*(z)}(x)\}_{x\in\Pi_Y, z\in\{0,1\}^*}$ and $\{\mathcal{S}(x, z)\}_{x\in\Pi_Y, z\in\{0,1\}^*}$ are statistically indistinguishable, the interactive proof system $\langle P, V\rangle$ is called statistical zero-knowledge.

2.4 Augmented black-box Zero knowledge

Let $\langle P, V\rangle$ be an interactive proof system for a promise problem Π. Since the last message of the execution is almost always sent to the verifier V by the prover P, assume that the verifier V sends the first message. If in some protocols P sends the real first

message, we assume that the verifier V sends an empty string λ before P's. And we assume that the round number of $\langle P, V\rangle$ is $2k(n)$, where n is the security parameter and $k(n)$ is a positive polynomial. V's next message depends on its random coins and all messages that have been received from P so far. And P's next message depends on its private input, random coins, and all messages received. In the $(2i-1)$th round, V sends α_i, and then receives β_i responded by P in the $(2i)$th round, $i = 1, \cdots, k$. We use $\alpha_i = \perp$ ($\beta_i = \perp$) to denote V (P) aborts. At the end of the interaction, the verifier V decides whether to accept or reject the proof.

We use $Next_V(x, z; \cdot)$ to denote the next message function of V with common input x and auxiliary z, and let $r_V = r_1 \cdots r_k$ be V's all random coins. In $(2i-1)$th round, V first computes $\alpha_i = Next_V(x, z; i, r_i, \{\beta_j\}_{j\in[i-1]})$ with random coins r_i and all received messages $\{\beta_j\}_{j\in[i-1]}$, and then sends α_i to P, $i \in [k]$. V's private information used to generate α_i is denoted by $state_V^{(i)}$, which will be stored when V sends α_i.

To introduce the augmented black-box simulator, we first recall the definition of the extended verifier $\widehat{V}$ (an imaginary verifier) for any V. $\widehat{V}$ does the same as V when it interacts with P except that it records its secret state on the private output tape, the tape V does not have. That is, in $(2i-1)$th round, $\widehat{V}$ computes α_i and then sends α_i to P as V, for $i \in [k]$, and at the same time $\widehat{V}$ honestly writes $state_V^{(i)}$ on its private output tape. Obviously, the augmented verifier $\widehat{V}$ differs from V only in the output of V's private information $state_V^{(i)}$, which is used to generate α_i. Hence then, the next message function of $\widehat{V}$, denoted by $Next_{\widehat{V}}(x, r_V, \cdot, \cdot)$, is defined as follows:

$$\left(\alpha_i, state_V^{(i)}\right) \leftarrow Next_{\widehat{V}}\left(x, r_V; i, \{\beta_j\}_{j\in[i-1]}\right), i = 1, \cdots, k$$

where $\alpha_i = Next_V(x, r_V, i, \{\beta_j\}_{j\in[i-1]})$ is written on $\widehat{V}$'s communication output tape, and V's private information $state_V^{(i)}$, used to generate α_i, is written on $\widehat{V}$'s private output tape. For any malicious verifier V^*, the corresponding extended verifier $\widehat{V}^*$ computes α_i with the next message function selected by V^*.

1. Uniformly select r_V, and set $i = 1$, $\widetilde{\beta}_0 = \lambda$.
2. Make a query to $\mathcal{O}_{\Pi}$ with x. It returns b.
3. Invoke $\mathcal{O}_{\widehat{V}}(x, r_V; \cdot)$ with $\left(i, \widetilde{\beta}_{i-1}\right)$, and receive $\left(\alpha_i, state_V^{(i)}\right)$.
4. Verify α_i as an honest prover. If the verification fails, output $\left(x, r_V, \{\widetilde{\beta}_j\}_{j\in[i-1]}\right)$ and stop.
5. Else, verify that α_i is computed correctly from $state_V^{(i)}$.
   - If the verification succeeds and $b = 1$, prepare $\widetilde{\beta}_i$ from x, $\{state_V^j\}_{j\in[i]}$, $\{\alpha_j\}_{j\in[i]}$ and $\{\widetilde{\beta}_j\}_{j\in[i-1]}$, such that $\widetilde{\beta}_i$ is acceptable by $\widehat{V}$. If no correct $\widetilde{\beta}_i$ can be obtained, fail and stop.
   - Else, randomly select $\widetilde{\beta}_i$ from a proper set.
6. $i \leftarrow i + 1$ and return to step 3 if $i \leq k$.
7. Output $\left(x, r_V, \{\widetilde{\beta}_j\}_{j\in[k]}\right)$.

Figure 1. Augmented black-box simulator $\boldsymbol{S^{\mathcal{O}_{\Pi}, \mathcal{O}_{\widehat{V}}}(x)}$

Let $\mathcal{O}_{\widehat{V}}$ be the oracle of the next message function of $\widehat{V}$. $\mathcal{O}_{\widehat{V}}(x, r; \cdot)$ indicates $\widehat{V}$ with common input x and random coins r. That is, when being queried with messages $(i, \{\beta_j\}_{j\in[i-1]})$, $\mathcal{O}_{\widehat{V}}(x, r; \cdot)$ first computes $\alpha_i = Next_V(x, r; i, \{\beta_j\}_{j\in[i-1]})$ with the private state $state_V^{(i)}$ and then answers with $(\alpha_i, state_V^{(i)})$.

Then, the augmented black-box simulator S with oracle access to $\mathcal{O}_{\widehat{V}}(\cdot)$ receives both the verifier's message α_i and its private state $state_V^{(i)}$ used to compute α_i, and then S can complete the simulation more effectively. For any V, the construction of the PPT algorithm S with access to oracle $\mathcal{O}_{\Pi}$, which outputs 1 when being queried with $x \in \Pi_Y$ and outputs 0 otherwise, and oracle $\mathcal{O}_{\widehat{V}}$, written by $S^{\mathcal{O}_{\Pi}, \mathcal{O}_{\widehat{V}}}$, is depicted in Figure 1.

Therefore, the augmented black-box zero knowledge is formally defined by requiring that for any verifier V^* with auxiliary input z (the corresponding extended verifier is $\widehat{V^*}$), there is a PPT algorithm S with the oracles $\mathcal{O}_{\Pi}$ and $\mathcal{O}_{\widehat{V^*}}$, such that the output of $S(x, z)$ and the real protocol view $View^P_{V^*(z)}(x)$ are indistinguishable.

Definition 5. ***Augmented Black-box Zero-knowledge Proof.*** *Let $\langle P, V\rangle$ be an interactive proof system for a promise problem Π. $\langle P, V\rangle$ is an called (auxiliary-input) augmented black-box computational zero-knowledge proof if for every PPT V^*, the corresponding extended verifier is denoted by $\widehat{V^*}$, there exists an augmented black-box simulator $S^{\mathcal{O}_{\Pi}, \mathcal{O}_{\widehat{V^*}}}$, such that*

1) *The probability that $S^{\mathcal{O}_{\Pi}, \mathcal{O}_{\widehat{V^*}}}(x, z)$ fails is at most $\frac{1}{2}$.*
2) *Under the condition that the augmented black-box simulator does not fail, the real protocol view $View^P_{V^*(z)}(x)$ and $S^{\mathcal{O}_{\pi}, \mathcal{O}_{\widehat{V^*}}}(x, z)$ are computationally indistinguishable. That is, $\left\{S^{\mathcal{O}_{\pi}, \mathcal{O}_{\widehat{V^*}}}(x, z)\right\} \stackrel{c}{=} \left\{View^P_{V^*(z)}(x)\right\}$*

If the simulation $S^{\mathcal{O}_{\Pi}, \mathcal{O}_{\widehat{V^*}}}(x, z)$ is statistically closed or identical to the real protocol view $View^P_{V^*(z)}(x)$ under the condition that the augmented black-box simulator does not fail, $\langle P, V\rangle(x)$ is an called statistical or perfect augmented black-box zero knowledge proof.

