An Efficiency-Preserving Transformation from Honest-Verifier Statistical Zero-Knowledge to Statistical Zero-Knowledge

Pavel Hubáček[1⋆], Alon Rosen[2⋆⋆], and Margarita Vald[3⋆⋆⋆]

[1] Computer Science Institute, Charles University, Czech Republic
hubacek@iuuk.mff.cuni.cz
[2] IDC Herzliya, Israel
alon.rosen@idc.ac.il
[3] Tel Aviv University, Israel
margarita.vald@cs.tau.ac.il

Abstract. We present an unconditional transformation from any honest-verifier statistical zero-knowledge (HVSZK) protocol to standard SZK that preserves round complexity and efficiency of both the verifier and the prover. This improves over currently known transformations, which either rely on some computational assumptions or introduce significant computational overhead. Our main conceptual contribution is the introduction of instance-dependent SZK proofs for NP, which serve as a building block in our transformation. Instance-dependent SZK for NP can be constructed unconditionally based on instance-dependent commitment schemes of Ong and Vadhan (TCC'08).

As an additional contribution, we give a simple constant-round SZK protocol for Statistical-Difference resembling the textbook HVSZK proof of Sahai and Vadhan (J.ACM'03). This yields a conceptually simple constant-round protocol for all of SZK.

1 Introduction

Zero-knowledge proof systems, introduced by Goldwasser, Micali, and Rackoff [9], give any powerful prover the ability to convince a verifier about validity of a statement without revealing any additional information other than its correctness. This power has been extensively exploited in constructions of various cryptographic protocols. Besides the many applications, great effort was invested

⋆ This work was performed while at the Foundations and Applications of Cryptographic Theory (FACT) center, IDC Herzliya, Israel. Partially supported by the PRIMUS grant PRIMUS/17/SCI/9 and by the Center of Excellence – ITI, project P202/12/G061 of GA ČR.

⋆⋆ Work supported by ISF grant no 1399/17 and by NSF-BSF Cyber Security and Privacy grant no. 2014/632.

⋆⋆⋆ Work supported by ISF grant no 1399/17 and by Google Europe Doctoral Fellowship in Security.

to improve our understanding of the limits of zero-knowledge proof systems with respect to different complexity measures such as round complexity or efficiency of prover and verifier.

Similarly to the requirement of soundness for interactive proof systems, there are many natural relaxations of zero-knowledge. In this work we study *statistical* zero-knowledge (SZK) proofs. In particular, we revisit the problem of immunizing any honest-verifier statistical zero-knowledge (HVSZK) protocol against malicious verifiers, while preserving the efficiency of the original protocol. Such transformation suggests a methodology for constructing zero-knowledge protocols: first construct an efficient proof system for the desired problem where the zero-knowledge property holds against honest verifiers, and then compile it to a full-blown zero-knowledge proof against malicious verifiers while preserving the efficiency.

Bellare, Micali, and Ostrovsky [3] initiated the study of general transformations from honest-verifier zero-knowledge protocols to protocols in which the zero-knowledge property holds against arbitrary verifiers. Their work presented such a transformation under the assumption of intractability of solving the discrete-logarithm problem. Later, Ostrovsky, Venkatesan, and Yung [16] presented a transformation under a weaker assumption of existence of one-way permutations. Okamoto [13] further weakened the assumption to existence one-way functions. However, relying on intractability assumptions prevents the zero-knowledge property to hold against computationally unbounded verifiers which might be a desirable property in some contexts.

Until recently, unconditional transformations of honest-verifier zero-knowledge to zero-knowledge against malicious verifiers were only known via public-coin proof system. Under the restriction to constant-round public-coin protocols [4, 5] gave first such unconditional transformations. The restriction to constant-round was lifted by [7] who gave a transformation achieving general statistical zero-knowledge starting from any *public-coin* honest-verifier statistical zero-knowledge protocol. Combining the transformation of [7] with the private-coin to public-coin transformation of [13, 8] yields a general transformation starting from any honest-verifier protocol. However, it follows from Vadhan [18] that any transformation from honest-verifier zero-knowledge to general cheating verifier that goes through public-coin protocol must result in a significant blow-up in the prover's complexity. Moreover, the private-coin to public-coin transformation of [13, 8] does not preserve the message complexity.

Ong and Vadhan [14] successfully avoided the standard private-coin to public-coin transformation by relying on their novel construction of a relaxed notion of commitments, called instance-dependent commitment. Instance-dependent commitments allow the hiding and binding properties of a commitment scheme not to hold simultaneously but rather to depend on a given instance. Specifically, they obtained a general transformation from honest-verifier statistical zero-knowledge to general statistical zero-knowledge by going via the transformation of honest-verifier statistical zero-knowledge to two-round Arthur-Merlin protocol due to Aiello and Håstad [1]. In the resulting statistical zero-knowledge protocol the verifier sends the first message of Arthur in the AM protocol and the prover then gives a statistical zero-knowledge proof for the NP statement of the form: there exists a message of Merlin that makes Arthur accept. The statistical zero-knowledge proof for this NP statement can be performed in constant number of rounds by instantiating known statistical zero-knowledge protocols for NP using the instance-dependent commitment scheme of Ong and Vadhan [14]. The transformation in [14] was the first to result in a protocol with constant number of rounds. However, the [14] transformation, as well as all of the above unconditional transformations, result in a significant blow-up in the complexity of the prover compared to the original honest-verifier protocol.

2 Our Results

We present a general efficiency-preserving compiler from any honest-verifier statistical zero-knowledge proof to a statistical zero-knowledge proof against malicious verifiers. Our compiler preserves both the round complexity and the prover's complexity of the original honest-verifier protocol. Our transformation yields a very simple constant-round statistical zero-knowledge protocol for every problem in honest-verifier statistical zero-knowledge.

Theorem 1 (honest-verifier SZK to SZK compiler). *For every promise problem $\Pi \in \mathsf{HVSZK}$, there exists a statistical zero-knowledge proof where the prover's complexity and the round complexity match the parameters of the best honest-verifier statistical zero-knowledge proof for Π.*

Applying Theorem 1 on the honest-verifier statistical zero-knowledge protocol of Sahai and Vadhan [17] for the HVSZK-complete problem Statistical-Difference yields the following:

Theorem 2 (Constant-round proof for SZK). *For every promise problem $\Pi \in \mathsf{HVSZK}$, there exists a constant-round statistical zero-knowledge proof.*

Additionally, we show how to achieve Theorem 2 via simple direct construction for Statistical-Difference. This is shown in Section 4.2.

Our transformation follows the classical approach of Goldreich, Micali and Wigderson [6] to immunize protocols against malicious behavior. In the context of zero-knowledge, an honest verifier follows the protocol specification using a uniformly random tape. The standard way to preserve zero-knowledge in the presence of a malicious verifier is to enforce the honest behavior. To this end, we leverage the fact that the protocol specification is a deterministic function of the verifier's view; at each round the verifier's view consists of its random tape and the messages received up to this round. Thus, the verifier can give a zero-knowledge proof for the NP statement attesting that its messages to the prover are indeed computed according to the specifications of the protocol.

Note that the quality of the employed zero-knowledge proof for NP determines the quality of the resulting protocol. Specifically, if we use as a building block a proof for NP that is zero-knowledge against polynomial-time verifiers then the resulting protocol will be a zero-knowledge *argument*. This follows from the fact that the roles of the prover and verifier are reversed in the intermediate proof for NP and our compiler cannot guarantee soundness against unbounded provers unless the simulator for the intermediate proofs can handle unbounded verifiers. To solve this issue, we use a relaxation of statistical zero-knowledge for NP that is sufficient for our compiler to result in a statistical zero-knowledge *proof*.

Instance-dependent commitment schemes [2, 10], in which the properties of the commitment protocol depend on a given instance of a language, proved to be useful in constructions of zero-knowledge protocols by Itoh, Ohta, and Shizuya [10]. Recently, Ong and Vadhan [14] constructed instance-dependent (ID) commitments relative to all of SZK. The ID commitments of Ong and Vadhan are statistically binding on Yes instances of the SZK problem and statistically hiding on No instances (and vice versa due to the fact that SZK is closed under complement).

In this work, we define a relaxation of zero-knowledge proofs, called *instance-dependent zero-knowledge*, and show that it suffices for the [6] approach when constructing a compiler from honest-verifier statistical zero-knowledge to general statistical zero-knowledge. Analogously to other instance-dependent primitives, soundness and zero-knowledge do not necessary hold simultaneously in instance-dependent zero-knowledge proofs but depending on the underlying instance of the given promise problem. We believe that this primitive is of independent interest and may find further applications beyond our compiler. We instantiate the instance-dependent zero-knowledge by employing the construction of instance-dependent commitments [14] in the constant-round zero-knowledge proof of knowledge for NP of Lindell [11] (see Section 4.1 for details). The instantiation and our compiler do not rely on any intractability assumption.

3 Preliminaries

Throughout the rest of the paper we use the following notation and definitions. For $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, \ldots, n\}$. A function $g : \mathbb{N} \to \mathbb{R}^+$ is *negligible* if it tends to 0 faster than any inverse polynomial, i.e., for all $c \in \mathbb{N}$ there exists $k_c \in \mathbb{N}$ such that for every $k > k_c$ it holds that $g(k) < k^{-c}$. We use $\mathsf{neg}(\cdot)$ to denote a negligible function if we do not need to specify its name.

