A Characterization of Non-interactive Instance-Dependent Commitment-Schemes (NIC)

Bruce Kapron, Lior Malka, and Venkatesh Srinivasan

Department of Computer Science
University of Victoria, BC, Canada
bmkapron,liorma,venkat@cs.uvic.ca

Abstract. We provide a new characterization of certain zero-knowledge protocols as *non-interactive instance-dependent commitment-schemes* (NIC). To obtain this result we consider the notion of V-bit protocols, which are very common, and found many applications in zero-knowledge. Our characterization result states that a protocol has a V-bit zero-knowledge protocol if and only if it has a NIC. The NIC inherits its hiding property from the zero-knowledge property of the protocol, and vice versa.

Our characterization result yields a framework that strengthens and simplifies many zero-knowledge protocols in various settings. For example, applying this framework to the result of Micciancio et al. [18] (who showed that some problems, including GRAPH-NONISOMORPHISM and QUADRATIC-RESIDUOUSITY, *unconditionally* have a concurrent zero-knowledge proof) we easily get that arbitrary, monotone boolean formulae over a large class of problems (which contains, e.g., the complement of any random self-reducible problem) *unconditionally* have a concurrent zero-knowledge proof.

Keywords: zero-knowledge, commitment-schemes, random self-reducibility.

1 Introduction

Zero-knowledge protocols are two party protocols that enable one party (the prover) to convince another party (the verifier) of an assertion, with the guarantee that the verifier learns nothing but the truth of the assertion [14]. These protocols play a central role in the theory of cryptography, and they are also interesting from a complexity theoretic perspective because they facilitate the study of **NP** through interaction and randomness.

Zero-knowledge protocols and cryptography heavily rely on commitment-schemes. For example, every language in **NP** has a computational zero-knowledge (**CZK**) protocol [13,5] if bit commitment-schemes (equivalently, one-way functions [15,20]) exit. Consequently, many results about zero-knowledge protocols, and cryptography in general, are based on unproven assumptions.

Recently, Vadhan [27] gave a characterization of **CZK**, called the SZK/OWF-CHARACTERIZATION, which leads to the construction of a special scheme from any zero-knowledge protocol. Utilizing this scheme and the techniques already known from the conditional study of zero-knowledge, Vadhan was able to prove many results about **CZK** without relying on any unproven assumptions. A similar approach was applied by Nguyen and Vadhan [21] in the context of zero-knowledge proofs with efficient

L. Arge et al. (Eds.): ICALP 2007, LNCS 4596, pp. 328–339, 2007.

provers[1], and by Ong and Vadhan [22] in the context of zero-knowledge arguments. The works of [27,21,22] demonstrate that we can prove unconditional results about zero-knowledge protocols. This can be done by characterizing zero-knowledge protocols as special bit commitment-schemes, and then using these special schemes instead of bit commitment-schemes[2].

We continue this line of research. That is, we construct special schemes from a specific class of zero-knowledge protocols, and then we use the special schemes instead of bit commitment-schemes. Our schemes are simply functions. That is, by restricting ourselves to a specific class of zero-knowledge protocols we are able to construct very simple non-interactive schemes. In contrast, the schemes of Vadhan [27] can be constructed from *any* zero-knowledge protocol, but they are interactive, and have an involved definition (similar in flavor to that of zero-knowledge protocols). We stress that although our schemes are constructed from specific zero-knowledge protocols, they can be used in other zero-knowledge protocols, and in various settings. That is, our characterization result yields a framework with wide applicability.

Our Results. We provide a new characterization of certain zero-knowledge protocols as special bit commitment-schemes. To obtain this result we consider the notion of V-bit protocols. Informally, in such protocols 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. These protocols are very common in zero-knowledge. Examples include the perfect zero-knowledge (**PZK**) proof of [4] for GRAPH-ISOMORPHISM, the statistical zero-knowledge (**SZK**) proof of [19] for certain lattice problems, the **SZK** and **PZK** proofs of [24] for variants of STATISTICAL-DISTANCE (SD), and more.

