Efficient Perfectly Sound One-message Zero-Knowledge Proofs via Oracle-aided Simulation

Vincenzo Iovino[1]

[1]University of Salerno,
vinciovino@gmail.com

Abstract. In this paper we put forth new efficient *one-message* proof systems for several practical applications, like proving that an El Gamal ciphertext (over a multiplicative group) decrypts to a given value and correctness of a shuffle. Our proof systems are built from multiplicative groups of hidden order, are not based on any setup/trust assumption like the RO or the common reference string model and are *perfectly sound*, that is they are written proofs in the sense of mathematics.

Our proof systems satisfy a generalization of zero-knowledge (ZK) that we call *harmless* zero-knowledge (HZK). The simulator of an O-HZK proof for a relation over a language L is given the additional capability of invoking an oracle O relative to which L is hard to decide. That is, the proof does not leak any knowledge that an adversary might not compute by itself interacting with an oracle O that does not help to decide the language.

Unlike ZK, non-interactivity and perfect soundness do not contradict HZK and HZK can replace ZK in any application in which, basically, the computational assumptions used in the application hold even against adversaries with access to O. An O-HZK proof is witness hiding (WH) for distributions hard against adversaries with access to O, and strong-WI when quantifying over distributions that are indistinguishable by adversaries with access to O. Moreover, an O-HZK proof is witness indistinguishable (and the property does not depend on the oracle).

We provide a specific oracle DHInvO that is enough powerful to make our main proof systems DHInvO-HZK but not trivial: indeed, we show concrete and practical cryptographic protocols that can be proven secure employing a DHInvO-HZK proof in the reduction and that are instead not achievable using traditional ZK (unless resorting to the CRS/RO models).

Efficient one-message proof systems with perfect soundness were only known for relations over bilinear groups and were proven only witness indistinguishable.

As byproduct, we also obtain a perfectly sound non-interactive ZAP, WH and HZK proof system for $\mathcal{NP}$ relations from number-theoretic assumptions over multiplicative groups of hidden order. No non-interactive WH proof system for $\mathcal{NP}$ (neither for simpler non-trivial relations) was previously known.

Keywords: zero-knowledge, NIZK, RSA, witness hiding, ZAP.

Table of Contents

1 Introduction

Mathematical proofs vs interactive proofs. Zero-knowledge (ZK) proofs [GMR85] [GMR89,GMW91] represent one of the most important concepts in computer science and turned out to be one of the key ingredients in modern cryptographic protocol design. Unlike traditional mathematical proofs, ZK proofs require interaction and allow the prover to cheat the verifier with a non-zero probability of error.

Non-interactive proofs. Since the discovery of ZK proofs, the importance of removing interaction led to the introduction of non-interactive (NI) zero-knowledge (NIZK) systems [DMP88]. NIZKs have been extensively studied for about 30 years [BFM88,DMP88,FLS90,RS92,Gol01,DDO+01]. Indeed, 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.

Truly non-interactive ZK proofs for non-$\mathcal{BPP}$ languages are provably impossible to achieve [Gol01], so the initial constructions for NIZKs worked in the common reference string (CRS) model [DMP88] and because of various limitations (e.g., the need of $\mathcal{NP}$ reductions, the non-reusability of the CRS, the expensive computations) their impact was mainly in the theoretical foundations of cryptography. In the CRS model, the party who generates the CRS has to be trusted. Indeed, such party might either setup the CRS in a malicious way for which the soundness does not hold or hand a trapdoor to an adversary who can exploit the trapdoor to prove false theorems.[1]

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 a dishonest 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 [BCC88] has a significantly weaker soundness guarantee since it applies to efficient adversarial provers only.[2].

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

[1] In the rest of this work, when we say that an adversary "proves a false theorem" using a proof system we actually mean that it produces a string π such that the verifier of the proof system accepts (x, π) for a false statement x. Such string π is an alleged proof of x, not a proof in the mathematical sense.

[2] 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".

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 (e.g., SHA-3 [BDPA11]). NIZK arguments in the RO model obtained via the FS transform are orders of magnitude more efficient than the most efficient NIZK arguments in the CRS model [GS08].

Even if the RO methodology has been shown to be controversial already in [CGH98] and further negative results were published next [DNRS99,Bar01,GK03,Kal06] [BLV03,DRV12,BDSG+13,GOSV14,KRR17], NIZK arguments via the FS transform are widely used in concrete cryptographic protocols (e.g., in the popular Helios voting system [Adi08]). We remark that one could also consider a 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, when analyzing FS-derived NIZKs, we consider a truly unbounded adversarial prover. This difference can be crucial in applications.

The importance of unconditional soundness. In e-voting privacy cannot be achieved unconditionally unless losing universal verifiability or unless all the voters actually vote [CMFP+10]. There is instead no barrier to attain unconditional universal verifiability and the soundness guarantee of a ZK proof/argument used in an e-voting scheme impinges on the quality of the universal verifiability: an adversary that can break the soundness condition can subvert the result of the election.

In the context of proofs of solvency for cryptocurrencies [DBB+15,BBB+18] and similar applications, the soundness of the proof/argument system is significantly more important than the ZK property: the privacy of the transaction is expendable but breaking the soundness of the proof system gives the possibility of generating arbitrary coins blasting away the whole digital economy. The proof systems we will propose are unconditionally sound but are not short, so cannot be used in most of the applications of confidential transactions. Moreover, under widely accepted complexity assumptions, no proof system with short communication may exist [GH98]. Despite this fact, the new ideas we will introduce may open up the possibility of removing CRSs/ROs from the design of argument systems for this kind of applications.

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] [GOS12,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 $\mathcal{NP}$ reduction) that make the NIZK proof completely impractical, and 2) it requires some complexity assumptions (e.g., trapdoor permutations in [FLS90]) therefore additionally incurring a significant security loss in the zero-knowledge guarantee.

Furthermore, both the RO and the CRS model rely on trust assumptions and unproven hypotheses.

For languages relative to bilinear groups the situation is better and we have NIZK proof systems and non-interactive ZAPs [DN00] with perfect soundness [GOS06b,GS08]. However, bilinear groups are less efficient and their security is less studied than other number-theoretic problems and the security of ZAPs is limited to witness indistinguishability (WI) that, for instance, is a vacuous guarantee for single witness relations as it is the case for the relation of valid DH tuples. Furthermore, even for non-single witness relations WI poses several threats and issues in several scenarios (see, e.g., discussion on verifiable functional encryption in Section 1.3.8).

These limitations of the FS-transform, of the above trivial approach, of ZK proofs in general (non-zero soundness error and non-interactivity inherently based on trust assumptions), and of weaker WI proofs, motivate the main questions of this work.

Practical open question: *is there any efficient non-interactive proof system (i.e., soundness is guaranteed also against unbounded adversarial provers) for practical languages not related to bilinear groups that can be used in relevant cryptographic applications and satisfies a meaningful notion of privacy?*

Theoretical open question: *is there any achievable, non-trivial, useful and usable variant of ZK that is compatible with perfect soundness and true non-interactivity?*

Later, we will answer positively these questions in a very strong sense by presenting completely non-interactive proof systems both for practical applications and general statements satisfying perfect soundness, not based on any trusted parameter and enjoying a close variant of ZK. Our proofs are in particular proper *mathematical proofs*.

The FS transform internals. Before digging in our results, we will first discuss the limitations of the FS transform, the most known technique to construct efficient NIZKAs, and review its 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]. 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.

$\mathcal{P}$ sends a first message a to $\mathcal{V}$, also called the commitment. Then $\mathcal{V}$ sends back a random challenge c. Finally $\mathcal{P}$ 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 $\mathcal{V}$ takes deterministically the decision of accepting or not the transcript.

The FS transform constructs $\mathsf{NIZK} = (\mathsf{NIZK}.\mathcal{P}, \mathsf{NIZK}.\mathcal{V})$ as follows. $\mathsf{NIZK}.\mathcal{P}$ computes a precisely as $\mathcal{P}$, but then the challenge c of $\mathcal{V}$ is replaced by the output of the RO on input the statement x and a, i.e., $c = H(x, a)$.[3] Finally $\mathsf{NIZK}.\mathcal{P}$ computes z precisely as $\mathcal{P}$ 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 $\mathcal{V}$. Soundness can not be claimed when instead the adversarial prover is unbounded and can therefore make an unbounded number of queries to the RO.

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

If $\mathsf{3HVZK}$ satisfies special soundness (i.e., there is a deterministic efficient extractor that from 2 different accepting transcripts for the same statement with the same first message outputs a witness), then the resulting NIZK argument system additionally enjoys witness extraction but limited to PPT adversarial provers. Known variations [Pas03a,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 [Pas03a] and Fischlin [Fis05]) only attain *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). Moreover, to our knowledge, previous works on NIZK argument systems in the RO model only attained extraction (i.e., the proof of knowledge property) against bounded adversaries.

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 $k(\lambda) \in \omega(\log(\lambda)), b(\lambda) \geq \lambda + k(\lambda)$ and λ being the security parameter, and the prover computes the answer z deterministically based on (a, c). Suppose further 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 for some z.

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

(A natural Σ-protocol satisfying the above requirements will be shown soon. The latter hypothesis can be also seen to hold assuming that for each $x \notin L$, the soundness error is non-zero.)

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

For each commitment a the probability that the RO maps (x, a) into c such that (a, c, z) is an accepted transcript for some z is, by hypothesis, $\geq 2^{-k(\lambda)}$. Thus, since there are $2^{b(\lambda)} \geq 2^{\lambda+k(\lambda)}$ commitments, $\mathsf{NIZK}.\mathcal{P}^\star$ fails in proving the false statement x with probability $< (1 - \frac{1}{2^{k(\lambda)}})^{2^{\lambda+k(\lambda)}}$. Therefore, $\mathsf{NIZK}.\mathcal{P}^\star$ succeeds with probability $\geq 1 - (1 - \frac{1}{2^{k(\lambda)}})^{2^{k(\lambda)} \cdot 2^\lambda} \approx 1 - (\frac{1}{e})^{2^\lambda}$.[4]

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) tuple.

The prover chooses a random $r \leftarrow \mathbb{Z}_p$, where p is the order of the group, and sends the commitment $a \stackrel{\triangle}{=} g^r, b \stackrel{\triangle}{=} h^r$. The verifier sends a random challenge $e \leftarrow \mathbb{Z}_p$. (Note we use here the letter e to denote the challenge instead of the previous c.) The prover sends back deterministically $z \stackrel{\triangle}{=} r + e \cdot w \mod p$ and the verifier accepts iff $g^z = a \cdot u^e$ and $h^z = b \cdot v^e$.

Let λ be the security parameter and $k(\lambda) \stackrel{\triangle}{=} \lambda$ equal the length of the group elements. Then, challenges have length $k(\lambda)$, commitments have length $2 \cdot k(\lambda)$ and $k(\lambda)$ is also the soundness error. By the special HVZK property, 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. More concretely, for each false statement $x \stackrel{\triangle}{=} (g, h, u, v)$ with $u = g^{w_1}, v = g^{w_2}$, for $w_1 \not\equiv w_2 \mod p$, and each commitment $(a \stackrel{\triangle}{=} g^{r_1}, b \stackrel{\triangle}{=} g^{r_2})$ for $r_1, r_2 \in \mathbb{Z}_p$, the challenge $e \stackrel{\triangle}{=} \frac{(r_2 - r_1)}{(w_1 - w_2)} \mod p$ and the response $z \stackrel{\triangle}{=} r_1 + e \cdot w_1 \mod p$ satisfy the wished property. 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

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

$\mathcal{RO}(x||a_i||i)$. Unluckily, this strategy does not improve the situation. In fact, while the number of possible challenges increases (each challenge now consists of $k \cdot p$ bits) the number of possible commitments also increases. A simple analysis shows that an attack similar to the previous one can be applied to such variant of the FS transform as well. Observe also that the previous attack can be viewed as a special case for $p(\lambda) = 1$.

In fact, consider a false statement x and an unbounded prover $\mathsf{NIZK}.\mathcal{P}^\star$ 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, $\mathsf{NIZK}.\mathcal{P}^\star$ 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 be much bigger than the challenge space, as it is indeed the case in general for natural Σ-protocols for relations in which deciding membership is non-trivial. In fact, if for instance the challenge and commitment spaces had the same cardinality, the lower-bound on the winning probability of the previous prover would just be $\left(1 - \frac{1}{e}\right)^{p(\lambda)}$, a negligible function.

Moreover, it is very easy to observe that any ZK proof system, even for the RO model, cannot satisfy *perfect* soundness.

1.1 Our Results and Roadmap

The main result of this work is a *positive* answer to the above open questions: we construct new efficient perfectly sound *one-message* proof systems for practical languages satisfying a variant of zero-knowledge that we call *harmless zero-knowledge* (HZK). We provide an overview of HZK in Section 1.3 and in Section 2 we recall standard definitions and provide formal definitions for harmless ZK proof of knowledge systems. In Sections 2.2.4, 2.2.5 and 2.2.6 we present relations of our new notion to related ones, in particular demonstrating that, under some computational assumptions, HZK proofs are witness hiding (sketch in Section 1.3.4) and strong-witness indistinguishable (sketch in Section 1.3.9).

In Section 3 (overview in Section 1.2) we present our new proof system for proving that an El Gamal ciphertext decrypts to a given value, for an El Gamal encryption scheme instantiated over a group of hidden order (more details in Sections 1.2 and 2). In Section 1.4.1 we show a variant of our proof system for proving correctness of a shuffle of El Gamal ciphertexts (instantiated over the same group).

In Section 1.4.2 we build OR proofs from proofs of correct shuffle, and in Section 1.4.3 we construct direct proofs for polynomial statements.

All our proof systems enjoy perfect soundness, are non-interactive and do not assume any trust assumptions like the Common Reference String (CRS) model or the RO model and do not assume any bound on the space of the verifier.

If the group parameter on which DH tuples are based is seen as a common parameter and is made public (though we do not require it to be setup in a trusted way), our proof for DH tuples also satisfies the standard definition of perfect extraction [GOS12] and additionally enjoys what we call harmless proof of knowledge (see Sections 1.3 and 2).

In Section 1.4.4 we construct a one-message perfectly sound WI proof (ZAP) for Boolean circuit satisfiability from a number-theoretic assumption related to DH over groups of hidden order. Our ZAP is also *computational* HZK and we sketch that, using know complexity leveraging arguments and techniques, it can be tweaked to be quasi-polynomial time simulatable.

Our proof systems are sufficient to construct an e-voting scheme in which authorities have zero probability of subverting the result of the election. This application is presented in Sections 1.2.3 and 1.3.8. In Section 1.3.8, we also show how to use harmless ZK proof of knowledge systems and other applications to verifiable functional encryption and to CCA1-security.

In Section 1.5 we survey the known literature in the field and compare our results to it.

1.2 Overview of our main proof system

Before describing our new proof system, we recall the standard NIZK argument in the programmable RO model for proving correct decryption of an El Gamal ciphertext due to Chaum and Pedersen [CP93].

1.2.1 Standard NIZKA for correctness of El Gamal decryption

Consider an (exponential) El Gamal ciphertext for public key $\mathsf{pk} = g^w$ and message m: $(a = g^r, b = \mathsf{pk}^r \cdot g^m)$, with g known generator of a group of prime order p (e.g., the group of quadratic residues modulo a prime q such that $q = 2p + 1$ for a prime p). To prove that this ciphertext decrypts to m without revealing any information on the secret-key w, one can prove that the following tuple is DH: $(g, h \stackrel{\triangle}{=} a, u \stackrel{\triangle}{=} \mathsf{pk} = g^w, v \stackrel{\triangle}{=} b/g^m = h^w)$. Therefore, the problem of proving that an El Gamal ciphertext decrypts to some message boils down to proving that a tuple (g, h, u, v) is DH.

The standard Chaum-Pedersen's interactive proof for DH tuples [CP93] is the following. Let (g, h, u, v) be a DH tuple with witness w, i.e., $u = g^w, v = h^w$ for some $w \in \mathbb{Z}_p$. The prover knows the witness but the verifier does not and both the prover and the verifier share the common input (g, h, u, v).

- The prover sends $a = g^r, b = h^r$ to the verifier.
- The verifier sends a challenge $e \leftarrow \mathbb{Z}_p$.
- The prover sends back $z = r + e \cdot w \mod p$.
- The verifier accepts iff $g^z = a \cdot u^e, h^z = b \cdot v^e$.

Let us analyze the soundness. Let g, h, u, v be a non-DH tuple, that is $g, h, u = g^{w_1}, v = h^{w_2}$ for $w_1 \not\equiv w_2 \mod p$. Let $a = g^{r_1}, b = h^{r_2}$ for some $r_1, r_2 \in \mathbb{Z}_p$ (the first message of the prover can be possibly ill-formed when $r_1 \not\equiv r_2 \mod p$).

If the verifier accepts, the equations checked by the verifier imply that 1) $z = r_1 + e \cdot w_1 \mod p$ and $z = r_2 + e \cdot w_2 \mod p$. Subtracting the equations together, it holds that $r_1 - r_2 = e \cdot (w_1 - w_2) \mod p$. This means that for each

w_1, w_2, r_1, r_2 there is exactly one value e (i.e., $e = (w_1 - w_2)/(e_1 - e_2) \mod p$) that satisfies the equations, and thus if e is randomly chosen the probability that the verifier accepts the false statement is $\frac{1}{p}$, a quantity negligible in $|p|$.

The proof is made non-interactive via the FS transform, i.e., computing $e = \mathcal{RO}((g, h, u, v), a, b)$. As we analyzed before, applying the FS transform statistical soundness is lost. For completeness, we sketch again the argument. Fix a false statement (g, h, u, v). The unbounded prover can search over all values $a = g^{r_1}$ and $b = g^{r_2}$ until it finds a pair (a, b) such that $\mathcal{RO}(g, h, u, v, a, b)$ equals the only value e that satisfies the equation $e = (w_1 - w_2)/(e_1 - e_2) \mod p$. As there are 2^{2k} possible pairs (a, b) and 2^k possible values of e, with $k \stackrel{\triangle}{=} |p|$, an unbounded prover succeeds with overwhelming probability in proving any false statement. Moreover, in practice one chooses a fixed hash function and in this case nothing can be said about the security: it might be that the hash function maps a false statement and a pair (a, b) into the only one "bad" e that satisfies the equations checked by the verifier.

In the following, we will propose a new proof system that both removes the use of the hash function (and thus it is not based on the RO heuristic or any other trust assumption or limitation) and achieves perfect soundness. Recall that perfect soundness cannot be attained even in the interactive case for ZK proofs. This system will also serve as base to build a proof system for general $\mathcal{NP}$ statements (see Section 1.4.4).

1.2.2 Our new non-interactive proof system Observe that in the above proof the prover can cheat only when $r_1 \not\equiv r_2 \mod p$. To prevent such possibility of cheating, one could require the prover to send the value r in the clear in the first round but this is insecure as the prover also sends the value $z \stackrel{\triangle}{=} r + e \cdot w \mod p$ in the last round. Sending instead $a \stackrel{\triangle}{=} g^z$ and $b \stackrel{\triangle}{=} h^z$ would not reveal z and would still allow the verifier to check the equations. In this case the verifier would need to additionally verify the well-formedness of the pair (a, b), i.e., that $a = g^r, b = g^r$, for some $r \in \mathbb{Z}_p$. Apparently, a way of proving (a, b) to be well-formed seems as difficult as proving a tuple to be DH. However, we will show that this can be done working in groups of hidden order.

Switching to groups of hidden order. Let us analyze the following completely non-interactive proof system (in particular, the verifier does not longer need to send any challenge to the prover). Let N be a Blum integer and consider the group of quadratic residues modulo N. In this group, we can construct an El Gamal-like encryption scheme (see Def. 26). Both the prover and the verifier share the tuple (g, h, u, v) and the modulus N. The order of the group of quadratic residues modulo N is hidden and equals $m \stackrel{\triangle}{=} \phi(N)/4$.

The aim of the prover is to convince the verifier that the tuple (g, h, u, v) is DH for witness $w \in \mathbb{Z}_{\phi(N)}$. In Section 1.2.3, we will show that our proof can be used to prove that a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$ for public key pk decrypts to g^0.

Our first version of the proof system NIDDH assumes u (resp. v) to be in the same subgroup generated by g (resp. h) and we will subsequently show another proof system NISG that will be used in the final version of NIDDH to remove this restriction. However, we stress that in the analysis of NIDDH and NISG we will never assume N to be well-formed.

To the aim of highlighting potential attacks, we will first present a NI proof system subject to an attack and we will later show how to patch it.

A first attempt. The prover of NIDDH (in its first insecure version), on input a DH tuple (g, h, u, v) and a factorization of N, sends the following (non-interactive) proof:

$$r, X \stackrel{\triangle}{=} g^z, Y \stackrel{\triangle}{=} h^z, z' \stackrel{\triangle}{=} z^{-1} \mod \phi(N),$$

with $r \leftarrow \mathbb{Z}^\star_{\phi(N)}$ and $z \stackrel{\triangle}{=} r + w \mod \phi(N)$ subject to the following constraints: $z \in \mathbb{Z}^\star_{\phi(N)}$ and z' have to be prime numbers. (The reason for z' to be prime will be explained later when we will also propose a change in the proof. We stress that we *require* z to be in $\mathbb{Z}^\star_{\phi(N)}$, that is the prover has to find randomness r such that $r + w \mod \phi(N)$ satisfies the constraint; it is easy to design an algorithm that computes values satisfying such constraints w.v.h.p.) Note that, notwithstanding the group is of "hidden order", the prover can compute $z^{-1} \mod \phi(N)$ from the factorization of N, that is the order of the group is hidden to the verifier but not to the prover.

The verifier of NIDDH is given N and the tuple (but not the factorization). The verifier accepts the proof if and only if z' is a prime number and all the following equalities hold:

$$X^{z'} = g, Y^{z'} = h, X = g^r \cdot u, Y = h^r \cdot v.$$

The idea is that z' should allow to verify that X and Y are such that $\mathbf{dlog}_g X = \mathbf{dlog}_h X$, i.e., $X = g^z, Y = h^z$ for some $z < \phi(N)$. Then, the soundness would follow from the observations highlighted above. However, the checks are not sufficient to guarantee soundness and we actually need some modifications as explained next.

A potential issue, how to fix it and soundness analysis of our first attempt. Notice that the soundness of the previous proof relies on the fact that z' should allow to check the well-formedness of the pair (X, Y), i.e., check that $X = g^t, Y = h^t$ for some non-negative integer $t < \phi(N)$. However, it might be that $X^{z'} = g, Y^{z'} = h$ but $\mathbf{dlog}_g X \neq \mathbf{dlog}_h Y$.[5] Thus, the previous checks are not sufficient. We guarantee this case cannot occur as follows.

Observe that if z' has no common factors with $\phi(N)$ and $X^{z'} = g, Y^{z'} = h$, then $X = g^t, Y = h^t$ for some integer $t < \phi(N)$. This can be seen by setting $t \stackrel{\triangle}{=} z'^{-1} \mod \phi(N)$. So, let us analyze the soundness supposing z' to be co-prime with $\phi(N)$. Recall that we are assuming u (resp. v) to be in the same subgroup

[5] For example, if $N = 35, g = 8, X = 2$, we have that $X^3 = g$ but $\mathbf{dlog}_g X$ does not exist, that is there is no x such that $8^x \equiv 2 \mod 35$.

generated by g (resp. h). We will later show how to remove this restriction. Let us assume for simplicity $\mathrm{ord}(g) = \mathrm{ord}(h)$ (see the general case in Theorem 15) and let $k \triangleq \mathrm{ord}(g) = \mathrm{ord}(h)$. The verifier of NIDDH checks that $X = g^r \cdot u$ and $Y = h^r \cdot v$. By hypothesis, $u = g^{w_1}$ and $v = h^{w_2}$ for some $w_1, w_2 < k$. Letting $t \triangleq z'^{-1} \mod \phi(N)$ and taking the discrete logs, resp. in base g and h, we have that $t \mod k = r + w_1 \mod k$ and $t \mod k = r + w_2 \mod k$. So, we have $w_1 \mod k = w_2 \mod k$, for some $k < \phi(N)$, thus $w_1 = w_2$. Therefore, there exists $w < k \leq \phi(N)$ such that $u = g^w$ and $v = h^w$, as it was to prove.

Therefore, what is left to guarantee soundness is to enforce z' to be co-prime with $\phi(N)$. This can be done as follows. The prover repeats the basic NIDDH protocol in parallel a sufficient number of times s setting in the i-th execution, for $i = 1, \ldots, s$, the value z'_i to be prime and setting all z_i's to be *different*. If a dishonest prover could set for each $i \in [s]$ the value z'_i to have a common factor with $\phi(N)$, we would have a contradiction. Indeed, it is not possible for all z'_i's to have a factor in common with the order of the group $\mathbb{Z}_N^\star$ assuming s to be, e.g., greater or equal than the maximum possible number of factors of $\phi(N)$. See details in Theorem 15.

Observe that the soundness of NIDDH, as described so far, does not to rely on the well-formedness of the modulus N: whatever the modulus N is, the prover cannot cheat.

Proof of knowledge with perfect extraction. Our proof system NIDDH (in its first insecure version) has perfect extraction according to the standard definition [GOS12]. The common reference string (that has not to be trusted in our case) can be set to the modulus N. The extractor computes the modulo N with knowledge of the factorization, and thus of the hidden order $\phi(N)$. Given a proof accepted by the verifier and $\phi(N)$, the prover inverts $z' = z^{-1} \mod \phi(N) = (r+w)^{-1} \mod \phi(N)$ to compute $z = r + w \mod \phi(N)$ and subtracts from it $r \mod \phi(N)$ to compute w, the witness.

So the NI satisfies perfect extraction assuming a common parameter (the modulus N) is set and made public at the beginning of the protocol. On the other hand, NIZKAs obtained via FS transform suffers annoying rewinding issues. We will later show that our proofs systems additionally enjoy a generalization of proof of knowledge to a purely non-interactive setting.

Guaranteeing that u belong to the subgroup generated by g. The previous proof of knowledge system NIDDH (in its first insecure version) can be simplified to a Schnorr-like non-interactive proof of knowledge system NISG to prove that an element u belongs to the subgroup generated by g, i.e., that $u = g^w$ for some $w < \phi(N)$. (This proof of knowledge system is still subject to the linear attacks we will describe next but later we will show a patch against them that applies both to NIDDH and NISG.)

Consider the following proof of knowledge system NISG. The prover sends $r, z' \triangleq z^{-1} \mod \phi(N)$, with $z \triangleq r + w \mod \phi(N) \in \mathbb{Z}^\star_{\phi(N)}$ and z' prime number. Let $H \triangleq g^r \cdot u$. The verifier checks that 1) $H^{z'} = g$.

The soundness should follow from the fact that, for each Y and X both $\neq 1$ and for each number z' co-prime with $\phi(N)$, if $Y^{z'} = X$ then Y belongs to the subgroup generated by X. This can be proven as follows. Let $t \triangleq z'^{-1} \mod \phi(N)$, then $X^t = Y$ and thus Y belongs to the subgroup generated by X (note that the inverse of z' exists as z' if is co-prime with $\phi(N)$). Guaranteeing z' to be co-prime with $\phi(N)$ can be done as shown above with the trick of the repetitions. For simplicity, henceforth we assume z' to be co-prime with $\phi(N)$.

The check 1) implies that $H = g^t$ for some $t \in \mathbb{Z}_\phi(N)$ and thus $u = H \cdot g^{-r} = g^{t-r}$, that is u is in the subgroup generated by g as well.

Previously, we assumed NIDDH to work under the hypothesis of u (resp. v) being in the same subgroup generated by g (resp. h). To remove such a restriction, we require the prover of NIDDH to first invoke the prover of NISG to prove u (resp. v) to be in the same subgroup generated by g (resp. h).

A linear attack against our first attempts of NIDDH *and* NISG. Hereafter, for simplicity we are not considering the aforementioned modification to NIDDH that introduces parallel repetitions and invokes NISG as sub-protocol. Recall that in NIDDH, $r \triangleq z - w \mod \phi(N)$. Let $s \triangleq z^{-1} \mod \phi(N)$. The verifier can multiply (over the integers) s by r to get $1 + s \cdot w \mod \phi(N)$ and in turn, subtracting 1 (over the integers), can get $s \cdot w \mod \phi(N)$. Given another pair of group elements g^t and h^t, an attacker can power g^t to $s \cdot w \mod \phi(N)$ and h^t to s to check that the tuple (g, h, g^t, h^t) is DH for witness w, a destructive attack.

Potential attacks on multiple proofs. Given two proofs, it is possible to get a multiple of the order of the group, from which it is possible to factorize the modulus N. Indeed, suppose to have two proofs for the same witness. That is, suppose we have $z_1 \triangleq r_1 + w \mod \phi(N)$ and $z_2 \triangleq r_2 + w \mod \phi(N)$. Combining them together we obtain $z_1 - z_2 \equiv r_1 - r_2 \mod \phi(N)$. Multiplying over the integers this value by $z'_1 \triangleq (r_1 + w)^{-1} \mod \phi(N)$, we have $z'_1 \cdot (r_1 - r_2) \equiv 1 - z_2 \cdot z'_1 \mod \phi(N)$. Subtracting over the integers by 1 and multiplying over the integers by $z'_2 \triangleq (r_2 + w)^{-1} \mod \phi(N)$ we finally obtain that $z'_2(z'_1 \cdot (r_1 - r_2) - 1) \equiv z'_1 \mod \phi(N)$.

Therefore, $z'_2 \cdot (z'_1 \cdot (r_1 - r_2) - 1) - z'_1$ is a multiple of $\phi(N)$ and, by standard techniques, the hidden order $\phi(N)$ can be computed.

Defense against linear attacks and how to patch NIDDH *and* NISG. We now show how to counter the previous attacks. Yet, for simplicity we are not considering the modification to NIDDH and NISG that introduces parallel repetitions and the need for NIDDH to invoke NISG as sub-protocol. The insecurity of the NI proof of knowledge systems NIDDH and NISG, as presented so far, comes from the fact that the proof contains the value r in the clear. Such value can be multiplied by z' to get a value of the form $z' \cdot w \mod \phi(N)$, a fatal attack.

To overcome this attack, we require the prover to send the pair $a \triangleq g^r, b \triangleq h^r$ as in the original Chaum-Pedersen's proof. In addition the prover has to send the value $r^{-1} \mod \phi(N)$ that can be used by the verifier to check the well-formedness of the pair (a, b) (all previous considerations and the need for parallel repetition apply in this case as well). In this case the prover of NIDDH (similar consideration holds for NISG) does not need to transmit g^z and h^z as they can be derived from g^r, h^r and u, v.

The previous soundness analysis stays roughly unchanged except for the following. As we detailed, the overall proof of NIDDH essentially consists of s repetitions of a basic sub-proof. For each $i \in [s]$, the sub-proof contains two values r'_i and z'_i that have to be prime numbers to guarantee that there exists some $j \in [s]$ such that, e.g., r'_j is co-prime with $\phi(N)$, and in turn this ensures that the j-th sub-proof is accepted. The issue is that it might be that the only values co-prime with $\phi(N)$ are r'_j and z'_{j_2}, for some $j \neq j_2$. To prevent this problem to arise, our verifier is slightly more intricate. We defer to Section 3 for the full description of NIDDH. Observe that a sort of perfect extraction still holds: from the factorization of N and $r^{-1} \mod \phi(N)$, the value r and thus w can be computed; we will later discuss how to exploit this fact.

What privacy? The so modified proof systems seems to withstand the aforementioned linear attacks: given $r^{-1} \mod \phi(N)$ and $(r + w)^{-1} \mod \phi(N)$, the attacker cannot seemingly form a multiple of the witness. This patch also appears to protect against the attacks on multiple proofs. Does the overall proof systems NIDDH and NISG satisfy a reasonable notion of "privacy" and what kind of security is it attained? This question will be discussed in depth in Section 1.3.

The overall detailed constructions and analysis for NIDDH and NISG are presented in Section 3.

Why is working in a group of hidden order not *a trust assumption?* One could naively think that working in a group of hidden order is a trust assumption. A trust assumption for proof systems requires a parameter to be chosen correctly and that the generator of the parameter be a trusted party who cannot collude with the adversaries against the proof system. If this is not the case (if the parameters are not correctly chosen or the generator colludes with the adversary), the security may not hold.

In our main proof systems NIDDH and NISG, *whatever* modulus N and generator are adversarially chosen by the prover, no proof for a false statement can be generated. That is, even if N and the group are setup in an incorrect way, no proof for a false statement can be produced. Notice that the prover must *not* convince the verifier that the group has the right form. Indeed, our guarantee is that there exists *no* proof for a false statement at all (relative to any modulus and generator), so the possibility of cheating is null.

What about privacy? If a parameter is ill-formed, can a proof leak information about the witness to the statement to prove? The answer is positive but this is *inherent* in proof systems. For any proof system, the prover could compute the proof in an ill-formed way (e.g., choosing the randomness not uniformly) so that the proof leak knowledge. Moreover, the prover can ever collude with an adversary to hand the adversary a witness to the statement to prove. Therefore, for any reasonable notion of privacy for proof systems, the randomness used to compute a proof has to be computed honestly and the prover cannot collude with adversaries against the privacy, and our NI proofs are no exception in this respect.

1.2.3 Proof of correct decryption and its applications In Section 1.3.8 we will show that (actually, we will analyze a more general case) no PPT adversary can win with non-negligible probability in the following game against a challenger $\mathcal{C}$. The challenger $\mathcal{C}$ selects a random bit b, computes a well-formed DH tuple $T \triangleq (, g, h, u, v)$ over $\mathbb{Z}_N^\star$ and gives the adversary two tuples (T_0^b, T_1^b), with $T_0^b \triangleq (g, h, u.v \cdot g^1)$ and $T_1^b \triangleq (g, h, u, v \cdot g^{-1})$, and additionally a NIDDH proof that T is a well-formed DH tuple. The adversary outputs a bit b' and wins iff $b = b'$. Note that this is a non-trivial problem. To our knowledge, it was not know how to prove that no PPT adversary can win with non-negligible probability in the previous game instantiated with any other (completely) non-interactive proof (unless assuming trusted parameters.

An issue in applying NIDDH *to proofs of correct decryption and how to solve it.* The proof system NIDDH for DH tuples can be also used to prove that a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$ for public key pk decrypts to g^0 as follows. Observe first that if ct_1 belongs to the subgroup generated by g, then the tuple $(g, \mathsf{ct}_1, \mathsf{pk}, \mathsf{ct}_2)$ is DH if and only if ct decrypts to $g^0 = 1$. However, ct_1 might not be in the group generated by g and in this case it might occur that $\mathsf{ct}_2 = \mathsf{ct}_1^{w_1} \cdot g^{m_1} = \mathsf{ct}_1^{w_2} \cdot g^{m_2}$ for integers w_1, w_2, m_1, m_2 such that $w_1 \not\equiv w_2 \mod \mathrm{ord}(\mathsf{ct}_1)$, $w_1 = w_2 \mod \mathrm{ord}(g)$ and $m_1 \not\equiv m_2 \mod \mathrm{ord}(g)$.

This issue can be solved in one of the following ways.

- **Solution 1: resorting to a trusted setup.** However, if N is generated properly as described in Section 2.1, QR_N is cyclic and g can be set to be a generator of QR_N. In this case (i.e, assuming (N, g) to be generated correctly) the issue can be overcome by having the decryption performed on $(\mathsf{ct}_1^2, \mathsf{ct}_2^2)$ with respect to public key pk^2; indeed, ct_1^2 belongs to QR_N, so in this case it is generated by g (we assume g to be generated correctly as a generator of QR_N).
- **Solution 2: modifying the encryption scheme or requiring interaction with the authority during the casting phase.** An alternative that does not require introducing trusted parameters would be for the encryptor to provide another ciphertext C_2 (beyond $C_1 \triangleq (\mathsf{ct}_1, \mathsf{ct}_2)$) encrypting the randomness r used in $\mathsf{ct}_1 \triangleq g^r$. In this case, the authority can recover r from C_2 and uses r to compute the proof of the fact that ct_1 belongs to the subgroup generated by g. The encryption scheme becomes weaker as the overall ciphertext consists of the pair (C_1, C_2) rather than just C_1. We conjecture the resulting cryptosystem to be IND-CPA secure. Such change in the encryption scheme would make our analysis slightly more complicated.

However, a variation of the previous solution is to require that at time of casting the ballot, the voter interacts with the authority in the following way. The voter sends to the authority the so modified ciphertext that consists of two ciphertexts $C_1 \stackrel{\triangle}{=} (\mathsf{ct}_1, \mathsf{ct}_2)$ and C_2 that encrypts the randomness r used in ct_1). The authority recovers r from C_2 and can check in a perfect way if the voter is cheating encrypting an invalid randomness in C_2. The authority provides a proof π_{ct_1}, using NISG, that ct_1 is in the subgroup generated by g. The proof π_{ct_1} is posted on the bulletin board, but C_2 is not posted on the bulletin board. Notice that in this way, the adversary attacking the privacy of the e-voting system does not observe the encryption C_2 of the randomness but only π_{ct_1} as part of the proof of correct decryption. See also Remark 11.
- **Solution 3: via $\mathcal{NP}$ reductions.** We can use our NI for $\mathcal{NP}$ of Section 1.4.4 to prove ct_1 to belong to the subgroup generated by g. Observe that this cannot be directly done using NIDDH since the prover of NIDDH needs the factorization of the modulus that the encryptor does not have. Instead, our NI for $\mathcal{NP}$ does not suffer this limitation as the prover for the $\mathcal{NP}$ system generates the modulus and all the group elements in its output using knowledge of the corresponding factorization and discrete logs.

Therefore, for the application to e-voting we have either to assume the parameters (N, g) to be generated in a trusted way or we have to use our proof system for $\mathcal{NP}$ of Section 1.4.4 to prove ct_1 to belong to the subgroup generated by g or we have to use the above solution 2 to allow the authority to recover the randomness. We suggest solution 2 as it is efficient, and it just requires a minor change in the flow of the protocol.

Notice that if we use one the above solutions, no unbounded authority may have a non-zero probability of subverting the election result when using our proofs. In contrast, if an authority were aware of trapdoor in a hash function used to instantiate known RO-based NIZKs, it could easily subvert the result of an election in, e.g., the Helios e-voting system [Adi08]. This comes at the cost of basing the privacy of the e-voting system on stronger oracle-based computational problems.

We also point out that, it was not known any *efficient* proof system, even in the CRS model and with statistical soundness, satisfying a non-trivial notion of privacy for proving that an El Gamal-like ciphertext decrypts correctly; in the bilinear group setting instead it was known how to prove efficiently with perfect soundness correct decryption of ciphertexts (for encryption schemes defined in the bilinear setting) with just WI security.