A random variable X is a function from a finite set S to the nonnegative reals with the property that $\sum_{s \in S} X(s) = 1$. We write $x \leftarrow X$ to indicate that x is selected according to X. We write U_n to denote the random variable that is uniform over $\{0,1\}^n$. We use the terms random variable and probability distribution interchangeably

A probability ensemble is a set of random variables $\{A_x\}_{x \in \{0,1\}^*}$, where A_x takes values in $\{0,1\}^{p(|x|)}$ for some polynomial p. We call such an ensemble samplable if there is a probabilistic polynomial-time algorithm such that for every x, the output of the algorithm is distributed according to A_x.

3.1 Interactive Proof Systems

Definition 1 (Interactive proof system). *A pair of interactive machines* $\langle \mathsf{P}, \mathsf{V} \rangle$ *is called an* interactive proof system *for a language L if* V *is a PPT machine and there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that* $\forall k \in \mathbb{N}$ *the following holds:*

Completeness: *For all* $x \in L$*,*

$$\Pr[\langle \mathsf{P}, \mathsf{V} \rangle(x, 1^k) = 1] = 1 \ .$$

Soundness: *For all* $x \notin L$*, and every interactive machine* P^**,*

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x, 1^k) = 1] \leq \mathsf{neg}(k) \ .$$

Definition 2 (Proof of knowledge). *Let* $L \in \mathsf{NP}$ *and let* R_L *be its witness relation. An interactive proof system* $\langle P, V \rangle$ *for L is called* a proof of knowledge *(*PoK*) if it satisfies the following property:*

Knowledge Soundness: *There exists a PPT machine* E*, called the* extractor*, such that for every* P^**, for every* $x \in L$*, auxiliary input z, random tape r, and* $k \in \mathbb{N}$

$$\Pr[\mathsf{E}^{\mathsf{P}^*}(x, z, r; 1^k) = w : (x, w) \in R_L] \geq \Pr[\langle \mathsf{P}^*(z; r), \mathsf{V} \rangle(x, 1^k) = 1] - \mathsf{neg}(k) \ .$$

If the soundness property (resp. the knowledge soundness) in $\langle \mathsf{P}, \mathsf{V} \rangle$ holds only with respect to PPT provers, we call it an *interactive argument system* (resp. an *argument of knowledge*).

3.2 Statistical Zero-Knowledge

We use the standard definition of statistical difference of two probability distributions X, Y over universe $\mathbf{U}$, i.e.,

$$\mathrm{SD}\,(X, Y) = \max_{S \subset \mathbf{U}} |\Pr[X \in S] - \Pr[Y \in S]| \ .$$

Definition 3 (Promise problems). *A* promise problem *is specified by two disjoint sets of strings* $\Pi = (\Pi_Y, \Pi_N)$*, where* Π_Y *is the set of YES instances and* Π_N *is the set of NO instances. Any promise problem* Π *is associated with the following algorithmic task: given an input string that is promised to lie in* $\Pi_Y \cup \Pi_N$*, decide whether it is in* Π_Y *or in* Π_N*.*

Recall that the zero-knowledge property is captured via an existence of a simulator, an entity that simulates the view of the verifier in its interaction with the prover.

Definition 4 (View of an interactive protocol). *Let $\langle A, B\rangle$ be an interactive protocol. B's view of $\langle A, B\rangle$ on common input x is the random variable $(A, B)(x) = (m_1, \ldots, m_t; r)$ consisting of all the messages $m_1, \ldots, m_t$ exchanged between A and B together with the string r containing all the random bits that B has read during the interaction.*[4]

Statistical zero knowledge requires that the statistical difference between the simulator's output distribution and the verifier's view is so small that polynomially many repetitions of the protocol cannot make it noticeable. The definition allows the simulator to occasionally fail and output `fail`, and it only measures the quality of the simulation conditioned on non-failure.

Definition 5 (Honest-Verifier Statistical Zero-Knowledge). *An interactive proof system $\langle P, V\rangle$ for a promise problem Π is said to be* honest-verifier statistical zero-knowledge *if there exists a PPT S that fails with probability at most $1/2$ and a negligible function $\mathsf{neg}(\cdot)$ such that $\forall x \in \Pi_Y, k \in \mathbb{N}$,*

$$\mathrm{SD}\left(\widetilde{\mathsf{S}}(x, 1^k), (\mathsf{P}, \mathsf{V})(x, 1^k)\right) \leq \mathsf{neg}(k)$$

where $\widetilde{\mathsf{S}}$ is the output distribution of S conditioned on not failing. HVSZK denotes the class of all promise problems admitting honest-verifier statistical zero-knowledge proofs.

Zero knowledge against arbitrary verifier is captured by exhibiting a single, universal simulator S that simulates an arbitrary verifier strategy V^* by using V^* as a subroutine (denoted by $\mathsf{S}^{\mathsf{V}^*}$). That is, the simulator does not depend on or use the code of V^*, and instead only has black-box access to V^*. More formally,

Definition 6 (Statistical Zero-Knowledge). *An interactive proof system $\langle P, V\rangle$ for a promise problem Π is said to be* statistical zero-knowledge *if there exists a PPT S that fails with probability at most $1/2$ such that for every nonuniform PPT V^* it holds that*

$$\mathrm{SD}\left(\widetilde{\mathsf{S}}^{\mathsf{V}^*}(x, 1^k), (\mathsf{P}, \mathsf{V}^*)(x, 1^k)\right) \leq \mathsf{neg}(k) \qquad \forall x \in \Pi_Y, k \in \mathbb{N}\ ,$$

where $\widetilde{\mathsf{S}}$ is the output distribution of S conditioned on not failing, and $\mathsf{neg}(\cdot)$ is some negligible function that may depend on V^*. SZK denotes the class of all promise problems admitting statistical zero-knowledge proofs.

3.3 Instance-Dependent Commitment Schemes

Definition 7 (Instance-dependent commitment schemes). *An instance-dependent commitment scheme is a family of commitment schemes $\{\mathsf{Com}_x\}_{x\in\{0,1\}^*}$ with the following properties:*

[4] Note that equivalently we can define the view to be the messages from A to B and B's random bits. This is since the messages sent by B are a deterministic function of the received messages and the B's random bits.

1. *Scheme* Com_x *proceeds in two stages: a commit stage and a reveal stage. In both stages, the sender and receiver receive instance* x *as common input, and hence we denote the sender and receiver as* S_x *and* R_x*, respectively, and write* $\mathsf{Com}_x = (\mathsf{S}_x, \mathsf{R}_x, \mathsf{Open}_x)$*.*
2. *At the beginning of the commit stage, sender* S_x *receives a private input* $b \in \{0,1\}$*, which denotes the bit that* S_x *is supposed to commit to. At the end of the commit stage, both sender* S_x *and receiver* R_x *output a commitment* c*.*
3. *In the reveal stage, sender* S_x *sends a pair* (b, d)*, where* d *is the decommitment string for bit* b*. Receiver* R_x *outputs* $\mathsf{Open}_x(c, b, d) \in \{\texttt{accept}, \texttt{reject}\}$*.*
4. *The sender* S_x *and receiver* R_x *algorithms are computable in polynomial time (in* $|x|$*), given* x *as auxiliary input.*
5. *For every* $x \in \{0,1\}^*$*,* $\mathsf{Open}_x(c, b, d) = \texttt{accept}$ *with probability* 1 *if both sender* S_x *and receiver* R_x *follow their prescribed strategy.*

Definition 8 (Statistical hiding). *Instance-dependent commitment scheme* $\mathsf{Com}_x = (\mathsf{S}_x, \mathsf{R}_x, \mathsf{Open}_x)$ *is* statistically hiding *on* $I \subseteq \{0,1\}^*$ *if for every* R^**, the ensembles* $\{\mathsf{view}_{\mathsf{R}^*}(\mathsf{S}_x(0), \mathsf{R}^*)\}_{x \in I}$ *and* $\{\mathsf{view}_{\mathsf{R}^*}(\mathsf{S}_x(1), \mathsf{R}^*)\}_{x \in I}$ *are statistically indistinguishable, where the random variable* $\mathsf{view}_{\mathsf{R}^*}(\mathsf{S}_x(b), \mathsf{R}^*)$ *denotes the view of* R^* *in the commit stage interacting with* $\mathsf{S}_x(b)$*. For a promise problem* $\Pi = (\Pi_Y, \Pi_N)$*, an instance-dependent commitment scheme* Com_x *for* Π *is statistically hiding on the YES instances if* Com_x *is statistically hiding on* Π_Y*.*

Definition 9 (Statistical binding). *Instance-dependent commitment scheme* $\mathsf{Com}_x = (\mathsf{S}_x, \mathsf{R}_x, \mathsf{Open}_x)$ *is* statistically binding *on* $I \subseteq \{0,1\}^*$ *if for every* S^**, there exists a negligible function* neg *such that for all* $x \in I$*, the malicious sender* S^* *wins in the following game with probability at most* $\mathsf{neg}(|x|)$*.*

- S^* *interacts with* R_x *in the commit stage obtaining commitment* c*.*
- *Then* S^* *outputs* d_0 *and* d_1*, and it wins if* $\mathsf{Open}_x(c, 0, d_0) = \mathsf{Open}_x(c, 1, d_1) = \texttt{accept}$*.*

For a promise problem $\Pi = (\Pi_Y, \Pi_N)$*, an instance-dependent commitment scheme* Com_x *for* Π *is* statistically binding on the NO instances *if* Com_x *is statistically binding on* Π_N*.*

Theorem 3 ([14]). *Every problem* $\Pi = (\Pi_Y, \Pi_N) \in \mathsf{HVSZK}$ *has an instance-dependent commitment scheme that is statistically hiding on the YES instances and statistically binding on the NO instances. Moreover, the instance-dependent commitment scheme is public-coin and constant-round.*

Since HVSZK is closed under complement, for every $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathsf{HVSZK}$, we can also obtain instance dependent commitments in which the security properties are reversed (i.e., statistically binding on YES instances and statistically hiding a on NO instances).

