An extended abstract of this paper is published in the proceedings of the 13th International Conference on Cryptology in India [21] – Indocrypt 2012. This is the full version.

On the Non-malleability of the Fiat-Shamir Transform

Sebastian Faust[1], Markulf Kohlweiss[2], Giorgia Azzurra Marson[3], and Daniele Venturi[1]

[1] *Aarhus University*
[2] *Microsoft Cambridge*
[3] *Technische Universität Darmstadt*

Abstract

The Fiat-Shamir transform is a well studied paradigm for removing interaction from public-coin protocols. We investigate whether the resulting non-interactive zero-knowledge (NIZK) proof systems also exhibit non-malleability properties that have up to now only been studied for NIZK proof systems in the common reference string model: first, we formally define simulation soundness and a weak form of simulation extraction in the random oracle model (ROM). Second, we show that in the ROM the Fiat-Shamir transform meets these properties under lenient conditions. A consequence of our result is that, in the ROM, we obtain truly efficient non malleable NIZK proof systems essentially for free. Our definitions are sufficient for instantiating the Naor-Yung paradigm for CCA2-secure encryption, as well as a generic construction for signature schemes from hard relations and simulation-extractable NIZK proof systems. These two constructions are interesting as the former preserves both the leakage resilience and key-dependent message security of the underlying CPA-secure encryption scheme, while the latter lifts the leakage resilience of the hard relation to the leakage resilience of the resulting signature scheme.

Contents

1 Introduction **3**

2 Preliminaries **4**

3 Properties of NIZKs in the Random Oracle Model **7**

4 On the Non-malleability of the Fiat-Shamir Transform **10**
4.1 Simulation Soundness 10
4.2 Weak Simulation Extractability 13

5 Applications **14**
5.1 Leakage Resilience 14
5.2 Key-Dependent Message Security 16

A Defining Proofs of Knowledge **21**

B Honest-verifier zero-knowledge and neighborhood **21**

C Proof of Proposition 1 **22**

D Details Omitted from Applications **24**
D.1 Definitions of Leakage-Resilient Primitives 24
D.2 Proof of Theorem 4 25
D.3 Revisiting Naor-Segev in the ROM 26
D.4 Σ-Protocol for BHHO 29
D.5 Revisiting Camenisch-Chandran-Shoup in the ROM 30

1 Introduction

Zero-knowledge proof systems [27] are a powerful tool for designing cryptographic primitives and protocols. They force malicious parties to behave according to specification while allowing honest parties to protect their secrets. Non-interactive zero-knowledge (NIZK) proofs [11] consist of a single proof message passed from the prover to the verifier. They are particularly useful for designing public-key encryption and signature schemes as the proof can be added to the ciphertext and signature respectively. Understanding the most efficient NIZK proofs that are sufficiently strong, i.e., sufficiently non-malleable, for building signature and encryption schemes with strong security properties is thus of fundamental importance in cryptography. It was shown by Goldreich and Oren [26] that NIZK proofs are unattainable in the standard model. To avoid this impossibility result, one must rely on additional assumptions, such as common reference strings [10] (CRS model) or idealizations of hash functions [8] (random oracle model, ROM).

With the aim of finding the "right" definition in the non-interactive case, several flavors of non-malleability [20] have been introduced for NIZK in the CRS model [39, 40, 25, 32]. The notion of *simulation soundness*, which bridges soundness and zero knowledge, guarantees that soundness holds even after seeing accepting proofs, for both true and *false* statements, produced by the simulator. This strengthened soundness notion was first proposed by Sahai in [39], and later improved by De Santis et al. [40]. The notion of simulation extraction [40, 29] in addition requires that accepting proofs allow to extract witnesses. Different variants of simulation extraction have been proposed by [16, 19].

Until recently, zero-knowledge in general and NIZK in particular were considered to be primarily of theoretical interest. Significant exceptions being efficient Σ-protocols [17, 18] and their non-interactive relatives based on the Fiat-Shamir (FS) transform [22]. A Σ-protocol is a three-move interactive scheme where the prover sends the first message and the verifier sends a random challenge as the second message. In a nutshell, the Fiat-Shamir transform removes the interaction by computing the challenge as the hash value of the first message and the theorem that is being proven. Σ-protocols and the Fiat-Shamir transform are widely used in the construction of efficient identification [22], anonymous credential [15], signature [37, 1], e-voting schemes [9], and many other cryptographic constructions [12, 6, 24].

Most work on the provable security of zero-knowledge has, however, been conducted either on interactive proof systems in the plain model or on NIZK in the CRS model, while practitioners often preferred Fiat-Shamir based NIZK proofs for their simplicity and efficiency. The use of the Fiat-Shamir transform was most thoroughly explored in the security proofs of signature schemes in the random oracle model [37, 1], but was otherwise often used heuristically. The question thus arises whether one can lay sound foundations for the FS transform in the light of recent research on CRS-based NIZKs. To this end, we provide non-malleability definitions for NIZK in the random oracle model that closely follow the established CRS-based definitions [29]. An earlier result oriented in the same direction, but concerning a Σ-protocol for a *specific* language,[1] was given by Fouque and Pointcheval [24]. Their proof strategy relies on the forking lemma [37] and (implicitly) on the fact that the Σ-protocol they consider has a particular property called *strong special honest-verifier zero-knowledge* (SS-HVZK). Since there exist Σ-protocols that do not satisfy the SS-HVZK property, Fouque and Pointcheval's proof cannot be immediately extended to the general case. Moreover, we make the random oracle explicit in our definition, which is crucial as definitions in the random oracle model can be brittle [42].

Our first observation is that much less is required to show simulation soundness for any FS-NIZK proof. Namely, in the random oracle model, simulation soundness simply follows from

[1] This is the language used in the Naor-Yung transform when the underlying encryption is the ElGamal scheme.

the soundness and the HVZK properties of the underlying interactive protocol. In particular, it is neither necessary to rely on the forking lemma, nor on the strong [2] property of SS-HVZK. We also show that the proof strategy of [24], when generalized properly to any Σ-protocol, yields something more than just simulation soundness. In fact, one gets some form of simulation extractability, which we call *weak simulation extractability*. In a nutshell, *full simulation extractability* requires that even after seeing many simulated proofs, whenever an adversary outputs a new accepted proof, we can build an algorithm to extract a valid witness. Sometimes, such a strong extraction property is called *online* extraction [23] because the extractor outputs a witness directly after receiving the adversary's proof. In comparison, our notion is weaker in that it allows the extractor to fully control the adversary (i.e., rewind it).

Our contribution. Our contributions are threefold. First, we formally define the notions of zero-knowledge (which holds trivially for the Fiat-Shamir transform), simulation soundness, and simulation extractability for NIZKs in the random oracle model. Second, we show that simulation soundness and a weak form of simulation extractability come for free if one uses the FS-transform for turning Σ-protocols into NIZK proof systems. Third, we investigate the consequences of this result by showing that our definitions are sufficient for instantiating the Naor-Yung paradigm for constructing CCA2-secure encryption schemes, and generic construction for signature schemes from hard relations and simulation-extractable NIZK proof systems [19]. These two constructions are particularly interesting as the former preserves both leakage resilience and key-dependent message security of the underlying CPA-secure encryption scheme, while the latter lifts the leakage resilience of the hard relation to the leakage resilience of the resulting signature scheme. To our knowledge, these are the most efficient schemes having such properties, if one is willing to rely on the ROM.[3]

Related work. The only other efficient transform for Σ-protocols yielding simulation soundness (again in the random oracle model) is Fischlin's transform [23] which is designed with the purpose of online extraction and is less efficient than the classical Fiat-Shamir transform. Therefore, it would be interesting to investigate whether Fischlin's transform achieves a stronger form of simulation extractability. We notice that in the interactive case, a general transform from any Σ-protocol to an (unbounded) simulation-sound Σ-protocol using one-time signatures has been proposed [25]. In the common reference string model the most efficient simulation-sound or simulation extractable NIZK proof system are based on Groth-Sahai proofs [30]. One has however to pay the price of proving a structure-preserving CCA secure encryption [19] (for true-simulation extraction) or a structure-preserving signature scheme [3] (for full simulation extraction).

2 Preliminaries

Notation. Let k be a security parameter. A function ν is called *negligible* if $\nu(k) \leq k^{-c}$ for any $c > 0$ and sufficiently large k. Given two functions f, g, we write $f \approx g$ if there exists a negligible function ν such that $|f(k) - g(k)| < \nu(k)$. Given an algorithm $\mathcal{A}$, $y \leftarrow \mathcal{A}(x)$ means that y is the output of $\mathcal{A}$ on input x; when $\mathcal{A}$ is randomized, then y is a random variable. We write $\mathcal{A}^H$ to denote the fact that $\mathcal{A}$ has oracle access to some function H. PPT stands for

[2] In Appendix B, we show a separation between the two notions of special HVZK [17] (S-HVZK) and SS-HVZK, by showing that any S-HVZK Σ-protocol can be turned into another Σ-protocol which is still S-HVZK but not SS-HVZK, assuming that one-way functions exist.

[3] In particular we obtain as a special case the Alwen et al. [4] leakage-resilient signature scheme based on the Okamoto identification scheme.

$$\begin{array}{lcl}
\underline{\textsf{Prover}\quad \mathcal{P}(x, w; 1^k)} & & \underline{\textsf{Verifier}\quad \mathcal{V}(x; 1^k)} \\
\alpha \leftarrow \mathcal{P}_0(x, w; \rho) & \xrightarrow{\alpha} & \\
 & \xleftarrow{\beta} & \beta \overset{\$}{\leftarrow} \mathcal{V}_0(x, \alpha) \\
\gamma \leftarrow \mathcal{P}_1(\alpha, \beta, x, w; \rho) & \xrightarrow{\gamma} & \text{Accept iff } \mathcal{V}_1(x, \alpha, \beta, \gamma) = 1
\end{array}$$

Figure 1: A Σ-protocol for a language $\mathcal{L}$.

probabilistic-polynomial time. A *decision problem* related to a language $\mathcal{L} \subseteq \{0,1\}^*$ consists in determining whether a string x is in $\mathcal{L}$ or not. Given an instance x, we say that $\mathcal{A}$ decides (or recognizes) $\mathcal{L}$ if, after a finite number of steps, the algorithm halts and outputs $\mathcal{A}(x) = 1$ if $x \in \mathcal{L}$, otherwise $\mathcal{A}(x) = 0$. (Sometimes, we may call "theorem" a string belonging to the language at hand.) We can associate to any NP-language $\mathcal{L}$ a polynomial-time recognizable relation $\mathcal{R}_\mathcal{L}$ defining $\mathcal{L}$ itself, that is $\mathcal{L} = \{x : \exists w \text{ s.t. } (x, w) \in \mathcal{R}_\mathcal{L}\}$, where $|w| \leq \mathit{poly}(|x|)$. The string w is called a *witness* or *certificate* for membership of $x \in \mathcal{L}$. For NP, w corresponds to the non-deterministic choices made by $\mathcal{A}$.

Interactive protocols. An interactive proof system (IPS) for membership in $\mathcal{L}$ is a two-party protocol, where a prover wants to convince an efficient verifier that a string x belongs to $\mathcal{L}$. In a zero-knowledge interactive proof system, a prover $\mathcal{P}$ can convince a verifier $\mathcal{V}$ that $x \in \mathcal{L}$ without revealing anything beyond the fact that the statement is indeed true. Informally, this means that $\mathcal{V}$ cannot exploit the interaction with $\mathcal{P}$ for gaining extra-knowledge. Such a property is formalized by requiring the existence of an efficient algorithm $\mathcal{S}$, the *zero-knowledge simulator*, which produces messages indistinguishable from conversations between an honest prover $\mathcal{P}$ and a malicious verifier $\mathcal{V}^*$. Besides the zero-knowledge property, any proof system satisfies two standard requirements: proving true statements is always possible, while it should be infeasible to convince the verifier to accept a false statement as correct. These two conditions are called *completeness* and *soundness* respectively. Related to the concept of interactive proof systems, but even more subtle, is the notion of *proof of knowledge.* In a proof of knowledge (PoK), $\mathcal{P}$ wants to convince $\mathcal{V}$ that he *knows* a secret witness which implies the validity of some assertion, and not merely that the assertion is true. To formalize the fact that a prover actually "knows something", we require that there exists an efficient algorithm $\mathcal{E}$, called *knowledge extractor*, that when given complete access to the program of the prover can extract the witness.

An IPS or an interactive PoK is called *public-coin* when the verifier's moves consist merely of tossing coins and sending their outcomes to the prover. (In contrast, in a *private-coin* IPS the verifier does not need to show the outcome of the coins to the prover [28].) We are mainly interested in a specific class of public-coin interactive PoK systems for NP-languages, called Σ-protocols. Here, the parties involved share a string x belonging to a language $\mathcal{L} \in$ NP and the prover also holds a witness w for membership of $x \in \mathcal{L}$. Thus, the prover $\mathcal{P}$ wants to convince the verifier $\mathcal{V}$ that it "knows" a witness w for x, i.e. that x is in the language, without revealing the witness itself. Σ-protocols have a 3-move shape where the first message α, called *commitment*, is sent by the prover and then, alternatively, the parties exchange the other messages β and γ, called (respectively) *challenge* and *response.* The interaction is depicted in Figure 1. Besides the standard properties held by any IPS, Σ-protocols satisfy a flavour of zero-knowledge — called *honest-verifier zero knowledge* (HVZK) — saying that an *honest* verifier taking part in the protocol does not learn anything beyond the validity of the theorem being proven.

Definition 1 (Σ-protocols)**.** *A Σ-protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ for an* NP*-language $\mathcal{L}$ is a three-round public-coin IPS where $\mathcal{P} = (\mathcal{P}_0, \mathcal{P}_1)$ and $\mathcal{V} = (\mathcal{V}_0, \mathcal{V}_1)$ are PPT algorithms, with the following* *additional proprieties:*

Completeness. *If $x \in \mathcal{L}$, any proper execution of the protocol between $\mathcal{P}$ and $\mathcal{V}$ ends with the verifier accepting $\mathcal{P}$'s proof.*

Honest-verifier zero knowledge (*HVZK*). *There exists an efficient algorithm $\mathcal{S}$, called zero-knowledge simulator, such that for any PPT distinguisher $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ and for any $(x, w) \in \mathcal{R}_{\mathcal{L}}$, the view of the following two experiments, real and simulated, are computationally indistinguishable:*

$$\begin{array}{l|l}
\textbf{Experiment } \mathsf{Exp}^{\mathrm{REAL}}_{\Sigma,\mathcal{D}}(1^k) & \textbf{Experiment } \mathsf{Exp}^{\mathrm{SIM}}_{\Sigma,\mathcal{D}}(\mathcal{S}, 1^k) \\
\quad (x, w, \delta) \leftarrow \mathcal{D}_0(1^k) & \quad (x, w, \delta) \leftarrow \mathcal{D}_0(1^k) \\
\quad \pi \leftarrow \langle \mathcal{P}(x, w; 1^k), \mathcal{V}(x; 1^k) \rangle & \quad \pi \leftarrow \mathcal{S}(x, 1^k) \\
\quad \textit{Output } \mathcal{D}_1(\pi, \delta) & \quad \textit{Output } \mathcal{D}_1(\pi, \delta)
\end{array}$$

where $\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle$ denotes the verdict returned at the end of the interaction between $\mathcal{P}$ and $\mathcal{V}$ on common input x and private input w.