We construct an efficient function $f(x, b; r)$ from any V-bit zero-knowledge protocol for a promise-problem $\Pi \stackrel{\text{def}}{=} \langle \Pi_{\text{Y}}, \Pi_{\text{N}} \rangle$. The inputs to f are a string x, a bit b, and randomness r. The output y of f hides b when x is a YES instance, and binds to b when x is a NO instance. More precisely, given $y = f(x, b; r)$, if $x \in \Pi_{\text{Y}}$, then b cannot be determined from y, and if $x \in \Pi_{\text{N}}$, then y can be a commitment to either 0 or 1, but not both (i.e., $y \neq f(x, 1-b; r')$ for all r'). Notice that unlike bit commitment-schemes, the hiding and the binding properties of f may not hold simultaneously. Since f is a non-interactive commitment-scheme for Π, we call f a *non-interactive instance-dependent commitment-scheme* (NIC). Using the techniques of [10,16] we get the following:

Main result (informal). A problem Π has a V-bit zero-knowledge protocol if and only if Π has a NIC.

The NIC f inherits its hiding property from the zero-knowledge property of the V-bit protocol, and vice versa. For example, the **SZK** protocols for the lattice problems of Micciancio and Vadhan [19] yield a statistically hiding NIC for these problems, and vice versa.

The notion of V-bit protocols is related to Cramer's notion of Σ-protocols [7]. These protocols are similar to V-bit protocols in that they are also 3-round public-coin

[1] A prover is *efficient* if given witness for input x it runs in time polynomial in $|x|$.

[2] The idea of replacing a bit commitment-scheme with a special scheme is due to Itoh et al. [16]. However, [16] construct a special scheme (different from that of [27,21]) for specific languages, whereas [27,21] provide a characterization result.

protocols, but instead of sending a bit b, the verifier sends a string e. However, if we consider V-bit *zero-knowledge* protocols, then the two notions are equivalent (the idea is to let e be the bit b, followed by zeroes [11]). Thus, our characterization result applies to Σ-protocols as well.

An immediate corollary to the characterization result is a transformation from V-bit *honest-verifier* zero-knowledge protocols to *dishonest-verifier* V-bit zero-knowledge protocols with *efficient provers*. The transformation preserves the zero-knowledge property of the original protocol. When we apply it to, .e.g., the protocol of [24] for variants of SD we immediately get a zero-knowledge protocol with an efficient prover for these variants, a result previously proved in [19] using similar ideas.

To show that our characterization result yields a useful framework we prove that NIC can be combined in a monotone boolean formula fashion (i.e., with AND and OR connectors). For example, if f is a NIC for GRAPH-ISOMORPHISM, and g is a NIC for the lattice problems of [19], then our lemma states that, e.g., $f \wedge g$ and $f \vee g$ are also NIC for the corresponding problems.

Second result (informal). The class of problems possessing NIC is closed under arbitrary monotone boolean formulae.

In addition, we prove that any random self-reducible (RSR) problem [2] has a perfectly hiding NIC. This folklore lemma follows from [26,25], but here we provide the proof for completeness. Let us see how combining these lemmas with our characterization result yields a very useful framework.

Removing computational assumptions. Our framework allows replacing the bit commitment-scheme in the protocol of Barak [3] with a NIC. The protocol inherits its zero-knowledge property from the hiding property of the NIC. For example, we get that if a problem has a perfectly hiding NIC, then it has a public-coin, round-efficient protocol (i.e., constant-round, with a negligible soundness error, and perfect completeness). The protocol is a **PZK** argument with a strict, polynomial-time *non-black-box* simulator. Notice that our protocol applies to problems that have a NIC, whereas the protocol of [3] applies to all of **NP**. As in [3], our protocol assumes the existence of collision-resistent hash functions. However, our result yields **PZK** protocols (as opposed to **CZK** in [3]), and it does not use bit commitment-schemes.