4 Constant-Round Statistical Zero-Knowledge Proofs

In this section, we define a relaxation of zero-knowledge called *instance-dependent statistical zero-knowledge proofs*. We show that for the class NP it is possible to obtain constant-round instance-dependent statistical zero-knowledge proofs of knowledge without relying on computational assumptions. Next, using this relaxation of zero-knowledge for NP, we construct a constant-round statistical zero-knowledge proof for any promise problem in HVSZK.

4.1 Instance-Dependent Statistical Zero-knowledge Proofs

Instance-dependent statistical zero-knowledge proofs are a relaxation of the standard notion of statistical zero-knowledge proofs that allows the proof to depend on a specific promise problem Π. Similarly to instance-dependent commitment schemes [2, 10, 12], the prover and the verifier receive an instance x of the problem Π as auxiliary input and a statement ψ to prove. The proof system is required to be sound proof of knowledge when $x \in \Pi_{\mathrm{Y}}$ and zero-knowledge when $x \in \Pi_{\mathrm{N}}$.

Looking ahead, instance-dependent zero-knowledge proofs will be used as a sub-protocol within some outer protocol. Note that there are two instances involved: 1) an instance of the promise problem Π, for which the outer protocol is constructed and 2) an instance of the language L for which the instance-dependent proof system is used.

Definition 10 (Instance-dependent statistical zero-knowledge). An instance-dependent statistical zero-knowledge proof of knowledge for language L with respect to a promise problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ *is a family of protocols* $\{\langle P_x, V_x\rangle\}_{x\in\{0,1\}^*}$ *with the following properties:*

- $\langle P_x, V_x\rangle$ *is complete on all instances of* Π*, i.e., for all* $x \in \Pi_Y \cup \Pi_N$*.*
- $\langle P_x, V_x\rangle$ *is statistical zero-knowledge on the NO instances, i.e., for all* $x \in \Pi_N$*.*
- $\langle P_x, V_x\rangle$ *is a sound proof of knowledge on the YES instances, i.e., for all* $x \in \Pi_Y$*.*

We show that the protocol of Lindell [11] instantiated with the instance-dependent commitments of Ong and Vadhan [15] gives rise to a constant-round instance-dependent statistical zero-knowledge proof of knowledge for NP.

Theorem 4. *For every promise problem* $\Pi = (\Pi_Y, \Pi_N) \in \mathsf{HVSZK}$ *and for every language* $L \in \mathsf{NP}$*, there exists a constant-round instance-dependent statistical zero-knowledge proof of knowledge for* L *with respect to* Π*. Moreover, the zero-knowledge property holds against unbounded verifiers.*

Similarly to instance-dependent commitments, for all $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathsf{HVSZK}$, we can obtain instance-dependent statistical zero-knowledge with the security properties reversed, i.e., with knowledge soundness on NO instances and statistical zero-knowledge on YES instances.

Let x be an instance of Π and let Com_x^{sh} and Com_x^{sb} be instance-dependent commitment schemes.
Input: a graph $G = (V, E)$, with $n = |V|$, and security parameter 1^k.
Prover's auxiliary input: a directed Hamiltonian cycle $C \subseteq E$ in G.
The protocol $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ for proving $G \in HC$ proceeds as follows:

1. P_x sends n independent copies of the first message for the basic proof of Hamiltonicity. That is, for $1 \leq i \leq n$, P_x selects a random permutation π_i over the vertices V and interacts with V_x to commit (using Com_x^{sb}) to the entries of the adjacency matrix of the resulting permuted graph. That is, P_x commits to an n-by-n matrix so that the entry $(\pi_i(\ell), \pi_i(j))$ contains a commitment to 1 if $(\ell, j) \in E$, and it contains a commitment to 0 otherwise.
   (a) V_x samples $q_1 \leftarrow \{0,1\}^n$ and interacts with P_x in Com_x^{sh}, so that P_x learns c_1, a commitment to q_1.
   (b) P_x samples $q_2 \leftarrow \{0,1\}^n$ and interacts with V_x in Com_x^{sb}, so that V_x learns c_2, a commitment to q_2.
   (c) V_x opens the commitment c_1 by sending q_1 and a decommitment string d_1.
   (d) If $\mathsf{Open}_x^{sh}(c_1, q_1, d_1) = \mathtt{reject}$, then P_x aborts and halts. Otherwise, P_x opens the commitment c_2 by sending q_2 and a decommitment string d_2.
2. P_x computes an n bit string $q = q_1 \oplus q_2$ and sends the second message for the basic proof of Hamiltonicity for each of the n copies, where P_x uses the i-th bit of q as the verifier's query in the i-th copy. That is, for $1 \leq i \leq n$ do:
   – If $q(i) = 0$, then send π_i and open all the commitments in the adjacency matrix of the i-th instance.
   – If $q(i) = 1$, open *only* the commitments of entries $(\pi_i(\ell), \pi_i(j))$ for which $(\ell, j) \in C$.
3. V_x computes $q = q_1 \oplus q_2$. If either $\mathsf{Open}_x^{sb}(c_2, q_2, d_2) = \mathtt{reject}$ or the response of the prover is not accepting in all n copies, based on the queries according to q, then output $\mathtt{reject}$. Otherwise, output $\mathtt{accept}$.

Fig. 1. The instance-dependent statistical zero-knowledge proof of knowledge $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ for NP-complete problem Hamiltonian Cycle with respect to a promise problem $\Pi \in \mathsf{HVSZK}$. The protocol builds on the constant-round zero-knowledge proof of knowledge of Lindell [11] which we instantiate with instance-dependent commitments relative to an instance x of Π.

Proof (Proof of Theorem 4).
Let $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathsf{HVSZK}$ be some promise problem and denote by HC the Hamiltonian Cycle language. Let x be an instance of Π, let Com_x^{sb} be an instance-dependent commitment scheme that is statistically binding on Π_{Y} and statistically hiding on Π_{N}. Let Com_x^{sh} be an instance-dependent commitment scheme that is statistically binding on Π_{N} and statistically hiding on Π_{Y}. The protocol is formally presented in Figure 1. Since HC is NP-complete, we obtain a proof system for any language in NP by a standard reduction.

Lindell [11] showed that if the verifier commits using a statistically hiding scheme Com_x^{sh} and the prover commits using a statistically binding scheme Com_x^{sb} then the protocol in Figure 1 is sound proof of knowledge for HC. Since Com_x^{sh} and Com_x^{sb} satisfy this requirement on Π_{Y}, we obtain that $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ is sound

Input: instance x of $\Pi \in$ HVSZK, a graph $G = (V, E)$, with $n = |V|$, and security parameter 1^k. Given oracle access to verifier V^*, the simulator S works as follows:

1. S chooses a random string $q \in \{0,1\}^n$. Then, for the prover's message in the i-th execution, S interacts in Com_x^{sb}, so that V^* learns a commitment to a random permutation of G if $q(i) = 0$, and to a simple n-cycle if $q(i) = 1$.
2. S honestly interacts with V^* in Com_x^{sh}, and learns the verifier's commitment c_1. S chooses a random q_2 and interacts with V^* in Com_x^{sb}, so that V^* learns c_2, a commitment to q_2.
3. S receives q_1 and the decommitment string d_1 from V^*. If $\mathsf{Open}_x^{sh}(c_1, q_1, d_1) =$ `reject`, then S simulates P_x aborting, outputs whatever V^* outputs and halts. Otherwise, S proceeds to the next step.
4. **Rewinding phase:**
   (a) S rewinds V^* back to the point before the interaction in Com_x^{sh}. S interacts honestly with V^* to produce a commitment c_2 for value $q_1 \oplus q$.
   (b) S receives q_1' and d_1' from V^* and proceeds as follows:
      - If $\mathsf{Open}_x^{sh}(c_1, q_1', d_1') =$ `reject` then abort on behalf of P_x, S outputs whatever V^* outputs and halts.
      - If $\mathsf{Open}_x^{sh}(c_1, q_1', d_1') =$ `accept` but $q_1' \neq q_1$ then S outputs `fail` and halts.
      - Otherwise, S opens the commitment c_2 and for each $i \in [n]$, opens the commitments either to the entire graph (for $q(i) = 0$) or the simple cycle (for $q(i) = 0$).
5. S outputs whatever V^* outputs.

Fig. 2. Simulator for the protocol in Figure 1.

proof of knowledge for HC with respect to all $x \in \Pi_{\mathrm{Y}}$. Therefore, it is only left to show that $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ is statistical zero-knowledge against unbounded verifiers with respect to all $x \in \Pi_{\mathrm{N}}$.

