Fiat–Shamir for Highly Sound Protocols Is Instantiable

Arno Mittelbach[1] and Daniele Venturi[2(✉)]

[1] Cryptoplexity, Technische Universität Darmstadt, Darmstadt, Germany
[2] Department of Information Engineering and Computer Science, University of Trento, Trento, Italy
daniele.venturi@unitn.it

Abstract. The Fiat–Shamir (FS) transformation (Fiat and Shamir, Crypto '86) is a popular paradigm for constructing very efficient non-interactive zero-knowledge (NIZK) arguments and signature schemes using a hash function, starting from any three-move interactive protocol satisfying certain properties. Despite its wide-spread applicability both in theory and in practice, the known positive results for proving security of the FS paradigm are in the random oracle model, i.e., they assume that the hash function is modelled as an external random function accessible to all parties. On the other hand, a sequence of negative results shows that for certain classes of interactive protocols, the FS transform cannot be instantiated in the standard model.

We initiate the study of complementary positive results, namely, studying classes of interactive protocols where the FS transform *does* have standard-model instantiations. In particular, we show that for a class of "highly sound" protocols that we define, instantiating the FS transform via a q-wise independent hash function yields NIZK arguments and secure signature schemes. For NIZK, we obtain a weaker "q-bounded" zero-knowledge flavor where the simulator works for all adversaries asking an a-priori bounded number of queries q; for signatures, we obtain the weaker notion of random-message unforgeability against q-bounded random message attacks.

Our main idea is that when the protocol is highly sound, then instead of using random-oracle programming, one can use complexity leveraging. The question is whether such highly sound protocols exist and if so, which protocols lie in this class. We answer this question in the affirmative in the common reference string (CRS) model and under strong assumptions. Namely, assuming indistinguishability obfuscation and puncturable pseudorandom functions we construct a compiler that transforms any 3-move interactive protocol with instance-independent commitments and simulators (a property satisfied by the Lapidot-Shamir protocol, Crypto '90) into a compiled protocol in the CRS model that is highly sound. We also present a second compiler, in order to be able to start from a larger class of protocols, which only requires instance-independent commitments (a property for example satisfied by the classical protocol for quadratic residuosity due to Blum, Crypto '81). For the second compiler we require dual-mode commitments.

We hope that our work inspires more research on classes of (efficient) 3-move protocols where Fiat–Shamir is (efficiently) instantiable.

V. Zikas and R. De Prisco (Eds.): SCN 2016, LNCS 9841, pp. 198–215, 2016.
DOI: 10.1007/978-3-319-44618-9_11

1 Introduction

The Fiat–Shamir (FS) transformation [26] is a popular[1] technique to build efficient non-interactive zero-knowledge (NIZK) arguments and signature schemes, starting from three-round *public-coin* (3PC) protocols satisfying certain properties. In a 3PC protocol the prover starts by sending a commitment α, to which the verifier replies with a challenge β drawn at random from some space $\mathcal{B}$; finally the prover sends a reply γ and the verifier's verdict is computed as a predicate of the transcript (α, β, γ).

1.1 Fiat–Shamir NIZK and Signatures

We briefly review both the main applications of the FS transform below.

- **NIZK.** A NIZK is a non-interactive protocol in which the prover—holding a witness w for membership of a statement x in some *NP*-language L—can convince the verifier—holding just x—that $x \in L$, by sending a single message π. NIZK should satisfy three properties. First, *completeness* says that an honest prover holding a valid witness (almost) always convinces an honest verifier. Second, *soundness* says that a malicious prover should not be able to convince the honest verifier into accepting a *false* statement, i.e. a statement $x \notin L$; we speak of *arguments* (resp., *proofs*) when the soundness requirement holds for all computationally bounded (resp., computationally unbounded) provers. Third, *zero-knowledge* requires that a proof does not reveal anything about the witness beyond the validity of the statement being proven.

  Apart from being a fascinating topic, NIZK have been demonstrated to be extremely useful for cryptographic applications (see, e.g., [11,12,22,24,28,36]). NIZK require a setup assumption, typically in the form of a common reference string (CRS).

  Starting with a 3PC protocol, the FS transform makes it a NIZK by having the prover compute the verifier's challenge as a hash of the commitment α via some hash function H (with "hash key" hk); this results in a single message $\pi = (\alpha, \beta, \gamma)$, where $\beta = \mathsf{H}(\mathsf{hk}, \alpha)$, that is sent from the prover to the verifier.[2] (The description of the hash function, i.e. key hk, is included as part of the CRS.)
- **Signatures.** Digital signatures are among the most important and well-studied cryptographic tools. Signature schemes allow a signer (holding a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$) to generate a signature σ on a message m, in such a way that anyone possessing the public key pk can verify the validity of (m, σ). Signatures must be unforgeable, meaning that it should be hard to forge a signature on a "fresh" chosen message (even after seeing polynomially many signatures on possibly chosen messages).

[1] There are over 3.000 Google-Scholar-known citations to [26], as we type.

[2] The value β is typically omitted from the proof, as the verifier can compute it by itself.

Starting with a 3PC protocol, the FS transform makes it a signature by having the signer compute the verifier's challenge as a hash of the commitment α, concatenated with the message m, via some hash function H (with "hash key" hk); this results in a signature $\sigma = (\alpha, \beta, \gamma)$, where $\beta = \mathsf{H}(\mathsf{hk}, \alpha||m)$.

1.2 Positive and Negative Results

We refer to the non-interactive system obtained by applying the FS transform to a 3PC protocol (i.e., a NIZK or a signature scheme) as the *FS collapse*. A fundamental question in cryptography is to understand what properties the initial 3PC protocol and the hash function should satisfy in order for the FS collapse to be a NIZK argument or a secure signature scheme. This question has been studied extensively in the literature; we briefly review the current state of affairs below.

