Non-Interactive Zero Knowledge *Proofs* in the Random Oracle Model

Vincenzo Iovino[1] and Ivan Visconti[2]

[1]University of Luxembourg
vinciovino@gmail.com
[2] DIEM - University of Salerno
visconti@unisa.it

Abstract. The Fiat-Shamir (FS) transform is a well known and widely used technique to convert any constant-round public-coin honest-verifier zero-knowledge (HVZK) proof or argument system $\mathsf{HVZK} = (\mathcal{P}, \mathcal{V})$ in a non-interactive zero-knowledge (NIZK) argument system
$\mathsf{NIZK} = (\mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$. The FS transform is secure in the random oracle (RO) model and is extremely efficient: it adds an evaluation of the RO for every message played by $\mathcal{V}$.

While a major effort has been done to attack the soundness of the transform when the RO is instantiated with a "secure" hash function, here we focus on a different limitation of the FS transform that exists even when there is a secure instantiation of the random oracle: the soundness of NIZK holds against polynomial-time adversarial provers only. Therefore even when HVZK is a proof system, NIZK is only an argument system.

In this paper we propose a new transform from 3-round public-coin HVZK proof systems for several practical relations to NIZK *proof* systems in the RO model. Our transform outperforms the FS transform protecting the honest verifier from unbounded adversarial provers with no restriction on the number of RO queries. The protocols our transform can be applied to are the ones for proving membership to the range of a one-way group homomorphism as defined by [Maurer - Design, Codes and Cryptography 2015] except that we additionally require the function to be endowed with a trapdoor and other natural properties. For instance, we obtain new efficient instantiations of NIZK *proofs* for relations related to quadratic residuosity and the RSA function.

As a byproduct, with our transform we obtain essentially for free the first efficient non-interactive zap (i.e., 1-round non-interactive witness indistinguishable *proof* system) for several practical languages in the non-programmable RO model and in an ideal-PUF model.

Our approach to NIZK proofs can be seen as an abstraction of the celebrated work of [Feige, Lapidot and Shamir - FOCS 1990].

Keywords: FS transform, NIZK, random oracle model.

1 Introduction

Non-Interactive Zero-Knowledge (NIZK) proof and argument systems have been studied for about 30 years [BFM88,FLS90,Gol01]. The concept of proving a statement in just one round without leaking any information has been intriguing for theoreticians and extremely useful as building block for designers of cryptographic protocols. The initial constructions for NIZK worked in the common reference string (CRS) model and because of various limitations (e.g., the need of NP reductions, the non-reusability of the CRS, the expensive computations) their impact was mainly in the theoretical foundations of cryptography.

Proofs vs arguments. The gap between NIZK proof (NIZKP) systems and NIZK argument (NIZKA) systems consists in a different soundness requirement. The soundness property aims to prevent an adversarial prover from convincing the verifier about the veracity of a false statement. The powerful concept of a NIZK proof requires the soundness guarantee to be unconditional, therefore the adversarial prover can be unbounded. Instead, the notion of a NIZK argument has a significantly weaker soundness guarantee since it applies to PPT (corresponding to non-uniform polynomial-time algorithms) adversarial provers only.[1].

The difference seems subtle but may be fundamental in real-world applications. Consider an e-voting system that uses crypgtographic proofs to ensure the election result claimed by the authorities to be authentic. If the system uses NIZK proofs, then there is a guarantee that the authorities cannot subvert the result of the election whatever computing power they have. If NIZK arguments are instead employed, then the guarantee is only *conditional* (it holds only if the authorities do not have enough computational power).

The bridge between theory and practice: the Fiat-Shamir (FS) transform. The traditional power of the simulator in a NIZK proof/argument system consists in programming the common reference string (CRS). A popular alternative to the CRS model is the Random Oracle (RO) model [BR93]. The RO model assumes the availability of a perfect random function to all parties. One of the most successful applications of the RO model in cryptography is the FS transform that allows to obtain very efficient NIZK arguments [FS87]. The simulator of such a NIZK argument programs the RO (i.e., the simulator replaces at least in part the RO in answering to RO queries of the adversary).

In concrete implementations of this transform, prover and verifier replace the RO by some "secure" hash function.

Even if the RO methodology has been shown to be controversial already in [CGH98] and further negative results were published next [DNRS99,Bar01,GK03] [BLV03,DRV12,GOSV14,KRR16], NIZK arguments via the FS transform are widely used in concrete cryptographic protocols (e.g., in e-voting). We remark that one could also consider an hybrid notion where the adversarial prover can be unbounded except that it can query the random oracle a polynomial number of times only. We stress that in this paper we consider a truly unbounded adversarial prover, and as such, a NIZK proof system does not impose any limitation on the number of RO queries. This difference can be crucial in applications.

[1] In literature this difference is often overlooked. Despite this subtle difference, for simplicity we will call *proof* the string generated by the prover, irrespective of whether the prover be part of a proof or an argument system. We will however be precise on using the words "proof system" and "argument system".

1.1 Problem statement

The FS transform induces a significant soundness loss. Indeed it receives as input a constant-round public-coin honest-verifier zero-knowledge (HVZK) *proof* system and outputs a NIZK *argument* system. This is a step back compared to the known NIZK *proofs* in the CRS model [BFM88,FLS90,GOS06b,GS08].

Of course if one is interested in a NIZK proof system in the RO model there is a trivial approach: just evaluate the RO on input the instance x to get a random string that can be used to compute a NIZK proof in the common reference string model (e.g., [FLS90]). However the trivial approach is very unsatisfying for the following two reasons: 1) it requires expensive computations (sometimes including an NP reduction) that make the NIZK proof completely impractical, and 2) it requires some complexity assumptions (e.g., trapdoor permutations in [FLS90]) therefore incurring a significant security loss in the zero-knowledge guarantee.

These limitations of the FS-transform and of the above trivial approach motivate the main question of this work.

Open question: *is there an alternative transform that outputs an efficient NIZK proof system (i.e., soundness is guaranteed also against unbounded adversarial provers) in the RO model for practical languages without introducing any additional unproven hypothesis?*

1.2 The FS transform internals

Formal definitions of NIZK proofs and arguments of knowledge in the RO model through the FS transform have been investigated in several papers [FKMV12,BPW12,BFW15] and are discussed in Appendix A.3. For simplicity here we will now discuss the specific case of a 3-round public-coin HVZK proof system $\mathsf{3HVZK} = (\mathcal{P}, \mathcal{V})$ where the decision of the verifier is deterministic. However our discussion can be generalized to any constant-round public-coin HVZK argument system.

P sends a first message a to V, also called the commitment. Then V sends back a random challenge c. Finally P outputs the final message z, the answer to c. The triple (a, c, z) is called the transcript of an execution of $\mathsf{3HVZK}$ for an instance x and V takes deterministically the decision of accepting or not the transcript.

The FS transform constructs $\mathsf{NIZK} = (\mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ as follows. $\mathsf{NIZK.Prove}$ computes a precisely as P, but then the challenge c of V is replaced by the output of the RO on input the statement x and a, i.e., $c = H(x, a)$.[2] Finally $\mathsf{NIZK.Prove}$ computes z precisely as P would compute it.

NIZK is only computationally sound (i.e., it is an argument system) in the random oracle model. Indeed one can easily see that computing with non-negligible probability an accepting transcript for a false statement when the

[2] When the challenge c is computed as $H(a)$, the FS transform offers weaker security guarantees (see [BPW12,CPS$^+$16]). In this work, we will consider the *strong* FS transform.

adversarial prover runs in polynomial time, implies that the challenge is the output of one out of a polynomially bounded number of evaluations of the RO, and this can be translated to proving with non-negligible probability a false statement to V. Soundness cannot be claimed when instead the adversarial prover is unbounded and can therefore make an unbounded number of queries to the RO.

If $\mathsf{3HVZK}$ is also HVZK (see Appendix A.1), then the resulting NIZK argument system is additionally a computational ZK argument system. Indeed the ZK simulator can program the queries therefore being able to produce a simulated proof using the HVZK simulator that is computationally indistinguishable from the a real proof.

If $\mathsf{3HVZK}$ satisfies special soundness (i.e., there is a deterministic efficient extractor that from 2 different accepting transcripts for the same statement with the same first message outputs a witness), then the resulting NIZK argument system additionally enjoys witness extraction but limited to PPT adversarial provers. Known variations [Pas03,Fis05,FKMV12] of the FS transform produce NIZK *argument* systems that suffer of the same limitation of witness extraction with respect to PPT provers. We also stress that, to our knowledge, all previous variants of the FS transform (e.g., the ones of Pass [Pas03] and Fischlin [Fis05]) only achieve *computational* soundness (i.e., there is no security guarantee against an unbounded adversarial prover that as such can have unlimited access to the random oracle). In this paper we call NIZK proof of knowledge (NIZKPoK) a NIZK *proof* (i.e., soundness unconditional) system that enjoys the above extraction property (i.e., limited to PPT adversarial provers).

1.3 The soundness degradation of the FS transform

Suppose that the underlying interactive protocol has the following properties. The space of prover commitments has cardinality $\geq 2^{b(\lambda)}$, the verifier's challenges have length $k(\lambda)$, the soundness error is $2^{-k(\lambda)}$, with $k(\lambda) \in \omega(\log(\lambda)), b(\lambda) \geq \lambda + k(\lambda)$ where λ is the security parameter. Suppose further that the prover computes the answer z deterministically based on (a, c) and suppose that for each $x \notin L$ and each commitment a, there exists at least one challenge c such that (a, c, z) is an accepted transcript (a natural Σ-protocol satisfying the above requirements will be shown soon).

Fix an $x \notin L$ and consider the following unbounded prover $\mathsf{NIZK.Prove}^\star$ that aims to compute an accepting proof for x. $\mathsf{NIZK.Prove}^\star$ searches over all pairs of challenges and commitments (a_c, c) such that the above property holds (i.e., (a_c, c, z) is an accepting tuple, where z is the deterministic answer of the prover to (a_c, c)) *and* RO maps (x, a_c) into c; if $\mathsf{NIZK.Prove}^\star$ can find a pair (a_c, c) that verifies such conditions, it outputs (a_c, c, z) as its proof, otherwise outputs some error $\perp$.

For each challenge and commitment pair (a_c, c) the probability that the RO maps (x, a_c) into c such that (a_c, c, z) is an accepted transcript is $\geq 2^{-k(\lambda)}$ (by hypothesis on the soundness error). Thus, since there are $2^{b(\lambda)} \geq 2^{\lambda+k(\lambda)}$ commitments, $\mathsf{NIZK.Prove}^\star$ fails in proving the false statement x with probability

$< (1-\frac{1}{2^{k(\lambda)}})^{2^{\lambda+k(\lambda)}}$. Therefore, $\mathsf{NIZK.Prove}^{\star}$ succeeds with probability $\geq 1-(1-\frac{1}{2^{k(\lambda)}})^{2^{k(\lambda)}\cdot 2^{\lambda}} \approx 1-(\frac{1}{e})^{2^{\lambda}}$.[3]

This example shows that an unbounded prover can break the soundness of the FS transform applied to some particular proof system satisfying the above requirements. This is not an artificial counter-example as such requirements are satisfied by very natural proof systems like the ones of [CP93,CDS94].

Example. Consider for instance the protocol of Chaum and Pedersen [CP93] for proving that a tuple (g, h, u, v) of 4 group elements, in a group of prime order q, is a Diffie-Hellman (DH, in short) tuple.[4]

The prover chooses a random $r \in \mathbb{Z}_q$, where q is the order of the group, and sends the commitment $a = g^r, b = h^r$. The verifier sends a random challenge $c \in \mathbb{Z}_q$. The prover sends back deterministically $z = r + cw \mod q$ and the verifier accepts iff $g^z = au^c$ and $h^z = bv^c$.

Let $k(\lambda) = \lambda$ with security parameter λ equals to the length of the group elements. Then, the challenges have length $k(\lambda)$, the commitments have length $2 \cdot k(\lambda)$ and $k(\lambda)$ is also the soundness parameter. By using the simulator (of the special HVZK), it is easy to see that for each false statement $x \notin L$ and for each challenge c, there exists (a, z) such that (a, c, z) is an accepted transcript for x. Thus, the Chaum and Pedersen's protocol satisfies the above requirements and the soundness can be broken in time $\approx 2^{k(\lambda)}$.

Ineffectiveness of parallel repetition. A natural approach to adjust the FS transform in order to circumventing the above attack would be to execute p instances of the protocol in parallel and computing each challenge c_i, for $i = 1, \ldots, p$, as $\mathcal{RO}(x||a_i||i)$. Unluckily, this strategy does not improve the situation. In fact, while the number of possible challenges increases (each challenge now consists of $k \cdot p$ bits) the number of possible commitments also increases. A simple analysis shows that an attack similar to the previous one can be applied to such variant of the FS transform as well. Observe also that the previous attack can be viewed as a special case for $p(\lambda) = 1$.

In fact, consider a false statement x and an unbounded prover $\mathsf{NIZK.Prove}^{\star}$ similar to before aiming at computing an accepting proof for x. By the previous analysis on the protocol without repetitions (that can be seen as a special case for $p(\lambda) = 1$) and since the $p(\lambda)$ executions are independent, $\mathsf{NIZK.Prove}^{\star}$ succeeds with probability $\left(1-(\frac{1}{e})^{2^{\lambda}}\right)^{p(\lambda)}$ that is overwhelming in λ.

It is fundamental for the previous analysis to hold that the space of commitments is much bigger than the challenge space, as it is indeed the case in general

[3] This follows from the fact that $\lim_{\lambda\to\infty} 2^{k(\lambda)} = \infty$ and thus $\lim_{\lambda\to\infty}(1-\frac{1}{2^{k(\lambda)}})^{-2^{k(\lambda)}} = e$.

[4] Our transform cannot be applied to Chaum and Pedersen's protocol. However there are examples of natural 3-round public-coin HVZK protocols that have a big ratio between space of commitments and space of challenges and can be made non-interactive through our transform (e.g., quadratic residuosity).

for natural Σ-protocols for languages where deciding membership is non-trivial. In fact, if for instance the space of the challenges and commitments were of the same cardinality, the lower-bound on the winning probability of the previous prover would be only $\left(1-\frac{1}{e}\right)^{p(\lambda)}$ that is a negligible function. As we will see next, our transform still uses parallel repetitions but in a more careful way achieving NIZK proof systems for several natural and practical languages.

2 Our Results

In the main result of this work we give a *positive* answer to the above open question: we show a transform that gives NIZK proof systems for practical languages.

We first (see Appendix A.3) provide formal definitions for NIZK proof/argument systems in the RO model following the lines of Faust *et al.* [FKMV12] and Bernhard *et al.* [BFW15] but taking into account unbounded adversarial provers, therefore considering statistical soundness. Then we propose a new transform from a specific class of 3-round public-coin HVZK proof systems for a given class of relations (see below) to NIZK *proof* systems in the RO model for the same class of relations.

The protocols and relations we support are a strengthening of the ones introduced by Maurer [Mau15]. Precisely, Maurer shows that most of the known practical sigma protocols can be viewed as special case of a sigma protocol for a group homomorphic one-way function (OWF). Sigma protocols are a special case of 3-round public-coin HVZK proof systems (see Appendix A.1). Similarly, our transform can be applied to sigma protocols for proving that an element y is in the range of a group homomorphic OWF but we also require additional properties on the function f. Namely, we require the following properties (this is only a sketch and the complete set of properties will be presented in Def. 11).

