Fully Collision-Resistant Chameleon-Hashes from Simpler and Post-Quantum Assumptions

David Derler[1], Stephan Krenn[2], Kai Samelin[3], and Daniel Slamanig[2]

[1] DFINITY, Zurich, Switzerland
david@dfinity.org
[2] AIT Austrian Institute of Technology, Vienna, Austria
{stephan.krenn, daniel.slamanig}@ait.ac.at
[3] Independent, Hamburg, Germany
kaispapers@gmail.com

Abstract. Chameleon-hashes are collision-resistant hash-functions parametrized by a public key. If the corresponding secret key is known, arbitrary collisions for the hash can be found. Recently, Derler et al. (PKC '20) introduced the notion of *fully* collision-resistant chameleon-hashes. Full collision-resistance requires the intractability of finding collisions, even with full-adaptive access to a collision-finding oracle. Their construction combines simulation-sound extractable (SSE) NIZKs with perfectly correct IND-CPA secure public-key encryption (PKE) schemes. We show that, instead of perfectly correct PKE, non-interactive commitment schemes are sufficient. For the first time, this gives rise to efficient instantiations from plausible post-quantum assumptions and thus candidates of chameleon-hashes with strong collision-resistance guarantees and long-term security guarantees. On the more theoretical side, our results relax the requirement to not being dependent on public-key encryption.

1 Introduction

Chameleon-hashes (CHs) are collision-resistant hash-functions parametrized by a public key. Knowledge of the corresponding secret key allows finding arbitrary collisions. Chameleon-hashes were initially introduced by Krawczyk and Rabin [KR00]. Similar underlying ideas even date back to the introduction of "trapdoor commitments" by Brassard et al. [BCC88]. They are an integral part of many cryptographic constructions, both in theory and practice. For instance, CHs find usage in on/offline signatures [CZSM07, EGM96, ST01], to generically lift non-adaptively secure signature schemes to adaptively secure ones [HW09, ST01], or as a building block for tightly-secure signatures [BKKP15]. Likewise, they find applications in strong one-time signatures [Moh10], the construction of IND-CCA secure public-key encryption [Zha07] or to extend Schnorr and RSA

This is the full version of a paper which appears in Security and Cryptography for Networks - SCN 2020 - 12th Conference on Security and Cryptography for Networks, Amalfi, Italy, September 14-16, 2020. The proceedings version is available online at: https://doi.org/10.1007/978-3-030-57990-6_21

signatures to the universal designated-verifier setting [SBWP03]. CHs are also widely used in sanitizable signatures [ACdMT05, BCD$^+$17, BFF$^+$09, CDK$^+$17], i.e., signatures where a designated entity can alter certain parts of a signed message while also deriving a valid new signature for the altered message. Bellare and Ristov have shown that chameleon-hashes and Σ-protocols (meaning three-round public-coin honest-verifier zero-knowledge proofs of knowledge), are equivalent [BR08, BR14]. Likewise, several extensions such as (hierarchical) identity-based [AdM04a, AdM04b, BDD$^+$11], policy-based chameleon-hash functions [DSSS19, SS20], or multi-trapdoor CHs [CDK$^+$17, KPSS18] have been studied.

Derler et al. [DSS20] recently studied existing collision-resistance notions of CHs and introduced the notion of full collision-resistance, which is the strongest known such notion and arguably the most natural one. Compared to prior notions, their definition requires that an adversary which has full adaptive access to a collision-finding oracle cannot find any collisions that it did not receive from the oracle. For comparison, the weakest meaningful notion (those satisfied by trapdoor commitments) does not allow the adversary to see *any* collision.

Contribution. Given the wide variety of application scenarios relying on CHs as building blocks, striving to find efficient instantiations and construction paradigms, based on minimal assumptions, yet with strong security guarantees, is an important task. Our contributions are along those lines, and, in particular, include:

- A black-box construction of fully collision-resistant chameleon-hashes based on SSE NIZKs and non-interactive commitment schemes. Most importantly, this construction manages to remove the requirement to rely on public-key encryption. While this is interesting from a practical point of view as it gives more freedom for possible instantiations, it is also interesting from a theoretical perspective, as we can instantiate our constructions from primitives which require weaker assumptions. Besides that, our construction offers strong indistinguishability, a strong privacy notion recently introduced by Derler et al. [DSSS19].
- An efficient instantiation from post-quantum assumptions. In particular, we present a concrete construction from the learning parity with noise (LPN) problem. This yields the first chameleon-hash from post-quantum assumptions that provides a collision-resistance notion stronger than that provided by trapdoor commitments (e.g., the lattice-based chameleon-hash by Cash et al. [CHKP10]). We note that although the security of the used SSE NIZKs obtained from the Fiat-Shamir transform are just argued in the random oracle model (ROM), there is a recent line of works [DFMS19, LZ19, DFM20] that prove security of (SSE) NIZKs obtained via Fiat-Shamir in the quantum accessible ROM (QROM) [BDF$^+$11]. Latter gives evidence that security in the ROM based on post-quantum assumptions is a meaningful security guarantee in practice. We leave it as an interesting open question to study

the security of instantiations from other post-quantum assumptions in the QROM.

- An efficient instantiation from the discrete logarithm (DL) assumption, in contrast to the DDH assumption used by Derler et al. [DSS20].
- The new notion of randomness unforgeability of chameleon-hashes. Intuitively, it requires that the adversary cannot find new randomness for an honestly generated hash. This notion is weaker than the uniqueness notion by Camenisch et al. [CDK+17], but may find its usage in cases where neither the holder of the secret key nor the hashing party is adversarial, protecting against outsiders tempering with the generated values.

2 Preliminaries

Notation. With $\lambda \in \mathbb{N}$ we denote our security parameter. All algorithms implicitly take 1^λ as an additional input. We write $a \leftarrow_r A(x)$ if a is assigned to the output of an algorithm A with input x (and use $a \leftarrow A(x)$ if A is deterministic). An algorithm is efficient, if it runs in probabilistic polynomial time (PPT) in the length of its input. All algorithms are PPT, if not explicitly mentioned otherwise. Most algorithms may return a special error symbol $\perp \notin \{0,1\}^*$, denoting an exception. Returning output ends execution of an algorithm or an oracle. In order to make the presentation in the security proofs more compact, we occasionally use $(a, \perp) \leftarrow_r A(x)$ to indicate that the second output is either ignored or not returned by A. If S is a finite set, we write $a \leftarrow_r S$ to denote that a is chosen uniformly at random from S. $\mathcal{M}$ denotes a message space of a scheme, and we generally assume that $\mathcal{M}$ is derivable from the scheme's public parameters or its public key. For a list we require that there is an injective, and efficiently reversible, encoding, that maps the list to $\{0,1\}^*$. A function $\nu : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible, if it vanishes faster than every inverse polynomial, i.e., $\forall k \in \mathbb{N}$, $\exists n_0 \in \mathbb{N}$ such that $\nu(n) \leq n^{-k}$, $\forall n > n_0$.

2.1 One-Way Functions

A one-way function f is a function, where computing the function is easy, but reversing the function is hard.

Definition 1 (One-Way Functions). *A function $f : \{0,1\}^* \to \{0,1\}^*$ is one-way, if (1) there exists a PPT algorithm $\mathcal{A}_1$ so that for all $\forall\ x \in \{0,1\}^* : \mathcal{A}_1(x) = f(x)$, and (2) for every PPT adversary $\mathcal{A}_2$ there exists a negligible function ν such that:*

$$\Pr[x \leftarrow_r \{0,1\}^\lambda, x' \leftarrow_r \mathcal{A}_2(f(x)) : f(x) = f(x')] \leq \nu(\lambda).$$

2.2 Non-Interactive Commitment Schemes

Non-interactive commitment schemes allow one party to commit itself to a value without revealing it [Blu81]. Later, the committing party can give some opening information to the receiver to “open” the commitment.