Note that when $x \in \Pi_{\mathrm{N}}$, the commitment Com_x^{sh} is statistically binding and Com_x^{sb} is statistically hiding. In Figure 2, we present a simulator that produces a distribution of transcripts which is statistically close to the real distribution of transcripts.

Lemma 5. *For all $x \in \Pi_N$, every input graph $G = (V, E)$, every security parameter $k \in \mathbb{N}$, and any verifier V^*, it holds that*

$$\Pr[\mathsf{S}^{\mathsf{V}^*}(x, G, 1^k) = \mathtt{fail}] \leq \mathsf{neg}(k) \ .$$

Proof. Given $x \in \Pi_{\mathrm{N}}$, let V^* be an arbitrary verifier. We get that

$$\begin{aligned} &\Pr[\mathsf{S}^{\mathsf{V}^*}(x, G, 1^k) = \mathtt{fail}] \\ &\qquad \leq \Pr[\exists c, q_1, d_1, q_1', d_1' \colon \mathsf{Open}_x^{sh}(c, q_1, d_1) = \mathsf{Open}_x^{sh}(c, q_1', d_1') = \mathtt{accept}] \end{aligned}$$

which is at most negligible in the security parameter since the commitment scheme Com_x^{sh} is statistically binding for any $x \in \Pi_{\mathrm{N}}$. □

Note that the simulator S rewinds V^* such that the initially chosen string q is the coin-flipping result. In this case, S can decommit appropriately and conclude the proof. The statistical closeness of the distribution of transcripts produced by the simulator and the real distribution of transcripts follows from the statistical hiding of Com_x^{sb} combined with the statistical binding of Com_x^{sh}.

Due to statistical hiding of Com_x^{sb}, the probability over q_2 and r_2 that V^* decommits to c_1 in the main thread (before rewinding) is basically equivalent to the probability that V^* decommits to c_1 in the rewind. Thus, the only difference between the output distribution generated by S and the output distribution generated in a real proof is that in the case that $q(i) = 1$ the unopened commitments in the simulated transcript are all to 0, and not to the rest of the graph apart from the cycle. However, due to the statistical hiding property of Com_x^{sb} on $x \in \Pi_{\mathrm{N}}$, the distributions are statistically close. This completes the proof of Theorem 4. $\square$

4.2 A Concrete Protocol for a SZK-Complete Problem

In this section, we show that $\mathsf{HVSZK} \subseteq \mathsf{SZK}[c]$, where $\mathsf{SZK}[c]$ is the class of all promise problems that admit constant-round statistical zero-knowledge poof. Concretely, in Figure 3 we present a simple constant-round statistical zero-knowledge protocol secure against any malicious verifier for a complete problem in HVSZK, called Statistical-Difference. The constant-round protocol for any problem in HVSZK would comprise of a reduction to Statistical-Difference (which can be performed locally by both P and V) and then running our protocol.

First, we recall the Statistical-Difference problem which was shown to be HVSZK-complete by Sahai and Vadhan [17]. In this work we consider the polarized form of Statistical-Difference, that can be obtained from the basic definition in polynomial-time.

Definition 11 (Statistical-Difference). *Given $k \in \mathbb{N}$, the promise problem* Statistical-Difference *is $\mathsf{SD} = (\mathsf{SD}_Y, \mathsf{SD}_N)$, where*

$$\mathsf{SD}_Y = \{(X_0, X_1) : \mathrm{SD}(X_0, X_1) \geq 1 - 2^{-k}\}\ ,$$
$$\mathsf{SD}_N = \{(X_0, X_1) : \mathrm{SD}(X_0, X_1) \leq 2^{-k}\}\ .$$

Above, X_0, X_1 are circuits encoding probability distributions.

Given $\overline{X} = (X_0, X_1)$, an instance of Statistical-Difference, our protocol builds on the standard honest-verifier statistical zero-knowledge proof for Statistical-Difference of Sahai and Vadhan [17]. To force the verifier to behave as in the original honest-verifier protocol, we use 1) a constant-round instance-dependent commitment scheme $\mathsf{Com}_{\overline{X}} = (\mathsf{S}_{\overline{X}}, \mathsf{R}_{\overline{X}}, \mathsf{Open}_{\overline{X}})$ that is statistically binding on SD_Y, and 2) a constant-round instance-dependent statistical zero-knowledge proof of knowledge $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$ for NP that is zero-knowledge on SD_N against any unbounded verifier. These building blocks are provided by Theorem 3 and Theorem 4, respectively. The protocol is formally presented in Figure 3.

Input: Given $\overline{X} = (X_0,\ X_1)$, a pair of circuits, and security parameter 1^k. Let $\mathsf{Com}_{\overline{X}} = (\mathsf{S}_{\overline{X}}, \mathsf{R}_{\overline{X}}, \mathsf{Open}_{\overline{X}})$ be an instance-dependent commitment scheme that is statistically binding on SD_Y and let $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$ be an instance dependent statistical zero-knowledge proof of knowledge for NP with knowledge soundness on SD_Y. The protocol $\mathsf{SD} = \langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}_{\mathsf{SD}} \rangle$ for proving $\overline{X} \in \mathsf{SD}_Y$ proceeds as follows:

1. **Coin flipping phase:**
   (a) V_{SD} samples $r_0 \leftarrow \{0,1\}^n$, $b_0 \leftarrow \{0,1\}$.
   (b) V_{SD} and P_{SD} interact in $\mathsf{Com}_{\overline{X}}$, so that P_{SD} learns c, a commitment to the pair (r_0, b_0).
   (c) P_{SD} samples $r_1 \leftarrow \{0,1\}^n$, $b_1 \leftarrow \{0,1\}$ and sends them to V_{SD}. Then, V_{SD} sets $b = b_0 \oplus b_1$ and $r = r_0 \oplus r_1$.
2. **Honest SD-protocol execution phase:**
   (a) V_{SD} sends $y = X_b(r)$ to P_{SD}.
   (b) Let $L_{\text{samp}} = \{(c, r', b', y) | \exists r, b, d : \mathsf{Open}_{\overline{X}}(c, (r, b), d) = \mathtt{accept} \wedge y = X_{b' \oplus b}(r' \oplus r)\}$. V_{SD} uses $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$ to prove to P_{SD} that $(c, r_1, b_1, y) \in L_{\text{samp}}$. Denote by θ the transcript of this proof.
   (c) If $\mathsf{V}_{\overline{X}}$ rejects θ then P_{SD} aborts, otherwise, it replies with $b'' \in \{0,1\}$ such that
   $$\Pr_{r \leftarrow U_n}[X_{b''}(r) = y] \geq \Pr_{r \leftarrow U_n}[X_{1-b''}(r) = y] .$$
   (d) If $b = b''$ then V_{SD} outputs $\mathtt{accept}$, otherwise outputs $\mathtt{reject}$.

Fig. 3. The statistical zero-knowledge proof $\langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}_{\mathsf{SD}} \rangle$ for Statistical-Difference. Our protocol builds on the honest-verifier statistical zero-knowledge proof of Sahai and Vadhan [17] with the following changes: 1) The verifier's randomness is picked mutually by the verifier and the prover (while maintaining the secrecy to the prover). 2) The verifier is required to provide a proof that it used the mutually chosen randomness.

Theorem 6. *The protocol presented in Figure 3 is constant-round statistical zero-knowledge proof for* Statistical-Difference*.*

By completeness of Statistical-Difference for HVSZK, we obtain a constant-round protocol secure against any verifier for every problem in the class.

Corollary 7. *There exists a constant-round statistical zero-knowledge proof for every $\Pi \in \mathsf{HVSZK}$, where the zero-knowledge holds against any malicious verifier.*

Proof of Theorem 6. Here we show that the protocol in Figure 3 is complete, sound and achieves statistical zero-knowledge.

Completeness. Due to the perfect completeness of the $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$ proof , it follows that the completeness error of our protocol is the same as the completeness error of the standard protocol for SD of [17], i.e., at most 2^{-k}.

Soundness. We present here a proof sketch. The full proof can be found in Section 5, where we present the general transformation. Given $\overline{X} = (X_0, X_1) \in \mathsf{SD}_N$, a NO instance of Statistical-Difference, let P^* be an arbitrary prover. Let

Input: Given $\overline{X} = (X_0,\ X_1)$ and security parameter 1^k. Let E be the extractor of $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$ scheme. The simulator S_{SD} with oracle access to V^* proceeds as follows:

1. Execute honestly the protocol up to the last round with $\mathsf{V}^*(x)$ in order to learn a commitment c, and a sample y. Let b_1 and r_1 be the values given to $\mathsf{V}^*(x)$ in the simulated coin-flipping phase. Participate as the honest $\mathsf{V}_{\overline{X}}$ in the proof of knowledge for the committed value in c and correctness of y. Denote this proof of knowledge θ.
2. If θ is not accepting then abort. Otherwise, use the knowledge extractor $\mathsf{E}^{\mathsf{V}^*}$ to extract the values r_0^*, b_0^*, d^*. If the extractor fails output `fail`.
3. Send $b = b_0^* \oplus b_1$ to V^*.
4. Output the simulated transcript and r_0^*, b_0^*, d^* as the randomness of V^*.

Fig. 4. Simulator $\mathsf{S}_{\mathsf{SD}}^{\mathsf{V}^*}$ for protocol $\langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}_{\mathsf{SD}} \rangle$. The simulator honestly participates in an execution with V^* but instead of sending the last message, it extracts the randomness of the verifier and uses it to generate the last message.