1. f is a *trapdoor* OWF with range $\subseteq \{0,1\}^{m(\lambda)}$ for some polynomial $m(\cdot)$. The witness for the relation includes the trapdoor, i.e., the prover needs the trapdoor to compute the proof. The trapdoor also allows to efficiently decide whether a string $y \in \{0,1\}^{m(\lambda)}$ is in the range of f or not.
2. The language of all strings $y \notin \mathsf{Range}(f), y \in \{0,1\}^{m(\lambda)}$ is in $\mathsf{co\text{-}NP}$ and using the trapdoor for f it is possible to compute a witness for the fact that $y \notin \mathsf{Range}(f)$. That is, there are: a) an algorithm Prove_f that on input a string y and a trapdoor trap for f computes a proof π; b) an algorithm Verify that on input y and a proof π accepts if and only if $y \notin \mathsf{Range}(f)$; c) a PPT simulator Sim_f that, with input the security parameter, outputs a pair (a, π) that is distributed identically to (a', π') where a' is selected at random in the space of strings $y \in \{0,1\}^{m(\lambda)}, y \notin \mathsf{Range}(f)$ and $\pi' \leftarrow \mathsf{Prove}_f(y, \mathsf{trap})$.
3. A random element in $\{0,1\}^{m(\lambda)}$ falls outside the range of f with probability $\leq \frac{1}{q}$ (up to a negligible factor) for some constant $q > 1$; this probability affects the length of the proof.

We call such a function a special one-way group homomorphic function (SOWGHF). To exemplify the requirements, consider the squaring function modulo a Blum integer N that acts on the group $\mathbb{Z}_N^\star$; sigma protocols for such f allow to prove whether a number is a quadratic-residue modulo N. The first condition requires the existence of a trapdoor that in this case is the factorization of N and the range of the function is $\mathbb{Z}_N$.

The second condition requires the existence of an efficient way for proving that a number is not a quadratic residue mod N. As N is a Blum integer, -1 is a quadratic non-residue and thus $-y$ is a quadratic residue mod N if and only if y is a quadratic non-residue mod N. Thus, there exists a witness for proving that a number y is not a quadratic residue. The simulator can simply pick a random number $r \leftarrow \mathbb{Z}_N$ and output $(-r^2 \mod N, r)$.

The third condition is also satisfied since a random number in $\mathbb{Z}_N^\star$ is a quadratic-residue modulo N with probability $\frac{1}{4}$ and only a negligible fraction of the integers in $\mathbb{Z}_N$ are not in $\mathbb{Z}_N^\star$.

The second and third conditions are trivially satisfied when f is a permutation, e.g., for the RSA permutation. In that case, it makes no sense to prove with our NIZKP that a string is in the range of the function because for permutations the soundness is trivially satisfied. Moreover, the knowledge extraction property is also guaranteed by the FS transform at a lower cost. Nevertheless, one might consider statements like $\exists x_1, x_2, x_3$ such that $((y_1 = f_1(x_1) \wedge y_2 = f_2(x_2)) \vee y_3 = f_3(x_3))$, where one or more of the functions f_1, f_2, f_3 are permutations and at least one is not a permutation and all the functions satisfy our requirements. Following Cramer *et al.* [CDS94], our transform can be likewise extended to support such compound statements.

One might be worried that the first condition is very restrictive in that we do not just require f to be a trapdoor OWF but in addition to feed the trapdoor as input to the prover. However, notice that for many practical statements this is the case, e.g., for a proof of correct decryption of a Goldwasser-Micali's ciphertext [GM84] we can assume that the prover is endowed with the factorization of N.

We defer the reader to Appendix A.2 for more details on what we call special one-way group homomorphic functions and special protocols. In Appendix B we show several examples of SOWGHFs that exemplify the usefulness and practicality of our notion. Combined with our transform, this gives efficient NIZK proof systems with statistical soundness for disparate relations of wide applicability.

Our transform preserves the same properties of the FS transform (except some efficiency loss) but maintains the unconditional soundness of the starting protocol (unlike the FS transform). Regarding knowledge extraction, if the starting protocol satisfies special soundness then NIZK will have the same guarantee of extractability (see Appendix E) of the FS transform (i.e., extraction is possible against a PPT adversarial prover). Our transform does not add any computational assumption and thus our NIZK proof will be secure in the RO model without any unproven hypothesis.

Therefore our work gives the first NIZK proof systems for a variety of useful languages in the RO model. See Theorems 10 and 12.

As noted and proved by Yung and Zhao [YZ06] (see also Ciampi *et al.* [CPSV16]), if the original 3-round public-coin HVZK proof system is witness indistinguishable (WI), then the FS-transformed argument is still WI, and the security proof for WI is RO-free. Since the same holds for our transform we get an efficient non-interactive WI *proof* system (also called non-interactive zap in previous work) [GOS06a,GS08,DN00] in the *non-programmable* RO model. The result is formally stated in Corollary G. In Appendix 5 we present applications of this result to hardware-assisted cryptography. In particular we achieve an unconditional NIWI proof system in an ideal-PUF model.

As shown earlier, if the starting interactive proof system has challenges of length λ (with λ security parameter) and space of commitments of cardinality 2^{λ} then the soundness guarantee of the FS transform is completely violated by adversaries running in $\Theta(2^{\lambda})$ steps. Instead, the soundness of our transform is preserved with respect to adversaries running in $O(2^{\lambda})$ steps, when the instantiation of the random oracle is resilient to adversaries running in time $O(2^{\lambda})$ (e.g., idealized hash functions, PUFs). We formally state it in Conjecture 1.

3 Overview of Our Transform

We next describe our transform. Given an $x \notin L$, we denote by "space of bad commitments" S_x for x of a 3-round public-coin proof system the set of all commitments a such that there exist e, z such that $\mathcal{V}(x, a, e, z)$ is accepted by the verifier. With a slight abuse of notation, we say that the space of bad commitments S of $\mathsf{3HVZK}$ has cardinality $\leq N$ if for all $x \notin L$, the cardinality of S_x is $\leq N$.

Let $\mathsf{3HVZK}$ be a 3-round public-coin HVZK proof system $\mathsf{3HVZK} = (\mathcal{P}, \mathcal{V})$ with space of bad commitments of cardinality $\leq 2^{b(\lambda)}$, challenges of length $k(\lambda)$ and soundness error bounded by $s(\lambda)$. In Lemma 9 we prove that the FS transform applied to a such $\mathsf{3HVZK}$ results into a NIZK proof system with statistical soundness that degrates "nicely" in relation to $s(\lambda)$ when the space of the bad commitments $2^{b(\lambda)}$ is not too "big" (see the Lemma and also Theorem 10 for a more precise statement).

As a consequence, the problem of transforming sigma protocols into NIZK proofs with statistical soundness can be reduced to the problem of transforming 3-round public-coin HVZK proof systems into ones having arbitrarily *small* ratio between soundness error and space of bad commitments. So, we first present a transform from interactive protocols (that do not use the RO) to interactive protocols in the RO model with shorter commitment space. Then, applying the FS transform to the latter protocol will result into a NIZK with statistical soundness.

Trapdoor one-way group homomorphism and special protocols. Before presenting our transform, we define the class of relations supported by our protocols. As in Maurer [Mau15], the class of relations we consider are associated with an homomorphic OWF that in our case satisfies some additional requirements. We first recall the abstraction of Maurer [Mau15] and then we proceed to state the additional properties we require.

Consider two groups $(G, \cdot)$, $(H, *)$ and a one-way homomorphic function from G to H, that is a OWF with the property that $f(x_1 \cdot x_2) = f(x_1) * f(x_2)$. By abstracting several known protocols in the literature, Maurer presents a sigma protocol for proving that an element $y \in H$. In the Maurer's protocol, the prover knows x and the verifier knows $y = f(x)$. The prover selects a random element r in G and sends $a = f(k)$ to the verifier. The verifier sends back a number c selected at random in a challenge space that is a set of integers. The prover sends $z = k \cdot x^c$ to the verifier that accepts the transcript if and only if $f(z) = a * y^c$.

If a protocol is so defined and if in addition the function f satisfies the three conditions given in Appendix 2 we say that the protocol is *special*. We now show how to transform a special protocol ($\mathsf{spec\text{-}prot}$ henceforth) into one with shorter commitment space.

Reducing the space of commitments in special protocols. We construct a 3-round public-coin HVZK protocol $\mathsf{3HVZK} = (\mathsf{3HVZK.Prove}, \mathsf{3HVZK.Verify})$ for proving that $y \in \mathsf{Range}(f)$ from a $\mathsf{spec\text{-}prot}$ $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ for the same relation. We denote by Prove and Verify the efficient algorithms to prove and verify that a string $y \notin \mathsf{Range}(f)$ guaranteed by a $\mathsf{spec\text{-}prot}$ for f. We recall that in a $\mathsf{spec\text{-}prot}$ (see. Def. 13) the prover $\mathsf{SpecP.Prove}$ computes a commitment as $f(r)$ where r is a string drawn at random in the domain of f.

The idea behind the transform is to make the space of the commitments to be arbitrarily shorter than the space of the challenges. Specifically, we repeat the protocol a sufficient number of times p to increase the space of the challenges but at the same time we have to avoid that the space of the commitment increases with the same ratio. To that aim, we force the space of the commitment to be short by computing each commitment via the RO as $a_i = RO(y||i), i \in [p]$. In this way the space of the commitment is limited by $2^{|y|} \cdot p$ and thus, e.g, doubling p just double the space of the commitments while quadrupling the space of the challenges.

Under one of the assumptions for any $\mathsf{spec\text{-}prot}$ we can assume that with noticeable probability $a_i = f(r_i)$ for some r_i. If this is the case the prover, by means of the trapdoor, can invert a_i and get r_i. As mentioned above, the value r_i is meant to be the randomness used by $\mathsf{SpecP.Prove}$ to compute a commitment. Thus, using r_i $\mathsf{3HVZK.Prove}$ can complete the protocol (i.e., computing the final answer to send to the verifier). Note that, by hypothesis, the trapdoor can be also employed to check whether $a_i \in \mathsf{Range}(f)$. On the other hand, if this is not the case, the prover can still use the trapdoor to show the verifier that $a_i \notin \mathsf{Range}(f)$. As in FS, the verifier has also to check that each commitment a_i received by the prover equals $\mathcal{RO}(y, i)$.

Overall transform. We define our transform to be the result of applying the above transform to a $\mathsf{spec\text{-}prot}$ SpecP to obtain a protocol $\mathsf{3HVZK}$ and then apply FS transform to $\mathsf{3HVZK}$ to obtain a NIZK argument. It can be seen that our transform guarantees completeness if SpecP is perfectly complete. It can be seen that our transform guarantees computational ZK (see Appendix A.3) if SpecP is HVZK exactly as it is the case for the FS transform. It can be seen that our transform guarantees computational witness extraction (see Appendix E) if SpecP satisfies special soundness exactly as it is the case for the FS transform. More details will be given in Section 7.

The most important property of this new transform is that starting from a 3-round public-coin proof system that matches our requirements (i.e., what we call a spec-prot), our transform gives in output a non-interactive *proof* system, assuming a suitable choice of the parameters as we will specify later.

The parameter $p(\cdot)$ in our transform depends on the cardinality of the challenge space $k(\cdot)$ and the probability $q(\cdot)$ that a random element in the space of the commitments falls to be in the range of f. A more precise statement will be given in Section 7.

Connection to FLS. The reader may have noticed a connection to the work of Feige, Lapidot and Shamir (FLS) [FLS90]. A CRS-based NIZK like FLS can be easily converted to a NIZK in the RO model by setting the CRS to be the string $\mathcal{RO}(1^\lambda)$. In that case, the CRS in the FLS' NIZK can be seen as the first message in our protocol and then, by using a trapdoor, the prover in FLS is able to open the bits to the verifier in a selected way.

As we want to avoid expensive NP-reductions, in our case the trapdoor depends on the language. Moreover we have to handle the case when f is not a permutation.

4 Comparison

Comparison. Here we compare in more detail the NIZK proofs obtained through our transform with other NIZK arguments and proofs discussed before.

In Table 1 we present a comparison of the NIZK proof resulting to other NIZK proofs and arguments known in the literature (see Section 6). The NIZK proof and argument system in the comparison are very different in that they admit so different and disparate relations or can prove general statements through expensive NP-reductions. Nevertheless, it makes sense to compare them in terms of properties achieved. We omit the comparison with the transform of Mittelbach and Venturi that can be instantiated only for specific classes of interactive protocols and uses strong computational assumptions.

The 3rd line in the table refers to a NIZK in the RO constructed from a CRS-based NIZK in the trivial way by replacing the CRS with the string $\mathcal{RO}(1^\lambda)$ and programming the RO in the obvious way. The ZK type is omitted but is implicitly assumed to be (multi-theorem adaptive) computational in the programmable RO model[5] for works in which the corresponding entry CRS is set to No and (multi-theorem adaptive) computational for the CRS model otherwise.

[5] This holds for NIZKAs resulting from the strong FS transform, not for the weak FS one [BPW12]

Efficiency: the case of quadratic residuosity. It is difficult to compare different NIZK proofs and arguments systems for practical statements when they can handle different classes of relations. However, it makes sense to compare FS-transformed NIZK argument to the NIZK proof systems resulting from our transform when both are for the same relation. As an example, we can compare a FS-transformed NIZK argument system for proving that an integer is a quadratic residue to a NIZK proof system resulting from our transformation for the same relation.

The basic sigma protocol for proving quadratic residuosity has soundness error $\frac{1}{2}$. To make the soundness error, let us say $2^{-\lambda}$, it is necessary to repeat the protocol λ times and in turn applying the FS transform to the latter protocol results into just a NIZK argument with computational soundness. Let us now compare the improvement offered by our transform.

As it will be shown in our transform $\mathsf{Trans}_{\mathsf{main}}$ of Construction 2, to get soundness error $2^{-\lambda}$ our transform will compute a NIZKP consisting of $p(\lambda)$ repetitions of a 3-round protocol with essentially the same efficiency in terms of communication that the basic sigma protocol for quadratic residuosity, where $p(\lambda)$ has to satisfy the equation (cf. Equation (1) in Construction 2):

$$2^{2\cdot\lambda+\log(p(\lambda))}\cdot\left(\frac{1}{q}+\left(1-\frac{1}{q}\right)\cdot\frac{1}{k(\lambda)}\right)^{p(\lambda)}\leq 2^{-\lambda}.$$

As $\frac{1}{q}\approx\frac{3}{4}$, the above equation can be simplified to $3\cdot\lambda+\log(p(\lambda))\leq c\cdot p(\lambda)$ where $c\overset{\triangle}{=}3-\log_2(7)\approx 0.2$.

Then it can be seen that $p(\lambda)\approx 16\cdot\lambda$ satisfies the equation. Therefore, our transform allows to upgrade from computational to statistical soundness at a cost of a moderate factor of inefficiency.

5 Applications

Efficient NIWI Proofs in the NPRO Model. Yung and Zhao [YZ06] (see also Ciampi *et al.* [CPSV16]) observed that if the original 3-round public-coin HVZK proof system is witness indistinguishable (WI), then the FS-transformed argument is still WI, and the security proof for WI is RO free. Since the same holds for our transform, we get an efficient non-interactive witness indistinguishable (NIWI) *proof* system (also called non-interactive zap in previous work) [GOS06a] [GS08,DN00] in the *non-programmable* RO model. Next we show an application of this primitive.

Unconditional NIWI proofs in the ideal-PUF model. In last decade, there has been a renewed interest about hardware-assisted cryptographic protocols and physically uncloneable functions (PUFs, in short) [PRTG02,GCvD02,TSS^{+}05] [Kat07,HL08,GKR08,DORS08,AMS^{+}09,GIS^{+}10,BFSK11,OSVW13,RvD13]. We note that our unconditional NIWI proof system in the NPRO can be turned in an unconditional NIWI *proof* system in the *ideal*-PUF model, in which the PUF acts like a RO.

More specifically, we consider the availability of an *ideal*-PUF. Note that this is different from assuming a RO. In the RO model, all parties need to have access to the same function. In the ideal-PUF model we envision, we just assume that an hardware token acting as an ideal-PUF can be attached to a proof and sent from a party to another (specifically, from the prover to the verifier). We observe that our unconditional NIWI proof system in the NPRO can be turned in an unconditional NIWI *proof* system in the ideal-PUF model.