Soundness. *If $x \notin \mathcal{L}$ then any malicious (even unbounded) prover $\mathcal{P}^*$ is accepted only with negligible probability.*

Special soundness. *There exists an efficient algorithm $\mathcal{E}$, called special extractor, such that given two accepting conversations (α, β, γ) and $(\alpha, \beta', \gamma')$ for a string x, where $\beta \neq \beta'$, then $w \leftarrow \mathcal{E}(\alpha, \beta, \gamma, \beta', \gamma', x)$ is such that $(x, w) \in \mathcal{R}_{\mathcal{L}}$.*

The special soundness property is strong enough to imply both soundness and that Σ-protocols are PoK [18]. Sometimes Σ-protocols are required to meet stronger notions of HVZK. [4]

A non-standard condition that many Σ-protocols satisfy, introduced by Fischlin in [23], requires that responses are quasi unique, i.e. given an accepting proof it should be infeasible to find a new valid response for that proof.

Definition 2 (Quasi unique responses)**.** *A Σ-protocol has* quasi unique responses *if for any PPT $\mathcal{A}$ and for any security parameter k it holds:*

$$\mathrm{Prob}[(x, \alpha, \beta, \gamma, \gamma') \leftarrow \mathcal{A}(1^k) \ : \ \mathcal{V}(x, \alpha, \beta, \gamma) = \mathcal{V}(x, \alpha, \beta, \gamma') = 1 \ \wedge \ \gamma \neq \gamma'] \approx 0.$$

A Σ-protocol has *unique responses* if the probability above is zero. The latter condition, defined by Unruh in [41], is also known as *strict soundness.*

Min-entropy of commitments. Following [1, 2], we use the concept of min-entropy to measure how likely it is for a commitment to collide with a fixed value.

Definition 3 (Min-entropy of commitment)**.** *Let k be a security parameter and $\mathcal{L}$ be an* NP*-language with relation $\mathcal{R}_{\mathcal{L}}$. Consider a pair $(x, w) \in \mathcal{R}_{\mathcal{L}}$ and let $(\mathcal{P}, \mathcal{V})$ be an arbitrary three-round IPS. Denote with $\mathsf{Coins}(k)$ the set of coins used by the prover and consider the set $A(x, w) = \{\mathcal{P}_0(x, w; \rho) : \rho \leftarrow \mathsf{Coins}(k)\}$ of all possible commitments associated to w. The min-entropy function associated to $(\mathcal{P}, \mathcal{V})$ is defined as $\varepsilon(k) = \min_{(x,w)}(-\log_2 \mu(x, w))$,where the minimum is taken over all possible (x, w) drawn from $\mathcal{R}_{\mathcal{L}}$ and $\mu(x, w)$ is the maximum probability that a commitment takes on a particular value, i.e., $\mu(x, w) = \max_{\alpha \in A(x,w)}(Prob[\mathcal{P}_0(x, w; \rho) = \alpha : \rho \leftarrow \mathsf{Coins}(k))])$.*

[4] We discuss these notions and implications and non implications between them in the Appendix B.

We say that $(\mathcal{P},\mathcal{V})$ is *non-trivial* if $\varepsilon(k) = \omega(\log(k))$ is super-logarithmic in k. Often, the commitment is drawn uniformly from some set. In order for $(\mathcal{P},\mathcal{V})$ to be non-trivial, this set must have size exponential in k. Notice that most of natural Σ-protocols meet such a condition and, in fact, non-triviality is quite easy to achieve, e.g. by appending redundant random bits to the commitment.

Forking lemma. To prove our second main result, we make use of the following version of the forking lemma, which appeared in [6].

Lemma 1 (General forking lemma)**.** *Fix an integer Q and a set $\mathcal{H}$ of size $h \geq 2$. Let P be a randomized program that on input $y, h_1, \ldots, h_Q$ returns a pair, the first element of which is an integer in the range $0, \ldots, Q$ and the second element of which we refer to as a* side output*. Let IG be a randomized algorithm that we call the input generator. The accepting probability of P, denoted* acc*, is defined as the probability that $J \geq 1$ in the experiment $y \leftarrow \mathsf{IG}; h_1, \ldots, h_Q \leftarrow \mathcal{H}; (J, s) \leftarrow \mathsf{P}(y, h_1, \ldots, h_Q)$.*

The forking algorithm $\mathsf{F_P}$ associated to P is the randomized algorithm that on input y proceeds as follows.

```
Algorithm $\mathsf{F_P}(y)$
    Pick coins $\rho$ for $\mathsf{P}$ at random
    $h_1, \ldots, h_Q \leftarrow \mathcal{H}$
    $(I, s) \leftarrow \mathsf{P}(y, h_1, \ldots, h_Q; \rho)$
    If $I = 0$ return $(0, \perp, \perp)$
    $h'_I, \ldots, h'_Q \leftarrow \mathcal{H}$
    $(I', s') \leftarrow \mathsf{P}(y, h_1, \ldots, h_{I-1}, h'_I, \ldots, h'_Q; \rho)$
    If $(I = I') \wedge (h_I \neq h'_I)$ return $(1, s, s')$ else return $(0, \perp, \perp)$
```

Let $\mathrm{ext} = \mathrm{Prob}[b = 1 : y \leftarrow \mathsf{IG}; (b, s, s') \leftarrow \mathsf{F_P}(y)]$*, then* $\mathrm{ext} \geq \mathrm{acc}\left(\frac{\mathrm{acc}}{Q} - \frac{1}{h}\right)$.

3 Properties of NIZKs in the Random Oracle Model

Removing interaction. The Fiat-Shamir transform was originally designed to turn three-round identification schemes into efficient signature schemes. As Σ-protocols are an extension of three-round identification schemes, it is not surprising that they can be considered as a starting point for the Fiat-Shamir transform. The Fiat-Shamir paradigm applies to any Σ-protocol (and more generally to any three-round public-coin proof system): We start from an interactive protocol $(\mathcal{P},\mathcal{V})$ and remove the interaction between $\mathcal{P}$ and $\mathcal{V}$ by replacing the challenge, chosen at random by the verifier, with a hash value $H(\alpha, x)$ computed by the prover, where H is a hash function modeled as a random oracle. Thus, the interactive protocol $(\mathcal{P},\mathcal{V})$ is turned into a non-interactive one: The resulting protocol, denoted $(\mathcal{P}^H,\mathcal{V}^H)$, is called *Fiat-Shamir proof system.*

Throughout this paper, we refer to the so called *explicitly programmable* random oracle model [42] (EPROM) where the simulator is allowed to program the random oracle explicitly. We model this by defining the zero-knowledge simulator $\mathcal{S}$ of a non-interactive zero-knowledge proof system as a stateful algorithm that can operate in two modes: $(h_i, st) \leftarrow \mathcal{S}(1, st, q_i)$ takes care of answering random oracle queries (usually by lazy sampling) while $(\pi, st) \leftarrow \mathcal{S}(2, st, x)$ simulates the actual proof. Note that calls to $\mathcal{S}(1, \cdots)$ and $\mathcal{S}(2, \cdots)$ share the common state st that is updated after each operation.

Definition 4 (Unbounded non-interactive zero knowledge)**.** *Let $\mathcal{L}$ be a language in* NP*. Denote with $(\mathcal{S}_1, \mathcal{S}_2)$ the oracles such that $S_1(q_i)$ returns the first output of $(h_i, st) \leftarrow \mathcal{S}(1, st, q_i)$ and $\mathcal{S}_2(x, w)$ returns the first output of $(\pi, st) \leftarrow \mathcal{S}(2, st, x)$ if $(x, w) \in \mathcal{R}_{\mathcal{L}}$. We say a protocol $(\mathcal{P}^H, \mathcal{V}^H)$ is a NIZK proof for language $\mathcal{L}$ in the random oracle model, if there exists a PPT simulator $\mathcal{S}$ such that for all PPT distinguishers $\mathcal{D}$ we have*

$$\mathrm{Prob}[\mathcal{D}^{H(\cdot),\mathcal{P}^H(\cdot,\cdot)}(1^k) = 1] \approx \mathrm{Prob}[\mathcal{D}^{\mathcal{S}_1(\cdot),\mathcal{S}_2(\cdot,\cdot)}(1^k) = 1],$$

where both $\mathcal{P}$ and $\mathcal{S}_2$ oracles output $\perp$ if $(x, w) \notin \mathcal{R}_{\mathcal{L}}$.

A well known fact is that, in the random oracle model, the Fiat-Shamir transform allows to efficiently design digital signature schemes [22] and non-interactive zero-knowledge protocols. In fact, an appealing characteristic of this transform is that many properties of the starting protocol are still valid after applying it. In particular, it has been proven that the Fiat-Shamir transform turns any three-round public-coin zero-knowledge interactive proof system into a NIZK proof system [8]. It is straightforward to prove that the same holds when the starting protocol is ZK only with respect to a honest verifier, as stated in the following Theorem.

Theorem 1 (Fiat-Shamir NIZKs)**.** *Let k be a security parameter. Consider a non-trivial three-round public-coin honest-verifier zero-knowledge interactive proof system $(\mathcal{P}, \mathcal{V})$ for a language $\mathcal{L} \in$ NP. Let H be a function with range equal to the space of the verifier's coins. In the random oracle model the proof system $(\mathcal{P}^H, \mathcal{V}^H)$, derived from $(\mathcal{P}, \mathcal{V})$ by applying the Fiat-Shamir transform, is unbounded non-interactive zero-knowledge.*

sketch. To prove that the proof system $(\mathcal{P}^H, \mathcal{V}^H)$ is non-interactive zero-knowledge it is sufficient to show that there exists a simulator $\mathcal{S}$ as required in Definition 4. This can be done by invoking the HVZK simulator associated with the underlying interactive proof system. In particular, $\mathcal{S}$ works as follows:

- To answer query $q = (x, \alpha)$ to $\mathcal{S}_1$, $\mathcal{S}(1, st, q)$ lazily samples a lookup table $\mathcal{T}_H$ kept in state st. It checks whether $\mathcal{T}_H(q)$ is already defined. If this is the case, it returns the previously assigned value; otherwise it returns and sets a fresh random value (of the appropriate length).
- To answer query x to $\mathcal{S}_2$ (respectively $\mathcal{S}_2'$), $\mathcal{S}(2, st, x)$ calls the HVZK simulator of $(\mathcal{P}, \mathcal{V})$ on input x to obtain a proof (α, β, γ). Then, it updates $\mathcal{T}_H$ in such a way that $\beta = \mathcal{T}_H(x, \alpha)$. If $\mathcal{T}_H$ happens to be already defined on this input, $\mathcal{S}$ returns failure and aborts.

We call this simulator *canonical*. The main result of Fiat-Shamir [22] (expressed for their particular identification protocol) is that $\mathcal{S}$ is a "good" NIZK simulator. The crucial step in the proof is that the starting protocol $(\mathcal{P}, \mathcal{V})$ is non-trivial (cf. Definition 3), thus the probability of failure in each of the queries to $\mathcal{S}_2'$ is upper-bounded by $\mathrm{Prob}[\mathsf{failure}] \leq 2^{-\varepsilon(k)}$, which is negligible in k. □

Simulation soundness. The soundness property of a proof system ensures that no malicious prover can come up with an accepting proof for a string that does not belong to the language in question (i.e., for a false theorem). However, it is not clear whether this condition still holds *after* the attacker observes valid proofs for adaptively chosen (true or false) statements. The notion of *simulation soundness* deals with this case.

Definition 5 (Unbounded simulation soundness)**.** *Let $\mathcal{L}$ be a language in* NP*. Consider a proof system $(\mathcal{P}^H, \mathcal{V}^H)$ for $\mathcal{L}$, with zero-knowledge simulator $\mathcal{S}$. Denote with $(\mathcal{S}_1, \mathcal{S}_2')$ the oracles such* *that $S_1(q_i)$ returns the first output of $(h_i, st) \leftarrow \mathcal{S}(1, st, q_i)$ and $\mathcal{S}_2'(x)$ returns the first output of $(\pi, st) \leftarrow \mathcal{S}(2, st, x)$. We say that $(\mathcal{P}^H, \mathcal{V}^H)$ is simulation sound with respect to $\mathcal{S}$ in the random oracle model, if for all PPT adversaries $\mathcal{A}$ the following holds:*

$$\text{Prob}[(x^\star, \pi^\star) \leftarrow \mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}_2'(\cdot)} : (x^\star, \pi^\star) \notin \mathcal{T} \ \wedge \ x^\star \notin \mathcal{L} \ \wedge \ \mathcal{V}^{\mathcal{S}_1}(x^\star, \pi^\star) = 1] \approx 0,$$

where $\mathcal{T}$ is the list of pairs (x_i, π_i), i.e., respectively queries asked to and proofs returned by the simulator.

We stress that the above definition relies crucially on the zero-knowledge property of $(\mathcal{P}^H, \mathcal{V}^H)$, as we use a probability experiment that defines a property of $\mathcal{S}$ to define a property about $(\mathcal{P}^H, \mathcal{V}^H)$. In particular the definition is most meaningful for a simulator $\mathcal{S}$ for which the simulation of the random oracle of $\mathcal{S}_1$ is consistent with a truly random oracle H. Also note that $\mathcal{S}_2'$ allows $\mathcal{A}$ to ask for simulated proofs of false statements.

The possibility to request proofs of false statements has an interesting consequence: simulation soundness holds only with respect to specific simulators and not in general for all NIZK simulators. In particular, one can construct a NIZK proof system $(\mathcal{P}^H, \mathcal{V}^H)$ that is simulation sound with respect to a simulator $\mathcal{S}$ but for which there exists a valid NIZK simulator $\hat{\mathcal{S}}$, such that $(\mathcal{P}^H, \mathcal{V}^H)$ cannot be simulation sound with respect to $\hat{\mathcal{S}}$. To see this, consider a $\mathcal{V}^H$ that accepts all proofs if $H(0) = 0$. $\hat{\mathcal{S}}$ simulates a consistent random oracle until it receives a proof of a false statement (one of which could be hard-coded in $\hat{\mathcal{S}}$ or easy to recognize) at which point it sets $\mathcal{T}_H(0) = 0$. Note that a similar counterexample exists for CRS-based NIZK [29]: $\hat{\mathcal{S}}_2$ can simply return the simulation trapdoor when queried on a false statement.