Definition 2 (Non-Interactive Commitments). *A non-interactive commitment scheme $\mathit{\Gamma}$ is a tuple of PPT algorithms defined as follows:*

$\mathsf{ParGen}_{\mathit{\Gamma}}$. *This algorithm takes as input a security parameter λ and outputs the public parameters $\mathsf{pp}_{\mathit{\Gamma}}$:*

$$\mathsf{pp}_{\mathit{\Gamma}} \leftarrow_r \mathsf{ParGen}_{\mathit{\Gamma}}(1^\lambda)$$

$\mathsf{Commit}_{\mathit{\Gamma}}$. *This algorithm takes as input a message m, and outputs a commitment C together with corresponding opening information O:*

$$(C, O) \leftarrow_r \mathsf{Commit}_{\mathit{\Gamma}}(\mathsf{pp}_{\mathit{\Gamma}}, m)$$

$\mathsf{Open}_{\mathit{\Gamma}}$. *This deterministic algorithm takes as input a commitment C, a message m, and some opening information O. It outputs a decision $d \in \{0,1\}$:*

$$d \leftarrow \mathsf{Open}_{\mathit{\Gamma}}(\mathsf{pp}_{\mathit{\Gamma}}, C, O, m)$$

Definition 3 (Correctness). *A non-interactive commitment scheme $\mathit{\Gamma}$ is said to be (perfectly) correct, if for all $\lambda \in \mathbb{N}$, all $\mathsf{pp}_{\mathit{\Gamma}} \leftarrow_r \mathsf{ParGen}_{\mathit{\Gamma}}(1^\lambda)$, for all messages $m \in \mathcal{M}$, for all $(C, O) \leftarrow_r \mathsf{Commit}_{\mathit{\Gamma}}(\mathsf{pp}_{\mathit{\Gamma}}, m)$, it holds that $\mathsf{Open}_{\mathit{\Gamma}}(\mathsf{pp}_{\mathit{\Gamma}}, C, O, m) = 1$.*

Definition 4 (Binding). *A non-interactive commitment scheme is binding, if for all PPT adversaries $\mathcal{A}$ there exists a negligible function ν such that:*

$$\Pr[\mathbf{Exp}^{\mathsf{Binding}}_{\mathcal{A},\mathit{\Gamma}}(\lambda) = 1] \leq \nu(\lambda),$$

where the corresponding experiment is depicted in Figure 1b.

Definition 5 (Hiding). *A non-interactive commitment scheme $\mathit{\Gamma}$ is hiding, if for any PPT adversary $\mathcal{A}$, there exists a negligible functions ν such that:*

$$\left|\Pr[\mathbf{Exp}^{\mathsf{Hiding}}_{\mathcal{A},\mathit{\Gamma}}(\lambda) = 1] - {}^1\!/_2\right| \leq \nu(\lambda),$$

where the corresponding experiment is depicted in Figure 1b.

2.3 Non-Interactive Proof Systems

Let L be an NP-language with associated witness relation R, i.e., such that $L = \{x \mid \exists w : R(x, w) = 1\}$. A non-interactive proof system allows to prove membership of some statement x in the language L. More formally, such a system is defined as follows.

Definition 6 (Non-Interactive Proof System). *A non-interactive proof system $\mathsf{\Pi}$ for language L consists of three algorithms $\{\mathsf{PG}_{\mathsf{\Pi}}, \mathsf{Prf}_{\mathsf{\Pi}}, \mathsf{Vfy}_{\mathsf{\Pi}}\}$, such that:*

$\mathsf{PG}_{\mathsf{\Pi}}$. *The algorithm $\mathsf{PG}_{\mathsf{\Pi}}$ outputs public parameters of the scheme, where λ is the security parameter:*

$$\mathsf{crs}_{\mathsf{\Pi}} \leftarrow_r \mathsf{PG}_{\mathsf{\Pi}}(1^\lambda)$$

```
$\mathbf{Exp}^{\mathsf{Binding}}_{\mathcal{A},\Gamma}(\lambda)$
  $\mathsf{pp}_\Gamma \leftarrow_r \mathsf{ParGen}_\Gamma(1^\lambda)$
  $(C^*, O^*, O'^*, m^*, m'^*) \leftarrow_r \mathcal{A}(\mathsf{pp}_\Gamma)$
  return 1, if $\mathsf{Open}_\Gamma(\mathsf{pp}_\Gamma, C^*, O^*, m^*) = 1 \wedge$
               $\mathsf{Open}_\Gamma(\mathsf{pp}_\Gamma, C^*, O'^*, m'^*) = 1 \wedge$
               $m^* \neq m'^*$
  return 0
```

(a) Binding

```
$\mathbf{Exp}^{\mathsf{Hiding}}_{\mathcal{A},\Gamma}(\lambda)$
  $\mathsf{pp}_\Gamma \leftarrow_r \mathsf{ParGen}_\Gamma(1^\lambda)$
  $b \leftarrow_r \{0,1\}$
  $b^* \leftarrow_r \mathcal{A}^{\mathsf{Commit}'_\Gamma(\mathsf{pp}_\Gamma,\cdot,\cdot,b)}(\mathsf{pp}_\Gamma)$
    where $\mathsf{Commit}'_\Gamma$ on input $\mathsf{pp}_\Gamma$, $m_0$, $m_1$, and $b$:
      return $\perp$, if $m_0 \notin \mathcal{M} \vee m_1 \notin \mathcal{M}$
      $(C, O) \leftarrow_r \mathsf{Commit}_\Gamma(\mathsf{pp}_\Gamma, m_b)$
      return $C$
  return 1, if $b^* = b$
  return 0
```

(b) Hiding

Fig. 1: Security Games for Non-Interactive Commitments

Prf_Π. *The algorithm* Prf_Π *outputs the proof* π*, on input of the CRS* crs_Π*, statement* x *to be proven, and the corresponding witness* w*:*

$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, x, w)$$

Vfy_Π. *The deterministic algorithm* Vfy_Π *verifies the proof* π *by outputting a bit* $d \in \{0,1\}$*, w.r.t. to some CRS* crs_Π *and some statement statement* x*:*

$$d \leftarrow \mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x, \pi)$$

Definition 7 (Correctness). *A non-interactive proof system is called correct, if for all* $\lambda \in \mathbb{N}$*, for all* $\mathsf{crs}_\Pi \leftarrow_r \mathsf{PG}_\Pi(1^\lambda)$*, for all* $x \in L$*, for all* w *such that* $R(x, w) = 1$*, for all* $\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, x, w)$*, it holds that* $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x, \pi) = 1$*.*

In the context of (zero-knowledge) proof-systems, correctness is sometimes also referred to as completeness. In addition, we require two standard security notions for zero-knowledge proofs of knowledge: zero-knowledge and simulation-sound extractability (also known as simulation-extractability). We define them analogously to the definitions given in [DS19].

Informally speaking, zero-knowledge says that the receiver of the proof π does not learn anything except the validity of the statement. It is required that the distribution of crs_Π output by SIM_1 is distributed identically to PG_Π.

Definition 8 (Zero-Knowledge). *A non-interactive proof system* Π *for language* L *is zero-knowledge, if for any PPT adversary* $\mathcal{A}$*, there exists an PPT simulator* $\mathsf{SIM} = (\mathsf{SIM}_1, \mathsf{SIM}_2)$ *such that there exist negligible functions* ν_1 *and* ν_2 *such that*

$$\Big| \Pr\left[\mathsf{crs}_\Pi \leftarrow_r \mathsf{PG}_\Pi(1^\lambda) : \mathcal{A}(\mathsf{crs}_\Pi) = 1\right] - \Pr\left[(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda) : \mathcal{A}(\mathsf{crs}_\Pi) = 1\right] \Big| \leq \nu_1(\lambda),$$

and that

$$\left| \Pr\left[\mathbf{Exp}^{\mathsf{Zero\text{-}Knowledge}}_{\mathcal{A},\Pi,\mathsf{SIM}}(\lambda) = 1\right] - {}^1\!/_2 \right| \leq \nu_2(\lambda),$$

where the corresponding experiment is depicted in Figure 2a.

```
$\mathbf{Exp}_{\mathcal{A},\Pi,\mathsf{SIM}}^{\mathsf{Zero\text{-}Knowledge}}(\lambda)$
  $(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda)$
  $b \leftarrow_r \{0,1\}$
  $b^* \leftarrow_r \mathcal{A}^{P_b(\cdot,\cdot)}(\mathsf{crs}_\Pi)$
    where $P_0$ on input $x$, $w$:
      return $\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, x, w)$, if $R(x, w) = 1$
      return $\perp$
    and $P_1$ on input $x$, $w$:
      return $\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, x)$, if $R(x, w) = 1$
      return $\perp$
  return 1, if $b^* = b$
  return 0
```

