A Framework for Non-Interactive Instance-Dependent Commitment Schemes (NIC)[*]

Bruce Kapron, Lior Malka, Venkatesh Srinivasan

Department of Computer Science
University of Victoria, BC, Canada V8W 3P6
Email:bmkapron,liorma,venkat@cs.uvic.ca

September 17, 2009

Abstract

Zero-knowledge protocols are often studied through specific problems, like GRAPH-ISOMORPHISM. In many cases this approach prevents an important level of abstraction and leads to limited results, whereas in fact the constructions apply to a wide variety of problems. We propose to address this issue with a formal framework of *non-interactive instance-dependent commitment schemes* (NIC). We define NIC in both the perfect, statistical, and computational settings, and formally characterize problems admitting NIC in all of these settings. We also prove other useful lemmas such as closure properties. Consequently, results that previously applied only to specific problems are now strengthened by our framework to apply to *classes of problems*. By providing formal yet intuitive tools, our framework facilitates the construction of zero-knowledge protocols for a wide variety of problems, in various settings, without the need to refer to a specific problem. Our results are unconditional.

Keywords: cryptography, zero knowledge, instant-dependent commitment schemes

1 Introduction

Zero-knowledge protocols enable one party (the prover) to prove an assertion to another party (the verifier), but without revealing anything to the verifier other than the truth of the assertion [20]. Intuitively, this is made possible using a cryptographic primitive called *a bit commitment scheme*. Such a scheme allows a *sender* to commit to a bit b such that the *receiver* cannot learn b from the commitment, and at the same time the sender cannot change the commitment to another value. The first property is called *hiding* and the later is called *binding*. Bit commitment schemes exist if and only if one-way functions exist [22, 30].

The key role that bit commitment schemes play in the study of zero-knowledge protocols explains why languages such as GRAPH-ISOMORPHISM appear in many zero-knowledge constructions. Intuitively, the graphs induce a primitive similar to a bit commitment scheme, and the primitive can be used to construct the protocol. That is, given graphs $\langle G_0, G_1 \rangle$, a commitment to a bit b can be computed by choosing a random permutation π and outputting $y = \pi(G_b)$, which perfectly hides b if the graphs are isomorphic, and

[*] A preliminary version of this paper appeared in Track C of ICALP 2007 [24] under the title ”A Characterization of Non-Interactive Instance-Dependent Commitment Schemes (NIC)”.

perfectly binds to b otherwise. There are numerous examples of works that utilize variants of this idea to construct zero-knowledge protocols (c.f., [38, 19, 5, 36, 5]). A sample of recent works includes the quantum zero-knowledge protocol of Watrous [41] for GRAPH-ISOMORPHISM, the linear locality zero-knowledge proof of Micali and Pass [27] for GRAPH-NONISOMORPHISM, the concurrent zero-knowledge protocol of Micciancio et al. [28] (based on [34]) for a variant of STATISTICAL-DISTANCE [35] called $\overline{\mathrm{SD}^{1}_{1/2}}$, and the efficient-prover zero-knowledge protocol of Micciancio and Vadhan [29] for lattice problems.

The issue that we address in this paper is that many works (such as these mentioned above) construct zero-knowledge protocols for *specific problems*. For example, the quantum zero-knowledge protocol of Watrous [41] is designed specifically for GRAPH-ISOMORPHISM. This is an issue because these works actually apply to a much larger class of problems, and therefore they can obtain stronger and more general results. Our goal is to provide a framework that will allow achieving these stronger and more general results. The other disadvantage in constructing zero-knowledge protocols for specific problems is that they prevent us from formally reasoning about other settings. For example, in many cases the protocol inherits its zero-knowledge property from the hiding property of the bit commitment scheme, and this can be argued formally when using bit commitment schemes, but not when using a specific problem such as GRAPH-ISOMORPHISM. Our framework will not suffer from this limitation. It will provide this level of abstraction while still being simple.

1.1 Our Results - The NIC Framework

One may suggest that instead of constructing protocols using specific problems like GRAPH-ISOMORPHISM, we should use the notions of random self-reducibility (RSR) [38, 2] or Σ-protocols [8]. The first issue with this idea is that, since these definitions are complex, they make protocol construction and analysis very cumbersome. Indeed, this may explain why most zero-knowledge works rather use the simpler and more intuitive problem GRAPH-ISOMORPHISM. The second issue is that some of the specific problems for which zero-knowledge protocols are constructed (such as variants of STATISTICAL-DISTANCE and the lattice problems of [29]) are not known to satisfy these notions.

The notion of a non-interactive commitment scheme is significantly simpler than the notions of random self-reducibility or Σ-protocols. Furthermore, commitment schemes are natural tools in the study of zero-knowledge protocols. Thus, we propose that the issue of constructing zero-knowledge protocols for specific problems be addressed using a primitive which we call a *non-interactive instance-dependent commitment scheme* (NIC). More formally, a NIC is an efficient function $f(x, b; r)$ that outputs a commitment to a bit b using randomness r, just like a non-interactive bit commitment scheme. In addition, it takes an instance x of a problem as an input, and the hiding and binding properties depend on whether x is a YES or a NO instance of the problem. The binding is perfect, and the hiding property can be either perfect, statistical, or computational (generalizing [23]).

To show that NIC can replace specific problems used in zero-knowledge constructions, we had to find a property of these problems that implies NIC, be implied by NIC, and would apply in both the computational, statistical, and perfect setting. As we mentioned earlier, random self-reducibility [38] and Σ-protocols [8] are not suitable candidates because variants of STATISTICAL-DISTANCE and the lattice problems of [29] are not known to satisfy these notions. Instead, we formalize a very simple property, which we call *V-bit protocols*. Informally, these are 3-round protocols with perfect completeness where the message of the verifier is one random bit. This property is already satisfied by all the problems we want to capture. We then prove that this notion implies NIC and vice versa.

Characterization of NIC **(informal).** A problem has a perfectly (respectively, statistically, computationally) hiding NIC if and only if it has a V-bit perfect (respectively, statistical, computational) zero-knowledge.

This theorem yields NIC for a wide variety of problems, such as Graph-Isomorphism, Quadratic-Residuousity, Discrete-Logarithm and any random self-reducible problem (RSR) [38, 2]. Problems that are neither known to be RSR nor to admit Σ-protocols [8] are also included. For example, 1MOD4 and its variants [23], the version $\overline{\mathrm{SD}_0^1}$ of Statistical-Distance (the general version is complete for statistical zero-knowledge [35]), and problems with statistically hiding NIC such as $\overline{\mathrm{SD}_{1/2}^1}$ and the lattice problems Shortest Vector Problem (SVP) and Closest Vector Problem (CVP) of [29]. Thus, any protocol using the notion of a NIC would immediately apply to a wide variety of problems (as opposed to a specific problem).

Our theorem facilitates the study of zero-knowledge protocols by providing useful abstractions. For example, consider the non-black-box zero-knowledge protocol of Barak [4] applied to Graph-Isomorphism. If we use the graphs as explained earlier, instead of using a bit commitment scheme, then we obtain a non-black-box *perfect* zero-knowledge protocol for Graph-Isomorphism.[1] However, if instead of using Graph-Isomorphism we replace the bit commitment scheme with a NIC, then we get a result that applies to a wide variety of problems, in both the perfect and the statistical setting. Moreover, since the notion of a NIC is very similar to that of a bit commitment-scheme, we can formally prove this claim by reusing the proof given in [4]. Finally, by using the NIC framework we guarantee that if more problems admitting NIC are discovered in the future, then our result would apply to these problems as well.

In the second part of this paper we extend our approach to closure results. That is, we consider works that provide closure results for specific problems such as Graph-Isomorphism, and we show that stronger and more general results can be achieved by using the notion of a NIC. For example, Itoh et al. [23] constructed what we call a perfectly hiding NIC for the OR and the AND variants of Graph-Isomorphism. Instead, we show that all the problems admitting NIC can be combined in any monotone boolean formula fashion.

Closure of NIC **(informal).** The class of problems possessing NIC is closed under *arbitrary* (as opposed to fixed) monotone boolean formulae.

Hence, any protocol using a NIC would automatically apply to any monotone boolean formula over all the problems admitting a NIC. The formula can combine problems with different hiding properties (e.g., perfect and statistical), and can be chosen after the input is given to the parties (as opposed to fixing the protocol for one formula). Our result is also stronger and more general than that of De Santis et. al [36]. They showed a zero-knowledge protocol for monotone boolean formula statements over random self-reducible (RSR) problems, whereas our closure result includes problems that are not known to be RSR and problems where the hiding property is statistical or computational. The literature offers a variety of related closure results (c.f., [10, 11, 9]), but these apply to more restricted classes of problems and provide constructions that do not preserve the properties of the building blocks.

1.2 Related Work

The idea of a commitment scheme whose hiding and binding properties depend on an instance were implicit in [5, 36]. Itoh, Ohta, and Shizuya [23] were the first to formalize this notion and show that such schemes

[1] This is the first evidence that *perfect* (as opposed to *computational*) non-black-box zero-knowledge protocols are possible, assuming only the existence of collision-resistant hash functions (as in [4]).

can replace the bit commitment scheme in the protocols of [6, 19] for **NP**. We extend their definition from the perfect setting to the statistical and the computational settings. We remark that a similar notion to our statistically hiding NIC (using a variant of STATISTICAL-DISTANCE) appeared in the work of Micciancio and Vadhan [29] and used in [28], but it is not suitable because it cannot be generalized to the computational setting, and even in the perfect setting there are technical issues that prevent it from representing problems that have NIC, like GRAPH-ISOMORPHISM.

One direction in our characterization result shows that NIC imply what we call *V-bit* zero-knowledge protocols. This is proved by applying the technique of [23] from perfect zero-knowledge to all settings (perfect, statistical, and computational). In the other direction, we adapt the technique of Damgård [12]. This technique was originally used in the context of Σ-protocols that are also proofs of knowledge for **NP**-hard relations. We show that it also applies in the case of V-bit zero-knowledge protocols.

Our proof that problems admitting NIC are closed under *arbitrary* (as opposed to *fixed*) monotone boolean formulae uses the technique of De Santis et. al [36], who used it in the context of random self-reducible (RSR) problems, where the hiding property is perfect and therefore the construction can be applied many times without leaking any information. In our case, on the other hand, the hiding property can be computational, statistical, or perfect, and we allow mixing NIC with different hiding properties. This introduces several technical difficulties in the reductions, which we overcome by inductively summing up the leakage and relating the size of the formula to the size of the input. Related results were given in [10, 11, 9], but whereas we show how to combine NIC such that the resulting construction is also a NIC, they show how to combine protocols such that the resulting construction does not belong to the same class of protocols.

The relationship between commitment schemes and zero-knowledge protocols is one of the long standing open questions in cryptography. On one hand, the existence of commitment schemes implies that of zero-knowledge proofs (e.g., [6, 19]). On the other hand, Ostrovsky and Wigderson [33, 31] showed that commitment schemes can be constructed from zero-knowledge proofs for hard on average problems (an alternative proof was given in [35, 39, 40]). More recently, Ong and Vadhan [32] showed that a problem has a *constant-round* instance-dependent commitment scheme if and only if it has a zero-knowledge proof, but this equivalence does not apply to perfect zero-knowledge (**PZK**) proofs. Malka [26] showed an alternative equivalence that applies to all settings, including **PZK**, but this equivalence does not have useful properties, such as being constant-round.