3 Constant-round zero-knowledge arguments for Exact Cover problem

The goal in this section is to construct constant-round augmented black-box zero-knowledge argument protocols for Exact Cover problem $\Pi_{EC} = (EC_Y, EC_N)$, an NP-complete promise problem, under the Decision Multilinear No-Exact-Cover Assumption.

Assume $x = (X; \mathcal{T}) \in \Pi$, $X = \{x_1, \cdots, x_n\}$, $\mathcal{T} = (T_1, \cdots, T_l)$. If $x \in EC_Y$ then we have $\exists I \subseteq \{1, \cdots, l\}, \cup_{i\in I} T_i = X; \forall i \neq j \in I, T_i \cap T_j = \emptyset$.

Let $\mathcal{G}(1^\lambda, n)$, where $n = |X|$, be a PPT generation algorithm such that the output of $\mathcal{G}(1^\lambda, n)$ consists of description of a sequence of groups $G_1, \cdots, G_n$ of the same prime order q, where q is exponential in λ, the corresponding generators $g_1, \cdots, g_n$, and a multilinear map e. The output of $\mathcal{G}(1^\lambda, n)$ is denoted by $pp = (\{(G_i, g_i), \}_{i=1}^n, e, q)$.

We start from a simple interactive proof for Exact-Cover problem. The construction of 3-round interactive protocol is in figure 2.

The completeness of the protocol is easy to see, and the soundness come from the Decision Multilinear No-Exact-Cover Assumption. The protocol is only honest verifier zero knowledge because any simulator is unable to generates $A = g_n^{\prod_{i=1}^{n} a_i}$ correctly when the malicious verifier randomly picks $A_j \in G_{|T_j|}$, being related to the prover's first message, even if it is an augmented black-box simulator.

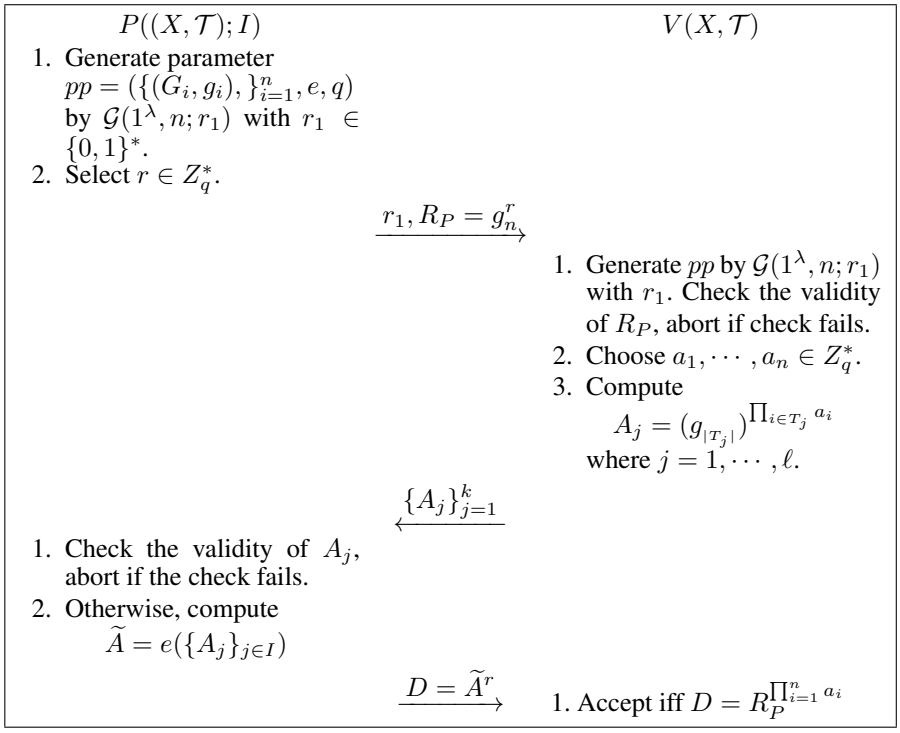

Figure 2: Interactive profs for Exact Cover Problem

In order to get zero knowledge property, we can require the verifier to prove $\{A_j\}_{j=1}^{l}$ is generated correctly after he sends $\{A_j\}_{j=1}^{l}$ to the prover. The prover sends the last message only if the verifier's proof is accepted. Specifically, for the instance $x = (X, \mathcal{T})$ and any given $pp = ((G_1, g_1), \cdots, (G_n, g_n), e, q)$, define set

$$L_{x-pp} = \Big\{\{A_j, T_j\}_{j=1}^{l} : T_j \in \mathcal{T};\ \exists a_1, \cdots, a_n \in Z_q^*, A_j = (g_{|T_j|})^{\prod_{i\in T_j} a_i}, j \in [\ell]\Big\}.$$

To prove $\{A_j\}_{j=1}^{l}$ is generated honestly is to prove that $\{(A_j, T_j)\}_{j=1}^{l} \in L_{x-pp}$. To this end, reduce L_{x-pp} to Exact-Cover Problem Π_{EC}. Assume that $y = (X', \mathcal{T}') \in \Pi_{EC}$ is an instance of Exact-Cover Problem obtained from $\{(A_j, T_j)\}_{j=1}^{l}$, where $|X'| = m$ and $\mathcal{T}' = \{T_1', \cdots, T_k'\}$. It is known that $\{(A_j, T_j)\}_{j=1}^{l} \in L_{x-pp}$ iff $y \in EC_{yes}$. Therefore, to prove $\{A_j\}_{j=1}^{l}$ is honestly generated, the verifier and the prover can execute the above 3-round interactive proof.