Applications of proofs of correct decryption to e-voting and how to replace proofs of correct encryption with proofs of correct decryption. In an universally verifiable e-voting, proofs of correct decryption are a necessary component used to compute the tally in a privacy-preserving way and guarantee universal verifiability, that is that every party, not just who took part in the election, may verify the result of the election. Proofs of correct *encryption* are also used to enforce that the encrypted plaintext belong to a valid message space. Our techniques cannot be used to construct efficient (if we do not consider efficiency, we can instead use our proofs for $\mathcal{NP}$ relations of Section 1.4.4) proofs of correct encryption as the prover crucially needs the factorization of the modulus. Notwithstanding, we show that proofs of correct decryption can profitably replace proofs of correct encryption in e-voting under the very basic assumption that for each candidate c there exists at least one ciphertext encrypting a vote for c.

We would like to stress that in the following we are considering a setting with a single authority. The single authority has the secret key so is ever able to obtain the preference of any voter. This is inherent in e-voting. Even in a multi-authority setting, the authorities can ever collude together to decrypt the votes. Extending our results to a multi-authority setting is possible but beyond the scope of the work. Note that, unlike privacy, our results show that verifiability can be instead guaranteed even if the authorities (or multiple authorities) are completely malicious. In our analysis, we are also ignoring issues of malleability, so ours is far from and does not aspire to being a complete e-voting solution. (Preventing malleability attacks can be done, e.g., encrypting in an onion way the voters' ciphertexts along with their corresponding signatures under the public key of another non-malleable cryptosystem. We skip the details.)

For simplicity, consider a referendum (voters should encrypt 0 or 1). We propose the following. The authority selects a random bit b and groups the ciphertexts in two classes Z_0, Z_1 putting in Z_b (resp. Z_{1-b}) the ciphertexts encrypting 0 (resp. 1). The authority proves that each pair of ciphertexts c and d in the same class encrypt the same bit by showing that the product of c and d^{-1} (where here we mean the usual operations between El Gamal ciphertexts; cf. Def. 27) decrypts to 0. Notice that this proves that each ciphertext in the same class decrypts to the same value. This fact, combined with the hypothesis that there is at least one ciphertext encrypting 0 and one ciphertext encrypting 1, implies that all ciphertexts in Z_0 encrypt a bit b and all ciphertexts in Z_1 encrypt $1 - b$.

Additionally, for each ciphertext that does not encrypt either 0 or 1, the authority provides a corresponding proof of the fact that the ciphertext decrypts to an invalid plaintext. In this way, the authority is able to prove that it tallies all and only the ciphertexts encrypting plaintexts that are 0 or 1. This can be extended to a larger space of voting options with the proof size growing linearly in the number of options.

Our El Gamal encryption scheme over the group of quadratic residues modulo N and our proofs of correct decryption for it can be used to construct an e-voting scheme satisfying an indistinguishability-based security notion stating that no PPT adversary can win in the following game with non-negligible advantage. The adversary is given the public key of the e-voting system and selects two tuples of n votes $\boldsymbol{v}_0, \boldsymbol{v}_1$ such that $\sum_{i\in[n]} v_{0,i} = \sum_{i\in[n]} v_{1,i}$. A challenge bit b is chosen at random and the adversary is given n ciphertexts $\mathsf{ct}_i, i \in [n]$ encrypting resp. $v_{b,i}$ along with the tally v $(= \sum_{i\in[n]} v_{b,i})$ and a proof of correct computation of the tally. The goal of the adversary is to guess the bit b. The notion can be extended to allow the adversary to choose ill-formed votes. See also Section 1.3.8 for more discussion about the e-voting application and on how to use our proofs to argue security.

1.3 Harmless Zero-Knowledge Proof of Knowledge

Let us recall a proof of the ZK property for the (interactive) Chaum-Pedersen's proof system. The simulator chooses random values $e, z \in \mathbb{Z}_p$ and sets $a \stackrel{\triangle}{=} g^z \cdot u^{-e}, b \stackrel{\triangle}{=} h^z \cdot v^{-e}$. It is easy to see that the transcript of the simulator is distributed identically to the output of the prover, thus the proof system satisfies *perfect* ZK.

In our proof system, the simulator could likewise generate a, b but cannot compute $r^{-1} \mod \phi(N)$ because it does not know r and $\phi(N)$. Any approach to design an efficient simulator is doomed to fail because a ZK proof system for a non-trivial language cannot be perfectly sound. The reason is that if the proof system satisfied perfect soundness, the simulator might be used to decide the language, a contradiction. Moreover, no ZK proof, even with statistical soundness, can be completely non-interactive (without trusted parameters); see Theorem 2.

1.3.1 Harmless ZK According to the ZK paradigm, a proof carries no additional information (i.e., it is zero-knowledge) if whatever you can compute after seeing the proof, you could compute by yourself by means of a simulator. The power of the simulator has to be restricted. Indeed, if the simulator were allowed to have unbounded time, proofs that leak the witness would be declared ZK just because there exists a simulator that can simulate the proof in unbounded time. The obvious restriction is to limit the simulator to run in polynomial-time. We contend that this definition can be generalized to achieve more properties and enable more applications while still sticking to the the ZK paradigm.

Let us analyze the ZK paradigm in more detail. Suppose there exists a language L that is hard to decide for adversaries of time $\mathcal{O}(n^2)$ but easy for adversaries of time $\omega(n^2)$. Then, a proof leaking part or all of the witness might be declared ZK only because there might exist a simulator running in time $\mathcal{O}(n^3)$. This example suggests that simulators running in arbitrary polynomial-time should not be allowed and that the running-time of the simulator should *depend* on the hardness of the language.

One can abstract this line of reasoning: if the language is not decidable by adversaries using a given set S of "resources“, the class of admissible simulators should include all algorithms having access to S. The resources comprise the time of execution of the algorithm but other resources can be considered as well. Indeed, if the language is not decidable by PPT adversaries interacting with some oracle O, the simulator should be allowed to both run in PPT and have oracle access to O. The oracle can be seen as an external entity handing some auxiliary information to the parties in the system.

This leads to our notion of *harmless ZK* (HZK).[6] HZK is a generalization of the traditional ZK formulation in that it allows the simulator to have access

[6] The name harmless zero-knowledge was suggested to us by Geoffroy Couteau.

to an oracle relative to which the language is still hard to decide. We stress that we do *not* allow the simulator to program the oracle. We will denote an algorithm/simulator with access to an oracle O an O-*aided* algorithm/simulator. Considering oracle machines in the analysis of the complexity of computational problems is as old as computer science and dates back to Turing himself. In Section 1.5 we compare our use of oracles to related notions of ZK and secure computation in general.

Our main proof system NIDDH is for the language of DH tuples (over some group of hidden order) and, conjecturing this language to be hard to decide even for adversaries given access to an oracle O (to be defined later), we will show that the proof system is HZK. To avoid trivial attacks, the adversary has to be restricted to not query the oracle on inputs not belonging to the language (see Remark 1 for a careful discussion on this point).

As we will show later, ZK essentially implies that the probability for a PPT adversary of computing a function of the witness given as input a statement for some language L and a ZK proof for it is bounded by the probability that any PPT adversary can compute the same function of the witness given only the statement. The latter probability is negligible if L is hard to decide for PPT adversaries. HZK makes this property more fine-grained by quantifying over PPT adversaries that attempt to compute a function of the witness given the statement but with additional access to some oracle O. Such probability may still be negligible for adversaries with oracle access to O.

Another way of comparing HZK to the standard ZK formulation is to look at an application in which a ZK proof system is employed. ZK proofs are used to enforce correctness in various applications like e-voting while preserving privacy. The privacy of an e-voting system that uses a ZK proof system can be based, e.g., on the (decisional) DH assumption. The running time of the simulator implicitly affects the assumption: if the simulator runs in time $\Theta(m)$, we have to assume DH to hold against adversaries running in time $\Omega(m)$. If the latter were not true, a simulator of time $\Theta(m)$ could not be used in a reduction to the assumption. This, again, is to reiterate the dependency of the resources of the simulator on the language for which the proof system is designed. Analogously, to make use of a simulator with access to an oracle O in a security reduction, the computational problem to which the security is reduced has to hold with respect to adversaries with access to O.

1.3.2 HZK of our main proof system To prove NIDDH to be HZK, we provide a simulator with access to an oracle relative to which we conjecture the language of DH tuples to be hard (under the constraint that the adversary cannot query the oracle on invalid tuples). The simulator needs to invoke the following oracle DHInvO. The oracle DHInvO takes as input a tuple (N, g, h, u, v), checks whether $u = g^w$ and $v = h^w$ for some $w \in \mathbb{Z}^\star_{\phi(N)}$; if such value does not exist, it outputs error and outputs $(g^r, h^r, r' \triangleq r^{-1} \mod \phi(N), z' \triangleq (r+w)^{-1} \mod \phi(N))$ for a random $r \in \mathbb{Z}^\star_{\phi(N)}$, otherwise. The oracle has to compute the randomness so as to guarantee that the inverses modulo $\phi(N)$ exist and that r'

and z' are prime numbers. Cf. Def. 30 for more precise details. Our NI NIDDH can be proven O-HZK with respect to this oracle; see Section 3 for the details. We conjecture the language L of DH tuples (over our group of hidden order) to be hard to decide with respect to DHInvO and thus our simulator belongs to the class of legal oracles for a HZK proof system for L (cf. Def. 5). Precisely, we need to restrict the adversary to not query the oracle on invalid statements (i.e., non-DH tuples). See also Remark 1. Similar considerations hold for NISG that is DHInvO-HZK with respect to a simulator with access to the same oracle DHInvO.

Languages vs relations. Having a proof system for a $\mathcal{NP}$ language L means having a proof system for a polynomial-time relation $\mathcal{R}$ such that $x \in L$ iff there exists w such $(x, w) \in \mathcal{R}$; in this case we say that $\mathcal{R}$ is a relation over L. However, for each $\mathcal{NP}$ language L there are *different* relations over L and a proof system for a relation $\mathcal{R}$ over L does not necessarily imply a proof system for any another relation $\mathcal{R}'$ over L.

Indeed, a subtle point is that in our proof system for DH tuples (as well as the ones for correctness of a shuffle and polynomial statements) the prover cannot be run on input just the statement and the natural witness w (i.e., the exponent for the DH tuple) but additionally needs the factorization of the modulus N. Formally, our proof system is for the relation $\mathcal{R}(x, (w, [p_i, m_i]_{i=1}^{l}))$ whose witness also includes the factorization of the modulus N and checks if the tuple $x \stackrel{\triangle}{=} (N, g, h, u, v)$ is a DH tuple with exponent w, with the group operation being the multiplication modulo N, and $N = \prod_{i=1}^{l} p_i^{m_i}$. Notice that $\mathcal{R}$ is still a relation over the language of valid DH tuples but is different from the "natural" relation whose witness consists of just the exponent (and nothing else). This is reflected in applications: NIDDH can be used to provide proofs of correct *decryption* but not proofs of correct encryption because our prover needs to be run with an input that includes the factorization that is not given to an encryptor.

Notwithstanding, we will see that our techniques can be used to construct an e-voting scheme. Moreover, our proof system for Boolean circuit satisfiability of Section 1.4.4 does not share this limitation (its prover does not need any trapdoor) but this comes at the cost of trading perfect simulatability for computational simulatability.

1.3.3 Harmless proof of knowledge

Traditional PoK is impossible for NI systems. We extend the above concepts to the proof of knowledge property that is challenging in a completely non-interactive setting (no trusted parameters, no RO...).

Observe that the traditional way of defining the proof of knowledge property via extractability is condemned to fail for completely non-interactive proof (or even argument) systems. In fact, suppose towards a contradiction that there exist a NI proof of knowledge system (not based on any trust assumption). Then, there exists an extractor that, given oracle access to the prover, can extract a witness from accepted proofs. But the oracle access to the prover just enables the extractor to see proofs, in particular the only form of rewinding an extractor can carry out is to see multiple proofs for the same statement computed with the same random string (i.e., to see the same *identical* proof multiple times). Clearly, any attack an extractor could perform with oracle access to the prover could be performed by an algorithm without oracle access to the prover. So, the existence of the extractor would imply that the existence of an algorithm with the same computing power that can extract a witness just from the statement.

Motivating and defining HPoK. First, we introduce the notion of a hard relation (cf. Def 13). A hard relation is one that is coupled with a distribution $\mathcal{D}$ over pairs in the relation and is such that no PPT adversary, with possibly access to some oracle, on input an instance x sampled from $\mathcal{D}$, can output (with non-negligible probability) the witness w sampled by $\mathcal{D}$. In the case of the relation of valid DH tuples, we conjecture the hardness of the relation of valid DH tuples associated with the following distribution $\mathcal{D}$: $\mathcal{D}$ outputs pairs (x, w), with x being an uniformly distributed DH tuple (over $\mathbb{Z}_N^\star$, for a Blum integer N) and w being the corresponding witness. For the rest of this discussion, we sometimes omit the dependency on the distribution.

Consider now the following motivating protocol. Parties A and B interact in the following way. A chooses a random DH tuple x in the group of the quadratic residues modulo N with knowledge of its witness w and sends x to B. Party B has access to an oracle $W(\cdot)$ that, given as input x, outputs w. Party B sends a proof π for the validity of x to A. If the proof is accepted, A sends back to B the witness w and B outputs it. The security property P we require is that a malicious $B^\star$ that does not invoke the oracle W should not be able to output a valid witness w for the instance x chosen by A.

We can reduce the security of P to the following assumption: a PPT adversary cannot compute a witness for a randomly chosen DH tuple (over the group of quadratic residues modulo N) even if it has access to a factorization oracle. The reduction algorithm runs B on input the random tuple x, gets from B the accepted proof π and from it, using the factorization oracle, can get a witness to x. The factorization oracle is likely a legitimate resource since the relation of valid DH tuples over the group of the quadratic residues modulo N is probably hard even for adversaries with access to the factorization oracle.

Our definition of harmless proof of knowledge (HPoK) (cf. Def. 16) for a hard relation R postulates that there exists a PPT extractor algorithm Ext with access to an oracle O relative to which R is hard such that Ext, on input any x and any accepted proof for x, can extract a witness to x with probability 1. A NI satisfying HPoK with respect to an extractor with access to an oracle O is said to be *O-extractable*.

As mentioned above, our NI NIDDH has also perfect extraction according to the standard definition [GOS12] when the modulus N is set to a common parameter (that does not have to be trusted). We call HZKPoK a proof system that is both HZK and HPoK.

HPoK does not contradict HZK. It is worth observing why HPoK does not contradict HZK. Consider the concrete example of NIDDH. Intuitively, one could think at an incompatibility of the two properties as it seems that there exists an efficient oracle adversary that can extract a witness to a randomly chosen DH tuple x as follows (and we conjectured such an adversary to not exist). The adversary could generate a proof for x using the DHInvO-aided simulator and then run the O-aided extractor on the proof to get a witness to x. The reasoning is mistaken as for the extraction the adversary needs to invoke an oracle O *different* from the oracle DHInvO for the simulation and it might not be longer hard, for adversaries with access to *both* DHInvO and O, to extract a witness to a randomly chosen instance. In the case of our HZKPoK for DH tuples, the oracle associated with the simulator used in combination with the oracle associated with the extractor allows indeed to compute a witness to a randomly chosen DH tuple.

Precisely, it is true that the existence of an O-HPoK proof is in contradiction with the existence of an O_2-HZK when the two oracles $O(\cdot)$ and $O_2(\cdot)$ are identical but it is not true in general.

HPoK implies hardness of obliviously computing accepted proofs. The immediate corollary of harmless proof of knowledge is that no efficient algorithm can output with non-negligible probability an accepted proof for a randomly chosen statement x (received from a challenger) for a hard relation. That is, an attacker, that has possibly observed other accepting proofs (that are encoded in its algorithm as a non-uniform advice), is unable to produce another accepted proof for a statement chosen according to some hard distribution, except with negligible probability. The formal statement is in Cor. 3. Indeed, if there existed such an algorithm, it could be used by the extractor with oracle access to O to break the hardness of the relation, contradicting the hypothesis that no efficient algorithm even with access to O can break the hardness of the relation.

As an example, an adversary against an e-voting system that observes a ciphertext encrypting a voter preference along with a proof is not able to produce a *different* accepted proof. Therefore, HPoK seems useful to prevent an attacker against an e-voting system to submit ballots encrypting the preferences of other voters, that is to prevent a so called replay attack [CS13].

1.3.4 O-HZK $\iff$ O-function hiding $\rightarrow$ witness hiding

HZK implies WH. Our proofs are additionally 1-message harmless witness hiding proofs, under some computational assumptions. Harmless witness hiding (HWH) is a natural *strengthening* of witness hiding [FS90]. Witness hiding (WH) requires that no efficient adversary (playing the role of the verifier) can extract a witness with non-negligible probability after interacting with the prover on a randomly chosen instance for a hard relation. O-HWH requires the same property to hold but quantifying over adversaries with access to O; see Def. 21. "WH is a natural security requirement and can replace ZK in many cryptographic protocols" [FS90].

Observe that WH is implied by WI with respect to relations with "computationally independent witnesses"; see [FS90]. The WI property of our NI proofs will be analyzed in Section 1.3.9. All NI proofs we construct in this paper, except the NI proof for Boolean circuit satisfiability of Section 1.4.4, are for relations that do not respect the computational independent witnesses property. So, it is a meaningful question to ask whether our NI proofs are WH, and more generally O-HWH.

If DHInvO is the oracle associated with the simulator of our NI NIDDH, then NIDDH is DHInvO-HWH for the hard relation of DH tuples. This holds under the assumption that no PPT adversary, with access to DHInvO, can extract a witness from a randomly selected DH tuple over QR_N, for a Blum integer N. The previous hardness assumption is formally stated as Assumption 4. Towards a contradiction, suppose there exist an adversary $\mathcal{A}$, with oracle access to DHInvO, that can extract with probability p a witness from a proof for a randomly selected DH tuple over this group. Then there exists a DHInvO-aided algorithm $\mathcal{B}$ that receives as input a randomly selected instance x, computes a proof π using the DHInvO-aided simulator, and returns the output $\mathcal{A}$ on (x, π). By definition of $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{B}$ outputs a witness to x with probability p, contradicting the hardness of Assumption 4. See Lemma 5 for more generality and details. (A version of Assumption 4 in which the adversary is restricted to query the oracle only a certain bounded number of times is also equivalent to the DHInvO-HWH of NIDDH (that is, the reduction goes both ways.)

In Appendix A we analyze Assumption 4 against "generic attacks" that we therein define; in particular we prove that in a generic model Assumption 4 is equivalent to an assumption that is identical to Assumption 4 except that the goal of the attacker is to factorize the modulus rather than also finding the exponent w.

Notice that, for any oracle $O(\cdot)$, O-HWH is *stronger* than WH and indeed, under Assumption 4, NIDDH is WH (cf. Cor. 17). It is worth observing why this fact does not contradict the impossibility results of [HRS09,Pas11] of the existence of black-box reductions of "standard assumptions" to WH. First of all, the aforementioned impossibility results apply to distributions that assign non-zero probability to instances with unique witness. However, the impossibility results extend to protocols whose goal is to hide some specific function g of the witness that is uniquely determined in the sense that for any two witnesses w_1, w_2 to a statement x, $g(w_1) = g(w_2)$; the standard WH property corresponds to the goal of hiding the identity function. $\mathcal{R}_{\mathsf{DDH}}$ (cf. Def. 28) is formulated as a multiple witness relation but it is trivial to see that our result on the WH of NIDDH extends to the property of hiding some uniquely determined function, see Remark 10. In the following, we skip this detail and for simplicity consider standard WH (that is, hiding the identity function). Furthermore, the standard assumptions in [HRS09] can be naturally extended to the case of assumptions against oracle-aided adversaries but even in this setting the black-box impossibility results of [HRS09,Pas11] would break down (see next).

The core of the previously cited black-box impossibility results is that if a reduction R using an adversary Adv, guaranteed to break the WH of a given proof system (in our case NIDDH), breaks a computational problem P (in our case Adv is DHInvO-aided and P is Assumption 4), then the reduction can be rewound to extract a witness. This would imply the existence of an adversary $\mathcal{B}$ that breaks P, in our case would imply the existence a DHInvO-aided adversary that breaks Assumption 4. The previous reasoning holds only in the case of proof systems in which the witness can be extracted via rewinding. In the case of our proof NIDDH, an extractor is guaranteed to extract a witness only with access to the factoring oracle. So, for the previous reasoning to hold, we would need to give the reduction access to the factoring oracle but then the black-box impossibility results would imply the existence of an adversary $\mathcal{B}$ breaking Assumption 4 with access to *both* DHInvO and the factoring oracle. This does not lead to a contradiction as Assumption 4 is *not* hard against adversaries with access to both DHInvO and the factoring oracle. Similar considerations hold for our proof for $\mathcal{NP}$ relations of Section 1.4.4 that can be likewise shown to be WH under some computational assumption (or directly conjectured to be so) but we omit the details.

Deng *et al.* [DSYC18] observed that the aforementioned black-box impossibility results hold only for reductions that invoke the adversarial verifier on input a sample from the distribution and showed as to bypass the lower bound by means of reductions that invoke the adversarial verifier on instances from indistinguishable distributions with multiple witnesses. Our NI proofs are WH with respect to a distribution D via reductions that invoke the adversary on instances sampled from D. So, in our case it is not the restriction on the reduction the key to bypass the impossibility results.

Note that the assumption "the NI proof NIDDH is WH" (cf. Def. 22 and Assumption 4) is a falsifiable assumption according to the classification of Gentry and Wichs [GW11], which is a more liberal classification that Naor's [Nao03]. Indeed, the assumption of the WH of NIDDH can be stated by means of the following game (skipping minor details) between an efficient interactive challenger and an adversary. The challenger chooses N with knowledge of the corresponding factorization, a random DH tuple X over QR_N, computes (by means of the factorization) a NIDDH's proof π of the fact that X is DH and runs the adversary on input (N, X, π). The adversary outputs an alleged witness w and the challenger outputs "win" to indicate that the adversary broke the assumption iff w is a witness to X (this fact can be efficiently checked by the challenger).

Finally, we mention that Bellare and Palacio [BP02] prove the security of the Schnorr identification protocol, that is a 3-round proof system, under the hardness of the one-more discrete logarithm problem that is similar in nature to our oracle-based assumptions.

An alternative formulation of HZK. An alternative definition of privacy for NI systems, that we call O-function (or feature) hiding (O-FH) (cf. Def. 24) that is implied by O-HZK and we show to be equivalent to HZK in the case of single witness relations is the following.[7] Assume L to be an $\mathcal{NP}$ language that is worst-case hard to decide relative to O. For any possibly randomized function f, for any PPT algorithm Adv, for any pair $(x, w) \in \mathcal{R}_L$, let $P_{x,f,\mathsf{Adv}}$ the distance between the random variable $\mathsf{Adv}(x, \pi)$ and $f(x)$, where π is a proof for x computed by running the prover on input (x, w) and an uniformly distributed random string, and Adv and f are executed/evaluated on uniformly random strings. Then, for any randomized function f, for any PPT algorithm Adv, there exists a PPT algorithm Adv' with oracle access to O such that, for any $(x, w) \in \mathcal{R}_L$, the distance between $\mathsf{Adv}'^{O(\cdot)}(x)$ and $f(x)$ is equal to $P_{x,f,\mathsf{Adv}}$.

Simplifying, the definition essentially says: for a given function, if no O-aided PPT adversary, on input $x \in L$, can compute the function of x with probability $\geq p$, then no PPT adversary (without access to O), given x and a proof for $x \in L$, can compute the function of x with probability $\geq p$. That is, whatever you can compute from the proof, you could compute without the proof with the same probability using "non-trivial" resources that do not help decide the language.

The notion of O-FH shares similarities with the original notion of semantic security for public key encryption of Goldwasser and Micali [GM84]. In the modern treatment of cryptography, semantic security is usually presented according to the simulation paradigm whereas the original pioneering work of Goldwasser and Micali adopted a function-based notion similar to the previous one. Moreover, the original seminal work of Goldwasser, Micali and Rackoff (GMR) first appeared in ACM STOC '85 [GMR85] was slightly syntactically different from what later appeared in the journal version of the same authors [GMR89]. To our knowledge, the first formal definition of ZK expliciting the role of the simulator was in Goldwasser, Micali and Wigderson (GMW) [GMW91], though the definition of ZK via approximability of random variables in the journal version of GMR [GMR89] implicitly defines a simulator and is equivalent to the simulation-based one of GMW.

An O-HZK proof system satisfies O-FH. Indeed, for any possibly randomized function f, for any PPT algorithm Adv, for any pair $(x, w) \in \mathcal{R}_L$, consider the following algorithm Adv' with access to O. The algorithm Adv' uses the simulator Sim with oracle access to O guaranteed by the definition of HZK to simulate a proof π that is identically distributed to a real proof (i.e., computed by the prover) and runs Adv on (x, π). By hypothesis the probability that Adv' outputs $f(x, w)$ equals $P_{x,f,\mathsf{Adv}}$.

The reverse also holds for a single witness relation (O-FH implies O-HZK). Indeed, consider the randomized function f that implements the prover algorithm that, on input x, outputs a proof for (x, w) computed with the NI system, where w is the unique witness to x. Consider an adversary Adv that, on input

[7] We would like to remark that, despite the name, our definition of O-FH is conceptually different from what is usually denoted as "feature hiding" as generalization of witness hiding [Pas06a]. Ours is a worst-case, non-distributional notion whereas feature hiding meant as generalization of witness hiding assumes inputs chosen from a distribution.

(x, π), just outputs π. The previous definition guarantees the existence of an O-aided algorithm Adv' that, on input x, is identical to the distribution $f(x)$, that is a proof for (x, w), except with statistical distance $P_{x,f,\mathsf{Adv}}$. By definition of Adv, the latter statistical distance equals 0. Such an adversary Adv' is thus a simulator with oracle access to O that generates a proof distributed identically to a proof for (x, w), as it was to show. See Lemma 7 for a more detailed proof of the previous implications.

It is also easy to see that ZK implies 1-FH[8] and, in the case of single witness relations, is equivalent to 1-FH, with $1(\cdot)$ being the oracle implementing the identity function. Therefore, as there exists no NI ZK proof for non-$\mathcal{BPP}$ languages, then there exists no NI FH proof for single witness relations associated to non-$\mathcal{BPP}$ languages.

In Remark 7 we discuss whether and why O-HZK and O-FH are equivalent for relations with multiple witnesses.

1.3.5 The impact of the oracle leakage in applications.

As byproduct, HZK implies that the hardness of computing functions of a witness to some statement, given the statement and a proof for it (and with no access to any oracle), is bounded by the hardness of computing functions of the witness (given only the statement) for adversaries with access to the oracle (associated with the simulator). Therefore, the hardness of computing functions of the witness is bounded by the amount of leakage the oracle provides.

Such leakage may be harmful in some applications in which adversaries are given some auxiliary input. For instance, suppose that a proof of membership of some string x in some worst-case hard $\mathcal{NP}$ language L contain the value $f(x, w)$, for some one-way function f and suppose it be hard to decide L given access to an oracle that on input x returns $f(x, w)$.[9] If f is a trapdoor one-way function, the proof might help an adversary with access to the trapdoor to carry out some task it could not perform without having the proof. So, such proof might be risky for applications in which potential attackers do have access to the trapdoor but might be suitable in other applications. Different oracles give different leakage and may be harmless or harmful depending on the application. In the case of our main HZK proof for DH tuples, for example, the proof cannot be composed with sub-protocols in which the factorization of the modulus is made public.

As a general example, consider the following 3-party protocol between PPT parties $\mathcal{P}$, $\mathcal{V}$ and an unbounded party Q. $\mathcal{P}$ sends to $\mathcal{V}$ a HZK proof π for the membership of x in some worst-case hard language L. Suppose that the proof leaks the value $O(x)$. The verifier can send to Q some pair (x, y) and if $y = O(x)$ the party O sends back to $\mathcal{V}$ a witness w for x if any, or $\perp$ otherwise. It is clear that this protocol is not "zero-knowledge" since the information leaked by the HZK proof allows $\mathcal{V}$ to get non-trivial knowledge from Q. Hence, in this specific

[8] Since there is no NI ZK proof, to give more sense to this implication, we should think more generally about generalizations of the definitions of O-HZK and O-FH for interactive proof systems.

[9] For simplicity, we are glossing over issues of non-uniformity.

protocol the HZK proof turns out to be indeed harmful. It is easy to design other counter-examples in which the leakage given by the oracle is deleterious. However, a proof should be designed to provide "harmless" leakage in natural and practical applications; see Section 1.3.8.

It is interesting noticing why the previous "attacks" do not contradict O-FH. Fix a given function and consider for simplicity an $\mathcal{NP}$ language L associated with a single witness relation. If a PPT adversary, with access to O and on input $x \in L$, can compute the function of the witness to x with probability $> p$, then we *cannot* conclude that no PPT adversary (without access to O), given x and a proof for $x \in L$, can compute the function of the witness with probability $> p$. Indeed, in the case of the oracle O associated to our NI NIDDH, an adversary that has embedded the factorization of a modulus N (that corresponds in the real world to receiving the factorization from a colluding party) can compute the witness of a DH tuple over QR_N with just one invocation to the oracle.

Contrast HZK with ZK that is equivalent to 1-FH. The following considerations apply more generally to interactive proof systems. On the one hand, it may be that there is an PPT oracle adversary that can use an oracle O to compute the function of the witness with probability $> p$, but no such non-oracle adversary exists. So, there may exist an O-FH (and thus O-HZK) proof system that is not 1-FH (and thus not ZK). On the other hand, not considering oracles (or equivalently restricting the oracle to be $1(\cdot)$) in the analysis of an proof system, we are potentially excluding "attacks" that may be instead harmless in a specific application. Viceversa, considering an oracle O in the analysis of an proof system, we are implicitly excluding as harmful all kind of attacks in which the adversary may gain knowledge using the oracle. For instance, if a particular application like e-voting assumes an adversary to not have access to the factorization, then a DHInvO-HZH proof system NIDDH is harmless in that particular application.

More in detail, fix an oracle O and consider a NI proof system for a single witness (this is for simplicity) $\mathcal{NP}$ relation. Let (f, Adv) be a pair consisting of a randomized function and a PPT (non-oracle) adversary (Adv is *not* given access to O or any other oracle). We call the pair (f, Adv) *bad* if there exists no PPT (non-oracle) adversary Adv' such that, for any $(x, w) \in \mathcal{R}_L$, the distance between $\mathsf{Adv}'(x)$ and $f(x, w; r)$ is equal to $P_{x,f,\mathsf{Adv}}$ (cf. Def. of O-FH) but there exists such an oracle adversary $\mathsf{Adv}_2'^{O(\cdot)}$ with access to O. In other words (simplifying), the pair is bad if (1) Adv, with input a proof for a statement x, can compute $f(x, w)$ with some probability $\geq p$ and (2) no PPT adversary can solve the same task without the proof and (3) instead a PPT oracle adversary with access to O (and without the proof) can. So, an oracle O induces a set of bad pairs. If such bad pairs are not harmful in the application in which the proof has to be utilized, then the proof is harmless for that particular application. For instance, the pair consisting of an adversary that has embedded the factorization and the identity function (meaning that the adversary can compute the entire witness) is bad but this may be not harmful in applications, like in e-voting, in which the verifier is not given the secret key. When analyzing the security via ZK (that is, 1-FH), the set of bad pairs (relative to the trivial oracle $1(\cdot)$) is ever empty but this comes at the cost of being unable to demonstrate useful security properties of the proof system in a particular application.

In other words, ZK is a powerful notion of privacy but protects against *any* possible leakage, even the ones that might not be harmful to the applications in which ZK protocols are usually employed. O-HZK instead exploits the fact that in most real-world protocols the corresponding ideal world harmlessly grants to the adversary an O-leakage. Consider the relation between ZK and WH, and more generally simulatability and WH. The existence of a simulator implies WH, under the assumption that no adversary, *as powerful as* the simulator, can compute a witness from a randomly selected instance for a hard relation. If we restrict the simulator to be PPT, we are not taking advantage that the fact that it is likely hard even for adversaries with oracle access to DHInvO to compute a witness from a randomly selected instance for a hard relation. So, in some case we might not able to reduce simulatability to WH. So, granting the simulator more resources may allow, as our results demonstrate, to deduce the WH of a specific proof system, fact that might not be provable without this leveraging of the simulator power.

Composition of HZK proofs. HZK proofs can raise issues of composition in a protocol I when the oracle associated with the simulator can make one of the hard problems, on which the security of I is based, easy. This is already implicit but less visible in the case of standard ZK proofs.

Consider a possible world (not ruled by our current knowledge on computational complexity) in which a problem P used in cryptographic constructions is provably hard for adversaries of time $\mathcal{O}(n^7)$ but it is provably breakable by an adversary of time $\mathcal{O}(n^{10})$. In this world, it would still make sense to design cryptographic protocols choosing appropriately the security parameter though one would have to carefully take in account the running-time of simulators for ZK proofs. Indeed, a simulator of time $\mathcal{O}(n^{10})$ should not be considered legitimate. Moreover, if a larger protocol were based both on P and on another problem P_2 with stronger hardness requirements (e.g., hard only for adversaries of time $\mathcal{O}(n^4)$), even a simulator of time n^6 could raise compositional issues. This scenario is however unlikely as we currently believe that PPT algorithms can be composed together without having "too much" effects on the computational assumptions used in the design of protocols.

Yet, the running-time of the simulator of a ZK proof may affect the quality of a reduction of some protocol I that uses the ZK proof to some hard problem P on which the security of I is based: more time the simulator needs to carry out the simulation, more time the reduction may need to break P and thus, for a given security parameter and for a given bound on the computational power of adversaries, a stronger hypothesis on the hardness of P may be required. Then, composing a sub-protocol for which a tight reduction is known with a ZK proof might imply a significant worsening of the tightness of the reduction.

Such compositional issues are more evident when using oracles. In particular, it may be easier to check whether an oracle help in breaking some computational problem and hence issues with oracle-aided simulation are made manifest.

In view of the above considerations, we advocate a pragmatic approach. ZK and HZK are tools for proving the security of larger protocols. HZK proofs can own additional properties provably impossible for ZK proofs, like non-interactivity and perfect soundness, but this comes at the cost of stronger "oracle-based" assumptions and potential composability issues with other protocols whose security relies on assumptions that do not hold relative to the oracle associated with the HZK simulator. In this paper we show real-world applications that crisply benefit from replacing ZK with HZK proofs.

Relation to computational ZK or formulations without auxiliary inputs. HZK, even in its perfect HZK formulation, is somehow qualitatively comparable to computational ZK without auxiliary input (cZK) in that the computational leakage gained by an adversary is harmless unless the adversary holds a trapdoor as it is illustrated by the following example.

Given an arbitrary $\mathcal{NP}$ relation $\mathcal{R}$, define the relation $\mathcal{R}'$ so that, for each string pk, $\mathcal{R}'(x||\mathsf{pk}, w)$ holds iff $\mathcal{R}(x, w)$ holds (pk is artificially ignored).

Consider the following ZK proof for $\mathcal{R}'$. The prover, on input $x||\mathsf{pk}$ and w, for a valid El Gamal public key pk, interacts with the verifier using a ZK proof for $\mathcal{R}$ but additionally in the last message sends an encryption of w computed under pk. The verifier ignores the ciphertext and accepts iff the ZK argument for $\mathcal{R}$ is accepting. This protocol is still simulatable (the simulator can set the ciphertext to be an encryption of an arbitrary string of the same length) but the output of the simulator is only computationally indistinguishable under some computational assumption[10] from the output of the honest prover, that is the protocol is cZK. Yet, an adversary having the secret key corresponding to the public key can compute the entire witness. So, similarly to HZK, a cZK protocol does not necessarily guarantee security against adversaries that may gain some trapdoor information.

1.3.6 On trivial and efficient oracles

Trivial oracles. A HZK simulator is coupled with an oracle. The choice of an oracle O introduces qualitatively different assumptions when plugging an O-HZK proof into a concrete application. In the extreme case, for a proof system for a relation $\mathcal{R}_L$, one could set the oracle to be the trivial "prover oracle". The prover oracle gets as input a statement $x \in L$, finds a witness w such that $\mathcal{R}(x, w) = 1$ and returns the output of the prover on input (x, w) (and uniformly chosen random bits). Then the assumption on the hardness of the language would boil down to require L to be hard even for adversaries seeing proofs for statements

[10] For each PPT adversary $\mathcal{D}$, for all sufficiently long valid El Gamal public keys pk, $\mathcal{D}$ has negligible advantage in distinguishing an encryption of 0 under pk from an encryption of 1 under pk.

in L. That, in turn, accounts to say that the protocol is "secure because it is secure".

One can object that the oracle associated with our main proof system is somehow trivial as well. We embrace a pragmatic approach. First of all, computational assumptions are such because of our current limitations on their provability or refusal. So, the real challenge is to reduce the security of protocols to assumptions that are at least enough simple to cryptanalyze. Simulators are a practical tool employed to prove the security of protocols and can be used in a neater way than witness indistinguishability that, not only introduces direct inefficiency in the security proofs due to a complicated and unnatural use of OR statements, but also results in severe limitations (see discussion on verifiable functional encryption in Section 1.3.8).

In Section 1.3.8 we show that our main proof system can be profitably used to prove the security of a concrete non-trivial simplified e-voting protocol and the overall security of such protocol can be reduced, via a reduction that uses the oracle-aided simulator, to Assumption 8 that is well-defined and seemingly cryptanalyzable. We stress that it was not know before how to construct an efficient protocol for the same task satisfying perfect verifiability (i.e., with proofs of correct decryption satisfying perfect soundness) without bilinear groups.

It appeared not obvious to us to come up with non-trivial oracles to design oracle-aided simulators for other known proof systems like the non-interactive ZAPs of Groth *et al.* [GOS12].

Note that the trivial proof system for a single witness DH relation in which the prover outputs the witness in the clear satisfies WI but does not satisfy (under some reasonable computational assumption) DHInvO-HZK with respect to the oracle DHInvO previously defined in Section 1.3.2. Indeed, the DHInvO-aided simulator could be used to extract a witness from a DH tuple (given oracle access to DHInvO) and this seems a hard problem.

Oracles with auxiliary input. Our oracle DHInvO for our proof system for DH tuples cannot be implemented in polynomial-time, not even whether the oracle is given the trapdoor as hint. It is a natural question to ask whether there may exist a one-message HZK proof system associated with an oracle that runs in polynomial-time given a trapdoor as auxiliary input. We speculate this situation to be unlikely. The intuition is that if this were the case, the HZK proof system might be converted into a traditional ZK proof system by somehow "obfuscating" the oracle program (with the trapdoor embedded in its representation).

The question might be related to the problem of removing the oracle and achieving super-polynomial (but sub-exponential) simulation [Pas03b] via complexity leveraging arguments [CGGM00]. Along this direction, we mention that our HZK proof for Boolean circuit satisfiability is computational HZK (cf. Def. 6). Our proof for $\mathcal{NP}$ can be tweaked, using known techniques, to achieve one-message arguments with uniform soundness and quasi-polynomial time simulation or alternatively two message-arguments with standard soundness. In Section 1.4.4 we briefly discuss this point.