(a) Zero-Knowledge

```
$\mathbf{Exp}_{\mathcal{A},\Pi,\mathcal{E}}^{\mathsf{SimSoundExt}}(\lambda)$
  $(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_r \mathcal{E}_1(1^\lambda)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(x^*, \pi^*) \leftarrow_r \mathcal{A}^{\mathsf{SIM}(\cdot)}(\mathsf{crs}_\Pi)$
    where $\mathsf{SIM}$ on input $x$:
      obtain $\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, x)$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(x, \pi)\}$
      return $\pi$
  $w^* \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, x^*, \pi^*)$
  return 1, if $\mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, x^*, \pi^*) = 1 \wedge$
    $R(x^*, w^*) = 0 \ \wedge \ (x^*, \pi^*) \notin \mathcal{Q}$
  return 0
```

(b) Simulation-Sound Extractability

Fig. 2: Security Games for Non-Interactive Proof Systems

Simulation-sound extractability says that every adversary which is able to come up with a proof π^* for a statement must know the witness, even when seeing proofs for statements potentially not in L [Sah99]. Clearly, this implies that the proofs output by a simulation-sound extractable proof-systems are non-malleable. Note that the definition of simulation-sound extractability of [Gro06] is stronger than ours in the sense that the adversary also gets the trapdoor ζ as input. However, in our context this weaker notion (previously also used e.g. in [ADK+13, DHLW10]) suffices.

Definition 9 (Simulation-Sound Extractability). *A zero-knowledge non-interactive proof system Π for language L is said to be simulation-sound extractable, if for any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$, such that*

$$\Big|\Pr\left[(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda) \ : \ \mathcal{A}(\mathsf{crs}_\Pi, \tau) = 1\right] - \\ \Pr\left[(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_r \mathcal{E}_1(1^\lambda) \ : \ \mathcal{A}(\mathsf{crs}_\Pi, \tau) = 1\right]\Big| = 0,$$

and that there exist a negligible function ν so that

$$\Pr\left[\mathbf{Exp}_{\mathcal{A},\Pi,\mathcal{E}}^{\mathsf{SimSoundExt}}(\lambda) = 1\right] \leq \nu(\lambda),$$

where the corresponding experiment is depicted in Figure 2b.

3 Syntax and Security of Chameleon-Hashes

We next present the formal framework for CHs used by Derler et al. [DSS20], which itself is based on prior work [AMVA17, BFF+09, CDK+17].

Definition 10. *A chameleon-hash* CH *is a tuple of five PPT algorithms* $(\mathsf{CHPG}, \mathsf{CHKG}, \mathsf{CHash}, \mathsf{CHCheck}, \mathsf{CHAdapt})$*, such that:*

CHPG. *The algorithm* CHPG*, on input a security parameter* λ *outputs public parameters of the scheme:*

$$\mathsf{pp_{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$$

We assume that $\mathsf{pp_{ch}}$ *contains* 1^λ *and is implicit input to all other algorithms.*

CHKG. *The algorithm* CHKG*, on input the public parameters* $\mathsf{pp_{ch}}$ *outputs the private and public keys of the scheme:*

$$(\mathsf{sk_{ch}}, \mathsf{pk_{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp_{ch}})$$

CHash. *The algorithm* CHash *gets as input the public key* $\mathsf{pk_{ch}}$*, and a message* m *to hash. It outputs a hash* h*, and some randomness* r*:*[4]

$$(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, m)$$

$\mathsf{CHCheck}$. *The deterministic algorithm* $\mathsf{CHCheck}$ *gets as input the public key* $\mathsf{pk_{ch}}$*, a message* m*, randomness* r*, and a hash* h*. It outputs a bit* $d \in \{0, 1\}$*, indicating whether the hash* h *is valid:*

$$d \leftarrow \mathsf{CHCheck}(\mathsf{pk_{ch}}, m, r, h)$$

$\mathsf{CHAdapt}$. *The algorithm* $\mathsf{CHAdapt}$ *on input of a secret key* $\mathsf{sk_{ch}}$*, the message* m*, the randomness* r*, hash* h*, and a new message* m' *outputs new randomness* r'*:*

$$r' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r, h)$$

Definition 11 (Correctness). *A chameleon-hash is called correct, if for all security parameters* $\lambda \in \mathbb{N}$*, for all* $\mathsf{pp_{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$*, for all* $(\mathsf{sk_{ch}}, \mathsf{pk_{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp_{ch}})$*, for all* $m \in \mathcal{M}$*, for all* $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, m)$*, for all* $m' \in \mathcal{M}$*, we have for all* $r' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r, h)$*, that* $1 = \mathsf{CHCheck}(\mathsf{pk_{ch}}, m, r, h) = \mathsf{CHCheck}(\mathsf{pk_{ch}}, m', r', h)$*.*

Full Collision-Resistance. Derler et al. [DSS20] recently defined the notion of full collision-resistance. Here, the adversary gets access to a collision-finding oracle $\mathsf{CHAdapt}'$, which outputs a collision for the adversarially chosen hash, but also keeps track of each of the queried and returned hash/message *pairs* (h, m) and (h, m'), using the list $\mathcal{Q}$. The adversary wins, if it comes up with a hash/message pair (h^*, m^*) colliding with (m'^*, r'^*), for the given public key, where (m'^*, r'^*) was never queried to or output from the collision-finding oracle.

Definition 12 (Full Collision-Resistance). *A chameleon-hash* CH *provides full collision-resistance, if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that*

$$\Pr[\mathbf{Exp}^{\mathsf{F\text{-}CollRes}}_{\mathcal{A},\mathsf{CH}}(\lambda) = 1] \leq \nu(\lambda)$$

The corresponding experiment is depicted in Figure 3a.

[4] We note that the randomness r is also sometimes called "check value" [AMVA17].

```
$\mathbf{Exp}^{\mathsf{F\text{-}CollRes}}_{\mathcal{A},\mathsf{CH}}(\lambda)$
  $\mathsf{pp_{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$
  $(\mathsf{sk_{ch}}, \mathsf{pk_{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp_{ch}})$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, r^*, m'^*, r'^*, h^*) \leftarrow_r \mathcal{A}^{\mathsf{CHAdapt}'(\mathsf{sk_{ch}},\cdot,\cdot,\cdot,\cdot)}(\mathsf{pk_{ch}})$
    oracle $\mathsf{CHAdapt}'$ on input $\mathsf{sk_{ch}}, m, m', r, h$:
      return $\perp$, if $\mathsf{CHCheck}(\mathsf{pk_{ch}}, m, r, h) \neq 1$
      $r' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r, h)$
      Return $\perp$, if $r' = \perp$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(h, m), (h, m')\}$
      return $r'$
  return 1, if $\mathsf{CHCheck}(\mathsf{pk_{ch}}, m^*, r^*, h^*) = 1 \wedge$
    $\mathsf{CHCheck}(\mathsf{pk_{ch}}, m'^*, r'^*, h^*) = 1 \wedge$
    $m^* \neq m'^* \ \wedge \ (h^*, m^*) \notin \mathcal{Q}$
  return 0
```

(a) Full Collision-Resistance

```
$\mathbf{Exp}^{\mathsf{S\text{-}Ind}}_{\mathcal{A},\mathsf{CH}}(\lambda)$
  $\mathsf{pp_{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$
  $(\mathsf{sk_{ch}}, \mathsf{pk_{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp_{ch}})$
  $b \leftarrow_r \{0, 1\}$
  $b^* \leftarrow_r \mathcal{A}^{\mathsf{HashOrAdapt}(\mathsf{sk_{ch}},\mathsf{pk_{ch}},\cdot,\cdot,b)}(\mathsf{sk_{ch}}, \mathsf{pk_{ch}})$
    where $\mathsf{HashOrAdapt}$ on input $\mathsf{sk_{ch}}, \mathsf{pk_{ch}}, m, m', b$:
      $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, m')$
      $(h', r') \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, m)$
      $r'' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r', h')$
      return $\perp$, if $r'' = \perp \ \vee \ r' = \perp \ \vee r = \perp$
      if $b = 0$, return $(h, r)$
      if $b = 1$, return $(h', r'')$
  return 1, if $b^* = b$
  return 0
```

(b) Strong Indistinguishability

Fig. 3: Security Games for Chameleon-Hashes

Strong Indistinguishability. Strong indistinguishability is a strong privacy notion [DSSS19]. It requires that a randomness r does not reveal whether it was generated using CHash or $\mathsf{CHAdapt}$, even if the adversary $\mathcal{A}$ knows all secret keys.