Positive Results. All security proofs for the FS transform follow the random oracle methodology (ROM) of Bellare and Rogaway [4], i.e., they assume that the function H behaves like an external random function accessible to all parties (including the adversary). In particular, a series of papers [1,26,40,42] establishes that the FS transform yields a secure signature scheme in the ROM provided that the starting 3PC is a passively secure identification scheme. The first definition of NIZK in the ROM dates back to [4] (where a particular protocol was analyzed); in general, it is well known that, always in the ROM, the FS transform yields a NIZK satisfying sophisticated properties such as simulation-soundness [25] and simulation-extractability [6].

Barak *et al.* [3] put forward a new hash function property (called entropy preservation[3]) that allows to prove soundness of the FS collapse without random oracles; their result requires that the starting 3PC protocol is statistically sound, i.e. it is a *proof*. Dodis *et al.* [21] show that such hash functions exist if a conjecture on the existence of certain "condensers for leaky sources" turns out to be true. Canetti *et al.* [13] study the correlation intractability of obfuscated pseudorandom functions and show a close connection between entropy preservation and correlation intractability, but it remains open whether their construction achieves entropy preservation or, in fact, whether entropy-preserving hash functions exist in the standard model. A negative indication to this question was recently presented by Bitansky *et al.* [7] who show that entropy-preservation security cannot be proven via a black-box reduction to a *cryptographic game*.

Negative Results. It is often difficult to interpret what a proof in the ROM means in the standard model. This is not only because concrete hash functions seem far from behaving like random oracles, but stems from the fact that there exist cryptographic schemes that can be proven secure in the ROM, but are *always* insecure in the standard model [14].

[3] Entropy preservation roughly says that for all efficient adversaries that get a uniformly random hash key hk and produce a correlated value α, the conditional Shannon entropy of $\beta = \mathsf{H}(\mathsf{hk}, \alpha)$ given α, but not hk, is sufficiently large.

The FS transformation is not an exception in this respect. In their study of "magic functions", Dwork *et al.* [23] establish that whenever the initial 3PC protocol satisfies the zero-knowledge property, its FS collapse can never be (computationally) sound for any implementation of the hash function. Goldwasser and Kalai [29], building on previous work of Barak [2], construct a specially-crafted 3PC *argument* for which the FS transform yields an insecure signature scheme for any standard model implementation of the hash function.

Bitansky *et al.* [8] and Dachman-Soled *et al.* [18] (see also [7]) show an unprovability result that also covers 3PC *proofs*. More in detail, [8] shows that the FS transform cannot always preserve soundness when starting with a 3PC proof, under a black-box reduction to any falsifiable assumption (even ones with an inefficient challenger). [18] shows a similar black-box separation (although only for assumptions with an efficient challenger) for any concrete proof that is honest-verifier zero-knowledge against sub-exponential size distinguishers. In a related paper, Goyal *et al.* [30] obtain a negative result for non-interactive information-theoretically secure witness indistinguishable arguments.

1.3 Our Contributions

The negative results show that, for certain classes of interactive protocols, the FS transform cannot be instantiated in the standard model. We initiate the study of complementary positive results, namely, studying classes of interactive protocols where the FS transform *does* have a standard-model instantiation. We show that for a class of "highly sound" protocols that we define, instantiating the FS transform via a q-wise independent hash function yields both a NIZK argument in the CRS model and a secure signature scheme. In the case of NIZK, we get a weaker "q-bounded" zero-knowledge flavor where the simulator works for all adversaries asking an a-priori bounded number of queries q; in the case of signatures, we get the weaker notion of random-message unforgeability against q-bounded random message attacks, where the forger can observe signatures on random messages and has to produce a forgery on a fresh random message.

Very roughly, highly sound protocols are a special class of 3PC arguments and identification schemes satisfying three additional properties: (**P1**) The honest prover computes the commitment α independently of the instance being proven and of the corresponding witness; (**P2**) The soundness error of the protocol is tiny, in particular the ratio between the soundness error and the worst-case probability of guessing a given commitment is bounded-away from one; (**P3**) Honest conversations between the prover and the verifier on common input x can be simulated knowing just x, and moreover the simulator can fake α independently of x itself.

We are not aware of natural protocols that are directly highly sound according to our definition. (But we will later discuss that, e.g., the Lapidot-Shamir protocol [37] partially satisfies our requirements.) Hence, the question is whether such highly sound protocols exist and, if so, which languages and protocols lie in this class. We answer this question in the affirmative in the CRS model and under strong assumptions. Namely, assuming indistinguishability obfuscation,

puncturable pseudorandom functions and equivocal commitments, we build a sequence of two compilers that transform any three-move interactive protocol with instance-independent commitments (i.e., property **P1**) into a compiled protocol in the CRS model that satisfies the required properties. Noteworthy, our compilers are language-independent, and we know that assuming one-way permutations three-move interactive protocols with instance-independent commitments exist for all of *NP*.

Our result avoids Dwork *et al.* [23], because we start from a protocol that is honest-verifier zero-knowledge rather than fully zero-knowledge. Note that our approach also circumvents the negative result of [8,30] as our technique applies only to a certain class of 3PC arguments. Furthermore, we circumvent the black-box impossibility result [18] by using complexity leveraging and sub-exponential security assumptions.

1.4 Perspective

The main contribution from our perspective is to initiate the study of restricted positive standard-model results for the FS transform. Namely, we show that for the class of highly sound protocols, the FS transform can be instantiated via a q-wise independent hash function (both for the case of NIZK and signatures). This is particularly interesting given the negative results in [7,23,29].

An important complementary question is, of course, to study the class of highly sound protocols. Under strong assumptions, our compilers show that highly sound protocols exist for all languages in *NP*. However, the compilers yield protocols in the CRS model and, at least for the case of NIZK, as we discuss now, one has to take care in interpreting positive results about the FS transform applied to 3PC protocols in the CRS model.

It is well known that in the CRS model one can obtain a NIZK both for *NP*-complete languages [10] and for specific languages [31]. Let L be a language. Given a standard 3PC protocol for proving membership of elements $x \in L$, and with transcripts (α, β, γ), consider the following dummy "compiler" for obtaining a 3PC protocol for L in the CRS model. The first message α^* and the second message β^* of the compiled protocol are equal to the empty string ε; the third message is a NIZK proof γ^* that $x \in L$. Note that the FS transform is easily seen to be secure (without random oracles) on such a dummy protocol, the reason for this being that α^* and β^* play no role at all in the obtained 3PC! Further note that this artificial "compiler" actually ignores the original protocol, and hence it does not rely on any of the security features of the underlying protocol. Regrettably, the above example does not shed any light on the security of the FS transform and when it applies.