1.3 Organization.

Section 2 gives standard definitions and Section 3 defines NIC. In Section 4 we prove that V-bit zero-knowledge protocols imply and are implied by NIC. Section 5 shows that random self-reducibility implies NIC, and Section 6 shows that NIC can be combined in monotone boolean formula fashion. Open questions are given in Section 7. In Section A we prove that the notions of V-bit zero-knowledge proofs and Σ-protocols are equivalent.

2 Definitions

2.1 Indistinguishability

The notion of zero-knowledge is based on indistinguishability of ensembles, which we now define. Let I be a countable set of strings x, and let $|x|$ denote the length of x. A sequence $\{Y_x\}_{x \in I}$ of random variables is called a *probability ensemble*. Indistinguishability is defined in terms of distance between ensembles. A function $f(n)$ is *negligible* on I if for any polynomial p there is N such that for all $x \in I$ of length at least N it holds that $f(|x|) < 1/p(|x|)$. When I is clear from the context we simply say that $f(n)$ is *negligible*.

Given a circuit $D : \{0,1\}^* \to \{0,1\}$ and two distributions Y_x and Z_x we define the *advantage* of D to distinguish Y_x from Z_x to be the function

$$\mathtt{adv}(D, Y_x, Z_x) \overset{\text{def}}{=} |\Pr[D(Y_x) = 1] - \Pr[D(Z_x) = 1]|,$$

where $\Pr[D(X) = 1]$ denotes the probability that D outputs 1 on input an element chosen according to a distribution X. Notice that if D takes randomness r as an input, then the probability is also over the uniform distribution on r. We say that D is a polynomial-size circuit if there is a polynomial p such that on inputs of length n the size of D is at most $p(|n|)$.

We also need a definition of advantage in the stronger, information theoretic sense. Let X and Y be two discrete distributions. The *statistical distance* between X and Y is defined as

$$\Delta(X, Y) \overset{\text{def}}{=} 1/2 \cdot \sum_{\alpha} |\Pr[X = \alpha] - \Pr[Y = \alpha]|.$$

Definition 2.1 (Indistinguishability) *Let $\{Y_x\}_{x\in I}$ and $\{Z_x\}_{x\in I}$ be probability ensembles. We say that $\{Y_x\}_{x\in I}$ and $\{Z_x\}_{x\in I}$ are* statistically identical *(respectively,* statistically indistinguishable*) if $\Delta(Y_x, Z_x)$ is identically 0 (respectively, negligible) on I. We say that $\{Y_x\}_{x\in I}$ and $\{Z_x\}_{x\in I}$ are* computationally indistinguishable *if $\mathtt{adv}(D, Y_x, Z_x)$ is negligible on I for all non-uniform, polynomial-size circuits D.*

The notion of a NIC is related to the promise problem STATISTICAL-DISTANCE [35]. Formally, $\mathrm{SD}^{\alpha,\beta}$ is the pair $(\mathrm{SD}^{\alpha}_{Y}, \mathrm{SD}^{\beta}_{N})$, where $\mathrm{SD}^{\alpha}_{Y} \overset{\text{def}}{=} \{(X,Y) | \Delta(X,Y) \geq \alpha\}$, $\mathrm{SD}^{\beta}_{N} = \{(X,Y) | \Delta(X,Y) \leq \beta\}$, and $\langle X, Y\rangle$ is a pair of circuits viewed as distributions (under the convention that the input to the circuit is uniformly distributed). In this paper we only consider the **NP** problem $\overline{\mathrm{SD}^{1,\beta}}$ for $\beta \leq 1/2$ (the problem $\mathrm{SD} \overset{\text{def}}{=} \overline{\mathrm{SD}^{2/3,1/3}}$, called STATISTICAL-DISTANCE, is complete for **SZK**).

2.2 Interactive Protocols and Zero-Knowledge

We use the standard definitions of interactive protocols, proofs, and zero-knowledge. We start with the notion of a protocol, which is a pair of parties communicating with each other.

Definition 2.2 (Interactive Protocols) *An interactive protocol is a pair $\langle P, V\rangle$ of functions. The* interaction *between P and V on common input x is the following random process.*

1. *Let r_P and r_V be random inputs to P and V, respectively.*
2. *repeat the following for $i = 1, 2, \ldots$*
   (a) *If i is odd, let $m_i = P(x, z, m_1, \ldots, m_{i-1}; r_P)$.*
   (b) *If i is even, let $m_i = V(x, z, m_1, \ldots, m_{i-1}; r_V)$.*
   (c) *If $m_i \in \{\mathtt{accept}, \mathtt{reject}, \mathtt{fail}\}$, then exit loop.*

Each interaction yields a transcript *$\langle x, z, m_1, \ldots, m_p; r_V\rangle$, and the strings m_i are called* messages*. The probability space containing all the transcripts is called* the view of V on x*, and is denoted $\langle P, V\rangle(x)$. We say that V* accepts x *if $m_i = \mathtt{accept}$ for an even i.*

We say that $\langle P, V\rangle$ is public coin *if V always sends independent portions of r_V, and its last message is a deterministic function of the messages exchanged. We say that $\langle P, V\rangle$ is* constant round *if there is c such that the number of messages exchanged in any interaction is at most c.*

We define interactive proofs, and remark that we consider complexity classes of *promise problems* [14] (or *problems* for short), as opposed to languages. A *problem* Π is a pair $\langle \Pi_Y, \Pi_N \rangle$ of disjoint sets, and the complement of Π is $\overline{\Pi} \stackrel{\text{def}}{=} \langle \Pi_N, \Pi_Y \rangle$. The set Π_Y contains `YES` *instances*, and the set Π_N contains `NO` *instances*. Notice that a language L can be defined as $\langle L, \overline{L} \rangle$. Thus, using the notion of problems makes our results more general.

Informally, a problem has an interactive proof if it has an interactive protocol with a polynomial-time verifier that accepts `YES` instances, and rejects `NO` instances.

Definition 2.3 (Interactive proofs and arguments) *Let $\Pi = \langle \Pi_Y, \Pi_N \rangle$ be a problem, and let $\langle P, V \rangle$ be an interactive protocol. We say that $\langle P, V \rangle$ is an* interactive proof *for Π if there is a, and $c(n), s(n) : \mathbb{N} \rightarrow [0, 1]$ such that $1 - c(n) > s(n) + 1/n^a$ for any n, and the following conditions hold.*

- *Efficiency: V is a probabilistic Turing machine whose running time over the entire interaction is polynomial in $|x|$ (this implies that the number of messages exchanged is polynomial in $|x|$).*
- *Completeness: if $x \in \Pi_Y$, then V accepts in $\langle P, V \rangle(x)$ with probability at least $1 - c(|x|)$. The probability is over r_P and r_V (the randomness for P and V, respectively).*
- *Soundness: if $x \in \Pi_N$, then for any function P^* it holds that V accepts in $\langle P^*, V \rangle(x)$ with probability at most $s(|x|)$. The probability is over the randomness r_V for V.*

If the soundness condition holds with respect to non-uniform polynomial-size circuits, then we say that $\langle P, V \rangle$ is an interactive argument *for* Π.

The function c is the completeness error*, and the function s is the* soundness error*. We say that $\langle P, V \rangle$ has* perfect completeness *(respectively,* perfect soundness*) if $c \equiv 0$ (respectively, $s \equiv 0$).*

We denote by **IP** the class of problems admitting interactive-proofs [20], and by **AM** the class of problems admitting *public-coin, constant-round* interactive-proofs [3, 25].

Definition 2.4 (Efficient prover) *Let $\langle P, V \rangle$ be an interactive proof or argument for an **NP** problem $\Pi = \langle \Pi_Y, \Pi_N \rangle$. We say that P is an* efficient prover *if on common input $x \in \Pi_Y$ it runs in time polynomial in $|x|$ given an arbitrary **NP** witness w for x.*

Finally, an interactive proof (or an interactive argument) is zero-knowledge if there is a simulator such that the view of the verifier and the output of the simulator are indistinguishable. In the following definition the simulator is not allowed to fail so that we can work with the perfect setting (in the statistical and the computational settings the simulator is allowed to fail). We use S^{V^*} to denote a Turing machine S with oracle access to Turing machine V^*.

Definition 2.5 (Zero-knowledge protocols) *A protocol $\langle P, V \rangle$ for a problem $\Pi = \langle \Pi_Y, \Pi_N \rangle$ is* perfect *(respectively,* statistical, computational*)* zero-knowledge *if there is a probabilistic, polynomial-time Turing machine S, called* the simulator*, such that*

$$\{\langle P, V^* \rangle(x)\}_{x \in \Pi_Y} \quad \text{and} \quad \{S^{V^*}(x)\}_{x \in \Pi_Y}$$

*are statistically-identical (respectively, statistically indistinguishable, computationally indistinguishable.) The class of problems having perfect (respectively, statistical, computational) zero-knowledge protocols is denoted **PZK** (respectively, **SZK**, **CZK**.) When the above ensembles are indistinguishable for $V^* = V$ we say that $\langle P, V \rangle$ is* honest-verifier, perfect *(respectively,* statistical, computational*)* zero-knowledge*, and we denote the respective classes by **HVPZK**,**HVSZK**, and **HVCZK**.*

3 Non-interactive, Instance-Dependent Commitment-Schemes (NIC)

In this section we define *non-interactive instance-dependent commitment schemes* (NIC). Notice that our definition applies to all settings (perfect, statistical, and computational) and that it can be used for studying problems admitting NIC as well as problems whose complement admits a NIC.

As a warm up, we start with the familiar notion of a *non-interactive bit commitment scheme*. Intuitively, such a scheme allows a sender to commit to a bit b such that the receiver cannot learn the value of b, yet the sender cannot change b. More precisely, the scheme is an efficient function $f(b;r)$, and to commit to b the sender chooses randomness r, computes $y = f(b;r)$, and sends y to the receiver. This is the *commit phase*. In the *reveal phase* the sender sends b and r to the receiver, who computes $f(b;r)$, thus confirming that y is indeed a commitment to b. The receiver does not send anything (hence the term *non-interactive*). The scheme is *hiding* if b cannot be determined from y, and *binding* if y binds the sender to b (that is, $f(0;r) \neq f(1;r')$ for any $r \neq r'$).

Intuitively, a NIC for a problem Π is a non-interactive commitment scheme where the hiding and the binding properties depend on instances of Π, and may not hold simultaneously. That is, instead of $f(b;r)$ we consider $f(x,b;r)$, and the hiding and binding properties depend on whether x is a `YES` or a `NO` instance of Π. NIC are attractive for many reasons. Firstly, since they are simple and non-interactive, they can be used in various settings. Secondly, unlike bit commitment schemes, they can be constructed without any assumptions, thus facilitating an unconditional study of zero-knowledge protocols. Finally, the hiding and binding properties can be statistical (because, unlike in bit commitment schemes, these properties are not required to hold simultaneously). This makes NIC suitable for the study of proofs in the statistical and perfect settings. The following definition extends the *positively opaque and negatively transparent* scheme of [23] to the statistical and the computational settings.