1.3.7 Undeniability of our proofs Though our proofs are non-interactive, they are *undeniable* as well. Indeed, you cannot claim to have generated the proof by yourself using the simulator. This is because you would need access to the oracle to run the simulator. In the case of interactive ZK proofs (as well as for NIZK proofs in the CRS and programmable RO model), the simulator can be also used to simulate proofs for false statements and thus having a transcript for a given statement does not represent evidence that you know the veracity of the statement. In contrast, if you have a perfectly sound HZK proof for a given statement, you *can* always check its validity and so you cannot deny that you *could* have learnt the validity of the statement since proofs for false statements cannot be simulated.

1.3.8 Using HZKPoK and applications to e-voting, FE, CCA1 In Section 1.3.4 we demonstrated that, under some computational assumption, a NI HZK proof is witness hiding and as such it inherits the applications of witness hiding protocols. Next, we show further applications of HZK proofs, and in particular we show how to make use of HZK proofs in security reductions.

E-voting. Consider the following e-voting application (see also Section 1.2.3). (We would like to stress that we are in a setting with a single authority. The single authority has the secret key so is ever able to obtain the preference of any voter. This is inherent in e-voting [CMFP+10]. This issue can be leveraged passing to a threshold multi-authority setting that is not analyzed in this work. However, even in a multi-authority setting, the authorities can ever collude together to decrypt the individual ballots.)

The authority adds a proof that the the product ciphertext $\mathsf{ct} \stackrel{\triangle}{=} \mathsf{ct}_1 * \mathsf{ct}_2$ (where "$*$“ has the usual meaning) decrypts to v, with v being v_1+v_2. We would like such a protocol to satisfy the following privacy requirement. The requirement (details in Assumption 7) states that a PPT adversary cannot distinguish whether either ct_1 encrypts 1 and ct_2 encrypts -1 or ct_1 encrypts -1 and ct_2 encrypts 1, given additionally a proof of the fact that the product ciphertext $\mathsf{ct}_1 * \mathsf{ct}_2$ decrypts to 0.

If the proof of correct decryption were ZK, we could reduce the security to the standard Decisional Diffie-Hellman (DDH) assumption, that is the problem of distinguishing a DH tuple from a random tuple. When using a HZK proof, one can naively think to reduce the privacy to a variant of DDH in which the adversary has access to the oracle DHInvO needed by the simulator of our DHInvO-HZK proof. Unfortunately, there is an issue.

Let us first analyze in more detail how the security reduction would work if the proof were ZK. Consider the following hybrid experiments (we do not include all the hybrid experiments necessary to argue the security and we do not take in consideration the CRS model).

- H_0. Hybrid experiment H_0 is identical to the real game except that ct_1 encrypts 1 and ct_2 encrypts -1.

– H_1. Hybrid experiment H_1 is identical to H_0 except that the proof is simulated (instead of being honestly generated).
– H_2. Hybrid experiment H_2 is identical to H_1 except that ct_1 encrypts a random group element (the proof is still simulated like in H_1).

One can construct an adversary against DDH from an adversary that distinguishes H_2 from H_1, that is an El Gamal encryption of 1 from an El Gamal encryption of a random element, given additionally a simulated proof. The point is that in H_2, the proof can be simulated even though the statement given as input to the simulator is *false*: ct_2 encrypts a random plaintext and hence the product ciphertext $\mathsf{ct} \stackrel{\triangle}{=} \mathsf{ct}_1 * \mathsf{ct}_2$ is not a valid statement (it does not decrypt to 0).

A ZK simulator *can* be run on false statements and indeed one can prove that simulated ZK proofs for false statements are always accepted by the verifier. In contrast, a HZK simulator cannot be generally run on false statements. Precisely, we cannot attempt to reduce the privacy to the variant of DDH in which the adversary has access to the oracle DHInvO because, by definition, DHInvO returns an error when the input is not a valid DH tuple (cf. Def. 30); the assumption would be trivially false since the oracle might be used to distinguish a DH tuple from a random one. Therefore, when using HZK proofs in reductions one has to guarantee that the simulator is invoked only on valid statements. This can be done considering the following alternative series of hybrid experiments.

– H_0'**.** Hybrid experiment H_0' is identical to the real game except that ct_1 encrypts 1 and ct_2 encrypts -1.
– H_1'**.** Hybrid experiment H_1' is identical to H_0' except that the proof is simulated (instead of being honestly generated).
– H_2'**.** Hybrid experiment H_2' is identical to H_1' except that ct_1 encrypts -1 and ct_2 encrypts 1 (the proof is still simulated like in H_1).
– H_3'**.** Hybrid experiment H_3' is identical to H_2' except that the proof is honestly computed (instead of being simulated).

Observe that the simulator is always invoked on valid statements. The perfect indistinguishability of H_1' from H_0' (resp. H_3' from H_2') follows from the perfect HZK simulatability. The indistinguishability of H_2' from H_1' follows from Assumption 7 that may be easily seen to be equivalent to the following "simpler" assumption: a PPT adversary cannot distinguish a DH tuple (g, h, u, v) for witness w (i.e., $u = g^w, v = h^w$) from a random tuple, given additionally oracle access to DHInvO and values h', v' such that (g, h', u, v') is another DH tuple for the same witness w. This is stated in Assumption 8 and the reduction of Assumption 8 to Assumption 7 is proven in Lemma 11.

Verifiable Functional Encryption. Badrinarayanan [BGJS16] *et al.* put forth the powerful primitive of verifiable functional encryption that extends functional encryption in that malicious behavior of the central authority and encryptors can be detected. The difficulty in the construction of verifiable functional encryption of Badrinarayanan *et al.* stems from the fact that, in order to not rely on trusted parameters, NIZKs in the CRS model have to avoided and replaced by non-interactive WI proofs. Unfortunately, a complicated and unnatural use of WI limits the security to be selective (the adversary has to announce the challenge before seeing the public key). Essentially, to engineer multiple witnesses, the public key has to contain a commitment that in a hybrid experiment is set to be a commitment to the challenge message. This shows that often the use of WI is not only unnatural and causes an efficiency loss but also suffers inherent limitations: it is not know how to construct, from a fully secure functional encryption scheme and non-interactive WI proofs, a fully secure verifiable functional encryption scheme.

A HZKPoK NI for $\mathcal{NP}$, as the one we construct in Section 1.4.4, makes the construction of a fully secure verifiable functional encryption from an arbitrary fully secure functional encryption scheme straightforward: add to ciphertexts and tokens of a fully secure functional encryption scheme proofs of their correct computation. The perfect verifiability follows from the perfect soundness of the proof and the IND-CPA security for the verifiable functional encryption scheme follows assuming the underlying functional encryption scheme to be IND-CPA secure against adversaries with access to the oracle SimNP used by simulator of our NI for $\mathcal{NP}$ (restricting the adversaries to never query the oracle on inputs returning error).

CCA1-*secure encryption.* A natural testbed to demonstrate the usefulness and usability of our proof systems is to figure out, e.g., whether our NI for $\mathcal{NP}$ of Section 1.4.4 can be used to construct an encryption scheme secure against non-adaptive chosen-ciphertext attacks (CCA1-secure, for short) [NY90]. A CCA1-secure encryption scheme differs from a semantically secure encryption scheme in that an adversary attacking the CCA1 system is given access to a decryption oracle for any ciphertext of its choice under the restriction that the queries can be asked only before seeing the challenge ciphertext.

There are two main approaches to construct CCA1-secure encryption schemes from standard NIZKs: the Naor-Yung double encryption paradigm [NY90] and the "proof of knowledge" (PoK) paradigm of Sahai [Sah99] (see also [DDO$^+$01]). The PoK paradigm uses a semantic secure encryption scheme $\mathcal{E}$ and constructs a CCA1-secure encryption scheme $\mathcal{E}'$ in the following way. The public key of $\mathcal{E}'$ contains the public key of $\mathcal{E}$ and a CRS for a NIZKPoK. A NIZKPoK is a NIZK with the additional property that the CRS can be computed with knowledge of a trapdoor that allows to efficiently extract a witness from any proof that is accepted by the verifier. A ciphertext of $\mathcal{E}'$ consists of a ciphertext of $\mathcal{E}$ along with a NIZKPoK proof that the ciphertext is computed correctly. The decryption algorithm returns error if a proof is not accepted and decrypts using $\mathcal{E}$ otherwise.

An adversary against the CCA1-security of $\mathcal{E}'$ can be converted to an adversary against the semantic security of $\mathcal{E}$ in the following way. Consider an hybrid game that is identical to the CCA1-security game except that the proof is simulated (instead of being computed by the prover algorithm). Consider the following reduction algorithm attacking this hybrid game. The reduction setups the NIZKPoK CRS with knowledge of the corresponding trapdoor. By means of the trapdoor, the reduction can answer all decryption queries of the CCA1 adversary: if the proof is not accepted it returns an error, and uses the trapdoor to extract the witness (that contains the message) and output the message otherwise. The reduction adds a simulated proof π to the $\mathcal{E}$'s challenge ciphertext ct received by the challenger of the semantic security game and returns the output of the adversary against the CCA1-security of $\mathcal{E}'$ on input (ct, π). Because simulated proofs are indistinguishable from real proofs, the previous hybrid game is indistinguishable from the real CCA1 game so it is straightforward to conclude that an adversary breaking the CCA1-security of $\mathcal{E}'$ can be converted in an adversary breaking the semantic security of $\mathcal{E}$.

Observe that, for the reduction to go through, it is crucial to restrict the CCA1 adversary to not make queries after seeing the challenge ciphertext. Without the restriction, an adversary in the previous hybrid game, after seeing a challenge $C = (\mathsf{ct}, \pi)$, might query the decryption oracle on a ciphertext $C' = (\mathsf{ct}', \pi')$ such that π' is equal to the simulated proof π. Moreover, removing the restriction the reduction would need to answer decryption queries even after the challenge ciphertext phase. However, indistinguishability of real proofs from simulated proofs is not guaranteed for adversaries with unlimited access to an extraction oracle, in particular for adversaries that can attempt to extract a witness from a simulated proof. Therefore, the previous hybrid game would be *distinguishable* from the real CCA1 game.

Next, we sketch how the PoK paradigm can be adapted so that our NI for $\mathcal{NP}$ of Section 1.4.4 can be used to construct a CCA1-secure encryption scheme. The issue in employing the PoK paradigm with our NI for $\mathcal{NP}$ is that our NI for $\mathcal{NP}$, as any HZKPoK proof, enjoys the property that there exists an extractor that ever succeeds in extracting a witness even from simulated proofs, not just from real proofs as for standard NIZKPoKs. We overcome this obstacle in the following way. In Section 1.4.4, we will see that a proof computed by our NI for $\mathcal{NP}$ contains a modulus N (the modulus N is part of the proof, not of the statement or witness) and there exists an efficient extractor that, with knowledge of the corresponding factorization of N, is able to extract a witness from any accepted proof for modulus N (i.e., from any proof that contains N in its description). We convert an adversary against the CCA1-security to an adversary against the following variant of the semantic security game. The game is identical to the semantic security game except that the adversary is given access to the factoring oracle and access to the SimNP associated with the simulator of our NI for $\mathcal{NP}$ under the following restrictions: the adversary can invoke the factoring oracle only before seeing the challenge ciphertext and can invoke the SimNP oracle only after seeing the challenge ciphertext. In this case, the factoring oracle can be used to answer the decryption queries and the SimNP oracle to simulate the proof for the challenge ciphertext.

For the construction to work, we need a semantic secure encryption scheme secure even against PPT adversaries with access to the factoring oracle and the SimNP oracle under the previous restrictions. We conjecture our variant of El Gamal (cf. Def. 26) as well as the standard El Gamal encryption scheme over multiplicative groups of prime order or over elliptic curves to satisfy the previous property. Notice that the restrictions on the oracle queries make the conjecture non-trivial: removing the restrictions, an attacker can easily break the game simulating a proof for the challenge ciphertext and extracting a witness for it using the factoring oracle.

In full generality, a CCA1-secure encryption scheme can be constructed from any encryption scheme $\mathcal{E}$ and any NI that is O-simulatable and E-extractable if the following holds: $\mathcal{E}'$ is semantically secure even against adversaries with oracle access to O and E under the restriction that a query to E can be done only before seeing the challenge ciphertexts and a query to O can be done only after seeing the challenge ciphertext.

The advantage of the CCA1-secure encryption scheme constructed from our NI for $\mathcal{NP}$ is in that ill-formed ciphertexts can be publicly filtered out without any error, that is no ill-formed ciphertext can pass the public verification test. In the CCA1-secure encryption scheme constructed using standard NIZKPoK arguments as the one of De Santis *et al.* [DDO+01] or De Santis and Persiano [DP92], there is no way to reject ill-formed ciphertexts if the attached proof is simulated. Moreover, in the De Santis *et al.* NIZKPoK, as in any same-string NIZKPoK, simulated proofs for false statements can be computed even if the CRS is setup correctly.

In the current version of our work, we neglect the treatment of adaptive chosen-ciphertext attacks (CCA2).

1.3.9 WI and *O*-strong-WI It is easy to see that an O-HZK proof is also witness indistinguishable (WI) (cf. Def. 19) [FS90]. Indeed, it is true that any proof system endowed with a simulator of unbounded time that outputs a string indistinguishable from a real proof satisfies WI: observe that the WI property does not depend on the simulator resources (oracle in our case) at all. In Section 1.4.4, we construct a one-message WI perfectly sound proof for $\mathcal{NP}$ relations.

Strong witness indistinguishability (strong-WI) [Gol01,Gol04] requires that for two computationally indistinguishable statement distributions X_1 and X_2, a proof for statement $x_1 \leftarrow X_1$ must be computationally indistinguishable from a proof for statement $x_2 \leftarrow X_2$. The previous application to e-voting makes manifest a close relation to strong-WI. Indeed, consider the following two distributions $X_b, b \in \{0,1\}$. Distribution X_b contains a randomly selected public key pk for our (variant of) El Gamal encryption scheme and a ciphertext ct computed as follows. Compute two ciphertexts $\mathsf{ct}_0, \mathsf{ct}_1$ encrypting resp. b and $1-b$ under pk and set ct to be $\mathsf{ct}_0 * \mathsf{ct}_1$. The statement output by the distribution states that ct is a ciphertext for public key pk that decrypts to 1 and $\mathsf{ct} = \mathsf{ct}_0 * \mathsf{ct}_1$. Assumption 7 (that is in turn equivalent to Assumption 8) may be then seen to be equivalent to conjecturing our main proof system to be strong-WI with respect to these two specific distributions.

More generally, an O-HZK proof system is O-strong-WI, that is strong-WI when quantifying distributions that are computationally indistinguishable even by adversaries with access to O; see Def. 23 and Corollary 6. We are not aware of any work analyzing whether known non-interactive ZAPs are also strong-WI.

1.3.10 Why DHInvO and not other "simpler" oracles?

We emphasized that the choice of the oracle O for an O-HZK proof reflects in the applications in which the proof is used. A natural question is whether a variant of NIDDH can be proven O-HZK with respect to an oracle O different from DHInvO such that the resulting proof system would still have similar applications as the original NIDDH but the assumptions introduced by O would be easier to cryptanalyze, i.e., O would be in some sense "simpler".

We would like to draw attention that care has to be taken when attempting to "simplifying" oracles and proof systems.

Consider for instance the oracle O that, on input a DH tuple over $\mathbb{Z}_N^\star$ for witness $w \in \mathbb{Z}_{\phi(N)}$, outputs (if it exists) $w' \stackrel{\triangle}{=} w^{-1} \mod \phi(N)$. The value w' can be used to check the well-formedness of the tuple in the obvious way. Let us denote by NIDDH' the O-HZK proof whose prover outputs w' and whose verifier checks the well-formedness using w'. The HPoK property of NIDDH' is based on the same oracle (i.e., the factoring oracle FactO) than the HPoK property of NIDDH. Is then NIDDH' useful in the same applications of HPoK as NIDDH does? Observe first that the proof computed by NIDDH' for two different DH tuples for the same witness is identical. Recall that HPoK implies the infeasibility for an efficient attacker of computing accepted proofs for random statements (of which the attacker does not know the corresponding witness). However, in the case of NIDDH' it is not possible to conclude that the latter property implies the hardness of carrying out replay attacks in *general*. Indeed, observing one proof $\pi \stackrel{\triangle}{=} w'$ for a random statement X for witness w allows to forge a proof for another random statement Y for the same witness w: the forged proof is just π itself! That is, NIDDH' is subject to a malleability attack [DDN91]. This opens the question of whether NIDDH be non-malleable and of the relations between our definitions and their non-malleable variants. In the current version of this work, we do not study these problems.

Furthermore, there exist distributions for which DHInvO-strong-WI holds but O-strong-WI does not hold. For example, consider the distributions $X_b, b \in \{0,1\}$ of Section 1.3.9. The ciphertexts in each distribution are encrypted with respect to the same secret key w, hence the oracle O gives the possibility of testing whether, e.g., ciphertext ct_0 decrypts to 1 by just testing whether $\mathsf{ct}_0 \cdot d$ is a DH pair for witness w, where d is any ciphertext encrypting -1.

The lesson is that apparently simpler oracles may give more power to an attacker.

1.4 Extensions

In this section we sketch several extensions of our main proof system for DH tuples. While we provide a comprehensive overview, we sometimes skip technicalities and details, in particular we mostly opt for presenting only the details needed to construct our HZK proof for $\mathcal{NP}$ relations of Section 1.4.4.

1.4.1 Verifiable shuffle

Our techniques can be extended to construct a non-interactive perfectly sound proof of correctness of a shuffle of (our variant of) El Gamal ciphertexts. Our starting point is the verifiable shuffle of Neff [Nef01].

The Iterated Logarithm Multiplication Problem. As in Neff, to construct a verifiable shuffle we first build proofs for the iterated logarithm multiplication problem (ILMP). In the ILMP for parameter k, one wants to prove that two tuples $X = (X_1, \ldots, X_k)$ and $Y = (Y_1, \ldots, Y_k)$ are such that $\prod_{i=1}^{k} \mathbf{dlog}_g X_i = \prod_{i=1}^{k} \mathbf{dlog}_g Y_i$. For simplicity, hereafter we consider tuples of 3 elements, that is we consider the ILMP for $k = 3$. (For our construction of a HZK NI proof for $\mathcal{NP}$ relations of Section 1.4.4, we would actually need to just consider the case $k = 4$, but we believe that the core ideas can be presented more clearly with a slight loss of generality and we limit the presentation to the case $k = 3$.)

In our setting, the tuples are over $\mathbb{Z}_N^\star$, so it might be that for some $i \in [3], \mathbf{dlog}_g X_i$ or $\mathbf{dlog}_g Y_i$ does not exist. Therefore, in the following we will often assume all the group elements in the tuples to belong to the subgroup generated by a public group element g (that, if computed honestly, is a generator of the group of quadratic residues modulo N). This limitation can be removed using our HZK NI proof NISG to prove each group element in the tuples to belong to the subgroup generated by g. Observe that ILMP for $k = 2$ corresponds to the problem of proving that a tuple of 4 group elements is DH, so ILMP can be seen as a generalization of the DH problem.

We assume the prover knows the discrete logs in base g of all the elements X_i's and Y_i's. In our proof system for Boolean circuit satisfiability of Section 1.4.4 the prover does know the discrete logs of all the group elements (because it is the prover to create such elements with knowledge of the corresponding exponents) and so this assumption is not a limitation for that application.

A HZK NI proof for the ILMP. Consider the following non-interactive proof (we will next show that, in order to make it sound, it has to be modified).

The prover sends values:

$$A_1 \stackrel{\triangle}{=} Y_1^{\Theta_1}, A_2 \stackrel{\triangle}{=} X_2^{\Theta_1} \cdot Y_2^{-\Theta_2}, A_3 \stackrel{\triangle}{=} X_3^{\Theta_2},$$

and values r_1, r_2 satisfying the following three equations:

$$Y_1^{r_1} = A_1 \cdot X_1, X_2^{r_1} \cdot Y_2^{-r_2} = A_2, X_3^{r_2} = A_3 \cdot Y_3.$$

Observe that the prover can efficiently do that as it knows the exponents $x_1, x_2, x_3, y_1, y_2, y_3$.

The verifier accepts iff all the last equations are satisfied.

Let us set $\bar{r}_i = r_i - \Theta_i$. Assuming that the values A_1, A_2, A_3 are generated correctly, the first equation implies (1) $\bar{r}_i \cdot y_1 = x_1$, the second implies (2) $\bar{r}_1 \cdot$

$x_2 - \bar{r}_2 \cdot y_2 = 0$ and the third (3) $\bar{r}_2 \cdot x_3 = y_3$. Replacing (1) and (3) in (2) we have $\frac{x_1 \cdot x_2}{y_1} + \frac{y_3 \cdot y_2}{x_3} = 0$ that implies $x_1 x_2 x_3 = y_1 y_2 y_3$.

Therefore, (perfect) soundness would hold if the values A_1, A_2, A_3 were ever computed correctly (if, e.g., $A_3 = X_3^{\Theta'_2}$ for $\Theta'_2 \neq \Theta_2$ the above analysis would break down). For this reason we have to modify the above system to prove correctness of the computation of the values A_i's.

To that purpose, the prover also sends $z_1 = \Theta_1^{-1} \mod \phi(N), z_2 = \Theta_2^{-1} \mod \phi(N)$. It is easy to see that such values can be used to check the validity of the A_i's: the prover first checks that $A_1^{z_2} = Y_1$ then compute $C = A_2 \cdot X_2^{z_1}$, checks that $C^{z_2} = Y_2^{-1}$ and that $A_3^{z_2} = X_3$. However, it might be that the value z_i's are not co-prime with $\phi(N)$ and in such case the check would not be sufficient to guarantee soundness (see Section 1.2). As for our NI proof NIDDH, we have thus to employ the trick of the parallel repetitions (see Section 1.2). In the following, for simplicity we will skip this detail and assume the above version of the proof without parallel repetitions.

Let DHInvO be the oracle associated to the simulator for the NI NIDDH. The DHInvO-HZK can be proved as follows. The DHInvO-aided simulator $\mathsf{SimILMP}$ computes random values $r_1, r_2 \leftarrow \mathbb{Z}_{\phi(N)}$ and computes elements A_1, A_2, A_3 satisfying the equations

$$Y_1^{r_1} = A_1 \cdot X_1, X_2^{r_1} \cdot Y_2^{-r_2} = A_2, X_3^{r_2} = A_3 \cdot Y_3.$$

The distribution of the elements $(r_1, r_2, A_1, A_2, A_3)$ computed by the simulator is identical to the distribution of the same elements in the distribution of real proofs. Finally, the simulator invokes the oracle on (N, Y_1, Y_1, A_1, A_1) to get $z_i = \Theta_1^{-1} \mod \phi(N)$ and invokes the oracle on (N, X_3, X_3, A_3, A_3) to get $z_2 = \Theta_2^{-1} \mod \phi(N)$ and outputs $r_1, r_2, z_1, z_2, A_1, A_2, A_3$. By definition of the oracle, the so computed values z_1, z_2 are distributed identically to the values z_1, z_2 in the distribution of real proofs.

Note that, as mentioned above, to prove that all the group elements in the tuples belong to the subgroup generated by g, we have to invoke NISG that is also DHInvO-HZK, for the same oracle $\mathsf{DHInvO}(\cdot)$. Furthermore, recall that we are skipping the fact that, for the soundness to hold, the proof has to be repeated in parallel a certain number of times (in this case, the previous simulator would have to run the same computation several times).

The previous protocol can be generalized to deal with any tuple of $k > 3$ elements. For simplicity, we do not present the details. We remark that for our main application to proofs of Circuit Satisfiability of Section 1.4.4, it is sufficient to consider proofs for the ILMP for parameter $k = 4$. Hereafter, we assume to have a proof for the ILMP for any $k > 3$.

A HZK interactive proof for proving the correctness of a shuffle. From ILMP we move to a simple n-shuffle problem. Constants $c, d \in \mathbb{Z}^\star_{\phi(N)}$ are known to the prover and commitments $C = g^c$ and $D = g^d$ are published. The prover has to convince the verifier that there is some permutation $p : [n] \to [n]$ such that:

$$Y_i^d = X^c_{p(i)},$$

for $i \in [n]$. We assume the prover to know the discrete logs in base g of all the elements X_i's and Y_i's; in the application to the proof for $\mathcal{NP}$ of Section 1.4.4, this assumption will not constitute a limitation as it will be the prover of such system to create the group elements with knowledge of the corresponding exponents.

We firstly consider an interactive protocol. The protocol proceeds as follows. The verifier sends to the prover a random $t \in \mathbb{Z}_{\phi(N)}$.

Prover and verifier publicly compute $U \triangleq D^t = g^{dt}$ and $W \triangleq C^t = g^{ct}$ and

$$\bar{X} \triangleq (\bar{X}_1, \ldots, \bar{X}_n) \triangleq (X_1/U, \ldots, X_n/U)$$

and

$$\bar{Y} \triangleq (\bar{Y}_1, \ldots, \bar{Y}_n) \triangleq (Y_1/W, \ldots, Y_n/W).$$

Prover sends a ILMP proof for the vectors $(\bar{X}, [C]^n)$ and $(\bar{Y}, [D]^n)$, where $[C]^n$ (resp. $[D]^n$) denotes a list containing the value C (resp. D) repeated n times. (Here, we are implicitly assuming to have available a NI system for the generalization of the ILMP for parameter $k = 2n$. For our proof for $\mathcal{NP}$ relations of Section 1.4.4 we will need only a proof of correct shuffle for parameter $n = 2$ and thus in turn a proof for the ILMP for parameter $k = 4$.)

By the perfect soundness of the ILMP proof, if the proof is accepted then it holds that:

$$c^n \cdot \prod_{i=1}^{n}(x_i - dt) = d^n \cdot \prod_{i=1}^{n}(y_i - ct)$$

$$\Longleftrightarrow$$

$$\prod_{i=1}^{n}(x_i - dt) = d^n \cdot \prod_{i=1}^{n}(y_i/c - t)$$

$$\Longleftrightarrow$$

$$\prod_{i=1}^{n}(x_i/d - t) = \prod_{i=1}^{n}(y_i/c - t).$$

$$\Longleftrightarrow$$

$$\prod_{i=1}^{n}(t - x_i/d) = \prod_{i=1}^{n}(t - y_i/c).$$

Denote by P (resp. Q) the left (resp. right) hand of the last equation, that is $P \triangleq \prod_{i=1}^{n}(t - x_i/d)$ and $Q \triangleq \prod_{i=1}^{n}(t - y_i/c)$. P and Q can be seen as polynomials $P[t]$ and $Q[t]$ over $\mathbb{Z}_{\phi(N)}$ in the variable t. If $P[t]$ and $Q[t]$ are identical polynomials (the lists of the coefficients are identical), then the statement (i.e., that there exists a permutation p over $[n]$ such that for any $i \in [n], Y_i^d = X_{p(i)}^c$), is true (the viceversa holds as well). Indeed, suppose $P[t], Q[t]$ to be identical polynomials (the list of coefficients of $P[t]$ is equal to the list of coefficients of $Q[t]$). By definition of $P[t]$ and $Q[t]$, and from the fact that a polynomial of degree n, as $P[t]$ and $Q[t]$ are, has at most n roots, it follows that there exists a permutation

p over $[n]$ such that for every $i \in [n], x_i/d = y_{p(i)}/c \iff x_i \cdot c = y_{p(i)} \cdot d \iff X_i^c = Y_{p(i)}^d$, as it was to show. Hence, if the statement does not hold, the two polynomials have to be different.

The soundness error is the probability, over the random choices of $t \in \mathbb{Z}_{\phi(N)}$ of the verifier, that the proof is accepted but the statement does not hold. Therefore, the soundness error is equal to the probability that the last equation holds but the two polynomials $P[t]$ and $Q[t]$ are different. The probability that the last equation holds is equal to the probability, over a random $r \in \mathbb{Z}_{\phi(N)}$, that $P[t](r) = Q[t](r)$ (that is, that the evaluation of $P[t]$ at the point r equals the evaluation of $Q[t]$ at the point r). As a consequence, the probability that the last equation holds is equal to the probability, over the choices of $r \in \mathbb{Z}_{\phi(N)}$, that r is a zero of the polynomial $(P-Q)[t]$. Since $P[t]$ and $Q[t]$ are monic polynomials of degree n and they are different, the polynomial $(P-Q)[t]$ has at most $n-1$ roots, so the soundness error is $\leq (n-1)/\phi(N)$.

It is easy to see that the protocol is DHInvO-HZK, with $\mathsf{DHInvO}(\cdot)$ being the same oracle associated to the simulator of NIDDH. The DHInvO-oracle aided simulator $\mathsf{SimShfl}$ for the above protocol works as follows. The simulator takes as input $(C, D, X_1, \ldots, X_n, Y_1, \ldots, Y_n)$, chooses a random $t \in \mathbb{Z}_{\phi(N)}$ and computes $U \triangleq D^t = g^{dt}, W \triangleq C^t = g^{ct}$ and

$$\bar{X} \triangleq (\bar{X}_1, \ldots, \bar{X}_n) = (X_1/U, \ldots, X_n/U)$$

and

$$\bar{Y} \triangleq (\bar{Y}_1, \ldots, \bar{Y}_n) = (Y_1/W, \ldots, Y_n/W).$$

Such values are distributed identically as in a real proof. Finally, $\mathsf{SimShfl}$ returns the output of $\mathsf{SimILMP}$ on input the vectors $(\bar{X}, [C]^n)$ and $(\bar{Y}, [D]^n)$. As the distribution of the $\mathsf{SimILMP}$'s output for an ILMP instance is identical to a real proof for the same ILMP instance, the distribution of the $\mathsf{SimShfl}$'s output is identical to the distribution of the real transcripts for the above protocol.

Removing interaction and de-randomizing the protocol. The protocol can be made non-interactive using a non-programmable RO by computing the verifier message as output of the RO (note that the verifier message is the first message in the interaction, not the second like in a sigma protocol).

To remove the RO and the negligible soundness error, we recall that, as we proved above, there are at most $n-1$ values of t that can make the verifier to err. Therefore, the protocol can be repeated n times with *different* values of t. It is straightforward to see that the above simulator $\mathsf{SimShfl}$ can be adapted to this modification as well.

(Note that de-randomizing in this way the first message in the interactive protocol of Neff is possible but does not offer any advantage. Indeed, the Neff's interactive proof consists in a first random message sent by the verifier followed by a sigma protocol between prover and verifier. De-randomizing the first message would remove the need of the RO in the first verifier message but not for the verifier's message sent in the next three-round protocol.)

From a shuffle of group elements to a shuffle of El Gamal ciphertexts. The previous protocol could be improved, as done in Neff [Nef01] and following our ideas, to make it non-interactive and perfectly sound, to remove the limitation that the shuffler needs to know the discrete logs of all the X_i's and Y_i's; the resulting protocol would have computational HZK, that is the simulator with access to the oracle $\mathsf{DHInvO}(\cdot)$ outputs a transcript computationally indistinguishable from the one of the prover (cf. Def. 6). Moreover, as in Neff, the protocol can be generalized to any tuple of $k \geq 2$ elements (we have been implicitly assuming that in the previous analysis) and to a shuffle of El Gamal ciphertexts (i.e., pairs of group elements). We skip further details.

We instead show how our previous shuffle of group elements (in its limited version in which the prover needs to know the discrete logs of all the elements in the statement) can be adapted in a simpler way to a shuffle of 2 El Gamal ciphertexts. This is sufficient for our application to proofs for Circuit Satisfiability of Section 1.4.4.

For a given ciphertext $\mathsf{ct} = (c_1, c_2)$, let us denote by ct^l the left part c_1 (resp. by ct^r the right part c_2).

Let ct_1 and ct_2 be two El Gamal ciphertexts and let $P_1 \stackrel{\triangle}{=} \mathsf{ct}_1^l \cdot \mathsf{ct}_1^r$ and $P_2 \stackrel{\triangle}{=} \mathsf{ct}_2^l \cdot \mathsf{ct}_2^r$. The shuffler computes random commitments $C = g^c, D = g^d$, selects a random permutation p over [2] and sets $\mathsf{ct}'_1 \stackrel{\triangle}{=} ((\mathsf{ct}_{p(1)}^l)^{d/c}, (\mathsf{ct}_{p(1)}^r)^{d/c})$ and $\mathsf{ct}'_2 \stackrel{\triangle}{=} ((\mathsf{ct}_{p(2)}^l)^{d/c}, (\mathsf{ct}_{p(2)}^r)^{d/c})$, $P'_1 \stackrel{\triangle}{=} \mathsf{ct}_1'^l \cdot \mathsf{ct}_1'^r$ and $P'_2 \stackrel{\triangle}{=} \mathsf{ct}_2'^l \cdot \mathsf{ct}_2'^r$. Note that P'_1 (resp. P'_2) is the re-randomized permutation of P_1 (resp. P_2).

Note that if the original ciphertexts ct_1 and ct_2 encrypt resp. messages g^{m_1} and g^{m_2}, the shuffled ciphertexts will encrypt resp. messages $g^{d/c \cdot m_{p(1)}}$ and $g^{d/c \cdot m_{p(2)}}$, hence the message space will be changed. We will show later how to handle this issue for our application to OR proofs of Section 1.4.2 (used to construct proofs for Circuit Satisfiability).

The shuffler uses our previous proof of correct shuffle to prove that all the following statements hold:

- (1). $(\mathsf{ct}_1'^l, \mathsf{ct}_2'^l)$ is a shuffle with respect to the commitments C, D of $(\mathsf{ct}_1^l, \mathsf{ct}_2^l)$.
- (2). $(\mathsf{ct}_1'^r, \mathsf{ct}_2'^r)$ is a shuffle with respect to the commitments C, D of $(\mathsf{ct}_1^r, \mathsf{ct}_2^r)$.
- (3). (P'_1, P'_2) is a shuffle with respect to the commitments C, D of (P_1, P_2).

The verifier checks that $P_1 = \mathsf{ct}_1^l \cdot \mathsf{ct}_1^r, P'_1 = \mathsf{ct}_1'^l \cdot \mathsf{ct}_1'^r$, $P_2 = \mathsf{ct}_2^l \cdot \mathsf{ct}_2^r, P'_2 = \mathsf{ct}_2'^l \cdot \mathsf{ct}_2'^r$ and that the three proofs above are correct. Furthermore, the verifier checks that $\mathsf{ct}_1^l \neq \mathsf{ct}_2^l$ and $\mathsf{ct}_1^r \neq \mathsf{ct}_2^r$.

Roughly speaking, the shuffler computes three shuffles, one for the left part of the two ciphertexts, one for the right part, and one for the products, and proves that each shuffle individually is correct, and additionally performs some tests to enforce the prover to use the same permutation in all the shuffles.

Let us analyze the soundness. By (1), $(\mathsf{ct}_1'^l, \mathsf{ct}_2'^l) = ((\mathsf{ct}_{p(1)}^l)^s, (\mathsf{ct}_{p(2)}^l)^s)$, for some permutation p and $s = d/c$. By (2), $(\mathsf{ct}_1'^r, \mathsf{ct}_2'^r) = ((\mathsf{ct}_{q(1)}^r)^s, (\mathsf{ct}_{q(2)}^r)^s)$, for some permutation q and $s = d/c$. By (3), $(\mathsf{ct}_1'^l \cdot \mathsf{ct}_1'^r) = (\mathsf{ct}_{v(1)}^l \cdot \mathsf{ct}_{v(1)}^r)^s$ and $(\mathsf{ct}_2'^l \cdot \mathsf{ct}_2'^r) = (\mathsf{ct}_{v(2)}^l \cdot \mathsf{ct}_{v(2)}^r)^s$, for some permutation v and $s = d/c$.

So, $P'_1 =$ (by the verifier's checks) $= \mathsf{ct}'^l_1 \cdot \mathsf{ct}'^r_1 =$ (by (1) and (2)) $= (\mathsf{ct}^l_{p(1)} \cdot \mathsf{ct}^r_{q(1)})^s =$ (by (3)) $= (\mathsf{ct}^l_{v(1)} \cdot \mathsf{ct}^r_{v(1)})^s$, and similarly for P'_2. Suppose towards a contradiction that the proof is accepted (all the verifier's checks pass) but the ciphertexts are not permuted correctly, that is that $p \neq q$. We analyze four mutually exclusive cases and we reach a contradiction.

- p and v are the identity permutations. In this case, $P'_1 = (\mathsf{ct}^l_1 \cdot \mathsf{ct}^r_2)^s = (\mathsf{ct}^l_1 \cdot \mathsf{ct}^r_1)^s$. This contradicts the fact that $\mathsf{ct}^r_1 \neq \mathsf{ct}^r_2$.
- p is the identity and v is not the identity permutation. In this case, $P'_1 = (\mathsf{ct}^l_1 \cdot \mathsf{ct}^r_2)^s = (\mathsf{ct}^l_2 \cdot \mathsf{ct}^r_2)^s$. This contradicts the fact that $\mathsf{ct}^l_1 \neq \mathsf{ct}^l_2$.
- p is not the identity and v is the identity permutation. In this case, $P'_2 = (\mathsf{ct}^l_1 \cdot \mathsf{ct}^r_2)^s = (\mathsf{ct}^l_2 \cdot \mathsf{ct}^r_2)^s$. This contradicts the fact that $\mathsf{ct}^l_1 \neq \mathsf{ct}^l_2$.
- p and v are both different from the identity permutation. In this case, $P'_2 = (\mathsf{ct}^l_1 \cdot \mathsf{ct}^r_2)^s = (\mathsf{ct}^l_1 \cdot \mathsf{ct}^r_1)^s$. This contradicts the fact that $\mathsf{ct}^r_1 \neq \mathsf{ct}^r_2$.

To demonstrate the DHInvO-HZK of this modified proof, the previously described DHInvO-aided simulator $\mathsf{SimShfl}$ has to be modified as follows. The simulator $\mathsf{SimShfl}$ on input the two commitments C, D, two pairs of ciphertext ct_1 and ct_2 and the re-encrypted (according to C and D) and permuted resulting ciphertexts ct'_1 and ct'_2, computes what follows. It simulates a proof (using the previously described version of the simulator for the proof of correct shuffle of group elements) of the fact that $(\mathsf{ct}'^l_1, \mathsf{ct}'^l_2)$ is a shuffle with respect to the commitments C, D of $(\mathsf{ct}^l_1, \mathsf{ct}^l_2)$, of the fact that $(\mathsf{ct}'^r_1, \mathsf{ct}'^r_2)$ is a shuffle with respect to the commitments C, D of $(\mathsf{ct}^r_1, \mathsf{ct}^r_2)$ and of the fact that (P'_1, P'_2) is a shuffle with respect to the commitments C, D of (P_1, P_2), where, by their definitions, P_1, P_2, P'_1, P'_2 are publicly computable from the statement. By the previous analysis, it is straightforward to see that this modified simulator carries out a perfect DHInvO-aided simulation.

We stress that in our proof system for Boolean circuit satisfiability of Section 1.4.4 the prover does know the discrete logs of all the group elements. Furthermore, in our proof for Circuit satisfiability, the prover only needs to compute shuffles of 2 El Gamal ciphertexts. So the previous limited version of the shuffle (i.e., in which the prover needs to know the discrete logs) extended with the above modification to a shuffle of El Gamal ciphertexts suffices for the application to proofs of Boolean circuit satisfiability.