Simulation extractability. Combining simulation soundness and knowledge extraction, we may require that even after seeing (polynomially) many simulated proofs, whenever $\mathcal{A}$ makes a new proof it is possible to extract a witness. This property is called simulation extractability, and implies simulation soundness. Indeed, if we can extract a witness from the adversary's proof even with small probability, then obviously the statement must belong to the language in question. We introduce a weaker flavor of simulation extractability which we call *weak simulation extractability*. The main difference with full simulation extractability is that the extractor $\mathcal{E}_\mathcal{A}$ is now given complete control over the adversary $\mathcal{A}$, meaning that it is allowed to rewind $\mathcal{A}$ and gets to see the answers of $(\mathcal{S}_1, \mathcal{S}_2')$. Moreover, we require that if $\mathcal{A}$ outputs an accepting proof with some probability, then $\mathcal{E}_\mathcal{A}$ can extract with almost the same probability.

Definition 6 (Weak simulation extractability)**.** *Let $\mathcal{L}$ be a language in* NP*. Consider a NIZK proof system $(\mathcal{P}^H, \mathcal{V}^H)$ for $\mathcal{L}$ with zero-knowledge simulator $\mathcal{S}$. Let $(\mathcal{S}_1, \mathcal{S}_2')$ be oracles returning the first output of $(h_i, st) \leftarrow \mathcal{S}(1, st, q_i)$ and $(\pi, st) \leftarrow \mathcal{S}(2, st, x)$ respectively. We say that $(\mathcal{P}^H, \mathcal{V}^H)$ is* weakly simulation-extractable *with* extraction error ν *and with respect to $\mathcal{S}$ in the random oracle model, if for all PPT adversaries $\mathcal{A}$ there exists an efficient algorithm $\mathcal{E}_\mathcal{A}$ with access to the answers $\mathcal{T}_H$, $\mathcal{T}$ of $(\mathcal{S}_1, \mathcal{S}_2')$ respectively such that the following holds. Let:*

$$\begin{aligned}\text{acc} &= \text{Prob}\big[(x^\star, \pi^\star) \leftarrow \mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}_2'(\cdot)}(1^k; \rho) : (x^\star, \pi^\star) \notin \mathcal{T}; \mathcal{V}^{\mathcal{S}_1}(x^\star, \pi^\star) = 1\big] \\ \text{ext} &= \text{Prob}\big[(x^\star, \pi^\star) \leftarrow \mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}_2'(\cdot)}(1^k; \rho); \\ &\qquad\qquad w^\star \leftarrow \mathcal{E}_\mathcal{A}(x^\star, \pi^\star; \rho, \mathcal{T}_H, \mathcal{T}) : \ (x^\star, \pi^\star) \notin \mathcal{T}; (x^\star, w^\star) \in \mathcal{R}_\mathcal{L}\big],\end{aligned}$$

where the probability space in both cases is over the random choices of $\mathcal{S}$ and the adversary's random tape ρ. Then, there exist a constant $d > 0$ and a polynomial p such that whenever $\text{acc} \geq \nu$, *we have* $\text{ext} \geq \frac{1}{p}(\text{acc} - \nu)^d$.

The above definition is inspired by similar notions in the context of proofs of knowledge [5, 31, 41]. The value ν is called *extraction error* of the proof system. We omit for better readability that values $\mathrm{acc}, \mathrm{ext}, p, \nu$ all depend on the security parameter k. Note that a non-negligible extractor error can be made exponentially small by sequential repetitions (see Appendix C for a proof).

Proposition 1 (Extraction error amplification). *Let $(\mathcal{P}^H, \mathcal{V}^H)$ be a weakly simulation extractable NIZK proof system with extraction error ν. Then, the proof system $(\mathcal{P'}^H, \mathcal{V'}^H)$ obtained by repeating sequentially $(\mathcal{P}^H, \mathcal{V}^H)$ for a number n of times yields a weakly simulation extractable NIZK proof system with extraction error ν^n.*

It is useful to look at the relation between weak simulation extractability and the following stronger property modeling online-extraction.

Definition 7 (Full Simulation extractability). *Let $\mathcal{L}$ be a language in* NP*. Consider a NIZK proof system $(\mathcal{P}^H, \mathcal{V}^H)$ for $\mathcal{L}$ with simulator $\mathcal{S}$. Let $(\mathcal{S}_1, \mathcal{S}'_2)$ be oracles returning the first output of $(h_i, st) \leftarrow \mathcal{S}(1, st, q_i)$ and $(\pi, st) \leftarrow \mathcal{S}(2, st, x)$ respectively. We say that $(\mathcal{P}^H, \mathcal{V}^H)$ is* strongly simulation extractable *with respect to $\mathcal{S}$ in the random oracle model, if there exists an efficient algorithm $\mathcal{E}$ such that for all PPT adversaries $\mathcal{A}$ the following holds. Let:*

$$\begin{aligned}\mathrm{Prob}\big[w^\star \leftarrow \mathcal{E}(st, x^\star, \pi^\star) \ : \ (x^\star, \pi^\star) \leftarrow \mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}'_2(\cdot)}(1^k; \rho);\\ (x^\star, \pi^\star) \notin \mathcal{T}; \quad \mathcal{V}^{\mathcal{S}_1}(x^\star, \pi^\star) = 1; \quad (x^\star, w^\star) \notin \mathcal{R}_{\mathcal{L}}\big] \approx 0\end{aligned}$$

where $\mathcal{T}$ is the list of transcripts (x_i, π_i) returned by the simulator and the probability space is over the random choices of $\mathcal{S}$ and the adversary's randomness ρ.

Note that both our unbounded simulation soundness and our full simulation extractability definitions can be instantiated by common reference string based schemes. H turns into a random constant function that always returns the reference string, and the state st consists of the simulation and extraction trapdoors.

4 On the Non-malleability of the Fiat-Shamir Transform

4.1 Simulation Soundness

We now show that NIZK proofs obtained via the Fiat-Shamir transform from any IPS of the public-coin type additionally satisfying the HVZK property are simulation sound. Since Σ-protocols are a special class of HVZK public-coin IPSs, we get as a corollary that Fiat-Shamir NIZK proofs obtained from Σ-protocols are simulation-sound.

Theorem 2 (Simulation soundness of the Fiat-Shamir transform). *Consider a* non-trivial *three-round public-coin HVZK interactive proof system $(\mathcal{P}, \mathcal{V})$ for a language $\mathcal{L} \in$* NP*, with quasi unique responses. In the random oracle model, the proof system $(\mathcal{P}^H, \mathcal{V}^H)$ derived from $(\mathcal{P}, \mathcal{V})$ via the Fiat-Shamir transform is a* simulation-sound *NIZK with respect to its canonical simulator $\mathcal{S}$.*

Proof. We assume that $(\mathcal{P}^H, \mathcal{V}^H)$ is a non-interactive zero-knowledge proof system with the simulator $\mathcal{S}$ described in the proof of Theorem 1, and show that $(\mathcal{P}^H, \mathcal{V}^H)$ is simulation sound. We proceed by contradiction. Suppose there exists a PPT adversary $\mathcal{A}$ that breaks the simulation soundness of the non-interactive protocol with non-negligible probability

$$\epsilon := \mathrm{Prob}\big[(x^\star, \pi^\star) \leftarrow \mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}'_2(\cdot)} : \ (x^\star, \pi^\star) \notin \mathcal{T} \ \wedge \ x^\star \notin \mathcal{L} \ \wedge \ \mathcal{V}^{\mathcal{S}_1}(x^\star, \pi^\star) = 1\big].$$

In such a case, we are able to build two reductions $\hat{\mathcal{P}}$ and $\mathcal{P}^*$ which, by using $\mathcal{A}$ as a black-box, violate either the quasi unique response or the soundness properties of the underlying interactive protocol $(\mathcal{P}, \mathcal{V})$ respectively, contradicting our hypothesis. Recall that $\mathcal{S}_1$ simulates answers to the RO, while $\mathcal{S}_2'$ replies with an accepting proof π. Without loss of generality we assume that whenever adversary $\mathcal{A}$ succeeds and outputs an accepting proof $(\alpha^\star, \gamma^\star)$, she has previously queried the oracle $\mathcal{S}_1$ on input $(x^\star, \alpha^\star)$. The argument for this is that it is straightforward to transform any adversary that violates this condition into an adversary that makes one additional query to $\mathcal{S}_1$ and wins with the same probability.

A simple but crucial observation is that adversary $\mathcal{A}$ may have learned $\alpha^\star$ by querying the oracle $\mathcal{S}_2'$ on input $x^\star$ or might have computed it itself. We denote the first by the event proof, the second by the event $\overline{\mathsf{proof}}$. As these events are mutually exclusive and exhaustive, we have:

$$\text{Prob}[\mathcal{A} \text{ wins}] \; = \text{Prob}[\mathcal{A} \text{ wins } \wedge \; \mathsf{proof}] \; + \; \text{Prob}[\mathcal{A} \text{ wins } \wedge \; \overline{\mathsf{proof}}].$$

Now we have two different cases to analyze, each of them corresponding to the probability in the expression above.

In the first case (when proof happens), we assume that $x^\star$ is asked to $\mathcal{S}_2'$ and the answer is a proof of the type $(\alpha^\star, -)$. We show how to use an adversary $\mathcal{A}$ that makes use of $(x^\star, \alpha^\star)$ in its fake proof to build a reduction $\hat{\mathcal{P}}$. In this way we bound $\text{Prob}[\mathcal{A} \text{ wins } \wedge \; \mathsf{proof}]$ by the probability that $\hat{\mathcal{P}}$ wins in breaking the quasi unique response property. First, observe that if $\mathcal{A}$ wins with a proof $(\alpha^\star, -)$ and makes the query $x^\star$ to S_2', she implicitly has to set also the value $\mathcal{S}_1(x^\star, \alpha^\star) := \beta$ such that $H(\alpha^\star, x^\star) = \beta$. This forces her to use such values for producing a proof $x^\star, \pi^\star$. Thus, the fake proof she outputs must be of the type $\pi^\star = (\alpha^\star, -)$ for $x^\star$ such that $H(x^\star, \alpha^\star) = \beta$. On the other hand, such a proof cannot be the same returned by the simulator, since a forgery cannot appear in the list $\mathcal{T}$. This means that a fake proof for $x^\star$ produced by $\mathcal{A}$ must have the first argument equal to $\alpha^\star$, gives by the simulator $\mathcal{S}_2'$. But as the proof of $\mathcal{A}$ is accepted the responses must be different, and so $\hat{\mathcal{P}}$ must be successful in breaking the quasi unique response property. This intuition can be formalized by giving an explicit reduction from $\mathcal{A}$ to $\hat{\mathcal{P}}$.

Consider an algorithm $\hat{\mathcal{P}}$ which runs $\mathcal{A}$ internally as a black-box. Thus, $\hat{\mathcal{P}}$ sees all queries $\mathcal{A}$ makes to the oracles $\mathcal{S}_1$ and $\mathcal{S}_2'$ and produces their answers. The internal description of $\hat{\mathcal{P}}$ follows:

- $\hat{\mathcal{P}}$ answers $\mathcal{S}_1$ and $\mathcal{S}_2'$ and keeps lists $\mathcal{T}_H$ and $\mathcal{T}$ respectively as the real simulator $\mathcal{S}$ would.
- When $\mathcal{A}$ outputs a fake proof $(\alpha^\star, \gamma^\star)$ for $x^\star$, $\hat{\mathcal{P}}$ looks through its lists $\mathcal{T}$ and $\mathcal{T}_H$ until it finds $(x^\star, (\alpha^\star, \gamma))$ and $((x^\star, \alpha^\star), \beta)$ respectively;
- It returns $(x^\star, \alpha^\star, \beta, \gamma^\star, \gamma)$.

We claim that algorithm $\hat{\mathcal{P}}$ breaks the quasi unique response property. Indeed, the proof produced by $\mathcal{A}$ is accepting by $\mathcal{V}^H$ on common input $x^\star$. On the other hand, the proof $(\alpha^\star, \gamma)$ is given by the simulator, therefore it must be accepting for $x^\star$. Given this, it holds $\mathcal{V}^H(x^\star, \alpha^\star, \gamma^\star) \; = \; \mathcal{V}^H(x^\star, \alpha^\star, \gamma) = 1$, that means

$$\mathcal{V}(x^\star, \alpha^\star, H(x^\star, \alpha^\star), \gamma^\star) \; = \; \mathcal{V}(x^\star, \alpha^\star, H(x^\star, \alpha^\star), \gamma) = 1,$$

where $H(x^\star, \alpha^\star) = \beta$. The conclusion is that either $\gamma = \gamma^\star$, that is excluded since $\mathcal{A}$ cannot win by printing a simulated proof, or algorithm $\hat{\mathcal{P}}$ succeeds in breaking the quasi unique response property. We obtain:

$$\text{Prob}[\mathcal{A} \text{ wins } \wedge \; \mathsf{proof}] \; = \; \text{Prob}[\hat{\mathcal{P}} \text{ wins}] \leq \mathit{negl}(k).$$

In case proof does not happen, we can use adversary $\mathcal{A}$ that does not query $\mathcal{S}_2'$ with input $x^\star$ to build a reduction $\mathcal{P}^*$ and bound $\text{Prob}[\mathcal{A} \text{ wins} \wedge \overline{\mathsf{proof}}]$ by the probability $\text{Prob}[\mathcal{P}^* \text{ wins}] \cdot Q$ of breaking the soundness of the underlying interactive scheme. $\mathcal{P}^*$ runs $\mathcal{A}$ as a black-box and has to simulate its environment by answering the queries to $\mathcal{S}_1$ and $\mathcal{S}_2'$ in a consistent way. More precisely, $\mathcal{P}^*$ works as follows. It guesses uniformly at random an index $j \in [Q]$ and replies to queries to $\mathcal{S}_1$ and $\mathcal{S}_2'$ in the following way:

1. Answer query (x_i, α_i) to $\mathcal{S}_1$:
   (a) Query $1 \le i \le j-1$: Returns $H(x_i, \alpha_i)$ if it is already defined; otherwise it samples a random value β_i and sets $H(x_i, \alpha_i) := \beta_i$.
   (b) Query j: Runs the protocol with the honest verifier $\mathcal{V}$ for statement x_j, using as a commitment the value α_j. Obtains challenge β_j from $\mathcal{V}$ and program the oracle as $H(x_j, \alpha_j) := \beta_j$. The answer to $\mathcal{A}$'s query is β_j.
   (c) Query $j+1 \le i \le Q$: Proceed as in Step 1a.
2. Answer query x to $\mathcal{S}_2'$: Run the HVZK simulator of the interactive protocol on input x to obtain an accepting proof (α, β, γ), and program the oracle H in such a way that $H(x, \alpha) := \beta$. If the NIZK simulator returns $\mathsf{failure}$, which happens when $H(x, \alpha)$ is already defined, output $\mathsf{failure}$ and abort, otherwise output (α, γ).
3. Answer $\mathcal{V}$'s challenge: Let $x^\star, (\alpha^\star, \gamma^\star)$ be the instance and the proof output by $\mathcal{A}$. Return $\gamma^\star$ to $\mathcal{V}$ as the response to challenge β_j in step 1b.