Definition 3.1 (NIC) *Let* $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \rangle$ *be a promise-problem, and let* $f(x,b;r)$ *be a probabilistic Turing machine running in time polynomial in* $|x|$*. The inputs to* f *are a string* x *(denoting an instance of* Π*), a bit* b*, and a string* r *(denoting the randomness of* f*).*

We say that f *is* **binding** *on* Π_{N} *if for any* $x \in \Pi_{\mathrm{N}}$*, and for any* r *and* r' *it holds that* $f(x,0;r) \neq f(x,1;r')$*. We say that* f *is* perfectly *(respectively,* statistically, computationally*)* **hiding** *on* Π_{Y} *if the ensembles* $\{f(x,0)\}_{x\in\Pi_{\mathrm{Y}}}$ *and* $\{f(x,1)\}_{x\in\Pi_{\mathrm{Y}}}$ *are statistically identical (respectively, statistically indistinguishable, computationally indistinguishable), where* $f(x,b)$ *is a random variable obtained by uniformly choosing* r*, and outputting* $f(x,b;r)$*.*

We say that f *is a* perfectly *(respectively,* statistically, computationally*)* hiding NIC *for* Π *if* f *is binding on* Π_{N}*, and perfectly (respectively, statistically, computationally) hiding on* Π_{Y}*.*

Perfectly and statistically hiding NIC are different from computationally hiding NIC. Firstly, in a perfectly or a statistically hiding NIC the hiding and the binding properties cannot hold at the same time, whereas in a computationally hiding NIC they may [23, 17]. Secondly, if Π has a perfectly or a statistically hiding NIC f, then as a class of problems **NP** contains Π. This is so because if $x \in \Pi_{\mathrm{Y}}$, then there is a pair $\langle r, r' \rangle$ such that $f(x,0;r) = f(x,1;r)$, and if $x \in \Pi_{\mathrm{N}}$, then no such pair exists. However, Π may not be in **NP** if f is computationally hiding. Finally, as was observed by [23], if a problem has a statistically hiding NIC, then it cannot be **NP**-complete, unless the polynomial hierarchy collapses [16, 1, 7]. We give an example of a NIC.

Example 3.2 *A* NIC *for the language* GRAPH-ISOMORPHISM *[5, 23]. Let* $f(x,b;r)$ *be a function that given a pair of graphs* $x = \langle G_0, G_1 \rangle$ *on* n *vertices uses* r *to define a random permutation* π *over* $\{1,\ldots,n\}$*,* *and outputs $y = \pi(G_b)$. If the graphs are isomorphic, then y is isomorphic to both G_0 and G_1, and b cannot be determined from y. Conversely, if the graphs are not isomorphic, then y cannot be isomorphic to both G_0 and G_1. Thus, f is a perfectly hiding* NIC *for* GRAPH-ISOMORPHISM.

Another example is the statistically hiding NIC for $\overline{\mathrm{SD}^{1,1/2}}$ [29]. Instances of this problem are pairs of circuits $\langle X_0, X_1\rangle$, treated as distributions (under the convention that the input to the circuit is uniformly distributed). The statistical distance between X_0 and X_1 is $1/2$ for YES instances, and 1 for NO instances. Notice that statistical distance of 1 means that $X_0(r) \neq X_1(r')$ for any r and r'. Also, using techniques that manipulate distributions, the statistical distance between X_0 and X_1 can be reduced from $1/2$ to $1/2^k$ [35, 29]. Hence, $\overline{\mathrm{SD}^{1,1/2}}$ defines a statistically hiding NIC: to commit to b we uniformly choose r and output $X_b(r)$. Notice that if f is a perfectly hiding NIC, then $\langle f(x,0), f(x,1)\rangle$ is a pair of circuits with statistical distance 0 when x is a YES instance, and statistical distance 1 when x is a NO instance. Thus, another way to look at our main result is that $\overline{\mathrm{SD}^{1,0}}$ is complete for the class of problems admitting perfectly hiding NIC (equivalently, the class of problems admitting V-bit perfect zero-knowledge proofs). However, since the NIC for GRAPH-ISOMORPHISM uses randomness drawn from a set of size $n!$ (the set of all permutations on graphs with n vertices), unless $n!$ is a power of 2, this randomness cannot be represented by a bit string. In other words, GRAPH-ISOMORPHISM is not known to be reducible to $\overline{\mathrm{SD}^{1,0}}$

4 Characterizing NIC as V-bit Zero-Knowledge Protocols

To characterize the problems admitting NIC, we had to find a definition that would imply NIC, be implied by NIC, and apply to both the computational, statistical, and perfect setting. The definition of random self-reducibility [38] is not suitable because the variants of STATISTICAL-DISTANCE and the lattice problems of [29] are not known to satisfy it. Similarly, these problems are not known to admit Σ-protocols [8], and therefore this notion is not suitable either. Instead, we formalize a natural notion, which we term *V-bit protocols*. Informally, these are 3-round public-coin protocols with perfect completeness, where the message of the verifier (i.e., the second message) is one random bit.[2] We then prove that this simple definition implies NIC and vice versa. We only consider proofs, but our result also applies to arguments, in which case it yields NIC where the binding property holds with respect to computationally bounded senders.

Theorem 4.1 *A promise-problem Π has a perfectly (respectively, statistically) hiding* NIC *if and only if Π has a* V*-bit* **PZK** *(respectively,* **SZK***) proof. Similarly, $\Pi \in$* ***NP*** *and Π has a computationally hiding* NIC *if and only if Π has a* V*-bit* **CZK** *proof.*

This theorem shows that there is a tight relationship between natural types of zero-knowledge protocols and commitment schemes. The notion of a V-bit protocol can be demonstrated using the protocol for GRAPH-ISOMORPHISM [19], or the protocols of [6, 19] for **NP**. These protocols are public-coin, they have perfect completeness, and they admit the following structure: the prover sends the first message m_1, the verifier sends back a random bit b, the prover replies with a message m_2, and the verifier accepts or rejects. Since V sends only one bit, we call these protocols V*-bit protocols*. Formally,

Definition 4.2 (V-bit protocol) *Let $\langle P, V\rangle$ be a proof or an argument for a problem $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}\rangle$. We say that $\langle P, V\rangle$ is* V-bit *if for any $x \in \Pi_Y$ the interaction between P and V is as follows: P sends m_1 to*

[2] Notice that, unlike Σ-protocols, V-bit protocols make no reference to zero-knowledge, **NP**-relations, or special soundness.

V, and V replies with a uniformly chosen bit b. P replies by sending m_2 to V, and V accepts or rejects x based on $\langle x, m_1, b, m_2\rangle$. If $x \in \Pi_Y$, then V always accepts.

We have mentioned more than once that variants of STATISTICAL-DISTANCE and certain lattice problems admit V-bit protocols [29, 35], but are not known to admit Σ-protocols. Thus, we do not know if any V-bit zero-knowledge protocol is also a Σ-protocol. However, from our characterization result it will follow that a problem admits a V-bit zero-knowledge proof if and only if it admits a Σ-protocol. Thus, all the consequences that follow from the NIC framework immediately extend to Σ-protocols. The proof for the following lemma can be found in Appendix A.

Lemma 4.3 *A problem Π has a V-bit **HVPZK** (respectively, **HVSZK**, **HVCZK**) proof if and only if Π has a perfect (respectively, statistical, computational) Σ-protocol.*

4.1 From NIC to V-bit Zero-Knowledge Protocols

We show that if a problem has a NIC, then it has a V-bit zero-knowledge protocol. Following [23], our construction plugs the NIC into the zero-knowledge protocols of [6, 19] for **NP**. Combined with our lemma from the next section (which proves that if a problem has a V-bit zero-knowledge proof, then it has a NIC), our lemma yields a compiler that transforms any V-bit, zero-knowledge proof (i.e., honest-verifier, inefficient prover) into a *malicious verifier* V-bit zero-knowledge *proof of knowledge* with *an efficient prover*. The idea is to extract the NIC for the V-bit zero-knowledge protocol and use it in the V-bit protocol of Blum [6], which has an efficient prover, and is zero-knowledge against malicious verifiers.

Lemma 4.4 *If a problem Π has a perfectly (respectively, statistically) hiding* NIC*, then Π has a public-coin* **PZK** *(respectively,* **SZK***) proof with an efficient prover. If $\Pi \in$* **NP***, and Π has a computationally hiding* NIC*, then Π has a public-coin* **CZK** *proof with an efficient prover.*

Proof:(sketch) We use the zero-knowledge protocol of [6] for the **NP**-complete problem HAMILTONIAN-CIRUIT (HC). Specifically, given input $x \in \Pi_Y \cup \Pi_N$, the prover and the verifier initially reduce x to an instance G of HC, and then execute the protocol of [6] using the NIC f for Π as a bit commitment scheme. Notice that the prover can transform its witness for x into a witness for G, and thus it is efficient. Now, if $x \in \Pi_{\mathrm{Y}}$, then G has a Hamiltonian circuit, and thus the verifier accepts. Also, since f is hiding, the protocol inherits its zero-knowledge property from the hiding property of the scheme. Thus, the protocol has an efficient prover, it is zero-knowledge with respect to cheating verifiers, and it is V-bit because it has perfect completeness. If $x \in \Pi_{\mathrm{N}}$, then G does not have a Hamiltonian circuit, and the scheme is binding, which implies that the verifier rejects with probability $1/2$ over its random coin. Thus, the protocol is sound. □

4.2 From V-bit Zero-Knowledge Protocols to NIC

In this section we show how to construct a NIC from a simulator S for any V-bit zero-knowledge protocol $\langle P, V\rangle$. We adapt the technique of Damgård [12] for Σ-protocols and apply it to V-bit protocols (see Feige and Shamir [15] for a similar technique). The differences are that [12] constructed an interactive commitment scheme from a proof of knowledge for any **NP**-hard relation, provided that the proof is a Σ-protocol. We, on the other hand, consider regular zero-knowledge proofs (as opposed to proofs of knowledge for **NP**-hard relations) and construct a *non-interactive* (as opposed to interactive) *instance-dependent* commitment scheme (as opposed to a bit commitment). Furthermore, the binding property of our NIC is statistical and follows from the soundness of the underlying V-bit protocol, whereas in [12] the binding property is computational and follows from the hardness of the underlying problem.

We start with the following idea: to commit to a bit b, execute $S(x)$ using randomness r, obtain a transcript $\langle m_1, b', m_2\rangle$ such that $b = b'$ and V accepts, and output m_1 as a commitment. Let us verify that this NIC is hiding on YES instances and binding on NO instances. If x is a YES instance, then the perfect completeness property guarantees that we always obtain transcripts where V accepts, and since b cannot be determined from such m_1, the commitment is hiding. Conversely, by the soundness property, if x is a NO instance, then there are no transcripts $\langle m_1, 0, m_2\rangle$ and $\langle m_1, 1, m_2'\rangle$ such that V accepts in both. However, the issue with this idea is that b' may not be equal to b. To overcome this issue we redefine the commitment to be $\langle m_1, b' \oplus b\rangle$. That is, we execute $S(x)$, obtain $\langle m_1, b', m_2\rangle$, and output $\langle m_1, b' \oplus b\rangle$. Intuitively, since b' is hidden, the bit $b' \oplus b$ is also hidden. Our lemma follows.

Lemma 4.5 *Let $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}\rangle$ be a promise-problem. If Π has a* V*-bit, public-coin* **HVPZK** *(respectively,* **HVSZK**, **HVCZK***) proof, then Π has a* NIC *that is perfectly (respectively, statistically, computationally) hiding on Π_{Y} and perfectly binding on Π_{N}.*

Proof: Fix a public-coin V-bit **HVPZK** (respectively, **HVSZK**, **HVCZK**) proof $\langle P, V\rangle$ for Π. We assume that $\langle P, V\rangle$ has a simulator S that outputs either `fail`, or transcripts in which V accepts. Using S we define a NIC f for Π as follows. Let $f(x, b; r)$ be the function that executes $S(x)$ with randomness r. If f obtains a transcript $\langle x, m_1', b', m_2'\rangle$ such that $V(x, m_1', b', m_2') = \mathtt{accept}$, then f outputs $\langle m_1', b' \oplus b\rangle$. Otherwise, f outputs b.