In turn, our result for FS NIZK has two interesting features. First, our instantiation of the FS transform works even if the starting 3PC is in the standard model (provided that it satisfies **P1**-**P3**). Second, our CRS-based compiler is very different from the above dummy compiler in that we do not simply "throw away" the initial 3PC but instead rely on all of its properties in order to obtain a 3PC satisfying **P1**-**P3**.

We remark that the above limitation does not apply to our positive result for FS signatures, since assuming the initial 3PC protocol works in the CRS model does not directly yield a dummy "compiler" as the one discussed above.

1.5 Related Work

On Fiat–Shamir. It is worth mentioning that using indistinguishability obfuscation and puncturable PRFs one can directly obtain a NIZK for all *NP* as shown by Sahai and Waters [43]. However, our main focus is not on constructions of NIZK, rather we aim at providing a better understanding of what can be proved for the FS transform without relying on random oracles. In this respect, our result shares similarities to the standard-model instantiation of Full-Domain Hash given in [34].

In the case of NIZK, an alternative version of the FS transform is defined by having the prover hashing the statement x together with value α, in order to obtain the challenge β. The latter variant is sometimes called the *strong* FS transform (while the variant we analyze is known as the *weak* FS transform). Bernhard *et al.* [6] show that the weak FS transform might lead to problems in certain applications where the statement to be proven can be chosen adversarially (this is the case, e.g., in the Helios voting protocol). Unfortunately, it seems hard to use our proof techniques to prove zero-knowledge of the strong FS collapse, because the simulator for zero-knowledge does not know the x values in advance.

Our positive result for FS signatures shares some similarities with the work of Bellare and Shoup [5], showing that "actively secure" 3PC protocols yield a restricted type of secure signature schemes (so-called two-tier signatures) when instantiating the hash function in the FS transform via any collision-resistant hash function.

Compilers. Our approach of first compiling any "standard" 3PC protocol into one with additional properties that suffice for proving security of the FS transform is similar in spirit to the approach taken by Haitner [32] who shows how to transform any interactive argument into one for which parallel repetition decreases the soundness error at an exponential rate.

Lindell recently used a similar idea to first transform a 3PC into a new protocol in the CRS model, and then show that the resulting 3PC when transformed with (a slightly modified version of) Fiat–Shamir satisfies zero-knowledge in the standard model [38]. His approach was later improved in [17]. We note that the use of a CRS-enhanced interactive protocol is only implicit in Lindell's work as he directly analyzes the collapsed non-interactive version. On the downside, to prove soundness Lindell still requires (non-programmable) random oracles. We note that one of our compilers is essentially equivalent to the compiler used by Lindell. Before Lindell's work, interactive protocols in the CRS model have also been studied by Damgård who shows how to build 3-round concurrent zero-knowledge arguments for all *NP*-problems in the CRS model [20].

Alternative Transforms. Other FS-inspired transformations were considered in the literature. For instance Fischlin's transformation [27] (see also [19]) yields a simulation-sound NIZK argument with an online extractor; as mentioned above, Lindell [38] defines a twist of the FS transform that allows to prove zero-knowledge in the CRS model, and soundness in the non-programmable random oracle model. It is an interesting direction for future research to apply our techniques to analyze the above transformations without random oracles.

Concurrent Paper. Recently, in a concurrent and independent work, Kalai, Rothblum and Rothblum [35] showed a positive result for FS in the plain model, under complexity assumptions similar to ours. More in details, assuming sub-exponentially secure indistinguishability obfuscation, input-hiding obfuscation for the class of multi-bit point functions, and sub-exponentially secure one-way functions, [35] shows that, when starting with any 3PC *proof*, the FS transform yields a *two-round* computationally-sound interactive protocol.

On the positive side, their result applies to any 3PC proof (while ours only covers a very special class of 3PC arguments). On the negative side, their technique only yields a positive result for a two-round interactive variant of the FS transform (while our techniques apply to the full FS collapse, both for NIZK and for signatures).

1.6 Roadmap

Section 2 contains a detailed informal overview of our positive result for the case of FS NIZK; the corresponding formal definitions and proofs are deferred to the full version [39]. We present an overview of our compilers for obtaining highly sound protocols (in the CRS model) in Sect. 3; a more detailed treatment appears in the full paper [39], where we also explain how to adapt our techniques to the case of FS signatures.

2 FS NIZK

Fiat–Shamir Transform. The Fiat–Shamir (FS) transform [26] is a generic way to remove interaction from certain argument systems, using a hash function. For the rest of the paper, we consider only interactive arguments consisting of three messages—which we denote by (α, β, γ)—where the first message is sent by the prover. We also focus on so-called *public-coin* protocols where the verifier's message β is uniformly random over some space $\mathcal{B}$ (e.g., $\beta \in \{0,1\}^k$ for some $k \in \mathbb{N}$). We call this a 3PC argument system for short, and denote it by $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V})$; here K generates a CRS crs,[4] whereas P and V correspond to the prover and verifier algorithms.

[4] For standard-model 3PC arguments, the CRS contains the empty string ε. The reason for considering a CRS is that, looking ahead, our compilers yield highly sound protocols in the CRS model.