Abstraction and closure. Our framework strengthens and simplifies the result of Micciancio, Ong, Sahai, and Vadhan [18], who showed that a NIC with reversed properties[3] can replace the bit commitment-scheme in the protocol of [23]. Unlike [18], since we already have a characterization result, we do not need to construct such a NIC for specific problems (e.g., GRAPH-NONISOMORPHISM) or to be familiar with their definition (e.g., the lattice problems of [19]). Also, our framework shows that such NIC are closed under monotone boolean formulae. Thus, when we apply our framework to the theorem of [18] we get that arbitrary, monotone boolean formulae over a large class of problems (which contains, e.g., the complement of any random self-reducible problem)

[3] By "reversed" we mean that the hiding property holds on NO instances of the problem (instead of YES instances), and the binding property holds on YES instances (instead of NO instances).

unconditionally have a concurrent zero-knowledge proof. Similar improvements apply to local zero-knowledge [17], and quantum zero-knowledge [28].

Unifying previous works. Our framework unifies under the theme of NIC the results of Tompa and Woll [26], De Santis, Di Crescenzo, Persiano, and Yung [25], and Itoh, Ohta, and Shizuya [16]. Actually, these works only consider the perfect setting, and focus mainly on RSR problems. In contrast, our framework includes problems that are not known to be RSR, and it also considers the statistical and the computational setting. Hence, we get stronger and more general results under one simple theme.

Related work. We use the idea of Damgård [10] to obtain a NIC from any V-bit zero-knowledge protocol. Feige and Shamir used a similar idea to construct a trapdoor commitment-scheme from a bit commitment-scheme. Notice that the context of the work of Damgård [10] was to investigate whether zero-knowledge imply bit commitment-schemes. That is, [10] constructed an interactive bit commitment-scheme (as opposed to a non-interactive, *instance-dependent* commitment-scheme) from a proof of knowledge for any **NP**-hard relation, provided that the proof is a Σ-protocol. In contrast, we construct a NIC from any V-bit zero-knowledge protocol, regardless of whether the underlying problem is **NP**-hard. Also, the binding property of our NIC follows from the soundness of the underlying V-bit protocol, whereas in [10] the binding property is computational, and follows from the hardness of the underlying problem.

Our lemma on the closure of NIC under monotone boolean formulae uses the ideas of [25]. These ideas were also used in [24,27] to show closure properties. Our lemma is related to the closure results of Damgård and Cramer [9], and Cramer, Damgård, and Mackenzie [8]. All these results are proved by modifying the original protocols to obtain the closure. In contrast, we prove our closure result in a simple combinatorial setting (using NIC), and we always use the same underlying protocol of Blum [5] for **NP**. In addition, the results of [9,8] change the properties of the original protocol. For example, in [9] the protocol becomes a private-coin protocol, and in [8] the protocol becomes a 4-round protocol. In contrast, since we work with NIC, our underlying protocol does not change.

Our NIC is related to versions of SD, a complete problem for **SZK** [24]. That is, a problem has a perfectly (respectively, statistically) hiding NIC if and only if it Karp-reduces to $\overline{\mathrm{SD}^{1,0}}$ (respectively, $\overline{\mathrm{SD}^{1,1/2}}$). The notion of a perfectly hiding NIC is implicit in [4], and formalized in [16]. The notion of a statistically hiding NIC was formalized by [19]. Here we provide the computational analogue.

2 Non-interactive, Instance-Dependent Commitment-Schemes

We define non-interactive, instance-dependent commitment-schemes (NIC). Using the technique of [16] we show that if a problem has a NIC, then it has a V-bit zero-knowledge protocol (this holds for computationally hiding NIC if, in addition, the problem is in **NP**). The protocol is also a proof of knowledge, and it inherits its zero-knowledge property from the hiding property of the NIC.