We need to estimate the probability that $\mathcal{P}^*$ succeeds in breaking the soundness of the interactive scheme $(\mathcal{P}, \mathcal{V})$ in terms of the probability that $\mathcal{A}$ outputs an accepting proof $(\alpha^\star, \gamma^\star)$ for a false statement $x^\star$. Suppose that $(x^\star, \alpha^\star)$ has been asked to the random oracle as the $j^\star$-th query and we have $j = j^\star$, i.e., $\mathcal{P}^*$ guesses the correct index for which $\mathcal{A}$ outputs an accepting proof for a false statement $x^\star$. In such a case, $\mathcal{P}^*$ breaks the soundness of $(\mathcal{P}, \mathcal{V})$. Hence, we get:

$$\begin{aligned} \text{Prob}[\mathcal{P}^* \text{ wins}] &= \text{Prob}[\mathcal{A} \text{ wins} \;\wedge\; j = j^\star \;\wedge\; \overline{\mathsf{proof}}] \\ &= \text{Prob}[\mathcal{A} \text{ wins} \;\wedge\; \overline{\mathsf{proof}}] \cdot \text{Prob}[j = j^\star], \end{aligned}$$

where the second equality comes from the fact that $\mathcal{P}^*$ guesses $j^\star$ correctly indipendently of the event that $\mathcal{A}$ is successful and $\overline{\mathsf{proof}}$ happens. Since the index j is chosen at random in $[Q]$, we have $\text{Prob}[\mathcal{P}^* \text{ wins}] = \frac{1}{Q} \cdot \text{Prob}[\mathcal{A} \text{ wins} \wedge \overline{\mathsf{proof}}]$. Whenever $\mathcal{P}^*$ wins, it breaks the soundness of the interactive scheme: by hypothesis, this happens only with negligible probability. Therefore:

$$\text{Prob}[\mathcal{A} \text{ wins} \wedge \overline{\mathsf{proof}}] = Q \cdot \text{Prob}[\mathcal{P}^* \text{ wins}] \le \mathit{negl}(k).$$

Now we can bound the probability that $\mathcal{A}$ succeeds. As we assume, $\mathcal{A}$ breaks the simulation soundness of the scheme with non-negligible probability ϵ:

$$\text{Prob}[\mathcal{A} \text{ wins}] \le \text{Prob}[\mathcal{A} \text{ wins} \wedge \mathsf{proof}] + \text{Prob}[\mathcal{A} \text{ wins} \wedge \overline{\mathsf{proof}}] \le \mathit{negl}(k),$$

thus $\epsilon \le \mathit{negl}(k)$, that is a contradiction. □

On the quasi-unique responses condition. We remark that assuming $(\mathcal{P}, \mathcal{V})$ has quasi-unique responses is not an artifact of the proof. In fact, without this property, proofs would be malleable and breaking the simulation soundness would be an easy task. Consider a FS-NIZK proof system for which responses are not quasi-unique. An efficient adversary $\mathcal{A}$ can always query the simulator on input a false statement $x^\star$, obtaining a simulated proof $\mathcal{S}_2'(x^\star) \to \pi^\star = (\alpha^\star, \beta^\star, \gamma^\star)$. Given $\pi^\star$, $\mathcal{A}$ might be able to find, with non-negligible probability, a new response $\gamma^{\star\star} \neq \gamma^\star$ such that $(\alpha^\star, \beta^\star, \gamma^{\star\star})$ is also accepting. Hence, the scheme cannot be simulation sound.

4.2 Weak Simulation Extractability

The argument Fouque and Pointcheval use in [24] to show that the proof system they consider is simulation sound is roughly as follows. Assume there exists an adversary $\mathcal{A}$ which outputs a pair $(x^\star, \pi^\star)$ breaking the simulation soundness, as in the experiment of Definition 5. Then, one can invoke a suitable version of the forking lemma to show that it is possible to "extract" a witness $w^\star$ for $x^\star$ from such an adversary, contradicting the fact that $x^\star$ is false. The reduction simulates the list $\mathcal{T}$ for $\mathcal{A}$ in the simulation soundness experiment, in particular one needs to fake accepting proofs for (adaptively chosen and potentially false) theorems selected by the attacker. In order to do so, Fouque and Pointcheval (implicitly) rely on the SS-HVZK property. The next theorem is a generalization of the above strategy which does not rely on the SS-HVZK property and indeed applies to arbitrary languages. Moreover, we are able to prove a stronger statement, namely that Fiat-Shamir proofs satisfy weak simulation extractability (and not only simulation soundness). For simplicity the following theorem assumes (perfect) unique responses, but could be generalized using the same reduction as for Theorem 2.

Theorem 3 (Weak simulation extractability of the Fiat-Shamir transform)**.** *Let* $\Sigma = (\mathcal{P}, \mathcal{V})$ *be a* non-trivial Σ*-protocol with unique responses for a language* $\mathcal{L} \in$ NP*. In the random oracle model, the* NIZK *proof system* $\Sigma_{FS} = (\mathcal{P}^H, \mathcal{V}^H)$ *resulting by applying the Fiat-Shamir transform to* Σ *is weakly simulation extractable with* extraction error $\nu = \frac{Q}{h}$ *for the canonical simulator* $\mathcal{S}$*. Here,* Q *is the number of random oracle queries and* h *is the number of elements in the range of* H*. Furthermore, the extractor* $\mathcal{E}_\mathcal{A}$ *needs to run* $\mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}_2'(\cdot)}$ *twice, where* $\mathcal{A}$ *and* $\mathcal{E}_\mathcal{A}$ *are both defined in Definition 6.*

Proof. Let $\mathcal{S}$ be the canonical zero-knowledge simulator described in the proof of Theorem 2. Denote with $(x^\star, \alpha^\star, \gamma^\star)$ the pair statement/proof returned by $\mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}_2'(\cdot)}$; we describe an extractor $\mathcal{E}_\mathcal{A}$, able to compute a witness $w^\star$ by rewinding $\mathcal{A}$ once.

We want to exploit the general forking lemma1. In order to do so, we define program $\mathsf{P}(1^k, h_1, \ldots, h_Q; \rho_\mathsf{P})$ as follows: P virtually splits ρ_P into two random tapes ρ and $\rho_\mathcal{S}$ (e.g. by using even bits for ρ and odd bits for $\rho_\mathcal{S}$) and runs internally $\mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}_2'(\cdot)}$ with randomness ρ. P uses values $(h_1, \ldots, h_Q)$ to simulate fresh answers of $\mathcal{S}_1$, and $\rho_\mathcal{S}$ to simulate answers of $\mathcal{S}_2$. If $\mathcal{A}^{\mathcal{S}_1(\cdot), \mathcal{S}_2'(\cdot)}$ outputs $(x^\star, (\alpha^\star, \gamma^\star))$, P checks that it is a valid proof and not in $\mathcal{T}$ (otherwise it returns $(0, \perp)$). Then, because of the unique response property, $(x^\star, \alpha^\star)$ must correspond to some fresh query to $\mathcal{S}_1$ and P outputs $(J, (x^\star, \alpha^\star, \gamma^\star))$, where $J > 0$ is the index corresponding to the random oracle query $(x^\star, \alpha^\star)$. We say that P is successful whenever $J \geq 1$, and we denote with acc the corresponding probability. Given program P, we consider two related runs of P with the same random tape but different hash values, as specified by the forking algorithm F_P of Lemma 1. Denote with $(I, (x^\star, \alpha^\star, \gamma^\star)) \leftarrow \mathsf{P}(1^k, h_1, \ldots, h_Q; \rho)$ and $(I', (x^{\star\star}, \alpha^{\star\star}, \gamma^{\star\star})) \leftarrow \mathsf{P}(1^k, h_1, \ldots, h_{I-1}, h_I', \ldots, h_Q'; \rho)$ the two outputs of $\mathcal{A}$ in these runs. By the forking lemma we know that with probability ext $\geq$ acc(acc$/Q - 1/h$) the forking algorithm will return indexes I, I' such that $I = I'$, $I \geq 1$ and $h_I \neq h_I'$.

Notice that since F_P's forgeries are relative to the *same* random oracle query $I = I'$, we must have $x^\star = x^{\star\star}$ and $\alpha^\star = \alpha^{\star\star}$; on the other hand we have $h_I \neq h_I'$. We are thus in a position to invoke the special extractor $\mathcal{E}$ for the underlying proof system, yielding a valid witness $w^\star \leftarrow \mathcal{E}(\alpha^\star, h_I, \gamma^\star, h_I', \gamma^{\star\star}, x^\star)$ such that $(x^\star, w^\star) \in \mathcal{R}_\mathcal{L}$.

Assume now that acc $\geq \nu$. By applying the general forking lemma1 we obtain that ext $\geq$ acc$^2/Q -$ acc$/h$. Since Q is polynomial while h is exponentially large in the security parameter, for sure $\frac{Q}{h} < 1$ (in particular, it is negligible in k). As $\nu := \frac{Q}{h}$, we have:

$$\frac{\mathrm{acc}^2}{Q} - \frac{\mathrm{acc}}{h} = \frac{1}{Q}(\mathrm{acc}^2 - \mathrm{acc} \cdot \nu).$$

Now, since $\mathrm{acc} \geq \nu$, we have $\mathrm{acc} \cdot \nu \geq \nu^2$, that is $\nu^2 - \mathrm{acc} \cdot \nu \leq 0$. Hence,

$$\frac{1}{Q}(\mathrm{acc}^2 - \mathrm{acc} \cdot \nu) \geq \frac{1}{Q}(\mathrm{acc}^2 - 2\mathrm{acc} \cdot \nu + \nu^2) = \frac{1}{Q}(\mathrm{acc} - \nu)^2.$$

The previous inequality matches the definition of weak extractability with values $p = Q$ and $d = 2$.

□

5 Applications

In the literature there is a large number of applications for simulation-sound or extractable NIZKs. One of the first request for simulation soundness comes from the setting of public key encryption, for the design of encryption schemes with *chosen-ciphertext security* using the Naor-Yung (NY) paradigm [36]. At a high level, the NY works as follows: given two key pairs (sk, pk) and (sk', pk') for CPA-secure encryption schemes Π and Π' respectively, a ciphertext consists of two encryptions c, c' of the same message m, under different keys pk, pk', and a NIZK proof π that both c and c' encrypt m. In order to achieve security against *adaptive* chosen-ciphertext attacks (CCA2), the underlying NIZK must be simulation-sound. While achieving CCA2 security is probably one of the most prominent application of simulation soundness, simulation-sound or extractable proofs have been used, e.g., to build also leakage-resilient signatures or KDM secure encryption. In this section, we review some important applications of such proof systems and show how our result provides more efficient constructions in the ROM or generalizes earlier results that use the Fiat-Shamir transform.

5.1 Leakage Resilience

Simulation-sound and simulation-extractable NIZK proofs have been very useful in constructing leakage-resilient encryption and signature schemes [19, 33, 35]. Here, we consider these works and show that our result immediately yields efficient leakage-resilient schemes in the random oracle model.

Leakage-resilient signatures. A signature scheme is leakage resilient if it is hard to forge a signature even given (bounded) leakage from the signing key. Obviously, this requires that the amount of leakage given to the adversary has to be smaller than the length of the secret key, as otherwise the leakage may just reveal such a key, trivially breaking the security of the signature scheme.

We instantiate the generic construction of leakage-resilient signatures based on leakage-resilient hard relations and simulation-extractable NIZKs of [19] using the Fiat-Shamir transform. Let $\mathcal{R}$ be a λ-leakage-resilient hard relation with sampling algorithm $\mathsf{Gen}_{\mathcal{R}}$ (for detailed definitions, see Definition 9 and Definition 10 in Appendix D.1). Let $(\mathcal{P}^H, \mathcal{V}^H)$ be a NIZK argument[5] for relation $\mathcal{R}'$ defined by $\mathcal{R}'((pk, m), sk) \Leftrightarrow \mathcal{R}(pk, sk)$. Consider the following signature scheme:

$\mathsf{KeyGen}(1^k)$: Calls $(pk, sk) \leftarrow \mathsf{Gen}_{\mathcal{R}}(1^k)$ and returns the same output.

$\mathsf{Sign}(sk, m)$: Outputs $\sigma \leftarrow \mathcal{P}^H((pk, m), sk)$.[6]

[5] As opposed to a proof system where soundness needs to hold unconditionally, in an *argument* system it is sufficient that soundness holds with respect to a computationally bounded adversary.

[6] Note that m is part of the instance being proven.

$\mathsf{Vrf}(pk, m, \sigma)$: Verifies the signature by invoking $\mathcal{V}^H((pk, m), \sigma)$.

Notice that $\sigma \leftarrow \mathcal{P}^H((pk, m), sk)$ is a NIZK proof for the hard relation obtained by applying the Fiat-Shamir transform.

We chose to state the theorem below using an argument system as this is the minimal requirement under which leakage resilience of the scheme can be proven. Since our FS-based protocols are weakly simulation-extractable NIZK *proof* systems, they automatically satisfy the hypothesis of Theorem 4.

Theorem 4. *If $\mathcal{R}$ is a 2λ-leakage-resilient hard relation and $(\mathcal{P}^H, \mathcal{V}^H)$ is a weakly simulation-extractable NIZK argument with negligible extraction error for relation $\mathcal{R}'((pk, m), sk) \Leftrightarrow \mathcal{R}(pk, sk)$, then the above scheme is a λ-leakage-resilient signature scheme in the random oracle model.*

The proof of the theorem from above follows the one of Theorem 4.3 in [19]. A couple of subtleties arise, though. The main idea of the proof is to build a reduction from an adversary $\mathcal{A}$ breaking λ-leakage resilience of the signature scheme to an adversary $\mathcal{B}$ breaking the hardness of the 2λ-leakage-resilient hard relation $\mathcal{R}$. Roughly speaking, in the reduction $\mathcal{B}$ is given some instance pk and simulates the signing queries of $\mathcal{A}$ by using the zero-knowledge simulator of the NIZK, and the leakage queries by using the leakage oracle for the relation $\mathcal{R}$. At some point $\mathcal{A}$ outputs a forgery $\sigma^\star$ and $\mathcal{B}$ invokes the extractor of Theorem 3 to get $sk^\star \leftarrow \mathcal{E}_{\mathcal{A}}(pk, \sigma^\star)$. The first issue is that we are only guaranteed *weak* simulation-extractability, whereas the proof of [19] relies on full simulation-extractability. [7] However, this is not a problem because we just need to show that $\mathcal{B}$ outputs a valid witness with non-negligible probability. A second issue involves the extractor of Theorem 3, which needs to rewind $\mathcal{A}$ once and, thus, to simulate twice its environment (including the leakage queries). This causes the loss of a factor 2 in the total amount of tolerated leakage. We refer the reader to Appendix D.2 for the details.