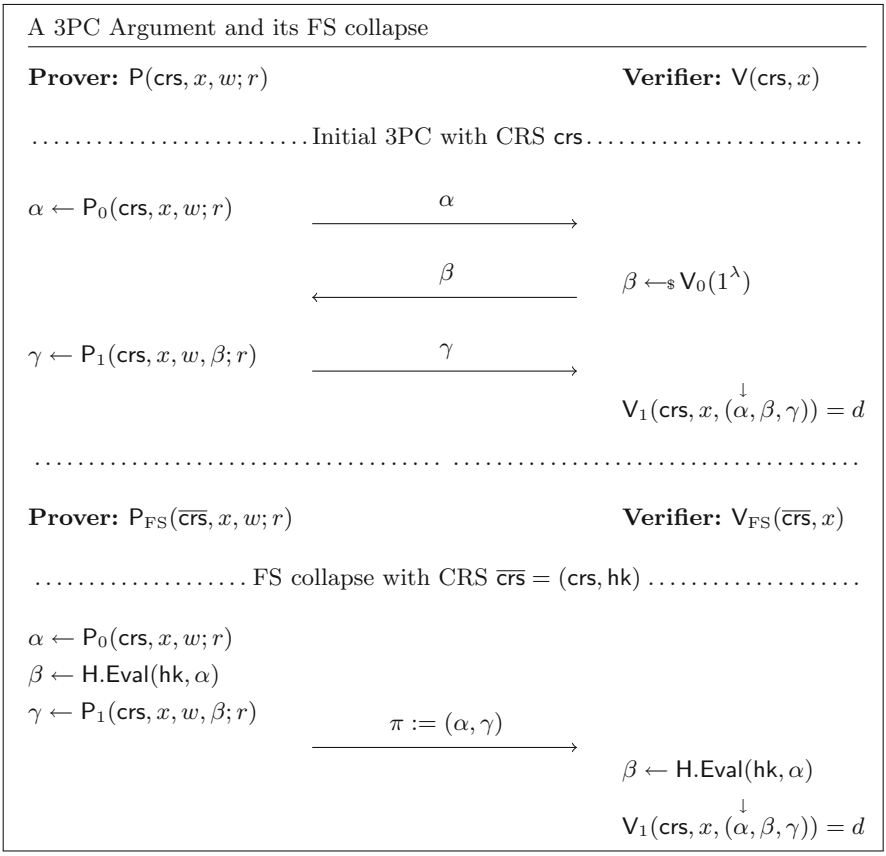

Fig. 1. Message flow of a typical 3PC argument system and its corresponding FS collapse.

For 3PC arguments we can think of the prover algorithm as being split into two sub-algorithms $\mathsf{P} := (\mathsf{P}_0, \mathsf{P}_1)$, where P_0 takes as input a pair (x, w) and outputs the prover's first message α (the so-called commitment) and P_1 takes as input (x, w) as well as the verifier's challenge β to produce the prover's second message γ (the so-called response). In general P_0 and P_1 are allowed to share the same random tape, which we denote by $r \in \{0,1\}^*$. In a similar fashion we can think of the verifier's algorithm as split into two sub-algorithms $\mathsf{V} = (\mathsf{V}_0, \mathsf{V}_1)$, where V_0 outputs a uniformly random value $\beta \in \mathcal{B}$ and V_1 is deterministic and corresponds to the verifier's verdict (i.e., V_1 takes as input x and a transcript (α, β, γ) and returns a decision bit $d \in \{0,1\}$).

The FS transform allows to remove interaction from any 3PC argument system for a polynomial-time computable relation R as specified below (see also Fig. 1). Let $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V})$ be the initial 3PC argument system. Additionally, consider a family of hash functions H consisting of algorithms $\mathsf{H.KGen}$, $\mathsf{H.kl}$, $\mathsf{H.Eval}$,

$\mathsf{H.il}$ and $\mathsf{H.ol}$; here $\mathsf{H.il}$ and $\mathsf{H.ol}$ correspond, respectively, to the bit lengths of messages α and β (as a function of the security parameter λ).

The FS collapse of Π using H is a triple of algorithms $\overline{\Pi}_{\mathrm{FS},\mathsf{H}} := (\mathsf{K}_{\mathrm{FS}}, \mathsf{P}_{\mathrm{FS}}, \mathsf{V}_{\mathrm{FS}})$:

- Algorithm K_{FS} takes as input the security parameter, samples $\mathsf{hk} \leftarrow\!\!\$\, \mathsf{H}.\mathsf{KGen}(1^\lambda)$, $\mathsf{crs} \leftarrow\!\!\$\, \mathsf{K}(1^\lambda)$, and publishes $\overline{\mathsf{crs}} := (\mathsf{crs}, \mathsf{hk})$.
- Algorithm P_{FS} takes as input $(\overline{\mathsf{crs}}, x, w)$ and runs $\mathsf{P}_0(\mathsf{crs}, x, w)$ in order to obtain the commitment $\alpha \in \{0,1\}^{\mathsf{H.il}(\lambda)}$; next P_{FS} defines the challenge as $\beta := \mathsf{H.Eval}(\mathsf{hk}, \alpha)$ and runs $\mathsf{P}_1(\mathsf{crs}, x, w, \beta)$ in order to obtain the response γ. Finally P_{FS} outputs $\pi := (\alpha, \gamma)$.
- Algorithm V_{FS} takes as input $(\overline{\mathsf{crs}}, x, \pi)$ and returns 1 if and only if verifier $\mathsf{V}_1(\mathsf{crs}, x, (\alpha, \beta, \gamma)) = 1$ where $\beta = \mathsf{H.Eval}(\mathsf{hk}, \alpha)$.

Briefly, the result of Fiat and Shamir says that if Π is a (standard-model) 3PC argument satisfying completeness, computational soundness, and computational honest-verifier zero-knowledge (in addition to a basic requirement on the min-entropy of the prover's commitment), its FS collapse $\overline{\Pi}_{\mathrm{FS},\mathsf{H}}$ is a NIZK argument system if H is modeled as a random oracle.

Our standard-model security proof proceeds in two modular steps. In the first step, we prove completeness and soundness of a "selective" variant of the FS transform; in the second step we analyze the standard FS transform using complexity leveraging. Details follow.

The Selective FS Transform. Consider a 3PC argument for a language L. For a hash family H, consider the following (interactive) selective adaptation of the FS transformation: The prover sends the commitment α as in the original protocol; the verifier, instead of sending the challenge $\beta \in \mathcal{B}$ directly, forwards a honestly generated hash key hk; finally the prover uses (hk, α) to compute $\beta = \mathsf{H}(\mathsf{hk}, \alpha)$ and then obtains the response γ as in the original 3PC argument.