Intuitively, a *bit commitment-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. Informally, a NIC

is a bit commitment-scheme in which the hiding and the binding properties depend on a string x, and thus 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 on a NO instance of some problem Π. Formally,

Definition 2.1 (NIC). *Let $\Pi = \langle \Pi_Y, \Pi_N \rangle$ be a promise-problem, and let $f(x,b;r)$ be a probabilistic, polynomial-time Turing machine on inputs x and $b \in \{0,1\}$. The string r denotes the randomness of f.*

We say that f is **binding** *on Π_N if for any $x \in \Pi_N$, 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 for any $x \in \Pi_Y$ and each $b \in \{0,1\}$ the ensembles $\{f(x,0)\}_{x\in\Pi_Y}$ and $\{f(x,1)\}_{x\in\Pi_Y}$ are statistically identical (respectively, statistically indistinguishable, computationally indistinguishable).*

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.*

When appropriate we will omit the random input r to f. Notice that if f is a perfectly or a statistically hiding NIC for Π, then as a class of problems **NP** contains Π. This is so because if $x \in \Pi_Y$, then there is a pair $\langle r, r' \rangle$ such that $f(x,0;r) = f(x,1;r)$, and if $x \in \Pi_N$, then no such pair exists. However, Π may not be in **NP** if f is computationally hiding . We give an example of a perfectly hiding NIC.

Example 2.1. NIC for the language GRAPH-ISOMORPHISM [4,16]. 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.

Our protocol follows the idea of [16], which uses the protocol of Blum [5] for the **NP**-complete problem HAMILTONIAN-CIRUIT (HC). In the protocol of [16] the prover and the verifier initially reduce the input x of the problem possessing a NIC to an instance G of HC, and then execute the zero-knowledge protocol of [5] using the NIC 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. When $x \in \Pi_Y$ the scheme is hiding, and thus the protocol is zero-knowledge. When $x \in \Pi_N$ the scheme is binding, and thus the protocol is sound. Our lemma follows. The proof is very similar to that of [16].

Lemma 2.1. *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.*

Itoh, Ohta, and Shizuya [16] observed that if Π has a statistically hiding NIC, then Π cannot be **NP**-complete, unless the polynomial hierarchy collapses [12,1,6]. In the next section we show that V-bit zero-knowledge protocols and NIC are equivalent. Thus, **NP**-complete languages cannot have V-bit **SZK** proofs, unless the polynomial hierarchy collapses.

3 Characterizing V-Bit Zero-Knowledge Protocols

We introduce the notion of V-bit protocols, and then show how to construct a NIC from a simulator of any V-bit zero-knowledge protocol. Since the zero-knowledge protocols constructed in Section 2 for problems possessing NIC are V-bit zero-knowledge protocols, we get our main theorem.

Theorem 3.1. *A promise-problem* Π *has a* V*-bit* **PZK** *(respectively,* **SZK***) proof if and only if* Π *has a perfectly (respectively, statistically) hiding* NIC*. Similarly,* Π *has a* V*-bit* **CZK** *proof if and only if* $\Pi \in$ ***NP*** *and* Π *has a computationally hiding* NIC*.*

We present the definition of V-bit protocols.

Definition 3.1 (V-bit protocol). *Let* $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \rangle$ *be a problem, and let* $\langle P, V \rangle$ *be a protocol for* Π *with* perfect completeness*. We say that* $\langle P, V \rangle$ *is* V-bit *if for any* $x \in \Pi_{\mathrm{Y}}$ *the interaction between* P *and* V *is as follows:* P *sends* m_1 *to* 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* $\langle x, m_1, b, m_2 \rangle$*.*

Using the idea of [10] we show how to construct a NIC from a simulator S for any V-bit zero-knowledge protocol $\langle P, V \rangle$. The NIC will be hiding on `YES` instances, and binding on `NO` instances. We start with the following idea to commit to a bit b: use randomness r to execute S on input x, obtain a transcript $\langle m_1, b', m_2 \rangle$ such that $b = b'$ and V accepts, and output m_1 as a commitment. 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 of $\langle P, V \rangle$, 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. The problem 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. Thus, the scheme is hiding. Our lemma follows.