We show that f is binding on Π_{N}. Let $x \in \Pi_{\mathrm{N}}$. Notice that for any r and b it holds that $f(x, b; r)$ outputs one bit if and only if $f(x, b; r) = b$. Thus, if f outputs one bit, then there are no r and r' such that $f(x, 0; r) = f(x, 1; r')$. For the case where $f(x, b; r)$ outputs a pair $\langle \tilde{m}_1, \tilde{b}\rangle$, recall that $\tilde{b} = b' \oplus b$, where b' is taken from some transcript $\langle x, m_1', b', m_2'\rangle$. Thus, by the definition of f, for any $\tilde{m}_1, \tilde{b}, r$ and r' it holds that $f(x, 0; r) = f(x, 1; r') = \langle \tilde{m}_1, \tilde{b}\rangle$ if and only if there are m_2 and m_2' and such that $V(x, \tilde{m}_1, 0, m_2) = V(x, \tilde{m}_1, 1, m_2') = \mathtt{accept}$. However, $\langle P, V\rangle$ is public coin, and by the soundness property of $\langle P, V\rangle$ there are no m_1, m_2 and m_2' such that $V(x, m_1, 0, m_2) = V(x, m_1, 1, m_2') = \mathtt{accept}$. Hence, if f does not output one bit, then there are no r and r' such that $f(x, 0; r) = f(x, 1; r')$. We conclude that f is perfectly binding on Π_{N}.

The rest of the proof shows that f is hiding on Π_{Y}. We start with the statistical setting. To show that f is statistically hiding we need to calculate the statistical distance between commitments to 0 and commitments to 1 over $x \in \Pi_{\mathrm{Y}}$. The following probabilities are over the randomness r for f.

$$\begin{aligned}
\Delta(f(x,0), f(x,1)) &= \frac{1}{2}\sum_{\alpha} |\Pr[f(x,0) = \alpha] - \Pr[f(x,1) = \alpha]| \\
&= \frac{1}{2}\sum_{m_1} |\Pr[f(x,0) = \langle m_1, 0\rangle] - \Pr[f(x,1) = \langle m_1, 0\rangle]| + \\
&\quad \frac{1}{2}\sum_{m_1} |\Pr[f(x,0) = \langle m_1, 1\rangle] - \Pr[f(x,1) = \langle m_1, 1\rangle]| + \\
&\quad \frac{1}{2}\sum_{b\in\{0,1\}} |\Pr[f(x,0) = b] - \Pr[f(x,1) = b]| \, .
\end{aligned}$$

Notice that the third sum (i.e., the sum over b) equals $\Pr[S(x) = \mathtt{fail}]$, the probability that S fails. Now, by Definition 2.5 of zero-knowledge, when S is a **HVPZK** simulator it never fails. Thus, $\Pr[S(x) =$

$\texttt{fail}] = 0$. It remains to deal with the sums over m_1. We show that the first sum is upper bounded by $\Delta(\langle P, V\rangle(x), S(x)) - \Pr[S(x) = \texttt{fail}]/2$, and since a symmetric argument applies to the second sum, the total will be upper bounded by $2 \cdot \Delta(\langle P, V\rangle(x), S(x))$. The following probabilities for $\langle P, V\rangle(x)$ and $S(x)$ are over the randomness to P, V and S, respectively.

$$\begin{aligned}
\tfrac{1}{2}\textstyle\sum_{m_1} \quad & |\Pr[f(x,0) = \langle m_1, 0\rangle] - \Pr[f(x,1) = \langle m_1, 0\rangle]| = \\
\tfrac{1}{2}\textstyle\sum_{m_1} \quad & |\sum_{m_2} \Pr[S(x) = \langle m_1, 0, m_2\rangle] - \sum_{m_2} \Pr[S(x) = \langle m_1, 1, m_2\rangle]| = \\
\tfrac{1}{2}\textstyle\sum_{m_1} \quad & |\sum_{m_2} \Pr[S(x) = \langle m_1, 0, m_2\rangle] - \sum_{m_2} \Pr[\langle P, V\rangle(x) = \langle m_1, 0, m_2\rangle] \\
& -(\sum_{m_2} \Pr[S(x) = \langle m_1, 1, m_2\rangle] - \sum_{m_2} \Pr[\langle P, V\rangle(x) = \langle m_1, 1, m_2\rangle])| \le \\
\tfrac{1}{2}\textstyle\sum_{m_1, m_2} \quad & (|\Pr[S(x) = \langle m_1, 0, m_2\rangle] - \Pr[\langle P, V\rangle(x) = \langle m_1, 0, m_2\rangle]| + \\
& |\Pr[S(x) = \langle m_1, 1, m_2\rangle] - \Pr[\langle P, V\rangle(x) = \langle m_1, 1, m_2\rangle]|) = \\
& \Delta(\langle P, V\rangle(x), S(x)) - \Pr[S(x) = \texttt{fail}]/2\,.
\end{aligned}$$

In the first equality above we used the fact that S outputs transcripts in which V accepts. In the second equality we used the fact that $\langle P, V\rangle$ is public-coin, which implies that for any m_1 the probability of choosing an element of $\langle P, V\rangle(x)$ whose prefix is $\langle m_1, 0\rangle$ equals the probability of choosing an element of $\langle P, V\rangle(x)$ whose prefix is $\langle m_1, 1\rangle$. In the last equality we used the fact that $\langle P, V\rangle(x)$ never outputs $\texttt{fail}$, whereas $S(x)$ outputs $\texttt{fail}$ with probability $\Pr[S(x) = \texttt{fail}]$. We conclude that $\Delta(f(x,0), f(x,1)) \le 2 \cdot \Delta(S(x), \langle P, V\rangle(x))$. Hence, if S is a **HVPZK** (respectively, **HVSZK**) simulator, then $\Delta(S(x), \langle P, V\rangle(x))$ is 0 for any $x \in \Pi_{\mathrm{Y}}$ (respectively, negligible on Π_{Y}), which implies that f is perfectly (respectively, statistically) hiding on Π_{Y}.

It remains to deal with the case that S is a **HVCZK** simulator. The analysis is analogues to the statistical setting, but in reverse. We define the function $f'(\cdot, b)$ just like f, except that instead of executing the simulator, f' receives a transcript $\langle m_1, b', m_2\rangle$ and outputs $\langle m_1, b' \oplus b\rangle$. Thus, $f'(S(x), b)$ and $f(x, b)$ are identically distributed for any $b \in \{0, 1\}$. Assume towards a contradiction that there is a non-uniform family D of polynomial-size circuits that distinguishes $\{f(x,0)\}_{x\in\Pi_{\mathrm{Y}}}$ and $\{f(x,1)\}_{x\in\Pi_{\mathrm{Y}}}$. Thus, D distinguishes $\{f'(S(x),0)\}_{x\in\Pi_{\mathrm{Y}}}$ and $\{f'(S(x),1)\}_{x\in\Pi_{\mathrm{Y}}}$, and the following expression is non-negligible:

$$\begin{aligned}
& |\Pr[D(f'(S(x),0)) = 1] - \Pr[D(f'(S(x),1)) = 1]| \le \\
& |\Pr[D(f'(S(x),0)) = 1] - \Pr[D(f'(\langle P, V\rangle(x),0)) = 1]| + \\
& |\Pr[D(f'(S(x),1)) = 1] - \Pr[D(f'(\langle P, V\rangle(x),1)) = 1]|\,.
\end{aligned}$$

Above we used the fact that $\langle P, V\rangle$ is V-bit, which implies that $f'(\langle P, V\rangle(x), 0)$ and $f'(\langle P, V\rangle(x), 1)$ are identically distributed for any $x \in \Pi_{\mathrm{Y}}$. It follows that there is $b \in \{0, 1\}$ such that D distinguishes $\{f'(\langle P, V\rangle, b)\}_{x\in\Pi_{\mathrm{Y}}}$ and $\{f'(S(x), b)\}_{x\in\Pi_{\mathrm{Y}}}$. This contradicts the fact that S is a **HVCZK** simulator. We conclude that f is computationally hiding on Π_{Y}. The lemma follows. □

Theorem 4.1, presented in the beginning of this section, immediately follows from Lemmas 4.4 and 4.5. Thus, we get a characterization of V-bit zero-knowledge protocols as NIC. We remark that Theorem 4.1 can be extended to arguments, in which case it yields NIC where the binding property holds with respect to computationally bounded senders. Also, it can be extended to relaxed notions of V-bit protocols (e.g., where perfect completeness or public-coins are not required), but we avoid these extensions because they require changing the definition of a NIC.

5 Random Self-Reducibility Implies NIC

We prove the folklore theorem that if a problem is random self-reducible (RSR), then it has a perfectly hiding NIC. By replacing the notion of random-self reducibility with the simpler notion of a NIC, we make protocol design and analysis simpler. Furthermore, we can include RSR problems in our closure result. This allows combinations of RSR problems with problems that are not known to be RSR (such as versions of SD, and the lattice problems of [29]). Thus, we strengthen and unify the results of [38, 36, 23] and achieve all the improvements claimed in the introduction.

The notion of random self-reducibility [2] considers a set of strings x, each associated with a polynomial-time relation R_x on pairs $\langle z, w\rangle$. Given x, there is an algorithm S that uses randomness r to sample the domain of R_x (that is, it outputs y such that $\langle y, w'\rangle \in R_x$ for some w'). The heart of this notion consists of two algorithms: A_1, which converts a witness for y into a witness for z, and A_2, which converts witness for z into a witness for y. Both A_1 and A_2 use r. Also, there is an algorithm G that generates random pairs $\langle z', w'\rangle$ from R_x. All of the algorithms are efficient. The following definition is similar to that of [36].

Definition 5.1 (A random self-reducible problem) *Let $\mathcal{N} \subset \{0,1\}^*$ be a countable set such that R_x is an* **NP***-relation for each $x \in \mathcal{N}$. The* domain *of R_x is denoted $d(R_x) \overset{def}{=} \{z | \exists w\ \langle z, w\rangle \in R_x\}$. The language $\mathrm{L} \overset{def}{=} \{\langle x, z\rangle | x \in \mathcal{N}, \exists w\ \langle z, w\rangle \in R_x\}$ is* random self-reducible (RSR) *if there are polynomial time algorithms G, A_1, A_2, and S such that $S(x, z; r) = y \in d(R_x)$ for any $x \in \mathcal{N}, z$, and r, and the following conditions hold.*

1. *If $z \in d(R_x)$, and r is uniformly distributed, then y is uniformly distributed in $d(R_x)$.*
2. *A witness for y yields a witness for z, and vice versa. That is, $\langle z, A_1(x, y, r, w')\rangle \in R_x$ for any $\langle y, w'\rangle \in R_x$, and $\langle y, A_2(x, z, r, w'')\rangle \in R_x$ for any $\langle z, w''\rangle \in R_x$.*
3. *$G(x; r) = \langle z', w'\rangle \in R_x$, and if r is uniformly distributed, then z' is uniformly distributed in $d(R_x)$, and w' is uniformly distributed in $\{w | \langle z, w\rangle \in R_x\}$.*

We prove that random self-reducible problems have a perfectly hiding NIC. Our proof uses the idea behind the construction of the subroutine in the protocol of [36] (see Section 3.3 in [36]). Given $\mathcal{N}$ and R_x as in Definition 5.1, we define the problem $\Pi^{\mathrm{L}} \overset{\mathrm{def}}{=} \langle \Pi^{\mathrm{L}}_{\mathrm{Y}}, \Pi^{\mathrm{L}}_{\mathrm{N}}\rangle$, where $\Pi^{\mathrm{L}}_{\mathrm{Y}} \overset{\mathrm{def}}{=} \{\langle x, z\rangle | x \in \mathcal{N}, \exists w\ \langle z, w\rangle \in R_x\}$, and $\Pi^{\mathrm{L}}_{\mathrm{N}} \overset{\mathrm{def}}{=} \{\langle x, z\rangle | x \in \mathcal{N}, \forall w\ \langle z, w\rangle \notin R_x\}$.