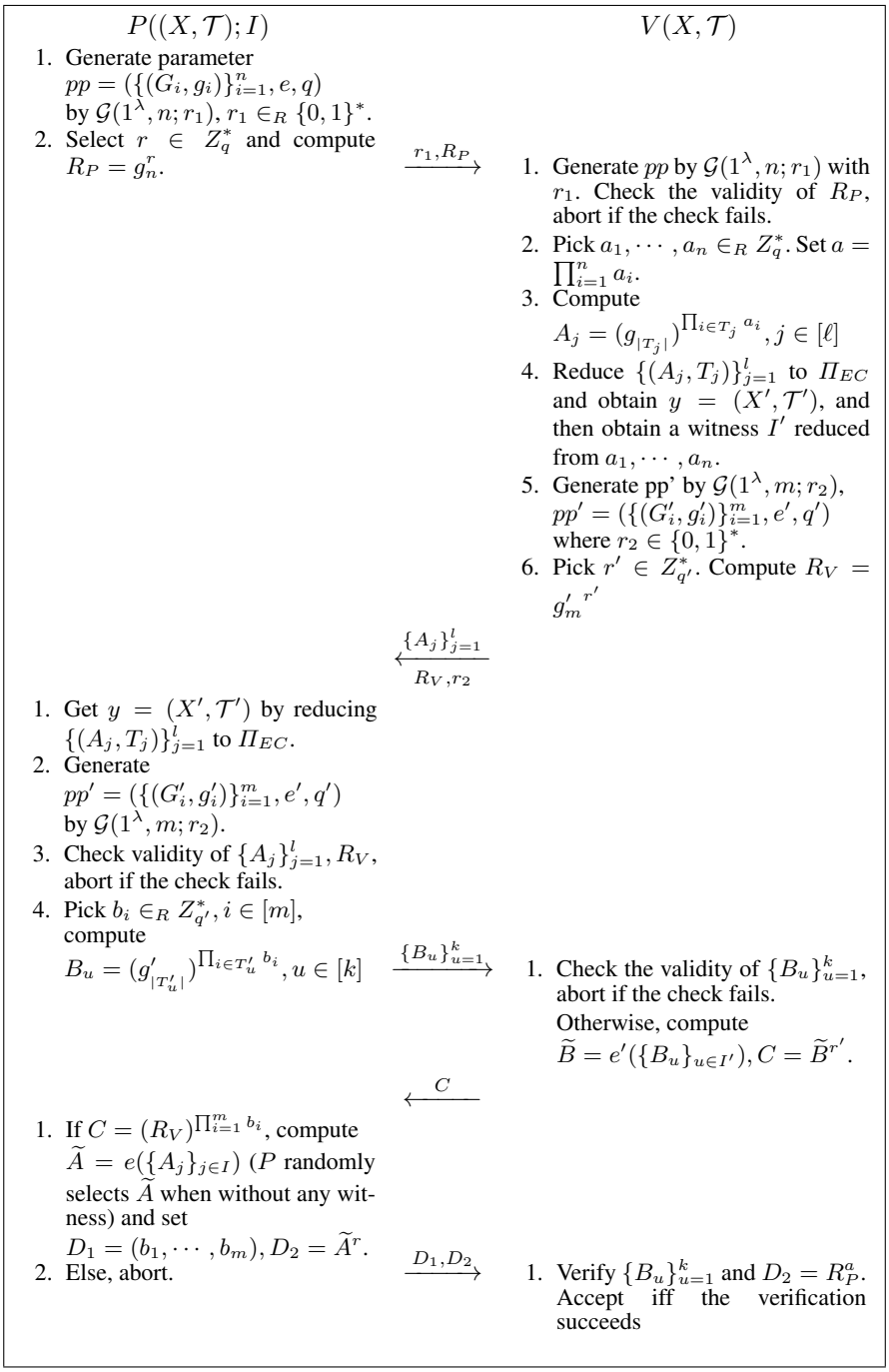

Figure 3: Statistical augmented black-box ZKA for Exact Cover Problem

Our zero-knowledge argument $\langle P, V\rangle$ consists of three stages. In the first stage, the prover P and the verifier V execute the first two round of the 3-round interactive proof protocol, and obtain $y \in \Pi_{EC}$ by reducing L_{x-pp} to Π_{EC}; In the second stage, P and V execute the 3-round interactive proof to let V prove to P that $y \in EC_{yes}$. In the last stage, P proceeds the last round of the 3-round interactive proof, that is, P sends D to V, when the proof in the second stage is accepted. The detailed construction is depicted in Figure 3.

Theorem 1. *Assume that Decision Multilinear No-Exact-Cover Assumption holds. The construction in Figure 3 is an interactive argument for $x \in EC_Y$.*

Proof. **Completeness**. It is obvious to see that, if P and V execute the protocol honestly, V can computes $C = e'(\{B_u\}_{u\in I'})^{r'}$ and P can computes $D_1 = R_P^a = e(\{A_j\}_{j\in I})^r$ when $x \in EC_{yes}$. And so, if $x \in EC_{yes}$, P can always convince V to accept.

Soundness. Suppose $x = (X; \mathcal{T}) \in EC_N$. Obviously, we only need to prove that the probability that the malicious prover P^* outputs the correct $D = \widetilde{A}^r$ is negligible. And notice that if V accepts, it means $\{B_u\}_u$ are honestly generated, and so $\widetilde{B}^{r'} = R_V^{b_1\cdots b_m}$ can be computed from R_V and $D_1 = (b_1, \cdots, b_m), D_2 = \widetilde{A}^r$. Therefore, if P^* can convince V, P^* must be able to compute $A = g_n^{\prod_{i=1}^n a_i}$ from $\{A_j\}_{j=1}^l$. By the assumption, it is impossible except for a negligible probability. Thus the soundness follows. ■

Theorem 2. *Assume that Decision Multilinear No-Exact-Cover Assumption holds. The construction in Figure 3 is statistical augmented black-box zero knowledge.*

For any V^*, the simulation is completed by a PPT simulator with oracles $\mathcal{O}_{\Pi}, \mathcal{O}_{\widehat{V}}$, where $\mathcal{O}_{\Pi}$ returns 1 if $x \in \Pi_Y$ and 0 otherwise, and $\mathcal{O}_{\widehat{V}}$ is the next message function of the extended verifier $\widehat{V}$ (see the details in subsections 2.3 and 2.4).

First, $S^{\mathcal{O}_{\Pi},\mathcal{O}_{\hat{V}}}$ does the same as P to interact with V^* in the first stage and the second stage. Next, if V^*'s proof in the stage 2 is correct, the simulator computes $A = g_n^a$ from the records of V^*'s private state, where $a = \prod_{i=1}^n a_i$. If it succeeds, set $D_2 = A^r$. Else, randomly select D_2.

Proof. For any verifier V^*, the augmented black-box simulator $S^{\mathcal{O}_{EC},\mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)}$ proceeds as follows:

1. Uniformly select $r_{V^*} \in \{0,1\}^{poly(n)}$ for $\mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)$.
2. Make a query to $\mathcal{O}_{EC}$ with x and obtain its return δ_x.
3. Execute the first two round of the protocol.
   - Randomly select $r_1 \in \{0,1\}^*$ and generate $pp = ((G_1, g_1), \cdots, (G_n, g_n), e, q)$ by $\mathcal{G}(1^\lambda, n; r_1)$. Select $r \in Z_q^*$.
   - Make a query to $\mathcal{O}_{\widehat{V^*}}(x, r_{V^*}; \cdot)$ with $(R_P = g_n^r, r_1)$. If it aborts, stop and output (x, r_{V^*}, R_P, r_1). Else, receive $(\{A_j\}_{j=1}^l, r_2, R_V)$. In addition, simulator obtains V^*'s current private state $s_1 = state_{V^*}^{(1)}$, which is used to produce $(\{A_j\}_{j=1}^l, r_2, R_V)$.

4. Get $y = (X', \mathcal{T}')$ from $\{A_j, T_j\}_{j=1}^{l}$, where $X' = \{v'_1, \cdots, v'_m\}$ and $\mathcal{T}' = \{T_i\}_{i=1}^{k}$, by reducing L_{x-pp} to Exact-Cover problem.
5. Generate $pp' = ((G'_1, g'_1), \cdots, (G'_n, g'_n), e', q')$ by $\mathcal{G}(1^\lambda, m; r_2)$
6. If $\{A_j\}_j$ or R_V is not valid, stop and output (x, r_{V^*}, r_1, R_P).
7. Otherwise,
   - Select $b_1, \cdots, b_m \in_R Z_{q'}^*$, and set $D_1 = (b_1, \cdots, b_m)$. Compute $B_u = (g'_{|T'_u|})^{\prod_{i \in T'_u} b_i}$, $u = 1, \cdots, k$.
   - Make a query to $\mathcal{O}_{\widehat{V^*}}(x, r_{V^*}; \cdot)$ with $(\{B_u\}_u)$. If V^* aborts, stop and output $(x, r_{V^*}, r_1, R_P, \{B_u\}_u)$. Else, obtain V^*'s return C and current private state $s_2 = state_{V^*}^{(2)}$.
8. If $C \neq (R_V)^{\prod_{i=1}^{n} b_i}$, stop and output $(x, r_{V^*}, r_1, R_P, \{B_u\}_u)$.
9. Else, compute $\widetilde{A} = g_n^{\prod_{i=1}^{n} a_i}$ from V^*'s private state s_1 and s_2, which should contain $a_1, \cdots, a_n \in Z_q^*$ from which $\{A_j\}_j$ is computed if the verifier is honest. If it succeeds, set $D_2 = \widetilde{A}^r$ when $\delta_x = 1$, or randomly select $D_2 \in G_n$ when $\delta_x = 0$; Otherwise, randomly select $D_2 \in G_n$ when $\delta_x = 0$, or fail and stop when $\delta_x = 1$
10. Output $(x, r_{V^*}, R_P, r_1, \{B_u\}_u, (D_1, D_2))$.

Next, we will prove that the output of the simulator $\{S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}(x)\}$ is statistically indistinguishable from the view $\{View_{V^*}^{P(x,I)}(x)\}$ of the verifier playing the protocol with an honest prover P with witness for the proven statement x.