Definition 13 (Strong Indistinguishability). *A chameleon-hash* CH *provides strong indistinguishability, if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that*

$$\left|\Pr[\mathbf{Exp}^{\mathsf{S\text{-}Ind}}_{\mathcal{A},\mathsf{CH}}(\lambda) = 1] - {}^1\!/_2\right| \leq \nu(\lambda)$$

The corresponding experiment is depicted in Figure 3b.

Randomness Unforgeability. Uniqueness, introduced by Camenisch et al. at PKC 20 [CDK+17], requires that an adversary controlling *all* values (but the public parameters) cannot find two distinct randomness values $r^* \neq r'^*$ for the same hash/message pair (h, m).[5]Uniqueness is a very strong notion that is hard to achieve, and, in this strong form, only seems to be required in one particular use case [BCD+17, CDK+17, HZM+20, SS20]. To this end, we introduce a slightly weaker variant that is easier to achieve while still being useful in other applications. It requires that an adversary cannot find new randomness for hashes it did not create by itself. We call this notion randomness unforgeability.

In our formalization, the challenger generates the key pair and parameters honestly, and uses $\mathsf{pk_{ch}}$ to initialize the adversary. The adversary gains access to two oracles. The oracle CHash' allows the adversary to adaptively receive hashes on messages of its choice. The generated hash/randomness pairs (h, r) are stored in a set $\mathcal{Q}$. The oracle $\mathsf{CHAdapt}'$ allows the adversary to adaptively find collisions for hashes. If the adversary queries a hash/randomness pair which is an element of $\mathcal{Q}$, the resulting (h, r') is also added to $\mathcal{Q}$. The adversary wins, if it can come up with a new randomness r^* (i.e., not stored in $\mathcal{Q}$) for whatever message m^*, verifying for a hash h^* which was output by CHash'.

[5] The definition of uniqueness is restated in Appendix A.

```
$\mathbf{Exp}^{\mathsf{Rand\text{-}Uf}}_{\mathcal{A},\mathsf{CH}}(\lambda)$
  $\mathsf{pp}_{\mathsf{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$
  $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, h^*, r^*) \leftarrow_r \mathcal{A}^{\mathsf{CHash}'(\mathsf{pk}_{\mathsf{ch}},\cdot),\mathsf{CHAdapt}'(\mathsf{sk}_{\mathsf{ch}},\cdot,\cdot,\cdot,\cdot)}(\mathsf{pk}_{\mathsf{ch}})$
    oracle $\mathsf{CHash}'$ on input $\mathsf{pk}_{\mathsf{ch}}, m$:
      $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(h, r)\}$
      return $(h, r)$
    oracle $\mathsf{CHAdapt}'$ on input $\mathsf{sk}_{\mathsf{ch}}, m, m', r, h$:
      $r' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)$
      return $\perp$, if $\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m', r', h) \neq 1$
      If $\exists(h, \cdot) \in \mathcal{Q}$:
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(h, r')\}$
      return $r'$
  return 1, if $\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m^*, r^*, h^*) = 1 \wedge$
    $(h^*, \cdot) \in \mathcal{Q} \ \wedge \ (h^*, r^*) \notin \mathcal{Q}$
  return 0
```

Fig. 4: Randomness Unforgeability

Definition 14 (Randomness Unforgeability). *A chameleon-hash* CH *offers randomness unforgeability, if for any PPT adversary $\mathcal{A}$ there exists a negligible function ν such that*

$$\Pr[\mathbf{Exp}^{\mathsf{Rand\text{-}Uf}}_{\mathcal{A},\mathsf{CH}}(\lambda) = 1] \leq \nu(\lambda)$$

The corresponding experiment is depicted in Figure 4.

4 Generic Construction

The main idea of our generic construction follows the original idea by Derler et al. [DSS20], but slightly altered to meet our requirements. Namely, hashing a message m means committing to it. The randomness r is a SSE NIZK proving membership of a tuple containing the opening O for the commitment, and the pre-image x of a one-way function f, fulfilling the following NP-relation:

$$L := \{(\mathsf{pp}_\Gamma, h, m, y) \mid \exists\ (O, x)\ :\ \mathsf{Open}_\Gamma(\mathsf{pp}_\Gamma, h, O, m) = 1\ \vee\ y = f(x)\} \quad (1)$$

Informally, this language requires the prover to demonstrate that it either knows an opening O such that h is a well-formed commitment of m under pp_Γ, *or* the pre-image x corresponding to $f(x)$ of a one-way function f is known. Our construction of a fully collision-resistant, strongly indistinguishable, and randomness unforgeable, CH is presented as Construction 1.

4.1 Security

Subsequently, we prove the security of our CH in Construction 1.

Theorem 1. *If Γ is correct, and Π is complete, then* CH *in Construction 1 is correct.*

$\underline{\mathsf{CHPG}(1^\lambda)}$: Fix a commitment scheme Γ, a one-way function f, and a compatible NIZK proof system for language L in (1). Return $\mathsf{pp_{ch}} = (f, \mathsf{pp}_\Gamma, \mathsf{crs}_\Pi)$, where

$$\mathsf{pp}_\Gamma \leftarrow_r \mathsf{ParGen}_\Gamma(1^\lambda), \text{ and } \mathsf{crs}_\Pi \leftarrow_r \mathsf{PG}_\Pi(1^\lambda).$$

$\underline{\mathsf{CHKG}(\mathsf{pp_{ch}})}$: Return $(\mathsf{sk_{ch}}, \mathsf{pk_{ch}}) = (x, y)$, where

$$x \leftarrow_r \{0,1\}^\lambda, y \leftarrow f(x).$$

$\underline{\mathsf{CHash}(\mathsf{pk_{ch}}, m)}$: Parse $\mathsf{pk_{ch}}$ as $((f, \mathsf{pp}_\Gamma, \mathsf{crs}_\Pi), y)$ and return $(h, r) = (C, \pi)$, where

$$(C, O) \leftarrow_r \mathsf{Commit}_\Gamma(\mathsf{pp}_\Gamma, m), \text{ and } \pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pp}_\Gamma, h, m, y), (O, \bot)).$$

$\underline{\mathsf{CHCheck}(\mathsf{pk_{ch}}, m, r, h)}$: Parse $\mathsf{pk_{ch}}$ as $((f, \mathsf{pp}_\Gamma, \mathsf{crs}_\Pi), y)$ and r as π, and return 1, if the following holds, and 0 otherwise:

$$m \in \mathcal{M} \ \wedge \ \mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pp}_\Gamma, h, m, y), \pi) = 1.$$

$\underline{\mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r, h)}$: Parse $\mathsf{sk_{ch}}$ as x, and set $y \leftarrow f(x)$. Check that $m' \in \mathcal{M}$ and $\mathsf{CHCheck}(y, m, r, h) = 1$. Return $\bot$, if not. Otherwise, return $r' = \pi'$, where

$$\pi' \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pp}_\Gamma, h, m', y), (\bot, x)).$$

Construction 1: Our Construction of a Fully Collision-Resistant CH

Correctness follows from inspection and the (perfect) correctness of the used primitives.