Lemma 5.2 *If L is a random self-reducible language, then Π^{L} has a perfectly hiding* NIC.

Proof: Let $\mathrm{L} \overset{\mathrm{def}}{=} \{\langle x, z\rangle | x \in \mathcal{N}, \exists w\ \langle z, w\rangle \in R_x\}$ be a random self-reducible language. Consider the algorithms S and G from Definition 5.1. Let $G'(x; r)$ be the algorithm that executes $G(x; r)$, obtains $\langle z', w'\rangle$, and outputs z'. We use S and G' to commit to 0 and 1, respectively. Formally, we define our NIC to be a probabilistic polynomial-time Turing machine $f(x, z, b; r)$ that on input $\langle x, z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{Y}} \cup \Pi^{\mathrm{L}}_{\mathrm{N}}$, bit b, and randomness r outputs $S(x, z; r)$ if $b = 0$, and $G'(x; r)$ if $b = 1$.

The efficiency of f follows from the efficiency of S and G. We show that f is perfectly hiding. By Definition 5.1, $S(x, z; r) = y$ is uniformly distributed over $d(R_x)$ if r is uniformly distributed, and $\langle x, z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{Y}}$. Similarly, $G(x; r) = \langle z', w'\rangle$, and z' is uniformly distributed over $d(R_x)$ if r is uniformly distributed and $x \in \mathcal{N}$. Since the output of f is uniformly distributed over $d(R_x)$ for any b and $\langle x, z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{Y}}$, the ensembles $\{f(x, z, 0; r)\}_{\langle x,z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{Y}}}$ and $\{f(x, z, 1; r)\}_{\langle x,z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{Y}}}$ are statistically identical, and therefore f is perfectly hiding on $\Pi^{\mathrm{L}}_{\mathrm{Y}}$.

We show that f is binding on $\Pi^{\mathrm{L}}_{\mathrm{N}}$. Let $\langle x, z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{N}}$. Assume towards a contradiction that there are r and r' such that $S(x, z; r) = f(x, z, 0; r) = f(x, z, 1; r') = G'(x; r)$. Let $y = S(x, z; r)$. By the definition of G', there is w' such that $G(x; r) = \langle G'(x; r), w'\rangle = \langle y, w'\rangle \in R_x$. By the property of A_1 from Definition 5.1, it follows that $\langle z, A_1(x, y, r, w')\rangle \in R_x$. Hence, $\langle x, z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{Y}}$, in contradiction to the choice of $\langle x, z\rangle \in \Pi^{\mathrm{L}}_{\mathrm{N}}$. Thus, f is binding on $\Pi^{\mathrm{L}}_{\mathrm{N}}$. □

Notice that in the above proof we did not use Algorithm A_2 from Definition 5.1. Neither did we use the fact that A_1 runs in polynomial time, nor did we use the witness that G outputs.

6 Closure of Problems Possessing NIC under Monotone Boolean Formulae

In this section we show that the class of problems possessing NIC is closed under *arbitrary* (as opposed to fixed) monotone boolean formulae. Such results have been traditionally proved in the perfect setting, where, intuitively, no information is leaked at any stage. This is not the case here. We are proving closure where the formula can be chosen after the protocol is fixed and our analysis applies to all settings. This makes the proofs significantly more technical.

We start with notation, and formalize our theorem in Section 6.1. Intuitively, our goal is to show that given instances $x_1, \ldots, x_n$ and a monotone boolean formula ϕ over n variables, the prover can prove to the verifier that the instances satisfy the formula. Notice that we first fix the protocol, and then choose n, $x_1, \ldots, x_n$ and ϕ. To formalize the above intuition we need the following definitions. A *boolean variable* is a variable that can only take the values 0 or 1. We say that ϕ is a monotone boolean formula if ϕ is a boolean variable, or ϕ is of the form $\phi_0 \wedge \phi_1$ or $\phi_0 \vee \phi_1$, where both ϕ_0 and ϕ_1 are monotone boolean formulae. Given a problem $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}\rangle$ and $x \in \Pi_{\mathrm{Y}} \cup \Pi_{\mathrm{N}}$, we define the *characteristic function* χ_Π of Π as follows: if $x \in \Pi_{\mathrm{Y}}$, then $\chi_\Pi(x) = 1$, and if $x \in \Pi_{\mathrm{N}}$, then $\chi_\Pi(x) = 0$. Let ϕ be a boolean formula over $a_1, \ldots, a_m$, and let $x_1, \ldots, x_n \in \Pi_{\mathrm{Y}} \cup \Pi_{\mathrm{N}}$ for some $n \geq m$. The *evaluation* of ϕ in $\vec{x} = \langle x_1, \ldots, x_n\rangle$ is denoted $\phi(\vec{x})$, and equals 1 if and only if ϕ is satisfied when a_i is assigned $\chi_\Pi(x_i)$ for each $1 \leq i \leq m$.

We say that a class C of problems is closed under *arbitrary*, monotone boolean formulae if $\Pi \in C$ implies that $\Phi(\Pi) \in C$, where $\Phi(\Pi)$ is defined as follows.

Definition 6.1 *Let* $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}\rangle$ *be a problem. The problem* $\Phi(\Pi) \overset{def}{=} \langle \Phi(\Pi)_{\mathrm{Y}}, \Phi(\Pi)_{\mathrm{N}}\rangle$ *is defined as*

$$\Phi(\Pi)_{\mathrm{Y}} \overset{def}{=} \{\langle \phi, x_1, \ldots, x_n\rangle | \phi(\chi_\Pi(x_1), \ldots, \chi_\Pi(x_n)) = 1\}$$
$$\Phi(\Pi)_{\mathrm{N}} \overset{def}{=} \{\langle \phi, x_1, \ldots, x_n\rangle | \phi(\chi_\Pi(x_1), \ldots, \chi_\Pi(x_n)) = 0\},$$

where ϕ *is a monotone boolean formula over* $a_1, \ldots, a_m$ *such that* $m \leq n$*, and* $x_i \in \Pi_{\mathrm{Y}} \cup \Pi_{\mathrm{N}}$ *for all* $1 \leq i \leq n$*. We define* $\Phi(\Pi)^k \overset{def}{=} \langle \Phi(\Pi)^{\mathrm{k}}_{\mathrm{Y}}, \Phi(\Pi)_{\mathrm{N}}\rangle$*, where* $\Phi(\Pi)^{\mathrm{k}}_{\mathrm{Y}}$ *is defined as*

$$\Phi(\Pi)^{\mathrm{k}}_{\mathrm{Y}} \overset{def}{=} \{\langle \phi, x_1, \ldots, x_n\rangle | \phi(\chi_\Pi(x_1), \ldots, \chi_\Pi(x_n)) = 1 \wedge \forall i\ |x_i|^k \geq |\phi, x_1, \ldots, x_n|\}.$$

The use of k in the definition is necessary for bounding the advantage of the adversary in the statistical and the computational settings. This issue does not arise in the considerably simpler perfect setting.

6.1 Combining NIC in a Monotone Boolean Formula Fashion

Our closure result states that if a problem Π has a NIC, then the problem $\Phi(\Pi)$ also has a NIC. Notice that the definition of $\Phi(\Pi)$ only considers instances of Π, but this was done only to simplify the presentation.

The proofs would work with instances from different problems. Consequently, we get that the class of problems possessing NIC (equivalently, V-bit zero-knowledge proofs) is closed under arbitrary, monotone boolean formulae. This is stronger than saying that the class of problems admitting NIC is closed under the AND and the OR operators.

Theorem 6.2 *For any problem Π that has a* NIC *f, and for any $k \in \mathbb{N}$, there is a* NIC *f' such that*

1. *if f is a perfectly hiding* NIC *for Π, then f' is a perfectly hiding* NIC *for $\Phi(\Pi)$.*
2. *if f is a statistically (respectively, computationally) hiding* NIC *for Π, then f' is a statistically (respectively, computationally) hiding* NIC *for $\Phi(\Pi)^{\mathrm{k}}$.*

We will prove the above theorem by constructing a new NIC for $\Phi(\Pi)$ from the NIC for Π. Compared to Σ-protocols and random self-reducibility, the advantage of this approach is that we do not need to work with involved notions such as interaction or zero-knowledge. Using the technique of [36] we construct NIC as follows. If f is a NIC for Π, then a NIC for instances of the form $z = \langle a \wedge b, x_1, x_2\rangle$ can be defined by $f'(z, b; r) = \langle f(x_1, b), f(x_2, b)\rangle$. Thus, if both x_1 and x_2 are YES instances of Π, then f' is hiding (because both $f(x_1, b)$ and $f(x_2, b)$ are hiding), and if x_1 or x_2 is a NO instance, then f' is binding (because at least one of $f(x_1, b)$ and $f(x_2, b)$ is binding). Notice that we omitted the randomness for f, but the intention is that f' uses independent randomness in each execution. A similar idea applies to the OR connector. That is, a NIC for instances of the form $z = \langle a \vee b, x_1, x_2\rangle$ can be defined by $f'(z, b; r) = \langle f(x_1, b_1), f(x_2, b_2)\rangle$, where b_1 is uniformly chosen, and b_2 is chosen such that $b_1 \oplus b_2 = b$. Thus, if at least one of x_1, x_2 is a YES instance of Π, then f' is hiding (because either $f(x_1, b_1)$ or $f(x_2, b_2)$ is hiding), and if both x_1 and x_2 are NO instances, then f' is binding (because both $f(x_1, b_1)$ and $f(x_2, b_2)$ are binding). The following construction generalizes these ideas to any monotone boolean formula.

Construction 6.3 *We define a recursive function $f'(\phi, \vec{x}, b; r)$. Let f be a* NIC, *and let $b \in \{0, 1\}$. Let ϕ be a monotone boolean formula over the variables $a_1, \ldots, a_m$, and let $\vec{x} = \langle x_1, \ldots, x_n\rangle$ be a vector of n strings, where $n \geq m$. The randomness r for f' is of length polynomial in $|\langle \phi, \vec{x}\rangle|$, and the polynomial is determined from the construction of f', described below.*

1. *If $\phi = a_i$ for some $1 \leq i \leq m$, then return $f(x_i, b, r)$.*
2. *Partition r into r_0 and r_1 (that is, the concatenation $r_0 r_1$ equals r).*
3. *If $\phi = \phi_0 \wedge \phi_1$, then return $\langle f'(\phi_0, \vec{x}, b, r_0), f'(\phi_1, \vec{x}, b, r_1)\rangle$.*
4. *If $\phi = \phi_0 \vee \phi_1$, then return $\langle f'(\phi_0, \vec{x}, b_0, r_0), f'(\phi_1, \vec{x}, b_1, r_1)\rangle$, where $b_0 \in \{0, 1\}$ is uniformly distributed, and b_1 is chosen such that $b_0 \oplus b_1 = b$.*

6.2 Proof of the Closure Result

In this section we prove Theorem 6.2 from the previous section. This theorem states that if f is a NIC for a problem Π, then f' defined from f as in Construction 6.3 is a NIC for $\Phi(\Pi)$. We use the technique of [36]. Intuitively, this technique admits a simple analysis in the perfect setting because the advantage of the adversary remains zero at every stage of construction 6.3, and therefore it sums up to zero. However, in the statistical and the computational settings the advantage is non-negligible, and the total may not be negligible. This is why we introduced the constant k in Definition 6.1, and this is why we need to provide a more involved analysis. We start with the binding property.

