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

Vincenzo Iovino[1(✉)] and Ivan Visconti[2]

[1] University of Luxembourg, Luxembourg City, Luxembourg
vinciovino@gmail.com
[2] DIEM, University of Salerno, Fisciano, Italy
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

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-16458-4_9) contains supplementary material, which is available to authorized users.

C. Carlet et al. (Eds.): C2SI 2019, LNCS 11445, pp. 118–141, 2019.
https://doi.org/10.1007/978-3-030-16458-4_9

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 cryptographic 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

[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".

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.

1.1 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 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 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 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 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 NIZK.Prove* that aims to compute an accepting proof for x. NIZK.Prove* searches over all pairs of challenges and commitments (a_c, c) such that the above property holds

[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.

(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 $p(\lambda) = 1$.

[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).

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 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 Definition 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 Sect. 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 Sect. 2 we say that the protocol is *special*. We now show how to transform a special protocol (spec-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 spec-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 spec-prot for f. We recall that in a spec-prot (see. Definition 13) the prover 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 spec-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 SpecP.Prove to compute a commitment. Thus, using r_i 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 spec-prot SpecP to obtain a protocol 3HVZK and then apply FS transform to 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 Sect. 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 Sect. 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 Sect. 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.

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. Eq. (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 $c\stackrel{\triangle}{=}3-\log_2(7)\approx 0.2$.

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.

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

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.

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.

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

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

[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.

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.

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 Sect. 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 spec-prot SpecP 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, 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 $|x| \stackrel{\triangle}{=} \lambda$ 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 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 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 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 spec-prot computes its first message as $f(r)$ where r is the chosen randomness, thus the first message corresponds to $f(f^{-1}(a_i)) = a_i$.

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

– [Round 1] $\mathsf{3HVZK.Prove}(y, (x, \mathsf{trap})) \rightarrow \mathsf{3HVZK.Verify}(y)$.
  For each $i \in [p(\lambda)]$, **do**
    $*$ **Send** $a_i \leftarrow \mathcal{RO}(y||i)$ to 3HVZK.Verify.
  • **endFor**

– [Round 2] $\mathsf{3HVZK.Verify}(y) \rightarrow \mathsf{3HVZK.Prove}(y, (x, \mathsf{trap}))$.
  For each $i \in [p(\lambda)]$, **do**
  * $e_i \leftarrow \{0,1\}^{k(\lambda)}$
  * **Send** e_i to $\mathsf{3HVZK.Prove}$.
  • **endFor**
– [Round 3] $\mathsf{3HVZK.Prove}(y, (x, \mathsf{trap})) \rightarrow \mathsf{3HVZK.Verify}(y)$.
  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] $\mathsf{3HVZK.Verify}(y) \rightarrow \{0,1\}$.
  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 $\mathsf{3HVZK}$ (resp. maximum between the length of the commitments and the length of the challenges of $\mathsf{3HVZK}$).

(In the following we assume that, e.g., if $\mathsf{3HVZK}$ 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 $\mathsf{3HVZK}$ 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 $\mathsf{3HVZK}$ 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 $\mathsf{3HVZK}$ 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 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.

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

Construction 3 [H-instantiation of our transform]. 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.

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 codomain $\{0,1\}^m$ 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 $H(\cdot)$.

References

[AABN02] Abdalla, M., An, J.H., Bellare, M., Namprempre, C.: From identification to signatures via the Fiat-Shamir transform: minimizing assumptions for security and forward-security. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 418–433. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_28

[AABN08] Abdalla, M., An, J.H., Bellare, M., Namprempre, C.: From identification to signatures via the Fiat-Shamir transform: necessary and sufficient conditions for security and forward-security. IEEE Trans. Inf. Theory **54**(8), 3631–3646 (2008)

[ABB+10] Almeida, J.B., Bangerter, E., Barbosa, M., Krenn, S., Sadeghi, A.-R., Schneider, T.: A certifying compiler for zero-knowledge proofs of knowledge based on Σ-protocols. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 151–167. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15497-3_10

[AF07] Abe, M., Fehr, S.: Perfect NIZK with adaptive soundness. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 118–136. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_7

[AMS+09] Armknecht, F., Maes, R., Sadeghi, A.-R., Sunar, B., Tuyls, P.: Memory leakage-resilient encryption based on physically unclonable functions. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 685–702. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_40

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

[BCNP04] Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: 45th Annual Symposium on Foundations of Computer Science, pp. 186–195. IEEE Computer Society Press, October 2004

[BDSG+13] Bitansky, N., et al.: Why "Fiat-Shamir for proofs" lacks a proof. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 182–201. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_11