Lemma 1. *The probability that the simulator fails without output is negligible.*

Proof. Denoted by $BadState$ the event that the simulator $S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}$ cannot compute $\widetilde{A} = g_n^{\prod_{i=1}^{n} a_i}$ from V^*'s private state s_1 and s_2. Notice the event "the simulator fails and stops" takes place iff the event $x \in EC_{yes} \wedge BadState \wedge (C = R_V^b)$ takes place. It is easy to see that

$$\Pr[(x \in EC_{yes}) \wedge BadState \wedge (C = R_V^b)] \leq \Pr[BadState \wedge (C = R_V^b)]$$

Furthermore, if $y \in EC_{yes}$ but the verifier V^* does not have a witness I' for $y \in EC_{yes}$ ($BadState$ takes place), then the probability that V^* can prove $y \in EC_{yes}$ is negligible. Therefore,

$$\Pr[(y \in EC_{yes}) \wedge (C = R_V^b) \wedge BadState] \leq neg(\cdot)$$

and we have

$$\begin{aligned}
&\Pr[(C = R_V^b) \wedge BadState] \\
&= \Pr[(y \in EC_{no}) \wedge (C = R_V^b) \wedge BadState] \\
&\qquad + \Pr[(y \in EC_{yes}) \wedge (C = R_V^b) \wedge BadState] \\
&\leq \Pr[(y \in EC_{no}) \wedge (C = R_V^b)] + neg(\cdot)
\end{aligned}$$

By Decision Multilinear No-Exact-Cover Assumption, $\Pr[(y \in EC_{no}) \wedge (C = R_V^b)]$ is negligible. All above, the probability that the simulator fails without output is negligible. $x \in EC_{yes}$ means the prover P honestly computes $\widetilde{A}$ by $\widetilde{A} = e(\{A_j\}_{j \in I})$

Since the simulator acts as an honest prover before computing $\mathcal{A}$, it is easy to see that the following two lemmas hold.

Lemma 2. *For any randomly selected* (r_{V^*}, r_1, R_P)*, we have*

$$\Pr\left[View_{V^*}^{P(x,I)}(x) = (x, r_{V^*}, r_1, R_P)\right] = \Pr\left[S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)}(x) = (x, r_{V^*}, r_1, R_P)\right]$$

Lemma 3. *For any randomly selected* $(r_{V^*}, pp, R_P, (b_1, \ldots, b_m))$*, we have*

$$\begin{aligned}&\Pr\left[View_{V^*}^{P(x,I)}(x) = (x, r_{V^*}, r_, R_P, \{B_u\}_u)\right]\\ =\ &\Pr\left[S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)}(x) = (x, r_{V^*}, r_1, R_P, \{B_u\}_u)\right]\end{aligned}$$

The only difference between the simulator and the prover is that, instead of computing $\mathcal{A}$ from $\{A_j\}_j$, the simulator need to compute $\mathcal{A}$ from the received state of the verifier. So, the simulation will fail when the verifier's malicious act in generating $\{A_j\}_j$. Concretely, the failure arises only under $(C = R_V^b) \wedge BadState$ ($BadState$ is defined in proof of Lemma 1), which takes place with a negligible .

Lemma 4. *For any* $(r_1, R_P, \{B_u\}_u, D = (D_1, D_2))$*, it holds that*

$$\begin{aligned}&\Pr\left[View_{V^*}^{P(x,I)}(x) = (x, r_{V^*}, r_1, R_P, \{B_u\}_u, D)\right]\\ &\quad = \Pr\left[S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)}(x) = (x, r_{v^*}, r_1, R_P, \{B_u\}_u, D)\right] \pm neg(\cdot)\end{aligned}$$

Proof. The probability that the simulator outputs $(x, r_{V^*}, r_1, R_P, \{B_u\}_u, D)$, by the Lemma 1–3, is almost equal to the probability that the view of V^* is in the form $(x, r_{V^*}, r_1, R_P, \{B_u\}_u, D)$ except for a negligible probability.

Because the simulator is same as an honest prover in step3-step7, the only difference between the output of the simulator and that of V^* is D_2. For convenience, the simulator's output is denoted by $(x, r_{V^*}, r_1, R_P, \{B_u\}_u, D' = (D_1, D_2'))$.

In the view of V^*, D is computed by $D_2 = \mathcal{A}^r$ or selected randomly from G_n, but the simulator generates D_2' from V^*'s state or selects randomly from G_n. It is known that the simulator randomly select $D_2' \in G_n$ iff $\delta_x = 0$, that is $x \in EC_{no}$. Since P select D_2 randomly from G_n when $x \in EC_{no}$, we obtain that $\Pr[D_2' \in_R G_n] = \Pr[D_2 \in_R G_n]$.

Next, we prove that D_2' has the same distribution as D_2 when the simulator computes D_2' with V^*'s private state. The simulator computes D_2 with V^*'s private state implies $(x \in EC_{yes}) \wedge (C = R_V^b)$, and then means $D_2 = e\,(\{A_j\}_{j\in I})$. So, let

$$\begin{aligned}\Gamma_{good} &= (x \in EC_{yes}) \wedge (C = R_V^b) \wedge (\neg BadState),\\ \Gamma_{bad} &= (x \in EC_{yes}) \wedge (C = R_V^b) \wedge (BadState),\end{aligned}$$

we obtain

$$\begin{aligned}&\Pr[D_2' = D_2 | (x \in EC_{yes}) \wedge (C = R_V^b)]\\ =\ &\Pr[\neg BadState \wedge (D_2' = D_2) | (x \in EC_{yes}) \wedge (C = R_V^b)]\\ &+ \Pr[BadState \wedge (D_2' = D_2) | (x \in EC_{yes}) \wedge (C = R_V^b)]\\ =\ &\Pr[\neg BadState | (x \in EC_{yes}) \wedge (C = R_V^b)] \Pr[(D_2' = D_2) | \Gamma_{good}]\\ &+ \Pr[BadState | (x \in EC_{yes}) \wedge (C = R_V^b)] \Pr[(D_2' = D_2) | \Gamma_{bad}].\end{aligned}$$

Furthermore, by the following fact:

$\Pr[D_2' = D_2 | \Gamma_{good}] = \Pr[D_2' = D_2 | (x \in EC_{yes}) \wedge (C = R_V^b) \wedge (\neg BadState)] = 1$

$\Pr[D_2' = D_2 | \Gamma_{bad}] = \Pr[D_2' = D_2 | (x \in EC_{yes}) \wedge (C = R_V^b) \wedge (BadState)] = 0$

we have

$$\begin{aligned}
&\Pr[D_2' = D_2 | (x \in EC_{yes}) \wedge (C = R_V^b)] \\
&= \Pr[\neg BadState | (x \in EC_{yes}) \wedge (C = R_V^b)] \\
&= 1 - \Pr[BadState | (x \in EC_{yes}) \wedge (C = R_V^b)] = 1 - neg(\cdot).
\end{aligned}$$

Combining with $\Pr[D_2' \in_R G_n | x \in EC_{no}] = \Pr[D_2 \in_R G_n | x \in EC_{no}]$, we obtain

$$\Pr[D_2' = D_2 | (C = R_V^b)] = 1 - neg(\cdot).$$

This Lemma is proved.

These four lemmas complete the proof of Theorem 2. ■

4 Leakage-resilient zero-knowledge argument

We recall the notions of leakage-resilient zero knowledge in [18, 32, 25], show the description of the augmented black-box simulator with access to leakage oracle, and present the definition of leakage-resilient zero knowledge according to our another work on augmented black-box zero-knowledge. We show a construction of statistical leakage-resilient zero-knowledge argument according to this definition.

Recently, Garg et al. [18] introduced leakage-resilient zero-knowledge (LRZK) such that in the setting where adversarial verifiers can obtain arbitrary leakage on the internal state (including the witness and the random coins) of the honest prover throughout the execution of the protocol. LRZK guarantees the zero-knowledge property in stronger adversarial models where an malicious verifier has the ability to obtain leakage of the secret states of the honest prover by launching a side-channel attack. In this model, a malicious verifier makes a series of leakage queries throughout the execution of the protocol, and whenever the prover sends message, he sends answers to the leakage queries as well. More concretely, let $\langle P, V \rangle$ be an interactive proof for a promise problem Π. In the prover's round of the interaction, P sends a message to V, and then updates its current private state $state$ (at the beginning of the interaction, $state$ is initialized to be the private auxiliary input w) by setting $state = state||r$ at that time, where r is a random coin used by P in the current round. And then P responds with $f(state)$ to the leakage query f launched by V.