$\mathsf{Com}_{\overline{X}}$ and $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$ be as defined above. Finally, let $\mathsf{Sim}_{\overline{X}}$ be the statistical zero-knowledge simulator for $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$.

We show that the soundness error in the above protocol is at most negligibly larger than the soundness error in the original honest-verifier protocol. This follows from the statistical zero-knowledge property against unbounded verifiers of $\langle \mathsf{P}_{\overline{X}}, \mathsf{V}_{\overline{X}} \rangle$, and the statistical hiding property of $\mathsf{Com}_{\overline{X}}$. Specifically, the distribution of transcripts $\langle \mathsf{P}^*, \mathsf{V}_{\mathsf{SD}} \rangle(\overline{X})$ is statistically close to the distribution of transcripts where the proof in Step 2b is performed using $\mathsf{Sim}_{\overline{X}}$ (this can be done since V is honest, and proves a true statement). Note that when Step 2b is performed using $\mathsf{Sim}_{\overline{X}}$, the acceptance probability of V is equivalent to its acceptance probability in a protocol where the proof of Step 2b is not performed at all. We can use the statistical hiding property of $\mathsf{Com}_{\overline{X}}$ to argue that the distribution of transcripts of the protocol without Step 2b is in turn statistically close to a distribution of transcripts where the verifier commits to a fixed value (r^*, b^*) and uses uniformly random r_0, b_0 to compute $y = X_{b_0 \oplus b_1}(r_0 \oplus r_1)$. However, this corresponds exactly to the original honest-verifier protocol of Sahai and Vadhan [17]. Therefore, the soundness error can be at most negligibly larger.

Statistical Zero-Knowledge. For any V^*, the simulator S_{SD} proceeds as described in Figure 4.

Lemma 8. *For all PPT V^*, $\overline{X} \in \mathsf{SD}_Y$, and $k \in \mathbb{N}$, it holds that*

$$\Pr[\mathsf{S}_{\mathsf{SD}}^{\mathsf{V}^*}(\overline{X}, 1^k) = \texttt{fail}] \leq 1/2 \ .$$

Proof. Let V^* be some PPT verifier, let $\overline{X} \in \mathsf{SD}_Y$ be some input, and let k be the security parameter. Note that $\mathsf{S}_{\mathsf{SD}}^{\mathsf{V}^*}$ fails only when V^* provides an accepting proof of knowledge of the value committed in c while the extractor fails to extract this value. Therefore,

$$\Pr[\mathsf{S}_{\mathsf{SD}}^{\mathsf{V}^*}(\overline{X}, 1^k) = \texttt{fail}]$$

$$\leq \Pr[\mathsf{V}_{\overline{X}}(c, r_1, b_1, y, \theta) = \texttt{accept} \wedge \mathsf{E}_{\overline{X}}^{\mathsf{V}^*}(c, r_1, b_1, y, \theta) = \texttt{fail}] \ ,$$

where (c, r_1, b_1, y, θ) is the partial transcript produced by $\mathsf{S}_{\mathsf{SD}}^{\mathsf{V}^*}(\overline{X}, 1^k)$ in Step 1 of the simulation. Since $\mathsf{S}_{\mathsf{SD}}^{\mathsf{V}^*}$ behaves in Step 1 exactly as the honest prover P_{SD}, we can switch to $(c, r_1, b_1, y, \theta) \leftarrow \langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}^* \rangle(\overline{X}, 1^k)$, and obtain the following series of inequalities.

$$\leq \Pr[\mathsf{V}_{\overline{X}}(c, r_1, b_1, y, \theta) = \texttt{accept}] \cdot (1 - \Pr[\mathsf{E}_{\overline{X}}^{\mathsf{V}^*}(c, r_1, b_1, y, \theta) \neq \texttt{fail}])$$

$$\leq \Pr[\mathsf{V}_{\overline{X}}(c, r_1, b_1, y, \theta) = \texttt{accept}] \cdot (1 - \Pr[\mathsf{V}_{\overline{X}}(c, r_1, b_1, y, \theta) = \texttt{accept}] + \mathsf{neg}(k))$$

$$< 1/2 \ ,$$

where $(c, r_1, b_1, y, \theta) \leftarrow \langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}^* \rangle(\overline{X}, 1^k)$. □

To complete the proof, we show that conditioned on not outputting `fail`, the output distribution of $\mathsf{S}_{\mathsf{SD}}^{\mathsf{V}^*}$ is statistically close to the view of V^*. Due to the statistical binding of $\mathsf{Com}_{\overline{X}}$, the extracted randomness is distributed statistically close to the randomness of V^*. Moreover, the simulated transcript in Step 1 is distributed identically to $\langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}^* \rangle$. Given this observation, it is sufficient to bound the probability that the last message of the simulated transcript differs from the last message of the real transcript (the real and the simulated transcript distributions are otherwise identical).

Lemma 9. *For all PPT V^*, $\overline{X} \in \mathsf{SD}_Y$, and $k \in \mathbb{N}$, it holds that*

$$\Pr[\widetilde{\mathsf{S}}_{\mathsf{SD}}^{\mathsf{V}^*}(\overline{X}, c, r_1, b_1, y, \theta) \neq b''] \leq \mathsf{neg}(k) \ ,$$

where $(c, r_1, b_1, y, \theta, b'') \leftarrow \langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}^* \rangle(\overline{X}, 1^k)$, and $\widetilde{\mathsf{S}}_{\mathsf{SD}}^{\mathsf{V}^*}(\overline{X}, c, r_1, b_1, y, \theta)$ denotes simulator's message in Step 3 on input $\overline{X}$ and transcript prefix (c, r_1, b_1, y, θ), conditioned on not outputting `fail`.

Proof. Let V^* be some PPT verifier, let $\overline{X} \in \mathsf{SD}_Y$ be some input, and let k be the security parameter. The claim follows from the fact that the transcripts may differ if either the statistical binding does not hold or the verifier samples a value from one of the distributions such that the probability of this value in the other distribution is higher (this event happens with 2^{-k} probability). That is,

$$\Pr[\widetilde{\mathsf{S}}_{\mathsf{SD}}^{\mathsf{V}^*}(\overline{X}, c, r_1, b_1, y, \theta) \neq b'']$$

$$\leq \Pr[c \text{ is not binding }] + \Pr[\exists r_0^*, d^* : \mathsf{Open}_{\overline{X}}(c, r_0^*, 1 - b'', d^*) = \texttt{accept}]$$

$$\leq \mathsf{neg}(k) \ ,$$

where $(c, r_1, b_1, y, \theta, b'') \leftarrow \langle \mathsf{P}_{\mathsf{SD}}, \mathsf{V}^* \rangle(\overline{X}, 1^k)$. □

Lemma 9 completes the proof of Theorem 6.

5 Efficient Transformation from Honest-Verifier SZK to SZK

The general transformation takes any honest-verifier statistical zero-knowledge protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ for promise problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathsf{HVSZK}$, an instance $x \in \Pi$, a constant-round instance-dependent commitment scheme Com_x that is statistically binding on Π_{Y} instances, and a constant-round instance-dependent statistical zero-knowledge proof of knowledge protocol $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ for NP (from Theorem 4), and constructs a statistical zero-knowledge proof for Π.

Theorem 10 (Theorem 1 restated). *For every promise problem $\Pi \in \mathsf{HVSZK}$, there exists a statistical zero-knowledge proof where the prover's complexity and the round complexity match the parameters of the best hones-verifier statistical zero-knowledge proof for Π.*

The transformation is given in Figure 5. We establish the proof of Theorem 10 by arguing its correctness, soundness, and zero-knowledge property below.

Correctness. Correctness of the compiled protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$ follows directly from correctness of the building blocks, i.e., the instance-dependent statistical zero-knowledge proof of knowledge for NP $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ and the honest-verifier statistical zero-knowledge proof $\langle \mathsf{P}, \mathsf{V} \rangle$.

Soundness. Soundness of the compiled protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$ follows from the soundness of the basic honest-verifier procol $\langle \mathsf{P}, \mathsf{V} \rangle$ combined with the instance-dependent zero-knowledge proofs for NP being statistical zero-knowledge against unbounded verifiers on Π_{N}. Moreover, the statistical hiding property of Com_x on Π_{N} allows V' to use random coins distributed almost identically as the randomness of V (the distribution of randomness might be influenced by a cheating prover only if the hiding property does not hold).

Proposition 11 (Soundness of $\langle \mathsf{P}', \mathsf{V}' \rangle$). *Let $\Pi = (\Pi_Y, \Pi_N) \in \mathsf{HVSZK}$, and let $\langle \mathsf{P}, \mathsf{V} \rangle$ be honest-verifier statistical zero-knowledge protocol for Π. For all $x \in \Pi_N$, $k \in \mathbb{N}$, and P^*, it holds that*

$$\Pr[\langle \mathsf{P}^*, V' \rangle(x, 1^k) = 1] = \eta_{\langle \mathsf{P}, \mathsf{V} \rangle} + \mathsf{neg}(k) \ ,$$

where $\eta_{\langle \mathsf{P}, \mathsf{V} \rangle}$ denotes the soundness error of $\langle \mathsf{P}, \mathsf{V} \rangle$.