Work	Efficiency	Soundness?	CRS?	PV?	Unconditional?*	PoK?
NIZKPoK of [GOS06b]	NP-reductions	Stat	Yes	Yes	No	Stat
NIZKPoK of [GS08]	Efficient	Stat	Yes	Yes	No	Stat
NIZKPoK of [GS08] with CRS set to $\mathcal{RO}(1^\lambda)$	NP-reductions	Stat	No	Yes	No	Stat
Transforms of [Lin15,CPSV16]	Efficient	Comp	Yes	Yes	No	No
Transforms of[DFN06] [VV09,CG15]	Efficient	Comp	Yes	No	No	No
Transforms of [Pas03,Fis05]	Efficient	Comp	No	Yes	Yes	CS**
Transform of FS	Very efficient	Comp	No	Yes	Yes	CR
Our transform	Efficient	Stat	No	Yes	Yes	CR

Table 1. Stat denotes statistical and Comp computational. PV denotes public verifiability: a YES refers to standard NIKZP/NIZKA and a NO to designated verifier ones. CR denotes computational extractability with rewinding extractors and CS denotes computational extractability with straight-line extractors. The ZK type is omitted but is implicitly assumed to be (multi-theorem adaptive) computational in the programmable RO model for works in which the corresponding entry CRS is set to No and (multi-theorem adaptive) computational for the CRS model otherwise. *: When referred to the transforms, a No means that the transform does not *add* any additional computational assumption (beyond assuming the RO model) beyond the ones of the underlying starting protocol (that could even be unconditional). **: Note that the definition of online extractability of Fischlin implicitly assumes that the adversary is possibly computationally unbounded but limited to a polynomial number of RO queries. Thus, according to our terminology, it is still an argument with computational extractability.

6 Related Work

CRS-based NIZK proof and argument systems have been intensively studied in the last 30 years in a sequel of works [BFM88,FLS90,RS92,BY96,Pas03,BCNP04,Ps05] [GOS06b,AF07,GS08,Pas13,BFS16]. One of the initial motivations for CRS-based NIZK proof was CCA-security [NY90,CS98,Sah99,CS03,Lin06]. In this setting, the CRS is computed by the receiver, while the NIZK proofs are computed by the sender of ciphertexts. Thus, for CCA-security the CRS model does not pose any issue. However, in e-voting the authority cannot compute the CRS because it must compute proofs that show the correctness of the tally and thus cannot be the same party that computes the CRS that thus has to be setup by a trusted party.

An alternative to the CRS model is the RO model that does not solve the issues of the CRS model but often leads to the design of more efficient protocols. The RO methodology has been introduced in the groundbreaking work of Bellare and Rogaway [BR93]. Canetti *et al.* [CGH98] show that the RO methodology is unsound in general and several works [DNRS99,Bar01,GK03,BLV03,BDSG$^+$13] [GOSV14,KRR16] study the security of the FS methodology. The first rigorous analysis of the FS transform (applied to the case of signature schemes) appeared in Pointcheval and Stern [PS00]. Since the introduction of the FS transform [FS87], a lot of works have investigated alternative transformations achieving further properties or mitigating some issues of FS.

Pass [Pas03] and Fischlin [Fis05] introduce new transformations with straight-line extractors to address some problems that arise when using the NIZK argument systems resulting from the FS transform in larger protocols [SG02]. The NIZK systems resulting from the Pass' and Fischlin's transforms share the same limitation of FS of being *arguments*, i.e., sound only against computationally bounded adversaries. Furthermore, as in our case, Fischlin's transform also results in a completeness error.

(Note that the definition of online extractability of Fischlin implicitly assumes that the list of RO queries given to the extractor has polynomial size and thus only withstands adversaries that are possibly computationally unbounded *but* limited to a polynomial number of RO queries; according to our terminology, this limitation brings to an argument system with computational extractability.[6])

Damgård *et al.* [DFN06] propose a new transformation for the standard model but it results in NIZK argument systems that are only *designated verifier*, rests on computational assumptions and has soundness limited to a logarithmic number of theorems. Designated verifier NIZK proofs are sufficient for some applications (e.g., non-malleable encryption [PsV06]) but not for others like e-voting in which public verifiability is a wished property. The limitation on the soundness of the Damgård's transformation has been improved in the works of Ventre and Visconti [VV09] and Chaidos and Groth [CG15].

Lindell [Lin15] (see also the improvement of Ciampi *et al.* [CPSV16]) puts forward a new transformation that requires both a *non-programmable* RO and a CRS and has computational complexity only slightly higher than FS. The transformations of Lindell and Ciampi *et al.* are based on computational assumptions whereas ours does not require any unproven hypothesis.

[6] Note that also the FS transform leads to statistically sound proof systems against computationally unbounded provers constrained to a polynomial number of RO queries. In this paper, we deem a non-interactive system in the RO a proof system only if it enjoys statistical soundness against unbounded adversaries without any limitation on the number of RO queries.

Mittelbach and Venturi [MV16] investigate alternative classes of interactive protocols where the FS transform does have standard-model instantiations but their result yields NIZK argument systems and is based on strong assumptions like indistinguishability obfuscation [GGH$^+$13], and as such is far from being practical. Moreover the result of Mittelbach and Venturi seems to apply only to the weak FS transform in which the statement is not hashed along with the commitment. The weak FS transform is known to be insecure in some applications [BPW12]. In this work, we only consider the strong FS transform.

The work of Mittelbach and Venturi has been improved by Kalai *et al.* [KRR16] that, building on [BLV03,DRV12], have shown how to transform any public-coin interactive proof system into a *two-round* argument system using strong computational assumptions. The latter work does *not* yield non-interactive argument systems.

Sigma protocols, on which efficient NIZK arguments (and our NIZK proofs) in the RO model are based, have been intensively studied [CP93,CDS94,FKI06] [BR08,ABB$^+$10,Mau15,GMO16]. Sigma protocols incorporate properties both of interactive proof systems and proofs of knowledge systems [GMR89,BG93]. Faust *et al.* [FKMV12] and Bernhard *et al.* [BFW15] provide a careful study of the definitions and security properties of the NIZK argument systems resulting from the FS transform but they do not investigate the possibility of achieving *statistically* sound proofs. Both works, as well as ours, make use of the general forking lemma of Bellare and Neven [BN06] that extends the forking lemma of Pointcheval and Stern [PS00]. We note that in our NIWI the RO can be replaced by an ideal PUF. In the last decade, a lot of works study constructions and applications of hardware-assisted cryptographic protocols and PUFs [PRTG02,GCvD02,Kat07,HL08,GKR08,DORS08,AMS$^+$09,BFSK11,OSVW13,RvD13].

Roadmap. In Appendix A we provide the necessary background and formal definitions of all the primitives and concepts used in this work, including our new framework of special one-way group homomorphic functions. Additional definitions regarding extractability will be given in Appendix E. In Section 7 we present our main transform, in Appendix D we analyze its soundness and in Appendices E-G zero-knowledge, extractability and additional properties. In Appendix B we present several instantiations of special one-way group homomorphic functions.

7 Our Transform

7.1 Step I: From spec-prot to 3-Round Public-Coin HVZK in the ROM

For the sake of exposition, we define our main transform as consisting of two transforms. The first one transforms a spec-prot into a 3-round public-coin HVZK protocol in the RO model.

Specifically, $\mathsf{Trans}(c(\cdot), k(\cdot), q), m(\cdot), f)$ converts a $\mathsf{spec\text{-}prot}$ SpecP $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f into a 3-round public-coin HVZK proof system $\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f] = (\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f].\mathsf{Prove}$, $\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f].\mathsf{Verify})$ with commitments of length $c(\lambda) \cdot p(\lambda)$, space of bad commitments of cardinality $2^{\lambda + \log(p(\lambda))}$, challenges of length $k(\lambda) \cdot p(\lambda)$. Moreover, $\mathsf{3HVZK}$ is associated with a polynomial $\mathsf{poly}_{\mathsf{inp}}(\cdot)$.

The algorithms of $\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f]$ when run on an input x with $|x| \triangleq \lambda$ need oracle access to a function $\mathcal{RO}$ with domain $\{0,1\}^{\mathsf{poly}_{\mathsf{inp}}(\lambda)}$ and co-domain $\{0,1\}^{c(\lambda)}$, and guarantee soundness bounded by $p(\lambda)$. We next define our transform $\mathsf{Trans}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f]$.

Construction 1 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q. Our transform $\mathsf{Trans}(c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f)$ is a polynomial-time algorithm that takes as input the description of f (and thus implicitly SpecP), the description of functions $c(\cdot), k(\cdot)$, $q, m(\cdot)$ and $p(\cdot)$ and outputs a pair $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f])$ that consists of the description of a polynomial and the description of a proof system computed as follows.

Compute $\mathsf{poly}_{\mathsf{inp}}(\cdot) = \lambda + \log(p(\cdot))$, and set $\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f] = (\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f].\mathsf{Prove}$, $\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f].\mathsf{Verify})$ according to the description of the following two algorithms that are algorithms with oracle access to a function $\mathcal{RO}$ with domain $\{0,1\}^{\mathsf{poly}_{\mathsf{inp}}(\lambda)}$ and co-domain $\{0,1\}^{c(\lambda)}$.

In the following we denote by $\mathsf{SpecP.Prove}(y, (x, \mathsf{trap}), f^{-1}(a_i), e_i)$ the output of $\mathsf{SpecP.Prove}$ when executed with theorem z, witness (y, trap), first message computed with randomness $f^{-1}(a_i)$ (where the inverse is computed with trapdoor trap) and after having received as challenge e_i from the verifier. Note that the prover of a $\mathsf{spec\text{-}prot}$ computes its first message as $f(r)$ where r is the chosen randomness, thus the first message corresponds to $f(f^{-1}(a_i)) = a_i$.

$\mathsf{3HVZK.Prove}$, with inputs x, y and the trapdoor trap and $\mathsf{3HVZK.Verify}$, with input y, performs the following three rounds of communication.

```
– [Round 1] $\mathsf{3HVZK.Prove}(y, (x, \mathsf{trap})) \to \mathsf{3HVZK.Verify}(y)$.
      For each $i \in [p(\lambda)]$, do
        ∗ Send $a_i \leftarrow \mathcal{RO}(y||i)$ to $\mathsf{3HVZK.Verify}$.
    • endFor
– [Round 2] $\mathsf{3HVZK.Verify}(y) \to \mathsf{3HVZK.Prove}(y, (x, \mathsf{trap}))$.
      For each $i \in [p(\lambda)]$, do
        ∗ $e_i \leftarrow \{0, 1\}^{k(\lambda)}$
        ∗ Send $e_i$ to $\mathsf{3HVZK.Prove}$.
    • endFor
– [Round 3] $\mathsf{3HVZK.Prove}(y, (x, \mathsf{trap})) \to \mathsf{3HVZK.Verify}(y)$.
      For each $i \in [p(\lambda)]$, do
        ∗ If $a_i \notin \mathsf{Range}(f)$ do
            · $\pi_i \leftarrow \mathsf{Prove}(y, \mathsf{trap})$.
            · Send $z_i = (\perp, \pi_i)$ to $\mathsf{3HVZK.Verify}$.
        ∗ endIf
        ∗ else
            · Send $z_i \leftarrow \mathsf{SpecP.Prove}(y, (x, \mathsf{trap}), f^{-1}(a_i), e_i)$ to $\mathsf{3HVZK.Verify}$.
        ∗ endElse
    • endFor.
– [Acceptance condition] $\mathsf{3HVZK.Verify}(y) \to \{0, 1\}$.
      For each $i \in [p(\lambda)]$, do
        ∗ If $a_i \neq \mathcal{RO}(y, i)$ then return $0$.
        ∗ If $z_i = (\perp, \pi_i)$ do
            · If $\mathsf{Verify}(y, \pi_i) = 1$ then return $0$.
        ∗ endIf
        ∗ else
            · If $\mathsf{SpecP.Verify}(y, a_i, e_i, z_i) = 0$ then return $0$.
        ∗ endElse
        ∗ return $1$.
    • endFor.
```

7.2 Step II: Composing with the FS Transform

$\mathsf{Trans}(c(\cdot), k(\cdot), q, m(\cdot)p(\cdot), f)$ converts a spec-prot $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ with space of bad commitments of cardinality $\leq 2^{b(\cdot)}$, commitments of length $c(\cdot)$, challenges of length $k(\cdot)$ into a proof system in the RO model $\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f] = (\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f].\mathsf{Prove}, \mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f].\mathsf{Verify})$ with commitments of length $c(\lambda) \cdot p(\lambda)$, space of bad commitments of cardinality $2^{\lambda + \log(p(\lambda))}$ and challenges of length $k(\lambda) \cdot p(\lambda)$. The protocol is associated with a polynomial $\mathsf{poly}_{\mathsf{inp}}(\cdot)$ that dictates the domain of the RO.

By appropriately setting the parameter $p(\cdot)$ and applying the FS transform to 3HVZK we can obtain a NIZK proof system with negligible soundness error (precisely, $p(\cdot)$ and the soundness error will be related). We now show our main transform that uses the previous one and the FS transform to achieve our goal.

Construction 2 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a spec-prot with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q. Our main transform $\mathsf{Trans}_{\mathsf{main}}(c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f)$ is a polynomial-time algorithm that takes as input the description of f (and thus implicitly SpecP), the description of functions $c(\cdot), k(\cdot)$, $q, m(\cdot)$ and a negligible function $\delta(\cdot)$ and outputs a pair
$(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{poly}_{\mathsf{out}}(\cdot), \mathsf{NIZK}[c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f])$ that consists of the description of two polynomials $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{poly}_{\mathsf{out}}(\cdot))$ and the description of a NIZKPoK proof system computed as follows.

Firstly, compute a polynomial $p(\cdot)$ satisfying the equation

$$2^{2\cdot\lambda+\log(p(\lambda))} \cdot \left(\frac{1}{q} + (1-\frac{1}{q}) \cdot \frac{1}{k(\lambda)}\right)^{p(\lambda)} \leq \delta(\lambda). \tag{1}$$

We will show in Theorem 10 that it is always possible to find such a polynomial.[7]

Then, apply the transform $\mathsf{Trans}(c(\cdot), k(\cdot), q, m(\cdot)p(\cdot), f)$ of construction 1 to obtain a 3-round public-coin HVZK proof system in the RO model $\mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f]$ and a polynomial $\mathsf{poly}'_{\mathsf{inp}}(\cdot)$. Set $\mathsf{poly}_{\mathsf{inp}}(\cdot)$ (resp. $\mathsf{poly}_{\mathsf{out}}(\cdot)$) to the maximum between $\mathsf{poly}'_{\mathsf{inp}}(\cdot)$ and the length of the commitments of 3HVZK (resp. maximum between the length of the commitments and the length of the challenges of 3HVZK).

(In the following we assume that, e.g., if 3HVZK was expecting an RO with domain $\{0,1\}^{m(\lambda)}$ and we execute with an RO with domain $\{0,1\}^{n(\lambda)}$, for $n(\lambda) > m(\lambda)$, the protocol 3HVZK is slightly modified to use the truncation of the output of the RO; similarly for the co-domain. Thus, the previous setting serves to guarantee that the RO has domain and co-domain enough large to be used both for the transform Trans (that uses domain $\{0,1\}^{\lambda+\log((p(\lambda))}$ and co-domain $c(\lambda)$) and the FS transform that uses domain $\{0,1\}^{\lambda+c(\lambda)\cdot p(\lambda)}$ and co-domain $\{0,1\}^{c(\lambda)\times p(\lambda)}$).

Then it applies the FS transform to 3HVZK to get a NIZKPoK proof system $\mathsf{NIZK} = (\mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ that uses an RO with domain (resp. co-domain) strings of length $\mathsf{poly}_{\mathsf{inp}}(\cdot)$ (resp. $\mathsf{poly}_{\mathsf{out}}(\cdot)$).

Note that our main transform $\mathsf{Trans}_{\mathsf{main}}$ can be viewed as the composition of Trans with the FS transform.