1.4.2 OR proofs from verifiable shuffle In an OR proof, the prover needs to convince the verifier that a compound statement like $x \in L \vee x_2 \in L$ holds. OR proofs [CDS94] for sigma protocols are achievable using the simulator constructively in the proof. The idea is that the prover uses the simulator to generate a proof for the part of the OR statement for which it does not know the witness. Proving that ciphertext c decrypts to 0 or 1 is easy using such a technique. This trick cannot be applied to NIDDH as its prover cannot execute the simulator without the oracle and in general cannot be applied to a HZK proof as the simulator for it can be only invoked on valid statements.

We can construct an OR proof for proving that an El Gamal ciphertext $\mathsf{ct} \stackrel{\triangle}{=} (c_1, c_2)$ (over $\mathbb{Z}_N^\star$) for public key pk decrypts to 0 or 1 from the previous proof of correctness of a shuffle of El Gamal ciphertexts.

We assume the prover to know the discrete logs in base g of c_1 and c_2. In our application to proofs for Circuit Satisfiability of Section 1.4.4 this is not a limitation as the prover of the NI for Circuit Satisfiability applies OR proofs from ciphertexts created by itself with knowledge of all the corresponding exponents.

The prover holds an El Gamal public key pk (over $\mathbb{Z}_N^\star$) and a ciphertext ct_1 encrypting a bit b and computes another ciphertext ct_2 encrypting $1-b$. Note that ct_1 encrypts the group element g^b and ct_2 encrypts the group element g^{1-b}, that is one of the two encrypts the group element $1 = g^0$ and the other one the group element g. Then the prover computes a shuffle $(\mathsf{ct}_{s,1}, \mathsf{ct}_{s,2})$ of $(\mathsf{ct}_1, \mathsf{ct}_2)$, that is it computes random commitments $C \stackrel{\triangle}{=} g^c, D \stackrel{\triangle}{=} g^d$ and a random permutation p over [2] and sets $\mathsf{ct}_{s,1}$ and $\mathsf{ct}_{s,2}$ such that $\mathsf{ct}_{s,i}^d = \mathsf{ct}_{s,p(i)}^c$, for $i = 1, 2$. (Given a ciphertext $\mathsf{ct} \stackrel{\triangle}{=} (c_1, c_2)$, we denote by ct^d the ciphertext (c_1^d, c_2^d).) The prover computes a proof of correctness of the previous shuffle of the two previous pairs of ciphertexts with respect to the commitments C, D. Furthermore, the prover uses NIDDH to show what values $\mathsf{ct}_{s,1}$ and $\mathsf{ct}_{s,2}$ decrypt to in the following way.

The prover proves that one of the two shuffled ciphertexts, w.lo.g. let us say $\mathsf{ct}_{s,1}$, decrypts to the group element $1 = g^0$ and that, w.l.o.g. $\mathsf{ct}_{s,2}$, decrypts to a group element Z such that the tuple $T \stackrel{\triangle}{=} (g, C = g^c, Z, D = g^d)$ is DH. Note that if the proof is accepted, both the original ciphertexts ct_1 and ct_2 encrypt a binary message (one of the two encrypts the group element g^0 and the other encrypts the group element g^1). This is because if T is DH, then $Z = g^{d/c}$ and so either ct_1 or ct_2 encrypts the group element g^1 (i.e., the bit 1). The verifier checks that the shuffle is correct and that one of the two shuffled ciphertexts, w.l.o.g. let us say $\mathsf{ct}_{s,1}$, decrypts to the group element $1 = g^0$ and that, w.l.o.g. $\mathsf{ct}_{s,2}$, decrypts to the group element Z and that the tuple T is DH. So, the verifier is convinced that ct_1 encrypts a bit (0 or 1).

It is easy to see that the proof inherits the same soundness guarantee of the underlying proof of correct shuffle, so it is perfectly sound. The DHInvO-aided simulation is straightforward given a DHInvO-aided simulator for the proof of correct shuffle and well-formedness of DH tuples. More in detail, consider the following DHInvO-aided simulator SimOR. The simulator SimOR, on input a public key pk and a ciphertext ct_1, computes what follows. Let b the bit encrypted in ct_1. SimOR computes a ciphertext encrypting 1 and uses the homomorphic properties of El Gamal to compute a ciphertext ct_2 encrypting $1-b$. Note that the simulator does not know the bit b but is able to obliviously compute a ciphertext encrypting the bit complement. Then, as the prover, the simulator computes a shuffle $(\mathsf{ct}_{s,1}, \mathsf{ct}_{s,2})$ of $(\mathsf{ct}_1, \mathsf{ct}_2)$, that is it computes random commitments $C \stackrel{\triangle}{=} g^c, D \stackrel{\triangle}{=} g^d$ and a random permutation p over [2] and sets $\mathsf{ct}_{s,1}$ and $\mathsf{ct}_{s,2}$ such that $\mathsf{ct}_{s,i}^d = \mathsf{ct}_{p(i)}^c$, for $i = 1, 2$. The DHInvO-aided simulator SimOR invokes $\mathsf{SimShfl}$

to simulate a proof of correctness of the previous shuffle. The simulator SimOR simulates a proof of the fact that one of the two shuffled ciphertexts, w.l.o.g. let us say $\mathsf{ct}_{s,1}$, decrypts to the group element $1 = g^0$ and that, w.l.o.g. $\mathsf{ct}_{s,2}$, decrypts to the group element $Z \triangleq g^{d/c}$, and simulates a proof that the tuple (g, C, Z, D) is DH. It is straightforward to see that the simulation of SimOR is perfect as the simulation of $\mathsf{SimShfl}$ and DHInvO is.

1.4.3 Polynomial statements The ideas and construction in this section are due to Geoffroy Couteau. An alternative approach to prove OR statements and more general polynomial statements is the following. We first present a proof subject to an issue and then we will show how to fix it. Let (g, h) be the El Gamal public key (our our group of quadratic residues modulo N), and let w be the secret key (hence $h = g^w$). Let (u, v) be a ciphertext that decrypts to a message m: $(u, v) = (g^r, h^r \cdot g^m)$. To prove that m is a bit, it suffices to prove that there exist values (w, m, x) such that the following four equations are satisfied:

1. $h = g^w$.
2. $v = u^w \cdot g^m$.
3. $1 = g^x \cdot h^m$. This equation ensures that $x = -w \cdot m$.
4. $1 = u^x \cdot (v/g)^m$. This equation ensures that $u^{-w \cdot m} \cdot (v/g)^m = 1$, which reduces to $g^{m \cdot (m-1)} = 1$, hence m is a bit.

The proof that these equations are satisfied works as follows.

Pick random masks w', m', and x' in $\mathbb{Z}^\star_m$, where m is the order of the group. Compute $c_0 = g^{w'}, c_1 = u^{w'} \cdot g^{m'}, c_2 = g^{x'} \cdot h^{m'}$, and $c_3 = u^{x'} \cdot (v/g)^{m'}$. The proof is:

$$(c_0, c_1, c_2, c_3, s_1, s_2, s_3, s_4, s_5, s_6),$$

with

$$s_1 = w'^{-1} \mod \phi(N), s_2 = m'^{-1} \mod \phi(N), s_3 = x'^{-1} \mod \phi(N),$$

$$s_4 = (w+w')^{-1} \mod \phi(N), s_5 = (m+m')^{-1} \mod \phi(N), s_6 = (x+x')^{-1} \mod \phi(N).$$

Then, similarly to the proof for DH tuples, the verifier uses s_1, s_2, s_3 to check that c_0, c_1, c_2, c_3 are well formed. To illustrate the check, suppose you want to check that $c_1 = u^{w'} \cdot g^{m'}$. This check is in fact equivalent to verifying:

$$(c_1^{s_2}/g)^{s_1} = u^{s_2},$$

which can be done using s_1 and s_2. Then, analogously the verifier uses s_4, s_5, s_6 to check that $(h \cdot c_0, v \cdot c_1, v \cdot c_2, v \cdot c_3)$ are well-formed.

There is a problem with the previous proof system: if we reveal $(m + m')^{-1}$ and m'^{-1}, then we leak whether m is equal to 0 or not. However, there is a simple fix: simply add 1 to m homomorphically and prove that m is equal to 1 or 2; the previous proof system has to be slightly modified in an obvious way[11] but we skip the details.

Again, we implicitly assumed all group elements to belong to the same subgroup. This limitation can be removed as detailed for the proof for DH tuples. To simulate a proof for polynomial statements, the simulator needs oracle access to DHInvO. We skip further details.

1.4.4 ZAP and computational HZK proof for $\mathcal{NP}$ relations The previous OR proofs or proofs for polynomial statements can be used to construct a one-message perfectly sound WI proof (non-interactive ZAP) for Boolean circuit satisfiability as follows. Our solution is inspired by [GOS12]. Assume the circuits consist only of NAND gates. If w_0, w_1 are the values corresponding to the input wires of a gate and w_2 is the value corresponding to its output wire, it is easy to see that w_0, w_1, w_2 are a valid assignment (i.e., $w_2 = \neg(w_0 \wedge w_1)$) iff $w_0 + w_1 + 2w_2 - 2 \in \{0, 1\}$ and $w_0, w_1, w_2 \in \{0, 1\}$.

Our NI proof for Boolean circuit satisfiability. The prover creates a public key (N, g, h) for our variant of El Gamal and associates a ciphertext to each wire of the circuit in the following way. To each input wire corresponding to a bit b of the witness, the prover associates a ciphertext encrypting b. The prover evaluates the circuit at each gate and associates to each output wire of a gate the encryption of the corresponding bit (computed homomorphically).

To each output wire of a gate and to each input wire of the circuit, the prover adds a proof of the fact that the associated ciphertext decrypts to 0 or 1.[12] Let t be a ciphertext encrypting the integer -2 (this can be done with trivial randomness, so that the verifier can verify that it be correctly computed). For each gate with ciphertexts $\mathsf{ct}_0, \mathsf{ct}_1$ associated to its input wires and ciphertext ct_2 associated to its output wire, the prover computes the ciphertext $G \stackrel{\triangle}{=} \mathsf{ct}_0 * \mathsf{ct}_1 * \mathsf{ct}_2^2 * t$ and adds a proof that G decrypts to 0 or 1. (Here, by "$*$" and exponentiation we mean the usual operations on El Gamal ciphertexts; cf. Def. 27.) Finally, the prover shows that the output gate decrypts to 1.

Using the homomorphic property of El Gamal and the above fact, it is easy to see that if all the proofs are accepted, the computation is consistent. Therefore, since the ciphertexts associated to the input wires decrypt to 0 or 1 and the output wire of the circuit decrypts to 1, the circuit is satisfiable. Notice that we can easily deal with the issue mentioned in Section 1.2.3. Indeed, for each ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$, the prover can add a proof that ct_1 belongs to the

[11] The essential modification is to adapt the equations so to ensure that $g^{(m-2)(m-1)} = 1$.

[12] Observe that the prover for the $\mathcal{NP}$ proof can use our OR proof from proof of correct shuffle in the simplified version in which the input to the prover for the proof of shuffle includes the discrete logs of the group elements in the statement. Indeed, in this case the prover for the $\mathcal{NP}$ proof can generate such group elements with knowledge of the corresponding discrete logs.

subgroup generated by g; this can be done by the prover since the prover can generate N with knowledge of the factorization.

computational HZK of our NI for Boolean circuit satisfiability. The previous NI is computational HZK: the simulator computes the public key and simulates an OR proof for each ciphertext associated with an internal wire and a proof of the fact that the ciphertext associated with the output gate decrypts to 1.

Therefore, for the simulator to carry out the simulation is sufficient to provide the simulator with access to the same oracle DHInvO associated with the HZK proof for DH tuples, and in addition another oracle Sim' (the two oracles can be seen as a single oracle). The oracle Sim', if invoked on a satisfiable Boolean circuit, computes the public key, and a bit-by-bit encryption of a witness, specifically the lexicographically first witness satisfying the statement, or returns error on input a non-satisfiable Boolean circuit.

Let SimNP be the oracle resulting from "joining" DHInvO and Sim'. The simulation is not perfect as the ciphertexts leak information (the witness chosen by the oracle may differ from the one in a real proof). However, our NI for Boolean circuit satisfiability can be conjectured to be SimNP-cHZK (cf. Def. 6). We state this as an assumption in itself and skip further details.[13] We also conjecture the proof to be strong cHZK (cf. Remark 2). We did not investigate whether the proof be simulatable via more "concise" oracles. Observe also that in the case of $\mathcal{NP}$ relations with single witness, the simulation is perfect and thus the NI is SimNP-HZK.

WI and WH of our NI for Boolean circuit satisfiability. Any O-cHZK proof is also WI: the computational WI property follows from the assumption that the simulated proof is computationally indistinguishable from a real proof and the computational WI property does not directly depend from the oracle. Later, we discuss in more detail this point.

In more detail, the computational WI property follows from the assumption that a PPT adversary cannot distinguish the encryption of one of two bits having in addition a proof (computed as described previously) of the fact that the ciphertext encrypts a bit (0 or 1) and access to an oracle (that can be invoked only on valid tuples) for computing proofs of valid statements of this sort. This can be seen to be equivalent to state that a PPT adversary cannot distinguish the encryption of one of two bits having the possibility of invoking the oracle DHInvO of the HZK simulator of our main proof system only on valid DH tuples. The precise statement is given in Assumption 6.

As discussed in Section 1.3.4 the existence of an O-aided simulator for a proof for a relation $\mathcal{R}$ implies that the proof is witness hiding with respect to distributions D (over pairs (x, w) such that $\mathcal{R}(x, w) = 1$) with the following property: there exists no PPT oracle adversary $\mathcal{B}$ with oracle access to O such

[13] The cHZK property might be reduced to "simpler" assumptions similar to the one used to prove the computational WI of our non-interactive ZAP, that is Assumption 6.

that if $(x, w) \leftarrow D$, then $\mathcal{B}$, on input x, outputs w with non-negligible probability. With respect to any such distribution over Boolean circuits and satisfying assignments for them, our proof is witness hiding.

NI proofs for all $\mathcal{NP}$ relations. Let R be an arbitrary $\mathcal{NP}$ relation and let L be the language associated with it. Since any instance of L can be polynomially reduced to an instance of the Boolean circuit satisfiability language via efficiently invertible transformations and there exists an efficient transformation that maps pairs in R to pairs in the Boolean circuit satisfiability relation, then the existence of a ZAP or cHZK proof for Boolean circuit satisfiability implies the existence of a ZAP or cHZK proof for R.

Oracle-aided simulatability as special case of unbounded simulatability. We saw that O-cHZK implies WI for *every* oracle O. That is, whatever oracle O is, an O-cHZK proof is also WI. This is because, by definition of O-cHZK, the simulated distribution is computationally indistinguishable from the real distribution irrespective of the fact that the simulated distribution is computed by an O-aided simulator. Observe that this actually holds even for proof systems endowed with simulators of *unbounded* time until such a simulator is able to output a simulated distribution that is computationally indistinguishable from the real distribution.

Simulatability in unbounded time may appear trivial but a careful thought shows it is not. Consider the following naive simulation strategy in unbounded time: the unbounded simulator can just find the witness by brute force and outputs what the prover would output on input that witness, so the output of the simulator is indistinguishable from the one of the prover. The flaw in the reasoning is that we implicitly assumed a construction of the prover and verifier algorithm but for which we did not state any property.

More in detail, the strategy would work straightforward for single-witness relations since the distribution of the simulator output on input a statement x would be identical to that of the prover algorithm on input (x, w), with w the unique witness to x. If instead the relation has multiple-witnesses and it is computationally hard given a witness w to x to find different witnesses w' to x, the previous strategy would be not well-defined: which witness to x has the simulator to find? The natural approach would be to use the first witness in lexicographic order. But then, the distribution of the prover output on input, let us say, (x, w) would have to be computationally indistinguishable from the distribution of the prover output on input (x, w'), with w' being the first witness to x in lexicographic order.

So, unbounded simulatability is non-trivial and useful and indeed it implies WI. The non-triviality comes in defining a complete and sound pair of prover and verifier algorithms and in addition ensuring that the prover outputs indistinguishable views on input (x, w) and (x, w'), with w, w' being any (possibly different) witnesses to the same statement x. Oracle-aided simulatability is a special case of unbounded simulatability as an oracle can be simulated in unbounded time, and so it is an useful and non-trivial notion as well. The relation between unbounded simulatability and WI for proof systems is similar to the relation between virtual black-box obfuscation (VBB) [BGI+01] with unbounded simulation and indistinguishability obfuscation (iO) [GGH+13]. In fact, iO is similarly implied by VBB with unbounded simulation, so the latter is not a trivial primitive.

Complexity leveraging and quasi-polynomial time simulation. Pass [Pas03b] showed how to use complexity leveraging arguments [CGGM00] to construct quasi-polynomial time simulatable two-message arguments extended in Barak and Pass [BP04] to quasi-polynomial time simulatable one-message arguments with uniform soundness.

The idea of Pass [Pas03b] is (simplifying) the following. The verifier sends some random challenge $c = f(r)$ for some one-way function f and the prover proves, using an argument of knowledge, that $x \in L$ or there exists a pre-image for c. The soundness follows from the one-wayness of f and the security of the argument of knowledge. The quasi-polynomial time simulatability holds setting the length of c so that the simulator can extract a pre-image in quasi-polynomial time. The protocol as described results in a four round argument (due to the use of the argument of knowledge), the rounds can be reduced to two using non-interactive ZAP and extractable commitments but we skip the details.

Barak and Pass [BP04] builds on this idea to remove the interaction. In essence, the prover sends a commitment c and a proof, using for instance a non-interactive ZAP, that $x \in L$ or there exists a pair of collisions for a hash function. Assuming uniform adversarial prover strategies only, the soundness holds because it is a difficult for a uniform adversary to come up with a pair of collisions for a hash function. The quasi-polynomial time simulatability follows setting the parameters appropriately so that the simulator can find a pair of collisions in quasi-polynomial time.

Our proof for Boolean circuit satisfiability is also harmless proof of knowledge, so it can be used in both constructions to achieve quasi-polynomial time simulation.

Chung *et al.* [CLMP12] present barriers to using black-box reductions for proving soundness of quasi-polynomial time simulatable arguments, essentially showing that the results in the aforementioned works in the area are optimal. In contrast, we showed that oracle-aided simulatability can overcome such barriers.

1.5 Related Work and Comparison

1.5.1 Zero-knowledge proofs and arguments of knowledge

Zero-knowledge proofs, introduced in the seminal work of Goldreich, Micali and Rackoff [GMR85,GMR89] and further refined in [GMW86,GMW91], have been one of the main building blocks of modern cryptography [GMW87,CCD88], have provided useful in complexity theory to establish that certain languages are unlikely to be $\mathcal{NP}$-complete [BHZ87,For87] and represent a fascinating concept in itself. *Argument systems* have been introduced by Brassard *et al.* [BCC88] as a natural relaxation of the notion of proof system in which the dishonest prover is restricted to be efficient.

Zero-knowledge Proof of knowledge systems. Proof of knowledge systems [TW87,BM88] [GMR89,BG93,Gol01] extend proof systems adding the property that if a prover succeeds in convincing the verifier that, e.g., some $\mathcal{NP}$ statement holds, then the prover "knows" a witness for the statement; this is formalized requiring the existence of an efficient extractor that can extract the witness given oracle access to the prover with probability higher than the success probability of the prover in convincing the verifier (this is a simplification, the actual definition has to take in account a knowledge error).

Limitations of zero-knowledge. Strong limitations for ZK proofs and arguments have been studied since its discovery. The impossibility of non-interactive ZK (in the plain model) is straightforward (see Goldreich [Gol01]). Goldreich and Oren [GO94] proved the impossibility of non-trivial 2-round ZK arguments. Regarding black-box simulation, the possibility of non-trivial 3-round ZK arguments has been ruled out by Goldreich and Krawczyk [GK90] while Canetti *et al.* [CKPR01] ruled out the existence of non-trivial constant-round concurrent ZK arguments. The limitations of strict polynomial-time simulators have been studied by Barak and Lindell [BL02]; the oracle-aided simulators in our proof systems run in strict polynomial-time.

Regarding ZK *proofs*, Katz [Kat08] showed that there exist no black-box simulatable 4-round ZK proofs. Recently, Fleischhacker *et al.* [FGJ18], improving Kalai *et al.* [KRR17], proved that if certain assumptions on program obfuscation hold, then 3-round ZK proofs for $\mathcal{NP}$, even with respect to non-black-box simulation, do not exist.

Fortnow [For87], and Aiello and Håstad [AH87] studied the limits of perfect and statistical ZK (that guarantee that the ensemble output by the simulator is, resp., distributed identically or statistically indistinguishable from the ensemble output by the prover) showing that is unlikely the existence of a perfect or statistical ZK proof system for an $\mathcal{NP}$-complete language.

1.5.2 CRS-based NIZKs

CRS-based*Non-interactive Zero-Knowledge* (NIZK) proof and argument systems have been intensively studied in the last 30 years in a sequel of works [BFM88] [DMP88,FLS90,RS92,DP92,BY93,BY96,Gol01,DDO$^+$01,Can01,Pas03a,BCNP04] [Ps05,GOS06b,GOS06a,AF07,GS08,GK08,GOS12,Pas13,BFS16,PS19].

One of the initial motivations for CRS-based NIZK proof was CCA-security [NY90,CS98,Sah99,DDO$^+$01,ES02,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.

In contrast, in *e-voting* [Cha81,CGS97,Adi08,RS06,JCJ10] the authority cannot be the same party to generate the CRS. Indeed, the authority must compute proofs of correctness of the tally and thus the CRS has to be setup by a trusted party. Our NI for $\mathcal{NP}$ of Section 1.4.4 is not based on any trusted parameter and may be used in several existing e-voting schemes (e.g., [Adi08]) to overcome this limitation.

NIZKs in the CRS model can be obtained from any trapdoor permutation [FLS90] (and thus from factoring), from bilinear groups [GOS06a], and recently from the learning with error problem [CLW18,PS19], only to cite the most notable constructions.

1.5.3 NIZKs in the RO model

Sigma protocols. Sigma protocols, which efficient NIZK arguments in the RO model are based on, have been intensively studied [GQ88,CP93,CDS94,FKI06,BR08] [Dam10,ABB$^+$10,YZ12,Mau15,GMO16] and incorporate properties both of interactive proof systems and proofs of knowledge systems.

Chaum and Pedersen [CP93] construct sigma protocols for the well-formedness of DH tuples that can be FS-transformed to non-interactive arguments in the programmable RO model. Our proof system of Section 3 (overview in Section 1.2) for the well-formedness of (a variant of) DH tuples is non-interactive and perfectly sound and does not rely on any trusted parameter.

Cramer *et al.* [CDS94] present techniques to achieve sigma protocol for compound statements and Maurer [Mau15] show that many known sigma protocols can be seen as a special case of an abstract protocol. Our techniques extend as well to OR proofs and polynomial statements; see Sections 1.4.2 and 1.4.3.

The RO model and NIZKs derived via FS-like transformations. An alternative to the CRS model is the RO model that assumes the availability of a perfect random function that in practice is implemented with a hash function. The RO model 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,KRR17] studied the security of the FS methodology. The FS transform can be applied to any sigma protocols as well as to any public coin honest-verifier ZK proof system like the protocol for Hamiltonicity [Blu86].

A 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], there has been a lot of work in investigating alternative transformations that achieve further properties or mitigate some issues of FS.

Pass [Pas03a] 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, 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.[14])

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

Lindell [Lin15] (see also the improvement of Ciampi *et al.* [CPSV16]) puts forward a new transformation that requires both a *non-programmable* RO and a CRS and has computational complexity only slightly higher than FS. The transformations of Lindell and Ciampi *et al.* are based on computational assumptions.

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].

The work of Mittelbach and Venturi has been improved by Kalai *et al.* [KRR17] 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.

Ishai *et al.* [IMS12] and Mahmoody and Xiao [MX13] construct unconditional sub-linear ZK *arguments* in the RO model using the FS transform.

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 make use of the general forking lemma of Bellare and Neven [BN06] that extends the forking lemma of Pointcheval and Stern [PS00]. Wee [Wee09] presents a hierarchy of RO models and some separations among them.

Recently, Canetti *et al.* [CLW18] showed how to instantiate correlation-intractable functions [CGH98] and turned a particular sigma protocol into a NIZK in the CRS model based on a variant of fully homomorphic encryption. The result of Canetti *et al.* has been exploited by Peikert and Shiehian [PS19] to construct the first NIZK in the CRS model based on learning with error (LWE), a widely studied post-quantum safe problem.

[14] The FS transform leads to statistically sound proof systems against computationally unbounded provers if constrained to a polynomial number of RO queries.

The FS and NIZKs in the quantum setting. In the quantum setting, the Fiat-Shamir and NIZK proof/argument systems in the RO model has been studied in several works [Unr12,ARU14,Unr15] but this goes beyond the scope of our work.

1.5.4 Verifiable shuffles The original concept of mix-nets has been introduced by Chaum [Cha81]. A variant of Chaum's mix-net, called *shuffle*, based on the re-encrypt and permute paradigm, has been introduced by Park *et al.* [PIK94] and made verifiable by Sako and Kilian [SK94]. To our knowledge, all known efficient (without passing through $\mathcal{NP}$ reductions) non-interactive "proofs" of correct shuffle of ciphertexts of any semantically secure cryptosystem, both in the CRS and RO model, and even the non-short ones, only achieve soundness against computationally bounded provers. Examples include [SK94,FS01,Nef01,Gro03,Gro05b] [Wik05,GL07a,GL07b,TW10]. Our verifiable shuffle of Section 1.4.1 extends the one of Neff [Nef01] making it non-interactive and perfectly sound and is not based on any trust assumption like the CRS or the RO model.

1.5.5 Witness indistinguishable systems Dwork and Naor [DN00] constructed 2-round *witness indistinguishable* proofs, called *ZAPs*, for any $\mathcal{NP}$ relation (assuming trapdoor permutations exist). The construction of Dwork and Naor allows for the first message (from verifier to prover) to be reused, thus only one message is required even if many statements have to be proven. Barak *et al.* [BOV03] constructed the first non-interactive ZAPs for any $\mathcal{NP}$ relation. Groth *et al.* [GOS12] construct non-interactive ZAPs for any $\mathcal{NP}$ relation from number-theoretic assumptions over bilinear groups. Using the same techniques of Groth *et al.* [GOS12], the NIZK proofs of Groth and Sahai [GS08] can be used to build non-interactive ZAPs. The latter are the only known non-interactive ZAPs for practical statements that do not employ $\mathcal{NP}$-reductions. Bitansky and Paneth [BP15] construct non-interactive ZAPs from indistinguishability obfuscation [GGH$^+$13] and one-way functions. In Section 1.4.4 we sketch the construction of a non-interactive ZAP with perfect soundness for Boolean circuit satisfiability. Our non-interactive ZAP is also computational HZK.

ZAPs only offer witness indistinguishability (WI) [FS90], a security guarantee strictly weaker than ZK. Indeed, for relations with single witness, a trivial proof system that outputs the witness as proof satisfies WI. On the other hand, HZK proofs can be applied to relations with single witness as well, as it is the case for the relation of well-formed DH tuples.[15] WI makes the construction of security protocols more complex; to make use of WI proofs one has to introduce relations with artificial witnesses to be able to switch witnesses in hybrid experiments. This problem does not only affect security reductions. In practice, the prover invoked by the actual protocol has to compute a proof for a more complex relation (that usually consists in an OR of different statements, some of which

[15] For simplicity, our formal definition of DH relation over groups of non-prime order is not single witness but can be done so.

are not related to the statement to prove, but are artificially needed to make the reduction go through), and this comes with an *additional* efficiency loss. As for ZK, HZK proofs instead do not pose this kind of efficiency loss.

As a reference, in the verifiable functional encryption construction of Badrinarayanan [BGJS16], four functional encryption instances have to be used in parallel along with non-interactive WI proofs for very complex relations. Moreover, due to the use of WI the construction of Badrinarayanan *et al.* is inherently selectively secure (i.e., the adversary has to choose the challenge messages before seeing the public key). On the other hand, our non-interactive HZK proof with perfect soundness for $\mathcal{NP}$ relations implies a fully (i.e., non-selective) secure verifiable functional encryption scheme based on just a single instance of a fully secure functional encryption in conjunction with HZK proofs.

To our knowledge, no efficient non-interactive WI statistically sound proof for DH-like languages was known (unless going through expensive $\mathcal{NP}$-reductions). Hash proof systems [CS02] provide a different avenue to obtain variants of designated verifier proof systems for several practical relations including the DH one.

As we show in Section 1.4.4, a proof system that is simulatable via a simulator of *unbounded* time is WI. Oracle-aided simulatability is a special case of unbounded simulatability and as such it is non-trivial and useful notion as it implies WI.

1.5.6 Other formulations of privacy alternative to ZK Pass [Pas03b] proposed a relaxation of ZK by allowing *quasi-polynomial time simulation* and in this setting presented two-message straight-line concurrent ZK arguments that are composable without requiring trusted parameters bypassing the impossibility results of [CKPR01,GK90]. The work of Pass is similar in spirit to ours in that the simulator is allowed more than PPT resources. HZK can be seen to be more general as a sub-exponential time computation can be simulated by an oracle.[16] In [Pas03b], the motivation was to extend the simulation techniques to enable advanced composition and reduce the round complexity of protocols. Pass [Pas03b] only provides computationally sound and interactive protocols, whereas our protocols are completely non-interactive and perfectly sound.

Barak and Pass [BP04] de-randomize the two-message protocol of Pass [Pas03b] constructing a one-message quasi-polynomial time simulatable argument system for $\mathcal{NP}$, in which the soundness condition holds only against uniform machines, under general non-standard complexity theoretic assumptions.

Chung *et al.* [CLMP12] present barriers to using black-box reductions for proving soundness of quasi-polynomial time simulatable arguments, essentially showing that the results in the aforementioned works are optimal.

[16] Let D be the oracle that gets as input the representation of a program P and an input x, and returns the output of the execution of P on x. For each sub-exponential simulator Sim, consider the D-aided PPT simulator Sim' that, on input x, invokes the oracle D on the bit representation of Sim concatenated to x, and returns the output of the oracle.

Our proofs can be tweaked using complexity leveraging arguments [CGGM00] along with the techniques in [Pas03b,BP04] to achieve one-message arguments with uniform soundness and quasi-polynomial time simulation or two-message arguments with standard soundness and quasi-polynomial simulation; we briefly discuss this point in Section 1.4.4.

Prabhakaran and Sahai [PS04] realized tasks in the Universal Composability framework [Can01], known to be impossible without trust assumptions, by allowing the adversary and the environment to have super-polynomial computational power. The work of Prabhakaran and Sahai also introduces the notion of imaginary angels that are oracles available to the environment and the adversary and shows this to be an useful abstraction at the aim of defining and analyzing the security of complex protocols. Our oracles are very similar in spirit to imaginary angels. Observe that, like for imaginary angels in the work of Prabhakaran and Sahai, our oracles are used to define and analyze the security but are not used by the actual parties in proof systems.

Strong-WI [Gol01,Gol04] is a stronger security guarantee than WI. Strong-WI requires that for two computationally indistinguishable statement distributions X_1 and X_2, a pair (x_1, π_1), in which π_1 is a proof for statement $x_1 \leftarrow X_1$, must be computationally indistinguishable from a pair (x_2, π_2), in which π_2 is a proof for statement $x_2 \leftarrow X_2$.

We are not aware of any work analyzing whether known non-interactive ZAPs are also strong-WI. Pass [Pas06a] show that it is not possible to reduce the strong-WI of a constant-round public-coin proof systems (with negligible soundness error) for Boolean satisfiability to one-way functions under black-box reductions. An O-HZK proof is O-strong-WI, that is strong-WI when quantifying distributions that are computationally indistinguishable even by adversaries with access to O; see Section 1.3.9, Def. 23 and Corollary 6.

Jain *et al.* [JKKR17] observe that there is no evidence of whether ZK is actually necessary to enforce honest behavior of protocols with indistinguishability-based security and, borrowing ideas from Aiello *et al.* [ABOR00], design variant of argument systems that can be used to recover several applications of ZK. Their argument systems satisfy strong-WI and a weaker simulation strategy, called distinguisher-dependent simulation, in which the simulator can depend on the distinguisher. The setting, called the *delayed-input distributional setting*, is different from WZK (see below) in that the instance is only determined by the prover in the last round of the interaction and the the statement being proven is chosen by the prover from a public distribution.

Khurana and Sahai [KS17] constructed the first 2-message arguments for $\mathcal{NP}$ achieving quasi-polynomial time simulation in which the simulated proofs are indistinguishable from real proofs by distinguishers running in time significantly larger than that of the simulator (improving on Pass [Pas03b], in which the simulated proofs were instead indistinguishable by distinguishers running in time significantly smaller than that of the simulator). Kalai *et al.* [KKS18] construct two-message arguments satisfying the above property but with simulated proofs statistically indistinguishable from real proofs by distinguishers running in time significantly smaller than that of the simulator. In the delayed-input distributional setting, with distinguisher-dependent (polynomial time) simulation, Kalai *et al.* [KKS18] improve the work of Jain *et al.* [JKKR17] achieving statistical privacy.

Witness hiding (WH) has been suggested by Feige and Shamir [FS90] and aims at guaranteeing that an adversary cannot extract a witness from a proof for a statement chosen according to some distribution. According to Feige and Shamir [FS90] "Witness hiding is a natural security requirement, and can replace zero knowledge in many cryptographic protocols". In [FS90,Gol01] has been shown that a WI argument for a *specific* relation with at least two independent witnesses is also WH with respect to some specific distribution. Note that this does not imply that any WI protocol for $\mathcal{NP}$ is also witness hiding, and so we cannot conclude that non-interactive ZAPs for $\mathcal{NP}$ are additionally WH proofs for all $\mathcal{NP}$ relations.

More generally, WI arguments are known be WH only in the special case of a relation that has at least two computationally independent witnesses, that is such that it is hard for an efficient algorithm, on input one witness, to compute another one. Note that our relation NIDDH (cf. Def. 28) is technically a relation with multiple witnesses but, given one witness as input, it is easy to compute another one.

Deshpande and Kalai [DK18] construct the first 2-message adaptively sound WH argument for $\mathcal{NP}$ (in the non-delayed input setting). All our proofs, and in particular our proof for $\mathcal{NP}$ of Section 1.4.4, are additionally 1-message harmless witness hiding perfectly sound proofs, under seemingly reasonable computational assumptions. *Harmless witness hiding* (HWH) is a natural *strengthening* of WH. WH requires that no efficient verifier can extract a witness after interacting with the prover on a randomly chosen instance (according to some distribution). HWH quantifies over efficient adversaries with access to some oracle. For instance, our NI NIDDH is HWH, under the assumption that no PPT adversary, with access to the oracle DHInvO (cf. Def. 30), can extract a witness from a randomly selected DH tuple over the multiplicative group $\mathbb{Z}_N^\star$, for a Blum integer N. For more details, see Section 1.3.4, Def. 21 and Lemma 5.

Haitner *et al.* [HRS09] and Pass [Pas11] present impossibility results for black-box reductions of standard assumptions to WH. Deng *et al.* [DSYC18] observed that such black-box impossibility results only hold for a particular type of restricted reductions and showed how to bypass them via different types of reductions. Our positive results for WH are established via reductions that satisfy the same restriction as in the lower bounds of Haitner *et al.* and Pass, so we cannot benefit from the techniques of Deng *et al.*. In Section 1.3.4 we elaborate on why the fact that our NI proofs are WH (under some computational assumption) does not contradict the aforementioned black-box impossibility results.

In Section 1.3.4, we argue that the assumption "the NI proof system NIDDH is WH" is a falsifiable assumption according to the classification of Gentry and Wichs [GW11] (see also Naor [Nao03]). We mention that Bellare and Palacio

[BP02] prove the security of the Schnorr identification protocol, that is a 3-round proof system, under the hardness of the one-more discrete logarithm problem that is similar in nature to our oracle-based assumptions.

Weak zero-knowledge [DNRS99] is a relaxation of ZK in which the simulator can depend on the distinguisher and it is known to imply WH. Bitansky *et al.* [BKP19] construct the first 2-message WZK arguments for $\mathcal{NP}$. WZK is impossible to achieve in 1-message [GO94].

Dwork and Stockmeyer [DS02] present two-round zero-knowledge proofs for *time-bounded provers* (e.g., running in time n^4) and simulators allowed to run in longer time than the prover (long enough to break the soundness of the system); they also observe that one-message proofs cannot be obtained in their setting.

De Santis *et al.* [DPY92] showed that for proof systems with *space-bounded verifiers*, there exists a one-message zero-knowledge proof system for $\mathcal{NP}$ with statistical soundness.

Bellare *et al.* [BFS16] study the problem of *subversion resistance*, that is they consider the case of an adversary colluding with the CRS generator. Our proofs are not based on any trusted parameter, hence are additionally subversion resistant.

Pass, Halpern and Raman [HPR09] present an *epistemic formula* that holds iff a proof system is ZK. It would be interesting to find a similar characterization for O-HZK proof systems.

Precise ZK (see Pass [Pas06b]) aims at bounding the knowledge gained by a verifier in an interaction in terms of the actual computation rather than the potential computation. The notion does not apply to NI systems. However, one-message HZK proofs are not precise in that the simulator is allowed access to a resource (the oracle) that a potential verifier *might* use in an attempt to attack the system. A more "precise" definition of O-HZK, meaningful also for NI systems, might be formulated as a stricter variant of O-FH (see Sections 1.3.4 and 2.2.6) requiring that for every adversary of time t computing a function of the witness from a proof with probability p, there exists a related O-aided adversary of time t that can compute the same function of the witness (without the proof) with the same probability p.

1.5.7 Cryptography in groups of hidden order. The power of *groups of hidden order* has been established in the seminal work of Rivest, Shamir and Adleman [RSA78] on the homonymous RSA cryptosystem. El Gamal-like encryption schemes based on trapdoors like ours have been already proposed in several works. Paillier [Pai99] and its elliptic curve variant [Gal02], Bresson *et al.* [BCP03] and [CL15] constructed linearly homomorphic encryption schemes based on variants of the discrete log problem with trapdoors. McCurley [McC88] and Schmuely [Shm85] observed that the computational Diffie-Hellman assumption in multiplicative groups modulo a composite number is equivalent to factoring. Hofheinz and Kiltz [HK09] proposed a cryptographic group in which variants of the El Gamal encryption scheme can be proven CCA-secure under the factoring assumption. Groth [Gro05a] show the potential usefulness of working in small subgroups of $\mathbb{Z}_N^\star$.

1.5.8 Summary of our improvements of the state of the art

Summing up, our work improves the state of the art as follows.