LRZK requires that for any cheating verifier V^* that can obtain l bits of leakage information about the prover's state via a series of leakage queries $f_1, f_2, ...$, there exists a simulator S with the leakage oracle L_w^n such that the simulator can output an indistinguishable view of the malicious verifier and simulate the leakage as well. The leakage oracle L_w^n is parametrized by the witness w of the proven statement and the security parameter n. On inputing a computable leakage function $f'(\cdot)$, $L_w^n(\cdot)$ returns

$f'(w)$ to V, where $f'(\cdot)$ is usually prepared by S from V^*'s leakage query $f'(\cdot,\cdot)$ about the witness and randomness.

In [18], Garg et al. proposed λ-leakage-resilient zero knowledge, which requires that for any V^* there exists a simulator S with leakage oracle $L_w^{n,\lambda}(\cdot)$ such that the output of S and $View_{V^*}$ are indistinguishable and $L_w^{n,\lambda}(\cdot)$ sends at most $\ell'_S = \lambda\ell_{V^*}$ bits leakage, where ℓ_V is the total number of leaking bits received by V. Clearly, $\lambda \geq 1$. And later the work in [32, 25] reduces λ to 1. Next, we recall leakage-resilient zero knowledge according to the definition of augmented black-box zero knowledge.

As assumed previously, let $\langle P, V\rangle$ be a $2k(n)$-round interactive proof system for a promise problem Π. For $i = 1, \cdots, k$, the verifier sends α_i to the prover and receives β_i from the prover. However, the next message functions here in the leakage setting is different. The verifier's next message function takes the leakages received as a part of input, while the prover's next message function takes the current leakage query as a part of input. That is, the prover P computes β_i from the current private state $state_P^{(i)}$ which contains w (the witness for $x \in \Pi_Y$) and his random coins, namely, $\beta_i = P(state_P^{(i)})$, and answers a leakage query f_i on $state_P^{(i)}$ issued by V. And the next message function of V is as follows:

$$(\alpha_i, f_i(\cdot)) = Next_V\left(x, r_V, \overline{\beta}_{i-1}, F_{i-1}\right), i = 1, \cdots, k$$

where $\overline{\beta}_{i-1} = (\beta_1, \cdots, \beta_{i-1})$, $F_{i-1} = (f_1(state_P^{(1)}), \cdots, f_{i-1}(state_P^{(i-1)})$, and F_{i-1} is all the leakage received by V. After sending α_i to P, V receives β_i and $f_i(state_P^{(i)})$ (if V does not issue any leakage query f_i, set $f_i(state_P^{(i)})$ to be the empty string λ). And for any verifier V described above, the extended verifier $\widehat{V}$ is defined as in subsection 2.5, and so $\widehat{V}$'s next message function is as follows:

$$\left(\alpha_i, f_i(\cdot), state_V^{(i)}\right) = Next_{\widehat{V}}\left(x, r_V, \overline{\beta}_{i-1}, F_{i-1})\right), i = 1, \cdots, k.$$

$\mathcal{O}_{\widehat{V}}(x, r; \cdot, \cdot)$ is also defined as in subsection 2.5. When queried with $(i, \overline{\beta}_{i-1}, F_{i-1})$, $\mathcal{O}_{\widehat{V}}(x, r; \cdot)$ returns $(\alpha_i, f_i, state_V^{(i)})$.

Leakage-resilient zero-knowledge proofs $\langle P, V\rangle(x)$ for a promise problem Π require that for any malicious verifier launching leakage attack, the verifier can learn nothing beyond $x \in \Pi_Y$ and the leakage. To formulate this, for any verifier V^*, the construction of the leakage-resilient simulator with the oracles $\mathcal{O}_\Pi$, $\mathcal{O}_{\widehat{V^*}}(x, \cdot; \cdot)$ and the leakage oracle L_w^n, written as $S^{\mathcal{O}_\Pi, \mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot), L_w^n}$, is depicted in Figure 4.

Let $LView_{V^(z)}^P(x)$ consist of the view of V with any auxiliary input z and the received leakage.

Definition 6. ***Leakage-resilient Augmented Black-box Zero-knowledge Proof (Argument).*** *Let $\langle P, V\rangle$ be an interactive proof system for some language L. $\langle P, V\rangle$ is called leakage-resilient augmented black-box zero-knowledge proof system if for any PPT verifier V^* with auxiliary z, the corresponding extended verifier is denoted by $\widehat{V^*}$, there exists an augmented black-box simulator S with oracles $\mathcal{O}_L$, $\mathcal{O}_{\widehat{V^*}}$ and L_w^n, such that $LView_{V^*(z)}^P(x)$ and $S^{\mathcal{O}_L, \mathcal{O}_{\widehat{V^*}}(x;\cdot), L_w^n}(x, z)$ are computationally indistinguishable.*

If $S^{\mathcal{O}_L,\mathcal{O}_{\widehat{V^*}}(x;\cdot),L_w^n}(x)$ is statistically closed or identical to $LView^P_{V^*(z)}(x)$ when it does not fail, $\langle P,V\rangle(x)$ is called statistical or perfect leakage-resilient augmented black-box zero-knowledge proof (argument).

If the soundness security holds for an all powerful prover, $\langle P,V\rangle$ is called an (statistical) leakage-resilient augmented black-box zero-knowledge argument.

1. Uniformly select r_V for $\mathcal{O}_{\widehat{V}}(x,\cdot;\cdot)$,and set $i=1$, $\beta'_0=\lambda$, $f_0=\lambda$.
2. Make a query to $\mathcal{O}_{\Pi}$ with x. It returns b.
3. Invoke $\mathcal{O}_{\widehat{V}}(x,r_V;\cdot)$ with $\big(i,\overline{\beta'}_{i-1},F_{i-1}\big)$, where $\overline{\beta'}_{i-1}=(\beta'_0,\cdots,\beta'_{i-1})$, $F_{i-1}=(f_1(state_P^{(1)}),\cdots,f_{i-1}(state_P^{(i-1)})$, and then receive $\Big(\alpha_i,f_i,state_V^{(i)}\Big)$.
4. Verify α_i as an honest prover. If the verification fails or $\alpha_i=\perp$, output $\big(x,r_V,\beta'_1,\cdots,\beta'_{i-1},F_{i-1}\big)$ and stop.
5. Else, Verify that V^* computes α_i correctly from $state_V^{(i)}$.
   - If the verification succeeds and $b=1$, prepare $state_P^{(i)}$ corresponding to $state_V^{(i)}$, $\overline{\beta'}_{i-1}$, and then compute β'_i such that it is acceptable by $\widehat{V}$. If no such β'_i can be obtained, fail and stop.
   - Else, randomly select random coins of $state_P^{(i)}$ to compute β'_i.
   - Construct $f'_i(\cdot)$ such that $f'_i(w)$ is identical to $f_i(state_P^{(i)})$ by letting f'_i be residual function of f_i with the random coins in $state_P^{(i)}$ hard-wired in. Then, query L_w^n with $f'_i(\cdot)$ and receive the response $f'_i(w)$. Assuming that L_w^n returns $f'_i(w)$ to V^* simultaneously.
6. $i\leftarrow i+1$. Return to step 3 if $i\le k$.
7. Output $(x,r_V,\beta'_1,\cdots,\beta'_k,F_k)$.