Remark 1 By defining $\mathsf{Trans}_{\mathsf{main}}$ to be the composition of the two transforms (i.e., Trans and the FS transform), for simplicity we skipped a detail. Namely, the proof system 3HVZK on which we apply the FS transform is a protocol for the RO model and thus care has to be taken in avoiding that the *added* RO queries are in the set of possible RO queries of the original protocol. This issue can be sorted out by letting the RO in the original protocol and in the FS-transformed

[7] Specifically, it does not hold for all negligible functions but does hold for functions like $2^{-c\cdot\lambda}$ for some constant $c > 0$.

protocol to query the RO on different prefixes, e.g., 0 and 1; that is, each query x of $\mathsf{3HVZK}$ (resp. each new query added by the FS transform) will invoke the RO on input $(0||x)$ (resp. $(1||x)$).

Next, we define the instantiation of a NIZKPoK resulting from our transform with a concrete hash function.

Construction 3 [H-instantiation of our transform] Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{poly}_{\mathsf{out}}(\cdot), \mathsf{NIZK}[\mathsf{3HVZK}, c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot)]) = \mathsf{Trans}(\mathsf{3HVZK}, c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot))$ be the NIZKPoK system resulting from the transform of Construction 1. Let $H(\cdot)$ be any function with domain $\{0,1\}^\star$ and co-domain $\{0,1\}^m$ for some integer $m > 0$.

We denote by $\mathsf{Trans}^{H(\cdot),m}_{\mathsf{main}}(\mathsf{3HVZK}, c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot))$ be the NIZKPoK system resulting from the transform of Construction 1 changed as follows. (In the following we assume for simplicity that $\mathsf{poly}_{\mathsf{out}}(\lambda)$ divides m. It is straightforward to remove the constraint.) When the prover (resp. verifier) needs to access the oracle $\mathcal{RO}(\cdot)$ on an input $y \in \{0,1\}^{\mathsf{poly}_{\mathsf{inp}}(\lambda)}$, the function $H(\cdot)$ is invoked on inputs $H(1^1||0||y), \ldots, H(1^{\mathsf{poly}_{\mathsf{out}}(\lambda)/m}||0||y)$ to get respective outputs $e_1, \ldots, e_{\mathsf{poly}_{\mathsf{out}}(\lambda)/m}$ and the concatenation of the e_i's as the oracle's answer is returned to the prover (resp. verifier).

With a slight abuse of notation, we call the output of $\mathsf{Trans}^{H(\cdot),m}$ the instantiation of the proof system with function $H(\cdot)$.

Acknowledgments

Research supported in part by the European Unions Horizon 2020 research and innovation programme under grant agreement No 780477 (project PRIViLEDGE) and by an FNR CORE-Junior grant of the Luxembourg National research fund (project no. C16/IS/11299247).

References

AABN02. Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 418–433. Springer, April / May 2002.

AABN08. Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the fiat-shamir transform: Necessary and sufficient conditions for security and forward-security. *IEEE Transactions on Information Theory*, 54(8):3631–3646, Aug 2008.

ABB$^+$10. José Bacelar Almeida, Endre Bangerter, Manuel Barbosa, Stephan Krenn, Ahmad-Reza Sadeghi, and Thomas Schneider. A certifying compiler for zero-knowledge proofs of knowledge based on sigma-protocols. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *ESORICS 2010: 15th European Symposium on Research in Computer Security*, volume 6345 of *Lecture Notes in Computer Science*, pages 151–167. Springer, September 2010.

AF07. Masayuki Abe and Serge Fehr. Perfect NIZK with adaptive soundness. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 118–136. Springer, February 2007.

AMS$^+$09. Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Berk Sunar, and Pim Tuyls. Memory leakage-resilient encryption based on physically unclonable functions. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 685–702. Springer, December 2009.

Bar01. Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115. IEEE Computer Society Press, October 2001.

BCNP04. Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *45th Annual Symposium on Foundations of Computer Science*, pages 186–195. IEEE Computer Society Press, October 2004.

BDSG$^+$13. Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. Why "fiat-shamir for proofs" lacks a proof. In *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013.*, pages 182–201. Springer, 2013.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th Annual ACM Symposium on Theory of Computing*, pages 103–112. ACM Press, May 1988.

BFS16. Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. Nizks with an untrusted CRS: security in the face of parameter subversion. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, pages 777–804, 2016.

BFSK11. Christina Brzuska, Marc Fischlin, Heike Schröder, and Stefan Katzenbeisser. Physically uncloneable functions in the universal composition framework. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 51–70. Springer, August 2011.

BFW15. David Bernhard, Marc Fischlin, and Bogdan Warinschi. Adaptive proofs of knowledge in the random oracle model. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 629–649, 2015.

BG93. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume

740 of *Lecture Notes in Computer Science*, pages 390–420. Springer, August 1993.

BLV03. Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. In *44th Annual Symposium on Foundations of Computer Science*, pages 384–393. IEEE Computer Society Press, October 2003.

BM88. László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

BN06. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 390–399. ACM Press, October / November 2006.

BPW12. David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 626–643. Springer, December 2012.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, November 1993.

BR08. Mihir Bellare and Todor Ristov. Hash functions from sigma protocols and improvements to VSH. In Josef Pieprzyk, editor, *Advances in Cryptology – ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 125–142. Springer, December 2008.

BY96. Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *Journal of Cryptology*, 9(3):149–166, 1996.

CDS94. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *Advances in Cryptology – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, August 1994.

CG15. Pyrros Chaidos and Jens Groth. Making sigma-protocols non-interactive without random oracles. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 650–670, 2015.

CGH98. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218. ACM Press, May 1998.

CP93. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer, August 1993.

CPS$^+$16. Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Online/offline OR composition of sigma protocols. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 63–92, 2016.

CPSV16. Michele Ciampi, Giuseppe Persiano, Luisa Siniscalchi, and Ivan Visconti. A transform for NIZK almost as efficient and general as the fiat-shamir transform without programmable random oracles. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 83–111, 2016.

CS98. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, August 1998.

CS03. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

Dam10. Ivan Damgård. On Σ-protocol. `http://www.cs.au.dk/~ivan/Sigma.pdf`, 2010.

DFN06. Ivan Damgård, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 41–59. Springer, March 2006.

DG03. Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *35th Annual ACM Symposium on Theory of Computing*, pages 426–437. ACM Press, June 2003.

DN00. Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st Annual Symposium on Foundations of Computer Science*, pages 283–293. IEEE Computer Society Press, November 2000.

DNRS99. Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th Annual Symposium on Foundations of Computer Science*, pages 523–534. IEEE Computer Society Press, October 1999.

DORS08. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

DRV12. Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 618–635. Springer, March 2012.

Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 152–168. Springer, August 2005.

FKI06. Jun Furukawa, Kaoru Kurosawa, and Hideki Imai. An efficient compiler from sigma-protocol to 2-move deniable zero-knowledge. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 46–57. Springer, July 2006.

FKMV12. Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012: 13th International Conference in Cryptology in India*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79. Springer, December 2012.

FLS90. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science*, pages 308–317. IEEE Computer Society Press, October 1990.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, August 1987.

GCvD02. Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In Vijayalakshmi Atluri, editor, *ACM CCS 02: 9th Conference on Computer and Communications Security*, pages 148–160. ACM Press, November 2002.

GGH$^+$13. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49. IEEE Computer Society Press, October 2013.

GIS$^+$10. Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 308–326. Springer, February 2010.

GK03. Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th Annual Symposium on Foundations of Computer Science*, pages 102–115. IEEE Computer Society Press, October 2003.

GKR08. Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 39–56. Springer, August 2008.

GM84. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

GMO16. Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 1069–1083, 2016.

GMR89. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

GMY06. Juan A. Garay, Philip D. MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. *Journal of Cryptology*, 19(2):169–209, April 2006.

Gol01. Oded Goldreich. *Foundations of Cryptography: Basic Techniques*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

GOS06a. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 97–111. Springer, August 2006.

GOS06b. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358. Springer, May / June 2006.

GOSV14. Vipul Goyal, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Black-box non-black-box zero knowledge. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 515–524. ACM Press, May / June 2014.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, April 2008.

HL08. Carmit Hazay and Yehuda Lindell. Constructions of truly practical secure protocols using standardsmartcards. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08: 15th Conference on Computer and Communications Security*, pages 491–500. ACM Press, October 2008.

Kat07. Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 115–128. Springer, May 2007.

KRR16. Yael T. Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of fiat-shamir for proofs. *IACR Cryptology ePrint Archive*, 2016:303, 2016.

Lin06. Yehuda Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *Journal of Cryptology*, 19(3):359–377, July 2006.

Lin15. Yehuda Lindell. An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 93–109, 2015.

Mau15. Ueli Maurer. Zero-knowledge proofs of knowledge for group homomorphisms. *Des. Codes Cryptography*, 77(2-3):663–676, 2015.

MP03. Daniele Micciancio and Erez Petrank. Simulatable commitments and efficient concurrent zero-knowledge. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 140–159. Springer, May 2003.

MV16. Arno Mittelbach and Daniele Venturi. Fiat-shamir for highly sound protocols is instantiable. In *Security and Cryptography for Networks - 10th International Conference, SCN 2016, Amalfi, Italy, August 31 - September 2, 2016, Proceedings*, pages 198–215, 2016.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437. ACM Press, May 1990.

OPV10. Rafail Ostrovsky, Omkant Pandey, and Ivan Visconti. Efficiency preserving transformations for concurrent non-malleable zero knowledge. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 535–552. Springer, February 2010.

OSVW13. Rafail Ostrovsky, Alessandra Scafuro, Ivan Visconti, and Akshay Wadia. Universally composable secure computation with (malicious) physically uncloneable functions. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 702–718. Springer, May 2013.

Pas03. Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 316–337. Springer, August 2003.

Pas13. Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 334–354. Springer, March 2013.

PRTG02. Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

Ps05. Rafael Pass and Abhi shelat. Unconditional characterizations of non-interactive zero-knowledge. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 118–134. Springer, August 2005.

PsV06. Rafael Pass, abhi shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 271–289. Springer, August 2006.

RS92. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, August 1992.

RSA78. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978.

RvD13. Ulrich Rührmair and Marten van Dijk. PUFs in security protocols: Attack models and security evaluations. In *2013 IEEE Symposium on Security and Privacy*, pages 286–300. IEEE Computer Society Press, May 2013.

Sah99. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society Press, October 1999.

SG02. Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, 15(2):75–96, 2002.

TSS$^+$05. Pim Tuyls, B. Skoric, S. Stallinga, Anton H. M. Akkermans, and W. Ophey. Information-theoretic security analysis of physical uncloneable functions. In Andrew Patrick and Moti Yung, editors, *FC 2005: 9th International Conference on Financial Cryptography and Data Security*, volume 3570 of *Lecture Notes in Computer Science*, pages 141–155. Springer, February / March 2005.

VV09. Carmine Ventre and Ivan Visconti. Co-sound zero-knowledge with public keys. In Bart Preneel, editor, *AFRICACRYPT 09: 2nd International Conference on Cryptology in Africa*, volume 5580 of *Lecture Notes in Computer Science*, pages 287–304. Springer, June 2009.

YZ06. Moti Yung and Yunlei Zhao. Interactive zero-knowledge with restricted random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 21–40. Springer, March 2006.

YZ07. Moti Yung and Yunlei Zhao. Generic and practical resettable zero-knowledge in the bare public-key model. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 129–147. Springer, May 2007.

Supplementary Material

A Definitions and Building Blocks

Notation. We use $\mathbb{N}$ to denote the set of all natural numbers. For any natural number m, we let U_m stand for the uniform distribution over binary strings of length m. A *negligible* function $\mathsf{negl}(\lambda)$ is a function that is smaller than the inverse of any polynomial in λ (starting from a certain point). We denote by $[n]$ the set of numbers $\{1, \ldots, n\}$, by $|x|$ the bit length of $x \in \{0,1\}^\star$ and by $x||y$ the concatenation of any two strings x and y in $\{0,1\}^\star$. For any integer $m > 0$, we denote by U_m the uniform distribution over $\{0,1\}^m$.

When we invoke a function with domain $\{0,1\}^s$ on input a string of length shorter than s, we implicitly mean that the input is padded with a sufficient number of 0's.

We let PPT stand for probabilistic polynomial time and EPT for expected polynomial time. Unless otherwise specified, all our adversaries are modelled as non-uniform PPT algorithms. For a probabilistic algorithm A, $A(x)$ denotes the probability distribution of the output of A when run with x as input. We use $A(x;r)$ instead to denote the output of A when run on input x and coin tosses r.

A *polynomial-time* relation $\mathcal{R}$ is a relation for which membership of (x, w) in $\mathcal{R}$ can be decided in time polynomial in $|x|$. If $(x, w) \in \mathcal{R}$ then we say that w is a *witness* for *instance* x. A polynomial-time relation $\mathcal{R}$ is naturally associated with the NP language $L_\mathcal{R}$ defined as $L_\mathcal{R} = \{x \mid \exists w : \ (x, w) \in \mathcal{R}\}$. Similarly, an NP language is naturally associated with a polynomial-time relation. Following [GMY06], we define $\hat{L}_\mathcal{R}$ to be the *input language* that includes both $L_\mathcal{R}$ and all well formed instances that do not have a witness. It follows that $L_\mathcal{R} \subseteq \hat{L}_\mathcal{R}$ and membership in $\hat{L}_\mathcal{R}$ can be tested in polynomial time. Given an NP language L, for any natural number $k > 0$, we denote by L_k the language $L \cap \{0,1\}^{\leq k}$.

Given two interactive machines M_0 and M_1, we denote by $\langle M_0(x_0), M_1(x_1)\rangle(x)$ the output of M_1 when running on input x_1 and interacting with M_0 running on input x_0 and common input x and by $\mathsf{view}_A\langle A(x_A), B(x_B)\rangle(x)$ the *view* of A during the interaction with B when both are executed on common input x and A (resp. B) is executed on input x_A (resp. x_B).

A.1 3-Round Public-Coin HVZK

In this section, we recall notions related to interactive proof systems.

Definition 1 [Interactive proof system [BM88,GMR89]] A pair $(\mathcal{P}, \mathcal{V})$ of PPT interactive machines is a *interactive proof system* for polynomial-time relation $\mathcal{R}$ associated with a language L if the following properties of completeness, soundness and high min-entropy of commitment hold:

- *Completeness.* For every $(x, w) \in \mathcal{R}$, it holds that:

$$\text{Prob}[\langle \mathcal{P}(w), \mathcal{V}\rangle(x) = 1] = 1.$$

- *Soundness.* For every non-uniform (possibly computationally unbounded) machine $\mathcal{P}^\star \stackrel{\triangle}{=} \{\mathcal{P}^\star_\lambda\}_\lambda$, it holds that for every polynomial $p(\cdot)$, there exists a constant n such that for every $\lambda \geq n$, for every $x \notin L, x \in \{0,1\}^{\leq \lambda}$, it holds that:
$$\mathrm{Prob}[\langle \mathcal{P}^\star_\lambda, \mathcal{V}\rangle(x) = 1] \leq 1/p(\lambda).$$
- *High min-entropy of commitment [AABN02,AABN08,FKMV12].* Consider a pair $(x, w) \in \mathcal{R}$ and let $\lambda = |x|$. Denote with $\mathsf{Coins}(\lambda)$ the set of coins used by $\mathcal{P}$ and consider the set $A(x, w) = \{\mathcal{P}(x, w; \rho) : \ \rho \leftarrow \mathsf{Coins}(\lambda)\}$ of all possible commitments associated to w. The min-entropy function associated to $(\mathcal{P}, \mathcal{V})$ is defined as $\epsilon(\lambda) = \min_{x,w}(-\log_2 \mu(x, w))$, where the minimum is taken over all possible $(x, w) \in \mathcal{R}$ with $|x| = \lambda$ and $\mu(x, w)$ is the maximum probability that a commitment takes on a particular value, i.e., $\mu(x, w) = \max_{\alpha \in A(x,w)}(\Pr[\mathcal{P}(x, w; \rho) = \alpha : \ \rho \leftarrow \mathsf{Coins}(\lambda))]$.
We require that $\epsilon(\lambda) \in \omega(\log(\lambda))$, i.e., it is super-logarithmic in λ.

Note that the high min-entropy of commitment condition is non-standard in the definitions of interactive proof systems but as it will be needed in our work, we prefer to subsume it in our definition. It will be only necessary to prove the ZK of the NIZK systems resulting from our transform.