- We construct the first *non-interactive* proofs not relying on trusted parameters for non-trivial relations. Known non-interactive proofs without trusted parameters were only proven WI, whereas ours satisfy the stronger HZK property, a close variant of ZK rooted in the simulation paradigm. Our use of oracles to define and analyze security of proof systems is novel but shares similarities with imaginary angels introduced by Prabhakaran and Sahai [PS04] in the context of multi-party computation and universal composability. In this work, we show that HZK is useful and usable to prove security of larger protocols by presenting concrete examples (see Section 1.3.8).
- We construct proof systems with *perfect* soundness enjoying HZK. Interactive ZK proof systems cannot be perfectly sound.
- For the relation of well-formed DH tuples and correctness of shuffles, only NIZK arguments in the RO model resulting from FS-transforming sigma protocols for the same relations were known. We construct proofs for these relations enjoying perfect soundness and not based on any trusted parameter whereas previously known efficient constructions only achieved computational soundness in the programmable RO model. Our techniques extend to compound and polynomial statements.
- Our proof systems are additionally proofs of knowledge. In a completely non-interactive setting like ours, a proof of knowledge guarantees that an adversary that gets a randomly chosen statement cannot output an accepted proof for it (to come up with an accepted proof, it would need to know a witness for the statement). If the group parameter for the DH tuples is seen as a common parameter made public (that, however, does not have to be trusted), our proof for DH tuples also satisfies the standard definition of perfect extraction [GOS12].
- We construct a one-message perfectly sound WI proof (i.e., non-interactive ZAP) for Boolean circuit satisfiability from a number-theoretic assumption related to multiplicative groups of hidden order. Previous non-interactive ZAPs for Boolean circuit satisfiability were based on bilinear group assumptions [GOS12]. Our non-interactive ZAP is also (computational) HZK.
- Our proofs, and HZK proofs in general, satisfy O-strong-WI. While strong-WI requires that for two computationally indistinguishable statement distributions X_1 and X_2, a proof for statement $x_1 \leftarrow X_1$ must be computationally indistinguishable from a proof for statement $x_2 \leftarrow X_2$, O-strong-WI has a similar requirement but quantifies over statement distributions that are computationally indistinguishable even by adversaries with access to O. It has been observed that strong-WI can recover several applications of ZK. Similarly, this is true for O-strong-WI at the cost of basing the security on assumptions that hold even with respect to adversaries with access to O. In

this paper we present concrete examples in which this is the case; see Section 1.3.8.

– Witness hiding requires that no efficient verifier can extract a witness after interacting with the prover on a randomly chosen instance. Harmless witness hiding quantifies over efficient adversaries with access to some oracle and thus is stronger than witness hiding. All our NI proofs are 1-message harmless witness hiding proof systems under seemingly reasonable oracle-based computational assumptions. In particular, the assumption that our NI NIDDH is witness hiding is falsifiable.

  In Appendix A we analyze Assumption 4 against "generic attacks" that we therein define; in particular we prove that in a generic model Assumption 4 is equivalent to an assumption that is identical to Assumption 4 except that the goal of the attacker is just to factorize the modulus rather than also finding the exponent w.

2 Definitions

Notation. We use $\mathbb{N}$ to denote the set of all natural numbers. For any natural number $m > 0$, we let U_m stand for the *uniform distribution* over binary strings of length m. A *negligible* function $\mathsf{negl}(\cdot)$ is a function that is smaller than the inverse of any polynomial in λ (starting from a certain point). We denote by $[n]$ the set of numbers $\{1, \ldots, n\}$, by $|x|$ the bit length of $x \in \{0,1\}^\star$ and by $x||y$ the *concatenation* of any two strings x and y in $\{0,1\}^\star$. A function $\epsilon(\cdot)$ is non-negligible if ϵ is not a negligible function.

If S is a finite set, we denote by $a \leftarrow S$ the process of setting a equal to a uniformly chosen element of S. We denote by $\perp$ a special symbol not in $\{0,1\}^\star$.

An algorithm is *stateful* when it can store internal information between two executions and is probabilistic (or randomized) when can choose random coins during in its execution.

We let PPT stand for probabilistic polynomial-time and nuPPT for non-uniform PPT (by Adleman's theorem [MR95] we could consider only non-uniform deterministic algorithms without loss of generality). Unless otherwise specified, all our adversaries are modeled as nuPPT algorithms. Precisely, a non-uniform algorithm Adv is a family of probabilistic algorithms $\{\mathsf{Adv}_\lambda\}_{\lambda>0}$ parameterized by λ such that there exists a polynomial $p(\cdot)$ such that for all $\lambda > 0$, Adv_λ takes as inputs strings of length $p(\lambda)$, runs in time at most $p(\lambda)$ and its description has length at most $p(\lambda)$. We do not advocate that the non-uniform is the right model of adversarial behavior, in particular when proving implications of the form $A \to B$ via non-uniform reductions, with A, B being hardness problems postulated in the non-uniform model, we are assuming the stronger hypothesis that B is breakable by non-uniform algorithms, hypothesis that might be unrealistic in some circumstances. The choice of the non-uniform model is for simplicity and our assumptions and results can be often translated to the uniform model as well. We will make remarks when some results and conclusions present issues in the non-uniform model. In particular, at the end of Lemma 7 we remark that, because of the non-uniform model adopted both in the definition of O-HZK and O-HWH, the statement of the Lemma is not totally precise.

We use $A(x; r)$ to denote the output of A when run on input x and coin tosses r. Analogously, we denote by $f(x; r)$ the output of a (possibly uncomputable) function on input x and random coins r. For a probabilistic algorithm A (resp. function f), $A(x)$ (resp. $f(x)$) denotes the probability distribution of the output of A (resp. f) when run (resp. evaluated) with x as input and random coins r.

A *language* L is a (possibly infinite) set of binary strings. A language L is in the class $\mathcal{BPP}$ (*bounded-error probabilistic polynomial-time*) if there exists a (uniform) PPT algorithm $\mathcal{A}$ such that for all $x \in L$, $\text{Prob}[\mathcal{A}(x) = 1] \geq 2/3$ and for all $x \notin L$, $\text{Prob}[\mathcal{A}(x) = 1] \leq 1/3$, where in both cases the probability is taken over the random coins of the algorithm.

An *oracle* O is a possibly computationally unbounded and stateful randomized algorithm that takes as input strings in $\{0,1\}^\star$ and outputs strings in $\{0,1\}^\star \cup \{\perp\}$. An algorithm that can invoke an oracle and gets its output during its execution is called an oracle algorithm. We assume that an oracle algorithm can invoke an oracle to get its output in 1 step, that is the time required by the oracle to compute the output is not counted in the time of execution of the oracle algorithm that invokes the oracle. We denote by $A^{O(\cdot)}$ the execution of A with access to an oracle $O(\cdot)$. When it is clear from the context, we denote by 1 the trivial identity oracle $1(\cdot)$ that, on input an arbitrary string x, outputs x. We call an oracle algorithm with access to an oracle O an O-*aided* algorithm and an algorithm with access to some oracle an *oracle-aided* algorithm.

Given two families of random variables $X_0 \stackrel{\triangle}{=} \{X_{0,\lambda}\}_\lambda$ and $X_1 \stackrel{\triangle}{=} \{X_{1,\lambda}\}$, a function $\epsilon(\cdot)$ and a nuPPT algorithm $\mathcal{D} = \{D_\lambda\}_{\lambda>0}$, we say that $\mathcal{D}$ cannot *distinguish* X_0 from X_1 with *advantage* more than $\epsilon(\cdot)$ whether $|\text{Prob}[\mathcal{D}_\lambda(X_0) = 1 | \boldsymbol{x} \leftarrow X_{0,\lambda}] - \text{Prob}[\mathcal{D}_\lambda(X_1) = 1 | \ x \leftarrow X_{1,\lambda}]| \leq \epsilon(\lambda)$. If in addition $\mathcal{D}$ is given access to an oracle $O(\cdot)$ we say that $\mathcal{D}^{O(\cdot)}$ cannot distinguish X_0 from X_1 with advantage more than $\epsilon(\lambda)$. We say that two families of random variables $X_0 = \{X_{0,\lambda}\}_\lambda$ and $X_1 = \{X_{1,\lambda}\}$ are computationally indistinguishable if there exists no nuPPT algorithm $\mathcal{D}$ and there exists no non-negligible function $\epsilon(\cdot)$ such $\mathcal{D}$ distinguishes X_0 from X_1 with advantage more than $\epsilon(\cdot)$.

A *polynomial-time relation* (or $\mathcal{NP}$ relation) $\mathcal{R}$ is a relation for which membership of (x, w) in $\mathcal{R}$ can be decided in time polynomial in $|x|$, that is $\mathcal{R}$ is associated to a deterministic algorithm $\mathcal{A}$ such that $\mathcal{A}$ runs in time polynomial in its first input, $\mathcal{A}$ outputs a bit (the binary decision) and for each $(x, w), \mathcal{A}(x, w) = 1$ iff $(x, w) \in \mathcal{R}$. If $(x, w) \in \mathcal{R}$ then we say that w is a *witness* to *instance* (or statement or theorem) x and sometimes we write $\mathcal{R}(x, w) = 1$. A language $L \in \mathcal{NP}$ [Coo71,Kar72,Lev84] if there exists a polynomial-time relation $\mathcal{R}_L$ such that for each $x \in \{0,1\}^\star, x \in L$ iff there exists w such that $(x, w) \in \mathcal{R}_L$. A polynomial-time relation $\mathcal{R}$ is naturally associated with the $\mathcal{NP}$ language $L_\mathcal{R}$ defined as $L_\mathcal{R} = \{x | \ \exists w : \ (x, w) \in \mathcal{R}\}$. Given a language L, for any natural number $k > 0$, we denote by L_k the language $L \cap \{0,1\}^{\leq k}$. A relation $\mathcal{R}$ is a polynomial-time *single witness* (or unique witness) relation if $\mathcal{R}$ is a polynomial-time relation and for any two pairs $(x, w_1), (x, w_2)$, if $(x, w_1) \in \mathcal{R}, (x, w_2) \in \mathcal{R}$ then $w_1 = w_2$.

We assume familiarity with *interactive algorithms* (see [Gol01] for more details). Given two interactive algorithms M_0 and M_1, we denote by $\langle M_0(x_0), M_1(x_1)\rangle(x)$ the output of M_1 when running on input x_1 and interacting with M_0 running on input x_0 and common input x and by $\mathsf{view}_A\langle A(x_A), B(x_B)\rangle(x)$ the *view* of A during the interaction with B when both are executed on common input x and A (resp. B) is executed on input x_A (resp. x_B).

For any $k > 0$, any distribution X over inputs of length k, and any Boolean predicate $f : \{0,1\}^k \to \{0,1\}$, we denote by $\mathrm{Prob}[f(x)|\ x \leftarrow X]$ the probability that $f(x) = 1$ for $x \leftarrow X$.

With a slight abuse of notation, in the description of algorithms we sometimes use the symbols "$\stackrel{\triangle}{=}$" and "$=$" interchangeably to denote a copy of a memory area (corresponding to a variable in pseudo-code) into another.

2.1 Number-theoretic facts and definitions.

We call a λ-bit integer $N = p \cdot q$ an RSA *modulus* [RSA78] for security parameter λ if p and q are two distinct $\lambda/2$-bit odd primes and p and q are both congruent to 3 modulo 4; such a modulus N is also called a *Blum integer*. (In this work, we use the terms RSA modulus and Blum integer interchangeably preferring the latter to stress that p and q are both congruent to 3 modulo 4.) The group $\mathbb{Z}_N^\star$ consists of all the integers of $\mathbb{Z}_N$ that have an inverse modulo N with group operation being the multiplication modulo N and has order $\phi(N) = (p-1)(q-1)$, where $\phi(N)$ is the *Euler's totient function*. We denote by QR_N the set of all the elements y of $\mathbb{Z}_N$ such that there exists an integer $x \in \mathbb{Z}_N$ satisfying $y = x^2 \mod N$. QR_N can be shown to be a group under multiplication modulo N and is called the *group of quadratic residues modulo* N. If N is a Blum integer, it can be seen that $-1 \notin \mathsf{QR}_N$.

The Legendre symbol $\left(\frac{x}{p}\right)$ of an integer $x \in \mathbb{Z}_p$ for a prime p is defined to be 1 if $x \in \mathsf{QR}_p, x \neq 0$, -1 if $x \notin \mathsf{QR}_p$ and 0 if $x = 0$. It can be proven that for p prime $\left(\frac{x}{p}\right) = x^{(p-1)/2} \mod p$. For an RSA modulus $N = p \cdot q$, we denote by $\left(\frac{x}{N}\right)$ the Jacobi symbol of x modulo N and we define it as $\left(\frac{x}{N}\right) \stackrel{\triangle}{=} \left(\frac{x}{p}\right) \cdot \left(\frac{x}{q}\right)$. There is a PPT algorithm to compute the Jacobi symbol of x modulo N for any RSA modulo N [BSJS96].

By the Chinese remainder theorem (CRT) [BSJS96], the group $\mathbb{Z}_N^\star$, for an RSA modulus $N = p \cdot q$, is isomorphic to the product group $(\mathbb{Z}_p^\star \times \mathbb{Z}_q^\star)$. $\mathbb{Z}_p^\star$ (resp. $\mathbb{Z}_q^\star$) is cyclic of order $p-1$ (resp. $q-1$). Therefore QR_p (resp. QR_q) is cyclic of order $(p-1)/2$ (resp. $(q-1)/2$). By the Chinese remainder theorem, the group QR_N, for an RSA modulus $N = p \cdot q$, has order $(p-1)/2 \cdot (q-1)/2 = \phi(N)/4$ and is isomorphic to the product group $(\mathsf{QR}_p \times \mathsf{QR}_q)$, and hence is cyclic if $\gcd((p-1)/2, (q-1)/2) = 1$. Therefore, with respect to an RSA modulus $N = p \cdot q$, with $p \stackrel{\triangle}{=} 2p' + 1$ and $q \stackrel{\triangle}{=} 2q' + 1$, for primes p', q', the group QR_N is cyclic.

We let $\mathsf{GenRSA}(1^\lambda)$ be a PPT algorithm that generates elements (N, p, q, g) such that $N = p \cdot q$ is an RSA modulus for security parameter λ, $\gcd((p-1)/2, (q-1)/2) = 1$ and g is a generator of QR_N. From the previous facts, it is easy to see that such an algorithm exists.

If g is an element of some group, we denote by $\mathrm{ord}(g)$ its order.

By a generalization of the CRT, a system of equations over the integers

$$x \equiv a \mod m_1,$$

$$x \equiv b \mod m_2,$$

has a unique integer solution modulo $m_1 m_2/\gcd(m_1, m_2)$ if $a \equiv b \mod \gcd(m_1, m_2)$, otherwise it has no solution.

We will make use of a PPT algorithm to recognize prime numbers [AKS04].[17]

2.2 Proof systems.

2.2.1 Interactive and NI proof systems We now recall notions related to interactive proof systems and put forth our definitions for non-interactive harmless zero-knowledge proof of knowledge systems.

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

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

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

  The property may be relaxed to hold statistically by requiring the probability to be negligible in $|x|$. For the sake of our results, we are satisfied with statistical completeness. Actually, our proof systems, as they are described, satisfy statistical completeness because, e.g., the prover has to find random values that are invertible under some constraint. If the provers in our proof systems select the randomness properly, the required property holds w.v.h.p., so the statistical completeness follows. We believe that the provers can be changed so as to enjoy perfect completeness but we did not investigate the details. See also Remark 9.
- *Soundness.* For every non-uniform (possibly computationally unbounded) algorithm $\mathcal{P}^\star \stackrel{\triangle}{=} \{\mathcal{P}^\star_\lambda\}_{\lambda>0}$, it holds that for every polynomial $p(\cdot)$, there exists a constant n such that for every $\lambda \geq n$, for every $x \in \bar{L} \cap \{0,1\}^\lambda$, it holds that:

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

$\triangle$

[17] If we replaced this algorithm with an algorithm that errs with negligible probability like the Miller-Rabin algorithm [Rab80], our proof systems would incur a negligible soundness error in which the probability of error is over the random choices of the verifier, that is our proof systems would be such that if the verifier fails (with some probability over its random coins) in detecting that an integer sent from the prover is *not* a prime, the verifier may accept a false statement.

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

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

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

$\triangle$

Definition 3 [Perfect soundness] An interactive proof system $(\mathcal{P},\mathcal{V})$ for polynomial-time relation $\mathcal{R}$ associated with a language L satisfies perfect soundness (or it is perfectly sound) if it satisfies 0-soundness. $\triangle$

Definition 4 [Non-interactive proof system] A pair of PPT (non-interactive) algorithms $(\mathcal{P},\mathcal{V})$ is a non-interactive proof (NI) system for polynomial-time relation $\mathcal{R}$ associated with a language L if $(\mathcal{P},\mathcal{V})$ is a perfectly sound proof system and $\mathcal{V}$ is deterministic. For any $(x,w) \in \mathcal{R}$, we call any string π in the range of $\mathcal{P}(x,w)$ a proof for x. $\triangle$

Note that in the above definition we require $(\mathcal{P},\mathcal{V})$ to be non-interactive algorithms and thus the prover, on input (x,w) (and the random coins), outputs a proof π and the verifier $\mathcal{V}$, on input two strings x and π, outputs its decision on whether $x \in L$.

According to the previous definition, the term NI is synonymous of a non-interactive perfectly sound proof system. Sometimes, in this paper we denote by NI just a pair of non-interactive algorithms prover and verifier, and we use instead the term "NI proof" to stress the perfect soundness condition.

2.2.2 *O*-HZK

Definition 5 [Legitimate oracle] We definite a legitimate oracle by first defining an illegitimate (non-legitimate) oracle. A (possibly stateful and randomized) oracle $O(\cdot)$ is illegitimate for polynomial-time relation $\mathcal{R}$ associated with a language L if there exists a nuPPT O-aided algorithm $\mathsf{Adv}^{O(\cdot)} \stackrel{\triangle}{=} \{\mathsf{Adv}_\lambda^{O(\cdot)}\}_{\lambda>0}$ such that the following conditions hold:

- For every $x \in L$, $\mathsf{Adv}_{|x|}^{O(\cdot)}$ outputs 1 with probability $\geq 2/3$ and $\mathsf{Adv}_{|x|}^{O(\cdot)}$ never queries $O(\cdot)$ on a point y such that $O(y) = \perp$.
- For every $x \notin L$, $\mathsf{Adv}_{|x|}^{O(\cdot)}$ outputs 1 with probability $\leq 1/3$ and $\mathsf{Adv}_{|x|}^{O(\cdot)}$ never queries $O(\cdot)$ on a point y such that $O(y) = \perp$.

That is, an illegitimate oracle for a relation $\mathcal{R}$ over a language L makes L easy to decide against O-aided nuPPT adversaries.

An oracle $O(\cdot)$ for a polynomial-time relation $\mathcal{R}$ is *legitimate* if $O(\cdot)$ is *not* illegitimate for $\mathcal{R}$, in such case we write $O(\cdot) \in \mathsf{LegOr}^{\mathcal{R}}$. $\triangle$

Remark 1 The requirement on restricting the adversary to not query the oracle on points on which the oracle returns $\perp$ (to indicate error) is necessary to prevent the adversary to decide the language in a trivial way, e.g., querying the oracle on instances not belonging to the language.

At first sight, this condition may appear "weird" for the following reason. If the oracle cannot be invoked on invalid instances, how could an adversary ever benefit from the access to the oracle? Does not an adversary that invokes the oracle already "know" that the instance is in the language? It would seem so that the oracle can be never invoked. A careful reflection shows instead that the oracle might be invoked on any other valid instance that is not necessarily related to the input of the adversary or might be invoked on a string that is just a part of the whole input of the adversary; we will later further discuss this point.

What does the definition of legitimate oracle model? Later on, we will conjecture some language to be hard with respect to an adversary with access to a legitimate oracle for that language. We will now argue that this conjecture is necessary (it is a minimal assumption) to make any kind of HZK proof useful in cryptographic protocols.

Let us briefly recall the setting for traditional ZK proofs. A verifier needs a ZK proof of the fact that $x \in L$ because it cannot decide whether $x \in L$ or $x \notin L$ by itself. If, for instance, it were easy to distinguish DH tuples from non-DH tuples, we would not need any ZK proof for the well-formedness of DH tuples. So, in this sense the hypothesis that deciding the language of DH tuples is *worst-case* hard is a minimal assumption to make ZK proofs for this language useful. Observe that in practice one uses such ZK proofs for the well-formedness of DH tuples in protocols whose overall security is based on a related average-case assumption, e.g., the problem of distinguishing a random DH tuple from a random tuple. However, if the worst-case assumption does not hold, the average-case assumption does not hold as well, so the former is ever a necessary assumption.

Consider now the possibility of a world in which deciding the language of DH tuples (in the worst-case) is hard but it is easy if the adversary is given access to some oracle O that is only invoked on valid DH tuples. Then, a verifier that is given access to O would not need any proof of the fact that $x \in L$. Then, assuming every party to have access to O (an implicit assumption in the setting of HZK), requiring O to be a legitimate oracle is a necessary condition.

As for the case of traditional ZK proofs for DH tuples, in practice one would use our HZK proof for the well-formedness of DH tuples in protocols whose overall security is based on a related average-case assumption, e.g., Assumption 8 (see Section 1.3.8 on how to use HZK proofs to argue security, and in particular reducing the security of an e-voting application that uses our HZK proofs to such assumption). The latter assumption essentially states that it is difficult to distinguish a random DH tuple for witness w from a random tuple given additionally another random DH tuple for the same witness w and access to an oracle O that can be only invoked on valid DH tuples. If the oracle were not legitimate, it is easy to see that the latter assumption would not hold; this again to stress that requiring the oracle to be legitimate is a necessary assumption.

Assumption 8 also exemplifies why restricting the adversary to not query the oracle on invalid tuples is natural and useful. It turns out that if an adversary can break Assumption 7, there exists an adversary breaking Assumption 8 (see Lemma 11). The latter assumption essentially states that it is difficult to distinguish whether two El Gamal ciphertexts encrypt resp. $(m, -m)$ or $(-m, m)$ given additionally a proof that their "product" decrypts to 0. When proving that Assumption 8 implies Assumption 7 via hybrid experiments, we will have that the adversary, in one case, has to simulate one hybrid with input $Z = A^w$, where w is the witness for the DH tuple, and in another case, has to simulate another hybrid but with input a random Z. However, the adversary also gets as input a related DH tuple for the same witness w and such tuple, that is by definition always well-formed, is the input on which the adversary invokes the oracle (the HZK proof that that the adversary has to simulate in both hybrids is set to be the output of the oracle on such input). So, notwithstanding the restriction on the oracle queries, the adversary makes a non-trivial use of the oracle.
Assumption 8 and more generally our oracle-based assumptions have somehow the flavor of "one-more" RSA and discrete log assumptions [BNPS02,BP02] in that an adversary is asked to break some problem given access to an oracle that breaks other instances. Indeed, access to oracles in our assumptions allows the check the membership in some language.

One can argue that the restriction on the oracle queries in Assumption 8 is more natural than the restriction on the oracle queries in, e.g., the following assumption (that we will state later): the oracle DHInvO (cf. Def. 30), that returns error on input an invalid DH tuple, is legitimate for the relation of valid DH tuple (that we will define in Section 2.3). It seems that, unlike Assumption 8, the adversary cannot benefit from the oracle access. We remark that if in this case an adversary cannot make any non-trivial use of the oracle, all the more we are justified to allow the simulator access to the oracle. Indeed, we stress again that we would like to give the simulator access to all resources that cannot help the adversary to decide the language (without seeing proofs); if one could prove that, for the case of the relation of valid DH tuples, the oracle does not add any power to an adversary attempting to break the worst-case decisional hardness of the corresponding language, all the more so the oracle is a harmless resource and as such we grant the simulator access to it. Note that the assumption that the oracle DHInvO be legitimate for the relation of valid DH tuples basically accounts to say that an efficient adversary cannot distinguish a DH tuple from a non-DH tuple seeing additionally proofs of well-formedness of valid DH tuples of its choice.
We also point out that the oracle we will consider in this paper for the case of extraction (i.e., the factoring oracle) never outputs error. $\triangle$

Definition 6 [Computational harmless zero-knowledge] A NI system $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is computational harmless zero-knowledge (cHZK) if there exists a, possibly stateful and randomized, oracle $O(\cdot) \in \mathsf{LegOr}^{\mathcal{R}}$ and a PPT algorithm $\mathsf{Sim}^{O(\cdot)}$ (called the simulator) with oracle access to $O(\cdot)$ such that, for for every sequence $\{(x_\lambda, w_\lambda)\}_{\lambda>0}$ such that $\forall \lambda > 0\ (x_\lambda, w_\lambda) \in \mathcal{R}$ and

$|x_\lambda| \geq \lambda$, for every nuPPT O-aided distinguisher $\mathcal{D}^{O(\cdot)} = \{\mathcal{D}_\lambda^{O(\cdot)}\}_{\lambda>0}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, $\mathcal{D}_{|x_\lambda|}^{O(\cdot)}$ has advantage $< 1/p(|x_\lambda|)$ in distinguishing the following two random variables:

- $(x_\lambda, \mathcal{P}(x_\lambda, w_\lambda; U_{m(|x_\lambda|)}))$. (Where $m(\lambda)$ is the number of random coins $\mathcal{P}$ uses on an input x of length λ.)
- $(x_\lambda, \mathsf{Sim}^{O(\cdot;U_{d(|\cdot|)})}(x_\lambda; U_{s(|x_\lambda|)}))$. (Where $d(\lambda)$ is the total number of random coins O uses in a single invocation when running on an input x of length λ, and $s(\lambda)$ is the number of random coins $\mathsf{Sim}^{O(\cdot)}$ uses on an input x of length λ.)

$\triangle$

Definition 7 [Statistical harmless zero-knowledge] A NI system $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is statistical harmless zero-knowledge (sHZK) if there exists a, possibly stateful and randomized, oracle $O(\cdot) \in \mathsf{LegOr}^{\mathcal{R}}$ and a PPT algorithm $\mathsf{Sim}^{O(\cdot)}$ (called the simulator) with oracle access to $O(\cdot)$ such that, for for every sequence $\{(x_\lambda, w_\lambda)\}_{\lambda>0}$ such that $\forall\lambda > 0$ $(x_\lambda, w_\lambda) \in \mathcal{R}$ and $|x_\lambda| \geq \lambda$, for every non-uniform (possibly unbounded) distinguisher algorithm $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda>0}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, $\mathcal{D}_{|x_\lambda|}$ has advantage $< 1/p(|x_\lambda|)$ in distinguishing the following two random variables:

- $(x_\lambda, \mathcal{P}(x_\lambda, w_\lambda; U_{m(|x_\lambda|)}))$. (Where $m(\lambda)$ is the number of random coins $\mathcal{P}$ uses on an input x of length λ.)
- $(x_\lambda, \mathsf{Sim}^{O(\cdot;U_{d(|\cdot|)})}(x_\lambda; U_{s(|x_\lambda|)}))$. (Where $d(\lambda)$ is the number of random coins O uses in a single invocation when running on an input x of length λ, and $s(\lambda)$ is the number of random coins $\mathsf{Sim}^{O(\cdot)}$ uses on an input x of length λ.)

$\triangle$

Definition 8 [Perfect harmless zero-knowledge] A NI system $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is perfect harmless zero-knowledge (HZK) or simply harmless zero-knowledge if there exists a, possibly stateful and randomized, oracle $O(\cdot) \in \mathsf{LegOr}^{\mathcal{R}}$ and a PPT algorithm $\mathsf{Sim}^{O(\cdot)}$ (called the simulator) with oracle access to $O(\cdot)$ such that, for for every sequence $\{(x_\lambda, w_\lambda)\}_{\lambda>0}$ such that $\forall\lambda > 0$ $(x_\lambda, w_\lambda) \in \mathcal{R}$ and $|x_\lambda| \geq \lambda$, there exists a number $n > 0$ such that for every $\lambda \geq n$, the following two random variables are identically distributed:

- $(x_\lambda, \mathcal{P}(x_\lambda, w_\lambda; U_{m(|x_\lambda|)}))$. (Where $m(\lambda)$ is the number of random coins $\mathcal{P}$ uses on an input x of length λ.)
- $(x_\lambda, \mathsf{Sim}^{O(\cdot;U_{d(|\cdot|)})}(x_\lambda; U_{s(|x_\lambda|)}))$. (Where $d(\lambda)$ is the number of random coins O uses in a single invocation when running on an input x of length λ, and $s(\lambda)$ is the total number of random coins $\mathsf{Sim}^{O(\cdot)}$ uses on an input x of length λ.)

$\triangle$

Definition 9 [O-HZK] We say that a NI NI is O-HZK for polynomial-time relation $\mathcal{R}$, for some oracle $O(\cdot)$, if NI satisfies definition 8 with respect to $\mathcal{R}$ and a simulator with oracle access to $O(\cdot)$. Analogously, we say that a NI is O-cHZK or O-sHZK. $\triangle$

In this work, sometimes we informally talk about HZK even when we are actually referring to cHZK.

Definition 10 [ZK] We say that a NI NI is zero-knowledge (ZK) for polynomial-time relation $\mathcal{R}$ if NI satisfies definition 8 with respect to $\mathcal{R}$ and a simulator with oracle access to the trivial identity oracle $1(\cdot)$. Analogously, we say that a NI is cZK or sZK. $\triangle$

Remark 2 Note that cHZK, sHZK and HZK proofs are resp. O-cHZK, O-sHZK, O-HZK for *some* oracle O. In this work we often talk about HZK, sHZK and cHZK when we actually mean O-HZK, O-sHZK and O-cHZK for a specific oracle O that, when it is clear from the context, we omit.

We make further remarks on the previous definitions.

- Composition of cHZK proofs. Observe that the distinguisher for the real and simulated random variables in the definition of cHZK is given access to the oracle. This is necessary to prove sequential composition (see next). Also, compare it with the definition of ZK in the non-programmable and explicitly programmable RO model given in Wee [Wee09] in which the distinguisher is likewise given access to the RO.
- Conditionality. Our definitions of O-cHZK, O-sHZK and O-HZK, for any oracle O, are conditional in that we require the oracle O to be legal with respect to the relation. That is, a NI for a relation $\mathcal{R}$ over a language L is not O-HZK if $O \notin \mathsf{LegOr}^{\mathcal{R}}$, even if there exists an O-aided simulator satisfying the definition. This constraint is just to stress that, e.g., in that case a proof for $\mathcal{R}$ would not be meaningful as L is decidable (with access to O). The condition can be removed without any effect on our results.
- ZK as special case of HZK. Our definition of ZK as special case of HZK is syntactically different from the standard definition of ZK but is equivalent regarding language recognition.
- The order of quantifiers in cHZK and sHZK. The standard definition of computational ZK (as well as statistical ZK) follows a different type of quantification, namely cZK, following the standard quantification, would roughly say: "for any distinguisher, for any polynomial p, there exists n such that for any $\lambda \geq n$, for all $(x, w) \in \mathcal{R}$ such that $x \in \{0,1\}^{\lambda}$, the distinguisher has advantage $< 1/p(|x|)$ in distinguishing the simulated random variable for x from the real random variable for (x, w)."
  We say that a NI is resp. strong O-cHZK, strong O-sHZK (or just strong cHZK and sHZK), strong cZK if NI satisfies resp. definitions O-cHZK, O-sHZK (or just cHZK and sHZK), cZK changed with the previous stronger type of quantification.

Notice that in strong cZK, the value n depends only on the polynomial p. Instead, in our definition of cHZK (as well as sHZK), the point n depends on p but also on the sequence $\{(x_\lambda, w_\lambda)\}_{\lambda>0}$ that is universally quantified along with the distinguisher. That is, the point from which the distinguishing advantage is $< 1/p(|x_\lambda|)$ depends on such sequence. Therefore, our definitions of cHZK and sHZK are weaker than strong cHZK and strong sHZK.
Our definitional choice is justified as follows. Most of our results regard perfect HZK that is equivalent to the formulation with the standard type of quantification. However, we are also interested in showing impossibility results for non-interactive ZK (without setup). In Lemma 2 we indeed prove that *even* ZK with our weaker style of quantification is impossible to achieve for non-trivial languages, and thus we obtain a stronger impossibility result (though technically incomparable since the ”non-triviality“ property is different). Our only result regarding cHZK is our proof for $\mathcal{NP}$ relations of Section 1.4.4; such proof can be conjectured to satisfy strong cZK as well. Finally, cHZK seems sufficient in concrete applications. An adversary that can distinguish an hybrid experiment containing a proof computed by the prover from an hybrid experiment containing a simulated proof, can be used to construct a distinguisher against cHZK.

$\triangle$

Definition 11 [Unbounded computational harmless zero-knowledge] A NI system $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is unbounded computational harmless zero-knowledge (ucHZK) if there exists a, possibly stateful and randomized, oracle $O(\cdot) \in \mathsf{LegOr}^{\mathcal{R}}$ and a PPT algorithm $\mathsf{Sim}^{O(\cdot)}$ (called the simulator) with oracle access to $O(\cdot)$ such that, for every $m > 0$, every sequence $\{X_\lambda \stackrel{\triangle}{=} [(x^i_\lambda, w^i_\lambda)]_{i\in[m]}\}_{\lambda>0}$ such that $\forall\lambda > 0, i \in [m]$, $(x^i_\lambda, w^i_\lambda) \in \mathcal{R}$ and $|x^i_\lambda| \geq \lambda$, for every nuPPT O-aided distinguisher $\mathcal{D}^{O(\cdot)} = \{\mathcal{D}^{O(\cdot)}_\lambda\}_{\lambda>0}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, $\mathcal{D}^{O(\cdot)}_{|X_\lambda|}$ has advantage $< 1/p(|X_\lambda|)$ in distinguishing the following two random variables:

- $R_{\lambda,m} \stackrel{\triangle}{=} [(x^i_\lambda, \mathcal{P}(x^i_\lambda, w^i_\lambda; U_{m(|x^i_\lambda|)}))]_{i\in[m]}$. (Where $m(\lambda)$ is the number of random coins $\mathcal{P}$ uses on an input x of length λ.)
- $S_{\lambda,m} \stackrel{\triangle}{=} [(x^i_\lambda, \mathsf{Sim}^{O(\cdot;U_{d(|\cdot|)})}(x^i_\lambda; U_{s(|x^i_\lambda|)}))]_{i\in[m]}$. (Where $d(\lambda)$ is the total number of random coins O uses in a single invocation when running on an input x of length λ, and $s(\lambda)$ is the total number of random coins $\mathsf{Sim}^{O(\cdot)}$ uses on an input x of length λ.)

A NI NI is a O-ucHZK for polynomial-time relation $\mathcal{R}$ if NI is ucHZK with respect to an oracle $O(\cdot)$. $\triangle$

Lemma 1 [Composition of cZK proofs] If $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ is a O-cZK NI for a polynomial-time relation $\mathcal{R}$, then NI is a O-ucHZK NI for $\mathcal{R}$. $\triangle$

Proof. (Sketch) This follows by a standard hybrid argument observing that the distinguisher against a cHZK proof is given access to the oracle, by means of which, the distinguisher can compute simulated proofs. $\triangle$

Lemma 2 [Impossibility of non-interactive ZK for non-trivial languages] If $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ is a statistically sound cZK proof for a polynomial-time relation $\mathcal{R}$ associated with a language L, then L is trivial in the following sense: there exists a nuPPT adversary $\mathsf{Adv} \stackrel{\triangle}{=} \{\mathsf{Adv}_\lambda\}_{\lambda>0}$ such that for every sequence $\{x_\lambda\}_{\lambda>0}$ such that for every $\lambda > 0, |x_\lambda| \geq \lambda$, there exists a value $k > 0$ such that for every $\lambda \geq k$, if $x_\lambda \in L$, $\mathsf{Adv}_{|x|}(x) = 1$ with probability $\geq 2/3$ and if $x_\lambda \notin L$, $\mathsf{Adv}_{|x|}(x) = 1$ with probability $\leq 1/3$.

If NI satisfies strong cZK (cf. Remark 2), then $L \in \mathcal{BPP}$. $\triangle$

Proof. Let Sim be the 1-aided oracle guaranteed by the cZK property. In the following, without loss of generality, we assume Sim to not be oracle-aided (any invocation to the trivial identity oracle $1(\cdot)$ can be simulated with a constant overhead). Let $\mathsf{Adv} \stackrel{\triangle}{=} \{\mathsf{Adv}_\lambda\}_{\lambda>0}$ be the following nuPPT adversary against the worst-case membership hardness of L. Algorithm $\mathsf{Adv}_{|x|}$ receives as input a string x, invokes Sim on x to get an output π and returns $\mathcal{V}(x, \pi)$ as its decision on whether $x \in L$ or $x \notin L$. (Recall that, according to our definition of NI (cf. Def. 4), the verifier is deterministic. The impossibility can be easily extended to the case of probabilistic verifiers.)

Let us analyze the behavior of Adv. Consider an arbitrary sequence $\{(x_\lambda, w_\lambda)\}_{\lambda>0}$ such that $\forall \lambda > 0$ $(x_\lambda, w_\lambda) \in \mathcal{R}$, $|x_\lambda| \geq \lambda$.

By the cZK property, for every nuPPT distinguisher $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda>0}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, $\mathcal{D}_{|x_\lambda|}$ has advantage $< 1/p(|x_\lambda|)$ in distinguishing the following two random variables R_λ and S_λ:

- $R_\lambda \stackrel{\triangle}{=} (x_\lambda, \mathcal{P}(x_\lambda, w_\lambda; U_{m(|x_\lambda|)}))$. (Where $m(\lambda)$ is the number of random coins $\mathcal{P}$ uses on an input x of length λ.)
- $S_\lambda \stackrel{\triangle}{=} (x_\lambda, \mathsf{Sim}(x_\lambda; U_{s(|x_\lambda|)}))$. (Where $s(\lambda)$ is the number of random coins Sim uses on an input x of length λ.)

By statistical completeness, there is a negligible function $\nu(\cdot)$ such that (1) for any $\lambda > 0$, $\mathcal{V}$ accepts (x_λ, π), with $\pi \leftarrow \mathcal{P}(x'_\lambda, w_\lambda)$, with probability $\geq 1-\nu(\lambda)$ over the random coins of $\mathcal{P}$ and $\mathcal{V}$.

Suppose towards a contradiction that (2) there exists a polynomial $p(\cdot)$ such that for every $n > 0$ there exists $\lambda \geq n$ such that $\mathcal{V}$ accepts (x_λ, π), with $\pi \leftarrow \mathsf{Sim}(x_\lambda)$, with probability $< 1 - \nu(\lambda) - \frac{1}{p(|x_\lambda|)}$ over the random coins of Sim and $\mathcal{V}$. Consider the following nuPPT distinguisher $\mathcal{D}' \stackrel{\triangle}{=} \{\mathcal{D}'_\lambda\}_{\lambda>0}$ against the families of ensembles $\{R_\lambda\}_{\lambda>0}$ and $\{S_\lambda\}_{\lambda>0}$. $\mathcal{D}'_{|x_\lambda|}$ receives a string (x_λ, π) and outputs $\mathcal{V}(x_\lambda, \pi)$.