[BFM88] Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th Annual ACM Symposium on Theory of Computing, pp. 103–112. ACM Press, May 1988

[BFS16] Bellare, M., Fuchsbauer, G., Scafuro, A.: NIZKs with an untrusted CRS: security in the face of parameter subversion. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 777–804. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_26

[BFSK11] Brzuska, C., Fischlin, M., Schröder, H., Katzenbeisser, S.: Physically uncloneable functions in the universal composition framework. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 51–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_4

[BFW15] Bernhard, D., Fischlin, M., Warinschi, B.: Adaptive proofs of knowledge in the random oracle model. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 629–649. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_28

[BG93] Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 390–420. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_28

[BLV03] Barak, B., Lindell, Y., Vadhan, S.P.: Lower bounds for non-black-box zero knowledge. In: 44th Annual Symposium on Foundations of Computer Science, pp. 384–393. IEEE Computer Society Press, October 2003

[BM88] Babai, L., Moran, S.: Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. J. Comput. Syst. Sci. **36**(2), 254–276 (1988)

[BN06] Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) 13th ACM Conference on Computer and Communications Security, CCS 2006 pp. 390–399. ACM Press, October/November 2006

[BPW12] Bernhard, D., Pereira, O., Warinschi, B.: How not to prove yourself: pitfalls of the Fiat-Shamir heuristic and applications to helios. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 626–643. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_38

[BR93] Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Ashby, V. (ed.) 1st ACM Conference on Computer and Communications Security, CCS 1993, pp. 62–73. ACM Press, November 1993

[BR08] Bellare, M., Ristov, T.: Hash functions from sigma protocols and improvements to VSH. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 125–142. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_9

[BY96] Bellare, M., Yung, M.: Certifying permutations: noninteractive zero-knowledge based on any trapdoor permutation. J. Cryptol. **9**(3), 149–166 (1996)

[CDS94] Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_19

[CG15] Chaidos, P., Groth, J.: Making sigma-protocols non-interactive without random oracles. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 650–670. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_29

[CGH98] Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th Annual ACM Symposium on Theory of Computing, pp. 209–218. ACM Press, May 1998

[CP93] Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_7

[CPS+16] Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Online/offline OR composition of sigma protocols. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 63–92. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_3

[CPSV16] Ciampi, M., Persiano, G., Siniscalchi, L., Visconti, I.: A transform for NIZK almost as efficient and general as the Fiat-Shamir transform without programmable random oracles. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 83–111. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_4

[CS98] Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055717

[CS03] Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. **33**(1), 167–226 (2003)

[Dam10] Damgård, I.: On Σ-protocol (2010). http://www.cs.au.dk/~ivan/Sigma.pdf

[DFN06] Damgård, I., Fazio, N., Nicolosi, A.: Non-interactive zero-knowledge from homomorphic encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 41–59. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_3

[DG03] Damgård, I., Groth, J.: Non-interactive and reusable non-malleable commitment schemes. In: 35th Annual ACM Symposium on Theory of Computing, pp. 426–437. ACM Press, June 2003

[DN00] Dwork, C., Naor, M.: Zaps and their applications. In: 41st Annual Symposium on Foundations of Computer Science, pp. 283–293. IEEE Computer Society Press, November 2000

[DNRS99] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science, pp. 523–534. IEEE Computer Society Press, October 1999

[DORS08] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)

[DRV12] Dodis, Y., Ristenpart, T., Vadhan, S.: Randomness condensers for efficiently samplable, seed-dependent sources. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 618–635. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_35

[Fis05] Fischlin, M.: Communication-efficient non-interactive proofs of knowledge with online extractors. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 152–168. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_10

[FKI06] Furukawa, J., Kurosawa, K., Imai, H.: An efficient compiler from Σ-protocol to 2-move deniable zero-knowledge. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 46–57. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_5

[FKMV12] Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the Fiat-Shamir transform. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 60–79. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34931-7_5

[FLS90] Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In: 31st Annual Symposium on Foundations of Computer Science, pp. 308–317. IEEE Computer Society Press, October 1990

[FS87] Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12

[GCvD02] Gassend, B., Clarke, D.E., van Dijk, M., Devadas, S.: Silicon physical random functions. In: Atluri, V. (ed.) 9th ACM Conference on Computer and Communications Security, CCS 2002, pp. 148–160. ACM Press, November 2002

[GGH+13] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th Annual Symposium on Foundations of Computer Science, pp. 40–49. IEEE Computer Society Press, October 2013

[GIS+10] Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding cryptography on tamper-proof hardware tokens. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 308–326. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_19

[GK03] Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th Annual Symposium on Foundations of Computer Science, pp. 102–115. IEEE Computer Society Press, October 2003