The soundness can be weakened to $s(\cdot)$-soundness as follows.

Definition 2 [$s(\cdot)$-soundness] Let $s(\cdot)$ be a function. An interactive proof system $(\mathcal{P}, \mathcal{V})$ for polynomial-time relation $\mathcal{R}$ associated with a language L satisfies $s(\cdot)$-soundness if the following holds. For every non-uniform (possibly computationally unbounded) machine $\mathcal{P}^\star \stackrel{\triangle}{=} \{\mathcal{P}^\star_\lambda\}_\lambda$, it holds that there exists a constant n such that for every $\lambda \geq n$, for every $x \notin L \cap \{0,1\}^{\lambda \leq \lambda}$, it holds that:

$$\mathrm{Prob}[\langle \mathcal{P}^\star_\lambda, \mathcal{V}\rangle(x) = 1] \leq 1/s(\lambda).$$

Definition 3 [Space of bad commitments] Let $\mathsf{3HVZK} = (\mathcal{P}, \mathcal{V})$ be a interactive proof system for a polynomial-time relation $\mathcal{R}$ associated with a language L. Given an $x \notin L$, we denote by "space of bad commitments" S_x for x of a 3-round public-coin proof system the set of all commitments a such that there exist e, z such that $\mathcal{V}(x, a, e, z)$ is accepted by the verifier. With a slight abuse of notation, we say that the space of bad commitments S of $\mathsf{3HVZK}$ has cardinality $\leq N$ if for all $x \notin L$, the cardinality of S_x is $\leq N$.

Definition 4 [Computational and statistical honest verifier zero-knowledge] A proof system for a polynomial-time relation $\mathcal{R}$ consisting of a pair $(\mathcal{P}, \mathcal{V})$ of PPT interactive machines is called a computational (resp. statistical) *honest verifier zero-knowledge* (HVZK, in short) if it satisfies the following computational (resp. statistical) honest verifier zero-knowledge property.

- *Computational Honest Verifier Zero-Knowledge:* There exists a PPT algorithm Sim (called the *simulator* for $\mathcal{V}$) such that or for every sequence $\{(x_\lambda, w_\lambda)\}_{\lambda>0}$ such that for every $\lambda > 0$, $(x_\lambda, w_\lambda) \in \mathcal{R}$, for every polynomial

$p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, no non-uniform PPT distinguisher algorithm can distinguish the following two sequences of random variables with advantage $> 1/p(\lambda)$:

- $\{\mathsf{view}_{\mathcal{V}} \langle \mathcal{P}(w_\lambda; U_m), \mathcal{V} \rangle (x_\lambda)\}_{\lambda \geq n}$. (Where m is the number of random coins $\mathcal{P}$ uses).
- $\{\mathsf{Sim}(x_\lambda)\}_{\lambda \geq n}$.

Statistical Honest Verifier Zero-Knowledge: This is identical to computational honest verifier zero-knowledge except that it is quantified for every non-uniform (possibly computationally unbounded) distinguisher algorithms.

Definition 5 [Witness indistinguishable proof system] A proof system for a polynomial-time relation $\mathcal{R}$ consisting of a pair $(\mathcal{P}, \mathcal{V})$ of PPT interactive machines is called witness indistinguishable (WI, in short) if it satisfies the following property.

– *Witness indistinguishability (WI, in short):* Let L be the language associated with $\mathcal{R}$.
  For every non-uniform PPT verifier $\mathcal{V}^\star \overset{\triangle}{=} \{\mathcal{V}^\star_\lambda\}_\lambda$, for every two sequences $\{(x_\lambda, w^1_\lambda)\}_{\lambda > 0}$, $\{(x_\lambda, w^2_\lambda)\}_{\lambda > 0}$ such that for every $\lambda, x_\lambda \in L_\lambda, (x_\lambda, w^1_\lambda) \in \mathcal{R}$ and $(x_\lambda, w^2_\lambda) \in \mathcal{R}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, no non-uniform PPT distinguisher algorithm can distinguish the following two sequences of random variables with advantage more than $1/p(\lambda)$:
  - $\{\mathsf{view}_{\mathcal{V}^\star_\lambda} \langle \mathcal{P}(w^1_\lambda; U_m), \mathcal{V}^\star_\lambda \rangle (x_\lambda)\}_{\lambda \geq n}$.
  - $\{\mathsf{view}_{\mathcal{V}^\star_\lambda} \langle \mathcal{P}(w^2_\lambda; U_m), \mathcal{V}^\star_\lambda \rangle (x_\lambda)\}_{\lambda \geq n}$.
    (Where m is the number of random coins $\mathcal{P}$ uses).

The main results of this paper will concern interactive proof systems $(\mathcal{P}, \mathcal{V})$ with at least three rounds of interaction with $\mathcal{P}$ sending the first message and with $\mathcal{V}$'s only message consisting solely of coin tosses. These pairs are called *public-coin protocols* [BM88] and have been object of intensive studies.

For simplicity and for not overburdening the presentation, we will focus on the special case of protocols executing in exactly three rounds, but we will later show how our results can be generalized further. Therefore, unless otherwise specified, whenever we say a proof system, we mean a *three-round public-coin interactive proof system.* This class includes Σ-protocols [CDS94], that are widely used in practice, have been designed for all useful languages and, moreover, they are easy to work with as already shown in transforms [DG03,MP03,YZ07,OPV10,Lin15,CPSV16].

We usually denote the *transcript* of an execution of a proof system $(\mathcal{P}, \mathcal{V})$ by a triple of messages (a, c, z), where a and z are sent by $\mathcal{P}$ and c, the *challenge*, is $\mathcal{V}$'s only message. We say that a transcript is *accepting* if $\mathcal{V}$ outputs 1.

Definition 6 [Proof system] A proof system $(\mathcal{P}, \mathcal{V})$ is a *Σ-protocol* for polynomial-time relation $\mathcal{R}$ if it enjoys the following properties:

– *Completeness.* For every $(x, w) \in \mathcal{R}$, it holds that

$$\mathrm{Prob}[\langle \mathcal{P}(w), \mathcal{V} \rangle (x) = 1] = 1.$$

- *Special Soundness.* There exists a PPT algorithm $\mathsf{Extract}$ that, on input x and any pair of accepting conversations for x, $(a, c, z), (a, c', z')$, where $e \neq e'$, outputs w such that $(x, w) \in R$.
- *Special Honest Verifier Zero Knowledge (SHVZK).* There exists a PPT *simulator* algorithm Sim that, on input an instance $x \in L$ and a challenge c, outputs (a, z) such that (a, c, z) has the same distribution of transcripts obtained when $\mathcal{V}$ sends c as challenge and $\mathcal{P}$ runs on common input x and any private input w such that $(x, w) \in \mathcal{R}$.

We also stress that SHVZK as defined above corresponds to the notion of *Perfect* SHVZK as distinct from *Computational* SHVZK. This latter notion has also been studied in the literature in the context of Σ-protocols [GMY06] but it will not be considered in this paper.

SHVZK is a weaker requirement than Zero Knowledge; nonetheless, it implies non-trivial security against adversarial verifiers.

Theorem 1 [[CDS94]] Let Π be a proof system that enjoys completeness and SHVZK for relation $\mathcal{R}$. Then Π is Perfect WI.

In a Σ-protocol security for $\mathcal{P}$ is unconditional. The following result implies instead that the challenge length acts as a security parameter for $\mathcal{V}$.

Theorem 2 Let Π be a proof system for polynomial-time relation $\mathcal{R}$ that is special sound.

Then Π is a proof of knowledge with knowledge error negligible in the challenge length.

Proof. Based on [Dam10].

The following theorem says that the challenge length can be increased by simple parallel repetition.

Theorem 3 [CDS94,Dam10] Let Π be a Σ-protocol for polynomial-time relation $\mathcal{R}$ with challenge length l. The k-wise parallel composition of Π is a Σ-protocol for $\mathcal{R}$ with challenge length $k \cdot l$.

3-Round Public-Coin HVZK in the RO model In the first step of our transform we convert a 3-round public-coin HVZK protocol into a 3-round public-coin HVZK protocol in the (programmable) RO model.

Definition 7 [3-round public-coin HVZK protocol in the RO model] A 3-round public-coin HVZK protocol in the programmable RO model is a 3-round public-coin HVZK protocol in which the prover and verifiers have both access to the RO and the ZK property (whether statistical or computational) is changed as follows. The simulator has access to a RO and is given the ability of programming the RO at any point of its choice. The distinguisher against the ZK property is given access to the RO modified by the simulator.

The following definition is the analogous one for proof systems (in the standard model). It is almost identical except that both parties have access to the RO. Observe that in this case it might happen (as it is the case in Trans of construction 1) that the commitments have length $n(\lambda)$ but they are computed as output of a RO with domain consisting of strings of length $m(\lambda) << n(\lambda)$. Moreover, the protocol may allow the verifier to check how the commitment is computed by the RO, and thus the space of bad commitments is dictated by the RO.

Definition 8 [Space of bad commitments for a proof system in the RO model] Let $\mathsf{3HVZK} = (\mathcal{P}, \mathcal{V})$ be a 3-round public-coin proof system in the RO model for a polynomial-time relation $\mathcal{R}$ associated with a language L. Given an $x \notin L$, we denote by "space of bad commitments" S_x for x of a 3-round public-coin proof system the set of all commitments a such that there exist e, z such that $\mathcal{V}^{\mathcal{RO}(\cdot)}(x, a, e, z)$ is accepted by the verifier. With a slight abuse of notation, we say that the space of bad commitments S of $\mathsf{3HVZK}$ has cardinality $\leq N$ if for all $x \notin L$, the cardinality of S_x is $\leq N$.

Definition 9 [High min-entropy of commitment for proof system in the RO model] The high min-entropy of commitment property for a proof system in the RO model is stated identically to the analogous property for proof systems (in the standard model) except that the probability is also taken over the choices of the RO and the prover is given access to the RO.

A.2 Special Functions and Special Protocols

We now define the class of protocols which our transform can be applied to.

Definition 10 [One-way homomorphic function] Consider two groups $(G, \cdot)$, $(H, *)$. A function $f : G \leftarrow H$ is a one-way homomorphic function if f is a OWF and for each $x_1, x_2 \in G$, $f(x_1 \cdot x_2) = f(x_1) * f(x_2)$.

In the following we consider family of groups $G = \{G_\lambda\}$ parameterized by the security parameter and functions acting on them. So, f will be a family of functions indexed by the security parameter, though for simplicity we will often write $f(G_\lambda)$ to refer to the set of all elements y such that there exists $x \in G_\lambda, f(x) = y$.

Definition 11 [Special one-way homomorphic function (SOWGHF)] Consider two family of groups $(G = \{G_\lambda\}_\lambda, \cdot)$, $(H = \{H_\lambda\}, *)$. A function $f : G \rightarrow H$ is a special one-way homomorphic function (SOWGHF) if f is a one-way homomorphic function and the following additional requirements hold.

1. Efficient representability. There is a polynomial $m(\cdot)$ (resp. $n(\cdot)$) such that for each λ, H_λ (resp. G_λ) can be described by $m(\lambda)$ (resp. $n(\lambda)$) bits and the group operations $\cdot$ and $*$ can be performed in polynomial-time.
   With slight abuse of notation, we call the set $\{0,1\}^{m(\lambda)}$ the *co-domain* of f and we distinguish it from the *range* of f in the following way. An element $y \in \{0,1\}^{m(\lambda)}$ is said to belong to the range of f and thus to H, in symbols

$f \in \mathsf{Range}(f)$, if there exists $x \in G$ such that $f(x) = y$. So, a string y can belong to the co-domain of f but not to the range of f. Likewise, we call $\{0,1\}^{n(\lambda)}$ the *domain* of f and we distinguish it from G whose elements can be represented by $n(\lambda)$ bits.

2. Trapdoor invertibility. f is a *trapdoor* OWF, that is there is a trapdoor trap and an efficient algorithm that with the help of trap can invert any string $y \in \mathsf{Range}(f(G_\lambda))$ for any value of the security parameter.
3. Membership decidability. The trapdoor for the function also allows to efficiently decide whether a string $y \in \{0,1\}^{m(\lambda)}$ is in the range of $f(G_\lambda)$ for any value of the security parameter.
4. Co-membership decidability. The language of all strings $y \notin \mathsf{Range}(f), y \in \{0,1\}^{m(\lambda)}$ is in $\mathsf{co\text{-}NP}$ and using the trapdoor trap for f it is possible to compute, for any $y \in \{0,1\}^{m(\lambda)}$, a witness for the fact that $y \notin \mathsf{Range}(f(G_\lambda))$. That is, there is an algorithm Prove that on input a string $y \in \{0,1\}^{m(\lambda)}$ and a trapdoor trap for f computing a proof π and an algorithm Verify that on input $y \in \{0,1\}^{m(\lambda)}$ and a proof π accepts if and only if $y \notin \mathsf{Range}(f(G_\lambda))$. Furthermore, there is a PPT simulator Sim_f that, with input the security parameter, outputs a pair (a, π) that is distributed identically to (a', π') where a'_i is selected at random in the space of strings $y \in \{0,1\}^{m(\lambda)}, y \notin \mathsf{Range}(f)$ and $\pi' \leftarrow \mathsf{Prove}_f(y, \mathsf{trap})$.
5. Quasi-compactness. There is a constant $q > 1$ such that the probability p that a random element in $\{0,1\}^{m(\lambda)}$ falls outside the range of $f(G_\lambda)$ is $\frac{1}{q}$ up to a factor $\leq \pm 2^{-c\cdot\lambda}$ for some constant $c > 0$. It is also possible to efficiently sample a binary variable that equals 0 with probability p, up to a negligible error in λ.

We say that f is a $(m(\cdot), q)$-SOWGHF if f is a OWGHF and the functions in the first and last conditions are fixed, resp., to $m(\cdot)$ and q.

Definition 12 [Special relation for a SOWGHF] Let f be a SOWGHF. Let $\mathcal{R}_f(y, (x, \mathsf{trap}))$ be the polynomial-time relation that holds if and only if $x \in G_\lambda$, $y \in \{0,1\}^{m(\lambda)}$ for some λ and $y = f(x)$ (i.e., the relation holds if and only if y is in the range of f). $\mathcal{R}_f$ is called the special relation for f.

By abstracting several known protocols in the literature, Maurer presents a sigma protocol for proving that an element $y \in \mathsf{Range}(f)$ for a one-way homomorphic function f. A special protocol ($\mathsf{spec\text{-}prot}$) has the same pattern as in Maurer but it is associated with a SOWGHF (a strengthening of a one-way homomorphic group function).

Definition 13 [Special protocol ($\mathsf{spec\text{-}prot}$)] Let $(G, \cdot)$ and $(H, *)$ be two family of groups and $f : G \leftarrow H$ be a SOWGHF with associated trapdoor trap. Let $\mathcal{R}_f$ be the special relation for f. Let $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ with each $\mathcal{C}_\lambda$ set of integers of the same bit length. A $\mathsf{spec\text{-}prot}$ for f with challenge space $\mathcal{C}$ is the following Σ-protocol for $\mathcal{R}_f$.

- The prover knows x, y and trap and the verifier knows y.

- In the first round, the prover select a random element r in G_λ and sends the commitment $a = f(r)$ to the verifier.
- In the second round, The verifier sends back a number e selected at random in $\mathcal{C}_\lambda$.
- The prover sends the answer $z = r \cdot x^e$ to the verifier.
- The verifier accepts if and only if $f(z) = a * y^e$.

We say that SpecP is a $\mathsf{spec\text{-}prot}$ if it is a $\mathsf{spec\text{-}prot}$ for some SOWGHF.

Note that, given a SOWGHF and a challenge space $\mathcal{C}$, a $\mathsf{spec\text{-}prot}$ for f with challenge space $\mathcal{C}$ is completely determined and that such protocol is for the polynomial-time relation $\mathcal{R}_f$. In our work, we will often implicitly set as challenge space the set of all strings of some length $k(\lambda)$ and we will interpret such strings as integers when used as powers of group elements.

Thus, the following theorem follows straightforward from the results of Maurer [Mau15].