Theorem 2. *If Γ is binding, f is a one-way function, and Π is simulation-sound extractable, then* CH *in Construction 1 is fully collision-resistant.*

The proof of this theorem is along the same lines as that in Derler et al. [DSS20].

Proof. We prove full collision-resistance using a sequence of games.

Game 0: The original full collision-resistance game.

Game 1: As Game 0, but we replace the CHPG algorithm with an algorithm CHPG′ and modify the CHAdapt′ oracle as follows:

$\underline{\mathsf{CHPG}'(1^\lambda)}$:

$$\mathsf{crs}_\Pi \leftarrow_r \mathsf{PG}_\Pi(1^\lambda) \rightsquigarrow \boxed{(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda)}.$$

$\underline{\mathsf{CHAdapt}'(\mathsf{sk_{ch}}, m, m', r, h)}$: In CHAdapt:

$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pp}_\Gamma, h, m, y), (x, \bot)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m, y)).}$$

Transition - Game 0 → Game 1: We bound the probability for an adversary to detect this game change by presenting a hybrid game, which, depending on a zero-knowledge challenger $\mathcal{C}^{\mathsf{zk}}$, either produces the distribution in Game 0 or Game 1, respectively. In particular, assume that we use the following algorithm CHPG′′ instead of CHPG and CHPG′:

$\underline{\mathsf{CHPG}''(1^\lambda)}$:

$$(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda) \rightsquigarrow \boxed{\mathsf{crs}_\Pi \leftarrow_r \mathcal{C}^{\mathsf{zk}}}.$$

$\underline{\mathsf{CHAdapt}'(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$: In $\mathsf{CHAdapt}$:

$$\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m', y)) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pp}_\Gamma, h, m', y), (\bot, x))}.$$

Clearly, if the challenger's internal bit is 0 we simulate the distribution in Game 0, whereas we simulate the distribution in Game 1 otherwise. We have that $|\Pr[S_0] - \Pr[S_1]| \leq \nu_{\mathsf{zk}}(\lambda)$.

Game 2: As Game 1, but we replace the CHPG' algorithm with an algorithm CHPG''' which works as follows:

$\underline{\mathsf{CHPG}'''(1^\lambda)}$:

$$(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda) \rightsquigarrow \boxed{(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_r \mathcal{E}_1(1^\lambda)}.$$

Transition - Game 1 → Game 2: Under simulation-sound extractability, Game 1 and Game 2 are indistinguishable. That is, $|\Pr[S_1] - \Pr[S_2]| = 0$.

Game 3: As Game 2, but we keep a list $\mathcal{Q}$ of all tuples (h, r, m) previously submitted to the collision-finding oracle which are accepted by the $\mathsf{CHCheck}$ algorithm, where h was never submitted to the collision-finding oracle before.

Transition - Game 2 → Game 3: This change is purely conceptual, i.e., it does not change the view of the adversary. $|\Pr[S_2] - \Pr[S_3]| = 0$ follows.

Game 4: As Game 3, but for every valid collision $(m^*, r^*, m'^*, r'^*, h^*)$ output by the adversary we observe that either (m^*, r^*) or (m'^*, r'^*) must be a "fresh" collision, i.e., one that was never output by the collision-finding oracle. We assume, without loss of generality, that (m'^*, r'^*) is the "fresh" collision. We run $(x', O') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pp}_\Gamma, h^*, m'^*, y), r'^*)$ and abort if the extraction fails. We call this event E_1.

Transition - Game 3 → Game 4: Game 3 and Game 4 proceed identically, unless E_1 occurs. Assume, towards contradiction, that event E_1 occurs with non-negligible probability. We now construct an adversary $\mathcal{B}$ which breaks the simulation-sound extractability property of the NIZK proof-system with non-negligible probability. We engage with a simulation-sound extractability challenger $\mathcal{C}^{\mathsf{sse}}$ and modify the algorithms as follows:

$\underline{\mathsf{CHPG}''''(1^\lambda)}$:

$$(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_r \mathcal{E}_1(1^\lambda) \rightsquigarrow \boxed{(\mathsf{crs}_\Pi, \bot, \bot) \leftarrow_r \mathcal{C}^{\mathsf{sse}}}.$$

$\underline{\mathsf{CHAdapt}''(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$:

$$\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m', y)) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{sse}}.\mathsf{SIM}(\mathsf{pp}_\Gamma, h, m', y)}.$$

In the end we output $((\mathsf{pp}_\Gamma, h^*, m'^*, y), r'^*)$ to the challenger. This shows that we have $|\Pr[S_3] - \Pr[S_4]| \leq \nu_{\mathsf{sse}}(\lambda)$.

Game 5: As Game 4, but we observe that if (m^*, r^*) does not correspond to a fresh collision for h^* in the above sense, then we will have an entry

$(h^*, r, m) \in \mathcal{Q}$ where (m, r) is a "fresh" collision, i.e., one computed by the adversary. We run the extractor for the fresh collision, i.e., either obtain $(x'', O'') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pp}_\Gamma, h^*, m^*, y), r^*)$ or $(x'', O'') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pp}_\Gamma, h^*, m, y), r)$, respectively. In case the extraction fails, we abort. We call the abort event E_2.

Transition - Game 4 → Game 5: Analogously to the transition between Game 3 and Game 4, we argue that Game 4 and Game 5 proceed identically unless E_2 occurs which is why we do not restate the reduction to simulation-sound extractability here. We have that $|\Pr[S_4] - \Pr[S_5]| \leq \nu_{\mathsf{sse}}(\lambda)$.

Reduction to Binding and One-Wayness: We are now ready to construct an adversary $\mathcal{B}$ which breaks either the binding property of the used one-way function or the binding property of the underlying Γ. Our adversary $\mathcal{B}$ proceeds as follows. It receives pp_Γ from its binding challenger, as well as, f and y from a one-way challenger. It embeds them straightforwardly as $\mathsf{pp}_{\mathsf{ch}}$ and $\mathsf{pk}_{\mathsf{ch}}$ to initialize $\mathcal{A}$. Now we know that we have extracted two witnesses (x, O) as well as (x'', O'') where one attests membership of $(\mathsf{pk}_\Omega, h^*, m'^*, y)$ in L and one attests membership of $(\mathsf{pk}_\Omega, h^*, m'', y)$ for some $m'' \neq m'^*$ in L. In either case, $\mathcal{B}$ can check whether $f(x) = y$ or $f(x'') = y$ holds. In this case, it can return x, or x'' resp., to its one-way challenger. In all other cases, O and O'' open the commitment h^* to different messages. Thus, $\mathcal{B}$ can directly return (h^*, O, O'', m^*, m'^*) as its own forgery. A union bound gives us $\Pr[S_5] \leq \nu_{\mathsf{owf}}(\lambda) + \nu_{\mathsf{binding}}(\lambda)$. This concludes the proof. □

Remark 1. Note that, like Derler et al. [DSS20], we conduct a full collision-resistance proof that only requires extracting twice. While the formal notion of simulation-sound extractability would allow us to simply extract in every oracle query, and, thus, obtain a more general result, this is to ensure that one can plug in proof systems that rely on a rewinding extractor without putting a restriction on the allowed adversarial queries. We note, however, that this way of proving the theorem implies some limitations and if one can not afford these limitations one would need to prove it via extracting in every oracle query, thus excluding some of the proof systems we can plausibly plug in when only extracting twice. The limitations are as follows: Observe that the extractor is formally only guaranteed to work as long as either the proof we want to extract from, or the corresponding statement does not correspond to an output of a query to the simulator. For the proof above to go through, this means that the concrete proof system plugged into our generic construction needs to have the property that for any given valid proof for some statement the probability that the proof output by an honest run of the simulator for the same statement will only collide with this proof with negligible probability. This is a pretty common property for proof systems, and all proof systems we can think of provide the required guarantees (e.g., Groth-Sahai proofs [GS08], or Fiat-Shamir transformed Σ protocols).

Theorem 3. *If Γ is hiding, and Π is zero-knowledge, then CH in Construction 1 is strongly indistinguishable.*

In the proof, we use $\boxed{\text{frameboxes}}$ and $\rightsquigarrow$ to highlight the changes we make in the algorithms throughout a sequence of games (and we only show the changes).