In the full paper [39] we prove that if the starting 3PC protocol has instance-independent commitments, is complete and computationally sound, so is the one obtained by applying the selective FS transform. The idea is to use a "programmable" q-wise independent hash function (e.g., a random polynomial of degree $q-1$ over a finite field) to "program" the hash function up-front; note that commitment α is computed before the hash key is generated and hence, we can embed the challenge value β into the hash function such that it maps α to β and reduce to the soundness of the underlying 3PC argument.

Complexity Leveraging. The second step in proving soundness of the FS collapse (we discuss zero-knowledge below) consists in applying complexity leveraging so that we can swap the order of α and β. Hence, this step can only be applied to protocols satisfying an additional property as we discuss next.

Let Π be the initial 3PC argument, and denote by $\overline{\Pi}$ its corresponding FS collapse. Given a malicious prover P^* breaking soundness of $\overline{\Pi}$, we construct a prover P attacking soundness of the selective FS transform as follows. P picks a random α from the space of all possible commitments, and forwards α to

the verifier; after receiving the challenge hash key hk, prover P runs P^* which outputs a proof (α^*, γ^*). Prover P simply hopes that $\alpha^* = \alpha$, in which case it forwards γ^* to the verifier (otherwise it aborts). It follows that if the selective FS has soundness roughly $s(\lambda)$ (for security parameter λ), the soundness of $\overline{\Pi}$ is roughly $s(\lambda)$ divided by the probability of guessing correctly the value α^* in the first step of the reduction.

Note that for the above argument to give a meaningful bound, we need that the soundness of $\overline{\Pi}$ is bounded away from one. This leads to the following (non-standard) requirement that the initial 3PC argument should satisfy.

> **P2:** $\varrho(\lambda) := s(\lambda)/2^{-a(\lambda)} < 1$, where $s(\lambda)$ is the soundness error and $a(\lambda)$ is the maximum bit-length associated to the commitment α.

Zero-Knowledge. We assume that the initial 3PC is honest-verifier zero-knowledge (HVZK)—i.e., that it is zero-knowledge for honest verifiers. We need to show that $\overline{\Pi}$ satisfies zero-knowledge. Here, we require two additional properties as explained below; interactive protocols obeying the first property already appeared in the literature under the name of "input-delayed" protocols [15,16,33].

> **P1:** The value α output by the prover is computed independently of the instance x being proven (and of the corresponding witness w).

> **P3:** The value α output by the simulator is computed independently of the instance x being proven.

We now discuss the reduction for the zero-knowledge property and explain where **P1** and **P3** are used. We need to construct an efficient simulator that is able to simulate arguments for adaptively chosen (true) statements—without knowing a witness for such statements. The output of the simulator should result in a distribution that is computationally indistinguishable from the distribution generated by the real prover. The simulator gets extra power, as it can produce a "fake" CRS together with some trapdoor information tk (on which the simulator can rely) such that the "fake" CRS is indistinguishable from a real CRS.

In order to build some intuition, it is perhaps useful to recall the random-oracle-based proof for the zero-knowledge property of the FS transform. There, values α_i and β_i corresponding to the i-th adversarial query are computed by running the HVZK simulator and are later "matched" relying on the programmability of the random oracle. Roughly speaking, in our standard-model proof we take a similar approach, but we cannot use *adaptive* programming of the hash function. Instead, we rely on **P1** and **P3** to program the hash function in advance. More specifically, the trapdoor information will consist of q random tapes r_i (one for simulating each proof queried by the adversary) and the corresponding q challenges β_i (that can be pre-computed as a function of r_i, relying on **P1**). Since the challenges have the correct distribution, we can use the underlying HVZK simulator to simulate the proofs; here is where we need **P3**, as the

simulator has to pre-compute the values α_i in order to embed the β_i values on the correct points.

A caveat is that our simulator needs to know the value of q in advance; for this reason we only get a weaker *bounded* flavor of the zero-knowledge property where there exists a "universal" simulator that works for all adversaries asking q queries, for some a-priori fixed value of q. Note, however, that the CRS—as it contains the description of a q-wise independent hash function—needs to grow with q, and hence bound q should be seen as a parameter of the construction rather than a parameter of the simulator.

It is an interesting open problem whether this limitation can be removed, thus proving that actually our transformation achieves unbounded zero-knowledge.

Putting it Together. We will call 3PC arguments satisfying properties **P1**-**P3** above (besides completeness and soundness) *highly sound* 3PC arguments. The theorem below summarizes the above discussion. Its proof is deferred to the full version [39].

Theorem 1. *Let* $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V})$ *be a* highly sound *3PC argument system for an NP language L, and* H *be a programmable q-wise independent hash function. Then, the FS collapse* $\overline{\Pi}_{\mathrm{FS},\mathsf{H}}$ *of* Π *using* H *yields a q-bounded NIZK argument system for L.*

3 Compilers

It remains to construct a highly sound 3PC argument, and to understand which languages admit such arguments. Unfortunately we do not know of a natural highly sound 3PC argument. However, we do know of protocols that partially satisfy our requirements. For instance the classical 3PC argument for quadratic residuosity due to Blum [9] satisfies **P1**, and moreover can be shown to achieve completeness, soundness, and HVZK, but it does not directly meet **P2** and **P3**. Another interesting example is given by the Lapidot-Shamir protocol for the *NP*-complete problem of graph Hamiltonicity [37] (see also [41, Appendix B]). Here, the prover's commitment consists of a (statistically binding) commitment to the adjacency matrix of a random k-vertex cycle, where k is the size of the Hamiltonian cycle.[5] Hence, the protocol clearly satisfies **P1**. Additionally the simulator fakes the prover's commitment by either committing to a random k-vertex cycle, or by committing to the empty graph. Hence, the protocol also satisfies **P3**. As a corollary, we know that assuming non-interactive statistically binding commitment schemes (which follow from one-way permutations [9]), for all languages in *NP*, there exist 3PC protocols that satisfy completeness, computational soundness, and HVZK, as well as **P1** and **P3**.