Theorem 4 Let f be a SOWGHF, and SpecP a $\mathsf{spec\text{-}prot}$ for f. Then, SpecP is a Σ-protocol for $\mathcal{R}_f$.

A.3 NIZKA/NIZKP in the RO model

Let R be an efficiently computable binary relation. For pairs $(x, w) \in R$ we call x the statement and w the witness. Let L be the language consisting of statements in R.

Definition 14 [NIZKA] A non-interactive zero-knowledge argument system (NIZKA, in short) NIZK in the programmable RO model (see [BR93,FS87,FKMV12,BFW15]) for a relation R consists of the following PPT algorithms with access to an oracle $\mathcal{RO}$ drawn uniformly at random from a space $\mathsf{ROSp}(\lambda)$ of functions with domain $\{0,1\}^{\mathsf{poly}_{\mathsf{inp}}(\lambda)}$ and co-domain $\{0,1\}^{\mathsf{poly}_{\mathsf{out}}(\lambda)}$, for some polynomials $\mathsf{poly}_{\mathsf{inp}}$ and $\mathsf{poly}_{\mathsf{out}}$ that are part of the specification of the system:

- $\mathsf{Prove}^{\mathcal{RO}(\cdot)}(x, w)$: this is a PPT algorithm that takes as input a statement x and a witness w for x, and with oracle access to O produces a proof π.
- $\mathsf{Verify}^{\mathcal{RO}(\cdot)}(x, \pi)$: this is a deterministic polynomial-time algorithm that takes as input a statement x and a proof π, and with oracle access to O outputs 1 if the proof is accepted and 0 otherwise.

We call NIZK a non-interactive zero-knowledge argument system for R if it has the properties described below.

- $\mathsf{Statistical\ Completeness}$. An argument system is statistically complete if an honest prover with a valid witness can convince an honest verifier with overwhelming probability over the choices of the RO. Formally we have that for every $(x, w) \in R$, such that $|x| = \lambda$ it holds that:

  $\Pr[\mathcal{RO} \leftarrow \mathsf{ROSp}(\lambda);\ \pi \leftarrow \mathsf{Prove}^{\mathcal{RO}(\cdot)}(x, w) : \mathsf{Verify}^{\mathcal{RO}(\cdot)}(x, \pi) = 1] \in 1 - \mathsf{negl}(\lambda).$

- $\mathsf{Computational\ Soundness}$. A non-interactive argument system is computational sound if it is infeasible to convince an honest verifier when the statement is false. More formally, for all non-uniform PPT adversaries $\mathcal{A}$ we have:

  $$\Pr[\mathcal{RO} \leftarrow \mathsf{ROSp}(\lambda);\ (x, \pi) \leftarrow \mathcal{A}^{\mathcal{RO}}(1^\lambda) : \mathsf{Verify}^{\mathcal{RO}(\cdot)}(x, \pi) = 1 \wedge x \notin L \wedge |x| = \lambda] \in \mathsf{negl}(|x|)\ .$$

- $\mathsf{(Adaptive\ Multi\text{-}theorem)\ Computational\ zero\text{-}knowledge}$ $\mathsf{[FKMV12,BFW15]}$. A non-interactive argument system is computational zero-knowledge if the proofs do not reveal any information about the witnesses to a bounded adversary. We say a non-interactive argument NIZK is (adaptive multi-theorem) computational zero-knowledge if there exists a PPT *stateful* simulator $\mathsf{Sim} = (\mathsf{Sim}.\mathcal{RO}, \mathsf{Sim})$ that without access to the witness can simulate proofs having in addition the capability of programming the oracle $\mathcal{RO}$ at any point, i.e, for any x and y it is able to set $\mathcal{RO}(x) \stackrel{\triangle}{=} y$. Precisely, there exists a PPT stateful simulator $\mathsf{Sim} = (\mathsf{Sim}.\mathcal{RO}, \mathsf{Sim})$ such that for all non-uniform PPT adversaries $\mathcal{A}$ with access to an oracle $\mathcal{RO}$, we have that the following quantity is negligible in λ:

  $$\begin{aligned}&|\Pr[\mathcal{RO} \leftarrow \mathsf{ROSp}(\lambda) : \mathcal{A}^{\mathcal{RO}(\cdot), \mathsf{Prove}_2^{\mathcal{RO}(\cdot)}(\cdot,\cdot)}(1^\lambda) = 1] -\\ &\Pr[\mathcal{RO} \leftarrow \mathsf{ROSp}(\lambda) : \mathcal{A}^{\mathsf{Sim}.\mathcal{RO}(\cdot), \mathsf{Sim}_2(\cdot,\cdot)}(1^\lambda) = 1]|\ ,\end{aligned}$$

  where $\mathsf{Prove}_2^{\mathcal{RO}(\cdot)}(x, w) \stackrel{\triangle}{=} \mathsf{Prove}^{\mathcal{RO}(\cdot)}(x, w)$ for $(x, w) \in R$, $\mathsf{Sim}_2(x, w) \stackrel{\triangle}{=} \mathsf{Sim}(x)$ for $(x, w) \in R$, the latter oracles output $\perp$ for $(x, w) \notin R$ and $\mathsf{Sim}.\mathcal{RO}$ simulates the oracle O possibly modifying it at an arbitrary number of points.

Definition 15 [NIZKP] A non-interactive zero-knowledge proof system (NIZKP, in short) NIZK in the programmable RO model for a relation R is identical to a NIZKAoK except that the computational soundness is replaced by statistical soundness as follows.
$\mathsf{Statistical\ Soundness}$. A non-interactive proof system is statistically sound if it is infeasible to convince an honest verifier when the statement is false. More formally, for all non-uniform adversaries $\mathcal{A}$ we have:

$$\Pr[\mathcal{RO} \leftarrow \mathsf{ROSp}(\lambda);\ (x, \pi) \leftarrow \mathcal{A}^{\mathcal{RO}(\cdot)}(1^\lambda) : \mathsf{Verify}^{\mathcal{RO}(\cdot)}(x, \pi) = 1 \wedge x \notin L \wedge |x| = \lambda] \in \mathsf{negl}(\lambda)\ .$$

If in the above definition we quantify over non-uniform PPT adversaries running in time bounded by $s(\lambda)$, we talk about statistical $s(\cdot)$-soundness.

Note that in our formulation both of interactive systems and non-interactive ones, sometimes the security parameter λ is defined implicitly as $|x|$.

B Instantiations of SOWGHFs

In this section, we provide several examples of SOWGHFs.

Square function modulo a Blum integer. Let $N = p \cdot q$ be a Blum integer and let $|N|$ equal the security parameter λ.[8] Consider the groups $G = H = \mathbb{Z}_N^\star$ with multiplication modulo N as group operation for both and the function $f : G \to H, f(x) \stackrel{\triangle}{=} x^2 \mod N$ and let its domain and co-domain (cf. Def. 11) be the set of binary strings representing integers in $\mathbb{Z}_N$.

The function f is homomorphic since $f(x \cdot y) = x^2 \cdot y^2 \mod N = f(x) \cdot f(x) \cdot f(y) \mod N = f(x) \cdot f(y)$ and is conjectured to be a one-way function [Gol01].

We now verify that f satisfies the other properties of a SOWGHF under appropriate computational assumptions.

1. Efficient representability. The group operations can be represented efficiently and the group elements can be represented by $|N| = \lambda$ bits. Thus we assume that the integers in $\mathbb{Z}_N$ can be represented in the set $\{0, 1\}^\lambda$.
2. Trapdoor invertibility. The trapdoor trap for f is the factorization (p, q) of N. Given trap it is possible to compute $y_p = y \mod p, y_q = y \mod q$, compute one of their square roots and output one square root of y computed via the Chinese remainder theorem.
3. Membership decidability. The above trapdoor also allows to efficiently decide whether a string $y \in \mathbb{Z}_N$ is in the range of f. This follows from the observation that a number y is a quadratic residue modulo N if and only both $y \mod p$ and $y \mod q$ are quadratic residues and the latter can be efficiently checked.
4. Co-membership decidability. As N is a Blum integer, -1 is a quadratic non-residue modulo $\mathbb{N}$ and thus $-y$ is quadratic residue modulo N if and only if y is a quadratic non-residue modulo N. Thus, there exists a witness for proving that a number y is not a quadratic residue modulo N. Specifically, the algorithm Prove_f for f with input y and the trapdoor trap outputs one square root of $-y$ as proof π; and it is easy to see that such proof can be efficiently computed using trap.

   The simulator Sim_f for f works as follows. The simulator Sim_f picks a random number $r \leftarrow \mathbb{Z}_N$ and output $(-r^2 \mod N, r)$. It is easy to see that the output of the simulator has the same distribution of the of a pair (a', π') where a'_i is a random non-quadratic residue modulo N and π' is computed as before using Prove_f with input a_i and trap.
5. Quasi-compactness. From the previous observations it is easy to see that an integer selected at random in $\mathbb{Z}_N^\star$ is a quadratic non-residue with probability $\frac{3}{4}$ and an integer selected at random in $\mathbb{Z}_N$ is not in $\mathbb{Z}_N^\star$ with negligible probability. Thus, f satisfies quasi-compactness with parameter $\approx \frac{3}{4}$. It is also easy to efficiently sample, up to a negligible error, a random binary variable that equals 0 with the probability that a random integer in $\mathbb{Z}_N$ is a quadratic residue modulo N.

[8] Formally, we should define a family of moduli indexed by the security parameter. In the following of this section, we skip these details.

RSA function squared. A NIZKP for proving membership to the range of the following function can be employed to prove that an RSA "encryption" decrypts to the square of some message.

Let $N = p{\cdot}q$ be a Blum integer, let $|N|$ equal the security parameter λ and let e be co-prime with $\phi(N)$. Consider the groups $G = H = \mathbb{Z}_N^\star$ with multiplication modulo N as group operation for both and the function $f : G \to H, f(x) \overset{\triangle}{=} x^{2\cdot e} \mod N$ and let its domain and co-domain be the set of binary strings representing integers in $\mathbb{Z}_N$.

The function f is homomorphic since $f(x \cdot y) = x^{2\cdot e} \cdot y^{2\cdot e} \mod N = f(x) \cdot f(x) \cdot f(y) \mod N = f(x) \cdot f(y)$. The one-wayness of f can be reduced to the one-wayness of the RSA function [RSA78].

We now verify that f satisfies the other properties of a SOWGHF under appropriate computational assumptions.

1. Efficient representability. The group operations can be represented efficiently and the group elements can be represented by $|N| = \lambda$ bits.
2. Trapdoor invertibility. The trapdoor trap for f is the factorization (p, q) of N. Given trap it is possible to perform the following steps. Compute the inverse d of e modulo $\phi(N)$. Compute $z = y^e = x^2$ and then compute $y_p = z \mod p, y_q = z \mod q$, compute one of their square roots and output one square root $x'$of z computed via the Chinese remainder theorem.
3. Membership decidability. The above trapdoor also allows to efficiently decide whether a string $y \in \mathbb{Z}_N$ is in the range of f. This follows from the observation that a number y is a quadratic residue modulo N if and only both $y \mod p$ and $y \mod q$ are quadratic residues and the latter can be efficiently checked.
4. Co-membership decidability. As the function $f'(x) \overset{\triangle}{=} x^e \mod N$ is a permutation, it is easy to see that the property follows from the analogous property for the function square root modulo a Blum integer.
5. Quasi-compactness. As the function $f'(x) \overset{\triangle}{=} x^e \mod N$ is a permutation, it is easy to see that f satisfies quasi-compactness.

Trapdoor one-way homomorphic permutations. It is easy to verify that many natural trapdoor one-way permutations $f : G \to H$ that are homomorphic (e.g., the RSA permutation) are also a SOWGHF.

In fact, when the function is a permutation the properties of membership and co-membership decidability and quasi-compactness are trivially verified. Precisely, this holds when the elements of H can be represented by $m(\lambda)$ bits in a compact way, i.e., when all except a negligible fraction of the elements of $\{0, 1\}^\lambda$ do represent elements of H; for instance, the RSA function has this property.

When f is a permutation, there is no advantage in using our transform to prove that an element y is not in the range of f because in this case the soundness is trivially satisfied and the knowledge extraction property is also guaranteed by the FS transform with the same guarantees and at a lower cost.

Nevertheless, one might consider statements like $\exists x_1, x_2, x_3$ such that $((y_1 = f_1(x_1) \wedge y_2 = f_2(x_2)) \vee y_3 = f_3(x_3))$, where one or more of the functions f_1, f_2, f_3

are permutations and at least one is not a permutation and all the functions satisfy our requirements. Following Cramer *et al.* [CDS94], our transform can be likewise extended to support such compound statements.

C ZK

Lemma 5 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a polynomial $c(\cdot)$ and for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $(\mathsf{poly}_{\mathsf{inp}(\cdot)}, \mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f])$ be the output of transform $\mathsf{Trans}(c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f)$. Then, $\mathsf{3HVZK}$ is a 3-round public-coin proof system in the RO model satisfying HVZK and has high min-entropy of commitment.

Proof. Since the commitment of $\mathsf{3HVZK}$ are computed as output of a RO with range $c(\lambda)$ and $c(\cdot)$ is polynomial in λ, the high min-entropy of commitment property holds.

Let Sim_f the simulator for f guaranteed by Def. 11 and let $\mathsf{Sim}_{\mathsf{SpecP}}$ be the HVZK simulator guaranteed by Theorem 4. We now show a simulator Sim for the HVZK of $\mathsf{3HVZK}$.

Let $\mathbb{X}$ be a biased binary random variable that equals 0 with probability $\frac{1}{q}$ (and thus 1 with probability $1 - \frac{1}{q}$) guaranteed by Def. 11. (In the following, we skip the negligible error in sampling that can occur in sampling from $\mathbb{X}$.)

The simulator will keep a random table T representing the points in which it programs the RO. For each $i \in [p(\lambda)]$, Sim does the following.

- Sim draws a coin $b \leftarrow \mathbb{X}$.
- Case $b = 0$. If $b = 0$ then Sim uses $\mathsf{Sim}_{\mathsf{SpecP}}$ to compute (a_i, e_i, z_i) and sets $T[x||i] = a_i$.
- Case $b = 0$. If $b = 1$ then Sim does the following.
  - Draw a string e_i at random in $\{0,1\}^{k(\lambda)}$ and uses Sim_f to compute (a, π).
  - Set $a_i = a$ and $z_i = (\perp, \pi)$.
  - Set $T[x||i] = a_i$.

Finally, the simulator outputs $((a_1, \ldots, a_{p(\lambda)}), (e_1, \ldots, e_{p(\lambda)}), (z_1, \ldots, z_{p(\lambda)}))$ as simulated transcript and sets the RO accordingly to the table T.

We now argue that output of the simulator is statistically indistinguishable from the transcript of an execution with a real prover. In an honest transcript, each triple (a'_i, e'_i, z'_i) has probability $\frac{1}{q}$ to be such that $a'_i \notin \mathsf{Range}(f)$ and conditioned on this event, by construction of Trans and by hypothesis on Sim_f, the triple (a'_i, e'_i, z'_i) is perfectly indistinguishable from one computed by Sim_f.

By hypothesis on $\mathbb{X}$, for each $i \in [p(\lambda)]$, a triple (a_i, e_i, z_i) output by the simulator has probability $\frac{1}{q}$ of being such that $a_i \notin \mathsf{Range}(f)$. Then, for each $i \in [p(\lambda)]$, the distribution of the triples (a_i, e_i, z_i) output by the simulator such that $a_i \notin \mathsf{Range}(f)$ is identically distributed to the analogous triples in a transcript with a real prover.

In an honest transcript, each triple (a'_i, e'_i, z'_i) has probability $1 - \frac{1}{q}$ to be such that $a'_i \in \mathsf{Range}(f)$ and conditioned on this event, by construction of Trans and by hypothesis on $\mathsf{Sim}_{\mathsf{SpecP}}$, the triple (a'_i, e'_i, z'_i) is perfectly indistinguishable from one computed by $\mathsf{Sim}_{\mathsf{SpecP}}$.