Lemma 6.4 *If a function f is binding on a set Π_N, then f' from Construction 6.3 is binding on $\Phi(\Pi)_N$.*

Proof: We prove the lemma by induction on the number ℓ of connectives in ϕ. For the base case, $\ell = 0$ and therefore f and f' are identical. Since f is binding on Π_N, we get that f' is binding on $\Phi(\Pi)_N$. Assume the induction hypothesis for all $\ell \geq 1$. Let ϕ be a monotone boolean formula with $\ell + 1$ connectives, and let $\langle \phi, \vec{x} \rangle \in \Phi(\Pi)_N$. Consider the case where $\phi = \phi_0 \wedge \phi_1$, and assume towards contradiction that there are r_0, r'_0 and r_1, r'_1 such that

$$f'(\phi, 0; r_0 r'_0) = \langle f'(\phi_0, 0; r_0), f'(\phi_1, 0; r'_0) \rangle = \langle f'(\phi_0, 1; r_1), f'(\phi_1, 1; r'_1) \rangle = f'(\phi, 1; r_1 r'_1).$$

Since $\langle \phi, \vec{x} \rangle \in \Phi(\Pi)_N$, we can fix $b \in \{0, 1\}$ for which $\phi_b(\vec{x}) = 0$. Hence, $f'(\phi_0, 0; r_b) = f'(\phi_0, 1; r'_b)$, and since ϕ_b has at most ℓ connectives, we get a contradiction to the induction hypothesis. The case where $\phi = \phi_0 \vee \phi_1$ is similar. Specifically, assume towards contradiction that there are r_0, r_0, r_1, r'_1 and $b_0, b'_0, b_1, b'_1 \in \{0, 1\}$ such that $b_0 \oplus b'_0 \neq b_1 \oplus b'_1$, and

$$f'(\phi, b_0 \oplus b'_0; r_0 r'_0) = \langle f'(\phi_0, b_0; r_0), f'(\phi_1, b'_0; r'_0) \rangle = \langle f'(\phi_0, b_1; r_1), f'(\phi_1, b'_1; r'_1) \rangle = f'(\phi, b_1 \oplus b'_1; r_1 r'_1).$$

Thus, there is $d \in \{0, 1\}$ such that $b_d \neq b'_d$ and $f'(\phi_0, 0; r_d) = f'(\phi_0, 1; r'_d)$. Since ϕ_d has at most ℓ connectives, we get a contradiction to the induction hypothesis. □

In the following section we prove the hiding property in the statistical setting, hence obtaining Theorem 6.2 for perfectly and statistically hiding NIC.

6.2.1 The Hiding Property in the Statistical Setting

Recall that f' outputs $\langle f(x_1, b), f(x_2, b) \rangle$ on input $z = \langle a \wedge b, x_1, x_2 \rangle \in \Phi(\Pi)$ and bit b, and that if both x_1 and x_2 are YES instances, then f' is hiding (because both $f(x_1, b)$ and $f(x_2, b)$ are hiding). This motivation works in the perfect setting, but in the statistical setting the output of $f(x_1, b)$ and $f(x_2, b)$ may not perfectly hide the bit b. Intuitively, both $f(x_1, b)$ and $f(x_2, b)$ may leak a small amount of information about b. Thus, we need to quantify this amount. We use the following lemma, which is similar to the *Direct Product Lemma* and the *XOR Lemma* from Vadhan's thesis [39]. To simplify the presentation we omit $\vec{x}$ and r from the parameters to f'. The proof is technical, and appears in Appendix B.

Lemma 6.5 *Let f' be a function, let $\vec{x}$ be a vector of strings, and let ϕ_0 and ϕ_1 be monotone boolean formula. Then,*

$$\begin{aligned} \Delta(f'(\phi_0 \wedge \phi_1, 0), f'(\phi_0 \wedge \phi_1, 1)) &\leq \Delta(f'(\phi_0, 0), f'(\phi_0, 1)) + \Delta(f'(\phi_1, 0), f'(\phi_1, 1)), \text{and} \\ \Delta(f'(\phi_0 \vee \phi_1, 0), f'(\phi_0 \vee \phi_1, 1)) &\leq \Delta(f'(\phi_0, 0), f'(\phi_0, 1)) \cdot \Delta(f'(\phi_1, 0), f'(\phi_1, 1)). \end{aligned}$$

Now we prove Theorem 6.2 in the statistical setting. The idea is to recursively apply the above lemma, and to carefully add the the amount of information leaked at each stage of Construction 6.3. For this purpose we introduce the notation of $P(\phi)$, which denotes the multiset containing all the indices of boolean variables in a formula ϕ (e.g., if $\phi = (\alpha_1 \vee \alpha_2) \wedge \alpha_1$, then $P(\phi) = \{1, 1, 2\}$).

Lemma 6.6 *If a function f is perfectly (respectively, statistically) hiding on a set Π_Y, then for any $k \in \mathbb{N}$ Construction 6.3 of f' is perfectly (respectively, statistically) hiding on $\Phi(\Pi)_Y$ (respectively, $\Phi(\Pi)^k_Y$).*

Proof: Let $k \in \mathbb{N}$. We start with the statistical setting, and the perfect setting will follow. Our goal is to show that the statistical distance between commitments to 0 and commitments to 1 is negligible. Thus, as a first step we prove that for any vector $\langle \phi, \vec{x} \rangle = \langle \phi, \langle x_1, \ldots, x_n \rangle \rangle$ it holds that

$$\Delta\big(f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)\big) \leq \sum_{i \in P(\phi)} \Delta\big(f(x_i, 0), f(x_i, 1)\big).$$

We prove the above hypothesis by induction on the number ℓ of connectives in ϕ. The base case is trivial because $\ell = 0$, and therefore f and f' are identical. Assume the induction hypothesis for all $\ell \geq 1$, and let $\langle \phi, \vec{x} \rangle = \langle \phi, x_1, \ldots, x_n \rangle \in \Phi(\Pi)^k_Y$. Notice that regardless of whether ϕ equals $\phi_0 \wedge \phi_1$ or $\phi_0 \vee \phi_1$, by Lemma 6.5 it holds that

$$\Delta(f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)) \leq \Delta(f'(\phi_0, \vec{x}, 0), f'(\phi_0, \vec{x}, 1)) + \Delta(f'(\phi_1, \vec{x}, 0), f'(\phi_1, \vec{x}, 1)).$$

Now we apply the induction hypothesis to both ϕ_0 and ϕ_1. Hence, we get that

$$\Delta(f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)) \leq \sum_{i \in P(\phi_0)} \Delta(f(x_i, 0), f(x_i, 1)) + \sum_{i \in P(\phi_1)} \Delta(f(x_i, 0), f(x_i, 1)).$$

Since $P(\phi) = P(\phi_0) \cup P(\phi_1)$, the induction follows. Notice that if f is perfectly hiding, then the above sum equals 0, and thus f' is perfectly hiding on $\Phi(\Pi)_Y$. In the statistical setting we are not done because we need to show that this sum is negligible in the length of $\langle \phi, \vec{x} \rangle$. Thus, we proceed to the next step.

Let $a \in \mathbb{N}$. Since f is statistically hiding on Π_{Y}, there is N such that $\Delta(f(x, 0), f(x, 1)) \leq 1/|x|^{ak+k}$ for any $x \in \Pi_{\mathrm{Y}}$ of length at least N. Notice that by Definition 6.1 of $\Phi(\Pi)^k$, there is N' such that for any $\langle \phi, x_1, \ldots, x_n \rangle \in \Phi(\Pi)^k_Y$ of length at least N' it holds that $|x_i| \geq N$ for each $1 \leq i \leq n$. Hence, fixing N' we are guaranteed that for any $\langle \phi, \vec{x} \rangle = \langle \phi, x_1, \ldots, x_n \rangle \in \Phi(\Pi)^k_Y$ of length of at least N' it holds that $\Delta(f(x_i, 0), f(x_i, 1)) \leq 1/|x_i|^{ak+k}$ for each $1 \leq i \leq n$, and by the fact that we proved using induction,

$$\Delta(f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)) \leq \sum_{i \in P(\phi)} \Delta(f(x_i, 0), f(x_i, 1)) \leq \sum_{i \in P(\phi)} 1/|x_i|^{ak+k}.$$

It remains to show that $\sum_{i \in P(\phi)} 1/|x_i|^{ak+k} \leq 1/|\langle \phi, \vec{x} \rangle|^a$ for any $\langle \phi, \vec{x} \rangle \in \Phi(\Pi)^k$ of length at least N'. This follows from Definition 6.1 of $\Phi(\Pi)^k$ because for any $\langle \phi, x_1, \ldots, x_n \rangle \in \Phi(\Pi)^k_Y$ and $1 \leq i \leq n$ it holds that $|\langle \phi, x_1, \ldots, x_n \rangle| \leq |x_i|^k$, which implies that for any $1 \leq i \leq n$ the total number of variables in ϕ is at most $|x_i|^k$. Hence, for any $\langle \phi, x_1, \ldots, x_n \rangle \in \Phi(\Pi)^k_Y$ there is $1 \leq j \leq n$ such that

$$\sum_{i \in P(\phi)} 1/|x_i|^{ak+k} \leq |P(\phi)| \cdot 1/|x_j|^{ak+k} \leq |x_j|^k \cdot 1/|x_j|^{ak+k} \leq 1/|\langle \phi, x_1, \ldots, x_n \rangle|^a.$$

We conclude that $\Delta(f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)) \leq 1/|\langle \phi, \vec{x} \rangle|^a$ for any $a \in N$ and sufficiently long $\langle \phi, \vec{x} \rangle \in \Phi(\Pi)^k_Y$. Thus, f' is statistically hiding on $\Phi(\Pi)^k_Y$. □

6.2.2 The Hiding Property in the Computational Setting

In the computational setting we need a lemma analogous to Lemma 6.5. Roughly speaking, we use a distinguisher D on $\phi = \phi_0 \wedge \phi_1$ to construct circuits C_0 and C_1 such that either C_0 is a distinguisher on ϕ_0 or C_1 is a distinguisher on ϕ_1. Notice that we also need to make sure that the size of C_0 and C_1 is related to the size of D, so that later, when we apply this lemma inductively, the size of the resulting distinguisher will still be polynomial. The proof is technical, and can be found in Appendix C.

Lemma 6.7 *Let f' be the function from construction 6.3, let ϕ_0 and ϕ_1 be monotone boolean formulae, and let $\vec{x}$ be a vector of strings. Given a circuit D, for each $i \in \{0,1\}$ there are circuits C_i and E_i of size at most $|D| + |f'| + |\phi_{\bar{i}}| + \sum_{j \in P(\phi_{\bar{i}})} |x_j|$ each such that*

$$\begin{aligned}&\texttt{adv}(D, f'(\phi_0 \wedge \phi_1, 0), f'(\phi_0 \wedge \phi_1, 1)) \leq \texttt{adv}(C_0, f'(\phi_0, 0), f'(\phi_0, 1)) + \texttt{adv}(C_1, f'(\phi_1, 0), f'(\phi_1, 1)),\\ &\textit{and } \texttt{adv}(D, f'(\phi_0 \vee \phi_1, 0), f'(\phi_0 \vee \phi_1, 1)) \leq \texttt{adv}(E_i, f'(\phi_i, 0), f'(\phi_i, 1)).\end{aligned}$$

Finally, we prove Theorem 6.2 in the computational setting. The proof is complicated because we start with a distinguisher D whose input contains an instance of $\Phi(\Pi)^k$, and from this distinguisher we need to construct a distinguisher C whose input is an instance of Π. To do this, we will define an infinite sequence of $x \in \Pi_Y$ from the sequence $\Phi(\Pi)^k_Y$, making sure that the size of C is polynomial in the size of D.