Figure 4: Leakage-resilient simulator $\boldsymbol{S^{\mathcal{O}_{\Pi},\mathcal{O}_{\widehat{V}}(x,\cdot;\cdot),L_w^n}}$

Theorem 3. *The construction presented in Figure 2 is a statistical leakage-resilient augmented black-box zero-knowledge argument, if the conditions in Theorem 2 hold.*

Proof. We need to construct leakage-resilient augmented black-box simulator. We can see that it is easy to construct $S^{\mathcal{O}_{EC},\mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot),L_I^n}$ by modifying $S^{\mathcal{O}_{EC},\mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)}$ defined in the proof of Theorem 2. For any verifier V^*, the leakage-resilient augmented black-box simulator $S^{\mathcal{O}_{EC},\mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot),L_I^n}$ proceeds as follows:

1. Uniformly select $r_{V^*}\in\{0,1\}^{poly(n)}$ for $\mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)$.
2. Make a query to $\mathcal{O}_{EC}$ with x and obtain its return δ_x.
3. Execute the first two round of the protocol.
   - Randomly select $r_1\in\{0,1\}^*$ and generate $pp=((G_1,g_1),\cdots,(G_n,g_n),e,q)$ by $\mathcal{G}(1^\lambda,n;r_1)$. Select $r\in Z_q^*$.
   - Make a query to $\mathcal{O}_{\widehat{V^*}}(x,r_{V^*};\cdot)$ with $(R_P=g_n^r,r_1)$. If it aborts, stop and output (x,r_{V^*},R_P,r_1). Else, and receive $(\{A_j\}_{j=1}^l,r_2,R_V)$ and a leakage query $f_1(\cdot,\cdot)$. In addition to, simulator obtains V^*'s current private state $s_1=state_{V^*}^{(1)}$.

- Generate leakage query function by setting $f_1'(\cdot) = f_1(\cdot, r)$. Query leakage oracle L_I^n with f_1' and obtain $f_1'(I)$.

4. Get $y = (X', \mathcal{T}')$ from $\{A_j, T_j\}_{j=1}^l$, where $X' = \{v_1', \cdots, v_m'\}$ and $\mathcal{T}' = \{T_i\}_{i=1}^k$, by reducing L_{x-pp} to Exact-Cover problem.
5. Generate $pp' = ((G_1', g_1'), \cdots, (G_n', g_n'), e', q')$ by $\mathcal{G}(1^\lambda, m; r_2)$
6. If $\{A_j\}_j$ or R_V is not valid, stop and output $(x, r_V, r_1, R_P; f_1(I))$.
7. Otherwise,
   - Select $b_1, \cdots, b_m \in_R Z_{q'}^*$ and set $D_1 = (b_1, \cdots, b_m)$.
     Compute $B_u = (g'_{|T_u'|})^{\prod_{i \in T_u'} b_i}$, $u = 1, \cdots, k$.
   - Make a query to $\mathcal{O}_{\widehat{V^*}}(x, r_{V^*}; \cdot)$ with $\{B_u\}_u$. If V^* aborts, stop and output $(x, r_{V^*}, r_1, R_P, \{B_u\}_u, f_1(I))$. Else, obtain V^*'s return C, a leakage query function $f_2(\cdot, \cdot)$ and current private state $s_2 = state_{V^*}^{(2)}$.
   - Generate leakage query function by setting $f_2'(\cdot) = f_2(\cdot, r, (b_1, \cdots, b_m))$. Query leakage oracle L_I^n with f_2' and obtain $f_2'(I)$.
8. If $C \neq R_V^{\prod_{i=1}^m b_i}$, stop and output $(x, r_{V^*}, r_1, R_P, \{B_u\}_u, (f_1'(I), f_2'(I)))$.
9. Else:
   - Compute $\widetilde{A} = g_n^{\prod_{i=1}^n a_i}$ from V^*'s private state s_1 and s_2. If it succeeds, set $D_2 = \widetilde{A}^r$ when $\delta_x = 1$, or randomly select $D_2 \in G_n$ when $\delta_x = 0$; Otherwise, randomly select $D_2 \in G_n$ when $\delta_x = 0$, or fail and stop when $\delta_x = 1$.
   - Make a query to $\mathcal{O}_{\widehat{V^*}}(x, r_{V^*}; \cdot)$ with $D = (D_1, D_2)$ and receive a leakage query function $f_3(\cdot, \cdot)$.
   - Generate leakage query function by setting $f_3'(\cdot) = f_3(\cdot, r)$. Query leakage oracle L_I^n with f_3' obtain $f_3'(I)$.
10. Output $(x, r_{V^*}, R_P, r_1, \{B_u\}_u, (D_1, D_2), (f_1'(I), f_2'(I), f_3'(I)))$.

Notice that P's response is independent with the holding witness I (although P uses I to compute his last message D_2). So, the simulator's leakages $(f_1'(I), f_2'(I), f_3'(I))$ are identical to the real leakages. By the proof of Theorem 2, the simulator's output $\{S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot), L_I^n}(x)\}$ is statistically indistinguishable from $\{LView_{V^*}^{P(x,I)}(x)\}$. That is, the theorem is correct.

5 2-round computational zero-knowledge arguments for Exact Cover problem

The goal in this section is to construct a 2-round augmented computational black-box zero-knowledge argument for Exact Cover problem $EC = (EC_Y, EC_N)$ under the Decision Multilinear No-Exact-Cover Assumption and the assumption of the existence of hash functions.

Let $\{H_K\}_K$ be a family of hash functions, where $K \leftarrow Gen(1^\lambda)$ and $H_K : G_n \to \{0,1\}^*$. For any instance $x = (X; \mathcal{T}) \in EC$, where $|X| = n$, $|\mathcal{T}| = l$, and $pp = ((G_1, g_1), \cdots, (G_n, g_n), e, q) \leftarrow \mathcal{G}(1^\lambda, 1^n; r)$, define the language

$$L_{x-pp} = \left\{ \begin{array}{c} (\{A_j, T_j\}_{j=1}^l, K, t) : 1) \exists a_1, \cdots, a_n \in Z_q^*, A_j = (g_{|T_j|})^{\prod_{i \in T_j} a_i}, j \in [\ell] \\ 2) t = H_K(g_n^{\prod_i a_i}) \end{array} \right\}.$$

Let G and G' be two groups. For $A \in G$ and $B \in G'$, $A \oplus B$ denotes the exclusive OR operation on the its binary coded values. The details of the 2-round augmented computational black-box zero-knowledge argument for Exact Cover problem $EC = (EC_Y, EC_N)$ is depicted in Figure 5.