Lemma 3.1. *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, and fix a simulator S for $\langle P, V \rangle$. Without loss of generality we can assume that S either outputs transcripts in which V accepts, or it outputs `fail`. 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 $V(x, m_1', b', m_2') = \texttt{accept}$, 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 $V(x, m_1, 0, m_2) = V(x, m_1, 1, m_2') = \texttt{accept}$. 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 $V(x, m_1, 0, m_2) = V(x, m_1, 1, m_2') = \texttt{accept}$. 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}. Starting with the statistical setting, we 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.

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

For any x we define $p_x \stackrel{\text{def}}{=} \Pr[S(x) = \texttt{fail}]$, where the probability is over the randomness to S. In addition, when S is a **HVPZK** simulator we are assuming that $p_x = 0$. By the definition of f, the above sum over b equals p_x. It remains to deal with the sums over m_1. We show that the first sum is upper bounded by $\Delta(\langle P, V\rangle(x), S(x)) - p_x/2$, 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.

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

Above we used the fact that S outputs transcripts in which V accepts, and then we used the fact that $\langle P, V\rangle$ is public-coin (which implies that for any m_1 the probability to choose an element of $\langle P, V\rangle(x)$ whose prefix is $\langle m_1, 0\rangle$ equals the probability to choose an element of $\langle P, V\rangle(x)$ whose prefix is $\langle m_1, 1\rangle$). We conclude that $\Delta(f(x,0), f(x,1)) \leq 2 \cdot \Delta(S(x), \langle P, V\rangle(x))$. 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 contradiction that there is a probabilistic, polynomial-time Turing machine D that distinguishes $\{f(x,0)\}_{x \in \Pi_Y}$ and $\{f(x,1)\}_{x \in \Pi_Y}$. Thus, D distinguishes $\{f'(S(x),0)\}_{x \in \Pi_Y}$ and $\{f'(S(x),1)\}_{x \in \Pi_Y}$, and the following expression is non-negligible:

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

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 $x \in \Pi_Y$. It follows that there is $b \in \{0,1\}$ such that D distinguishes $\{f'(\langle P, V \rangle, b)\}_{x \in \Pi_Y}$ and $\{f'(S(x), b)\}_{x \in \Pi_Y}$. Since f' is efficient, this contradicts the fact that S is a **HVCZK** simulator. We conclude that f is computationally hiding on Π_Y. The lemma follows.

Theorem 3.1 presented in the beginning of this section immediately follows from Lemmas 2.1 and 3.1. Thus, we get a characterization of V-bit zero-knowledge protocols as NIC. We remark that Theorem 3.1 can be extended to arguments, and to relaxed notions of V-bit protocols.

4 Random Self-reducibility Implies NIC

We prove the folklore theorem that if a problem Π is random self-reducible, then Π has a perfectly hiding NIC. Our proof uses the idea behind the construction of the subroutine in the protocol of [25] (see Section 3.3 in [25]). Combining this theorem with our closure result from the next section allows us to strengthen and unify the results of [26,25,16], and achieve all the improvements claimed in the introduction. We define random self-reducibility.

Definition 4.1 (Random self-reducible language [2]). *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 $d(R_x) \stackrel{def}{=} \{z | \exists w \, \langle z, w \rangle \in R_x\}$. The language $\mathrm{L} \stackrel{def}{=} \{\langle x, z \rangle | x \in \mathcal{N}, \exists w \, \langle z, w \rangle \in R_x\}$ 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. Given $\mathcal{N}$ and R_x as in Definition 4.1 we define the problem $\Pi^{\mathrm{L}} \stackrel{\text{def}}{=} \langle \Pi^{\mathrm{L}}_{\mathrm{Y}}, \Pi^{\mathrm{L}}_{\mathrm{N}} \rangle$, where $\Pi^{\mathrm{L}}_{\mathrm{Y}} \stackrel{\text{def}}{=} \{\langle x, z \rangle | x \in \mathcal{N}, \exists w \, \langle z, w \rangle \in R_x\}$, and $\Pi^{\mathrm{L}}_{\mathrm{N}} \stackrel{\text{def}}{=} \{\langle x, z \rangle | x \in \mathcal{N}, \forall w \, \langle z, w \rangle \notin R_x\}$.