Lemma 6.8 *If f is a computationally hiding* NIC *on a set Π_Y, then for any $k \in \mathbb{N}$ Construction 6.3 of f' is computationally hiding on $\Phi(\Pi)^k_Y$.*

Proof: Let $k \in \mathbb{N}$. Our goal is to show that if a circuit D distinguishes commitments of formula ϕ (i.e., commitments to 0 and commitments to 1), then there is a circuit C_i that distinguishes commitments on one of the x_i, and the size of C_i is polynomial in the size of D. First, we prove that for any circuit D and any vector $\langle \phi, \vec{x} \rangle$ there are circuits C_i, each of size at most $|D| + |P(\phi)| \cdot |f'| + |\phi| + \sum_{j \in P(\phi)} |x_j|$, such that

$$\texttt{adv}\big(D, f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)\big) \leq \sum_{i \in P(\phi)} \texttt{adv}\big(C_i, f(x_i, 0), f(x_i, 1)\big).$$

We prove the above hypothesis by induction on the number ℓ of connectives in ϕ. The base case is trivial because $\ell = 0$, and ϕ is a boolean variable (i.e., $\phi = a_i$). Thus, f and f' are identical, and we can take $C_i = D$. Assume the induction hypothesis for all $\ell \geq 1$, let D be a circuit, and let $\langle \phi, \vec{x} \rangle = \langle \phi, x_1, \ldots, x_n \rangle$ be a vector. We only treat the case $\phi = \phi_0 \wedge \phi_1$ because the case $\phi_0 \vee \phi_1$ is similar. Omitting $\vec{x}$, by Lemma 6.7, there are circuits C_0 and C_1 such that

$$\texttt{adv}(D, f'(\phi_0 \wedge \phi_1, 0), f'(\phi_0 \wedge \phi_1, 1)) \leq \texttt{adv}(C_0, f'(\phi_0, 0), f'(\phi_0, 1)) + \texttt{adv}(C_1, f'(\phi_1, 0), f'(\phi_1, 1)),$$

and the size of C_0 is at most $|D| + |f'| + |\phi_1| + \sum_{j \in P(\phi_1)} |x_j|$. Thus, by the induction hypothesis for ϕ_0, there are circuits C_i^0 such that

$$\texttt{adv}\big(C_0, f'(\phi_0, \vec{x}, 0), f'(\phi_0, \vec{x}, 1)\big) \leq \sum_{i \in P(\phi_0)} \texttt{adv}\big(C_i^0, f(x_i, 0), f(x_i, 1)\big),$$

and the size of each of the circuits C_i^0 is at most $\big(|D| + |P(\phi_1)| \cdot |f'| + |\phi_1| + \sum_{j \in P(\phi_1)} |x_j|\big) + |P(\phi_0)| \cdot |f'| + |\phi_0| + \sum_{j \in P(\phi_0)} |x_j|$, which equals $|D| + |P(\phi)| \cdot |f'| + |\phi| + \sum_{j \in P(\phi)} |x_j|$. A similar argument applies to C_1. Thus, denoting the circuits corresponding to C_1 by C_i^1 we get that

$$\begin{aligned}\texttt{adv}(D, f'(\phi_0 \wedge \phi_1, 0), f'(\phi_0 \wedge \phi_1, 1)) \quad &\leq \quad \sum_{i \in P(\phi_0)} \texttt{adv}\big(C_i^0, f(x_i, 0), f(x_i, 1)\big) + \\ &\qquad \sum_{i \in P(\phi_1)} \texttt{adv}\big(C_i^1, f(x_i, 0), f(x_i, 1)\big).\end{aligned}$$

Since the size of the circuits C_i^0 and C_i^1 is as stated in hypothesis, the induction follows.

In the rest of the proof we show that the advantage is negligible in the length of $\langle \phi, \vec{x} \rangle$. Formally, we assume towards a contradiction that there is $a \in \mathbb{N}$, a polynomial-size circuit D, and an infinite sequence I of vectors $\langle \phi, \vec{x} \rangle \in \Phi(\Pi)_Y^k$ such that $\mathtt{adv}(D, f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)) \geq 1/|\langle \phi, \vec{x} \rangle|^a$ for all $\langle \phi, \vec{x} \rangle \in I$, and then we show that this contradicts the fact that f is a computationally hiding NIC on Π_Y.

Fix D, I and a. Recall that for any vector $\langle \phi, \vec{x} \rangle$ it holds that $|P(\phi)| \leq |\langle \phi, \vec{x} \rangle|$, and that the size of D and f' is polynomial in the size of $|\langle \phi, \vec{x} \rangle|$. Thus, by the fact that we proved using induction there is a polynomial p such that for each $\langle \phi, \vec{x} \rangle \in I$ there are circuits C_i of size at most $p(|\langle \phi, \vec{x} \rangle|)$ and $\sum_{i \in P(\phi)} \mathtt{adv}\big(C_i, f(x_i, 0), f(x_i, 1)\big) \geq \mathtt{adv}\big(D, f'(\phi, \vec{x}, 0), f'(\phi, \vec{x}, 1)\big) \geq 1/|\langle \phi, \vec{x} \rangle|^a$. Now, by Definition 6.1 of $\Phi(\Pi)^k$, for any $\langle \phi, \vec{x} \rangle = \langle \phi, x_1, \ldots, x_n \rangle \in \Phi(\Pi)_Y^k$ and $1 \leq j \leq n$ it holds that $|x_j|^k \geq |\langle \phi, \vec{x} \rangle|$. Thus, for each each $\langle \phi, \vec{x} \rangle \in I$ there is $1 \leq j \leq n$ and a circuit C_{x_j} of size at most $p(|x_j|^k)$ such that $\mathtt{adv}\big(C_{x_j}, f(x_j, 0), f(x_j, 1)\big) \geq |x_j|^{-k} \cdot 1/|\langle \phi, \vec{x} \rangle|^a \geq 1/|x_j|^{ak-k}$. Since there are infinitely many such x_j, we get a contradiction to the premise that f is a computationally hiding NIC on Π_Y. □

7 Open Questions

In this paper we showed a V-bit protocol for any problem that has a NIC. This protocol has soundness error $1/2$, which is inherent to public-coin black-box zero-knowledge protocols [18]. Our open question is whether the soundness error can be reduced to $1/2^n$ while maintaining a constant number of rounds. Indeed, the protocol of [5] achieves this for random-self reducible problems, but it does not seem to apply to problems admitting NIC.

Acknowledgements. We thank Oded Goldreich, Allan Scott, Warren Shenkenfelder, Salil Vadhan, and Ivan Visconti for helpful discussions on this version. Also, many thanks to the referees of Crypto 2005 and ICALP 2007 for their comments on earlier versions of this paper.

References

[1] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. of Computer and System Sciences*, 42(3):327–345, June 1991.

[2] Dana Angluin and David Lichtenstein. Provable security in cryptosystems: a survey. Technical Report 288, Department of Computer Science, Yale University, 1983.

[3] László Babai. Trading group theory for randomness. In *STOC*, pages 421–429, 1985.

[4] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001.

[5] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *22nd STOC*, pages 482–493, 1990.

[6] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the ICM,pp*, pages 1444–1451, 1986.

[7] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, 1987.

[8] Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and Uni.of Amsterdam, 1996.

[9] Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *Public Key Cryptography*, pages 354–372, 2000.

[10] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.

[11] Ivan Dåmgard and Ronald Cramer. On monotone function closure of perfect and statistical zero-knowledge, 1996.

[12] Ivan B. Damgård. On the existence of bit commitment schemes and zero-knowledge proofs. In *CRYPTO '89: Proceedings on Advances in cryptology*, pages 17–27, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[13] Ivan B. Damgård. On Σ-protocols. Available online at www.daimi.au.dk/ ivan/Sigma.pdf, 2005.

[14] Shimon Even, Alan L. Selman, and Yacov Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, May 1984.

[15] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In *CRYPTO*, pages 526–544, 1989.

[16] Lance Fortnow. The complexity of perfect zero-knowledge. In Silvio Micali, editor, *Advances in Computing Research*, volume 5, pages 327–343. JAC Press, Inc., 1989.

[17] Oded Goldreich. *Foundations of Cryptography*, volume 1. Cambridge University Press, 2001.

[18] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

[19] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[20] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[21] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In *EUROCRYPT*, pages 123–128, 1988.

[22] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[23] Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *J. Cryptology*, 10(1):37–50, 1997.

[24] Bruce Kapron, Lior Malka, and Venkatesh Srinivasan. Characterizing non-interactive instance-dependent commitment-schemes (NIC). In *34th International Colloquium on Automata, Languages and Programming (ICALP 2007)*, volume 4596 of *LNCS*, pages 328–339, 2007.

[25] Babai László and Shlomo Moran. Arthur-merlin games: A randomized proof system and a hierarchy of complexity classes. *J. of Computer and System Sciences*, 36:254–276, 1988.

[26] Lior Malka. Instance-dependent commitment schemes and the round complexity of perfect zero-knowledge proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(068), 2008.

[27] Silvio Micali and Rafael Pass. Local zero knowledge. In *STOC*, pages 306–315, 2006.

[28] Daniele Micciancio, Shien Jin Ong, Amit Sahai, and Salil P. Vadhan. Concurrent zero knowledge without complexity assumptions. In *TCC*, pages 1–20, 2006.

[29] Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO*, pages 282–298, 2003.

[30] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[31] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000.

[32] Shien Jin Ong and Salil P. Vadhan. An equivalence between zero knowledge and commitments. In *TCC*, pages 482–500, 2008.

[33] Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *ISTCS*, pages 3–17, 1993.

[34] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *FOCS*, pages 366–375, 2002.

[35] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero-knowledge. *J. ACM*, 50(2):196–249, 2003.

[36] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. On monotone formula closure of SZK. In *IEEE Symposium on Foundations of Computer Science*, pages 454–465, 1994.

[37] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[38] Martin Tompa and Heather Woll. Random self-reducibility and zero-knowledge interactive proofs of possession of information. In *28th FOCS*, pages 472–482, 1987.

[39] Salil P. Vadhan. *A study of statistical zero-knowledge proofs*. PhD thesis, MIT, 1999.

[40] Salil P. Vadhan. An unconditional study of computational zero knowledge. *SIAM J. Comput.*, 36(4):1160–1214, 2006.

[41] John Watrous. Zero-knowledge against quantum attacks. In *STOC*, pages 296–305, 2006.

A Σ-protocols

Sigma protocols play an important role in cryptography. Such protocols were given by Schnorr [37] and Guillou and Quisquater [21], and the notion of Σ-protocols was later formalized in the thesis of Cramer [8]. We recall the definition of Σ-protocols, and discuss their relationship to V-bit protocols.