Motivated by the above examples, we turn to the question whether it is possible to compile a 3PC protocol (with completeness, soundness, and HVZK) satisfying either **P1** or **P1** and **P3**, into a highly sound argument. Our compilers

[5] Note that the value k can be included in the language, and thus considered as public.

rely on several cryptographic tools (including indistinguishability obfuscation, puncturable PRFs, complexity leveraging and equivocal commitment schemes), and yield a 3PC in the CRS model; note that this means that we obtain an interactive protocol with a CRS even if the original protocol was in the standard model. It is an intriguing open problem if a highly sound argument can be constructed in the standard model, or whether a CRS is, in fact, necessary.

3.1 First Compiler

We present a compiler that turns a 3PC argument (possibly in the CRS model) with instance-independent commitments and HVZK (i.e., properties **P1** and **P3**) into a 3PC argument which has the soundness-error-to-guessing ratio (i.e., property **P2**) needed for the complexity leveraging in our positive result for FS NIZK. The idea for the compiler is to provide a mechanism that allows to produce many challenges β given only a single commitment α. To this effect the CRS will contain two obfuscated circuits to help the prover and the verifier run the protocol. For obfuscation we use an indistinguishability obfuscator. The first circuit C_0 is used by the prover to generate a *pre-commitment* α^* which it sends over to the verifier. The verifier will then use the second circuit C_1 and run it on α^* to obtain multiple commitments. For this $C_1[\mathsf{k}, \mathsf{crs}]$ has a PRF key (for function F) and the crs for algorithm P_0 of the underlying protocol hardcoded, and computes ℓ commitments as follows:

$C_1[\mathsf{k}, \mathsf{crs}](\alpha^*)$

for $i = 1, \ldots, \ell$ **do**
  $r^* \leftarrow \mathsf{F.Eval}(\mathsf{k}, \alpha^* + i)$
  $\boldsymbol{\alpha}[i] \leftarrow \mathsf{P}_0(\mathsf{crs}; r^*)$
return $\boldsymbol{\alpha}$

Using C_1 the compiled verifier V^* can generate ℓ real commitments $\boldsymbol{\alpha}[1]$ to $\boldsymbol{\alpha}[\ell]$ given the single (short) pre-commitment α^*. The verifier will then run the underlying verifier V on all these commitments to receive $\beta_1, \ldots, \beta_\ell$ which it sends back to the prover.

In order to correctly continue the prover's computation (which was started on the verifier's side) the compiled prover P^* needs to somehow obtain the randomnesses r^* used within C_1. For this, we will build a backdoor into C_1 which allows to obtain the randomness r^* if one knows the randomness that was used to generate α^*. Once the prover has recovered randomnesses $r_1^*, \ldots, r_\ell^*$ it can run the underlying prover P on this randomness and the corresponding challenges β_i to get correct values γ_i which it sends back to the verifier. In a final step verifier V^* runs the original verifier on the implicit transcripts $(\alpha_i, \beta_i, \gamma_i)_{i=1,\ldots,\ell}$ and returns 1 if and only if the original verifier returns 1 on all the transcripts.

Compiler Description. Let $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V})$ be a 3PC argument system where the prover generates instance-independent commitments and that satisfies instance-independent HVZK. Let rl denote an upper bound on the randomness used by the prover (i.e., $\mathsf{P.rl}$) and HVZK simulator (i.e., $\mathcal{S}.\mathsf{rl}$). Let F_1 be a puncturable pseudorandom function which is length doubling. Let F_2 be a puncturable pseudorandom function with $\mathsf{F}_2.\mathsf{il} = \mathsf{F}_1.\mathsf{ol}$ and with $\mathsf{F}_2.\mathsf{ol} = \mathsf{rl}$. Let ℓ be a polynomial. We construct an argument system $\Pi^* = (\mathsf{K}^*, \mathsf{P}^*, \mathsf{V}^*)$ in the CRS model as follows. On input the security parameter K^* will construct an obfuscation of the following two circuits:

$\mathsf{K}^*(1^\lambda)$

$\mathsf{crs} \leftarrow\!\!\$\ \mathsf{K}(1^\lambda)$
$\mathsf{k}_1 \leftarrow\!\!\$\ \mathsf{F}_1.\mathsf{KGen}(1^\lambda)$
$\mathsf{k}_2 \leftarrow\!\!\$\ \mathsf{F}_2.\mathsf{KGen}(1^\lambda)$
$\overline{C}_0 \leftarrow\!\!\$\ \mathsf{iO}(C_0[\mathsf{k}_1])$
$\overline{C}_1 \leftarrow\!\!\$\ \mathsf{iO}(C_1[\mathsf{k}_1, \mathsf{k}_2, \ell, \mathsf{crs}])$
$\overline{\mathsf{crs}} \leftarrow (\mathsf{crs}, \overline{C}_0, \overline{C}_1)$
return $\overline{\mathsf{crs}}$

$C_0[\mathsf{k}_1](\tau)$

$\alpha^* \leftarrow \mathsf{F}_1.\mathsf{Eval}(\mathsf{k}_1, \tau)$
return α^*

$C_1[\mathsf{k}_1, \mathsf{k}_2, \ell, \mathsf{crs}](\alpha^*, \tau)$

for $i = 1, \ldots, \ell$ **do**
  $\mathbf{r}^*[i] \leftarrow \mathsf{F}_2.\mathsf{Eval}(\mathsf{k}_2, \alpha^* + i)$
  $\mathsf{a}[i] \leftarrow \mathsf{P}_0(\mathsf{crs}; \mathbf{r}^*[i])$
if $\alpha^* \neq \mathsf{F}_1.\mathsf{Eval}(\mathsf{k}_1, \tau)$ **then**
  $\mathbf{r}^*[i] \leftarrow \bot$
return $(\mathsf{a}, \mathbf{r}^*)$

Note that we assume that the underlying protocol is in the CRS model and has a setup algorithm K. If this is not the case one recovers the transformation for a 3PC in the standard model by assuming that K outputs the empty string ε. The compiled 3PC $\Pi^* = (\mathsf{K}^*, \mathsf{P}^*, \mathsf{V}^*)$ is then constructed as in Fig. 2.