Lemma 4.1. *If* L *is a random self-reducible language, then* Π^{L} *has a perfectly hiding* NIC.

Proof. Let $\mathrm{L} \stackrel{\text{def}}{=} \{\langle x, z \rangle | x \in \mathcal{N}, \exists w \, \langle z, w \rangle \in R_x\}$ be a random self-reducible language. Consider the algorithms S and G from Definition 4.1. 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 the 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 4.1, $S(x, z; r) = y$ 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 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 4.1, 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 4.1. Neither did we use the fact that A_1 runs in polynomial time, nor did we use the witness outputted by G.

5 Closure of Problems Possessing NIC Under Monotone Boolean Formulae

We use the technique of [25] to show that the class of problems possessing NIC is closed under *arbitrary* (as opposed to fixed) monotone boolean formulae. For perfectly hiding NIC the analysis is simple, but for statistically and computationally hiding NIC the analysis is more complicated.

Motivation. Let f be a perfectly hiding NIC for a problem Π. Consider a prover and a verifier who are given instances $x_0, \ldots, x_n \in \Pi_{\mathrm{Y}} \cup \Pi_{\mathrm{N}}$, and suppose that the prover wants to prove to the verifier that more than half of the x_i's are in Π_{Y}. This statement can be expressed using the logical connectors AND (denoted $\wedge$) and OR (denoted $\vee$). The

prover can prove this statement if we can construct a NIC f' that is hiding when more than half of the x_i are in Π_Y, and binding otherwise. This is so because the statement is an **NP** statement, and the prover can use f' in the protocol of Blum [5] (as in Section 2). Later we will give a general construction that yields such f'. For now we consider the simple case where $n = 2$. That is, the prover proves that both x_0 and x_1 are in Π_Y.

To formulate the fact that the statement being proved is $x_0 \in \Pi_Y \wedge x_1 \in \Pi_Y$ we define the common input as $\langle \phi, x_0, x_1 \rangle$, where $\phi = a \wedge b$. Recall that we want to use the NIC f for Π to construct a NIC f' which is hiding when $x_0 \in \Pi_Y \wedge x_1 \in \Pi_Y$, and binding otherwise. We can construct such f by defining $f'(x_0, x_1, b) \stackrel{\text{def}}{=} \langle f(x_0, b), f(x_1, b) \rangle$. Thus, if $x_0, x_1 \in \Pi_Y$, then both $f(x_0, b)$ and $f(x_1, b)$ hide b, which implies that f' is hiding, and if $x_i \in \Pi_N$ (for some $i \in \{0, 1\}$), then $f(x_i, b)$ binds to b, and f' is binding. Notice that we omitted the randomness of f' from the notation, but the intention is that f' uses independent randomness in each execution of f.

We can formulate other statements too. For example, consider a prover and a verifier who are given x_0, x_1, and the prover wants to prove that either $x_0 \in \Pi_Y$ or $x_1 \in \Pi_Y$. Again, we can formulate this statement by defining $\langle \phi, x_0, x_1 \rangle$ as the input, where $\phi = a \vee b$. Recall that we want to use the NIC f for Π to construct a NIC f' which is hiding when $x_0 \in \Pi_Y \vee x_1 \in \Pi_Y$, and hiding otherwise. We can construct such f by defining $f'(x_0, x_1, b) \stackrel{\text{def}}{=} \langle f(x_0, b_0), f(x_1, b_1) \rangle$, where b_0 is uniformly chosen, and b_1 is chosen such that $b_0 \oplus b_1 = b$. Thus, if $x_0, x_1 \in \Pi_N$, then both $f(x_0, b)$ and $f(x_1, b)$ bind to b, which implies that f' is binding, and if $x_i \in \Pi_Y$ (for some $i \in \{0, 1\}$), then $f(x_i, b)$ hides b_i, and thus f hides b. Based on these $\wedge$ and $\vee$ cases we can give a general construction of a NIC f' from a NIC f.