By hypothesis on $\mathbb{X}$, for each $i \in [p(\lambda)]$, a triple (a_i, e_i, z_i) output by the simulator has probability $1 - \frac{1}{q}$ of being such that $a_i \in \mathsf{Range}(f)$. Then, for each $i \in [p(\lambda)]$, the distribution of the triples (a_i, e_i, z_i) output by the simulator such that $a_i \in \mathsf{Range}(f)$ is identically distributed to the analogous triples in a transcript with a real prover.

Therefore, it is easy to see that the distribution of the transcript output by the simulator is identically distributed to a transcript in an execution with a real prover and that the table T is also randomly distributed. So, the theorem follows.

Lemma 6 Let $\mathsf{3HVZK} = (\mathcal{P}, \mathcal{V})$ be a 3-round public-coin computational HVZK proof system in the RO model for polynomial-time relation $\mathcal{R}$ having commitments of length $c(\cdot)$, challenges of length $k(\cdot)$. Let $\mathsf{NIZK} = (\mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ be the NIZKP resulting from the FS transform on $\mathsf{3HVZK}$.

Then, NIZK satisfies (adaptive multi-theorem) computational zero-knowledge.

Proof (Sketch). The proof is only slightly different from the ones of [FKMV12,BFW15] for the FS transform and as such we sketch it.

We have to show that there exists a simulator $\mathsf{Sim} = (\mathsf{SIM}.\mathcal{RO}, \mathsf{Sim})$ that satisfies the computational zero-knowledge required in Definition 14. As for the FS case, Sim can invoke the simulator for $\mathsf{3HVZK}$. In particular, Sim works as follows.

- To answer query α to $\mathsf{Sim}.\mathcal{RO}$, the simulator samples a lookup table T kept in its internal state (recall that $\mathsf{Sim}.\mathcal{RO}$ and Sim are stateful algorithms communicating through an internal state). It checks whether $T(\alpha)$ is already defined. If this is the case, it returns the previously assigned value; otherwise it returns and sets a fresh random value (of the appropriate length).
- To answer query x to Sim, the simulator computes what follows. The simulator sets $\lambda = |x|$ and then the simulator does the following.
- Call the simulator of $\mathsf{3HVZK}$ on input x to obtain (a, e, z).
- If T happens to be already defined on $(x||a)$, then abort else set $T(x||a)$ to be a random string in $\{0,1\}^{k(\lambda)}$.

Consider the following hybrid experiments. H_2 is identical to the real experiment except that the prover, as the NIZKP simulator, keeps the table T and returns failure and aborts when queried on an already defined input x.

The crucial observation is that, as $\mathsf{3HVZK}$ satisfies high min-entropy of commitment (cf. Definition 1), the probability of failure in each of the queries to the prover is upper-bounded by $Q(\lambda) \cdot 2^{-\epsilon(\lambda)}$, where $Q(\cdot)$ is the total number of queries to $\mathcal{RO}$ at any stage and $\epsilon(\lambda)$ is the min-entropy of commitment, and thus, by assumption on $\epsilon(\cdot)$, is negligible in λ. As the number of steps of the adversary is bounded by a polynomial in λ, the number of queries are also bounded by a polynomial and thus the overall probability of failure is bounded by a negligible function in λ. Therefore, in H_2 the event of failure occurs with negligible probability and thus H_2 is statistically indistinguishable from the real experiment.

Assuming that the simulated transcript (a, e, z) for $x \in L$ is computationally indistinguishable (to non-uniform PPT adversaries) from real proofs for $(x, w) \in \mathcal{R}$, by a standard hybrid argument, it can be seen that the distribution of the query answers in the simulated experiment is computationally indistinguishable from the ones in H_2.

Therefore, no non-uniform PPT adversary has non-negligible advantage in distinguishing the two experiments of (adaptive multi-theorem) computational zero-knowledge.

Remark 2 The same comment or Remark 1 about composing the RO applies to the previous simulation but for simplicity we skipped such details.

Combining Lemmata 8 and 6 we have the following theorem.

Theorem 7 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $\delta(\cdot)$ be a negligible function and let $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{poly}_{\mathsf{out}}(\cdot), \mathsf{NIZK}[c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f])$ be the output of transform $\mathsf{Trans}_{\mathsf{main}}(c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f)$.

Then, NIZK satisfies (adaptive multi-theorem) computational zero-knowledge.

D Soundness

In this section we analyze the soundness of the NIZK systems obtained through our transform and in the Appendices C-G we study the other security properties.

Lemma 8 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{3HVZK}[c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f])$ be the output of transform $\mathsf{Trans}(c(\cdot), k(\cdot), q, m(\cdot), p(\cdot), f)$. Then, $\mathsf{3HVZK}$ is a 3-round public-coin proof system in the RO model with commitments of length $c(\lambda) \cdot p(\lambda)$, challenges of length $k(\lambda) \cdot p(\lambda)$, space of bad commitments of cardinality $\leq 2^{\lambda + \log(p(\lambda))}$ and soundness error $s(\lambda) = \left(\frac{1}{q} + (1 - \frac{1}{q}) \cdot \frac{1}{k(\lambda)}\right)^{p(\lambda)}$, up to a negligible factor.

Proof. It is easy to check that $\mathsf{3HVZK}$ is a 3-round and public-coin protocol in the RO model and that satisfies perfect completeness and the claim about the length of the commitments and challenges.

Since, for each $i \in [p(\lambda)]$, the verifier of $\mathsf{3HVZK}$ rejects if $a_i \neq \mathcal{RO}(x, i)$, it is easy to verify the claim on the cardinality of the space of bad commitments.

Let $x \in \{0,1\}^\lambda, x \notin \mathcal{R}_f$ (cf. Def. 12). The verifier accepts a transcript $((a_1, \ldots, a_{p(\lambda)}), (e_1, \ldots, e_{p(\lambda)}), (z_1, \ldots, z_{p(\lambda)}))$ if and only if for each $i \in [p(\lambda)]$

one of the following two events $(i.1)$ or $(i.2)$ occurs. Event $(i.1)$ occurs if and only if the prover claims that a_i falls outside the range of f and the proof for such claim is accepted. Event $(i.2)$ occurs if and only if event $(i.1)$ does not happen and the transcript (a_i, e_i, z_i) is accepted by the verifier of SpecP.

For each $i \in [p(\lambda)]$, the probability that $(i.1)$ occurs is, by the hypothesis of quasi-compactness of f (cf. Def. 11), $\frac{1}{q}$ and, by the fact that SpecP is a Σ-protocol (cf. Thm. 4) and the fact that the soundness error of a Σ-protocol is $\frac{1}{k(\lambda)}$, the probability that $(i.2)$ occurs is, $(1 - \frac{1}{q}) \cdot \frac{1}{k(\lambda)}$.

Therefore, for each $i \in [p(\lambda)]$, the probability that $(i.1)$ or $(i.2)$ occurs equals $\frac{1}{q} + (1 - \frac{1}{q}) \cdot \frac{1}{k(\lambda)}$ up to a negligible factor (cf. the quasi-compactness property of Def. 11).

As for all $i \in [p(\lambda)]$ the events "$(i.1)$ or $(i.2)$" are independent, the probability that a verifier of $\mathsf{3HVZK}$ accepts a transcript equals $(\frac{1}{q} + (1 - \frac{1}{q}) \cdot \frac{1}{k(\lambda)})^{p(\lambda)}$ up to a negligible factor.

Lemma 9 Let $\mathsf{3HVZK} = (\mathcal{P}, \mathcal{V})$ be a three-round public-coin proof system in the RO model for polynomial-time relation $\mathcal{R}$ having commitments of length $b(\lambda)$, space of bad commitments of cardinality $\leq 2^{b(\lambda)}$, challenges of length $k(\lambda)$ and soundness error $s(\lambda)$.

Let $\mathsf{NIZK} = (\mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ be the result of the FS transform on $\mathsf{3HVZK}$ and let $\{0,1\}^{\mathsf{poly_{inp}}(\lambda)}$ (resp. $\{0,1\}^{\mathsf{poly_{out}}(\lambda)}$) be the domain (resp. co-domain) of the RO in this transform. The polynomial $\mathsf{poly_{inp}}(\cdot)$ (resp. $\mathsf{poly_{out}}(\cdot)$ may be arbitrary until the domain (resp. co-domain) is sufficiently large to contain all strings of length $c(\lambda)$ (resp. $k(\lambda)$).[9]

Then, NIZK satisfies perfect completeness and statistical soundness with soundness error $\leq 2^{\lambda + b(\lambda)} \cdot s(\lambda)$.

Proof. Perfect completeness trivially holds (assuming implicitly that it holds for $\mathsf{3HVZK}$).

[9] Recall that $\mathsf{3HVZK}$ is a proof system in the RO model and thus it might need to get input/output from the RO on strings longer than the ones needed for the FS transform. For this reason we explicitly state that the RO may have different domain and co-domain until they are sufficiently large to be used in the FS transform.

Let us analyze the soundness error. Let $\mathsf{ROSp}(\lambda)$ be the space of all functions with domain $\{0,1\}^{\mathsf{poly}_{\mathsf{inp}}(\lambda)}$ and co-domain $\{0,1\}^{\mathsf{poly}_{\mathsf{out}}(\lambda)}$.

$$
\begin{aligned}
&\Pr_{\mathcal{RO}\leftarrow\mathsf{ROSp}(\lambda)}[\,\exists\ (x,\pi)\ x\notin L\ \wedge\ x\in\{0,1\}^{\lambda}\ \wedge\ \mathsf{NIZK.Verify}(x,\pi)=1\,] = \\
&\qquad\text{(where } e \text{ is computed as } \mathcal{RO}(a)) \\
&\Pr_{\mathcal{RO}\leftarrow\mathsf{ROSp}(\lambda)}[\,\exists\ (x,\pi=(a,z))\ x\notin L\wedge x\in\{0,1\}^{\lambda}\ \wedge\ \mathcal{V}(a,e,z)=1\,] \le \\
&\qquad\text{(by the union bound and the fact that the probability is 0 when } x\notin\{0,1\}^{\lambda}, x\notin L) \\
&\sum_{x\in\{0,1\}^{\lambda},x\notin L}\Pr_{\mathcal{RO}\leftarrow\mathsf{ROSp}(\lambda)}[\,\exists\ \pi=(a,z)\ \wedge\ \mathcal{V}(a,e,z)=1\,] = \\
&\qquad\text{(by the union bound)} \\
&\sum_{x\in\{0,1\}^{\lambda},x\notin L}\ \sum_{a\in\{0,1\}^{c(\lambda)}}\Pr_{\mathcal{RO}\leftarrow\mathsf{ROSp}(\lambda)}[\,\exists\ z\ \mathcal{V}(a,e,z)=1\,] = \\
&\qquad\text{(by definition of the set } S \text{ of bad commitments of } \mathsf{3HVZK}) \\
&\sum_{x\in\{0,1\}^{\lambda},x\notin L}\ \sum_{a\in S}\Pr_{\mathcal{RO}\leftarrow\mathsf{ROSp}(\lambda)}[\,\exists\ z\ \mathcal{V}(a,e,z)=1\,] = \\
&\qquad\text{(since, by assumption on } \mathsf{3HVZK}\text{, for all } a\in S\text{, the probability} \\
&\qquad\text{that a random } e\in\{0,1\}^{k(\lambda)} \text{ is s.t. } \exists\ z \text{ s.t. } (a,e,z) \text{ is accepting for } x\text{, is } \le s(\lambda)) \\
&\sum_{x\in\{0,1\}^{\lambda},x\notin L}\ \sum_{a\in S}s(\lambda) \le \\
&\qquad\text{(since } S \text{ has cardinality } 2^{b(\lambda)}) \\
&2^{\lambda+b(\lambda)}\cdot s(\lambda)
\end{aligned} \tag{2}
$$

as it was to prove.

Remark 3 For simplicity, we state the following theorem, as well as any other theorem in the rest of the work, stating that it works for any negligible function δ. As the proof shows, the theorem does not hold for all negligible functions, but does apply to all " nice" functions like $2^{-c\lambda}$ for some constant $c > 0$.

Theorem 10 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $\delta(\cdot)$ be a negligible function and let $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{poly}_{\mathsf{out}}(\cdot), \mathsf{NIZK}[c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f])$ be the output of transform $\mathsf{Trans}_{\mathsf{main}}(c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f)$. Then, NIZK satisfies completeness and has soundness error bounded by $\delta(\lambda)$, up to a fixed negligible factor.[10] Moreover, $\mathsf{Trans}_{\mathsf{main}}$ is a PPT algorithm.

Proof. Transform $\mathsf{Trans}_{\mathsf{main}}$ consists of the composition of Trans and the FS transform with polynomial $p(\lambda)$ set to satisfy the equation

[10] The negligible factor here is given by the quasi-compactness of Def 11.

$$2^{2\cdot\lambda+\log(p(\lambda))}\cdot\left(\frac{1}{q}+\left(1-\frac{1}{q}\right)\cdot\frac{1}{k(\lambda)}\right)^{p(\lambda)}\leq\delta(\lambda)$$

.

By Lemma 8, Transform Trans returns a proof system $\mathsf{3HVZK}$ with soundness error $s(\lambda)$ bounded by $(\frac{1}{q}+(1-\frac{1}{q})\cdot\frac{1}{k(\lambda)})^{p(\lambda)}$, up to a fixed negligible factor given by the quasi-compactness of f, and space of bad commitments of cardinality $2^{b(\lambda)}$ bounded by $2^{\lambda+\log(p(\lambda))}$. By Lemma 9, The FS transform applied to $\mathsf{3HVZK}$ returns a NIZKP NIZK with soundness error bounded by $2^{\lambda+b(\lambda)}\cdot s(\lambda)$. By the setting of $p(\cdot)$, NIZK has thus soundness error bounded by $\delta(\lambda)$.

To prove that $\mathsf{Trans}_{\mathsf{main}}$ is a PPT algorithm, it is necessary to show that it is possible to efficiently find a polynomial $p(\cdot)$ satisfying the previous equation. In fact, since $\frac{1}{k(\lambda)}\leq\frac{1}{2}$, q is a constant (cf. Def. 11), the soundness error of NIZK is bounded by $2^{2\cdot\lambda+\log(p(\lambda))}\cdot(\frac{1}{q'})^{p(\lambda)}$, up to a fixed negligible factor, for some constant $q'>1$ and thus it is easy to efficiently find a polynomial $p(\lambda)$ satisfying the equation (as stated in the remark this might not hold for all functions $\delta(\cdot)$ but it does hold, for instance, when $\delta(\lambda)=2^{-\lambda}$). It is also easy to see that NIZK satisfies completeness.

E Extractability

In this section, we define the notion of NIZK proof of knowledge system (NIZKPoK), and prove that our transform converts a public-coin HVZK with special soundness into a NIZKPoK system.

First, we recall a notion for interactive proof systems that will not be used directly in our results but which the corresponding notion for non-interactive systems will be inspired from.

Definition 16 [Proof of knowledge [GMR89,Dam10]] A pair $(\mathcal{P},\mathcal{V})$ of PPT interactive machines is called a *proof of knowledge with knowledge error* $k(\cdot)$ for polynomial-time relation $\mathcal{R}$ if completeness and the following property (that is a strengthening of soundness) hold.

- *Knowledge Soundness:* there exists a probabilistic oracle machine $\mathsf{Extract}$, called the *extractor*, such that for every interactive machine $\mathcal{P}^\star$ and for every input x accepted by $\mathcal{V}$ when interacting with $\mathcal{P}^\star$ with probability $\epsilon(|x|)>k(|x|)$, $\mathsf{Extract}^{\mathcal{P}^\star}(x)$ outputs a witness w for x. Moreover, the expected number of steps performed by $\mathsf{Extract}$ is bounded by $p(|x|)/(\epsilon(|x|)-k(|x|))^d$, for some polynomial $p(\cdot)$ and constant d.