We emphasize that the leakage-resilient signature scheme of Alwen et al. [4], obtained by applying the Fiat-Shamir transform to the Okamoto identification scheme, follows essentially the above paradigm. Here, one may view the public and secret keys of the Okamoto ID scheme as forming an instance of a leakage-resilient hard relation, while the NIZK proof corresponds to the Fiat-Shamir transform applied to the Okamoto identification protocol.

Naor-Yung with leakage. The definition of IND-CPA and IND-CCA security of an encryption scheme can be extended to the leakage setting by giving the adversary access to a leakage oracle. Naor and Segev [35] show that the Naor-Yung paradigm instantiated with a simulation-sound NIZK allows to leverage CPA-security to CCA-security even in the presence of leakage. In other words, if Π is CPA-secure against λ-key-leakage attacks, the encryption scheme obtained by applying the Naor-Yung paradigm to (Π, Π), using a simulation-sound NIZK, is CCA2-secure against λ-key-leakage attacks. In Appendix D.3 we revisit their proof in the ROM, dealing with the issue that the leakage queries can potentially depend on H. We stress that for the proof only *simulation soundness* is needed (i.e., our result from Theorem 2) and not weak simulation extractability.

In what follows, we propose a concrete instantiation of the result above, relying on the BHHO encryption scheme from [13]. Let $\mathbb{G}$ be a group of prime-order q. For randomly selected generators $g_1, \ldots, g_\ell \xleftarrow{\$} \mathbb{G}$, the public key is a tuple $pk = (g_1, \ldots, g_\ell, h)$, where $h = \prod_{i=1}^{\ell} g_i^{z_i}$ for a secret key $sk = (z_1, \ldots, z_\ell) \in \mathbb{Z}_q^\ell$. To encrypt a message $m \in \mathbb{G}$, choose a random $r \xleftarrow{\$} \mathbb{Z}_q$ and output $c = (c_1, \ldots, c_{\ell+1}) = (g_1^r, \ldots, g_\ell^r, m \cdot h^r)$. The message m can be recovered by computing $m = c_{\ell+1} \cdot (\prod_{i=1}^{\ell} c_i^{z_i})^{-1}$.

[7] Actually, they rely on a weaker property called *true* simulation-extractability [19].

Assuming that the DDH problem is hard in $\mathbb{G}$, Naor and Segev [35] showed that the BHHO encryption scheme is CPA-secure against λ-key-leakage attacks for any $\ell = 2 + \frac{\lambda+\omega(\log k)}{\log q}$, where k is the security parameter. Applying the Naor-Yung paradigm, consider the language:

$$\mathcal{L} = \big\{(c, pk, c', pk') : \ \exists r, r' \in \mathbb{Z}_q, m \in \mathbb{G} \text{ s.t.} \\ c = (g_1^r, \ldots, g_\ell^r, h^r \cdot m), \ c' = (g_1^{r'}, \ldots, g_\ell^{r'}, h'^{r'} \cdot m)\big\},$$

where $c = (c_1, \ldots, c_{\ell+1})$ and $c' = (c'_1, \ldots, c'_{\ell+1})$ are BHHO encryptions with randomness r and r', using public keys $pk = (g_1, \ldots, g_\ell, h)$ and $pk' = (g_1, \ldots, g_\ell, h')$ respectively. The pair $w = (r, r')$ is a witness for a string $x = (c, pk, c', pk') \in \mathcal{L}$. Consider the following interactive protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ for the above language:

1. $\mathcal{P}$ chooses s, s' at random from $\mathbb{Z}_q$ and computes the commitment:
$$\vec{\alpha} = ((\alpha_1, ..., \alpha_\ell), (\alpha'_1, ..., \alpha'_\ell), \alpha'') = ((g_1^s, ..., g_\ell^s), (g_1^{s'}, ..., g_\ell^{s'}), h^s \cdot (h')^{s'}).$$
2. The verifier $\mathcal{V}$ chooses a random challenge $\beta \xleftarrow{\$} \mathbb{Z}_q$.
3. The prover computes the response $\vec{\gamma} = (\gamma, \gamma') = (s - \beta \cdot r, s' + \beta \cdot r')$.
4. Given a proof $\pi = (\vec{\alpha}, \beta, \vec{\gamma})$, the verifier $\mathcal{V}$ checks that:
$$\begin{aligned}(\alpha_1, \ldots, \alpha_\ell) &= (g_1^\gamma \cdot c_1^\beta, \ldots, g_\ell^\gamma \cdot c_\ell^\beta) \\ (\alpha'_1, \ldots, \alpha'_\ell) &= (g_1^{\gamma'} \cdot (c'_1)^{-\beta}, \ldots, g_\ell^{\gamma'} \cdot (c'_\ell)^{-\beta}) \\ \alpha'' &= h^\gamma \cdot (h')^{\gamma'} \cdot (c_{\ell+1} \cdot (c'_{\ell+1})^{-1})^\beta.\end{aligned}$$

In Appendix D.4 we prove that the above protocol is a Σ-protocol for the language $\mathcal{L}$. With the Naor-Yung paradigm applied to the BHHO encryption scheme we get a ciphertext (c, c', π) consisting of $4\ell + 3$ elements in $\mathbb{G}$ plus 2 elements in $\mathbb{Z}_q$. Moreover, the fact that the BHHO encryption scheme is CPA-secure against key leakage together with the result of Naor-, show that the above instantiation is CCA-secure against key-leakage attacks.

Corollary 1. *Let k be a security parameter. Assuming that the DDH problem is hard in $\mathbb{G}$, the Naor-Yung paradigm applied to the BHHO encryption scheme yields an encryption scheme that is CCA-secure against λ-key-leakage attacks in the random oracle model for $\lambda = \ell \log q(1 - \frac{2}{\ell} - \frac{\omega(\log k)}{\ell \log q}) = L(1 - o(1))$, where L is the length of the secret key. An encryption consists of $4\ell + 3$ elements in $\mathbb{G}$ plus 2 elements in $\mathbb{Z}_q$.*

5.2 Key-Dependent Message Security

Key-dependent message (KDM) security of a public-key encryption scheme requires that the scheme remains secure even against attackers allowed to see encryptions of the value $f(sk)$, where $f \in \mathcal{F}$ for some class of functions $\mathcal{F}$.

Camenisch, Chandran and Shoup [14] show that a variation of the Naor-Yung paradigm instantiated with a simulation-sound NIZK can still leverage CPA-security to CCA-security, even in the context of KDM security. We revisit their proof in the random oracle model in Appendix D.5. Also in this case, only *simulation soundness* is needed for the proof.

Roughly, for some function family $\mathcal{F}$, if Π is KDM[$\mathcal{F}$]-CPA secure and Π' is CPA-secure, the scheme Π'' obtained by applying the Naor-Yung paradigm to (Π, Π') — i.e., an encryption of $m \in \mathcal{M}$ is a tuple $c'' = (c, c', \pi)$ where c encrypts m under Π, c' encrypts m under Π' and π

is a simulation-sound NIZK proof that c and c' encrypt the same message — is KDM[$\mathcal{F}$]-CCA secure.

Let $sk_i[j]$ denote the j-th bit of sk_i. The BHHO encryption scheme was the first KDM-CPA secure encryption scheme, with respect to the class of all projection functions $\mathcal{F}_\downarrow = \mathcal{F}_{\mathrm{read}} \cup \mathcal{F}_{\mathrm{flip}}$, where

$$\mathcal{F}_{\mathrm{read}} = \left\{ f_{i,j} : \; \vec{sk} \to sk_i[j] \right\}_{i,j} \text{ and } \mathcal{F}_{\mathrm{flip}} = \left\{ f_{i,j} : \; \vec{sk} \to 1 - sk_i[j] \right\}_{i,j}.$$

More generally, when the message space is a linear space over $\mathbb{Z}_q$, we define the function class $\mathcal{PJ}(\mathcal{F}_\downarrow)$ as the class of all affine combinations of elements in $\mathcal{F}_\downarrow$.

Now we can instantiate the general transform of [14] as follows. We choose Π to be BHHO, Π' to be ElGamal (say with $pk' = h' = g_1^{z'_1}$) and we build a Σ-protocol Σ' for the Naor-Yung language relative to Π and Π'. Protocol Σ' can be easily derived from protocol Σ of the last section, by just compressing the commitment as in $\vec{\alpha} = ((g_1^s, \ldots, g_\ell^s), g_1^{s'}, h^s \cdot (h')^{s'})$ (and simplifying the verification procedure accordingly). Hence, Theorem 2 yields the following result.

Corollary 2. *Assuming that the DDH problem is hard in $\mathbb{G}$, the Naor-Yung paradigm instantiated with BHHO and ElGamal encryption schemes yields a* KDM[$\mathcal{PJ}(\mathcal{F}_\downarrow)$]-CCA *secure encryption scheme in the random oracle model. An encryption consists of $\ell + 3$ elements in $\mathbb{G}$ plus* 3 *elements in $\mathbb{Z}_q$.*

Beyond Naor-Yung. Another paradigm that yields chosen-ciphertext security from NIZKs, based on *proving knowledge of the plaintext*, was suggested by Rackoff and Simon [38]. Such a construction is somewhat more natural and more efficient than the twin-encryption paradigm: a message m is encrypted (only once) under a CPA-secure encryption scheme, and a NIZK proof of knowledge of the plaintext is attached to the ciphertext. However, (to the best of our knowledge) truly efficient constructions for sufficiently strong NIZK proofs of knowledge are not available even using random oracles. One can hope that using the weaker from of extractability afforded by the Fiat-Shamir transform one could at least obtain NM-CPA secure encyption, and this is indeed what is aimed at in the ongoing work of [9].

Acknowledgments

We thank Marc Fischlin and Ivan Damgård for the useful feedbacks provided on earlier versions of the paper. Sebastian Faust and Daniele Venturi acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, and also from the CFEM research center (supported by the Danish Strategic Research Council) within which part of this work was performed.

References

[1] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *EUROCRYPT*, pages 418–433, 2002.

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

[3] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristian Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology - CRYPTO '10*, pages 209–237, 2010.

[4] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[5] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *CRYPTO*, pages 390–420, 1992.

[6] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM Conference on Computer and Communications Security*, pages 390–399, 2006.

[7] Mihir Bellare and Todor Ristov. Hash functions from sigma protocols and improvements to VSH. In *ASIACRYPT*, pages 125–142, 2008.

[8] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[9] David Bernhard, Olivier Pereira, and Bogdan Warinschi. On necessary and sufficient conditions for private ballot submission. Cryptology ePrint Archive, Report 2012/236, 2012. `http://eprint.iacr.org/`.

[10] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988.

[11] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM J. Comput.*, 20(6):1084–1118, 1991.

[12] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

[13] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *CRYPTO*, pages 108–125, 2008.

[14] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *EUROCRYPT*, pages 351–368, 2009.

[15] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Proceedings of Eurocrypt 2001*, volume 2045, pages 93–118. Springer-Verlag, 2001.

[16] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 78–96. Springer, 2006.

[17] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.

[18] Ivan Damgård. On Σ-protocols, 2002. Available at `http://www.daimi.au.dk/~ivan/Sigma.ps`.

[19] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In *ASIACRYPT*, pages 613–631, 2010.

[20] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[21] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In *INDOCRYPT*, pages 60 – 79, 2012.

[22] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[23] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *CRYPTO*, pages 152–168, 2005.

[24] Pierre-Alain Fouque and David Pointcheval. Threshold cryptosystems secure against chosen-ciphertext attacks. In *ASIACRYPT*, pages 351–368, 2001.

[25] Juan A. Garay, Philip D. MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. *J. Cryptology*, 19(2):169–209, 2006.

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

[27] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[28] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *STOC*, pages 59–68, 1986.

[29] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *Proceedings of Asiacrypt 2006*, volume 4284, pages 444–459. Springer-Verlag, 2006.

[30] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[31] Shai Halevi and Silvio Micali. More on proofs of knowledge. Cryptology ePrint Archive, Report 1998/015, 1998. `http://eprint.iacr.org/`.

[32] Abhishek Jain and Omkant Pandey. Non-malleable zero knowledge: Black-box constructions and definitional relationships. Cryptology ePrint Archive, Report 2011/513, 2011. `http://eprint.iacr.org/`.

[33] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT*, pages 703–720, 2009.

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

[35] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[36] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.

[37] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[38] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.

[39] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.

[40] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO*, pages 566–598, 2001.

[41] Dominique Unruh. Quantum proofs of knowledge. In *EUROCRYPT*, pages 135–152, 2012.

[42] Hoeteck Wee. Zero knowledge in the random oracle model, revisited. In *ASIACRYPT*, pages 417–434, 2009.

A Defining Proofs of Knowledge

Many different definitions of proof of knowledge have been proposed in the literature. According to Unruh [41], it is possible to identify three main distinct notions, which he refers to as A, B and C-style PoKs.

Definition 8 (PoK). *Let $\nu : \{0,1\}^* \to [0,1]$ be a function and $\mathcal{R}$ be a binary relation, with corresponding language $\mathcal{L}_{\mathcal{R}}$. A protocol $(\mathcal{P},\mathcal{V})$ is a* proof of knowledge *for $\mathcal{R}$, with* knowledge error ν *if the following properties hold:*

Completeness. *For any pair $(x,w) \in \mathcal{R}$, if $\mathcal{P}$ and $\mathcal{V}$ follow the protocol specification on public input x and private input w (held by $\mathcal{P}$), then the verifier accepts.*

Validity. *There exist a probabilistic oracle machine $\mathcal{E}$ and a constant $c > 0$ such that, for any $x \in \mathcal{L}_{\mathcal{R}}$, whenever a prover $\mathcal{P}^*$ convinces $\mathcal{V}$ on common input x with probability $\epsilon(|x|) > \nu(|x|)$, then the extractor $\mathcal{E}^{\mathcal{P}^*}$, taking x as an input and having oracle access to $\mathcal{P}^*$, outputs a witness w such that $\mathcal{R}(x,w) = 1$,*

(A-style) *working in* strict *polynomial time, with probability at least*

$$\frac{(\epsilon(|x|) - \nu(|x|))^c}{p}$$

for $c > 0$ and a polynomial p.