Construction 5.1. *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 $n \geq m$. Let $r \in \{0, 1\}^*$ be a uniformly distributed input to f'.*

The recursive function $f'(\phi, \vec{x}, f, b; r)$ is defined as follows.

1. *If $\phi = a_i$ for some $1 \leq i \leq m$, then return $f(x_i, b, r)$.*
2. *Otherwise, there are monotone boolean formulae ϕ_0 and ϕ_1 such that $\phi = \phi_0 \wedge \phi_1$ or $\phi = \phi_0 \vee \phi_1$. Partition r into r_0 and r_1.*
3. *If $\phi = \phi_0 \wedge \phi_1$, then return $\langle f'(\phi_0, \vec{x}, f, b, r_0), f'(\phi_1, \vec{x}, f, b, r_1) \rangle$.*
4. *If $\phi = \phi_0 \vee \phi_1$, then return $\langle f'(\phi_0, \vec{x}, f, b_0, r_0), f'(\phi_1, \vec{x}, f, b_1, r_1) \rangle$, where $b_0 \in \{0, 1\}$ is uniformly distributed, and b_1 is chosen such that $b_0 \oplus b_1 = b$.*

Our next step is define a problem that allows the prover to prove *arbitrary* (as opposed to fixed) monotone, boolean formula statements. 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 $\phi_0 \wedge \phi_1$ or $\phi_0 \vee \phi_1$, where both ϕ_0 and ϕ_1 are monotone boolean formulae. Let $\Pi = \langle \Pi_Y, \Pi_N \rangle$ be a promise-problem, and let $x \in \Pi_Y \cup \Pi_N$. The *characteristic function* χ_Π of Π is defined as follows: if $x \in \Pi_Y$, then $\chi_\Pi(x) = 1$, and if $x \in \Pi_N$, then $\chi_\Pi(x) = 0$. Let ϕ be a boolean formula over $a_1, \ldots, a_m$, and let $x_1, \ldots, x_n \in \Pi_Y \cup \Pi_N$ 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 5.1. *Let* $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \rangle$ *be a problem. The problem* $\Phi(\Pi) \stackrel{def}{=} \langle \Phi(\Pi)_{\mathrm{Y}}, \Phi(\Pi)_{\mathrm{N}} \rangle$ *is defined as*

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

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* $\Phi(\Pi)^k \stackrel{def}{=} \langle \Phi(\Pi)^{\mathrm{k}}_{\mathrm{Y}}, \Phi(\Pi)_{\mathrm{N}} \rangle$*, where* $\Phi(\Pi)^{\mathrm{k}}_{\mathrm{Y}}$ *is defined as*

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

The definition of $\Phi(\Pi)$ allows the prover to prove *arbitrary* (as opposed to fixed) monotone, boolean formula statements, and so does the definition of $\Phi(\Pi)^{\mathrm{k}}$. This formulation has the advantage that the formula does not have to be hardwired into the protocol, or known in advance. Our theorem follows.