For every $\lambda > 0$, if $(x_\lambda, \pi) \leftarrow R_\lambda$, then, by (1) and definitions of R_λ and $\mathcal{D}'_{|x_\lambda|}$, $\mathcal{D}'_{|x_\lambda|}(x_\lambda, \pi) = 1$ with probability $\geq 1 - \nu(\lambda)$. By (2), for every $n > 0$, there exists $\lambda \geq n$ such that if $(x_\lambda, \pi) \leftarrow S_\lambda$, then, by definitions of S_λ and $\mathcal{D}'_{|x_\lambda}$, $\mathcal{D}'_{|x_\lambda|}(x_\lambda, \pi) = 1$ with probability $< 1 - \nu(\lambda) - 1/p(|x_\lambda|)$. Therefore, there exists a polynomial $p(\cdot)$ such that for every $n > 0$ there exists $\lambda \geq n$ such that $\mathcal{D}'_{|x_\lambda|}$ can distinguish the distributions R_λ and S_λ with advantage $\geq \frac{1}{p(|x_\lambda|)}$, a contradiction to the cZK property.

Since (2) does not hold, then there exists a negligible function $\nu'(\cdot)$ such that (3) for any $\lambda > 0$, $\mathcal{V}$ accepts (x_λ, π), with $\pi \leftarrow \mathsf{Sim}(x_\lambda)$, with probability $\geq 1 - \nu'(\lambda)$ over the random coins of Sim and $\mathcal{V}$. Since ν' is a negligible function, there exists a value $k_1 \geq 0$ such that for every $\lambda > k_1$, $\nu'(\lambda) \leq 1/3$. Hence, by definition of Adv, we have that (4) there exists a value $k_1 \geq 0$ such that for every $\lambda > k_1$, $\mathsf{Adv}_\lambda(x_\lambda) = 1$ with probability $\geq 2/3$.

By statistical soundness there exists a negligible function $\mu(\cdot)$ such that for every string π, every $x \notin L$, $\mathcal{V}(x, \pi) = 1$ with probability $\leq \mu(|x|)$ over its random coins. Since μ is a negligible function, there exists a value $k_2 \geq 0$ such that for every $\lambda > k_2$, $\mu(\lambda) \leq 1/3$. Hence, there exists a value $k_2 \geq 0$ such that for every $\lambda > k_2$, string π, $x_\lambda \notin L \cap \{0,1\}^{\geq \lambda}$, $\mathcal{V}(x_\lambda, \pi) = 1$ with probability $\leq \mu(\lambda) \leq 1/3$ over its random coins. Therefore, (5) for every sequence $\{x_\lambda\}_\lambda$ such that for every $\lambda > 0, |x_\lambda| \geq \lambda$, there exists a value $k_2 \geq 0$ such that for every $\lambda > k_2$, string π, if $x_\lambda \notin L$, $\mathcal{V}(x_\lambda, \pi) = 1$ with probability $\leq \mu(\lambda) \leq 1/3$ over its random coins. By definition of Adv and (5), we have that (6) for every sequence $\{x_\lambda\}_\lambda$ such that for every $\lambda > 0, |x_\lambda| \geq \lambda$, there exists a value $k_2 \geq 0$ such that for every $\lambda > k_2$, string π, if $x_\lambda \notin L$, $\mathsf{Adv}(x_\lambda) = 1$ with probability $\leq \mu(\lambda) \leq 1/3$ over its random coins.

Let k be the maximum of k_1 and k_2. Then, by (4) and (6) we have that for every sequence $\{x_\lambda\}_{\lambda>0}$ such that for every $\lambda > 0, |x_\lambda| \geq \lambda$, there exists a value $k > 0$ such that for every $\lambda \geq k$, if $x_\lambda \in L$, $\mathsf{Adv}(x) = 1$ with probability $\geq 2/3$ and if $x_\lambda \notin L$, $\mathsf{Adv}(x) = 1$ with probability $\leq 1/3$. This concludes the proof.

Finally, it is easy to see that the previous proof can be adapted for strong cZK to conclude that $L \in \mathcal{BPP}$. $\triangle$

Remark 3 Note that considering in the above lemma only cZK, rather than perfect ZK, and statistical soundness, rather than perfect soundness, makes the impossibility result stronger. $\triangle$

2.2.3 Hard relations and *O*-HPoK

Definition 12 [Hard relation] A polynomial-time relation $\mathcal{R}$ associated with a language L is said to be hard with respect to an algorithm Gen (called the generator) if:

- Gen, on input 1^λ, outputs a pair $(x, w) \in \mathcal{R}$ where $|x| = \lambda$.
- For all nuPPT algorithms $\mathcal{A} = \{A_\lambda\}_{\lambda>0}$, the quantity $\epsilon(\lambda) \stackrel{\triangle}{=} \mathrm{Prob}[(x, w) \in \mathcal{R}|\ x \leftarrow \mathsf{Gen}(1^\lambda);\ \mathsf{w} \leftarrow \mathcal{A}_{|\mathsf{x}|}(\mathsf{x})]$ is a negligible function in λ.

A polynomial-time relation $\mathcal{R}$ associated with a language L is said to be hard if it is hard with respect to some PPT algorithm Gen. $\triangle$

Definition 13 [Hard relation with respect to an oracle] Let $O(\cdot)$ be a, possibly stateful and randomized, oracle. A polynomial-time relation $\mathcal{R}$ associated with a language L is said to be hard with respect to an algorithm Gen and an oracle $O(\cdot)$ if:

- Gen, on input 1^λ, outputs a pair $(x, w) \in \mathcal{R}$ where $|x| = \lambda$.
- For all nuPPT algorithms $\mathcal{A}^{O(\cdot)} = \{A_\lambda^{O(\cdot)}\}_{\lambda>0}$ with access to $O(\cdot)$, the quantity $\epsilon(\lambda) \stackrel{\triangle}{=} \text{Prob}[(x, w') \in \mathcal{R}|\ (x, w) \leftarrow \mathsf{Gen}(1^\lambda);\ \mathsf{w}' \leftarrow \mathcal{A}_{|\mathsf{x}|}^{\mathsf{O}(\cdot)}(\mathsf{x})]$ is a negligible function in λ.

Let $O(\cdot)$ be a, possibly stateful and randomized, oracle. A polynomial-time relation $\mathcal{R}$ associated with a language L is said to be hard with respect to $O(\cdot)$ if it is hard with respect to some PPT algorithm Gen and to $O(\cdot)$. $\triangle$

It may be that computing the whole witness (w, p, q) to $\mathcal{R}_{\mathsf{DDH}}$ is hard but is instead easy to compute just the exponent w. The following definition captures the hardness of computing "harmful“ parts of the witness. In the previous example, the function f to hide would be the function that on input (w, p, q) outputs just w (ignoring the factorization).

Definition 14 [f-hard relation with respect to an oracle] Let f be a function and $O(\cdot)$ be a, possibly stateful and randomized, oracle. A polynomial-time relation $\mathcal{R}$ associated with a language L is said to be f-hard with respect to an algorithm Gen and an oracle $O(\cdot)$ if:

- Gen, on input 1^λ, outputs a pair $(x, w) \in \mathcal{R}$ where $|x| = \lambda$.
- For all nuPPT algorithms $\mathcal{A}^{O(\cdot)} = \{A_\lambda^{O(\cdot)}\}_{\lambda>0}$ with access to $O(\cdot)$, the quantity $\epsilon(\lambda) \stackrel{\triangle}{=} \text{Prob}[\exists w_2\ (x, f(w'||w_2)) \in \mathcal{R}|\ (x, w) \leftarrow \mathsf{Gen}(1^\lambda);\ \mathsf{w}' \leftarrow \mathcal{A}_{|\mathsf{x}|}^{\mathsf{O}(\cdot)}(\mathsf{x})]$ is a negligible function in λ.

Let $O(\cdot)$ be a, possibly stateful and randomized, oracle. A polynomial-time relation $\mathcal{R}$ associated with a language L is said to be f-hard with respect to $O(\cdot)$ if it is hard with respect to some PPT algorithm Gen and to $O(\cdot)$. $\triangle$

Definition 15 [Legitimate oracle for extraction] A, possibly stateful and randomized, oracle $O(\cdot)$ is said to be a legitimate oracle for extraction for a polynomial-time relation $\mathcal{R}$ if $\mathcal{R}$ is a hard relation with respect to $O(\cdot)$. In such case, we write $O(\cdot) \in \mathsf{LegOrHR}^{\mathcal{R}}$. $\triangle$

Definition 16 [Harmless proof of knowledge] A NI system $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is said to be harmless proof of knowledge (HPoK) if there exists a, possibly stateful and randomized, oracle $O(\cdot) \in \mathsf{LegOrHR}^{\mathcal{R}}$ and a PPT algorithm $\mathsf{Ext}^{O(\cdot)}$ (called the extractor) with oracle access to $O(\cdot)$ such that the following holds:

- For any strings $x, \pi \in \{0, 1\}^\star$, if $\mathcal{V}(x, \pi) = 1$ then $\text{Prob}[(x, w) \in \mathcal{R}(x, w)|\ w \leftarrow \mathsf{Ext}^{O(\cdot)}(x, \pi)] = 1$.

$\triangle$

Definition 17 [O-HPoK] We say that a NI NI is O-HPoK (or O-extractable) for polynomial-time relation $\mathcal{R}$, for some oracle $O(\cdot)$, if NI satisfies definition 16 with respect to $\mathcal{R}$ and an extractor with oracle access to $O(\cdot)$. $\triangle$

It is easy to see that HPoK implies the following.

Corollary 3 If $(\mathcal{P}, \mathcal{V})$ is a HPoK NI system for some polynomial-time relation $\mathcal{R}$, then the following holds:

– Let Gen be a PPT algorithm such that $\mathcal{R}$ is hard with respect to Gen. For any nuPPT algorithm $\mathsf{Adv} = \{\mathsf{Adv}_\lambda\}_{\lambda>0}$, $\epsilon(\lambda) \stackrel{\triangle}{=} \text{Prob}[\mathcal{V}(x, \pi) = 1 |\ x \leftarrow \mathsf{Gen}(1^\lambda);\ \pi \leftarrow \mathsf{Adv}_{|\mathsf{x}|}(\mathsf{x})]$ is a negligible function in λ.

$\triangle$

We call a NI system HZKPoK if it satisfies both HZK and HPoK.

Definition 18 [NIHZKPoK] A NIHZK is a NI system satisfying HZK and a NIHZKPoK is a NI system satisfying both HZK and HPoK. $\triangle$

2.2.4 WI, *O*-WI, WH and *O*-WH

Definition 19 [Witness indistinguishable NI system][FS90] A NI system for a polynomial-time relation $\mathcal{R}$ associated with a language L, consisting of a pair $(\mathcal{P}, \mathcal{V})$ of PPT algorithms, is called witness indistinguishable (WI) if it satisfies the following property.

– *Witness indistinguishability (WI):*
For every two sequences $\{(x_\lambda, w_\lambda^0)\}_{\lambda>0}$, $\{(x_\lambda, w_\lambda^1)\}_{\lambda>0}$, such that $\forall\lambda > 0$ $(x_\lambda, w_\lambda^0) \in \mathcal{R}$ and $(x_\lambda, w_\lambda^1) \in \mathcal{R}$ and $|x_\lambda| \geq \lambda$, for every nuPPT distinguisher $\mathcal{D} = \{\mathcal{D}_\lambda\}_\lambda$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$, $\mathcal{D}_{|x_\lambda|}$ has advantage $< 1/p(|x_\lambda|)$ in distinguishing the following two random variables:
  - $\mathcal{P}(x_\lambda, w_\lambda^0; U_{m(|x_\lambda|)})$.
  - $\mathcal{P}(x_\lambda, w_\lambda^1; U_{m(|x_\lambda|)})$.
    (Where $m(\lambda)$ is the number of random coins $\mathcal{P}$ uses on an input x of length λ.)

The above definition can be naturally extended to nuPPT distinguishers with access to an oracle O and in this case we talk about O-WI. $\triangle$

A NI system that satisfies WI is also called one-message (or non-interactive) *ZAP*.

The following corollary follows straightforward from the definition of O-cHZK.

Corollary 4 Let O be an oracle and $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ be an O-cHZK system for a polynomial-time relation $\mathcal{R}$. Them, NI is WI and O-WI. $\triangle$

Proof (Sketch). The proof trivially follows from the transitivity of the computational indistinguishability property.

Definition 20 [Harmless witness hiding] A NI system $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a hard polynomial-time relation $\mathcal{R}$ associated to generator Gen is said to be harmless witness hiding (HWH) if there exists a, possibly stateful and randomized, oracle $O(\cdot) \in \mathsf{LegOrHR}^{\mathcal{R}}$ such that the following holds:

– For all nuPPT algorithms $\mathcal{A}^{O(\cdot)} = \{A_\lambda^{O(\cdot)}\}_{\lambda>0}$ with access to $O(\cdot)$, the quantity $\epsilon(\lambda) \stackrel{\triangle}{=} \mathrm{Prob}[(x, w') \in \mathcal{R}|\ (x, w) \leftarrow \mathsf{Gen}(1^\lambda);\ \pi \leftarrow \mathcal{P}(\mathsf{x}, \mathsf{w});\ \mathsf{w}' \leftarrow \mathcal{A}_\lambda^{\mathsf{O}(\cdot)}(\mathsf{x}, \pi)]$ is a negligible function in λ.

$\triangle$

Definition 21 [O-HWH] We say that a NI NI is O-HWH, or simply O-WH, for a hard polynomial-time relation $\mathcal{R}$, for some oracle $O(\cdot)$, if NI satisfies definition 20 with respect to $\mathcal{R}$ and oracle $O(\cdot)$. $\triangle$

Definition 22 [WH] We say that a NI NI is witness hiding (WH) for a hard polynomial-time relation $\mathcal{R}$ if NI satisfies definition 20 with respect to $\mathcal{R}$ and the trivial identity oracle $1(\cdot)$. $\triangle$

Remark 4 [O-WH $\rightarrow$ WH] It is easy to see that, for any oracle O, O-WH implies WH as defined above that is equivalent to the traditional notion of witness hiding [FS90]. $\triangle$

Lemma 5 [O-HZK $\rightarrow$ O-WH] If a polynomial-time relation $\mathcal{R}$ is hard with respect to an oracle O (cf. Def. 13), then an O-HZK NI $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for $\mathcal{R}$ is O-WH. $\triangle$

Proof. Let $\mathcal{R}$ be hard with respect to generator Gen and oracle O. Suppose towards a contradiction NI to not be O-WH. Then, there exists a nuPPT algorithm $\mathcal{A}^{O(\cdot)} = \{A_\lambda^{O(\cdot)}\}_{\lambda>0}$ with access to $O(\cdot)$ such that the quantity $\epsilon(\lambda) \stackrel{\triangle}{=} \mathrm{Prob}[(x, w') \in \mathcal{R}|\ (x, w) \leftarrow \mathsf{Gen}(1^\lambda);\ \pi \leftarrow \mathcal{P}(\mathsf{x}, \mathsf{w});\ \mathsf{w}' \leftarrow \mathcal{A}_\lambda^{\mathsf{O}(\cdot)}(\mathsf{x}, \pi)]$ is a non-negligible function in λ. Consider the adversary $\mathsf{Adv}'^{O(\cdot)}$ with access to O that, on input x, runs the simulator $\mathsf{Sim}^{O(\cdot)}$ guaranteed by the O-HZK property, to compute an identically (to the proof computed by the prover) distributed proof π and outputs $\mathsf{Adv}^{O(\cdot)}(x, \pi)$. Then, $\epsilon'(\lambda) \stackrel{\triangle}{=} \mathrm{Prob}[(x, w') \in \mathcal{R}|\ (x, w) \leftarrow \mathsf{Gen}(1^\lambda);\ \mathsf{w}' \leftarrow \mathsf{Adv}_\lambda'^{\mathsf{O}(\cdot)}(\mathsf{x})] = \epsilon(\lambda)$ is a non-negligible function in λ, contradicting the fact that $\mathcal{R}$ is hard with respect to Gen and O. $\triangle$

2.2.5 *O*-strong-WI

Definition 23 [O-strong-WI] Let O be a, possibly stateful and randomized, oracle and $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ be a NI system for a polynomial-time relation $\mathcal{R}$. We say that NI is O-strong-WI if the following holds.

Let $\{X^b_\lambda\}_{\lambda>0}, b \in \{0,1\}$ be two ensembles of distributions such that for any $b \in \{0,1\}$, X^b_λ outputs a pair $(x_\lambda, w_\lambda) \in \mathcal{R}$ such that $|x_\lambda| \geq \lambda$. Suppose that for every O-aided nuPPT adversary $\mathsf{Adv}^{O(\cdot)} = \{\mathsf{Adv}^{O(\cdot)}_\lambda\}_{\lambda>0}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$,

$$\left|\text{Prob}[\mathsf{Adv}^{O(\cdot)}_\lambda(x^0_\lambda) = 1|\ (x^0_\lambda, w^0_\lambda) \leftarrow X^0_\lambda] - \text{Prob}[\mathsf{Adv}^{O(\cdot)}_\lambda(x^1_\lambda) = 1|\ (x^1_\lambda, w^1_\lambda) \leftarrow X^1_\lambda]\right| \leq 1/p(|x_\lambda|).$$

Then, for every nuPPT (non-oracle) adversary $\mathcal{B} = \{\mathcal{B}_\lambda\}_{\lambda>0}$, for every polynomial $p(\cdot)$, there exists a number $n > 0$ such that for every $\lambda \geq n$,

$$\big|\text{Prob}[\mathcal{B}_\lambda(x^0_\lambda, \mathcal{P}(x^0_\lambda, w^0_\lambda)) = 1|\ (x^0_\lambda, w^0_\lambda) \leftarrow X^0_\lambda] -$$

$$\text{Prob}[\mathcal{B}_\lambda(x^1_\lambda, \mathcal{P}(x^1_\lambda, w^1_\lambda)) = 1|\ (x^1_\lambda, w^1_\lambda) \leftarrow X^1_\lambda]\big| \leq 1/p(|x_\lambda|).$$

That is, O-strong-WI relaxes strong-WI [Gol01] by quantifying over distributions $\{X^b_\lambda\}_{\lambda>0}, b \in \{0,1\}$ that are computationally indistinguishable by O-aided nuPPT adversaries. △

Remark 5 It is easy to see that the standard definition of strong-WI is equivalent to 1-strong-WI for the trivial identity oracle $1(\cdot)$. △

The following corollary follows straightforward from the definition of O-HZK.

Corollary 6 Let O be an oracle and $\mathsf{NI} \triangleq (\mathcal{P}, \mathcal{V})$ be an O-HZK system for a polynomial-time relation $\mathcal{R}$. Them, NI is O-strong-WI. △

2.2.6 *O*-FH.

An alternative definition of privacy for NI systems, that we call O-function (or feature) hiding (O-FH) is the following.

Definition 24 [O-FH] Let $\mathsf{NI} \triangleq (\mathcal{P}, \mathcal{V})$ be a NI for a polynomial-time relation $\mathcal{R}$. In the following we will only consider randomized functions f with the property that f has finite range and is associated with a polynomial d such that, for any string x, $f(x)$ uses at most $d(|x|)$ random coins. Apart from this, f may be an arbitrary function. For any possibly randomized function f with finite range R, for any nuPPT algorithm $\mathsf{Adv} \triangleq \{\mathsf{Adv}_\lambda\}_{\lambda>0}$, for any pair $(x, w) \in \mathcal{R}$, let $P_{x,w,f,\mathsf{Adv}}$ be the following quantity:

$$\sum_{y \in R}\left|\text{Prob}[\mathsf{Adv}_{|x|}(x, \pi; r) = y|\ r, s \leftarrow \{0,1\}^{d(|x|)};\ \pi \leftarrow \mathcal{P}(x, w; s)] - \text{Prob}[f(x; r) = y|\ r \leftarrow \{0,1\}^{d(|x|)}]\right|,$$

where $d(\lambda)$ is the maximum of the random coins used by $\mathcal{P}$ when invoked on an input of length λ and by Adv_λ.

A NI system $\mathsf{NI} \triangleq (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is O-function (or feature) hiding (O-FH) if there exists a, possibly stateful and randomized, oracle $O(\cdot) \in \mathsf{LegOr}^{\mathcal{R}}$ such that the following holds. For any randomized function f with finite range R, for any nuPPT algorithm $\mathsf{Adv} \triangleq \{\mathsf{Adv}_\lambda\}_{\lambda>0}$ using at most $d(\lambda)$ coins on inputs of length λ, there exists a nuPPT O-aided algorithm $\mathsf{Adv}'^{O(\cdot)} \triangleq \{\mathsf{Adv}'^{O(\cdot)}_\lambda\}_{\lambda>0}$ such that, for any $(x, w) \in \mathcal{R}$, we have that:

$$
\begin{aligned}
&P_{x,w,f,\mathsf{Adv}} = \\
&\sum_{y \in R} \Big| \mathrm{Prob}[\mathsf{Adv}'^{O(\cdot)}_{|x|}(x; r||s) = y|\ r \leftarrow \{0,1\}^{d(|x|)};\ s \leftarrow \{0,1\}^{d'(|x|)-d(|x|)}] \\
&- \mathrm{Prob}[f(x; r) = y|\ r \leftarrow \{0,1\}^{d(|x|)}] \Big|,
\end{aligned} \tag{1}
$$

where $d(\lambda)$ is the maximum of the random coins used by Adv_λ and $d'(\lambda) \geq d(\lambda)$ is the maximum of the random coins used by $\mathsf{Adv}'^{O(\cdot)}$ when invoked on an input of length λ. $\triangle$

Definition 25 [FH] A NI system $\mathsf{NI} \triangleq (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is function (or feature) hiding (FH) if NI is 1-FH for the trivial identity oracle $1(\cdot)$. $\triangle$

Lemma 7 [O-HZK $\iff$ O-FH] A NI system $\mathsf{NI} \triangleq (\mathcal{P}, \mathcal{V})$ for a polynomial-time relation $\mathcal{R}$ is O-FH if NI is O-HZK. A NI system $\mathsf{NI} \triangleq (\mathcal{P}, \mathcal{V})$ for a single witness polynomial-time relation $\mathcal{R}$ is O-FH if and only if NI is O-HZK. (See note at the end of the proof.) $\triangle$

Proof.
- **If.** Let $\mathsf{Sim}^{O(\cdot)}$ be the O-aided simulator guaranteed by the O-HZK of NI. Let $s(\lambda)$ be the maximum number of coins (including the coins used in all the invocations to its oracle) that $\mathsf{Sim}^{O(\cdot)}$ uses when executed on an input of length λ. For any possibly randomized function f with finite range R, for any nuPPT algorithm $\mathsf{Adv} \triangleq \{\mathsf{Adv}_\lambda\}_\lambda$, for any pair $(x, w) \in \mathcal{R}$, consider the following non-uniform oracle algorithm $\mathsf{Adv}'^{O(\cdot)} \triangleq \{\mathsf{Adv}'^{O(\cdot)}_\lambda\}_{\lambda>0}$ with access to O. The algorithm $\mathsf{Adv}'^{O(\cdot)}_{|x|}$, on input x, computes $\pi \leftarrow \mathsf{Sim}^{O(\cdot)}(x)$ simulating an invocation of Sim to O with its own oracle O, and outputs $\mathsf{Adv}_{|x|}(x, \pi)$. The algorithm $\mathsf{Adv}_{|x|}$ uses $d'(|x|)$ coins where $d'(\lambda)$ is the sum

of $s(\lambda)$ and the maximum number of coins $d(\lambda)$ used by Adv_λ.

$$
\begin{aligned}
&P_{x,w,f,\mathsf{Adv}} \triangleq \\
&= \sum_{y\in R} \Big| \mathrm{Prob}[\mathsf{Adv}_{|x|}(x,\pi;r)=y|\ r,s\leftarrow\{0,1\}^{d(|x|)};\ \pi\leftarrow\mathcal{P}(x,w;s)] \\
&- \mathrm{Prob}[f(x;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)}]\Big| = \\
&\text{(by the perfect } O\text{-HZK of } \mathsf{NI}) \\
&= \sum_{y\in R} \Big| \mathrm{Prob}[\mathsf{Adv}_{|x|}(x,\pi;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)};\ s\leftarrow\{0,1\}^{s(|x|)};\ \pi\leftarrow\mathsf{Sim}^{O(\cdot)}(x;s)] \\
&- \mathrm{Prob}[f(x;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)}]\Big| = \\
&\text{(by the definition of } \mathsf{Adv}') \\
&= \sum_{y\in R} \Big| \mathrm{Prob}[\mathsf{Adv}'^{O(\cdot)}_{|x|}(x,\pi;r||s)=y|\ r\leftarrow\{0,1\}^{d(|x|)};\ s\leftarrow\{0,1\}^{s(|x|)}] \\
&- \mathrm{Prob}[f(x;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)}]\Big|,
\end{aligned}
\tag{2}
$$

as it was to show.

- **Only if (in the case of single witness relations).** Let L the language associated with the single witness polynomial-time relation $\mathcal{R}$. Consider the randomized function $f(x;r)$ that, on input a string x, outputs $\perp$ if $x\notin L$, and $\mathcal{P}(x,w;r)$, where w is such that $(x,w)\in\mathcal{R}$, otherwise. That is:

$$
f(x;r) \triangleq \begin{Bmatrix} \perp, & \text{if } x\notin L \\ \mathcal{P}(x,w;r), & \text{where } (x,w)\in\mathcal{R},\ \text{otherwise.} \end{Bmatrix}
$$

  Note that the function is well-defined as $\mathcal{R}$ is a single witness relation, so for any $x\in L$, there is a *unique* witness w such that $(x,w)\in\mathcal{R}$. Let R be the range of f.

  Consider an adversary Adv that, on input (x,π), just outputs π and thus is deterministic (uses 0 random coins). By O-FH, there exists a nuPPT oracle algorithm $\mathsf{Adv}'^{O(\cdot)} \triangleq \{\mathsf{Adv}'^{O(\cdot)}_\lambda\}_{\lambda>0}$ with oracle access to O such that, for any $(x,w)\in\mathcal{R}$, $P_{x,w,f,\mathsf{Adv}}$ is equal to the following quantity:

$$
\begin{aligned}
&\sum_{y\in R} \Big| \mathrm{Prob}[\mathsf{Adv}'^{O(\cdot)}_{|x|}(x;s)=y|\ s\leftarrow\{0,1\}^{d'(|x|)}] \\
&- \mathrm{Prob}[f(x;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)}]\Big|,
\end{aligned}
\tag{3}
$$

  where $d'(\lambda)$ is the maximum of the random coins used by Adv_λ.

We have that:

$$
\begin{aligned}
P_{x,w,f,\mathsf{Adv}} \stackrel{\triangle}{=}\ & \\
& \sum_{y\in R}\Big|\mathrm{Prob}[\mathsf{Adv}_{|x|}(x,\pi)=y|\ s\leftarrow\{0,1\}^{d(|x|)};\ \pi\leftarrow\mathcal{P}(x,w;s)] \\
& -\mathrm{Prob}[f(x;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)}]\Big| = \\
& \text{(by the definition of } \mathsf{Adv}\text{)} \\
& =\sum_{y\in R}\Big|\mathrm{Prob}[\pi=y|\ s\leftarrow\{0,1\}^{d(|x|)};\ \pi\leftarrow\mathcal{P}(x,w;s)] \\
& -\mathrm{Prob}[f(x;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)}]\Big| = \\
& \text{(by the definition of } \pi\text{)} \\
& =\sum_{y\in R}\Big|\mathrm{Prob}[\mathcal{P}(x,w;s)=y|\ s\leftarrow\{0,1\}^{d(|x|)}] \\
& -\mathrm{Prob}[f(x;r)=y|\ r\leftarrow\{0,1\}^{d(|x|)}]\Big| = \\
& \text{(by the definition of } f\text{)} \\
& =\sum_{y\in R}\Big|\mathrm{Prob}[\mathcal{P}(x,w;s)=y|\ s\leftarrow\{0,1\}^{d(|x|)}]-\mathrm{Prob}[\mathcal{P}(x,w;s)=y|\ s\leftarrow\{0,1\}^{d(|x|)}]\Big| = \\
& =0.
\end{aligned} \tag{4}
$$

By equations 3 and 4 and definition of f, we conclude that

$$
\begin{aligned}
& \sum_{y\in R}\Big|\mathrm{Prob}[\mathsf{Adv}'^{O(\cdot)}_{|x|}(x;s)=y|\ s\leftarrow\{0,1\}^{d'(|x|)}] \\
& -\mathrm{Prob}[\mathcal{P}(x,w;s)=y|\ s\leftarrow\{0,1\}^{d(|x|)}]\Big| = 0.
\end{aligned} \tag{5}
$$

Therefore, adversary $\mathsf{Adv}'^{O(\cdot)}$ is an O-aided simulator such that, for any $(x,w)\in\mathcal{R}$, the distribution $\mathsf{Adv}'^{O(\cdot)}_{|x|}(x)$ is distributed identically to the distribution $\mathcal{P}(x,w)$, as it was to show (see next note).

Note. Note that in the proof for the "only if part" we actually constructed a nuPPT simulator whereas in the definition of O-HZK we required the simulator to be PPT. This is an artifact of the definition of O-FH. To formally prove the Lemma, we should change either the definition of O-HZK weakening the simulator to be nuPPT or considering PPT algorithms in the definition of O-FH. To not overburden the presentation, we skip these details.

$\triangle$

Corollary 8 If $\mathsf{NI}\stackrel{\triangle}{=}(\mathcal{P},\mathcal{V})$ is a NI system for a single witness polynomial-time relation $\mathcal{R}$ associated with a non-easy language in the sense of Lemma 2, then NI is not FH. $\triangle$

Proof. This follows from Lemma 2 and the fact that a NI system $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a single witness polynomial-time relation $\mathcal{R}$ is FH if and only if NI ZK.

$\triangle$

Remark 6 It is easy to observe that the previous proof that O-FH (resp. FH) implies O-HZK (resp. ZK) extends to the case of a NI $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P}, \mathcal{V})$ for a general polynomial-time relation $\mathcal{R}$ such that, for any $(x, w_1) \in \mathcal{R}$ and $(x, w_2) \in \mathcal{R}$, the random variable $\mathcal{P}(x, w_1)$ is identically distributed to $\mathcal{P}(x, w_2)$.

For instance, the relation $\mathcal{R}_{\mathsf{DDH}}$ (cf. Def. 28), as we have formulated, is technically a relation with multiple witnesses but for any x, w_1, w_2 such that $(x, w_1) \in \mathcal{R}_{\mathsf{DDH}}$ and $(x, w_2) \in \mathcal{R}_{\mathsf{DDH}}$, the random variable $\mathsf{ProveDDH}(x, w_1)$ (cf. Construction 2) is identically distributed to $\mathsf{ProveDDH}(x, w_2)$. $\triangle$

Remark 7 [Equivalence between O-FH and O-HZK for relations with multiple witnesses] It is worth observing why the previous proof of equivalence strongly uses the hypothesis of the single witness relation. For simplicity, let us neglect the oracle in the analysis and only consider the relation between FH and ZK for NI proofs. Recall that according to our definition, ZK requires perfect simulation.

FH requires that for any randomized function of a statement x, if the output of an adversary, on input x, and a proof for a pair (x, w') in the relation, has distance p from $f(x)$, then there exists a simulator Adv' such that $\mathsf{Adv}'(x)$ is distributed at the same distance p from $f(x)$. Consider the function f_i such that $f_i(x; r) \stackrel{\triangle}{=} \mathcal{P}(x, w_i; r)$, where w_i is the i-th witness (in lexicographical order) to x. Let w' be an arbitrary witness to x. Suppose an adversary Adv, on input x, and a proof π for (x, w') is such that $\mathsf{Adv}(x, \pi)$ has distance p from the distribution $f_i(x) = \mathcal{P}(x, w_i)$. Then, FH implies the existence of a simulator Sim_i such that $\mathsf{Sim}_i(x)$ has distance p from $\mathcal{P}(x, w_i)$. Notice that the simulator guaranteed by FH can depend on the function and thus on the witness w_i.

In the proof that O-FH implies O-HZK for single witness relations, we considered the function that computes a proof for the unique witness to the statement and analyzed the simulator for that specific function. In the case of general relations, if we change the function to compute a proof for another witness (e.g., the second witness in lexicographical order), the simulator changes accordingly and we cannot conclude the existence of *fixed* simulator for O-HZK.

ZK (with perfect simulation) implies that the simulated proof for a statement x has to be distributed identically to a real proof for *any* pair (x, w) in the relation. Therefore, ZK forces the distribution of proofs for any two pairs $(x, w_1), (x, w_2)$ in the relation to be identical. In a previous version of this manuscript, we believed that this fact does not hold for O-FH and thus in an inequivalence between the two notions of O-HZK and O-FH (and so between ZK and FH). However, a careful analysis shows that also O-FH implies that two proofs for any two pairs $(x, w_1), (x, w_2)$ in the relation have the same distribution. This fact, combined with Remark 6, implies that O-FH is actually equivalent to O-HZK. We omit the details.

Similar considerations would apply to formulations of FH and ZK for interactive proof systems. $\triangle$

2.3 Multiplicative groups of hidden order

2.3.1 El Gamal over groups of hidden order Let GenRSA be defined as in Section 2.1, i.e., on input the security parameter 1^λ generates elements (N, p, q, g) such that $N = p \cdot q$ is an RSA modulus for security parameter λ, $\gcd((p-1)/2, (q-1)/2) = 1$ and g is a generator of QR_N. Henceforth, we will often denote by m the order of QR_N that, as shown in Section 2.1, equals $\phi(N)/4$.

Assumption 1 [DDH over GenRSA] Let $X_{0,\lambda}$ be the random variable (N, g, h, u, v), with $(N, p, q, g) \leftarrow \mathsf{GenRSA}(1^\lambda)$, $w \leftarrow \mathbb{Z}_m^\star, h \stackrel{\triangle}{=} g^w, u \leftarrow \mathsf{QR}_N, v \stackrel{\triangle}{=} u^w$. Let $X_{1,\lambda}$ be the random variable (N, g, h, u, v), with $(N, p, q, g) \leftarrow \mathsf{GenRSA}(1^\lambda)$, $h, u, v \leftarrow \mathsf{QR}_N$. We say that the Decisional Diffie-Hellman assumption (DDH) holds for generator GenRSA if for every nuPPT algorithm $\mathsf{Adv} = \{\mathsf{Adv}_\lambda\}_{\lambda>0}$, the following quantity is negligible in λ:

$$|\mathrm{Prob}[\mathsf{Adv}_\lambda(x) = 1|\ x \leftarrow X_{0,\lambda}] - \mathrm{Prob}[\mathsf{Adv}_\lambda(X_1) = 1|\ x \leftarrow X_{1,\lambda}]|\,.$$

DDH holds if it holds for generator GenRSA. $\triangle$

We assume the reader have familiarity with the notion of public key encryption scheme. We define an exponential El Gamal encryption scheme over the group QR_N and message space $\mathcal{M} \stackrel{\triangle}{=} \{0, 1, \ldots, d\}$ where d is an integer such that for any $M \in \mathcal{M}$, M can be computed by g^M in polynomial-time.

Definition 26 [El Gamal over groups of hidden order] Our El Gamal encryption scheme $\mathsf{ElGamal} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ over message space $\mathcal{M}$ is a tuple of 3 PPT algorithms.

$\mathsf{Setup}(1^\lambda)$: on input the security parameter λ it runs $(N, q, q, g) \leftarrow \mathsf{GenRSA}(1^\lambda)$, computes $h = g^w$ for randomly chosen integer in $\mathbb{Z}_{\phi(N)}^\star$ and outputs *public key* $\mathsf{pk} \stackrel{\triangle}{=} (N, g, h)$ and *secret key* $\mathsf{sk} \stackrel{\triangle}{=} (\mathsf{p}, \mathsf{q}, \mathsf{w})$.

$\mathsf{Enc}(\mathsf{pk}, M)$: on input public key $\mathsf{pk} \stackrel{\triangle}{=} (N, g, h)$ and a message $M \in \mathcal{M}$, outputs *ciphertext* $\mathsf{ct} \stackrel{\triangle}{=} (g^r, h \cdot g^M)$.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: on input secret key $\mathsf{sk} \stackrel{\triangle}{=} (\mathsf{p}, \mathsf{q}, \mathsf{w})$ and ciphertext $\mathsf{ct} \stackrel{\triangle}{=} (\mathsf{ct}_1, \mathsf{ct}_2)$, compute $y = \mathsf{ct}_2 \cdot \mathsf{ct}_1^{-w}$ and, by brute force search over all elements $M \in \mathcal{M}$ until an element M such that $y = g^M$ is found and in such case output M; if no such element can be found, output $\perp$.
When it is clear from the context, we sometimes assume Dec to just output y, that is Dec computes y as above and outputs it.

$\triangle$

It is easy to see that the following facts hold.

Fact 9 $\mathsf{ElGamal}$ satisfies *(perfect) correctness*, that is that for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $M \in \mathcal{M}$, for all $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$, $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{M}$. $\triangle$

Fact 10 If DDH holds over generator GenRSA, then $\mathsf{ElGamal}$ is *indistinguishable against chosen message attack* ($\mathsf{IND\text{-}CPA}$), that is for every 2 messages $M, M' \in \mathcal{M}$, and every PPT adversary $\mathcal{A}$, the following quantity is negligible in λ:

$$\left|\text{Prob}[\mathcal{A}(\mathsf{Enc}(\mathsf{pk}, M)) = 1 \mid \mathsf{pk} \leftarrow \mathsf{Setup}(1^\lambda))] - \text{Prob}[\mathcal{A}(\mathsf{Enc}(\mathsf{pk}, M')) = 1 \mid \mathsf{pk} \leftarrow \mathsf{Setup}(1^\lambda))]\right|.$$

$\triangle$

Definition 27 [Operations on El Gamal ciphertexts]

- Multiplication of El Gamal ciphertexts. Let $\mathsf{ct}_1 \stackrel{\triangle}{=} (\mathsf{ct}_1^l, \mathsf{ct}_1^r), \mathsf{ct}_2 \stackrel{\triangle}{=} (\mathsf{ct}_2^l, \mathsf{ct}_2^r)$ be two El Gamal ciphertexts for the same public key pk. We denote by $\mathsf{ct}_1 * \mathsf{ct}_2$ the ciphertext $(\mathsf{ct}_1^l \cdot \mathsf{ct}_2^l, \mathsf{ct}_2^r \cdot \mathsf{ct}_2^r)$, that is we multiply the two ciphertexts entry by entry.
- Exponentiation of an El Gamal ciphertext to a constant. Let y be an integer and $\mathsf{ct} \stackrel{\triangle}{=} (\mathsf{ct}^l, \mathsf{ct}^r)$ an El Gamal ciphertext. We denote by ct^y the ciphertext $((\mathsf{ct}^l)^y, (\mathsf{ct}^r)^y))$, that is we exponentiate each entry of the ciphertext to y.