Proof. The proof of soundness follows from a series of lemmas. First, we define protocol $\langle \mathsf{P}_r, \mathsf{V}_r \rangle$ to be the same as the compiled protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$ but without the proofs of correctness provided by V'. We use $\langle \mathsf{P}_r, \mathsf{V}_r \rangle$ to argue that the-coin flipping phase alone increases the soundness error by at most a negligible amount over $\eta_{\langle \mathsf{P}, \mathsf{V} \rangle}$.

Lemma 12. *For all $x \in \Pi_N$, $k \in \mathbb{N}$, and P_r^*, it holds that*

$$\Pr[\langle \mathsf{P}_r^*, \mathsf{V}_r \rangle(x, 1^k) = 1] \le \eta_{\langle \mathsf{P}, \mathsf{V} \rangle} + \mathsf{neg}(k) \ .$$

Input: Given $x \in \Pi$ and security parameter 1^k. Let $\mathsf{Com}_x = (\mathsf{S}_x, \mathsf{R}_x, \mathsf{Open}_x)$ be a constant-round instance-dependent commitment scheme that is statistically binding on Π_{Y}, and let $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ be a constant-round instance-dependent statistical zero-knowledge proof of knowledge for NP with knowledge soundness on Π_{Y}. The protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$ for proving $x \in \Pi_{\mathrm{Y}}$ proceeds as follows:

1. **Coin-flipping phase:**
   (a) V' samples $r_{\mathsf{V}} \leftarrow \{0,1\}^{t_{\mathsf{V}}}$, where t_{V} is a bound on the running time of V.
   (b) V' and P' interact in Com_x, so that P' learns c, a commitment to r_{V}.
   (c) V' and P' run $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$, where V' proves that it knows an opening for c. Denote the transcript this proof θ_c. P' aborts if θ_c is not accepting.
   (d) P' samples $r_{\mathsf{P}} \leftarrow \{0,1\}^{t_{\mathsf{V}}}$ and sends r_{P} to V' that sets $r = r_{\mathsf{V}} \oplus r_{\mathsf{P}}$.
2. **Honest-verifier protocol execution phase:** V' and P' engage in an execution of the honest-verifier protocol $\langle \mathsf{P}, \mathsf{V} \rangle$. For each round $1 \leq i \leq t$ of $\langle \mathsf{P}, \mathsf{V} \rangle$ they proceed as follows:
   (a) denote by $\tau_{i-1} = (\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$ the transcript of $\langle \mathsf{P}, \mathsf{V} \rangle$ up to round $i-1$ (included).
   (b) V' computes the i-th message $\alpha_i = \mathsf{V}_i(x, \tau_{i-1}; r)$ of V and sends α_i to P'.
   (c) Let $L_i = \{(c, r, \tau, \alpha) | \exists \tilde{r}, d : \mathsf{Open}_x(c, \tilde{r}, d) = \mathtt{accept} \wedge \alpha = \mathsf{V}_i(x, \tau; \tilde{r} \oplus r\})\}$. V' proves to P' that $(c, r_{\mathsf{P}}, \tau_{i-1}, \alpha_i) \in L_i$ using $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$. Denote the transcript of this proof θ_i. P' aborts if θ_i is not accepting.
   (d) P' computes the i-th message $\beta_i \leftarrow \mathsf{P}_i(x, \tau_{i-1}, \alpha_i)$ of P and sends β_i to V'.
3. If $\mathsf{V}_{t+1}(x, \tau_t; r) = \mathtt{accept}$ then V' outputs $\mathtt{accept}$, and otherwise $\mathtt{reject}$.

Fig. 5. Compiled protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$. A compiler from honest-verifier protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ for promise problem Π to protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$ that is zero-knowledge against general verifiers. For a t-round protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ we denote by V_i the next-message function of V in round i computed on the input, the $(i-1)$-rounds transcript, and the random tape of V (where V_{t+1} refers to the output of V in the protocol). The next-message function is similarly defined for P.

Proof. We consider an intermediate protocol, denoted by $\langle \mathsf{P}_1, \mathsf{V}_1 \rangle$. The protocol $\langle \mathsf{P}_1, \mathsf{V}_1 \rangle$ is the same as $\langle \mathsf{P}_r, \mathsf{V}_r \rangle$ with the difference that V_1 commits to $0^{t_{\mathsf{V}}}$ and uses a uniformly random string independent of r_{P} as its randomness.

First, we show that for all $x \in \Pi_{\mathrm{N}}$, $k \in \mathbb{N}$, and P_1^*, it holds that

$$\Pr[\langle \mathsf{P}_1^*, \mathsf{V}_1 \rangle(x, 1^k) = 1] \leq \eta_{\langle \mathsf{P}, \mathsf{V} \rangle} \ .$$

This is shown by constructing a prover P^* that wins the security game for $\langle \mathsf{P}, \mathsf{V} \rangle$ with the same probability as P_1^*. The constructed P^* simulates for P_1^* the coin-flipping phase using a commitment to all-zero string, receives r_{P} and answers all messages from V with messages from P_1^*. It follows from construction of P_1^* that $\Pr[\langle \mathsf{P}_1^*, \mathsf{V}_1 \rangle(x, 1^k) = 1] = \Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x, 1^k) = 1] \leq \eta_{\langle \mathsf{P}, \mathsf{V} \rangle}$.

Next, we show that for all $x \in \Pi_{\mathrm{N}}$, $k \in \mathbb{N}$, and P_r^*, it holds that

$$\Pr[\langle \mathsf{P}_r^*, \mathsf{V}_r \rangle(x, 1^k) = 1] \leq \eta_{\langle \mathsf{P}, \mathsf{V} \rangle} + \mathsf{neg}(k) \ .$$

The bound follows from the statistical hiding property of Com_x on NO instances, i.e., on Π_{N}. Specifically, the transcripts of the coin-flipping phase in

$\langle \mathsf{P}_r^*, \mathsf{V}_r \rangle$ and in $\langle \mathsf{P}_r^*, \mathsf{V}_1 \rangle$ are statistically indistinguishable. This completes the proof of Lemma 12. $\square$

We now define a sequence of hybrid protocols that gradually move between the interaction in $\langle \mathsf{P}_r, \mathsf{V}_r \rangle$ (where the verifier does not provide any proof of correctness for its messages) and the interaction in $\langle \mathsf{P}', \mathsf{V}' \rangle$ (where every message of V' is followed by a proof of correctness). Let t be the number of rounds in $\langle \mathsf{P}, \mathsf{V} \rangle$, we define $t+2$ protocols as follows:

Protocol $\langle \mathsf{P}', \mathsf{V}'_0 \rangle$ is defined similarly to $\langle \mathsf{P}', \mathsf{V}' \rangle$, where V'_0 behaves as V', except that V'_0 provides simulated proofs using the simulator for $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$.

Protocol $\langle \mathsf{P}', \mathsf{V}'_i \rangle$ is defined for $1 \leq i \leq t+1$. The protocol $\langle \mathsf{P}', \mathsf{V}'_i \rangle$ is the same as $\langle \mathsf{P}', \mathsf{V}'_{i-1} \rangle$, except that V'_i performs the i-th proof using the actual witness instead of the simulator.

Note that $\langle \mathsf{P}', \mathsf{V}'_{t+1} \rangle$ is equivalent to $\langle \mathsf{P}', \mathsf{V}' \rangle$. Moreover, the soundness error of $\langle \mathsf{P}', \mathsf{V}'_0 \rangle$ is equal to the soundness error of $\langle \mathsf{P}_r, \mathsf{V}_r \rangle$. This can be seen by converting any cheating prover P'^* for $\langle \mathsf{P}', \mathsf{V}'_0 \rangle$ to a cheating prover P_r^* for $\langle \mathsf{P}_r, \mathsf{V}_r \rangle$. Concretely, on input x, the constructed prover P_r^* internally runs P'^* and provides it with simulated proof after each message from V_r. It follows that $\Pr[\langle \mathsf{P}'^*, \mathsf{V}'_0 \rangle(x, 1^k) = 1] = \Pr[\langle \mathsf{P}_r^*, \mathsf{V}_r \rangle(x, 1^k) = 1] \leq \eta_{\langle \mathsf{P}, \mathsf{V} \rangle} + \mathsf{neg}(k)$.