Security Analysis. It remains to show that the compiled protocol is computationally sound, achieves (bounded) instance-independent HVZK, is complete, and that it has instance-independent commitments and a sufficient soundness-error-to-guessing ratio:

Theorem 2. *Let $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V})$ be a 3PC argument system for a polynomial-time computable relation R such that Π is c-complete and s-sound and has instance-independent commitments and satisfies q-bounded instance-independent HVZK. Let iO be an indistinguishability obfuscator and F_1 and F_2 puncturable pseudorandom functions. Let ℓ be a polynomial. Then, in the CRS model, the compiled protocol $\Pi^* = (\mathsf{K}^*, \mathsf{P}^*, \mathsf{V}^*)$ is $(\ell \cdot c)$-complete, $(2 \cdot s^{-\ell} + 2^{\mathsf{F}_1.\mathsf{ol}(\lambda)} s^{-\ell})$-sound, has a worst-case collision probability of $2^{-\mathsf{F}_1.\mathsf{il}(\lambda)}$, and satisfies q/ℓ-bounded instance-independent HVZK. Furthermore the compiled protocol has instance-independent commitments.*

The proof to the above theorem appears in the full version [39].

3.2 Second Compiler

Next, we present a compiler that turns a 3PC protocol with HVZK and instance-independent commitments (i.e., property **P1**) into a 3PC protocol in the CRS

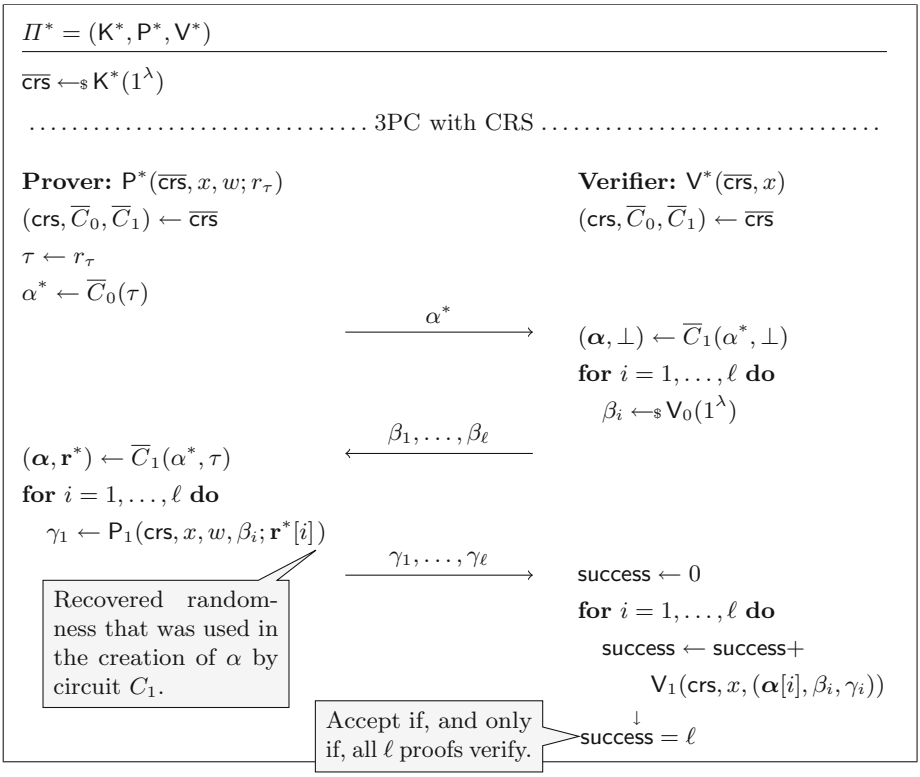

Fig. 2. The compiled protocol from Sect. 3.1 to turn a 3PC protocol into one that has a small soundness-error-to-guessing ratio (in the CRS model).

model that has instance-independent commitments *and* instance-independent simulators, that is, the HVZK simulator produces α and β independently of the instance (i.e., property **P3**).

The idea is inspired by Lindell's compiler [38]. Namely, we replace α by a commitment α^* to α where the deployed commitment scheme can come in one of two modes: if honestly generated the commitment will be *perfectly binding* thus allowing us to directly argue that the resulting compiled protocol retains soundness and completeness. On the other hand, the commitment scheme can be initialized to be *equivocal* (looking indistinguishably from the honest commitment setup) such that a simulator can open a commitment to arbitrary values. This way, the simulator can first commit to an arbitrary α^* and then, using the trapdoor in the CRS, it can open α^* to some arbitrary value α. In particular, in the reduction to the HVZK property, the verifier can choose α^* before knowing the statement that the simulator of the underlying protocol needs in order to produce α.

We refer the reader to the full paper [39] for a formal description of the above compiler, and for its security analysis.

4 Fiat–Shamir Signatures

Our techniques can be generalized in order to obtain a standard model instantiation of FS signatures, under similar complexity assumptions as in the case of FS NIZK. In particular it is possible to identify a certain class of so-called highly sound identification (ID) schemes, such that one can instantiate the hash function in the corresponding FS collapse via a q-wise independent hash function. As discussed in the introduction, the obtained signature scheme satisfies the weaker property of q-bounded random-message unforgeability against random-message attacks. Since the actual details of the instantiation are somewhat similar to the case of FS NIZK discussed above, we refer the reader to the full paper [39] for a more throughout discussion.

Acknowledgments. We are grateful to Christina Brzuska for her active participation in this research. Her ideas, feedback and suggestions played an essential part in the development of this work.

We thank Nils Fleischhacker and Markulf Kohlweiss for helpful comments on the presentation. We are grateful to an anonymous reviewer of TCC 2016 for pointing out that the constant hash function already suffices for obtaining a 1-bounded NIZK assuming properties **P1**-**P3** and thereby inspiring using a q-wise independent hash-function as instantiation. Before, we used a more complicated construction based on indistinguishability obfuscation and puncturable PRFs. We also thank the reviewer for pointing out the Blum-Lapidot-Shamir protocol, and we thank Ivan Visconti for helpful discussions and clarifications on the Blum-Lapidot-Shamir protocol.