$\triangle$

2.3.2 Our relations $\mathcal{R}_{\mathsf{DDH}}$ and $\mathcal{R}_{\mathsf{SG}}$

Definition 28 [$\mathcal{R}_{\mathsf{DDH}}$] The relation of well-formedness of Diffie-Hellman (DH) tuples is defined as follows: $\mathcal{R}_{\mathsf{DDH}}((N, g, h, u, v), (w, [p_i, m_i]_{i=1}^l)) = 1$ iff $g, h, u, v \in \mathbb{Z}_N$ and $u = g^w, v = h^w$ for some non-negative integer $w < \phi(N)$ (with group operation being the multiplication modulo N) and $N = \prod_{i=1}^l p_i^{m_i}$. $\triangle$

Definition 29 [$\mathcal{R}_{\mathsf{SG}}$] The relation of *subgroup membership* between two group elements is defined as follows: $\mathcal{R}_{\mathsf{SG}}((N, g, u), (w, [p_i, m_i]_{i=1}^l)) = 1$ iff $g, u \in \mathbb{Z}_N$ and $u = g^w$ for some non-negative integer $w < \phi(N)$ (with group operation being the multiplication modulo N) and $N = \prod_{i=1}^l p_i^{m_i}$. $\triangle$

Remark 8 We refer the reader to Section 1.3 for a discussion about the need of the factorization of the modulus in the definition of our relations. Note also that the relations do not guarantee N to have the correct form or g to be in QR_N. If this is not the case, $\mathsf{ElGamal}$ might be not secure.

The definition of $\mathcal{R}_{\mathsf{DDH}}$ is very general, in particular it is trivially satisfied for particular choices of g and h (e.g., when g and h have co-prime orders). In relevant applications, the relation will be used for g and h belonging to the same subgroup or in conjunction with a proof for $\mathcal{R}_{\mathsf{SG}}$ to guarantee g and h to be in the same subgroup. $\triangle$

2.3.3 Our main oracle DHInvO

Definition 30 [Oracle DHInvO] The oracle DHInvO takes as input a tuple (N, g, h, u, v) and checks whether $u = g^w$ and $v = h^w$ for some $w \in \mathbb{Z}^\star_{\phi(N)}$; if such value w

does not exist, it outputs $\perp$ to indicate an error; otherwise, it outputs $(g^r, h^r, r^{-1} \mod \phi(N), (r+w)^{-1} \mod \phi(N))$, with w being an integer $< \mathrm{ord}(g)$ such that $u = g^w, v = h^w$ and $r \leftarrow \mathbb{Z}^\star_{\phi(N)}$ under the constraints that r' be prime, $(r+w) \mod \phi(N) \in \mathbb{Z}^\star_{\phi(N)}$, and $(r+w)^{-1} \mod \phi(N)$ be prime. $\triangle$

Remark 9 Actually, our proof systems, as they are described, satisfy statistical completeness because, e.g., the prover has to find random values that are invertible under some constraint. As our provers might incur a statistical completeness error, in order to get perfect simulation the simulator should err as the prover. Hence, we implicitly assume for all our NIs that if a prover or simulator fails in some bounded polynomial time to find values satisfying the constraints, it outputs an error symbol (different from $\perp$), e.g., $(0, 0, 0, 0)$. We believe that our provers (and thus the corresponding simulators) can be changed so as to enjoy perfect completeness but we did not investigate the details. $\triangle$

Assumption 2 The assumption states that $\mathsf{DHInvO} \in \mathsf{LegOr}^{\mathcal{R}_{\mathsf{DDH}}}$ and $\mathsf{DHInvO} \in \mathsf{LegOr}^{\mathcal{R}_{\mathsf{SG}}}$. That is, DHInvO is a legitimate oracle for both the polynomial-time relations $\mathcal{R}_{\mathsf{DDH}}$ and $\mathcal{R}_{\mathsf{SG}}$. $\triangle$

Note that the latter assumption basically accounts to say that an efficient adversary cannot distinguish a DH tuple from a non-DH tuple seeing additionally proofs of well-formedness of valid DH tuples of its choice. See also Remark 1.

2.3.4 Hardness assumptions

Assumption 3 Let $\mathsf{GenDDH}(1^\lambda)$ be the generator that, on input security parameter 1^λ, computes $(N, p, q, g) \leftarrow \mathsf{GenRSA}(1^\lambda)$ and outputs statement (N, g, g^r, g^w, g^{rw}) and witness (w, p, q), with $r, w \leftarrow \mathbb{Z}^\star_{\phi(N)}$, for relation $\mathcal{R}_{\mathsf{DDH}}$. The assumption postulates that $\mathcal{R}_{\mathsf{DDH}}$ is hard (cf. Def. 13) with respect to GenDDH and oracle DHInvO. $\triangle$

In the previous assumption, the adversary's goal is to compute a witness (w, p, q) for relation $\mathcal{R}_{\mathsf{DDH}}$.

In the following assumption we require the adversary to compute only w (without the factorization). This captures the hiding of the "most harmful" part of the witness, the exponent of the DH tuple that allows to test whether the tuple is DH or related tuples for the same witness are DH.

Assumption 4 Let $\mathsf{GenDDH}(1^\lambda)$ be the generator that, on input security parameter 1^λ, computes $(N, p, q, g) \leftarrow \mathsf{GenRSA}(1^\lambda)$ and outputs statement (N, g, g^r, g^w, g^{rw}) and witness (w, p, q), with $r, w \leftarrow \mathbb{Z}^\star_{\phi(N)}$, for relation $\mathcal{R}_{\mathsf{DDH}}$. Let f be the function that on input (w, p, q, g) returns w (ignoring the factorization). The assumption postulates that $\mathcal{R}_{\mathsf{DDH}}$ is f-hard (cf. Def. 13) with respect to GenDDH and oracle DHInvO. $\triangle$

In Appendix A we show that the previous assumption is immune from "generic attacks" that we therein define.

Remark 10 Consider the function g that takes as input a witness $(w, [p_i, m_i]_{i=1}^{l})$ to an instance of $\mathcal{R}_{\mathsf{DDH}}$ and outputs $w \mod \phi(\prod_{i=1}^{l} p_i^{m_i})/4$. Then, g is uniquely determined for GenDDH in the sense of Haitner *et al.* [HRS09], that is, for any two pairs $(x, w_1), (x, w_2)$ in the support of GenDDH, $g(w_1) = g(w_2)$. Indeed, GenDDH outputs a tuple of elements in QR_N, that is a tuple of elements of order $\phi(N)/4$, and thus two witnesses reduced modulo $\phi(N)/4$ are identical. Therefore, the black-box impossibility results of Haitner *et al.* for WH apply to any NI for $\mathcal{R}_{\mathsf{DDH}}$ with respect to the distribution GenDDH. In Section 1.3.4 we discuss how and why our results bypass this and similar black-box impossibility results. $\triangle$

Definition 31 The factoring oracle FactO takes as input N and outputs the list of all prime factors of N. $\triangle$

Assumption 5 Let GenSG be the following PPT algorithm. The algorithm GenSG, on input the security parameter 1^λ, computes $(N, p, q, g) \leftarrow \mathsf{GenRSA}(1^\lambda)$, and $u = g^w$, with $w \leftarrow \mathbb{Z}^\star_{\phi(N)}$, and outputs statement $x \stackrel{\triangle}{=} (N, g, u)$ and witness (w, p, q). The assumption states that $\mathcal{R}_{\mathsf{DDH}}$ and $\mathcal{R}_{\mathsf{SG}}$ are hard with respect to GenSG and FactO. Thus, FactO is a legitimate oracle for extraction for both the polynomial-time relations $\mathcal{R}_{\mathsf{DDH}}$ and $\mathcal{R}_{\mathsf{SG}}$, that is $\mathsf{FactO} \in \mathsf{LegOrHR}^{\mathcal{R}_{\mathsf{DDH}}}$ and $\mathsf{FactO} \in \mathsf{LegOrHR}^{\mathcal{R}_{\mathsf{SG}}}$. $\triangle$

We assume that the reader is familiar with the notion of cryptographic games.

Assumption 6 Let $\mathsf{ElGamal} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ be the encryption scheme as in definition 26 and DHInvO the oracle as in Definition 30. The assumption holds if no nuPPT adversary $\mathsf{Adv}^{\mathsf{DHInvO}} = \{\mathsf{Adv}^{\mathsf{DHInvO}}_\lambda\}_{\lambda>0}$ with access to DHInvO can win with non-negligible advantage in λ in the following game between Adv and a challenge $\mathcal{C}$. For security parameter λ, the game is the following.

- **Setup Phase.** The challenger $\mathcal{C}$ picks a pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives pk to Adv_λ which outputs two pair of messages m_0 and m_1 under the constraint that $m_0, m_1 \in \{0, 1\}$.
- **Challenge Phase.** The challenger picks a random bit $b \leftarrow \{0, 1\}$ and gives to Adv_λ a ciphertext ct encrypting m_b.
- **Winning Condition.** Adv_λ outputs a bit b' and wins iff $b' = b$ and DHInvO is never invoked on a input y such that $\mathsf{DHInvO}(y) = \perp$.

$\triangle$

Assumption 7 Let $\mathsf{ElGamal} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ be the encryption scheme as in Def. 26 and let NIDDH be the NIHZK system for $\mathcal{R}_{\mathsf{DDH}}$ of Section 3. The assumption holds if no nuPPT adversary $\mathsf{Adv} = \{\mathsf{Adv}_\lambda\}_{\lambda>0}$ can win with non-negligible advantage in λ in the following game between Adv and a challenge $\mathcal{C}$. For security parameter λ, the game is the following.

- **Setup Phase.** The challenger $\mathcal{C}$ picks a pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives pk to Adv_λ which outputs two pair of messages $(v_{0,0}, v_{0.1})$ and $(v_{1,0}, v_{1,1})$ under the constraint that $v_{0,0} + v_{0,1} = v_{1,0} + v_{1,1} = 0$.

- **Challenge Phase.** The challenger picks a random bit $b \leftarrow \{0,1\}$ and gives to Adv_λ two ciphertexts $\mathsf{ct}_0, \mathsf{ct}_1$ encrypting respectively $v_{b,0}, v_{b,1}$ along with a proof π of the fact that $\mathsf{ct} \stackrel{\triangle}{=} \mathsf{ct}_1 * \mathsf{ct}_2$ decrypts to 0 computed with NIDDH using statement ct and witness sk. (See Remark 11.)
- **Winning Condition.** Adv_λ outputs a bit b' and wins iff $b' = b$.

$\triangle$

Remark 11 Note that to compute a proof of decryption π, the issue of Section 1.2.3 has to be taken in account. Let us assume that solution 2 is used. Let $\mathsf{ct}_b \stackrel{\triangle}{=} (\mathsf{ct}_{b,1}, \mathsf{ct}_{b,2})$ for $b \in \{0,1\}$. Then a proof of correct decryption π would have to additionally include a proof of the fact that $\mathsf{ct}_{b,1}$ belongs to the subgroup generated by g, computed with NISG using statement $\mathsf{ct}_{b,1}$ and witness $(\mathsf{sk}, \mathsf{r})$, for randomness $r \stackrel{\triangle}{=} \mathbf{dlog}_g \mathsf{ct}_{b,1}$.

The previous assumption is a simplification in that we implicitly assume that the proof of correct decryption does not include such proof $\pi_{\mathsf{ct}_{b,1}}$. Furthermore, in a real e-voting application, the authority should add proofs that each individual ciphertext encrypts a bit; this can be done as outlined in Section 1.2.3. However, in the following we will show that such simplifications are without loss of generality. $\triangle$

The following assumption and its relation with the previous one are due to Geoffroy Couteau; it basically states that it is difficult to distinguish a random DH tuple for witness w from a random tuple given additionally another random DH tuple for the same witness w and access to DHInvO that can be only invoked on valid DH tuples.

Assumption 8 The assumption holds if no nuPPT adversary $\mathsf{Adv}^{\mathsf{DHInvO}(\cdot)} = \{\mathsf{Adv}_\lambda^{\mathsf{DHInvO}(\cdot)}\}_{\lambda>0}$ can win with non-negligible advantage in λ in the following game between $\mathsf{Adv}^{\mathsf{DHInvO}(\cdot)}$ and a challenge $\mathcal{C}$. For security parameter λ, the game is the following.

The challenger $\mathcal{C}$ picks an instance $(N, p, q, g) \leftarrow \mathsf{GenRSA}(1^\lambda)$, random values $w, a \leftarrow \mathbb{Z}^\star_{\phi(N)}, h, h', Z_1 \leftarrow \mathsf{QR}_N$, a random bit $b \in \{0,1\}$, sets $A = g^a, Z' = h'^w, Z_0 = A^w$ and runs $\mathsf{Adv}_\lambda^{O(\cdot)}$ on input $(N, g, h, A, Z_b, h', Z')$.

Adv_λ outputs a bit b' and wins iff $b' = b$ and Adv_λ never queried the oracle on an input y such that $\mathsf{DHInvO}(y) = \perp$. $\triangle$

Lemma 11 If there exists an adversary $\mathsf{Adv}^{\mathsf{DHInvO}(\cdot)}$ breaking Assumption 7 with advantage ϵ, there exists an adversary $\mathcal{B}^{\mathsf{DHInvO}(\cdot)}$ breaking Assumption 8 with advantage 2ϵ. $\triangle$

Proof. We consider the following series of hybrid experiments.

- H_0. Hybrid H_0 is identical to the game of Assumption 8 except that the bit $b = 0$ instead of being chosen randomly.
- H_1. Hybrid H_1 is identical to H_0 except that $v_{0,0}$ is a random message in $\mathbb{Z}_{\phi(N)}$ and $v_{0,1} = -v_{0,0}$.

– H_0. Hybrid H_0 is identical to the game of Assumption 8 except that the bit $b = 1$ instead of being chosen randomly.

W.l.o.g., we will show that if Adv can distinguish H_0 from H_1 with advantage ϵ, there exists an adversary $\mathcal{B}$ breaking Assumption 8 with advantage ϵ. This will prove the lemma.

The adversary $\mathcal{B}$ receives as input a tuple (g, h, A, Z, h', Z') and works as follows. The adversary $\mathcal{B}$ runs Adv on input the public key (N, g, h) and gets from Adv two pairs of messages $(v_{0,0}, v_{0.1})$ and $(v_{1,0}, v_{1,1})$ such that $v_{0,0} + v_{0,1} = v_{1,0} + v_{1,1} = 0$ (if this is not the case $\mathcal{B}$ aborts outputting an arbitrary bit).

The adversary $\mathcal{B}$ sets $\mathsf{ct}_0 = (A, Z \cdot g^{v_{0,0}})$, $\mathsf{ct}_1 = (h' \cdot A^{-1}, Z' \cdot Z^{-1} \cdot g^{-v_{0,0}})$. It is easy to verify that the "product" of ct_0 and ct_1 is the ciphertext (h', Z') encrypting 0. Therefore, $\mathcal{B}$ invokes the oracle DHInvO on input (N, g, h, h', Z') and returns its output to Adv as proof π of the fact that (h', Z') encrypts 0. By definition of the oracle, π has the right distribution in both the experiments.

(Here, it is where the simplifications discussed in Remark 11 has to be dealt with. In the case of using the solution 2 outlined in Section 1.2.3, the proof of correct decryption has also to include a proof that h' belongs to the subgroup generated by g, but such proof can be simulated via the oracle DHInvO, so the reduction would go through. Furthermore, in a real e-voting application, the authority should add proofs to guarantee that each individual ciphertext encrypts a bit (a valid vote); this can be done as outlined in Section 1.2.3. The simulator would have to simulate such proofs but such proofs can be computed via the oracle DHInvO, so the reduction would go through in this case as well.)

Notice that if $Z = A^w$, ct_0 encrypts $v_{0,0}$ and ct_1 encrypts $-v_{0,0}$, that is $\mathcal{B}$ simulated to Adv the experiment H_0. On the other hand, if Z is a random element in QR_N, ct_0 encrypts a random message $v_{0,0}$ and ct_1 encrypts $-v_{0,0}$, that is $\mathcal{B}$ simulated to Adv the experiment H_1. This concludes the proof. $\triangle$

Remark 12 Assumption 7 is also equivalent (the reduction goes both ways) to a variant of Assumption 8 in which the adversary is restricted to call the oracle DHInvO only a certain bounded number of times. $\triangle$

3 Our HZKPoK proofs for subgroup membership and DH well-formedness

In this Section we present our main HZPoKs for the relations $\mathcal{R}_{\mathsf{SG}}$ (cf. Def. 29) and $\mathcal{R}_{\mathsf{DDH}}$ (cf. Def. 28).

3.1 HZKPoK for $\mathcal{R}_{\mathsf{SG}}$

We firstly construct a HZKPoK for polynomial-time relation $\mathcal{R}_{\mathsf{SG}}$ that will be used as building block in the HZKPoK for $\mathcal{R}_{\mathsf{DDH}}$.

Construction 1 The NI system $\mathsf{NISG} = (\mathsf{ProveSG}, \mathsf{VerifySG})$ for polynomial-time relation $\mathcal{R}_{\mathsf{SG}}$ (cf. Def. 29) consists of the following algorithms.

Note: In the following, we let $s \triangleq |N|$. We will later show how to optimize the parameter s.

$\mathsf{ProveSG}$, on inputs statement (N, g, u) and witness $(w, [p_i, m_i]_{i=1}^{l})$ for $\mathcal{R}_{\mathsf{SG}}$, computes the following proof. We assume w to be the integer $< \mathrm{ord}(g)$ such that $u = g^w$. If this is not the case, the prover finds the value w' with such property and executes the below protocol with witness w'. We skip this detail in the algorithm description.

Algorithm $\mathsf{ProveSG}$:
Inputs: statement (N, g, u) and witness $(w, [p_i, m_i]_{i=1}^{l})$ for $\mathcal{R}_{\mathsf{SG}}$.

- **For each** $i \in [s]$, **do**
  - **Set** $r_i \leftarrow \mathbb{Z}^{\star}_{\phi(N)}, z_i = r_i + w \mod \phi(N)$ under the constraint that $r_i^{-1} \mod \phi(N)$ be prime, $(r_i + w) \mod \phi(N) \in \mathbb{Z}^{\star}_{\phi(N)}$, $z_i \in \mathbb{Z}^{\star}_{\phi(N)}$ and $z_i^{-1} \mod \phi(N)$ be prime.
  - **Set** $r'_i = r_i^{-1} \mod \phi(N), z'_i = z_i^{-1} \mod \phi(N)$.
  - **Set** $R_i = g^{r_i}$.
- **endFor**
- **Output** $(R_i, r'_i, z'_i)_{i \in [s]}$.

$\mathsf{VerifySG}$, on inputs statement (N, g, u) and proof $(R_i, r'_i, z'_i)_{i \in [s]}$, outputs a binary decision, 0 to denote rejection and 1 to denote acceptance.

Algorithm $\mathsf{VerifySG}$:
Inputs: statement (N, g, u) and proof $(R_i, r'_i, z'_i)_{i \in [s]}$.

1. **If** $g^{\prod_{i \in [s]} r'_i \cdot z'_i} = 1$ **then Return** 0.
2. **For each** $i \in [s]$, **do**
   (a) **If** r'_i OR z'_i is not prime **then Return** 0.
   (b) **If** $R_i^{r'_i} \neq g$ **then Return** 0.
   (c) **Set** $H_i = R_i \cdot u$.
   (d) **If** $H_i^{z'_i} \neq g$ **then Return** 0.
3. **endFor**
4. **If** $\exists i, j \in [s], i \neq j, r'_i = r'_j$ OR $z'_i = z'_j$ **then Return** 0.
5. **If** $\exists i, j \in [s], r'_i = z'_i$ **then Return** 0.
6. **Output** 1.

$\triangle$

Theorem 12 The NI system $\mathsf{NISG} = (\mathsf{ProveSG}, \mathsf{VerifySG})$ for polynomial-time relation $\mathcal{R}_{\mathsf{SG}}$ of Construction 1 is complete and perfectly sound. $\triangle$

Proof. The proof for completeness is trivial; see Remark 9.

Let us analyze soundness. Suppose that $\mathsf{VerifySG}$, on inputs statement (N, g, u) and proof $(R_i, r'_i, z'_i)_{i \in [s]}$, outputs 1 (i.e., it accepts the proof). We will prove that

$u = g^w$ for some $w < \phi(N)$ (the order of $\mathbb{Z}_N^\star$) and thus $\mathcal{R}_{\mathsf{SG}}((N, g, u), (w, [p_i, m_i]_{i=1}^l)) = 1$, with $N = \prod_{i=1}^l p_i^{m_i}$.

We firstly argue there exists at least one index $j \in [s]$ such that $r_j \cdot z'_j$ is co-prime with $\phi(N)$. Suppose towards a contradiction this to be false, that is, for each $i \in [s]$, $r'_i \cdot z'_i$ has a common factor with $\phi(N)$. By check 2.a, for each $i \in [s], r'_i, z'_i$ are primes and, by checks 4 and 5, the primes $r'_1, \ldots, r'_s, z'_1, \ldots, z'_s$ are all pairwise different. Then, since $s = |N|$, the product $t \stackrel{\triangle}{=} \prod_{i \in [s]} r'_i \cdot z'_i$ has to be a multiple of $\phi(N)$. Since $\phi(N)$ is the order of $\mathbb{Z}_N^\star$ and the order of each group element divides the order of the group, we have that $g^t = 1$ and hence in check 1 $\mathsf{VerifySG}$ refuses the proof, contradicting the hypothesis.

Let $j \in [s]$ be such that $r'_j \cdot z'_j$ is co-prime with $\phi(N)$. Then r'_j is co-prime with $\phi(N)$ as well. Check 2.b implies $R_j^{r'_j} = g$. Being r'_j co-prime with $\phi(N)$, it has an inverse modulo $\phi(N)$. Let $y \stackrel{\triangle}{=} r_j'^{-1} \mod \phi(N)$. Then, powering both sides of the equation $R_j^{r'_j} = g$ to y, we have $R_j = g^y$. Analogously, checks 1, 2.a, 2.d, 4 and 5 imply $H_j = g^{y_2}$ for some $y_2 < \phi(N)$.

Then, by check 2.c, $H_j = R_j \cdot u$ and this implies $u = H_j \cdot R_j^{-1} = g^{y_2} \cdot g^{-y} = g^{y_2 - y}$, as we had to prove. $\triangle$

Theorem 13 Let DHInvO be the oracle of Def. 30. If Assumption 2 holds, then the NI system $\mathsf{NISG} = (\mathsf{ProveSG}, \mathsf{VerifySG})$ of Construction 1 is DHInvO-HZK for polynomial-time relation $\mathcal{R}_{\mathsf{SG}}$. $\triangle$

Proof. By Assumption 2, DHInvO is a legitimate oracle for polynomial-time relation $\mathcal{R}_{\mathsf{SG}}$. What is left to prove is to show a PPT simulator algorithm for NISG satisfying (perfect) HZK. Consider the following simulator $\mathsf{SimSG}^{\mathsf{DHInvO}(\cdot)}$ with oracle access to $\mathsf{DHInvO}(\cdot)$. The simulator takes as input a statement (N, g, u) and computes a simulated proof as follows with the help of the oracle DHInvO.

Algorithm SimSG:
Inputs: statement (N, g, u).
Oracle: DHInvO.

- **For each** $i \in [s]$, **do**
  - $(R_i, Y_i, r'_i, z'_i) = \mathsf{DHInvO}(N, g, g, u, u)$.
- **endFor**
- **Output** $(R_i, r'_i, z'_i)_{i \in [s]}$.

By the definition of DHInvO, it is easy to see that for each $i \in [s]$, the tuple (R_i, r'_i, z'_i) in the output of the oracle on input (N, g, u, g, u) has the same distribution as a tuple (R_i, r'_i, z'_i) output by the prover; see Remark 9. Then, the output of $\mathsf{SimSG}^{\mathsf{DHInvO}(\cdot)}(N, g, u)$ is distributed identically to the output of $\mathsf{ProveSG}((N, g, u), (w, p, q))$, where $N = p \cdot q$ and $u = g^w$. $\triangle$

Theorem 14 Let FactO be the oracle of Def. 31. If Assumption 5 holds, then the NI system $\mathsf{NISG} = (\mathsf{ProveSG}, \mathsf{VerifySG})$ of Construction 1 is FactO-HPoK for polynomial-time relation $\mathcal{R}_{\mathsf{SG}}$. $\triangle$

Proof. By Assumption 5, $\mathcal{R}_{\mathsf{SG}}$ is hard with respect to FactO, that is $\mathsf{FactO} \in \mathsf{LegOrHR}^{\mathcal{R}_{\mathsf{SG}}}$. What is left to prove is to show a PPT extractor $\mathsf{ExtSG}^{\mathsf{FactO}(\cdot)}$ with oracle access to $\mathsf{FactO}(\cdot)$ such that the following holds: for any strings $x, \pi \in \{0,1\}^\star$, if $\mathsf{VerifySG}(x,\pi) = 1$ then $\mathrm{Prob}[(x,w) \in \mathcal{R}_{\mathsf{SG}}(x,w) |\, w \leftarrow \mathsf{ExtSG}^{\mathsf{FactO}(\cdot)}(x,\pi)] = 1$.

Consider the following extractor ExtSG with oracle access to FactO. The extractor ExtSG takes as input a statement (N,g,u) and a proof $(R_i, r'_i, z'_i)_{i\in[s]}$ and computes what follows. ExtSG invokes the oracle to factorize N and gets its factorization $[p_i, m_i]_{i=1}^l$ from which it can compute $\phi(N)$, the order of $\mathbb{Z}_N^\star$. By the analysis of the soundness (cf. Thm. 12), if $\mathsf{VerifySG}(x,\pi)$ then there exist values $y, y_2 \in \mathbb{Z}^\star_{\phi(N)}$ (i.e., integers in $\mathbb{Z}_{\phi(N)}$ that are invertible modulo $\phi(N)$) such that $u = g^{y-y_2}$.

Using $\phi(N)$, one can invert y and y_2 and compute $w = y - y_2 \mod \phi(N)$ that, along with the factorization, forms a valid witness for $\mathcal{R}_{\mathsf{SG}}$. $\triangle$

3.2 HZKPoK for $\mathcal{R}_{\mathsf{DDH}}$

Construction 2 The NI system $\mathsf{NIDDH} = (\mathsf{ProveDDH}, \mathsf{VerifyDDH})$ for polynomial-time relation $\mathcal{R}_{\mathsf{DDH}}$ (cf. Def. 28) consists of the following algorithms.

Note: In the following, we let $s \stackrel{\triangle}{=} |N|$. We will later show how to optimize the parameter s.

Let $\mathsf{NISG} = (\mathsf{ProveSG}, \mathsf{VerifySG})$ be the NI for polynomial-time relation $\mathcal{R}_{\mathsf{SG}}$ of Construction 1. $\mathsf{ProveDDH}$, on inputs statement (N,g,h,u,v) and witness $(w, [p_i, m_i]_{i=1}^l)$ for $\mathcal{R}_{\mathsf{DDH}}$, computes the following proof. We assume w to be the integer $< \mathrm{ord}(g)$ such that $u = g^w, v = h^w$. If this is not the case, the prover finds the value w' with such property and executes the below protocol with witness w'. We skip this detail in the algorithm description.

Algorithm $\mathsf{ProveDDH}$:
Inputs: statement (N,g,h,u,v) and witness $(w, [p_i, m_i]_{i=1}^l)$ for $\mathcal{R}_{\mathsf{DDH}}$.

- **For each** $i \in [s]$, **do**
  - **Set** $r_i \leftarrow \mathbb{Z}^\star_{\phi(N)}, z_i = r_i + w \mod \phi(N)$ under the constraint that $r_i^{-1} \mod \phi(N)$ be prime, $(r_i + w) \mod \phi(N) \in \mathbb{Z}^\star_{\phi(N)}$, $z_i \in \mathbb{Z}^\star_{\phi(N)}$ and $z_i^{-1} \mod \phi(N)$ be prime.
  - **Set** $r'_i = r_i^{-1} \mod \phi(N), z'_i = z_i^{-1} \mod \phi(N)$.
  - **Set** $X_i = g^{r_i}, Y_i = h^{r_i}$.
- **endFor**
- **Set** $\pi_u \leftarrow \mathsf{ProveSG}((N,g,u),(w,p,q))$ and $\pi_v \leftarrow \mathsf{ProveSG}((N,h,v),(w,p,q))$.
- **Output** $(\pi_u, \pi_v, X_i, Y_i, r'_i, z'_i)_{i\in[s]}$.

$\mathsf{VerifyDDH}$, on inputs statement (N,g,h,u,v) and proof $(\pi_u, \pi_v, (X_i, Y_i, r'_i, z'_i)_{i\in[s]})$, outputs a binary decision, 0 to denote rejection and 1 to denote acceptance.

Algorithm $\mathsf{VerifyDDH}$:
Inputs: statement (N, g, h, u, v) and proof $(\pi_u, \pi_v, (X_i, Y_i, r'_i, z'_i)_{i\in[s]})$.

1. **If** $g^{\prod_{i\in[s]} r'_i \cdot z'_i} = 1$ **then Return** 0.
2. **For each** $i \in [s]$, **do**
   (a) **If** r'_i OR z'_i is not prime, **then Return** 0.
   (b) **If** $X_i^{r'_i} \neq g \vee Y_i^{r'_i} \neq h$ **then Return** 0.
   (c) **Set** $H_i = X_i \cdot u, Z_i = Y_i \cdot v$.
   (d) **If** $H_i^{z'_i} \neq g \vee Z_i^{z'_i} \neq h$ **then Return** 0.
3. **endFor**
4. **If** $\mathsf{VerifySG}((N, g, u), \pi_u) = 0 \vee \mathsf{VerifySG}((N, h, v), \pi_v) = 0$ **then Return** 0.
5. **If** $\exists i, j \in [s], i \neq j, r'_i = r'_j$ **then Return** 0.
6. **If** $\exists i, j \in [s], r'_i = z'_i$ **then Return** 0.
7. **Output** 1.

$\triangle$

Theorem 15 The NI system $\mathsf{NIDDH} = (\mathsf{ProveDDH}, \mathsf{VerifyDDH})$ for polynomial-time relation $\mathcal{R}_{\mathsf{DDH}}$ of Construction 2 is complete and perfectly sound. $\triangle$

Proof. The proof for completeness is trivial; see Remark 9.

Let us analyze soundness. Suppose that $\mathsf{VerifyDDH}$, on inputs statement (N, g, h, u, v) and proof $(\pi_u, \pi_v, (X_i, Y_i, r'_i, z'_i)_{i\in[s]})$, outputs 1 (i.e., it accepts the proof). We will prove that $u = g^w, v = h^w$ for some $w < \phi(N)$ (the order of $\mathbb{Z}_N^\star$) and thus $\mathcal{R}_{\mathsf{DDH}}((N, g, h, u, v), (w, [p_i, m_i]_{i=1}^l)) = 1$, with $N = \prod_{i=1}^l p_i^{m_i}$.

We firstly argue there exists at least one index $j \in [s]$ such that $r'_j \cdot z'_i$ is co-prime with $\phi(N)$. Suppose towards a contradiction this to be false, that is, for each $i \in [s]$, $r'_i \cdot z'_i$ has a common factor with $\phi(N)$. By check 2.a, for each $i \in [s], r'_i, z'_i$ are primes and, by checks 5 and 6, the primes $r'_1, \ldots, r'_s, z'_1, \ldots, z'_s$ are all pairwise different. Then, since $s = |N|$, the product $t \stackrel{\triangle}{=} \prod_{i\in[s]} r'_i \cdot z'_i$ has to be a multiple of $\phi(N)$. Since $\phi(N)$ is the order of $\mathbb{Z}_N^\star$ and the order of each group element divides the order of the group, we have that $g^t = 1$ and hence in check 1 $\mathsf{VerifyDDH}$ refuses the proof, contradicting the hypothesis.

Let $j \in [s]$ be such that $r'_j \cdot z'_j$ is co-prime with $\phi(N)$. Check 2.b implies $X_j^{r'_j} = g$ (resp. $Y_j^{r'_j} = h$). Being r'_j co-prime with $\phi(N)$, it has an inverse modulo $\phi(N)$. Let $y \stackrel{\triangle}{=} r'^{-1}_j \mod \phi(N)$. Then, powering both sides of the equation $X_j^{r'_j} = g$ (resp. $Y_j^{r'_j} = h$) to y, we have $X_j = g^y$ (resp. $Y_j = h^y$). Analogously, checks 1, 2,a, 2.d, 5 and 6 imply $H_j = g^{y_2}$ (resp. $Z_j = h^{y_2}$) for some $y_2 < \phi(N)$.

By perfect soundness of NISG, check 4 guarantees that $u = g^{w_1}$ for some $w_1 < \phi(N)$ and $v = h^{w_2}$ for some $w_2 < \phi(N)$. Let k_1 (resp. k_2) be the order of g (resp. h). Then, by step 2.c, $H_j = X_j \cdot u$ and $Z_j = Y_j \cdot v$ and this implies:

$$g^{w_1} = u = H_j \cdot X_j^{-1} = g^{y_2} \cdot g^{-y} = g^{y_2 - y},$$

and

$$h^{w_2} = v = Z_j \cdot Y_j^{-1} = h^{y_2} \cdot h^{-y} = h^{y_2 - y}.$$

Taking the discrete logs, resp. in base g and h of the last two equations, we have $w_1 \equiv y_2 - y \mod k_1$ and $w_2 \equiv y_2 - y \mod k_2$. Recall that the system of equations

$$x \equiv a \mod m_1,$$

$$x \equiv b \mod m_2,$$

has a unique solution modulo $m_1 m_2 / \gcd(m_1, m_2)$ if $a \equiv b \mod \gcd(m_1, m_2)$. Thus, there exists integer x such that $x \equiv y_2 - y \mod k_1$ and $x \equiv y_2 - y \mod k_2$ and setting $w \stackrel{\triangle}{=} x \mod \phi(N)$ we have that

$$g^w = g^{w \mod k_1} = g^{(x \mod \phi(N)) \mod k_1} = g^{x \mod k_1} = g^{y_2 - y \mod k_1} = g^{w_1 \mod k_1} = g^{w_1} = u$$

and

$$h^w = h^{w \mod k_2} = h^{(x \mod \phi(N)) \mod k_2} = h^{x \mod k_2} = h^{y_2 - y \mod k_2} = g^{w_2 \mod k_2} = g^{w_2} = v,$$

as it was to prove. $\triangle$

Theorem 16 Let DHInvO be the oracle of Def. 30. If Assumption 2 holds, then the NI system $\mathsf{NIDDH} = (\mathsf{ProveDDH}, \mathsf{VerifyDDH})$ of Construction 2 is DHInvO-HZK for polynomial-time relation $\mathcal{R}_{\mathsf{DDH}}$. $\triangle$

Proof. By Assumption 2, DHInvO is a legitimate oracle for polynomial-time relation $\mathcal{R}_{\mathsf{DDH}}$. What is left to prove is to show a PPT simulator algorithm for NIDDH satisfying (perfect) HZK. Let $\mathsf{SimSG}^{\mathsf{DHInvO}(\cdot)}$ be the PPT simulator with oracle access to $\mathsf{DHInvO}(\cdot)$ of Theorem 13. Consider the following simulator $\mathsf{SimDDH}^{\mathsf{DHInvO}(\cdot)}$ with oracle access to $\mathsf{DHInvO}(\cdot)$. The simulator takes as input a statement (N, g, h, u, v) and computes a simulated proof as follows using SimSG and with the help of the oracle DHInvO.

Algorithm SimDDH:
Inputs: statement (N, g, h, u, v).
Oracle: DHInvO.

- **For each** $i \in [s]$, **do**
  - $(X_i, Y_i, r'_i, z'_i) = \mathsf{DHInvO}(N, g, h, u, v)$.
- **endFor**
- $\pi_u \leftarrow \mathsf{SimSG}^{\mathsf{DHInvO}(\cdot)}((N, g, u)), \pi_v \leftarrow \mathsf{SimSG}^{\mathsf{DHInvO}(\cdot)}(N, h, v)$.
- **Output** $(\pi_u, \pi_v, (X_i, Y_i, r'_i, z'_i)_{i \in [s]})$.

By the definition of DHInvO, it is easy to see that for each $i \in [s]$, the tuple (X_i, Y_i, r'_i, z'_i) output by the oracle on input (N, g, u, h, v) has the same distribution as a tuple (X_i, Y_i, r'_i, z'_i) output by the prover; see Remark 9. By Theorem

13, the output of $\mathsf{SimSG}^{\mathsf{DHInvO}(\cdot)}(N,g,u)$ (resp. $\mathsf{SimSG}^{\mathsf{DHInvO}(\cdot)}(N,h,v)$) is distributed identically to $\mathsf{ProveSG}((N,g,u),(w,[p_i,m_i]_{i=1}^l))$ (resp. $\mathsf{ProveSG}((N,h,v),(w,[p_i,m_i]_{i=1}^l)))$ for any witness $(w,[p_i,m_i]_{i=1}^l)$ for $\mathcal{R}_{\mathsf{SG}}$.

Then, the output of $\mathsf{SimDDH}^{\mathsf{DHInvO}(\cdot)}(N,g,h,u,v)$ is distributed identically to the output of $\mathsf{ProveDDH}((N,g,h,u,v),(w,[p_i,m_i]_{i=1}^l))$ for any witness $(w,[p_i,m_i]_{i=1}^l)$ for $\mathcal{R}_{\mathsf{DDH}}$. $\triangle$

Corollary 17 If Assumption 4 holds, then the NI system $\mathsf{NIDDH} = (\mathsf{ProveDDH}, \mathsf{VerifyDDH})$ of Construction 2 is WH (cf. Def. 22) for polynomial-time relation $\mathcal{R}_{\mathsf{DDH}}$. $\triangle$

Proof. By Assumption 4, $\mathcal{R}_{\mathsf{DDH}}$ is a hard relation with respect to oracle DHInvO and, by Theorem 16, NIDDH is DHInvO-HZK. By Lemma 5, if a polynomial-time relation $\mathcal{R}$ is hard with respect to an oracle O (cf. Def. 13), then an O-HZK NI $\mathsf{NI} \stackrel{\triangle}{=} (\mathcal{P},\mathcal{V})$ for $\mathcal{R}$ is O-WH. (Recall that O-WH implies WH.) Therefore, NIDDH is WH for $\mathcal{R}_{\mathsf{DDH}}$. $\triangle$

Remark 13 If Assumption 4 is weakened to require $\mathcal{R}_{\mathsf{DDH}}$ to be hard in the sense of Def. 12, that is without considering oracle-aided adversaries, Corollary 17 still holds. $\triangle$

Theorem 18 Let FactO be the oracle of Def. 31. If Assumption 5 holds, then the NI system $\mathsf{NIDDH} = (\mathsf{ProveDDH}, \mathsf{VerifyDDH})$ of Construction 2 is FactO-HPoK for polynomial-time relation $\mathcal{R}_{\mathsf{DDH}}$. $\triangle$

Proof. By Assumption 5, $\mathcal{R}_{\mathsf{DDH}}$ is hard with respect to FactO, that is $\mathsf{FactO} \in \mathsf{LegOrHR}^{\mathcal{R}_{\mathsf{DDH}}}$. What is left to prove is to show a PPT extractor $\mathsf{ExtDDH}^{\mathsf{FactO}(\cdot)}$ with oracle access to $\mathsf{FactO}(\cdot)$ such that the following holds: for any strings $x,\pi \in \{0,1\}^\star$, if $\mathsf{VerifyDDH}(x,\pi) = 1$ then $\mathrm{Prob}[(x,w) \in \mathcal{R}_{\mathsf{DDH}}(x,w) |\ w \leftarrow \mathsf{ExtDDH}^{\mathsf{FactO}(\cdot)}(x,\pi)] = 1$.