Theorem 5.2. *Let* $\Pi = \langle \Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \rangle$ *be a promise-problem with a* NIC f*, and let* f' *be the function constructed from* f*, given in Construction 5.1. Let* $k \in \mathbb{N}$*.*

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* $\Phi(\Pi)^{\mathrm{k}}$*.*

References

1. Aiello, W., Håstad, J.: Statistical zero-knowledge languages can be recognized in two rounds. J. of Computer and System Sciences 42(3), 327–345 (1991)
2. Angluin, D., Lichtenstein, D.: Provable security in cryptosystems: a survey. Technical Report 288, Department of Computer Science, Yale University (1983)
3. Barak, B.: How to go beyond the black-box simulation barrier. In: FOCS, pp. 106–115 (2001)
4. Bellare, M., Micali, S., Ostrovsky, R.: Perfect zero-knowledge in constant rounds. In: 22nd STOC, pp. 482–493 (1990)
5. Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the ICM, pp. 1444–1451 (1986)
6. Boppana, R.B., Håstad, J., Zachos, S.: Does co-NP have short interactive proofs? Inf. Process. Lett. 25(2), 127–132 (1987)
7. Cramer, R.: Modular Design of Secure yet Practical Cryptographic Protocols. PhD thesis, CWI and Uni. of Amsterdam (1996)
8. Cramer, R., Damgård, I., MacKenzie, P.D.: Efficient zero-knowledge proofs of knowledge without intractability assumptions. In: Public Key Cryptography, pp. 354–372 (2000)
9. Dåmgard, I., Cramer, R.: On monotone function closure of perfect and statistical zero-knowledge (1996)
10. Damgård, I.B.: On the existence of bit commitment schemes and zero-knowledge proofs. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 17–27. Springer, Heidelberg (1990)

11. Damgård, I.B.: On Σ-protocols (2005) Available online at www.daimi.au.dk/~ivan/Sigma.pdf
12. Fortnow, L.: The complexity of perfect zero-knowledge. In: Micali, S. (ed.) Advances in Computing Research, vol. 5, pp. 327–343. JAC Press (1989)
13. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. ACM 38(3), 691–729 (1991)
14. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. 18(1), 186–208 (1989)
15. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. 28(4), 1364–1396 (1999)
16. Itoh, T., Ohta, Y., Shizuya, H.: A language-dependent cryptographic primitive. J. Cryptology 10(1), 37–50 (1997)
17. Micali, S., Pass, R.: Local zero knowledge. In: STOC, pp. 306–315 (2006)
18. Micciancio, D., Ong, S.J., Sahai, A., Vadhan, S.P.: Concurrent zero knowledge without complexity assumptions. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 1–20. Springer, Heidelberg (2006)
19. Micciancio, D., Vadhan, S.P.: Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 282–298. Springer, Heidelberg (2003)
20. Naor, M.: Bit commitment using pseudorandomness. J. Cryptology 4(2), 151–158 (1991)
21. Nguyen, M.-H., Vadhan, S.: Zero knowledge with efficient provers. In: STOC '06. Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, Seattle, WA, USA, pp. 287–295. ACM Press, New York (2006)
22. Ong, S.J., Vadhan, S.: Zero knowledge and soundness are symmetric. Electronic Colloquium on Computational Complexity (ECCC) (TR06-139) (2006)
23. Prabhakaran, M., Rosen, A., Sahai, A.: Concurrent zero knowledge with logarithmic round-complexity. In: FOCS, pp. 366–375 (2002)
24. Sahai, A., Vadhan, S.P.: A complete problem for statistical zero-knowledge. J. ACM 50(2), 196–249 (2003)
25. De Santis, A., Di Crescenzo, G., Persiano, G., Yung, M.: On monotone formula closure of SZK. In: IEEE Symposium on Foundations of Computer Science, pp. 454–465. IEEE Computer Society Press, Los Alamitos (1994)
26. Tompa, M., Woll, H.: Random self-reducibility and zero-knowledge interactive proofs of possession of information. In: 28th FOCS, pp. 472–482 (1987)
27. Vadhan, S.P.: An unconditional study of computational zero knowledge. In: FOCS, pp. 176–185 (2004)
28. Watrous, J.: Zero-knowledge against quantum attacks. In: STOC, pp. 296–305 (2006)