(B-style) *within* expected *number of steps bounded by*

$$\frac{x^c}{\epsilon(|x|) - \nu(|x|)}.$$

(C-style) *working in* expected *polynomial time, with probability at least*

$$\frac{\epsilon(|x|) - \nu(|x|)}{p}.$$

B Honest-verifier zero-knowledge and neighborhood

The HVZK property essentially says there exists a simulator $\mathcal{S}$ that produces, on input x, transcripts that are indistinguishable from real conversations between $\mathcal{P}$ and $\mathcal{V}$ sharing input x, as long as $\mathcal{V}$ behaves honestly. It is possible to give stronger flavors of this property. A first example is the case of *special honest-verifier zero-knowledge* (SHVZK), as introduced by Cramer, Damgård and Schoenmakers [17]. Intuitively, this property says that the simulator $\mathcal{S}$ can take *any* β as input and produce a conversation indistinguishable from the space of all conversations between honest parties in which β is the challenge. To formalize this, the experiments of Definition 1 are modified in such a way that both the verifier and the simulator get also β as input. Even though the S-HVZK property is seemingly stronger than the plain HVZK property, the two notions are essentially equivalent. In fact, as Fischlin showed in [23], any Σ-protocol where the challenge has size logarithmic in the security parameter, satisfies the stronger requirement of S-HVZK.

A stronger variant of zero-knowledge — first introduced by Bellare and Ristov [7] — is *strong special honest-verifier zero-knowledge* (SS-HVZK), which differs from the previous one in the fact that the simulator has to produce indistinguishable triplets (α, β, γ) where the string γ can be chosen at random and $\mathcal{S}$ can compute the commitment α trough a *deterministic* function ϕ of

Prover $\mathcal{P}'(x, w)$		Verifier $\mathcal{V}'(x)$
$r \xleftarrow{\$} \{0,1\}^*$, $\alpha \leftarrow \mathcal{P}_0(x, w)$	$\xrightarrow{\alpha'=(\alpha,r)}$	
	$\xleftarrow{\beta}$	$\beta \leftarrow \mathcal{V}_0(x, \alpha)$
$z = f(r)$, $\gamma \leftarrow \mathcal{P}_1(\alpha', \beta', x, w)$	$\xrightarrow{\gamma'=(\gamma,z)}$	Accept iff $\mathcal{V}_1(x, \alpha, \beta, \gamma) = 1$ and $z = f(r)$

Figure 2: The protocol Σ', separating SS-HVZK and S-HVZK.

x, β and γ, i.e. $\alpha := \phi(x, \beta, \gamma)$. We exhibit a separation between the SS-HVZK and S-HVZK properties, by showing that *any* Σ-protocol which is SS-HVZK can be turned into another Σ-protocol which is still S-HVZK but not SS-HVZK. For proving such a claim, we just need to assume the existence of one-way functions. The transform is depicted in Figure 2. Let $f : \{0,1\}^* \to \{0,1\}^*$ be a one-way function. We start with an arbitrary Σ-protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ which satisfies the SS-HVZK property. The protocol $\Sigma' = (\mathcal{P}', \mathcal{V}')$ is essentially identical to Σ, however the prover now also samples a random r and adds r to the commitment α; moreover it also appends $z = f(r)$ to the last message. The verifier $\mathcal{V}'$ runs the verifier $\mathcal{V}$, but also checks that $z = f(r)$.

Proposition 2 (Separation between S-HVZK and SS-HVZK)**.** *Assuming the existence of one-way functions, there exists a Σ-protocol which is* S-HVZK *but not* SS-HVZK. *More precisely, let $\Sigma = (\mathcal{P}, \mathcal{V})$ be a Σ-protocol which is* SS-HVZK *and f be a one-way function. Then, the protocol $\Sigma' = (\mathcal{P}', \mathcal{V}')$ of Figure 2 is* S-HVZK *but not* SS-HVZK.

Proof. We show that the protocol Σ' cannot satisfy the stronger SS-HVZK property, even if the original protocol Σ does. The intuition for this is that, if it could be possible to compute the commitment α' *after* having chosen the challenge β and the response γ', then it could also be possible to invert f. We now turn this intuition into a formal proof. Assume, towards a contradiction, that Σ' is SS-HVZK. This means that there exists a simulator $\mathcal{S}'$ which can choose γ' at random (for all β), and compute $\alpha' = (\alpha, r)$ as a deterministic function $\alpha' = \phi(x, \beta, \gamma')$ of the other values. We now show how to use $\mathcal{S}'$ to build a PPT adversary $\mathcal{B}$ able to break one-wainess of the function f. The attacker $\mathcal{B}$ simply chooses a random image $z \xleftarrow{\$} \{0,1\}^*$, samples β, γ at random and runs $\mathcal{S}'(x, \beta, \gamma' = (\gamma, z), 1^k)$ which returns $\alpha' = \phi(x, \beta, \gamma')$. Since $\alpha' = (\alpha, r)$ is such that $f(r) = z$, the adversary $\mathcal{B}$ has found a pre-image of z, namely $r = f^{-1}(z)$, breaking the one-wayness of f. Note, however, that the protocol Σ' is still S-HVZK. In fact, for every β, the (special) HVZK simulator $\mathcal{S}'$ can just run the SS-HVZK simulator $\mathcal{S}$ of the underlying protocol Σ, yielding $\alpha \leftarrow \mathcal{S}(x, \beta, \gamma, 1^k)$, and later append $r \xleftarrow{\$} \{0,1\}^*$ to α and $z = f(r)$ to γ by itself. □

C Proof of Proposition 1

The proof is similar to the proof of [41, Theorem 2]. Consider an adversary $\mathcal{A}$ having access to $(\mathcal{S}_1, \mathcal{S}_2')$ that outputs a pair (x, π) which makes the verifier $\mathcal{V}'$ of the composed system accept, i.e. $\mathcal{V}'^{\mathcal{S}_1}(x, \pi) = 1$. Without loss of generality, we can assume that $\mathcal{A}$ consists of n machines $\mathcal{A}_i$, each machine running the i-th execution in the composed system. For $i \geq 2$, $\mathcal{A}_i$ expects some state st from $\mathcal{A}_{i-1}$; such a state includes a proof π and the internal coin tosses ρ of $\mathcal{A}_{i-1}$. Let $\mathcal{E}$ be the extractor for $(\mathcal{P}, \mathcal{V})$. We construct an extractor $\mathcal{E}'$ for $(\mathcal{P}', \mathcal{V}')$ as follows. $\mathcal{E}'$ chooses an index $i \in [n]$ uniformly at random and internally simulates the first $i-1$ executions of the composed system (with $\mathcal{A}_1, \ldots, \mathcal{A}_{i-1}$). Let $st = (\pi, \rho)$ be the state that $\mathcal{A}_{i-1}$ passes to $\mathcal{A}_i$. $\mathcal{E}'$ computes $w \leftarrow \mathcal{E}_{\mathcal{A}_i}(x, st) = \mathcal{E}_{\mathcal{A}_i}(x, \pi; \rho)$.

Denote with a_i the probability that $\mathcal{A}$ is successful in the first i executions. (Note that in particular a_{i-1} is the probability that in an execution of $\mathcal{E}'$ the internal simulation of the first $i-1$ executions succeeds.) Let c_i denote the probability that the i-th execution of the composed system succeeds, conditioned on the event that the first $i-1$ executions succeed. We have $a_0 = 1$ and $a_i = c_i \cdot a_{i-1}$ for all $i = 1, \ldots, n$.

Denote with ext' the probability that $\mathcal{E}'$ extracts a witness and with acc' the probability that $\mathcal{A}$ is successful in the composed system. Moreover, for some fixed index i, let ext'_i be the probability that $\mathcal{E}'$ succeeds given that index i was chosen. Hence, we must have $\mathrm{ext}' = \sum_{i=1}^{n} \frac{1}{n}\mathrm{ext}'_i \geq \max_{1 \leq i \leq n} \frac{1}{n}\mathrm{ext}'_i$.

We proceed to bound ext'_i. Denote with D_{i-1} the probability distribution of the state output by $\mathcal{A}_{i-1}$ conditioned on the fact that the first $i-1$ executions were successful. Clearly, the probability that $\mathcal{E}'$ succeeds conditioned on the event that the first $i-1$ executions were successful is $\mathrm{E}[\mathrm{ext}_i(st)]$, where $\mathrm{ext}_i(st)$ denotes the probability that $\mathcal{E}_{\mathcal{A}_i}(x, st)$ outputs a witness. Hence, $\mathrm{ext}'_i = a_{i-1}\mathrm{E}[\mathrm{ext}_i(st)]$, where st is distributed accordingly to D_{i-1}. Using the hypothesis that $(\mathcal{P}, \mathcal{V})$ has extraction error ν, we get that there exists a constant $d > 0$ and a polynomial p such that $\mathrm{ext}_i(st) \geq \frac{1}{p}(\mathrm{acc}_i(st) - \nu)^d$ for all st, where $\mathrm{acc}_i(st)$ denotes the probability that $\mathcal{A}_i$ is successful using state st. Putting all together, we can conclude:

$$\begin{aligned} \mathrm{ext}'_i &= a_{i-1}\mathrm{E}\left[\mathrm{ext}_i(st)\right] \geq a_{i-1} \cdot \mathrm{E}\left[\frac{1}{p}(\mathrm{acc}_i(st) - \nu)^d\right] \\ &\geq \frac{a_{i-1}}{p}\left(\mathrm{E}\left[\mathrm{acc}_i(st)\right] - \nu\right)^d = \frac{a_{i-1}}{p}(c_i - \nu)^d, \end{aligned}$$

where the second inequality in the chain is obtained by applying Jensen's inequality $\mathrm{E}[f(X)] \geq f(\mathrm{E}[X])$ (for any random variable X and convex function f).

Let now $\delta = (\mathrm{acc}' - \nu^n)$. Assume that $\delta > 0$ otherwise there is nothing to prove.

Claim 1. *There exists an index $i \in [n]$ such that*

$$a_{i-1} < \nu^{i-1} + (i-1)\frac{\delta}{n} \qquad \textit{and} \qquad a_i \geq \nu^i + i\frac{\delta}{n}.$$

Proof of claim. We need to prove that there exists some $i \in [n]$ such that

$$a_{i-1} - a_i \leq \nu^{i-1} + (i-1)\frac{\delta}{n} - \nu^i - i\frac{\delta}{n} = \nu^{i-1} - \nu^i - \frac{\delta}{n}.$$

Assume, towards a contradiction, that for all indexes $i \in [n]$ we have $a_{i-1} - a_i > \nu^{i-1} - \nu^i - \frac{\delta}{n}$. Then, we get

$$\begin{aligned} a_0 - a_n = \sum_{i=1}^{n}(a_{i-1} - a_i) > \sum_{i=1}^{n}\left(\nu^{i-1} - \nu^i - \frac{\delta}{n}\right) = -\delta + \sum_{i=1}^{n}(\nu^{i-1} - \nu^i) \\ = \nu^n - \mathrm{acc}' + 1 - \nu^n = 1 - \mathrm{acc}'. \end{aligned}$$

However this is impossible, because by construction $a_0 - a_n = 1 - \mathrm{acc}'$. □

For the index i of the above claim, we have

$$a_{i-1}(c_i - \nu) = a_i - a_{i-1}\nu \geq \left(\nu^i + i\frac{\delta}{n}\right) - \left(\nu^i - (i-1)\frac{\delta}{n}\right) = \frac{\delta}{n},$$

and hence

$$\begin{aligned}\mathrm{ext}' &\geq \frac{1}{n}\max_{1\leq i\leq n}\frac{a_{i-1}}{p}(c_i-\nu)^d \geq \frac{1}{n}\max_{1\leq i\leq n}\frac{a_{i-1}^d}{p}(c_i-\nu)^d \\ &\geq \frac{1}{n}\max_{1\leq i\leq n}\left(\frac{\delta}{n}\right)^d = \frac{1}{p\cdot n^{d+1}}(\mathrm{acc}'-\nu^n)^d.\end{aligned}$$

Since $p \cdot n^{d+1} = p'$, the proposition follows.

D Details Omitted from Applications

D.1 Definitions of Leakage-Resilient Primitives

Leakage-Resilient Hard Relations and Signatures. For completeness, we recall some definitions from [19] that we mention in section 5.1.

Definition 9 (Leakage-resilient hard relation)**.** *A binary relation* $\mathcal{R} \subset \{0,1\}^* \times \{0,1\}^*$ *is called* λ-leakage-resilient hard relation *if: (i) there exists an efficient sampling algorithm* $\mathsf{Gen}_{\mathcal{R}}$ *such that for any pair* $(pk, sk) \leftarrow \mathsf{Gen}_{\mathcal{R}}(1^k)$*, it holds* $\mathcal{R}(pk, sk) = 1$*, (ii) deciding whether a pair satisfies* $\mathcal{R}(pk, sk) = 1$ *or not can be done in polynomial time, (iii) any efficient adversary* $\mathcal{A}$ *has negligible advantage in winning the experiment defined below:*

Experiment $\mathsf{Exp}^{\text{LK-REL}}_{\mathcal{R},\mathcal{A}}(1^k)$

$(pk, sk) \leftarrow \mathsf{Gen}_{\mathcal{R}}(1^k)$

$sk^\star \leftarrow \mathcal{A}^{\mathcal{O}^\lambda_{sk}(\cdot)}(pk)$

Output 1 *if* $\mathcal{R}(pk, sk^\star) = 1$

that is, for any PPT $\mathcal{A}$*, it holds:*

$$\mathrm{Prob}[\mathsf{Exp}^{\text{LK-REL}}_{\mathcal{R},\mathcal{A}}(1^k) = 1] \leq negl(k)$$

Definition 10 (Leakage-resilient signature)**.** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrf})$ *be a signature scheme. The following experiment models an unforgeability game where the adversary gets leakage information on the signing key.*

Experiment $\mathsf{Exp}^{\text{LKR-SIGN}}_{\Pi,\mathcal{A}}(1^k)$

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$

$(m^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}(sk,\cdot),\mathcal{O}^\lambda_{sk}(\cdot)}(pk)$

Output 1 *if and only if:*

1. $\mathsf{Vrf}(pk, m^\star, \sigma^\star) = 1$
2. $(m^\star, \sigma^\star) \notin \mathcal{T}$

In the experiment above, $\mathcal{T}$ *is a list containing queries to* Sign *and corresponding answers. A signature scheme* Π *is* λ-leakage resilient *if any PPT adversary* $\mathcal{A}$ *has only negligible advantage in winning the unforgeability game:*

$$\mathrm{Prob}[\mathsf{Exp}^{\text{LKR-SIGN}}_{\Pi,\mathcal{A}}(1^k) = 1] \leq negl(k)$$

Leakage-Resilient Public-Key Encryption. We recall the definitions of CPA and CCA security against λ-key-leakage attacks from [35]. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key encryption scheme and denote with $\mathcal{PK}_k$, $\mathcal{SK}_k$ the sets of public/secret keys that are produced by $\mathsf{Gen}(1^k)$. We define an oracle $\mathcal{O}^{\lambda}_{sk}$ depending on a secret key $sk \in \mathcal{SK}_k$, which takes as input (the description of) functions $f_i : \mathcal{SK}_k \to \{0,1\}^{\lambda_i}$. An oracle machine $\mathcal{A}$ is a λ-key-leakage adversary if $\mathcal{A}$ has access to $\mathcal{O}^{\lambda}_{sk}$ and the total leakage is bounded by $\sum_i \lambda_i \leq \lambda$. Consider the following experiments:

```
**Experiment** $\mathsf{Exp}^{\text{LKG-CPA}}_{\Pi,\mathcal{A}}(1^k)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathcal{O}^{\lambda}_{sk}(\cdot)}(pk)$
$c_b \leftarrow \mathsf{Enc}(pk, m_b)$ for $b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}_1(c_b)$
Output 1 if and only if:
    1. $b' = b$
    2. $|m_0| = |m_1|$
```

```
**Experiment** $\mathsf{Exp}^{\text{LKG-CCA}}_{\Pi,\mathcal{A}}(1^k)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{Dec}(sk,\cdot),\mathcal{O}^{\lambda}_{sk}(\cdot)}(pk)$
$c_b \leftarrow \mathsf{Enc}(pk, m_b)$ for $b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(c_b)$
Output 1 if and only if:
    1. $b' = b$
    2. $|m_0| = |m_1|$
    3. $c_b$ is never submitted to $\mathsf{Dec}(sk, \cdot)$
```

We stress that in the experiments above, $\mathcal{A}$ is not allowed to query the leakage oracle after seeing the challenge ciphertext. (In fact, a single bit of leakage on the challenge ciphertext would allow her to win the game with probability 1.)