Definition 17 [NIZKPoK] A non-interactive zero-knowledge proof of knowledge system (NIZKPoK, in short) $\mathsf{NIZK}=(\mathsf{poly}_{\mathsf{inp}}(\cdot),\mathsf{poly}_{\mathsf{out}}(\cdot),\mathsf{NIZK.Prove},\mathsf{NIZK.Verify})$ in the programmable RO model for a relation R is identical to a NIZKP except that it additionally satisfies the following computational extractability property.

- $\mathsf{Computational\ extractability}$. Computational extractability with error $\nu(\cdot)$ requires the existence of a PPT knowledge extractor $(\mathsf{Ext}, \mathsf{ExtRO})$. Ext and ExtRO are stateful and can communicate each other.

  For all non-uniform PPT adversaries Adv running in time bounded by $t(\cdot)$, Ext extracts w from a valid proof with overwhelming probability having the possibility of simulating a RO to the adversary through the algorithm ExtRO. The algorithm Ext has the possibility of rewinding the adversary on the same random tape. More formally, NIZK satisfies computational extractability with error $\nu(\cdot)$ if there exists a PPT extractor $(\mathsf{Ext}, \mathsf{ExtRO})$ such that for all non-uniform PPT adversaries Adv the following holds. Let:

$$\mathsf{acc}(\lambda) = \Pr\begin{bmatrix} r \leftarrow \{0,1\}^{t(\lambda)};\ (x,\pi) \leftarrow \mathsf{Adv}^{\mathsf{ExtRO}(\cdot)}(1^\lambda; r) : \\ \mathsf{Verify}^{\mathsf{ExtRO}(\cdot)}(x,\pi) = 1 \ \wedge\ (x,w) \in R \ \wedge\ |x| = \lambda \end{bmatrix}$$

$$\mathsf{ext}(\lambda) = \Pr\begin{bmatrix} r \leftarrow \{0,1\}^{t(\lambda)};\ (x,\pi) \leftarrow \mathsf{Ext}^{\mathsf{Adv}^{\mathsf{ExtRO}(\cdot)}(1^\lambda;\mathsf{r})};\ \mathsf{w} \leftarrow \mathsf{Ext}(1^\lambda, \mathsf{x}, \pi) : \\ \mathsf{Verify}^{\mathsf{ExtRO}(\cdot)}(x,\pi) = 1 \ \wedge\ (x,w) \in R \ \wedge\ |x| = \lambda \end{bmatrix}$$

  Then, there exists a constant $d \geq 0$ and a polynomial $p(\cdot)$ such that if $\mathsf{acc}(\lambda) \geq \nu(\lambda)$, we have that $\mathsf{ext}(\lambda) \geq \frac{1}{p(\lambda)} \cdot (\mathsf{acc}(\lambda) - \nu(\lambda))^d$.

For the proof of extractability of the NIZK systems resulting from our transform, we make use of the following version of the forking lemma, which appeared in Bellare and Neven [BN06] and generalizes the forking lemma of [PS00].

Lemma 11 [General forking lemma] Fix a polynomial $Q(\cdot)$ and a family of sets $\{\mathcal{H}\}_{\lambda \geq 0}$ of size $h(\lambda) \overset{\triangle}{=} |\mathcal{H}_\lambda| \geq Q(\lambda)$. Let P be a non-uniform PPT algorithm that on input $y, h_1, \ldots, h_{Q(|y|)}$ returns a pair, the first element of which is an integer in $[Q(|y|)]$ and the second element of which we refer to as a side output and runs in at most $t(\lambda)$ steps, for some polynomial $t(\cdot)$, for all $\lambda \geq 0$ and all inputs $y \in \{0,1\}^\lambda$. Let $\mathsf{IG}(1^\lambda)$ be a randomized algorithm that we call the input generator that on input a security parameter 1^λ outputs a string $y \in \{0,1\}^\lambda$.

The accepting probability of P, denoted by $\mathsf{acc}(\lambda)$, is defined as the probability that $J \geq 1$ in the experiment $y \leftarrow \mathsf{IG}(1^\lambda);\ h_1, \ldots, h_{Q(\lambda)} \leftarrow \mathcal{H}(\lambda);\ (J, s) \leftarrow P(y, h_1, \ldots, h_{Q(\lambda)})$. The forking algorithm Fork_P associated to P is the probabilistic algorithm that on input y proceeds as follows.

```
$\mathsf{Fork}_P(y)$
$\rho \leftarrow \{0,1\}^{t(\lambda)}$;
$h_1, \ldots, h_{Q(|y|)} \leftarrow \mathcal{H}(|y|)$;
$(I, s) \leftarrow P(y, h_1, \ldots, h_{Q(|y|)}; \rho)$;
If $I = 0$ return $(0, \perp, \perp)$;
$h'_I, \ldots, h'_{Q(|y|)} \leftarrow \mathcal{H}(|y|)$;
$(I', s') \leftarrow P(y, h_1, \ldots, h_{I-1}, h_{I'}, \ldots, h_{Q(|y|)'}; \rho)$;
If $I = I' \wedge (h_I \neq h_{I'})$ return $(1, s, s')$;
else return $(0, \perp, \perp)$;
```

(The efficiency of the above algorithm is polynomially related to the efficiency of P assuming an efficient way to sample elements from $\mathcal{H}$.)

Let $\mathsf{ext}(\lambda) = \Pr[b = 1 : \; y \leftarrow \mathsf{IG}(\lambda); \; (b, s, s') \leftarrow \mathsf{Fork}_P(y)]$, then $\mathsf{ext}(\lambda) \geq \mathsf{acc}(\lambda) \cdot (\frac{\mathsf{acc}(\lambda)}{Q(\lambda)} - \frac{1}{h(\lambda)})$.

We next prove the following theorem.

Theorem 12 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $\delta(\cdot)$ be any negligible function and let $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{poly}_{\mathsf{out}}(\cdot), \mathsf{NIZK}[c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f])$ be the output of transform $\mathsf{Trans}_{\mathsf{main}}(c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f)$.

Then, there exists some negligible function $\nu(\cdot)$ such that NIZK satisfies computational extractability with error $\nu(\cdot)$.

Proof. The proof follows the lines of the one of [FKMV12] except that ours does not address simulation extractability and as such is simplified and does not need the assumption of unique responses.

Let Adv be a non-uniform PPT adversary against the verifiability and let $t(\cdot)$ be a function such that for all $\lambda \geq 0$, the algorithm runs in at most $t(\lambda)$ steps on all inputs of length λ. We invoke the general forking lemma of Lemma 11.

In order to do so, we define program $P(1^\lambda, h_1, \ldots, h_{t(\lambda)}; \rho)$ as follows. P runs internally $\mathsf{Adv}(1^\lambda; \rho^{t(\lambda)})^{\mathsf{ExtRO}(\cdot)}$ on a fresh random string $\rho \leftarrow \{0,1\}^{t(\lambda)}$. Note that $t(\cdot)$ is also an upper-bound on the number of RO queries of Adv.

P uses values $(h_1, \ldots, h_t(\lambda))$ to simulate fresh answers of ExtRO. If $\mathsf{Adv}(1^\lambda; \rho)^{\mathsf{ExtRO}(\cdot)}$ outputs $(x^\star, (a^\star, z^\star))$, P checks if it is a valid proof and outputs $(J, x^\star, a^\star, z^\star)$, where $J > 0$ is the index corresponding to the random oracle query $(x^\star, a^\star)$. If the proof is not valid, P rejects outputting $(0, \bot)$. We say that P is successful whenever $J \geq 1$, and we denote with $\mathsf{acc}_f(\lambda)$ the corresponding probability (in the following we distinguish the probabilities $\mathsf{acc}_f, \mathsf{ext}_f$ of the forking lemma from the probabilities $\mathsf{acc}, \mathsf{ext}$ of the computational extractability for NIZK and Adv). Given program P, we consider two related runs of P with same random string ρ and different hash values, as specified by the forking algorithm Fork_P of Lemma 11.

Denote by $(I, (x^\star, a^\star, z^\star)) \leftarrow P(y, h_1, \ldots, h_{t(|y|)})$ and $(I', (x^{\star\star}, a^{\star\star}, z^{\star\star})) \leftarrow P(y, h_1, \ldots, h_{I-1}, h_{I'}, \ldots, h_{t(|y|)})$.

By the forking lemma we know that with probability $\mathsf{ext}_f(\lambda) \geq \mathsf{acc}_f(\lambda) \cdot (\mathsf{acc}_f(\lambda)/t(\lambda) - 1/2^{\mathsf{poly}_{\mathsf{out}}(\lambda)})$ the forking algorithm will return indexes I, I' such that $I = I', I \geq 1$ and $h_I \neq h_{I'}$. Since $I = I'$, we must have $x^\star = x^{\star\star}, a^\star = a^{\star\star}, z^\star = z^{\star\star}$. Moreover, we have that $h_I \neq H_{I'}$.

Let $\nu(\lambda) = \frac{t(\lambda)}{2^{\mathsf{poly}_{\mathsf{out}}(\lambda)}}$. Assume now that $\mathsf{acc}_f(\lambda) \geq \nu(\lambda)$. Recall that $\mathsf{ext}_f(\lambda) \geq (\mathsf{acc}_f(\lambda)^2/t(\lambda) - \mathsf{acc}_f(\lambda)/2^{\mathsf{poly}_{\mathsf{out}}(\lambda)})$. Since $t(\cdot)$ is polynomial while $2^{\mathsf{poly}_{\mathsf{out}}(\lambda)}$ is exponentially large in the security parameter, $\nu(\cdot)$ is a negligible function. Thus, $\mathsf{acc}_f(\lambda)^2/t(\lambda) - \mathsf{acc}_f(\lambda)/2^{\mathsf{poly}_{\mathsf{out}}(\lambda)} = \frac{1}{t(\lambda)} \cdot (\mathsf{acc}_f(\lambda)^2 - \mathsf{acc}_f(\lambda) \cdot \nu(\lambda))$ that, by the fact that $\mathsf{acc}_f(\lambda) \geq \nu(\lambda)$ and by algebraic manipulation, $\geq \frac{1}{t(\lambda)} \cdot (\mathsf{acc}_f(\lambda) - \nu(\lambda))^2$.

Therefore $\mathsf{ext}_f(\lambda) \geq \frac{1}{t(\lambda)} \cdot (\mathsf{acc}_f(\lambda) - \nu(\lambda))^2$.

Given that, the extractor Ext works as follows. The extractor invokes first the forking algorithm Fork_P to get the above values and then the extractor A guaranteed by the special soundness computing a witness $w^\star = \mathsf{Ext}(a^\star, h_I, z^\star, a^{\star\star}, h_{I'}, z^{\star\star})$ such that $(x^\star, w^\star) \in \mathcal{R}$.

Let $p(\cdot)$ be the polynomial used in the transform $\mathsf{Trans}_{\mathsf{main}}$ (recall that $\mathsf{Trans}_{\mathsf{main}}$ computes $p(\cdot)$ based on $\delta(\cdot)$ and $p(\lambda)$ represents the number of parallel iterations of the proof system to which the FS transform is applied).

Note that the extractor Ext might not work if the commitment $a^\star = (a^\star_1, \ldots, a^\star_{p(\lambda)})$ is such that for each $i \in [p(\lambda)]$, $a_i \notin \mathsf{Range}(f)$. In fact, in such case, the special soundness cannot be invoked. However, by the checks $a_i = \mathcal{RO}(x||i)$'s, by the fact that the output of the RO is uniformly distributed in the commitment space, and by hypothesis on q (cf. Def. 11), the probability that for each $i \in [p(\lambda)]$, $a_i = \mathcal{RO}(x||i)$ and $a_i \notin \mathsf{Range}(f)$ is $(\frac{1}{q}^{p(\lambda)}) \leq 1 - \frac{1}{p'(\lambda)}$, for some polynomial $p'(\cdot)$. Therefore, the probability ext of successful extraction is $\geq \frac{1}{p'(\lambda)} \cdot \mathsf{ext}_f(\lambda) \geq \frac{1}{p'(\lambda) \cdot t(\lambda)} \cdot (\mathsf{acc}_f(\lambda) - \nu(\lambda))^2$, as it was to prove.

F Separating FS from Our Transform

The next conjecture strictly separates FS transform (that when applied to some protocols only provides soundness breakable from adversaries running in $\Omega(2^\lambda)$ steps) from ours.

Conjecture 1 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $\delta(\cdot)$ be a negligible function. There exists a function $f(\lambda) \in o(2^\lambda)$, an integer $m > 0$ and an hash function $H(\cdot)$ with domain $\{0,1\}^\star$ and co-domain $\{0,1\}^m$ such that the following holds.

Let $(\mathsf{poly}_{\mathsf{inp}}(\cdot), \mathsf{poly}_{\mathsf{out}}(\cdot), \mathsf{NIZK}$ be the output of transform $\mathsf{Trans}^{H(\cdot), m(\mathrm{cot})}_{\mathsf{main}}(c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f)$ of construction 3.

Then NIZK' is a NIZKPoK for $\mathcal{R}_f$ (cf. Def. 12) satisfying $f(\lambda)$-soundness.

G WI

As noted and proved by Yung and Zhao [YZ06], and Ciampi *et al.* [CPSV16], if the original three-round public-coin HVZK proof system is witness indistinguishable, then the FS-transformed protocol is still witness indistinguishable, and the proof of witness indistinguishability is RO-free. Same considerations hold for our transform.

We first define the notion of non-interactive witness indistinguishable system.

Definition 18 [NIWI] A non-interactive witness indistinguishable proof system (NIWI, in short) $\mathsf{NIWI} \stackrel{\triangle}{=} (\mathsf{poly_{inp}}, \mathsf{poly_{out}}, \mathsf{NIWI.Prove}, \mathsf{NIWI.Verify})$ in the (non-programmable) RO model for a relation R is identical to a NIZKPoK except that the (adaptive multi-theorem) zero-knowledge property is replaced by the following witness indistinguishability property.

– *Witness indistinguishability (WI, in short):* Let L be the language associated with $\mathcal{R}$.
  For every function $\mathcal{RO} : \{0,1\}^{\mathsf{poly_{inp}}(\lambda)} \rightarrow \{0,1\}^{\mathsf{poly_{out}}(\lambda)}$, every two sequences $\{(x_\lambda, w^1_\lambda)\}_{\lambda>0}$, $\{(x_\lambda, w^2_\lambda)\}_{\lambda>0}$ such that for every $\lambda, x_\lambda \in L_\lambda, (x_\lambda, w^1_\lambda) \in \mathcal{R}$ and $(x_\lambda, w^2_\lambda) \in \mathcal{R}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, no non-uniform PPT distinguisher algorithm can distinguish the following two sequences of random variables with advantage more than $1/p(\lambda)$:
  - $\{\mathsf{NIWI.Prove}^{\mathcal{RO}(\cdot)}(x_\lambda, w^1_\lambda; U_m)\}_{\lambda \geq n}$.
  - $\{\mathsf{NIWI.Prove}^{\mathcal{RO}(\cdot)}(x_\lambda, w^2_\lambda; U_m)\}_{\lambda \geq n}$.
  (Where m is the number of random coins $\mathsf{NIWI.Prove}$ uses).

Corollary 13 Let $\mathsf{SpecP} = (\mathsf{SpecP.Prove}, \mathsf{SpecP.Verify})$ be a $\mathsf{spec\text{-}prot}$ with challenges of length $k(\cdot)$ and commitments of length $c(\cdot)$ for a $(m(\cdot), q)$-SOWGHF f. Note that according to our formulation, SpecP is induced by f, $k(\cdot)$, $m(\cdot)$ and q.

Let $\delta(\cdot)$ be a negligible function and let $(\mathsf{poly_{inp}}(\cdot), \mathsf{poly_{out}}(\cdot), \mathsf{NIZK}[c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f])$ be the output of transform $\mathsf{Trans_{main}}(c(\cdot), k(\cdot), q, m(\cdot), \delta(\cdot), f)$.

Then, NIZK is a NIWI proof system in the (*non-programmable*) RO model for $\mathcal{R}_f$ (cf. Def. 12).

Proof. The theorem follows from the proof of Theorem 10 and adapting the results of Yung and Zhao [YZ06] from FS to our transform. We omit further details.