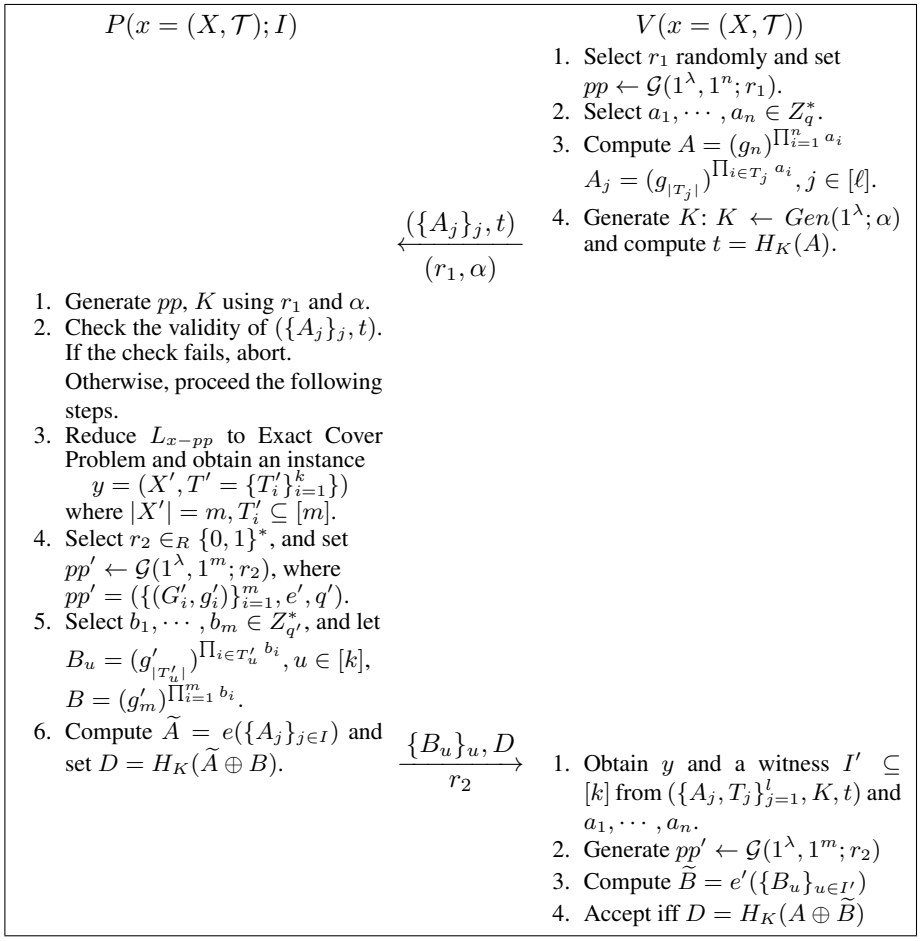

Figure 5: 2-round computational ZKA for Exact Cover Problem

Theorem 4. *Assume $\{H_K\}_K$ is a family of collision-resist hash functions and that the Decision Multilinear No-Exact-Cover Assumption holds. The construction in Figure 4 is an interactive argument for $x \in EC_Y$.*

Proof. **Completeness**. It is obvious to see that if $x \in EC_Y$ and V executes the protocol honestly, P can compute the committed value of $\{A_j\}_j$ by $\widetilde{A} = e(\{A_j\}_{j \in I}) = g_n^{\prod_{i \in [n]} a_i}$ using the witness I for $x \in EC_Y$, and V can compute the committed value of $\{B_u\}_u$ by $\widetilde{B} = e'(\{B_u\}_{u \in I'}) = g_m'^{\prod_{i \in [m]} b_i}$ using the witness I' for $y \in EC_Y$, hence then $D = H_K(\widetilde{A} \oplus B) = H_K(A \oplus \widetilde{B})$. Thus, the completeness follows.

Soundness. If $x = (X; \mathcal{T}) \in EC_N$, the probability that the malicious prover P^* outputs the correct $\{B_u\}_u, D = H_K(\widetilde{A} \oplus B)$ is negligible. In fact, by the Decision Multilinear No-Exact-Cover Assumption, P^* can get $\widetilde{A} = g_n^a$ only from the message $(\{A_j\}_j, t, r_1, \alpha)$ except for a negligible probability.

On the contrary, assume that there is a PPT adversary $\mathcal{A}$ that can obtain $\widetilde{A} = g_n^a$ from $(\{A_j\}_j, H_K(A), pp, K)$ with non-negligible probability, then we can construct a PPT distinguisher $\mathcal{D}$ that can distinguish $(\{A_j\}_j, g_n^a)$ and $(\{A_j\}_j, g_n^{\widetilde{r}})$, where $\widetilde{r} \in_R Z_q^*$, with non-negligible probability. $\mathcal{D}$ works as follows:

Given $(\{A_j\}_j, A^* \in G_n, pp)$, $\mathcal{D}$ generates $K \leftarrow Gen(1^\lambda)$, and invokes $\mathcal{A}$ with $(\{A_j\}_j, H_K(A^*), pp, K)$, if $\mathcal{A}$ outputs A^*, $\mathcal{D}$ outputs 1, and outputs 0 otherwise.

Since $x = (X; \mathcal{T}) \in EC_N$, if $A^* = g_n^{\widetilde{r}}$, $H_K(A^*)$ is independent of $\{A_j\}_j$, the property of the hash functions guarantees that no PPT algorithm can obtain A^* from $H_K(A^*)$ except for a negligible probability, thus $\mathcal{D}$ can distinguish $(\{A_j\}_j, g_n^a)$ and $(\{A_j\}_j, g_n^{\widetilde{r}})$ with non-negligible probability. It contradicts the Decision Multilinear No-Exact-Cover Assumption. Therefore, there is no PPT algorithm can obtain $\widetilde{A} = g_n^a$ from $(\{A_j\}_j, t, r_1, \alpha)$ except for a negligible probability.

In a word, P^* can produce correct $D = H_K(\widetilde{A} \oplus B)$ with negligible probability and make honest V accept his proof with negligible probability. Thus the soundness follows.

This completes the proof. ■

Theorem 5. *Under the same condition as Theorem 4, the construction in Figure 5 is augmented computational black-box zero-knowledge.*

Proof. For any verifier V^*, the augmented black-box simulator $S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)}$ proceeds as follows:

1. Uniformly select $r_{V^*} \in \{0,1\}^{poly(n)}$ for $\mathcal{O}_{\widehat{V^*}}(x, \cdot; \cdot)$.
2. Make a query to $\mathcal{O}_{EC}$ with x and obtain its return b_x.
3. Invoke $\mathcal{O}_{\widehat{V^*}}(x, r_{V^*}; \cdot)$ and receive the message $\{A_j\}_j, t, (r_1, \alpha)$ and V^*'s current private state $s_1 = state_{V^*}^{(1)}$
4. Generate pp, K using r_1 and α respectively. Check the validity of $(\{A_j\}_j, t)$. If the check fails, abort. Otherwise, proceed the following steps.
5. Obtain y and pp' as an honest prover.
6. Compute all $\{B_u\}_u$ and B as an honest prover.
7. Compute $\widetilde{A}$ from V^*'s current state s_1 such that $t = H_K(\widetilde{A})$. If it fails or $b_x = 0$, select $\widetilde{A} \in_R G_n$. And then compute $D = H_K(\widetilde{A} \oplus B)$.
8. Output $(x, r_{V^*}, \{B_u\}_u, r_2, D)$.

Next, we will prove that in the condition that the verifier V^* plays the protocol with an honest prover P holding witness for the proven statement x, the output of the simulator $\{S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot;\cdot)}(x)\}$ is computationally indistinguishable from the view $\{View_{V^*}^{P(x,I)}(x)\}$ of the verifier V^*.

Honest zero knowledge: If V^* produces $(\{A_j\}_j, t)$ honestly, V^*'s private state s_1 contains $a_1, \cdots, a_n \in Z_q^*$ such that $A_j = (g_{|T_j|})^{\prod_{i \in T_j} a_i}$ and $t = H_K(g_n^{\prod_{i=1}^n a_i})$. So, the simulator can get the same A as the prover will do. Therefore, Iif V^* is honest,for

any randomly selected $(r_{V^*}, r_2, (b_1, \cdots, b_m))$, it holds that

$$\Pr\left[View_{V^*}^{P(x,I)}(x) = (x, r_{V^*}, \{B_u\}_u, r_2, D)\right]$$
$$= \Pr\left[S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}(x) = (x, r_{V^*}, \{B_u\}_u, r_2, D)\right]$$

Zero knowledge: For the malicious verifier V^*, $\{S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}(x)\}$ may differ from $\{View_{V^*}^{P(x,I)}(x)\}$ only when V^* produces $(\{A_j\}_j, t)$ dishonestly. That is, V^*'s private state s_1 does not contain $a_1, \cdots, a_n \in Z_q^*$ such that $A_j = (g_{|T_j|})^{\prod_{i \in T_j} a_i}$ and $t = H_K(g_n^{\prod_{i=1}^n a_i})$. In this condition, the prover P will compute $\widetilde{A} = e(\{A_j\}_{j \in I})$ and set $D = H_K(\widetilde{A} \oplus B)$ while the simulator $S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}$ randomly selects $\widetilde{A} \in_R G_n$ and sets $D = H_K(\widetilde{A} \oplus B)$.