Proof. To prove strong indistinguishability, we use a sequence of games:

Game 0: The original strong indistinguishability game.

Game 1: As Game 0, but we modify the algorithms CHPG and the $\mathsf{HashOrAdapt}$ oracle as follows:

$\underline{\mathsf{CHPG}'(1^\lambda)}$:

$$\mathsf{crs}_\Pi \leftarrow_r \mathsf{PG}_\Pi(1^\lambda) \rightsquigarrow \boxed{(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda)}.$$

$\underline{\mathsf{HashOrAdapt}'(\mathsf{pk_{ch}}, \mathsf{sk_{ch}}, \mathsf{m}, \mathsf{m}', \mathsf{b})}$: In CHash:

$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\ldots) \rightsquigarrow \boxed{\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m, \mathsf{pk_{ch}}))}$$

and $\mathsf{CHAdapt}$:

$$\pi' \leftarrow_r \mathsf{Prf}_\Pi(\ldots) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, h, m', f(\mathsf{sk_{ch}}))).}$$

Transition - Game 0 → Game 1: We bound the probability for an adversary to detect this game change by presenting a hybrid game, which, depending on a zero-knowledge challenger $\mathcal{C}^{\mathsf{zk}}$, either produces the distribution in Game 0 or Game 1, respectively. In particular, assume that we use the following changes:

$\underline{\mathsf{CHPG}''(1^\lambda)}$:

$$(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda) \rightsquigarrow \boxed{\mathsf{crs}_\Pi \leftarrow_r \mathcal{C}^{\mathsf{zk}}}.$$

$\underline{\mathsf{HashOrAdapt}''(\mathsf{pk_{ch}}, \mathsf{sk_{ch}}, \mathsf{m}, \mathsf{m}', \mathsf{b})}$: In CHash:

$$\pi \leftarrow_r \mathsf{SIM}_2(\ldots) \rightsquigarrow \boxed{\pi \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pp}_\Gamma, h, m, \mathsf{pk_{ch}}), (O, \bot))}.$$

and $\mathsf{CHAdapt}$:

$$\pi' \leftarrow_r \mathsf{SIM}_2(\ldots) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pp}_\Gamma, h, m', f(\mathsf{sk_{ch}})), (\bot, x))}.$$

Clearly, if the challenger's internal bit is 0 we simulate the distribution in Game 0, whereas we simulate the distribution in Game 1 otherwise. We have that $|\Pr[S_0] - \Pr[S_1]| \leq \nu_{\mathsf{zk}}(\lambda)$.

Game 2: As Game 1, but we modify the $\mathsf{HashOrAdapt}$ oracle as follows:

$\underline{\mathsf{HashOrAdapt}'''(\cdot, \cdot, \cdot, \mathsf{b})}$: In CHash

$$(C, O) \leftarrow_r \mathsf{Commit}_\Gamma(\mathsf{pp}_\Gamma, m) \rightsquigarrow \boxed{(C, O) \leftarrow_r \mathsf{Commit}_\Gamma(\mathsf{pp}_\Gamma, 0)}.$$

Transition - Game 1 → Game 2: We bound the probability for an adversary to distinguish between two consecutive games by introducing a hybrid game which uses a hiding challenger to interpolate between two consecutive games.

$\underline{\mathsf{CHPG}'''(\lambda)}$:

$$\mathsf{pp}_{\Gamma} \leftarrow_r \mathsf{ParGen}_{\Gamma}(1^{\lambda}) \rightsquigarrow \boxed{\mathsf{pp}_{\Gamma} \leftarrow_r \mathcal{C}^{\mathsf{hiding}}}.$$

$\underline{\mathsf{HashOrAdapt}''''(\mathsf{pk}_{\mathsf{ch}}, \mathsf{sk}_{\mathsf{ch}}, \mathsf{m}, \mathsf{m}', \mathsf{b})}$: In CHash:

$$(C, O) \leftarrow_r \mathsf{Commit}_{\Gamma}(\mathsf{pp}_{\Gamma}, 0) \rightsquigarrow \boxed{(C, \perp) \leftarrow_r \mathcal{C}^{\mathsf{hiding}}.\mathsf{Commit}'_{\Gamma}(m, 0)}.$$

Now, depending on the challenger's bit, we either simulate Game 1 or Game 2. Thus we have that $|\Pr[S_1] - \Pr[S_2]| \leq \nu_{\mathsf{hiding}}(\lambda)$.

Now, the strong indistinguishability game is independent of the bit b, proving strong indistinguishability. □

Theorem 4. *If Γ is binding and hiding, f is a one-way function, Π is simulation-sound extractable, and* CH *fully collision-resistant, then* CH *in Construction 1 is randomness unforgeable.*

The proof of this theorem is presented in Appendix B.

5 Concrete Instantiations

5.1 Concrete Instantiation from Pre-Quantum Primitives

Our pre-quantum instantiation follows our generic compiler. As instantiation for Γ we use Pedersen commitments [Ped91] in discrete-logarithm (DL) hard groups. For f we use is the exponentiation in the aforementioned group, which is a one-way function under the DL assumption. For the non-interactive proof system, we use Fiat-Shamir (FS) transformed Σ-protocols for DLOG relations in the random-oracle model [FS86] and additionally apply the compiler by Faust et al. [FKMV12] to make it simulation-sound extractable. This compiler requires additionally including the statement x upon hashing in the challenge computation. In addition the Σ-protocol needs to provide a property called quasi-unique responses for this compiler to apply, which is straightforward for our statements. See, e.g., [DS18], for a detailed discussion of this transformation. Although when using FS we have to rely on a rewinding extractor, this choice is suitable as in our security proofs we only need to extract a bounded number of times (i.e., twice).

We provide this concrete instantiation as Construction 2, where we let $(\mathbb{G}, g_1, q) \leftarrow_r \mathsf{GGen}(1^{\lambda})$ be an instance generator which returns a prime-order, and multiplicatively written, group $\mathbb{G}$, where the DL problem is hard, along with two generators g_1, g_2 as the Pedersen parameters (we compute $g_2 = H'(g_1)$ where H' is a random oracle to avoid a trusted setup). Note that an SSE NIZK for the required L in (2) is obtained using an *or* composition of a proof of a discrete logarithm [CDS94] of Fiat-Shamir transformed Σ-protocols.

$$L := \{(y, h, m) \mid \exists\ (x, \xi)\ :\ h = (g_1^m g_2^{\xi})\ \vee\ y = g_1^x\}. \tag{2}$$

$\underline{\mathsf{CHPG}(1^\lambda)}$: Outputs the public parameters $(\mathbb{G}, g_1, g_2, q, H)$, where $\mathsf{pp_{ch}} = (\mathbb{G}, g_1, q) \leftarrow_r \mathsf{GGen}(1^\lambda)$, $g_2 \leftarrow H'(g_1)$, and a hash-functions $H : \{0,1\}^* \to \mathbb{Z}_q$ and $H' : \{0,1\}^* \to \mathbb{G}$ (which we assume to behave like a random oracle and to be implicitly available to all algorithms below).

$\underline{\mathsf{CHKG}(\mathsf{pp_{ch}})}$: Return $(\mathsf{sk_{ch}}, \mathsf{pk_{ch}}) = (x, y)$, where $x \leftarrow_r \mathbb{Z}_q$ and $y \leftarrow g_1^x$.