Definition A.1 (Σ-protocol [13]) *Let p be a polynomial, and let R be a relation such that $|w| \leq p(|x|)$ for any $\langle x, w\rangle \in R$. An interactive protocol $\langle P, V\rangle$ is a* Σ-protocol *for R if V runs in polynomial time, and the following properties hold.*

- **Public-coin, 3-round:** *on common input x, the prover P sends a, the verifier V replies with a uniformly chosen string e, the prover sends back z, and V accepts or rejects based on $\langle x, a, e, z\rangle$.*
- **Perfect completeness:** *if there is w such that $\langle x, w\rangle \in R$, then V accepts x with probability 1 over the randomness for P and V.*
- **Special soundness:** *there is a polynomial-time Turing machine M such that for any x, if $\langle a, e, z, \texttt{accept}\rangle$ and $\langle a, e', z', \texttt{accept}\rangle$ are in $\langle P, V\rangle(x)$ and $e \neq e'$, then $M(a, e, e', z, z') = w$ and $\langle x, w\rangle \in R$.*
- **Special honest-verifier zero-knowledge:** *there is a probabilistic, polynomial-time Turing machine S, called* the simulator, *such that for any $\langle x, w\rangle \in R$ and e, the output of $S(x, e)$ is identically distributed to $\langle P, V_e\rangle(x)$, where V_e is the verifier that sends e as its random string.*

Definition 4.2 of a V-bit protocol is similar to that of a Σ-protocol in that both of them consider 3-round, public-coin protocols with perfect completeness. The difference between the notions is that V-bit protocols make no reference to relations, zero-knowledge, or special soundness.

Notice that in V-bit protocols the verifier sends only one bit, whereas in Σ-protocols the verifier sends a string e. However, as was observed by Damgård, a protocol remains Σ-protocol even if instead of sending e the verifier sends one bit b, and e is defined as b followed by zeroes [13]. Thus, if a relation R has a Σ-protocol, then R has a V-bit zero-knowledge protocol. Now we show that the opposite is also true, thus proving Lemma 4.3.

Proof of Lemma 4.3: (sketch) Let $\Pi = \langle \Pi_Y, \Pi_N\rangle$ be a problem, and let $\langle P, V\rangle$ be a V-bit zero-knowledge proof for Π. We show that Π has Σ-protocol with the same zero-knowledge property. We start with the observation that as a class of promise problems, Π is in **NP**. This is so because if $x \in \Pi_Y$, then there are prover messages m_1, m_2, m_2' such that V accepts on both transcripts $\langle x, m_1, 0, m_2\rangle$ and $\langle x, m_1, 1, m_2'\rangle$, and if $x \in \Pi_N$, then no such transcripts exist. Thus, $\langle m_1, m_2, m_2'\rangle$ is a witness for x. By our characterization result, Π has a NIC f. Thus, the protocol $\langle P', V'\rangle$ of Blum [6], where P' proves to V' that $x \in \Pi_Y$ using the witness $\langle m_1, m_2, m_2'\rangle$, is a zero-knowledge proof for Π. Notice that the resulting proof inherits its zero-knowledge property from the hiding property of the NIC, and it has perfect completeness. Since the proof is also V-bit, it follows that it satisfies the special soundness and the special honest-verifier zero-knowledge conditions. □

B Proof of Lemma 6.5

We start with the case where $\phi = \phi_0 \wedge \phi_1$. Recall that for any bit b it holds that $\Pr[f'(\phi, b) = \langle\alpha, \beta\rangle] = \Pr[f'(\phi_0, b) = \alpha] \cdot \Pr[f'(\phi_1, b) = \beta]$. Hence,

$$\begin{aligned}
&\Delta(f'(\phi,0), f'(\phi,1)) = \\
&\sum_{\alpha,\beta} |\Pr[f'(\phi_0,0)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,1)=\beta]| = \\
&\sum_{\alpha,\beta} |\Pr[f'(\phi_0,0)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] + \\
&\qquad \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,1)=\beta]| \leq \\
&\sum_{\alpha,\beta} |\Pr[f'(\phi_0,0)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta]| + \\
&\sum_{\alpha,\beta} |\Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,1)=\beta]| = \\
&\sum_{\beta} \Pr[f'(\phi_1,0)=\beta]\cdot\Big(\sum_{\alpha} |\Pr[f'(\phi_0,0)=\alpha] - \Pr[f'(\phi_0,1)=\alpha]|\Big) + \\
&\sum_{\alpha} \Pr[f'(\phi_0,1)=\alpha]\cdot\Big(\sum_{\beta} |\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_1,1)=\beta]|\Big) = \\
&\Delta(f'(\phi_0,0), f'(\phi_0,1)) + \Delta(f'(\phi_1,0), f'(\phi_1,1)).
\end{aligned}$$

The case where $\phi = \phi_0 \vee \phi_1$ is different because the bit b is shared between two bits b_0 and b_1. Specifically,

$$\Pr[f'(\phi,0)=\langle\alpha,\beta\rangle] = \frac{1}{2}\cdot\Pr[f'(\phi_0,0)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] + \frac{1}{2}\cdot\Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,1)=\beta]$$

and

$$\Pr[f'(\phi,1)=\langle\alpha,\beta\rangle] = \frac{1}{2}\cdot\Pr[f'(\phi_0,0)=\alpha]\cdot\Pr[f'(\phi_1,1)=\beta] + \frac{1}{2}\cdot\Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta].$$

Hence,

$$\begin{aligned}
&\Pr[f'(\phi,1)=\langle\alpha,\beta\rangle] - \Pr[f'(\phi,0)=\langle\alpha,\beta\rangle] = \\
&1/2(\Pr[f'(\phi_0,0)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta] + \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,1)=\beta]) - \\
&1/2(\Pr[f'(\phi_0,0)=\alpha]\cdot\Pr[f'(\phi_1,1)=\beta] + \Pr[f'(\phi_0,1)=\alpha]\cdot\Pr[f'(\phi_1,0)=\beta]) = \\
&1/2\cdot\Pr[f'(\phi_1,0)=\beta](\Pr[f'(\phi_0,0)=\alpha] - \Pr[f'(\phi_0,1)=\alpha]) - \\
&1/2\cdot\Pr[f'(\phi_1,1)=\beta](\Pr[f'(\phi_0,0)=\alpha] - \Pr[f'(\phi_0,1)=\alpha]) = \\
&1/2\cdot(\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_1,1)=\beta])(\Pr[f'(\phi_0,0)=\alpha] - \Pr[f'(\phi_0,1)=\alpha]).
\end{aligned}$$

Using the above equality we conclude that

$$
\begin{aligned}
&\Delta(f'(\phi,0),f'(\phi,1)) = \\
&\frac{1}{2}\sum_{\langle\alpha,\beta\rangle} |\Pr[f'(\phi,1)=\langle\alpha,\beta\rangle] - \Pr[f'(\phi,0)=\langle\alpha,\beta\rangle]| = \\
&\frac{1}{2}\cdot\frac{1}{2}\sum_{\langle\alpha,\beta\rangle} |(\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_1,1)=\beta])(\Pr[f'(\phi_0,0)=\alpha] - \Pr[f'(\phi_0,1)=\alpha])| = \\
&\Big(\frac{1}{2}\sum_{\beta} |\Pr[f'(\phi_1,0)=\beta] - \Pr[f'(\phi_1,1)=\beta]|\Big)\cdot\Big(\frac{1}{2}\sum_{\alpha} |\Pr[f'(\phi_0,0)=\alpha] - \Pr[f'(\phi_0,1)=\alpha]|\Big) \\
&= \Delta(f'(\phi_1,0),f'(\phi_1,1))\cdot\Delta(f'(\phi_0,0),f'(\phi_0,1)).
\end{aligned}
$$

C Proof of Lemma 6.7

Fix $\phi_0, \phi_1, \vec{x}$ and D. To simplify the presentation we omit $\vec{x}$. We start with the $\wedge$ operator. Let C_0 (respectively, C_1) be the circuit that on input y obtains a random sample y' of $f'(\phi_0,0)$ (respectively, $f'(\phi_1,1)$), and outputs $D(y',y)$ (respectively, $D(y,y')$). Thus, by Construction 6.3 of f',

$$
\begin{aligned}
&\texttt{adv}(D, f'(\phi_0\wedge\phi_1,0), f'(\phi_0\wedge\phi_1,1)) = \\
&|\Pr[D(\langle f'(\phi_0,0), f'(\phi_1,0)\rangle)=1] - \Pr[D(\langle f'(\phi_0,1), f'(\phi_1,1)\rangle)=1]| = \\
&|\Pr[D(\langle f'(\phi_0,0), f'(\phi_1,0)\rangle)=1] - \Pr[D(\langle f'(\phi_0,0), f'(\phi_1,1)\rangle)=1] + \\
&\Pr[D(\langle f'(\phi_0,0), f'(\phi_1,1)\rangle)=1] - \Pr[D(\langle f'(\phi_0,1), f'(\phi_1,1)\rangle)=1]| \leq \\
&\texttt{adv}(C_0, f'(\phi_1,0), f'(\phi_1,1)) + \texttt{adv}(C_1, f'(\phi_0,0), f'(\phi_0,1)).
\end{aligned}
$$

We turn our attention to the $\vee$ operator. Let E_0 (respectively, E_1) be the circuit that on input y uniformly picks $b'\in\{0,1\}$, obtains a random sample y' of $f'(\phi_1,b')$ (respectively, $f'(\phi_0,b')$), and outputs $b'\oplus D(y,y')$ (respectively, $b'\oplus D(y',y)$). We only consider the case of E_0 because the case of E_1 is symmetric. Thus, by Construction 6.3 of f',

$$
\begin{aligned}
&\Pr[D(f'(\phi_0\vee\phi_1,0))=1] = \\
&1/2\cdot\Pr[D(\langle f'(\phi_0,0), f'(\phi_1,0)\rangle)=1] + 1/2\cdot\Pr[D(\langle f'(\phi_0,1), f'(\phi_1,1)\rangle)=1], \text{ and} \\
&\Pr[D(f'(\phi_0\vee\phi_1,1))=1] = \\
&1/2\cdot\Pr[D(\langle f'(\phi_0,1), f'(\phi_1,0)\rangle)=1] + 1/2\cdot\Pr[D(\langle f'(\phi_0,0), f'(\phi_1,1)\rangle)=1],
\end{aligned}
$$

and therefore

$$
\begin{aligned}
&\texttt{adv}(E_0, f'(\phi_0,0), f'(\phi_0,1)) = \\
&|\Pr[E_0(f'(\phi_0,0))=1] - \Pr[E_0(f'(\phi_0,1))=1]| = \\
&|1/2\cdot\Pr[D(\langle f'(\phi_0,0), f'(\phi_1,0)\rangle)=1] - 1/2\cdot\Pr[D(\langle f'(\phi_0,0), f'(\phi_1,1)\rangle)=1] - \\
&1/2\cdot\Pr[D(\langle f'(\phi_0,1), f'(\phi_1,0)\rangle)=1] + 1/2\cdot\Pr[D(\langle f'(\phi_0,1), f'(\phi_1,1)\rangle)=1]| = \\
&|\Pr[D(f'(\phi_0\vee\phi_1,0))=1] - \Pr[D(f'(\phi_0\vee\phi_1,1))=1]| = \\
&\texttt{adv}(D, f'(\phi_0\vee\phi_1,0), f'(\phi_0\vee\phi_1,1)).
\end{aligned}
$$

It remains to show that the size of C_i and E_i is as stated. Notice that each circuit takes a string y as input, and invokes D on y and y', where y' is obtained by using Construction 6.3 of f', with $\phi_{\bar{i}}$ and $\{x_j | j \in P(\phi_{\bar{i}})\}$ hardwired into it. Thus, the size of each of C_i and E_i is at most $|D| + |f| + |\phi_{\bar{i}}| + \sum_{j \in P(\phi_{\bar{i}})} |x_j|$.