[GKR08] Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-time programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_3

[GM84] Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)

[GMO16] Giacomelli, I., Madsen, J., Orlandi, C.: Zkboo: faster zero-knowledge for boolean circuits. In: 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, 10–12 August 2016, pp. 1069–1083 (2016)

[GMR89] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)

[GMY06] Garay, J.A., MacKenzie, P.D., Yang, K.: Strengthening zero-knowledge protocols using signatures. J. Cryptol. **19**(2), 169–209 (2006)

[Gol01] Goldreich, O.: Foundations of Cryptography: Basic Techniques, vol. 1. Cambridge University Press, Cambridge (2001)

[GOS06a] Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 97–111. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_6

[GOS06b] Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_21

[GOSV14] Goyal, V., Ostrovsky, R., Scafuro, A., Visconti, I.: Black-box non-black-box zero knowledge. In: Shmoys, D.B. (ed.) 46th Annual ACM Symposium on Theory of Computing, pp. 515–524. ACM Press, May/June 2014

[GS08] Groth, J., Sahai, A.: Efficient Non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_24

[HL08] Hazay, C., Lindell, Y.: Constructions of truly practical secure protocols using standardsmartcards. In: Ning, P., Syverson, P.F., Jha, S. (eds.) 15th ACM Conference on Computer and Communications Security, CCS 2008, pp. 491–500. ACM Press, October 2008

[Kat07] Katz, J.: Universally composable multi-party computation using tamper-proof hardware. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 115–128. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_7

[KRR16] Kalai, Y.T., Rothblum, G.N., Rothblum, R.D.: From obfuscation to the security of Fiat-Shamir for proofs. IACR Cryptology ePrint Archive 2016:303 (2016)

[Lin06] Lindell, Y.: A simpler construction of CCA2-secure public-key encryption under general assumptions. J. Cryptol. **19**(3), 359–377 (2006)

[Lin15] Lindell, Y.: An efficient transform from sigma protocols to NIZK with a CRS and Non-programmable random oracle. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 93–109. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_5

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

[MP03] Micciancio, D., Petrank, E.: Simulatable commitments and efficient concurrent zero-knowledge. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 140–159. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_9

[MV16] Mittelbach, A., Venturi, D.: Fiat–Shamir for highly sound protocols is instantiable. In: Zikas, V., De Prisco, R. (eds.) SCN 2016. LNCS, vol. 9841, pp. 198–215. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44618-9_11

[NY90] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd Annual ACM Symposium on Theory of Computing, pp. 427–437. ACM Press, May 1990

[OPV10] Ostrovsky, R., Pandey, O., Visconti, I.: Efficiency preserving transformations for concurrent non-malleable zero knowledge. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 535–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_32

[OSVW13] Ostrovsky, R., Scafuro, A., Visconti, I., Wadia, A.: Universally composable secure computation with (Malicious) physically uncloneable functions. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 702–718. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_41

[Pas03] Pass, R.: On deniability in the common reference string and random oracle model. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 316–337. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_19

[Pas13] Pass, R.: Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 334–354. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_19

[PRTG02] Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical one-way functions. Science **297**(5589), 2026–2030 (2002)

[PS00] Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**(3), 361–396 (2000)

[Ps05] Pass, R., Shelat, A.: Unconditional characterizations of non-interactive zero-knowledge. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 118–134. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_8

[PsV06] Pass, R., Shelat, A., Vaikuntanathan, V.: Construction of a non-malleable encryption scheme from any semantically secure one. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 271–289. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_16

[RS92] Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_35

[RSA78] Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signature and public-key cryptosystems. Commun. Assoc. Comput. Mach. **21**(2), 120–126 (1978)

[RvD13] Rührmair, U., van Dijk, M.: PUFs in security protocols: Attack models and security evaluations. In: 2013 IEEE Symposium on Security and Privacy, pp. 286–300. IEEE Computer Society Press, May 2013

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

[SG02] Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. J. Cryptol. **15**(2), 75–96 (2002)

[TSS+05] Tuyls, P., Škorić, B., Stallinga, S., Akkermans, A.H.M., Ophey, W.: Information-theoretic security analysis of physical uncloneable functions. In: Patrick, A.S., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 141–155. Springer, Heidelberg (2005). https://doi.org/10.1007/11507840_15

[VV09] Ventre, C., Visconti, I.: Co-sound zero-knowledge with public keys. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 287–304. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02384-2_18

[YZ06] Yung, M., Zhao, Y.: Interactive zero-knowledge with restricted random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 21–40. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_2

[YZ07] Yung, M., Zhao, Y.: Generic and practical resettable zero-knowledge in the bare public-key model. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 129–147. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_8