$\underline{\mathsf{CHash}(\mathsf{pk_{ch}}, m)}$: Parse $\mathsf{pk_{ch}}$ as y and $m \in \mathbb{Z}_q$, choose $(\xi, k_{1,1}, k_{1,2}, k_2, e_2, s_2) \leftarrow_r \mathbb{Z}_q^6$, set $u_1 \leftarrow g_1^{k_{1,1}} g_2^{k_{1,2}}$, $u_2 \leftarrow g_1^{s_2} \cdot y^{-e_2}$, $e \leftarrow H((y, h, m), (u_1, u_2))$ and $e_1 \leftarrow e - e_2 \bmod q$. Then compute $s_{1,1} \leftarrow k_{1,1} + e_1 m \bmod q$, $s_{1,2} = k_{1,2} + e_1 \xi$ and finally, return $(h, r) = (O, \pi)$, where

$$O \leftarrow g_1^m g_2^\xi \text{ , and } \pi \leftarrow (e_1, e_2, s_{1,1}, s_{1,2}, s_2).$$

$\underline{\mathsf{CHCheck}(\mathsf{pk_{ch}}, m, r, h)}$: Parse $\mathsf{pk_{ch}}$ as y and r as $(e_1, e_2, s_{1,1}, s_{1,2}, s_2)$, and h as O. Return 1 if the following holds, and 0 otherwise:

$$m \in \mathbb{Z}_q \ \wedge \ e_1 + e_2 = H((y, h, m), (g_1^{s_{1,1}} g_2^{s_{1,2}} \cdot O^{-e_1}, g^{s_2} \cdot y^{-e_2})).$$

$\underline{\mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r, h)}$: Parse $\mathsf{sk_{ch}}$ as x, and h as O. Set $y \leftarrow g_1^x$. Verify whether $m' \in \mathbb{Z}_q$, and $\mathsf{CHCheck}(y, m, r, h) = 1$. Return $\perp$ if not. Otherwise, choose $(k_{1,1}, k_{1,2}, e_1, s_{1,1}, s_{1,2}) \leftarrow_r \mathbb{Z}_q^5$, set $u_1 \leftarrow g_1^{s_{1,1}} \cdot g_2^{s_{1,2}} \cdot O^{-e_1}$, $u_2 \leftarrow g_1^{k_2}$, $e \leftarrow H((y, h, m'), (u_1, u_2))$, and $e_2 \leftarrow e - e_1 \bmod q$. Finally compute $s_2 \leftarrow k_2 + e_2 x \bmod q$, and return $r' = \pi'$, where

$$\pi' \leftarrow (e_1, e_2, s_{1,1}, s_{1,2}, s_2).$$

Construction 2: Concrete instantiation from DLOG

5.2 Concrete Instantiation from Post-Quantum Primitives

Our post-quantum instantiation follows the paradigm of the previous instantiation, however leveraging the hardness of the Learning Parity with Noise (LPN) problem instead of that of DLOG, cf., e.g., Pietrzak [Pie12] for an overview. The computational LPN assumption says that it is computationally infeasible (and actually NP hard) to distinguish samples of the form $(A, As \oplus e)$ from such of the (A, r), where $A \leftarrow_r \{0,1\}^{k \times \lambda}$, $s \leftarrow_r \{0,1\}^\lambda$, $x \leftarrow_r \{0,1\}^k$, and $e \leftarrow_r \chi$; the computational problem is defined analogously. In the standard LPN problem, χ is an k-dimensional Bernoulli distribution with parameter τ, i.e., each entry of e equals 1 with probability τ and 0 otherwise. Following Jain et al. [JKPT12], we will rely on the *exact* LPN (xLPN) problem in the following, where χ is an k-dimensional Bernoulli distribution conditioned on $\|e\|_1 = \lceil k\tau \rfloor$ and $\lceil . \rfloor$ denotes rounding to the nearest integer. It is easy to see that xLPN is computationally related to the standard LPN problem. Let the message length be denoted by v, let $\tau \in [0, 0.25)$ and $k \in \mathcal{O}(v + \lambda)$ such that the linear code generated by $A \leftarrow_r \{0,1\}^{k \times (v+\lambda)}$ has a distance of more than $2\lceil k\tau \rfloor$ with overwhelming probability. The commitment scheme in [JKPT12] now works as follows. The public parameters consist of a matrix $A \leftarrow_r \{0,1\}^{k \times (v+\lambda)}$ and the value τ. A commitment to $m \in \{0,1\}^v$ is now given by choosing $\xi_1 \leftarrow_r \{0,1\}^\lambda$ and $\xi_2 \leftarrow_r \chi$, and

$\underline{\mathsf{CHPG}(1^{\lambda})}$: Outputs the public parameters (A, τ, H, H'), where $\mathsf{pp}_{\mathsf{ch}} = (A, \tau)$ for $A \leftarrow_r \{0,1\}^{(v+\lambda)\times k}$ and $\tau \in [0, 0.25)$ as above, $H : \{0,1\}^* \to \{0,1,2\}^{\ell}$ is a hash-function (which we assume to behave like a random oracle and to be implicitly available to all algorithms below), and $H' : \{0,1\}^* \to \{0,1\}^{2\lambda}$ is a cryptographic hash function (which will be used as a computationally binding and statistically hiding auxiliary commitment scheme).

$\underline{\mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})}$: Return $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) = ((x_1, x_2), y)$, where $x_1 \leftarrow_r \{0,1\}^{v+\lambda}$, $x_2 \leftarrow_r \chi$, and $y \leftarrow Ax_1 \oplus x_2$, where as before χ indicates a Bernoulli distribution with parameter τ conditioned on $\|x_2\|_1 = \lceil \tau k \rfloor$.

$\underline{\mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)}$: Parse $\mathsf{pk}_{\mathsf{ch}}$ as y, then proceed as follows:

- *Compute h*: Draw $\xi_1 \leftarrow_r \{0,1\}^{\lambda}$ and $\xi_2 \leftarrow_r \chi$. Set $h \leftarrow A(m\|\xi_1) \oplus \xi_2$.
- *Simulate proof for y*: Choose $e_2 \leftarrow_r \{0,1,2\}$ and $r_{2,0}, r_{2,1}, r_{2,2} \leftarrow_r \{0,1\}^{2\lambda}$.
  - If $e_2{=}0$, let $\pi_2 \leftarrow_r \mathcal{S}_k$, $v_2 \leftarrow_r \{0,1\}^{v+\lambda}$, $f_2 \leftarrow_r \{0,1\}^k$, $c_{2,0} \leftarrow_r H'((\pi_2, Av_2 \oplus f_2), r_{2,0})$, $c_{2,1} \leftarrow_r H'(\pi_2(f_2), r_{2,1})$ and $c_{2,2} \leftarrow_r H'(0, r_{2,3})$. Set $s_0 \leftarrow_r (\pi_2, Av_2 \oplus f_2, r_{2,0})$, $s_1 \leftarrow_r (\pi_2(f_2), r_{2,1})$, $s_2 \leftarrow_r \perp$.
  - If $e_2{=}1$, let $\pi_2 \leftarrow_r \mathcal{S}_k$, $b \leftarrow_r \{0,1\}^{v+\lambda}$, $a \leftarrow_r \{0,1\}^k$, $c_{2,0} \leftarrow_r H'((\pi_2, Ab \oplus y \oplus a), r_{2,0})$, $c_{2,1} \leftarrow_r H'(0, r_{2,1})$ and $c_{2,2} \leftarrow_r H'(\pi_2(a), r_{2,3})$. Set $s_0 \leftarrow_r (\pi_2, Ab \oplus y \oplus a, r_{2,0})$, $s_1 \leftarrow_r \perp$, $s_2 \leftarrow_r (\pi_2(a), r_{2,2})$.
  - If $e_2{=}2$, let $b \leftarrow_r \chi$, $a \leftarrow_r \{0,1\}^k$, $c_{2,0} \leftarrow_r H'(0, r_{2,0})$, $c_{2,1} \leftarrow_r H'(a, r_{2,1})$ and $c_{2,2} \leftarrow_r H'(a{\oplus}b, r_{2,3})$. Set $s_0 \leftarrow_r \perp$, $s_1 \leftarrow_r (a, r_{2,1})$, $s_2 \leftarrow_r (a{\oplus}b, r_{2,2})$.
- *Compute first message for h*: Choose $r_{1,0}, r_{1,1}, r_{1,2} \leftarrow_r \{0,1\}^{2\lambda}$. Draw $\pi_1 \leftarrow_r \mathcal{S}_k$, $v_1 \leftarrow_r \{0,1\}^{v+\lambda}$, $f_1 \leftarrow_r \{0,1\}^k$, $c_{1,0} \leftarrow_r H'((\pi_1, Av_1 \oplus f_1), r_{1,0}$, $c_{1,1} \leftarrow_r H'(\pi_1(f_1), r_{1,1})$ and $c_{1,2} \leftarrow_r H'(\pi(f_1 \oplus \xi_2), r_{1,2})$
- *Compute challenge for h*: Compute $e \leftarrow H((y, h, m), (c_{1,0}, c_{1,1}, c_{1,2}, c_{2,0}, c_{2,1}, c_{2,2}))$. Set $e_1 \leftarrow e - e_2 \bmod 3$.
- *Compute proof for h*:
  - If $e_1{=}0$, set $s_0 \leftarrow_r (\pi_1, Av_1 \oplus f_1, r_{1,0})$, $s_1 \leftarrow_r (\pi_1(f_1), r_{1,1})$, $s_2 \leftarrow_r \perp$.
  - If $e_1{=}1$, set $s_0 \leftarrow_r (\pi_1, Av_1 \oplus f_1, r_{1,0})$, $s_1 \leftarrow_r \perp$, $s_2 \leftarrow_r (\pi_1(f_1 \oplus \xi_2), r_{1,2})$.
  - If $e_1{=}2$, set $s_0 \leftarrow_r \perp$, $s_1 \leftarrow_r (\pi_1(f_1), r_{1,1})$, $s_2 \leftarrow_r (\pi_1(f_1 \oplus \xi_2), r_{1,2})$.
- *Generate output*: Return (h, r), where $r \leftarrow ((c_{1,0}, c_{1,1}, c_{1,2}, c_{2,0}, c_{2,1}, c_{2,2}), (e_1, e_2), (s_{1,0}, s_{1,1}, s_{1,2}, s_{2,0}, s_{2,1}, s_{2,2}))$.