If V^* produces $(\{A_j\}_j, t)$ dishonestly and $y \notin EC_Y$, P randomly selects B and computes $D = H_K(\widetilde{A} \oplus B)$ while the simulator randomly selects $\widetilde{A}$ and B, computes $D = H_K(\widetilde{A} \oplus B)$. In this case, for $\widetilde{A}$ computed by P, $(\{A_j\}_{j=1}^l, \{B_u\}_{u=1}^k, D = H_K(\widetilde{A} \oplus B))$ and $(\{A_j\}_{j=1}^l, \{B_u\}_{u=1}^k, D = H_K(\widetilde{A} \oplus U_{G'_m}))$ are indistinguishable, and meanwhile, $(\{A_j\}_{j=1}^l, \{B_u\}_{u=1}^k, D = H_K(\widetilde{A} \oplus U_{G'_m}))$ and $(\{A_j\}_{j=1}^l, \{B_u\}_{u=1}^k, D = H_K(U_{G_n} \oplus B))$ are indistinguishable. Therefore, $\{S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}(x)\}$ is computationally indistinguishable from the view of V^*. Here, U_{G_n} ($U_{G'_m}$) is uniformly distributed on G_n (G'_m).

Assume V^* produces $(\{A_j\}_j, t)$ dishonestly but $y \in EC_Y$. This means that there exist $a_1, \cdots, a_n \in Z_q^*$ such that $A_j = (g_{|T_j|})^{\prod_{i \in T_j} a_i}$ and $t = H_K(g_n^{\prod_{i \in [n]} a_i})$. If V^* is unable to get $a_1, \cdots, a_n \in Z_q^*$, V^* will be unable to compute correct $\widetilde{B} = B$. As above, we can obtain $\{S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}(x)\}$ is computationally indistinguishable from the view of V^*. If V^* is able to get $a_1, \cdots, a_n \in Z_q^*$, the simulator can also obtain $a_1, \cdots, a_n \in Z_q^*$, then the simulator is the same as P. In the case that the simulator cannot obtain $a_1, \cdots, a_n \in Z_q^*$ but V^* can, the simulator fails. This happens with a negligible probability. On the contrary, if V^* computes $a_1, \cdots, a_n \in Z_q^*$ with noticeable probability, V^* must can determine whether $T_j \in I$ for any $T_j \in \mathcal{T}$ with noticeable probability (for convenience, we consider when there exists only one witness I for the statement $x = (X; \mathcal{T})$): V^* computes $a_1, \cdots, a_n \in Z_q^*$ from $\mathcal{T} - \{T_j\}$, if succeeds then accepts "$T_j \notin I$", otherwise, accepts "$T_j \in I$".

Combining the cases above, $\{S^{\mathcal{O}_{EC}, \mathcal{O}_{\widehat{V^*}}(x,\cdot,\cdot)}(x)\}$ and $\{View_{V^*}^{P(x,I)}(x)\}$ are computationally indistinguishable in the condition that V^* produces $(\{A_j\}_j, t)$ dishonestly. This completes the proof. ■

References

1. M. Abe, S. Fehr. Perfect NIZK with adaptive soundness. In TCC, pages 118-136, 2007.
2. W. Aiello, J. Hstad. Statistical zero-knowledge languages can be recognized in two rounds. Journal of Computer and System Science, 42(3):327-345, 1991.
3. B. Barak. How to go beyond the black-box simulation barrier. In FOCS, pages 106-115, 2001.
4. G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. J. Comput. Syst. Sci., 37(2):156C189, 1988.

5. G. Brassard, C. Crépeau and M. Yung. Constant round perfect zero knowledge computationally convincing protocols. Theoretical Computer Science, Vol. 84, No. 1, 1991.
6. B. Barak, Y. Lindell. Strict polynomial-time in simulation and extractor. In 34th ACM Symposium on the Theory of Computing, 2002:484-493.
7. M. Bellare, A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In CRYPTO, pages 273-289, 2004.
8. N. Bitansky, O. Paneth. Point obfuscation and 3-round zero knowledge. TCC 2012. LNCS, Volume 7194, pages 189-207, 2012.
9. N. Bitansky, O. Paneth. On the impossibility of approximate obfuscation and application to resettable cryptography. In STOC 2013, pages 241-250.
10. N. Bitansky, O. Paneth. On Non-Black-Box Simulation and the Impossibility of Approximate Obfuscation. SIAM J. Comput. 44(5), 1325-1383(2015).
11. Kai-Min Chung, Huijia Lin, R. Pass. On constant round concurrent zero knowledge from falsifiable. http://eprint.iacr.org/2012/563.pdf
12. Kai-Min Chung, R. Pass, K. Seth. Non-black-box simulation from one-way functions and applications to resettable security. STOC, 2013, pp. 231-240.
13. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography (extended abstract). In STOC, pages 542-552, 1991.
14. C. Dwork, M. Naor, O. Reingold, and L. Stockmeyer. Magic functions. In Memoriam: Bernard M. Dwork 1923-1998. Journal of the ACM, 50(6):852-921, 2003.
15. C. Dwork, M. Naor, and A. Sahai. Concurrent zero-knowledge. In STOC, pages 409-418, 1998.
16. L. Fortnow. The complexity of perfect zero-knowledge. In Advance in Computing Research, Volume 5, pages 327-343,1989.
17. S. Garg, C. Gentry, A. Sahai. Witness encryption and its applications. In STOC 2013.
18. S. Garg, A. Jain, A. Sahai. Leakage-resilient zero knowledge. In CRYPTO, pages 297-315, 2011.
19. V. Goyal, R. Moriarty, R. Ostrovsky, A. Sahaiy. Concurrent statistical zero-knowledge arguments for NP from one way functions. http://eprint.iacr.org/2006/400.
20. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. SIAM J. Comput., 18(1):186-208, 1989.
21. O. Goldreich, Y. Oren. Definition And Properties of Zero-Knowledge Proof Systems. J. Cryptology, 7(1): 1-32, 1994.
22. J. Groth. Short pairing-based non-interactive zero-knowledge arguments. In ASI- ACRYPT, pages 321-340, 2010.
23. S. Hada, T. Tanaka. On the existence of 3-round zero-knowledge protocols. In CRYPTO, pages 408-423, 1998.
24. S. Kiyoshima. Statistical concurrent non-malleable zero-knowledge from one-way functions. In CRYPTO, LNCS 9216, pages 85-106, 2015.
25. S. Kiyoshima. Constant-round leakage-resilient zero-knowledge from collision resistance. In: Fischlin, M., Coron, J.-S. (eds.): EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 93-123(2016).
26. Hongda Li, Dengguo Feng. Constant-Round Zero-Knowledge Proofs of Knowledge with Strict Polynomial-time Extractors for NP. Science China Information Sciences, 57(1):012111:1-012111:13
27. Daniele Micciancio, Shien Jin Ong, Amit Sahai, Salil P. Vadhan. Concurrent zero knowledge without complexity assumptions. In TCC, pages 1-20, 2006.
28. Minh-Huyen Nguyen, Shien Jin Ong, and Salil P. Vadhan. Statistical zero- knowledge arguments for np from any one-way function. In FOCS, pages 3-14. IEEE Computer Society, 2006.

29. M. Naor, R. Ostrovsky, R. Venkatasan, M. Yung. Perfect zero knowledge arguments for NP can be based on general complexity assumptions. Advances in cryptology - Crypto 92 Proceedings, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.
30. C. Orlandi, R. Ostrovsky, V. Rao, I. Visconti. Statistical Concurrent Non-Malleable Zero Knowledge. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 167-191. Springer, Heidelberg (2014).
31. R. Ostrovsky, G. Persiano, I. Visconti. Constant-round concurrent NMWI and its relation to NMZK. ECCC Report No. 95, 2006. 5 Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.
32. O. Pandey. Achieving constant round leakage-resilient zero knowledge. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 146-166. Springer, Heidelberg (2014).
33. R. Pass, A. Rosen. New and improved constructions of non-malleable cryptographic protocols. In Proc. 37th STOC, ACM, 2005, pages 533-542.
34. D. Gupta, A. Sahai. On Constant-Round Concurrent Zero-Knowledge from a Knowledge Assumption. http://eprint.iacr.org/2012/572