Definition 11 (ATK-secure encryption against λ-key-leakage attacks)**.** *For* ATK $\in$ {CPA, CCA}, *we say a public key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* ATK*-secure against λ-key-leakage attacks if, for every PPT λ-key-leakage adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ as above, we have:*

$$\text{Prob}[\mathsf{Exp}^{\text{LKG-ATK}}_{\Pi,\mathcal{A}}(1^k) = 1] \leq \frac{1}{2} + negl(k).$$

D.2 Proof of Theorem 4

This proof follows the same strategy given in [19], Theorem 4.3, with minor changes. We consider the following two games:

Game 0: This is the unforgeability game of Definition 10 tailored for leakage-resilient signatures described in 5.1.

Game 1: In this game, we change the way in which the signing oracle answers to $\mathcal{A}$'s queries. Instead of giving valid proofs π_i, it answers signature queries m_i with simulated proofs $\mathcal{S}(pk, m_i)$. Notice that we use a simplified notation and denote $\mathcal{S}$ the simulator that combines $\mathcal{S}_1$ (which answers queries to the random oracle) and $\mathcal{S}_2$ (which generates the proofs) from Definition 4.

In Game 1, the winning condition of $\mathcal{A}$ is still to compute a valid forgery, i.e., to compute $(m^\star, \sigma^\star)$ such that $\mathsf{Vrf}(pk, m^\star, \sigma^\star) = 1$. The transition from Game 0 to Game 1 is based on the indistinguishability between proofs produced by the prover and proofs computed by the simulator. For the unbounded non-interactive zero-knowledge property of the scheme, the success probability in the two games turns out to be negligibly close. Indeed, the adversary does not notice whether she is talking to the signing oracle Sign_{sk} or to the simulator $\mathcal{S}$. Suppose for sake of contradiction that the success probability of $\mathcal{A}$, that we denote ϵ, is non-negligible (wlog, we bound the success probability of $\mathcal{A}$ for Game 1).

We next construct an adversary $\mathcal{B}$ that runs $\mathcal{A}$ as a subroutine and breaks the security of the leakage-resilient hard relation $\mathcal{R}$. Adversary $\mathcal{B}$ behaves as follows: after receiving pk as the challenge for the hard relation, it simulates the environment for $\mathcal{A}$ and answers her oracle queries. To this end, $\mathcal{B}$ forwards $\mathcal{A}$'s leakage queries, addressed to oracle $\mathcal{O}_{sk}^{\lambda}$, to its own oracle $\mathcal{O}_{sk}^{2\lambda}$, and returns the answer to $\mathcal{A}$. To answer signing queries for m_i, $\mathcal{B}$ creates simulated proofs $\sigma_i \leftarrow \mathcal{S}(pk, m_i)$. When $\mathcal{A}$ outputs a forgery $(m^\star, \sigma^\star)$, adversary $\mathcal{B}$ runs the extractor $\mathcal{E}_{\mathcal{A}}(pk, \sigma^\star)$ to obtain $sk^\star$, hence outputs this value as a witness for pk. Notice that by running the extractor $\mathcal{E}_{\mathcal{A}}(pk, \sigma^\star)$, we need to rewind $\mathcal{A}$ which requires to simulate its leakage queries again. Hence, the need for a 2λ leakage oracle (cf. Theorem 3).

It remains to analyze the success probability of $\mathcal{B}$ in breaking the security of the leakage-resilient hard relation $\mathcal{R}(pk, sk)$. Let Win be the event that $\mathcal{A}$ wins in Game 1, that is she outputs a pair $(m^\star, \sigma^\star)$ accepted by $\mathcal{V}^H$, and let Ext be the event that the extractor $\mathcal{E}_{\mathcal{A}}$ computes a valid witness $sk^\star$ (cf. Definition 6). We have

$$\begin{aligned} \mathrm{Prob}[\mathsf{Exp}_{\mathcal{R},\mathcal{B}}^{\text{LK-REL}}(1^k) = 1] &\geq \mathrm{Prob}[\mathsf{Win} \wedge \mathsf{Ext}] && (1)\\ &= \mathrm{Prob}[\mathsf{Ext}|\mathsf{Win}] \cdot \mathrm{Prob}[\mathsf{Win}] && (2)\\ &\geq \frac{(1 - negl(k))^d}{p(k)} \cdot \epsilon && (3) \end{aligned}$$

In the above, Eq. (1) uses the fact that if $\mathcal{A}$ produces a valid forgery $(m, \sigma^\star)$ and the extractor $\mathcal{E}_{\mathcal{A}}$ succeeds and outputs a valid witness $sk^\star$ from such forgery, then $\mathcal{B}$ wins: indeed, it simply returns $sk^\star$. This implies that:

$$\mathrm{Prob}[\mathsf{Exp}_{\mathcal{R},\mathcal{B}}^{\text{LK-REL}}(1^k) = 1] \geq \mathrm{Prob}[\mathsf{Win} \wedge \mathsf{Ext}]$$

In Eq. (2), we use the definition of conditional probability. In Eq. (3) we use the fact that $(\mathcal{P}^H, \mathcal{V}^H)$ has negligible extraction error and that conditioned on Win the verification always succeeds, i.e., acc = 1 in Theorem 3. Finally, in Eq. (3) we use our assumption that $\epsilon := \mathrm{Prob}[\mathsf{Win}]$ is non-negligible, which implies that for any polynomial p and constant d, $\frac{\epsilon}{p(k)} \cdot (1 - negl(k))^d$ is also non-negligible. This yields a contradiction to the security of the leakage-resilient hard relation $\mathcal{R}$ and concludes the proof.

D.3 Revisiting Naor-Segev in the ROM

Naor-Yung paradigm. In the random oracle model, the Naor-Yung (NY) construction considers two encryption schemes with CPA-security $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ and a NIZK proof system $(\mathcal{P}^H, \mathcal{V}^H)$ for the following language:

$$\mathcal{L}_{NY} = \left\{(c, c', pk, pk') :\ \exists m, r, r' \text{ s.t. } c = \mathsf{Enc}(pk, m; r),\ c' = \mathsf{Enc}'(pk', m; r')\right\},$$

where the public keys pk and pk' are generated by Gen and Gen' respectively. The new encryption scheme $\Pi^* = (\mathsf{Gen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ is as follows:

Key generation: Runs $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, $(pk', sk') \leftarrow \mathsf{Gen}'(1^k)$, and outputs keys $\vec{pk} = (pk, pk')$ and $\vec{sk} = (sk, sk')$.

Encryption: Computes ciphertexts $c = \mathsf{Enc}(pk, m; r)$ and $c' = \mathsf{Enc}'(pk', m; r')$, invokes the prover $\mathcal{P}^H$ on public input (c, c', pk, pk') and private input (m, r, r') to get a NIZK proof π for $(c, c', pk, pk') \in \mathcal{L}_{NY}$.

Decryption: Checks the validity of proof π by calling the (public) verification procedure $\mathcal{V}^H((c, c'), \pi)$. If π is accepted, uses decryption algorithm $\mathsf{Dec}(sk, \cdot)$ on ciphertext c.

The extension of the Naor-Yung paradigm of twin encryption to the key-leakage framework has been proven by Naor and Segev [35] for NIZK proofs in the CRS model. They show that, when the NY paradigm is applied to leakage-resilient schemes, the resulting CCA-secure encryption scheme is still leakage-resilient. More precisely, starting from an encryption schemes Π which is CPA-secure against λ-key-leakage attacks, one gets a CCA-secure encryption scheme Π^* secure against λ-key-leakage attacks. The proof of Naor and Segev is similar to the one appeared in [34], except for the presence of a leakage oracle $\mathcal{O}^{\lambda}_{sk}$. The intuition is that a ciphertext (c, c', π) can be decrypted by using *only one* secret key. Indeed, given knowledge of one of the two secret keys, decryption can be carried out by checking the validity of π (the verification procedure is public) and decrypting the corresponding ciphertext with the known secret key. In the leakage setting, the CPA-attacker has also to answer to leakage queries made by the CCA-attacker. Again, a single secret key is sufficient for this purpose.

Below, we prove the ROM-equivalent of the result above when NIZKs are built in the random oracle model (in particular, when using the Fiat-Shamir transform). The only potential issue is that, since the random oracle is public, leakage functions might also depend on it. However, since we make use of Fiat-Shamir NIZKs, this fact does not affect the proof because the random oracle takes as input only public data (namely, the statement x to be proven and the commitment α in the NIZK proof).

Theorem 5 (ROM equivalent of [35]). *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *and* $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *be two public-key encryption schemes secure against* λ*-key-leakage chosen-plaintext attacks, and let* $(\mathcal{P}^H, \mathcal{V}^H)$ *be a simulation-sound NIZK proof system for the Naor-Yung language* $\mathcal{L}_{NY}$ *associated to* (Π, Π'). *In the random oracle model, the encryption scheme* Π^* *obtained via the Naor-Yung paradigm is secure against adaptive* λ*-key-leakage chosen-ciphertext attacks.*

In the random oracle setting, the proof presented in [35] can be mirrored in a straightforward way: it suffices to simply replace the CRS-based NIZK with an analogous NIZK proof system in the ROM. In fact, irrespective of the model being considered, the proof π can be computed by the encryption algorithm Enc^* and validated by the decryption procedure Dec^* by invoking, in turn, prover and verifier. The presence of random oracles does not change the behavior of the leakage oracle. Nevertheless, we propose a new approach which traces out the proof appeared in [14], in the context of KDM security.

Proof. Consider the LKG-CCA experiment adapted for the NY construction:

```
Experiment $\mathsf{Exp}^{\text{LKG-CCA}}_{\Pi^*,\mathcal{A}}(1^k)$:
$(pk, pk', sk, sk') \leftarrow \mathsf{Gen}^*(1^k)$; $b \xleftarrow{\$} \{0,1\}$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{Dec}^*(sk,\cdot),\mathcal{O}^{\lambda}_{sk}(\cdot)}(pk, pk')$
$c_b = \mathsf{Enc}(pk, m_b; r)$; $c'_b \leftarrow \mathsf{Enc}'(pk', m_b; r')$
$\pi \leftarrow \mathcal{P}^H((c_b, c'_b, pk, pk'), (m_b, r, r'))$; $\vec{c}_b := (c_b, c'_b, \pi)$
$b' \leftarrow \mathcal{A}_1^{\mathsf{Dec}^*(sk,\cdot),\mathcal{O}^{\lambda}_{sk}(\cdot)}(pk, pk', \vec{c}_b)$
Output 1 if and only if:
    1. $b' = b$
    2. $|m_0| = |m_1|$
    3. $c_b$ not asked to $\mathsf{Dec}^*(sk, \cdot)$

$\mathsf{Dec}^*(c, c', \pi)$:
if $\mathcal{V}^H((c, c'), \pi) = 1$
$m = \mathsf{Dec}(sk, c)$
Return $m$

$\mathcal{O}^{\lambda}_{sk}(f)$:
Return $f(sk)$
```

We stress that, even if not explicitly written (for better readability), the leakage functions f may depend on the random oracle H. However, such a dependence does not give any significant advantage to the adversary. In what follows, we derive a series of games whose outcomes cannot be mutually distinguished, as long as one does not violate the hypothesis of the theorem.

Game 0. We consider the LKG-CCA experiment defined above, in the case the random bit is $b = 0$. Observe that the challenge ciphertext (c_b, c'_b, π) contains encryptions of the same message m_0.

Game 1. This game differs from the previous one in that the proof π attached to the challenge ciphertext is not properly generated by the prover, but computed by the ZK simulator.

Game 2. As in Game 1, but in the challenge phase the ciphertext c'_b is computed as $c'_b \leftarrow \mathsf{Enc}'(pk', m_1)$. As a consequence, the attached proof π is a fake one, since c_b and c'_b encrypt m_0 and m_1 respectively.

Game 3. This game is defined as Game 2 but the decryption oracle uses the secret key sk' instead of sk, i.e. answers to queries (c, c', π) by computing $\mathsf{Dec}'(sk', c')$.

Game 4. Again, we change the encryption oracle in such a way that the first ciphertext c encrypts the message m_1. This means that the challenge ciphertext contains $\mathsf{Enc}(pk, m_1)$ and $\mathsf{Enc}'(pk', m_1)$, both encryptions of the same message m_1, hence it is a valid ciphertext.

Game 5. As in Game 4, but we restore the prover $\mathcal{P}^H$ to compute proofs.

Game 6. Game 6 differs from the previous one in that the decryption oracle restart to invoke $\mathsf{Dec}(sk, \cdot)$ on input the first ciphertext c (and the leakage oracle also uses sk). This is the experiment LKG-CCA for $b = 1$.