References

1. 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)
2. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd Annual Symposium on Foundations of Computer Science, 14–17 October 2001, pp. 106–115. IEEE Computer Society Press, Las Vegas (2001)
3. Barak, B., Lindell, Y., Vadhan, S.P.: Lower bounds for non-black-box zero knowledge. In: 44th Annual Symposium on Foundations of Computer Science, 11–14 October 2003, pp. 384–393. IEEE Computer Society Press, Cambridge (2003)
4. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 93: 1st Conference on Computer and Communications Security, 3–5 November 1993, pp. 62–73. ACM Press, Fairfax (1993)
5. Bellare, M., Shoup, S.: Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 201–216. Springer, Heidelberg (2007)
6. 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)

7. Bitansky, N., Dachman-Soled, D., Garg, S., Jain, A., Kalai, Y.T., López-Alt, A., Wichs, D.: Why "Fiat-Shamir for proofs" lacks a proof. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 182–201. Springer, Heidelberg (2013)
8. Bitansky, N., Garg, S., Wichs, D.: Why Fiat-Shamir for proofs lacks a proof. Cryptology ePrint Archive, Report 2012/705 (2012). http://eprint.iacr.org/2012/705
9. Blum, M.: Coin flipping by telephone. In: Gersho, A. (ed.) Advances in Cryptology - CRYPTO 1981. ECE Report 82–04, pp. 11–15. U.C. Santa Barbara, Department of Electrical and Computer Engineering, Santa Barbara, CA, USA (1981)
10. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th Annual ACM Symposium on Theory of Computing, 2–4 May 1988, pp. 103–112. ACM Press, Chicago (1988)
11. Camenisch, J.L., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 302–321. Springer, Heidelberg (2005)
12. Camenisch, J.L., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002)
13. Canetti, R., Chen, Y., Reyzin, L.: On the correlation intractability of obfuscated pseudorandom functions. Cryptology ePrint Archive, Report 2015/334 (2015). http://eprint.iacr.org/
14. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th Annual ACM Symposium on Theory of Computing, 23–26 May 1998, pp. 209–218. ACM Press, Dallas (1988)
15. Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Improved OR-composition of sigma-protocols. In: Kushilevitz, E., et al. (eds.) TCC 2016-A. LNCS, vol. 9563, pp. 112–141. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49099-0_5
16. Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Online/offline or composition of sigma protocols. Cryptology ePrint Archive, Report 2016/175 (2016). http://eprint.iacr.org/
17. 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., et al. (eds.) TCC 2016-A. LNCS, vol. 9563, pp. 83–111. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49099-0_4
18. Dachman-Soled, D., Jain, A., Kalai, Y.T., López-Alt, A.: On the (in)security of the Fiat-Shamir paradigm, revisited. IACR Cryptology ePrint Archive 2012, 706 (2012). http://eprint.iacr.org/2012/706
19. Dagdelen, Ö., Venturi, D.: A second look at Fischlin's transformation. In: Pointcheval, D., Vergnaud, D. (eds.) AFRICACRYPT 2014. LNCS, vol. 8469, pp. 356–376. Springer, Heidelberg (2014)
20. Damgård, I.B.: Efficient concurrent zero-knowledge in the auxiliary string model. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 418–430. Springer, Heidelberg (2000)
21. 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)
22. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: 23rd Annual ACM Symposium on Theory of Computing, 6–8 May 1991, pp. 542–552. ACM Press, New Orleans (1991)

23. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science, 17–19 October 1999, pp. 523–534. IEEE Computer Society Press, New York (1999)
24. Elkind, E., Lipmaa, H.: Interleaving cryptography and mechanism design. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 117–131. Springer, Heidelberg (2004)
25. 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)
26. 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)
27. 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)
28. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th Annual ACM Symposium on Theory of Computing, 25–27 May 1987, pp. 218–229. ACM Press, New York City (1987)
29. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th Annual Symposium on Foundations of Computer Science, 11–14 October 2003, pp. 102–115. IEEE Computer Society Press, Cambridge (2003)
30. 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, May 31–June 3 2014, pp. 515–524. ACM Press, New York (2014)
31. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
32. Haitner, I.: A parallel repetition theorem for any interactive argument. In: 50th Annual Symposium on Foundations of Computer Science, 25–27 October 2009, pp. 241–250. IEEE Computer Society Press, Atlanta (2009)
33. Hazay, C., Venkitasubramaniam, M.: On the power of secure two-party computation. Cryptology ePrint Archive, Report 2016/074 (2016). http://eprint.iacr.org/
34. Hohenberger, S., Sahai, A., Waters, B.: Replacing a random oracle: full domain hash from indistinguishability obfuscation. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 201–220. Springer, Heidelberg (2014)
35. Kalai, Y.T., Rothblum, G.N., Rothblum, R.D.: From obfuscation to the security of Fiat-Shamir for proofs. Cryptology ePrint Archive, Report 2016/303 (2016). http://eprint.iacr.org/
36. Kiayias, A., Zacharias, T., Zhang, B.: End-to-end verifiable elections in the standard model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 468–498. Springer, Heidelberg (2015)
37. Lapidot, D., Shamir, A.: Publicly verifiable non-interactive zero-knowledge proofs. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 353–365. Springer, Heidelberg (1991)
38. 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, Part I. LNCS, vol. 9014, pp. 93–109. Springer, Heidelberg (2015)
39. Mittelbach, A., Venturi, D.: Fiat-Shamir for highly sound protocols is instantiable. IACR Cryptology ePrint Archive 2016, 313 (2016). http://eprint.iacr.org/2016/313

40. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (1993)
41. Ostrovsky, R., Visconti, I.: Simultaneous resettability from collision resistance. Electronic Colloquium on Computational Complexity (ECCC) 19, 164 (2012). http://eccc.hpi-web.de/report/2012/164
42. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**(3), 361–396 (2000)
43. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th Annual ACM Symposium on Theory of Computing, May 31–June 3 2014, pp. 475–484. ACM Press, New York (2014)