Lemma 13. *For all $x \in \Pi_N$, for every $k \in \mathbb{N}$, any prover P'^*, and $1 \leq i \leq t+1$, it holds that*

$$\mathrm{SD}\left(\langle \mathsf{P}'^*, \mathsf{V}'_i \rangle(x, 1^k), \langle \mathsf{P}'^*, \mathsf{V}'_{i-1} \rangle(x, 1^k) \right) \leq \mathsf{neg}(k) \ .$$

Proof. The only difference in two consecutive hybrid protocols $\langle \mathsf{P}'^*, \mathsf{V}'_{i-1} \rangle$ and $\langle \mathsf{P}'^*, \mathsf{V}'_i \rangle$ is the simulated vs. the real proof in the i-th round when executing $\langle \mathsf{P}', \mathsf{V}' \rangle$. Assume towards a contradiction that there exists $x \in \Pi_{\mathrm{N}}$, a prover P'^*, and $1 \leq j \leq t+1$ such that for some polynomial p it holds that

$$\mathrm{SD}\left(\langle \mathsf{P}'^*, \mathsf{V}'_j \rangle(x, 1^k), \langle \mathsf{P}'^*, \mathsf{V}'_{j-1} \rangle(x, 1^k) \right) \geq p(k) \ .$$

We show that there exists an unbounded verifier V_x^*, and a partial transcript (c, r, τ, α) up to round j such that $(c, r, \tau, \alpha) \in L_j$ and

$$\mathrm{SD}\left((\mathsf{P}_x, \mathsf{V}_x^*)(c, r, \tau, \alpha; 1^k), \mathsf{S}^{\mathsf{V}_x^*}(c, r, \tau, \alpha; 1^k) \right) \geq p(k) \ .$$

We define V_x^* and the partial transcript as follows. To obtain the partial transcript, run P'^* and simulate V' honestly during the first $j-1$ rounds of $\langle \mathsf{P}', \mathsf{V}' \rangle$ and compute the j-th round message α. Let (c, r, τ, α) be the partial transcript so far. We define V_x^* to be identical to the behavior of P'^* in the proof of the j-th round. Note that we can complete the partial transcript to a full transcript of $\langle \mathsf{P}', \mathsf{V}' \rangle$ by continuing with the internal run of P'^* and providing it with simulated proofs for the remaining rounds $j+1, \ldots, t+1$, as if they were generated by the honest V'. Thus, if the proof provided at round j is simulated then the complete transcript is drawn from $\langle \mathsf{P}'^*, \mathsf{V}'_{j-1}\rangle(x, 1^k)$ and otherwise it is drawn from $\langle \mathsf{P}'^*, \mathsf{V}'_j\rangle(x, 1^k)$. Therefore, we obtain that

$$\begin{aligned}\mathrm{SD}\left((\mathsf{P}_x, \mathsf{V}_x^*)(c, r, \tau, \alpha; 1^k), \mathsf{S}^{\mathsf{V}_x^*}(c, r, \tau, \alpha; 1^k)\right) \\ \geq \mathrm{SD}\left(\langle \mathsf{P}'^*, \mathsf{V}'_j\rangle(x, 1^k), \langle \mathsf{P}'^*, \mathsf{V}'_{j-1}\rangle(x, 1^k)\right) .\end{aligned}$$

Hence,

$$\mathrm{SD}\left((\mathsf{P}_x, \mathsf{V}_x^*)(c, r, \tau, \alpha; 1^k), \mathsf{S}^{\mathsf{V}_x^*}(c, r, \tau, \alpha; 1^k)\right) \geq p(k) \ ,$$

contradicting the statistical zero-knowledge property (against unbounded verifiers) of $\langle \mathsf{P}_x, \mathsf{V}_x\rangle$. □

Given that we have polynomially many hybrids and they are all statistically close, Lemma 13 completes the proof of soundness. □

Statistical zero-knowledge. At a high level, the zero-knowledge property of the compiled protocol $\langle \mathsf{P}', \mathsf{V}'\rangle$ follows from the zero-knowledge property of the underlying honest-verifier protocol $\langle \mathsf{P}, \mathsf{V}\rangle$. That is, the proofs of correctness provided at each round by the verifier force the produced transcript to follow the same distribution as in the execution with an honest verifier, which ensures that the resulting protocol also achieves zero-knowledge. We formally show that the simulator given in Figure 6 satisfies the statistical zero-knowledge requirement.

Proposition 14. *For all PPT* V^*, $x \in \Pi_Y$, *and* $k \in \mathbb{N}$, *there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\mathrm{SD}\left(\widetilde{\mathsf{S}}^{\mathsf{V}^*}(x, 1^k), (\mathsf{P}', \mathsf{V}^*)(x, 1^k)\right) \leq \mathsf{neg}(k) \ ,$$

where $\widetilde{\mathsf{S}}^{\mathsf{V}^*}$ *is the output distribution of* $\mathsf{S}^{\mathsf{V}^*}$ *conditioned on not outputting* `fail`.

We prove Proposition 14 via a series of lemmas about the capability of any malicious verifier to deviate from the honest behavior, both in the real execution and in the simulated execution. We start by showing that in Step 2 of $\langle \mathsf{P}', \mathsf{V}'\rangle$ any verifier must produce a transcript distribution that is statistically close to the transcript distribution of the honest verifier.

Lemma 15. *For all PPT* V^*, $x \in \Pi_Y$, *and* $k \in \mathbb{N}$, *there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\Pr[\langle \mathsf{P}, \mathsf{V}(r)\rangle(x) \neq \tau_t \wedge \mathtt{transcript} \neq \bot] \leq \mathsf{neg}(k) \ ,$$

where $(\mathtt{transcript}, r) \leftarrow (\mathsf{P}', \mathsf{V}^*)(x, 1^k)$, *and* $\mathtt{transcript} \neq \bot$ *denotes that all the intermediate proofs of correctness in the transcript are accepting,* τ_i *is the projection of* `transcript` *on the messages in* $\langle \mathsf{P}, \mathsf{V}\rangle$ *up to round* i *(including), and* $\langle \mathsf{P}, \mathsf{V}(r)\rangle(x)$ *denotes the transcript produced in the honest execution of* $\langle \mathsf{P}, \mathsf{V}\rangle$ *on input* x *with verifier's randomness* r.

Input: Given $x \in \Pi_{\mathrm{Y}}$ and security parameter 1^k. Let E be the extractor for $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ and let S^{V} be the honest-verifier simulator for $\langle \mathsf{P}, \mathsf{V} \rangle$. The simulator S with oracle access to V^*, denoted by $\mathsf{S}^{\mathsf{V}^*}$, proceeds as follows:

1. Sample $(\mathtt{view}, r) \leftarrow \mathsf{S}^{\mathsf{V}}(x, 1^k)$, where $\mathtt{view} = (\beta_1, \ldots \beta_t)$ and β_i is the i-th message of P in the simulated execution of $\langle \mathsf{P}, \mathsf{V} \rangle$, and r is the randomness of V.
2. Proceed with $\mathsf{V}^*(x)$ in the *coin-flipping phase* of $\langle \mathsf{P}', \mathsf{V}' \rangle$ in order to learn a commitment c. Participate as honest V_x in the proof of knowledge for the committed value in c. Denote the transcript of this proof of knowledge θ_c. If θ_c is accepting then use the knowledge extractor $\mathsf{E}^{\mathsf{V}_x}$ to extract the committed value r_{V}. If the extractor fails output $\mathtt{fail}$.
3. Send $r_{\mathsf{P}} = r \oplus r_{\mathsf{V}}$ to V^*, and proceed to the *honest-verifier protocol execution phase*. To simulate each round $1 \le i \le t$ of $\langle \mathsf{P}, \mathsf{V} \rangle$ in $\langle \mathsf{P}', \mathsf{V}' \rangle$ proceed as follows:
   (a) Denote by $\tau_{i-1} = (\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$ the transcript of $\langle \mathsf{P}, \mathsf{V} \rangle$ up to round $i-1$ (included).
   (b) Upon receiving a message α_i from V^*, engage in a proof that $(c, r_{\mathsf{P}}, \tau_{i-1}, \alpha_i) \in L_i$ as the honest verifier V_x. Denote the transcript of this proof θ_i.
   (c) If V_x on θ_i rejects then abort, otherwise send β_i to V^*.
4. Output the simulated transcript and the induced randomness r.

Fig. 6. Simulator $\mathsf{S}^{\mathsf{V}^*}$ for the compiled protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$. The simulator $\mathsf{S}^{\mathsf{V}^*}$ samples a simulated transcript for the honest-verifier protocol which it uses to provide answers to V^* in the *honest-verifier protocol execution phase*, as well as to force the prover's randomness in the *coin-flipping phase*.

Proof. For $(\mathtt{transcript}, r) \leftarrow (\mathsf{P}', \mathsf{V}^*)(x, 1^k)$, we denote by $\langle \mathsf{P}, \mathsf{V}(r) \rangle (x)_i$ the message of V at round i. We denote by α_i the message of V^* and by θ_i the transcript of the proof at round i in $\mathtt{transcript}$.

$$
\begin{aligned}
&\Pr[\langle \mathsf{P}, \mathsf{V}(r) \rangle (x) \neq \tau_t \wedge \mathtt{transcript} \neq \bot] \\
&\quad \le \Pr[\exists i \in [t] : \alpha_i \neq \langle \mathsf{P}, \mathsf{V}(r) \rangle (x)_i \wedge \theta_i \text{ is accepting }] \\
&\quad \le \sum_{i \in [t]} \Pr[\alpha_i \neq \langle \mathsf{P}, \mathsf{V}(r) \rangle (x)_i \wedge \theta_i \text{ is accepting }] \\
&\quad \le \mathsf{neg}(k) \ ,
\end{aligned}
$$

where $(\mathtt{transcript}, r) \leftarrow (\mathsf{P}', \mathsf{V}^*)(x, 1^k)$, and the last inequality follows from the soundness of $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ using the union bound. □

Lemma 16. *For all PPT V^*, $x \in \Pi_Y$, and $k \in \mathbb{N}$, there exists a negligible function $\mathsf{neg}(\cdot)$ such that*

$$\Pr[\mathsf{V}(x, \mathtt{view}; r) \neq \tau_t \wedge \mathtt{transcript} \neq \bot] \le \mathsf{neg}(k) \ ,$$

where $(\mathtt{view}, r) \leftarrow \mathsf{S}^{\mathsf{V}}(x, 1^k)$, and $\mathtt{transcript}$ is a simulated transcript produced by $\widetilde{\mathsf{S}}^{\mathsf{V}^*}(x, 1^k)$ using $(\mathtt{view}, r)$ as described in Figure 6. We use $\mathtt{transcript} \neq \bot$

to denote that all the intermediate proofs of correctness in the transcript are accepting, τ_i is the projection of $\texttt{transcript}$ *on the* $\langle \mathsf{P}, \mathsf{V} \rangle$ *messages up to round i (included), and* $\mathsf{V}(x, \texttt{view}; r)$ *denotes the transcript produced by* V *on input x with randomness r and receiving messages in* $\texttt{view}$.