Denote by S_i the adversary success in Game i, for $i = 0, 1, \ldots, 6$. Observe that $|\text{Prob}[S_0] - \text{Prob}[S_6]|$ bounds the advantage of a (λ-key-leakage) CCA-adversary attacking the scheme Π^*. We show that the difference $|\text{Prob}[S_i] - \text{Prob}[S_{i+1}]|$ is negligible for any $i = 1, \ldots 5$, which implies the former.

$0 \to 1$ The transition between games 0 and 1 is based on the indistinguishability of NIZK proofs properly computed from proofs generated by the simulator. Indeed, if the probabilities of winning in the two games were not negligibly close, one could build an efficient distinguisher able to detect whether a proof comes from the prover $\mathcal{P}^H$ or from the simulator $\mathcal{S}$.

$1 \to 2$ CPA-security of the encryption scheme Π' means that no efficient adversary can distinguish between encryptions of different messages under the same public key. In particular, encryption of different messages under the same public key pk' are indistinguishable. Game 1 and Game 2 differ only in the way the challenge ciphertext (c_b, c'_b, π) is generated: ciphertext c'_b equals $\mathsf{Enc}'(pk', m_0)$ in the first case, while in Game 2 it is $\mathsf{Enc}'(pk', m_1)$. Thus, any PPT distinguisher who recognizes which games is involved with could be directly used to discern which one of the two plaitexts has been encrypted.

$2 \to 3$ The difference between Game 2 and Game 3 lies in the way the decryption oracle answers adversarial queries (c, c', π), namely by decrypting c with key sk or c' with key sk', respectively. The only chance to tell apart the two experiments is to query the decryption oracle with an invalid ciphertext (c, c', π) such that c and c' are the encryptions of different messages. This would require to produce an accepting proof $\pi^\star$ for a pair $(c, c') \notin \mathcal{L}_{NY}$ (the decryption oracle checks the validity of the proof before allowing decryption). Observe that in both games, the adversary actually receives a fake (simulated) accepting proof for a false statement (since one of the ciphertext encrypts m_0 and the other encrypts m_1), and she might potentially gain some information from that proof. However, the simulation soundness of the proof system forbids her to produce a new accepting proof for a false statement.

3 → 4 The transition between Game 3 and Game 4 is equivalent to the one involving games 1 and 2, that we have already discussed. Since the scheme Π is CPA-secure, indistinguishability of ciphertexts $\mathsf{Enc}(pk, m_0)$ and $\mathsf{Enc}(pk, m_1)$ implies that the probabilities of succeeding in the two games are negligibly close.

4 → 5 Another symmetry makes similar the transition from Game 4 to Game 5 with the one between games 0 and 1: indistinguishability follows from the zero-knowledge property of the non-interactive proof system.

5 → 6 Game 6 differs from Game 5 in that the decryption oracle restarts to decrypt ciphertext c by using secret key sk, as in the original experiment. To notice the difference, an efficient distinguisher should be able to produce an invalid ciphertext (c, c', π), hence an accepted proof π for a false statement, as for the transition between games 2 and 3. We have already seen that no adversary is allowed to do so, otherwise the simulation soundness of the proof system would be broken. Moreover, in games 5 and 6, the challenge ciphertext contains two encryptions of the same message m_1, that means it is a valid ciphertext. The adversary is not even allowed to get fake simulated proof, thus producing fresh fake proofs would in fact violate the soundness property of the proof system.

□

D.4 Σ-Protocol for BHHO

Proof. Completeness condition is trivially satisfied. In order to prove that the protocol is special sound, let $\pi_1 = (\vec{\alpha}, \beta_1, \vec{\gamma_1})$ and $\pi_2 = (\vec{\alpha}, \beta_2, \vec{\gamma_2})$ be two different accepted proofs for a given theorem $x = (c, pk, c', pk')$, with challenges $\beta_1 \neq \beta_2$. We show how to extract a witness for $x \in \mathcal{L}$. The validity of both proofs implies that, for any $i = 1, \ldots, \ell$, we have:

$$\alpha_i = g_i^{\gamma_1} \cdot c_i^{\beta_1} = g_i^{\gamma_2} \cdot c_i^{\beta_2}$$
$$\alpha'_i = g_i^{\gamma'_1} \cdot (c'_i)^{-\beta_1} = g_i^{\gamma'_2} \cdot (c'_i)^{-\beta_2}$$

and

$$\alpha'' = h^{\gamma_1} \cdot (h')^{\gamma'_1} \cdot (c_{\ell+1} \cdot (c'_{\ell+1})^{-1})^{\beta_1} = h^{\gamma_2} \cdot (h')^{\gamma'_2} \cdot (c_{\ell+1} \cdot (c'_{\ell+1})^{-1})^{\beta_2}.$$

From the first pair of equations we can easily compute $g_i^{(\gamma_1-\gamma_2)} = c_i^{(\beta_1-\beta_2)}$ and $g_i^{(\gamma'_1-\gamma'_2)} = (c'_i)^{(\beta_2-\beta_1)}$. Hence, since the challenges are distinct, we can invert $(\beta_2 - \beta_1)$, obtaining $c_i = g_i^{(\gamma_1-\gamma_2)(\beta_1-\beta_2)^{-1}}$ and $c'_i = g_i^{(\gamma'_1-\gamma'_2)(\beta_2-\beta_1)^{-1}}$. Finally, setting $\rho = (\gamma_1 - \gamma_2)(\beta_1 - \beta_2)^{-1}$ and $\rho' = (\gamma'_1 - \gamma'_2)(\beta_2 - \beta_1)^{-1}$ yields a witness $w = (\rho, \rho')$ for the statement x. Indeed, for any index $i = 1, \ldots, \ell$ it holds $c_i = g_i^{\rho}$, $c'_i = g_i^{\rho'}$ and

$$c_{\ell+1} \cdot (c'_{\ell+1})^{-1} = h^{\rho} \cdot (h^{\rho'})^{-1}.$$

The last equations prove that the ciphertexts encrypt the same message, that is to say the statement x is in $\mathcal{L}$.

It's not difficult to see that the protocol is not only HVZK, but also SS-HVZK. Indeed, computing the response $\vec{\gamma}$ as the third move of the prover is the same as choosing it at random in $\mathbb{Z}_q \times \mathbb{Z}_q$, because both s and β, like so s' and β', are chosen at random. Once a response $\vec{\gamma}$ is chosen, for any challenge β we can easily compute a commitment $\vec{\alpha}$ in such a way that the triplet $(\vec{\alpha}, \beta, \vec{\gamma})$ satisfies the check: this is a consequences of the special form the verification procedure has, that directly shows how to compute $\vec{\alpha}$. Given this, a probabilistic polynomial-time algorithm which acts as an SS-HVZK simulator can be described. □

D.5 Revisiting Camenisch-Chandran-Shoup in the ROM

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme and denote with $\mathcal{PK}_k$, $\mathcal{SK}_k$ the sets of public/secret keys that are produced by $\mathsf{Gen}(1^k)$. Also, for integer n, denote with $\mathsf{Gen}^{(n)}(1^k)$ the algorithm which outputs two vectors $(\vec{pk}, \vec{sk})$ containing n public/secret keys. Given the message space $\mathcal{M}$, consider the family of functions:

$$\mathcal{F}^{(n)} \subset \{f:\ \mathcal{SK}_k^n \to \mathcal{M}\} \qquad\qquad \mathcal{F} = \bigcup_{n=1}^{\infty} \mathcal{F}^{(n)}.$$

We define an oracle $\Delta_b(i, f)$ depending on a bit $b \in \{0,1\}$ and taking as input an index $i \in [n]$ and a function $f \in \mathcal{F}$. Whenever $b = 1$, the oracle returns $\mathsf{Enc}(pk_i, f(\vec{sk}))$ (i.e., an encryption of $f(\vec{sk})$ under the i-th public key in $\vec{pk}$); otherwise it returns $\mathsf{Enc}(pk_i, 0)$ (i.e., the encryption of a fixed message in $\mathcal{M}$). Consider the following experiments:

```
Experiment $\mathsf{Exp}^{\text{KDM-CPA}}_{\Pi,\mathcal{A},\mathcal{F}}(1^k)$
$(\vec{pk}, \vec{sk}) \leftarrow \mathsf{Gen}^{(n)}(1^k)$
$b' \leftarrow \mathcal{A}^{\Delta_b(\cdot,\cdot)}(\vec{pk})$
Output 1 if and only if:
    1. $b' = b$
```

```
Experiment $\mathsf{Exp}^{\text{KDM-CCA}}_{\Pi,\mathcal{A},\mathcal{F}}(1^k)$
$(\vec{pk}, \vec{sk}) \leftarrow \mathsf{Gen}^{(n)}(1^k)$
$b' \leftarrow \mathcal{A}^{\Delta_b(\cdot,\cdot),\mathsf{Dec}(\cdot,\cdot)}(\vec{pk})$
Output 1 if and only if:
    1. $b' = b$
    2. $\mathsf{Dec}(i, \cdot)$ not queried on $c \leftarrow \Delta_b(i, \cdot)$
```

Note that oracle $\mathsf{Dec}(i, c)$ outputs $\mathsf{Dec}(sk_i, c)$.

Definition 12 (KDM-ATK secure encryption)**.** *For* ATK $\in$ {CPA, CCA}*, we say a public-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* KDM[$\mathcal{F}$]-ATK *secure if, for every probabilistic polynomial-time adversary* $\mathcal{A}$ *as above and for all n, we have:*

$$\text{Prob}[\mathsf{Exp}^{\text{KDM-ATK}}_{\Pi,\mathcal{A},\mathcal{F}}(1^k) = 1] \leq \frac{1}{2} + negl(k).$$

Simulation soundness and CCA-KDM security. As shown in [14, Appendix A.1], a variation of the Naor-Yung paradigm instantiated with a simulation-sound NIZK can leverage CPA-security to CCA-security in the context of KDM security, provided that one of the two encryption scheme already satisfy KDM-CPA security. The basic elements are an encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with KDM-CPA security with respect a given class of function $\mathcal{F}$, a CPA-secure encryption scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ and a simulation sound NIZK proof system $(\mathcal{P}^H, \mathcal{V}^H)$ for the NY language associated to (Π, Π'). The combined encryption scheme $\Pi^* = (\mathsf{Gen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ is defined as in the original NY paradigm of twin encryption (see Section D.3).

The statement below is the ROM-equivalent of [14]. The proof strategy is essentially the same already used in the proof of Theorem 5, the only difference being the instantiation of Π with a KDM-CPA scheme.

Theorem 6 (ROM-equivalent of [14])**.** *Let* Π *be a KDM-CPA secure encryption scheme with respect to a class of function* $\mathcal{F}$*,* Π' *be a CPA-secure encryption scheme and let* $(\mathcal{P}^H, \mathcal{V}^H)$ *be a simulation-sound NIZK proof system for the Naor-Yung language* $\mathcal{L}_{NY}$ *associated to* (Π, Π')*. In the random oracle model, the encryption scheme* Π^* *obtained via the Naor-Yung paradigm is KDM-CCA secure with respect to the class of function* $\mathcal{F}$*.*

Proof. We only sketch the proof. Consider the KDM-CCA experiment adapted for the NY construction:

```
**Experiment** $\mathsf{Exp}^{\text{KDM-CCA}}_{\Pi^*,\mathcal{A},\mathcal{F}}(1^k)$:
$(\vec{pk}, \vec{sk}) \leftarrow \mathsf{Gen}^*(1^k)$
$b' \leftarrow \mathcal{A}^{\Delta_b(\cdot,\cdot),\mathsf{Dec}^*(\cdot,\cdot)}(\vec{pk})$
Output 1 if and only if:
    1. $b' = b$
    2. $\mathsf{Dec}(i, \cdot)$ not queried on $c \leftarrow \Delta_b(i, \cdot)$
```

```
$\mathsf{Dec}^*(i, (c, c', \pi))$:
if $\mathcal{V}^H((c, c'), \pi) = 1$
$m = \mathsf{Dec}(sk_i, c)$
Return $m$

$\Delta_b(i, f)$ :
if $b = 1$ output $\mathsf{Enc}^*(pk_i, f(\vec{sk}))$
else output $\mathsf{Enc}^*(pk_i, 0^{|f(\vec{sk})|})$
```

Note that the i-th key in vector $\vec{pk}$ has a type (pk_i, pk'_i) (one key for Π and the other one for Π'), and the same holds for $\vec{sk}$. Consider the following games:

Game 0. Is the KDM-CCA experiment defined above, when $b = 0$.

Game 1. This game differs from the previous one in that the proof π and queries to the random oracle are not computed using $\mathcal{P}^H$ and H respectively, but are produced by the ZK simulator. *This is indistinguishable by the unbounded zero-knowledge property of the proof system.*

Game 2. As in Game 1, but encryption queries for index i are answered by computing the second ciphertext as $c' \leftarrow \mathsf{Enc}'(pk'_i, 0^{|f(\vec{sk})|})$. (Hence, the attached proof π is a simulated proof of a false statement.) *This is indistinguishable due to the CPA-security of* Π'.

Game 3. This game is defined as Game 2 but decryption queries for index i are answered by using secret key sk'_i instead of sk_i, i.e. answer to queries $(i, (c, c', \pi))$ by computing $\mathsf{Dec}'(sk'_i, c')$ whenever π is accepted. *This is indistinguishable due to the simulation soundness of the proof system.*

Game 4. We change the encryption oracle in such a way that the first ciphertext is computed as $c \leftarrow \mathsf{Enc}(pk_i, 0^{|f(\vec{sk})|})$. *This is indistinguishable due to the KDM-CPA security of* Π.

Game 5. As in Game 4, but we restore the prover $\mathcal{P}^H$ to compute proofs. *This is indistinguishable by the unbounded zero-knowledge property of the proof system.*

Game 6. Game 6 differs from the previous one in that the decryption oracle restart to invoke $\mathsf{Dec}(sk_i, \cdot)$ on input the first ciphertext c. *This is indistinguishable due to the soundness of the proof system.*

Note that this is exactly the experiment KDM-CCA in the case $b = 1$.

Denote by S_i the adversary success in Game i, for $i = 0, 1, \ldots, 6$. Observe that $|\text{Prob}[S_0] - \text{Prob}[S_6]|$ bounds the advantage of a KDM-CCA adversary attacking the scheme Π^*. As in the proof of Theorem 5, the difference $|\text{Prob}[S_i] - \text{Prob}[S_{i+1}]|$ is negligible for all $i = 1, \ldots 5$, which makes negligible the advantage of a KDM-CCA adversary against the scheme Π^*. The only difference here is in the scheme Π, that is KDM-CPA secure instead of simply CPA-secure. In fact, the combined encryption scheme Π^* inherits KDM-security just from the scheme Π (when we move from Game 3 to Game 4). □