Let ExtDDH be the extractor with oracle access to FactO for NISG guaranteed by Theorem 14. Consider the following extractor ExtDDH with oracle access to FactO that uses ExtDDH. The extractor ExtDDH takes as input a statement (N,g,h,u,v) and a proof $(\pi_u,\pi_v,(X_i,Y_i,r_i',z_i')_{i\in[s]})$ and computes what follows. ExtDDH invokes ExtSG with input (N,g,u) and proof π_u simulating to ExtSG the oracle FactO. ExtDDH outputs the witness output by ExtSG. By Theorem 14, the witness $(w,[p_i,m_i]_{i=1}^l)$ output by ExtSG is such that $\mathcal{R}_{\mathsf{SG}}((N,g,u),(w,[p_i,m_i]_{i=1}^l)) = 1$, that is $u = g^w$. By perfect soundness of NIDDH, $h = g^w$ as well. Then, ExtDDH computes a valid witness. $\triangle$

3.3 Optimizations and a more efficient NIZK proof in the CRS model

Reducing the length of the proof. In the Constructions 1 and 2, the parameter s is set to $|N|$. What is actually needed for the perfect soundness to hold is just having s larger than the possible maximum number of prime factors of an integer N of λ bits.

There is a better upper bound on the maximum number s of prime factors of a composite number N of λ bits. Indeed, if the number of *different* prime factors of an integer N is $\geq s$, then $N \geq s!$ and thus, by Stirling's approximation, $N \geq \sqrt{2\pi} \cdot s^{s+1/2} \cdot e^{-s}$. Taking the discrete log in base 2 of both sides, we have that (1) $\lambda \geq 1.32 \cdot (\lfloor \log_2(s) \rfloor (s + 1/2) - \lfloor \log_2(e) \rfloor \cdot s) \geq 1.32 \cdot s \cdot (\lfloor \log_2(s) \rfloor - 2)$. This equation can be used to compute an integer s such that no integer of λ bits can have s different prime factors, that is to compute an integer s that represents a number of parallel repetitions sufficient to guarantee perfect soundness when the statement is with respect to a modulus N of λ bits. For instance, setting $\lambda = 1024$ (resp. $\lambda = 2048$) in (1), we see that $s = 156$ (resp. $s = 259$) repetitions are sufficient to guarantee perfect soundness.

We can also optimize the parameter s and reduce the length of the proof as follows. The verifier rejects if N has a prime factor of bit length $< k$. In this case, each factor of N has to be of bit length $\geq k$ and so s can be set to $\lceil |N|/k \rceil + 1$. As a consequence, we have a trade-off between the length of the proof and the computational power of the verifier.

NIZK proof in the CRS model. There are two sources of inefficiency in the NI NIDDH. Checking whether a number is co-prime with the group order induces the need for parallel repetitions, and additionally checking if an element belongs to the correct group. Can be NIDDH improved moving to the CRS model?

A possibility is to work in the group of signed quadratic residues modulo a Blum integer N [HK09] where one can efficiently check if a group element belongs to the group of signed quadratic residues. The properties of signed quadratic residues are guaranteed when -1 is not a quadratic residue and this is the case when the modulus N is setup honestly as a Blum integer. In the CRS model, the CRS can be set to be a pair consisting of a Blum integer N and a generator of the group of signed quadratic residues modulo N. Moreover, this can be done so that the generator have prime order. Therefore, having a CRS setup in that way, the need for parallel repetitions disappears and checking whether an element belong to the correct group is easy.

Even in the CRS model, to our knowledge there is no known efficient perfectly sound proof system for proving correct decryption of El Gamal ciphertexts over multiplicative groups.

The resulting proof system we obtain moving to the CRS model still seems to require oracle-aided simulation.

4 Conclusions

Since the introduction of zero-knowledge proofs [GMR89], the importance of removing coordination has been recognized as fundamental both from a theoretical point of view and for practical applications like e-voting that require universal verifiability. Unfortunately, one-message zero-knowledge proofs provably do not exist, so non-interactive zero-knowledge proof systems have been proposed subject to some limitations like the existence of a trusted player that setups a shared common reference string [DMP88] or in the so called random oracle model [BR93] or assuming a known bound on the space of the verifier [DPY92].

In this work we have put forth proofs for non-trivial and useful cryptographic relations that (1) can be communicated in one-message and enjoy zero soundness error, that is they are proofs in the mathematical sense, (2) are efficient and (3) satisfy a new privacy notion that we call harmless zero-knowledge. In addition, we presented proofs with the properties (1) and (3) for general $\mathcal{NP}$ relations.

Harmless zero-knowledge is rooted in the simulation paradigm and represents a generalization of zero-knowledge in that it allows the simulator to have access to an oracle relative to which the language is still hard to decide. Essentially, we exploit the fact that in several real-world protocols that use cryptographic proofs, we can assume adversaries to *not* have access to some trapdoor information (e.g., a secret key); restricted to this class of adversaries, a harmless zero-knowledge proof does not leak knowledge that enables the adversary to attack a larger system in which the proof is employed.

For example, taking advantage of the fact that adversaries against the privacy of an encryption scheme do not have access to the secret-key, we can construct a perfectly sound one-message harmless zero-knowledge proof of correct decryption of El Gamal ciphertexts that is not based on any trust assumption.[18] Instead, the soundness of proofs of correct decryption obtained via the FS transform is completely breakable by adversaries discovering a trapdoor, e.g., in the hash function used to instantiate the random oracle. The drawback is that the privacy of the application obtained using our harmless zero-knowledge proof of correct decryption (for instance, the task of distinguishing whether two ciphertexts encrypt $(1, -1)$ or $(-1, 1)$ given a proof that the product ciphertext decrypts to 0) is based on a less-studied oracle-based assumption. Therefore, one has trade a qualitatively different assumption used for the privacy for removing trust and computational assumptions used for the verifiability.

Contrast this state of affairs with other "implementations" of the simulation paradigm. For instance, the variant of zero-knowledge secure in the Universal Composability model [Can01] is stronger and offers more general composability guarantees than standard zero-knowledge but this comes at the cost of limiting the achievability only to the CRS, RO or restricted models. Analogously, zero-knowledge is stronger and can be composed with a larger class of protocols than harmless zero-knowledge but this comes at the cost of sacrificing non-interactivity and zero soundness error. For specific applications, harmless zero-knowledge may be useful, usable and secure.

[18] Due to the limitations highlighted in Section 1.2.3, we have to use either our NI for $\mathcal{NP}$ or to use our efficient proof but assuming the pair (N, g) to be correctly generated (we additionally sketch an alternative solution that requires a change in the encryption scheme). To our knowledge, it was not known how to construct a one-message perfectly sound proof for correct decryption satisfying a non-trivial notion of privacy beyond WI useful and usable in security proofs.

More in detail, an oracle O induces a set of bad pairs (f, Adv) (see Section 1.3.5) that, roughly speaking, represent the "non-simulatable" adversaries Adv that, given a proof of some statement x, can compute $f(x, w)$ for some witness w to x. Such an adversary may be, for instance, the one that is able to compute the witness to a DH tuple X over $\mathbb{Z}_N^\star$ given a NIDDH proof for X and additionally the factorization of N as auxiliary input. In some applications like e-voting, we can safely assume adversaries against the e-voting scheme to not have access to such factorization (such adversaries only observe the public-key), thus in those particular applications such an "attack" is harmless. Therefore, analyzing the security of an O-HZK proof deployed in a larger protocol accounts to studying whether the bad pairs induced by O correspond to real attacks in the protocol.

Another way of thinking about about the limitations and power of O-HZK proofs is to look at the fact that O-HZK implies O-strong-WI (cf. Def. 23 and discussion in Section 1.3.9): some distributions of statements X and Y are not computationally indistinguishable by distinguishers with access to the oracle and thus may not hold that (X, π) is computationally indistinguishable (by distinguishers without access to the oracle) from (Y, π). In the analysis of a security protocol, this fact can be not necessary to prove the security, and so this apparent limitation turns out to be harmless (moreover, without introducing the oracle in the analysis of the security, we could be unable to prove the security). That is, O-strong-WI punctures out a set of "bad" distributions, the ones that are indistinguishable by distinguishers without access to the oracle but distinguishable by distinguishers with access to the oracle and are non-simulatable (by simulators without access to the oracle). If such bad distributions never occur in the analysis of a protocol using an O-HZK proof, then O-HZK suffices to prove the security of that particular protocol (and in some applications it may be necessary to consider HZK since non-interactive ZK proofs do not exist). In view of these considerations, for future work it would be interesting to characterize O-HZK in terms of epistemic logic [HPR09].

Our work is far from being a comprehensive or completely satisfactory study of alternative models to zero-knowledge proofs compatible with zero verification error and pure non-interactivity and introduces novel computational assumptions. We hope, however, it can shed light on this intriguing possibility.

Acknowledgments

I dedicate this work to the memory of my father, Salvatore Iovino (1951-2019), and to all people with ALS and their relatives and friends.

Part of this research has been done while the author was at the University of Luxembourg supported by an FNR CORE-Junior grant of the Luxembourg National research fund (project no. C16/IS/11299247).

References

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

ABOR00. William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, pages 463–474, 2000.

Adi08. Ben Adida. Helios: Web-based open-audit voting. In *USENIX Security Symposium*, volume 17, pages 335–348, 2008.

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

AH87. William Aiello and Johan Håstad. Perfect zero-knowledge languages can be recognized in two rounds. In *28th Annual Symposium on Foundations of Computer Science*, pages 439–448. IEEE Computer Society Press, October 1987.

AKS04. Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, September 2004.

ARU14. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th Annual Symposium on Foundations of Computer Science*, pages 474–483. IEEE Computer Society Press, October 2014.

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

BBB+18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334. IEEE, 2018.

BCC88. Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of computer and system sciences*, 37(2):156–189, 1988.

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

BCP03. Emmanuel Bresson, Dario Catalano, and David Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In Chi-Sung Laih, editor, *Advances in Cryptology – ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 37–54. Springer, November / December 2003.

BDPA11. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. The Keccak reference, 2011. `http://keccak.noekeon.org/`.

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

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

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

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

BG93. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer, August 1993.

BGI$^+$01. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, August 2001.

BGJS16. Saikrishna Badrinarayanan, Vipul Goyal, Aayush Jain, and Amit Sahai. Verifiable functional encryption. In *Advances in Cryptology – ASIACRYPT 2016*, pages 557–587, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

BHZ87. R. B. Boppana, J. Hastad, and S. Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, May 1987.

BKP19. Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 1091–1102, New York, NY, USA, 2019. ACM.

BL02. Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. In *34th Annual ACM Symposium on Theory of Computing*, pages 484–493. ACM Press, May 2002.

Blu86. Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pages 444–451, 1986.

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

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

BN06. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and

Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 390–399. ACM Press, October / November 2006.

BNPS02. Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In Paul F. Syverson, editor, *FC 2001: 5th International Conference on Financial Cryptography*, volume 2339 of *Lecture Notes in Computer Science*, pages 319–338. Springer, February 2002.

BOV03. Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315. Springer, August 2003.

BP02. Mihir Bellare and Adriana Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 162–177. Springer, August 2002.

BP04. Boaz Barak and Rafael Pass. On the possibility of one-message weak zero-knowledge. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 121–132. Springer, February 2004.

BP15. Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *Theory of Cryptography Conference*, pages 401–427. Springer, 2015.

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

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

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

BSJS96. E. Bach, J.O. Shallit, S. Jeffrey, and P.J. Shallit. *Algorithmic Number Theory: Efficient algorithms.* Algorithmic Number Theory. MIT Press, 1996.

BY93. Mihir Bellare and Moti Yung. Certifying cryptographic tools: The case of trapdoor permutations. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 442–460. Springer, August 1993.

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

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Com-*

puter Science, pages 136–145. IEEE Computer Society Press, October 2001.

CCD88. David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (abstract) (informal contribution). In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, page 462. Springer, August 1988.

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

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

CGGM00. Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *32nd Annual ACM Symposium on Theory of Computing*, pages 235–244. ACM Press, May 2000.

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

CGS97. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 103–118. Springer, May 1997.

Cha81. David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

CKPR01. Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires omega (log n) rounds. In *33rd Annual ACM Symposium on Theory of Computing*, pages 570–579. ACM Press, July 2001.

CL15. Guilhem Castagnos and Fabien Laguillaumie. Linearly homomorphic encryption from DDH. In *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, pages 487–505, 2015.

CLMP12. Kai-Min Chung, Edward Lui, Mohammad Mahmoody, and Rafael Pass. Unprovable security of two-message zero knowledge. Cryptology ePrint Archive, Report 2012/711, 2012. `http://eprint.iacr.org/2012/711`.

CLW18. Ran Canetti, Alex Lombardi, and Daniel Wichs. Fiat-Shamir: From practice to theory, part ii (NIZK and correlation intractability from circular-secure FHE). Cryptology ePrint Archive, Report 2018/1248, 2018. `https://eprint.iacr.org/2018/1248`.

CMFP+10. Benoît Chevallier-Mames, Pierre-Alain Fouque, David Pointcheval, Julien Stern, and Jacques Traoré. On some incompatible properties of voting schemes. In *Towards Trustworthy Elections*, pages 191–199. Springer, 2010.

Coo71. Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

CP93. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume

740 of *Lecture Notes in Computer Science*, pages 89–105. Springer, August 1993.

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

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

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

CS02. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, April / May 2002.

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

CS13. Véronique Cortier and Ben Smyth. Attacking and fixing helios: An analysis of ballot secrecy. *Journal of Computer Security*, 21(1):89–148, 2013.

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

DBB+15. Gaby G Dagher, Benedikt Bünz, Joseph Bonneau, Jeremy Clark, and Dan Boneh. Provisions: Privacy-preserving proofs of solvency for bitcoin exchanges. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 720–731. ACM, 2015.

DDN91. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *23rd Annual ACM Symposium on Theory of Computing*, pages 542–552. ACM Press, May 1991.

DDO+01. Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pages 566–598, 2001.

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

DK02. Ivan Damgård and Maciej Koprowski. Generic lower bounds for root extraction and signature schemes in general groups. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 256–271. Springer, April / May 2002.

DK18. Apoorvaa Deshpande and Yael Kalai. Proofs of ignorance and applications to 2-message witness hiding. Cryptology ePrint Archive, Report 2018/896, 2018. `https://eprint.iacr.org/2018/896`.

DMP88. Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, pages 52–72. Springer, August 1988.

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

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

DP92. Alfredo De Santis and Giuseppe Persiano. Zero-knowledge proofs of knowledge without interaction (extended abstract). In *33rd Annual Symposium on Foundations of Computer Science*, pages 427–436. IEEE Computer Society Press, October 1992.

DPY92. Alfredo De Santis, Giuseppe Persiano, and Moti Yung. One-message statistical zero-knowledge proofs and space-bounded verifier. In *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings*, pages 28–40, 1992.

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

DS02. Cynthia Dwork and Larry J. Stockmeyer. 2-round zero knowledge and proof auditors. In *34th Annual ACM Symposium on Theory of Computing*, pages 322–331. ACM Press, May 2002.

DSYC18. Yi Deng, Xuyang Song, Jingyue Yu, and Yu Chen. On the security of classic protocols for unique witness relations. In *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II*, pages 589–615, 2018.

ES02. Edith Elkind and Amit Sahai. A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attack. Cryptology ePrint Archive, Report 2002/042, 2002. `http://eprint.iacr.org/2002/042`.

FGJ18. Nils Fleischhacker, Vipul Goyal, and Abhishek Jain. On the existence of three round zero-knowledge proofs. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, pages 3–33, 2018.

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

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

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

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

For87. Lance Fortnow. The complexity of perfect zero-knowledge (extended abstract). In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 204–209. ACM Press, May 1987.

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

FS90. Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd Annual ACM Symposium on Theory of Computing*, pages 416–426. ACM Press, May 1990.

FS01. Jun Furukawa and Kazue Sako. An efficient scheme for proving a shuffle. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 368–387. Springer, August 2001.

Gal02. Steven D. Galbraith. Elliptic curve Paillier schemes. *Journal of Cryptology*, 15(2):129–138, 2002.

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

GH98. Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Inf. Process. Lett.*, 67(4):205–214, 1998.

GK90. Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. In *International Colloquium on Automata, Languages, and Programming*, pages 268–282. Springer, 1990.

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

GK08. Vipul Goyal and Jonathan Katz. Universally composable multi-party computation with an unreliable common reference string. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 142–154. Springer, March 2008.

GL07a. Jens Groth and Steve Lu. A non-interactive shuffle with pairing based verifiability. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 51–67. Springer, December 2007.

GL07b. Jens Groth and Steve Lu. Verifiable shuffle of large size ciphertexts. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007: 10th International*

Conference on Theory and Practice of Public Key Cryptography, volume 4450 of *Lecture Notes in Computer Science*, pages 377–392. Springer, April 2007.

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

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

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304, 1985. Accessible at: `http://groups.csail.mit.edu/cis/crypto/classes/6.876/papers/gmr-ZK.pdf`.

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

GMW86. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th Annual Symposium on Foundations of Computer Science*, pages 174–187. IEEE Computer Society Press, October 1986.

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 218–229. ACM Press, May 1987.

GMW91. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

GO94. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

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

Gol04. Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004.

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

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

GOS12. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for non-interactive zero-knowledge. *Journal of the ACM (JACM)*, 59(3):11, 2012.

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

GQ88. Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *Advances in Cryptology – EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 123–128. Springer, May 1988.

Gro03. Jens Groth. A verifiable secret shuffle of homomorphic encryptions. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 145–160. Springer, January 2003.

Gro05a. Jens Groth. Cryptography in subgroups of zn. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 50–65. Springer, February 2005.

Gro05b. Jens Groth. Non-interactive zero-knowledge arguments for voting. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *ACNS 05: 3rd International Conference on Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 467–482. Springer, June 2005.

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

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 99–108. ACM Press, June 2011.

HK09. Dennis Hofheinz and Eike Kiltz. The group of signed quadratic residues and applications. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 637–653. Springer, August 2009.

HPR09. Joseph Y Halpern, Rafael Pass, and Vasumathi Raman. An epistemic characterization of zero knowledge. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 156–165. ACM, 2009.

HRS09. Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im)possibility of Arthur-Merlin witness hiding protocols. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 220–237. Springer, March 2009.

IMS12. Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. On efficient zero-knowledge PCPs. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 151–168. Springer, March 2012.

Jag12. Tibor Jager. *On black-box models of computation in cryptology.* PhD thesis, Ruhr University Bochum, 2012.

JCJ10. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Towards Trustworthy Elections*, pages 37–63. Springer, 2010.

JKKR17. Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In *Annual International Cryptology Conference*, pages 158–189. Springer, 2017.

Kal06. Yael Tauman Kalai. *Attacks on the Fiat-Shamir paradigm and program obfuscation*. PhD thesis, Massachusetts Institute of Technology, 2006.

Kar72. Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

Kat08. Jonathan Katz. Which languages have 4-round zero-knowledge proofs? In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 73–88. Springer, March 2008.

KKS18. Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 34–65. Springer, 2018.

KRR17. Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of fiat-shamir for proofs. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 224–251, 2017.

KS17. D. Khurana and A. Sahai. How to achieve non-malleability in one or two rounds. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 564–575, Oct 2017.

Lev84. Leonid A. Levin. Problems, complete in "average" instance. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pages 465–, New York, NY, USA, 1984. ACM.

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

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

Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12. Springer, December 2005.

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

McC88. Kevin S. McCurley. A key distribution system equivalent to factoring. *Journal of Cryptology*, 1(2):95–105, 1988.

MR95. Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.

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

MX13. Mohammad Mahmoody and David Xiao. Languages with efficient zero-knowledge PCPs are in SZK. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 297–314. Springer, March 2013.

Nao03. Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume

2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer, August 2003.

Nec94. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

Nef01. C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In *ACM CCS 01: 8th Conference on Computer and Communications Security*, pages 116–125. ACM Press, November 2001.

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

Pai99. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, May 1999.

Pas03a. Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 316–337. Springer, August 2003.

Pas03b. Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 160–176. Springer, May 2003.

Pas06a. Rafael Pass. Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on NP-hardness. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 13–pp. IEEE, 2006.

Pas06b. Rafael Pass. *A precise Computational Approach to Knowledge*. PhD thesis, Massachusetts Institute of Technology, 2006.

Pas11. Rafael Pass. Limits of provable security from standard assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 109–118. ACM Press, June 2011.

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

PIK94. Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. Efficient anonymous channel and all/nothing election scheme. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 248–259. Springer, May 1994.

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

PS04. Manoj Prabhakaran and Amit Sahai. New notions of security: Achieving universal composability without trusted setup. In László Babai, editor, *36th Annual ACM Symposium on Theory of Computing*, pages 242–251. ACM Press, June 2004.

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

PS19. Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. Cryptology ePrint Archive, Report 2019/158, 2019. `https://eprint.iacr.org/2019/158`.

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

Rab80. Michael O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory, 12(1)*, pages 128–138, 1980.

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

RS06. Peter Y. A. Ryan and S. A. Schneider. Prêt à voter with re-encryption mixes. Technical Report CS-TR-956, University of Newcastle, 2006.

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

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

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

Shm85. Zahava Shmuely. Composite diffie-hellman public-key generating schemes are hard to break. *Technical Repart No. 356, Computer Science Department, Technion-Israel Institute of Technology*, February 1985.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, May 1997.

SK94. Kazue Sako and Joe Kilian. Secure voting using partially compatible homomorphisms. In Yvo Desmedt, editor, *Advances in Cryptology – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 411–424. Springer, August 1994.

TW87. Martin Tompa and Heather Woll. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *28th Annual Symposium on Foundations of Computer Science*, pages 472–482. IEEE Computer Society Press, October 1987.

TW10. Björn Terelius and Douglas Wikström. Proofs of restricted shuffles. In Daniel J. Bernstein and Tanja Lange, editors, *AFRICACRYPT 10: 3rd International Conference on Cryptology in Africa*, volume 6055 of *Lecture Notes in Computer Science*, pages 100–113. Springer, May 2010.

Unr12. Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152. Springer, April 2012.

Unr15. Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *Annual International Conference on the The-*

ory and Applications of Cryptographic Techniques (TCC)*, pages 755–784. Springer, 2015.

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

Wee09. Hoeteck Wee. Zero knowledge in the random oracle model, revisited. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 417–434. Springer, December 2009.

Wik05. Douglas Wikström. A sender verifiable mix-net and a new proof of a shuffle. In Bimal K. Roy, editor, *Advances in Cryptology – ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 273–292. Springer, December 2005.

YZ12. Andrew C. Yao and Yunlei Zhao. Digital signatures from challenge-divided sigma-protocols. Cryptology ePrint Archive, Report 2012/001, 2012. `http://eprint.iacr.org/2012/001`.

A Analysis of Assumption 4 in a generic model

In Section 1.2.2, while sketching our new main proof systems, we highlighted potential attacks against simplified versions of them and patches to seemingly counter such attacks. In this section we aim at analyzing in more detail the security our main proof system NIDDH.

In this work we do not provide a comprehensive analysis of all the security properties we claim NIDDH enjoys except for completeness and soundness that we have already shown to hold unconditionally. However, in the following we will further examine Assumption 4 that essentially states that NIDDH is WH. Precisely, we will show that (a simplified version of) Assumption 4 holds in some variant of the generic group and ring model; this is a first step to argue the reasonableness of such assumption. We assume the reader to be familiar with the generic group and ring models [Nec94,Sho97,Mau05,DK02,Jag12].

Assumptions T, T' and T^F. For simplicity, we consider the following simplified version of Assumption 4 that we call hereafter Assumption T. An adversary against Assumption T receives as input the modulus N, a random DH tuple $(g, h, u \stackrel{\triangle}{=} g^w, v \stackrel{\triangle}{=} h^w) \in \mathsf{QR}_N^4$ for witness $w \in \mathbb{Z}^\star_{\phi(N)}$, a pair $(X = g^r, Y = h^r)$ for random $r \in \mathbb{Z}^\star_{\phi(N)}$, values $r' \stackrel{\triangle}{=} r^{-1} \mod \phi(N)$ and $z' \stackrel{\triangle}{=} (r+w)^{-1} \mod \phi(N))$. Hereafter we denote by $m \stackrel{\triangle}{=} \phi(N)/4$ the order of QR_N. The adversary wins the game if it outputs an integer $y \equiv w \mod m$. An adversary breaks the assumption if it wins with non-negligible probability.

Note that the DH tuple corresponds to the statement input to the adversarial prover, w corresponds to the witness to the statement and the remaining part of the input corresponds to the proof. Assumption T captures the essence of Assumption 4 while avoiding to introduce details (mainly the "parallel repetitions" and the constraint on the primality of the exponents) that do not benefit our analysis.

It is easy to see that Assumption T is equivalent to the following assumption T'. T' is identical to T except that the adversary does not get as input h, v, Y. Hereafter, we will thus analyze Assumption T'.

Assumption T^F is identical to Assumption T' except that the adversary wins if it outputs with non-negligible probability a factor of N. That is, the input to an adversary against T^F is identical to the input of an adversary against T' but the goal in T^F is to factor N rather than computing w. We say that Assumption T^F holds or is hard if no PPT adversary can break it.

Our variant of the generic group and ring models. The generic group and ring models (see [Jag12] for a comprehensive overview) are insufficient to model Assumption T'. The reason is that, not only the adversary may perform group operations starting from the elements g, u, X, but it also has available the elements r', z' that can be both subject to integer operations among them and used as exponents in an exponentiation with the previous set of elements. To exemplify, the adversary could compute $z_1 = r' + z', z_2 = z_1 * r'$ (where $+$ and $*$ are resp. addition and multiplication over the integers) and then $Z = X^{z_1} \cdot X^{z_2}$.

In our generic model, we consider two sets S and I resp. subsets of of the group $\mathbb{Z}_N^\star$ and of the integers. The generic attacks our model takes in account are the following:

- Two elements $g_1, g_2 \in S$ can be multiplied together via the group operation $\cdot$ to obtain $g = g_1 \cdot g_2$. The group operation $\cdot$ corresponds to the multiplication modulo N.
- Two elements $x, y \in I$ can be added and multiplied together over the integers to get another integer in I. We do not allow division but consider the attack in which two elements of I can added and multiplied together to get an element $z \equiv 1 \mod m$. This is a devastating attack because it allows to get a multiple of m (the order of the group that is equal to $\phi(N)/4$) and, by standard techniques, factorize N.

Therefore, in our model we assume the adversary gets as input the modulus N and has access to an oracle $\mathsf{GenericO}$ via which the adversary can perform the following operations:

- Group operations among elements of S.
- Addition, subtraction, multiplication of elements of I.
- Exponentiating an element of S to an element of I.

The result of the operation between two elements of I (resp. S) is a new element of I (resp. S), and the result of an exponentiation between an element of S and an element of I is a new element of S. That is, the sets are updated during the computation and we skip the details of their internal representation. The sets S and I are initialized according to the specific game played by the adversary, that is different computational assumptions stated in our generic model are parameterized by different sets S and I.

Moreover, we assume the adversary can issue the following queries:

- Test equality among two elements in S. The oracle is invoked on two elements $x, y \in S$ and returns 1 if and only if x and y are the same group element.
- Test equality among two elements in I. The oracle is invoked on two elements $x, y \in I$ and returns 1 if and only if x and y are equal as integers (we stress that the equality has to hold over the integers and *no* modular reduction is performed in the test).
- Test whether an element y of I equals $1 \mod m$ (recall that m is the order of the group that equals $\phi(N)/4$).

The latter test allows to take in account attacks in which the adversary is able to factorize by computing a multiple of $\phi(N)$. Observe that the latter query may be not needed because an adversary can power an element $g \in S$ to y and check that the result be equal to g itself. However, we keep such query for generality, for instance for assumptions in which the set S is empty.

Furthermore, we assume the set I to contain the integer 1 by means of which an arbitrary integer can be computed and the set S to contain the identity of the group. At the end of the game, the adversary can return some output string.

A computational assumption stated in our model is thus defined by the "initial inputs" S and I and some winning conditions indicating if the adversary won in an execution of the game and a function ϵ of the security parameter λ meaning that an adversary is considered successful in breaking the assumption if it wins in the game with probability $\geq \epsilon(\lambda)$. The winning conditions may depend on the output of the adversary and on on the sequence of oracle queries the adversary asks. We skip a precise formalization of the model; the interested reader can refer to Chapter 2 in [Jag12] and adapt that formalism to our case.

Note that in our generic model the adversary has no direct access to the integers in I. For instance, attacks not covered by our model include the ones in which the adversary can perform arbitrary operations among the elements in S. Since we are not aware of any attack in which the adversary can exploit the "exponents" except by computing a multiple of $\phi(N)$, such model seems reasonable.

Assumption T' is hard in our generic model. To state the hardness of a computational assumption in our generic model, we have to define its initial input (the sets S and I that are manipulated only via the oracle) and the winning conditions for the adversary playing the game against the challenger.

In Assumption T' in our generic model, the set S initially contains $1, g, u, X$ and the set I initially contains $1, r', z'$. An adversary playing the game of Assumption T' in our generic model gets N as input and can access via the oracle $\mathsf{GenericO}$ the elements in S and I in the way previously described. An adversary with access to $\mathsf{GenericO}$ wins the game of Assumption T' if one of the following winning conditions are satisfied:

1. The adversary successfully issues a query to test whether an element y of I is such that $y > 1, y \equiv 1 \mod m$.

2. During its computation, the oracle stores in its internal state an integer y such that $y \equiv w \mod m$.

We say that an adversary breaks Assumption T' if it wins the game with non-negligible probability. We say that Assumption T' (and similarly T) holds or is hard if no PPT adversary can break it.

Observe that it is crucial to include the latter two events in the winning conditions as the occurrence of these events corresponds to successful attacks against T'. Note that the winning condition 2 also covers attacks in which the adversary is able to effectively output w since such an adversary can be converted into another adversary that, once computed w, issues a sum query to 0.

We prove the following theorem.

Theorem 19 If Assumption T^F holds (in the standard model), then Assumption T' (and thus T) is hard in our generic model. $\triangle$

Proof (Sketch). We first show that if in the game of Assumption T' (that is played in our generic model), the first winning condition is satisfied with probability ϵ, then there exists an adversary $\mathcal{B}$ breaking (in the standard model) Assumption T^F with probability ϵ.

Indeed, an adversary $\mathcal{B}$ against T^F (in the standard model) can simulate to an adversary $\mathcal{A}$ the game of Assumption T' in our generic model by internally simulating the oracle.[19] More in detail, though $\mathcal{B}$ has just N (but not $\phi(N)$), in order to simulate a query to test whether an element $y \equiv 1 \mod m$, $\mathcal{B}$ can check whether $g^y = g$ and analogously can test whether an element $y \equiv w \mod m$ by testing whether $g^y = u$, and obviously can simulate integer operations between elements in I, group operations and exponentiations.

If a query to test whether an element y is such that $y > 1, y \equiv 1 \mod m$ issued to the oracle is successful, then it must be that $y - 1$ is a non-trivial multiple of $m = \phi(N)/4$, and so N can be factored by standard techniques.

Let W^T be the probability that Adv wins in the game of Assumption T in our generic model, W^{T^F} the probability that Adv wins in the game of Assumption T^F in the standard model and W_2^T the probability that *only* the second winning condition is satisfied in the game of Assumption T in our generic model.

(Hereafter, we will use the fact that an event A can be expressed as the event $C_1 \vee C_2$ in which $C_1 \stackrel{\triangle}{=} (A \ \wedge \ B)$ and $C_2 \stackrel{\triangle}{=} (A \ \wedge \ \neg B)$ are disjoint events, so $\mathrm{Prob}[A] = \mathrm{Prob}[C_1 \ \vee \ C_2] = \mathrm{Prob}[C_1] + \mathrm{Prob}[C_2] = \mathrm{Prob}[A|\ B] \cdot \mathrm{Prob}[B] + \mathrm{Prob}[A|\ \neg B] \cdot \mathrm{Prob}[\neg B] \leq \mathrm{Prob}[B] + \mathrm{Prob}[A|\ \neg B]$.)

We thus proved that (1) $W \leq W^{T_F} + W_2^T$.

We define the hybrid experiment H_1 to be identical to the experiment of Assumption T' in our generic model except that the first winning condition is missing, that is the adversary wins if and only if the second winning condition is satisfied. Let W^{H_1} be the probability that Adv wins in the experiment H_1 in

[19] Notice that $\mathcal{B}$ is an adversary against Assumption T^F in the standard model, that is it can manipulate the input directly, whereas $\mathcal{A}$ is an adversary against T' in our generic model and so can manipulate the input only via oracle calls.

our generic model. It is clear that $W^{H_1} = W_2^T$ and thus, by (1), we have that (2) $W \leq W^{T_F} + W^{H_1}$.

We now consider an hybrid experiment H_2 that is identical to the experiment H_1 except for the following. First, recall that by our setting (see Section 2.1), the modulus N is product of two primes p, q such that $p = 2p' + 1, q = 2q' + 1$ for two primes p', q'. Note that the order of the group elements m is $p'q'$. When the adversary issues an equality query between two elements x, x' of I (resp. S), the query is answered by checking whether $x \mod p' = x' \mod p'$ (resp. $x \mod p = x \mod p$). That is, the "q-part" of the computation is ignored.

Let us denote by Bad the event that at some point in the computation one of the two following events happens:

- The oracle adds to its internal state an element $x \in I$ and the internal state already contains an element $x' \in I$ such that $x \neq x'$ (as integers) and $x \equiv x' \mod p'$
- The oracle adds to its internal state an element $x \in S$ and the internal state already contains an element $x' \in S$ such that $x \neq x'$ and $x \equiv x' \mod p$.

We now show that if in experiment H_1 the event Bad occurs with probability ϵ, then there exists an adversary $\mathcal{B}$ breaking (in the standard model) Assumption T^F with probability ϵ.

Indeed, an adversary $\mathcal{B}$ against T^F can simulate to an adversary $\mathcal{A}$ experiment H_1 by internally simulating the oracle as shown before. When an element x of I (resp. S) is stored in the internal representation of the oracle, $\mathcal{B}$ computes $\gcd(x, x')$ for each other element x' of I (resp. S) internally stored before. It is easy to see that if the event Bad occurs then either there exist x, x' belonging to I such that $x - x'$ is a multiple of p' or there exist x, x' belonging to S such that $x - x'$ is a multiple of p. In both cases, $\mathcal{B}$ can easily compute a factor of N from $\gcd(x, x')$.

Let W^{H_2} be the probability that Adv wins in the experiment H_2 in our generic model, let $W^{H_1,\neg\mathsf{Bad}}$ be the probability that Adv wins in the experiment H_2 in our generic model when the event Bad does not occur. We thus proved that (3) $W^{H_1} \leq W^{T_F} + W^{H_1,\neg\mathsf{Bad}}$.

It is clear that if event Bad does not occur, the output of experiment H_1 is distributed identically to experiment H_2, so $W^{H_1,\neg\mathsf{Bad}} = W^{H_2}$. From (2), (3) and the previous fact, it follows that (4) $W \leq 2 \cdot W^{T_F} + W^{H_2}$.

We now consider an hybrid experiment H_3 that is identical to the experiment H_2 except that the "q-part" of each operand in an oracle query is ignored. That is, if two integers x and y have to be summed, subtracted or multiplied, the operation is done between $x \mod p'$ and $y \mod p$. If a group operation between g and h has to be done, the group operation is done between $g \mod p$ and $h \mod p$. If an exponentiation between g and x has to be done, the exponentiation is done between $g \mod p$ and $x \mod p'$.

Let W^{H_3} the probability that Adv wins in the experiment H_3. For $i = 2, 3$, let y^{H_i} be the result of an arbitrary integer (resp. group or exponentiation) operation in experiment H_i and let z^{H_i} be $y^{H_i} \mod p'$ (resp. $y^{H_i} \mod p$).

Let us analyze the following cases.

– Addition, subtraction and multiplication between $x, y \in I$. Let $z^{H_2} \stackrel{\triangle}{=} (x+y) \mod p'$ and $z^{H_3} \stackrel{\triangle}{=} \big((x \mod p') + (y \mod p')\big) \mod p'$. Then $z^{H_2} = z^{H_3}$, and similarly for subtraction and multiplication.
– Group operation between $g, h \in S$. Let $z^{H_2} \stackrel{\triangle}{=} (g \cdot h) \mod p$ and $z^{H_3} \stackrel{\triangle}{=} \big((g \mod p) \cdot (h \mod p)\big) \mod p$. Then $z^{H_2} = z^{H_3}$.
– Exponentiation between $g \in S$ and $x \in I$. Let $z^{H_2} \stackrel{\triangle}{=} (g^x) \mod p$ and $z^{H_3} \stackrel{\triangle}{=} \big((g \mod p)^{(x \mod p')}\big) \mod p$. Note that g has order $p'q'$ and "its p-part" has order p'. Then $z^{H_2} = z^{H_3}$.

So, in all cases z^{H_2} is equal to z^{H_3}. By definition of experiments H_2 and H_3, the view of Adv in experiment H_2 (resp. H_3) depends only on z^{H_2} (resp. z^{H_3}); this is because the only output that Adv gets from the oracle are the binary answers to each equality query between two elements $x, y \in I$ (resp. $g, h \in S$) that, by definitions of experiments H_2 and H_3, is equal to 1 if and only if $x \mod p' = y \mod p'$ (resp. $g \mod p = h \mod p$). Therefore, it follows that the output of experiment H_3 is distributed identically to the one of experiment H_2, that is $W^{H_2} = W^{H_3}$ and thus, by (4), we have that (5) $W \leq 2 \cdot W^{T_F} + W^{H_3}$.

Notice that in experiment H_3, the q'-part of w is information theoretically hidden so the probability that the second winning condition is satisfied is $\leq 1/q'$, a negligible quantity, that is (6) $W^{H_3} \leq 1/q'$.

From (5) and (6), it follows that $W \leq 2 \cdot W^{T_F} + 1/q'$, as we had to prove. $\triangle$

Remark 14 If the modular operation $x \mod x_2$ between two integers $x, x_2 \in I$ were allowed in our generic model (that is, if for each two integers $x, x_2 \in I$ the adversary might request to the oracle to update I adding the integer $x \mod x_2$), the view of Adv in experiment H_2 would not be identical to the view of Adv in experiment H_3. Indeed, in general it may be that the integer $\Big(\big((x \mod p') \mod x_2\big) \mod p'\Big)$ is different (as integer) from the integer $\big((x \mod x_2) \mod p'\big)$. So, the proof of Theorem 19 would break down if our generic model included the modular operation. In general, the proof of the theorem would work for any function f with the following property: $f(x \mod p, y \mod p) \mod p = f(x, y) \mod p$ (resp. $f(x \mod p', y \mod p') \mod p' = f(x, y) \mod p'$) for any $x \in S$ (resp. $x \in I$).

This is consistent with the fact that our model extended with the modular operation would likely become meaningless. Indeed, by application of the Chinese remainder theorem it can be seen that, just via oracle calls to the modular operation, the adversary is able to deterministically extract each encoded element in the clear.[20] $\triangle$

[20] This observation has been communicated to us by Tibor Jager.