Proof. We denote by $\mathsf{V}(x, \texttt{view}; r)_i$ the message of V at round i in $\langle \mathsf{P}, \mathsf{V} \rangle$, and by α_i and θ_i the message and proof of V^* in $\texttt{transcript}$ at round i. We denote by c the commitment of V^* to r_V in $\texttt{transcript}$.

$$\begin{aligned}
&\Pr\left[\mathsf{V}(x, \texttt{view}; r) \neq \tau_t \wedge \texttt{transcript} \neq \bot\right] \\
&\quad \leq \Pr\left[\mathsf{V}(x, \texttt{view}; r) \neq \tau_t \wedge \texttt{transcript} \neq \bot \wedge \mathsf{E}^{\mathsf{V}^*} \neq \texttt{fail}\right] \\
&\quad \leq \Pr\left[c \text{ is not binding }\right] + \Pr\left[\substack{\exists i \in [t] : \alpha_i \neq \mathsf{V}(x, \texttt{view}; r)_i \wedge \theta_i \text{ is accepting } \wedge \\ \mathsf{E}^{\mathsf{V}^*} \neq \texttt{fail} \wedge \exists!\, r^*, d^* : \mathsf{Open}_x(c, r^*, d^*) = \texttt{accept}}\right] \\
&\quad \leq \mathsf{neg}(k) + \sum_{i \in [t]} \Pr\left[\substack{\alpha_i \neq \mathsf{V}(x, \texttt{view}; r)_i \wedge \theta_i \text{ is accepting } \wedge \\ \mathsf{E}^{\mathsf{V}^*} \neq \texttt{fail} \wedge \exists!\, r^*, d^* : \mathsf{Open}_x(c, r^*, d^*) = \texttt{accept}}\right] \\
&\quad \leq \mathsf{neg}(k) \ ,
\end{aligned}$$

where $(\texttt{view}, r) \leftarrow \mathsf{S}^{\mathsf{V}}(x, 1^k)$, and $\texttt{transcript}$ is a simulated transcript produced by $\widetilde{\mathsf{S}}^{\mathsf{V}^*}(x, 1^k)$ using $(\texttt{view}, r)$ as described in Figure 6. □

Proof (Proposition 14). For any PPT verifier V^*, conditioned on the simulator not outputting $\texttt{fail}$, it follows from the statistical binding of Com_x together with the honest-verifier statistical zero-knowledge property provided by S^{V} that the distribution of the simulated transcript in the *coin-flipping phase* produced by $\widetilde{\mathsf{S}}^{\mathsf{V}^*}$ is statistically close to the transcript distribution of the coin-flipping phase in $\langle \mathsf{P}', \mathsf{V}^* \rangle$. In particular, the produced randomness for V^* in $\widetilde{\mathsf{S}}^{\mathsf{V}^*}$ is statistically close to uniform. From the following facts we obtain the desired:

1. From Lemma 15 it follows that only a $\mathsf{neg}(k)$ fraction of $\langle \mathsf{P}', \mathsf{V}^* \rangle$ transcripts disagree with $\langle \mathsf{P}, \mathsf{V} \rangle$ and the randomness distribution of $\langle \mathsf{P}', \mathsf{V}^* \rangle$ is uniform as in $\langle \mathsf{P}, \mathsf{V} \rangle$.
2. From Lemma 16 it follows that only a $\mathsf{neg}(k)$ fraction of transcripts produced by $\widetilde{\mathsf{S}}^{\mathsf{V}^*}$ disagree with S^{V} and the randomness distribution of $\widetilde{\mathsf{S}}^{\mathsf{V}^*}$ is statistically close to uniform, as in S^{V}.
3. The behavior of $\widetilde{\mathsf{S}}^{\mathsf{V}^*}$ in all the $\langle \mathsf{P}_x, \mathsf{V}_x \rangle$ proofs is identical to the behavior of P'.

Combining the above we obtain that for all PPT V^*, $x \in \Pi_{\mathrm{Y}}$, and $k \in \mathbb{N}$, it holds that the full transcript distribution of $\widetilde{\mathsf{S}}^{\mathsf{V}^*}(x, 1^k)$ is statistically close to the transcript distribution of $\langle \mathsf{P}', \mathsf{V}^* \rangle(x, 1^k)$. □

We complete the proof of statistical zero-knowledge by bounding the probability of $\mathsf{S}^{\mathsf{V}^*}$ outputting $\texttt{fail}$.

Proposition 17. *For all PPT* V^**,* $x \in \Pi_Y$*, and* $k \in \mathbb{N}$*, it holds that*

$$\Pr[\mathsf{S}^{\mathsf{V}^*}(x, 1^k) = \mathtt{fail}] \leq 1/2 \ .$$

Proof. Let V^* be any PPT verifier and let $x \in \Pi_{\mathrm{Y}}$ be some input. Note that $\mathsf{S}^{\mathsf{V}^*}$ fails only when V^* provides an accepting proof of knowledge θ_c of the value committed in c while the extractor fails to extract this value. That is,

$$\Pr[\mathsf{S}^{\mathsf{V}^*}(x, 1^k) = \mathtt{fail}] \leq \Pr[\mathsf{V}_x(c, \theta_c) = \mathtt{accept} \wedge \mathsf{E}^{\mathsf{V}^*}(x, c, \theta_c) = \mathtt{fail}] \ ,$$

where $(c, \theta_c) \leftarrow \mathsf{S}^{\mathsf{V}^*}(x, 1^k)$. Since $\mathsf{S}^{\mathsf{V}^*}$ behaves exactly as P' during the commitment c and the proof θ_c in Step 2 of the simulation, we can switch to $(c, \theta_c) \leftarrow \langle \mathsf{P}', \mathsf{V}^* \rangle(x, 1^k)$ and obtain the following series of inequalities:

$$\begin{aligned} &\leq \Pr[\mathsf{V}_x(c, \theta_c) = \mathtt{accept}] \cdot (1 - \Pr[\mathsf{E}^{\mathsf{V}^*}(x, c, \theta_c) \neq \mathtt{fail}]) \\ &\leq \Pr[\mathsf{V}_x(c, \theta_c) = \mathtt{accept}] \cdot (1 - \Pr[\mathsf{V}_x(c, \theta_c) = \mathtt{accept}] + \mathsf{neg}(k)) \\ &< 1/2 \ , \end{aligned}$$

where $(c, \theta_c) \leftarrow \langle \mathsf{P}', \mathsf{V}^* \rangle(x, 1^k)$. $\square$

Acknowledgements

We wish to thank Salil Vadhan and the anonymous EUROCRYPT 2018 referees for their helpful advice.

References

1. Aiello, W., Håstad, J.: Statistical zero-knowledge languages can be recognized in two rounds. J. Comput. Syst. Sci. 42(3), 327–345 (1991)
2. Bellare, M., Micali, S., Ostrovsky, R.: Perfect zero-knowledge in constant rounds. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA. pp. 482–493 (1990)
3. Bellare, M., Micali, S., Ostrovsky, R.: The (true) complexity of statistical zero knowledge. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA. pp. 494–502 (1990)
4. Damgård, I.: Interactive hashing can simplify zero-knowledge protocol design without computational assumptions (extended abstract). In: Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. pp. 100–109 (1993)
5. Damgård, I., Goldreich, O., Wigderson, A.: Hashing functions can simplify zero-knowledge protocol design(too). In: Technical Report RS94-39, BRICS, November (1994)
6. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA. pp. 218–229 (1987)

7. Goldreich, O., Sahai, A., Vadhan, S.P.: Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In: Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998. pp. 399–408 (1998)
8. Goldreich, O., Vadhan, S.P.: Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In: Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999. p. 54 (1999)
9. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. 18(1), 186–208 (1989)
10. Itoh, T., Ohta, Y., Shizuya, H.: A language-dependent cryptographic primitive. J. Cryptology 10(1), 37–50 (1997)
11. Lindell, Y.: A note on constant-round zero-knowledge proofs of knowledge. J. Cryptology 26(4), 638–654 (2013)
12. Micciancio, D., Vadhan, S.P.: Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. pp. 282–298 (2003)
13. Okamoto, T.: On relationships between statistical zero-knowledge proofs. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 649–658 (1996)
14. Ong, S.J., Vadhan, S.P.: An equivalence between zero knowledge and commitments. In: Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008. pp. 482–500 (2008)
15. Ong, S.J., Vadhan, S.P.: An equivalence between zero knowledge and commitments. In: Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008. pp. 482–500 (2008)
16. Ostrovsky, R., Venkatesan, R., Yung, M.: Interactive hashing simplifies zero-knowledge protocol design. In: Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings. pp. 267–273 (1993)
17. Sahai, A., Vadhan, S.P.: A complete problem for statistical zero knowledge. J. ACM 50(2), 196–249 (2003)
18. Vadhan, S.P.: On transformation of interactive proofs that preserve the prover's complexity. In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA. pp. 200–207 (2000)