Construction 3: Instantiation from LPN: Key Generation and Hashing

setting $h = A(m\|\xi_1) \oplus \xi_2$; upon receiving the opening (ξ_1, ξ_2) of a commitment h, one checks that h has the correct form and that $\|\xi_2\|_1 = \lceil \tau k \rfloor$.

As before, we use plain xLPN as a one-way function using the same generator matrix A, leading to the following language underlying our construction:

$$L := \{(y, h, m) \mid \exists (x_1, x_2, \xi_1, \xi_2) \ : \ h = A(m\|\xi_1) \oplus \xi_2 \ \vee \ y = Ax_1 \oplus x_2\}. \quad (3)$$

The zero-knowledge proofs for xLPN presented in [JKPT12] are based on those by Stern [Ste93] and come with a soundness error of $2/3$, therefore requiring about $\ell = 1.7\lambda$ parallel repetitions to achieve a soundness error or $2^{-\lambda}$. We note that the compiler to obtain simulation-sound extractability due to Faust et al. [FKMV12] also applies here: violating quasi-unique responses would imply

$\underline{\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h)}$: Parse $\mathsf{pk}_{\mathsf{ch}}$ as y and r as $((c_{1,0}, c_{1,1}, c_{1,2}, c_{2,0}, c_{2,1}, c_{2,2}), (e_1, e_2), (s_{1,0}, s_{1,1}, s_{1,2}, s_{2,0}, s_{2,1}, s_{2,2}))$, then proceed as follows:

- *Check message and challenges*: Return 0 if $m \notin \{0,1\}^v$ or $e_1 + e_2 \neq H((y, h, m), (c_{1,0}, c_{1,1}, c_{1,2}, c_{2,0}, c_{2,1}, c_{2,2}))$.
- *Verify proofs*: Return 1 if all following tests succeed for $j = 1, 2$, where $z \leftarrow h$ for $j = 1$ and $z \leftarrow y$ for $j = 2$, and return 0 otherwise:
  - If $e_j = 0$, parse $s_{j,0}$ as $((\pi_j, t_{j,0}), r_{j,0})$ and $s_{j,1}$ as $(t_{j,1}, r_{j,1})$. Check if $c_{j,0} = H'((\pi_j, t_{j,0}), r_{j,0})$ and $c_{j,1} = H'(t_{j,1}, r_{j,1})$. Check if $\pi_j \in \mathcal{S}_k$ and $t_{j,0} \oplus \pi_j^{-1}(t_{j,1}) \in \operatorname{img} A$.
  - If $e_j = 1$, parse $s_{j,0}$ as $((\pi_j, t_{j,0}), r_{j,0})$ and $s_{j,2}$ as $(t_{j,2}, r_{j,2})$. Check if $c_{j,0} = H'((\pi_j, t_{j,0}), r_{j,0})$ and $c_{j,2} = H'(t_{j,2}, r_{j,2})$. Check if $\pi_j \in \mathcal{S}_k$ and $t_{j,0} \oplus \pi_j^{-1}(t_{j,2}) \oplus z \in \operatorname{img} A$.
  - If $e_j = 2$, parse $s_{j,1}$ as $(t_{j,1}, r_{j,1})$ and $s_{j,2}$ as $(t_{j,2}, r_{j,2})$. Check if $c_{j,1} = H'(t_{j,1}, r_{j,1})$ and $c_{j,2} = H'(t_{j,2}, r_{j,2})$. Check if $\|t_{j,1} \oplus t_{j,2}\|_1 = \lceil \tau k \rfloor$.

$\underline{\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$: Parse $\mathsf{sk}_{\mathsf{ch}}$ as (x_1, x_2), and with $y = Ax_1 \oplus x_2$:

- *Check inputs*: Return $\perp$ if $\mathsf{CHCheck}(y, m, r, h) = 0$ or $m' \notin \{0,1\}^v$.
- *Simulate proof for h*: Choose $e_1 \leftarrow_r \{0,1,2\}$ and $r_{1,0}, r_{1,1}, r_{1,2} \leftarrow_r \{0,1\}^{2\lambda}$.
  - If $e_1{=}0$, let $\pi_1 \leftarrow_r \mathcal{S}_k$, $v_1 \leftarrow_r \{0,1\}^{v+\lambda}$, $f_1 \leftarrow_r \{0,1\}^k$, $c_{1,0} \leftarrow_r H'((\pi_1, Av_1 \oplus f_1), r_{1,0})$, $c_{1,1} \leftarrow_r H'(\pi_1(f_1), r_{1,1})$ and $c_{1,2} \leftarrow_r H'(0, r_{1,3})$. Set $s_0 \leftarrow_r (\pi_1, Av_1 \oplus f_1, r_{1,0})$, $s_1 \leftarrow_r (\pi_1(f_1), r_{1,1})$, $s_2 \leftarrow_r \perp$.
  - If $e_1{=}1$, let $\pi_1 \leftarrow_r \mathcal{S}_k$, $b \leftarrow_r \{0,1\}^{v+\lambda}$, $a \leftarrow_r \{0,1\}^k$, $c_{1,0} \leftarrow_r H'((\pi_1, Ab \oplus y \oplus a), r_{1,0})$, $c_{1,1} \leftarrow_r H'(0, r_{1,1})$ and $c_{1,2} \leftarrow_r H'(\pi_1(a), r_{1,3})$. Set $s_0 \leftarrow_r (\pi_1, Ab \oplus y \oplus a, r_{1,0})$, $s_1 \leftarrow_r \perp$, $s_2 \leftarrow_r (\pi_1(a), r_{1,2})$.
  - If $e_1{=}2$, let $b \leftarrow_r \chi$, $a \leftarrow_r \{0,1\}^k$, $c_{1,0} \leftarrow_r H'(0, r_{1,0})$, $c_{1,1} \leftarrow_r H'(a, r_{1,1})$ and $c_{1,2} \leftarrow_r H'(a \oplus b, r_{1,3})$. Set $s_0 \leftarrow_r \perp$, $s_1 \leftarrow_r (a, r_{1,1})$, $s_1 \leftarrow_r (a \oplus b, r_{1,2})$.
- *Compute first message for y*: Choose $r_{2,0}, r_{2,1}, r_{2,2} \leftarrow_r \{0,1\}^{2\lambda}$. Draw $\pi_2 \leftarrow_r \mathcal{S}_k$, $v_2 \leftarrow_r \{0,1\}^{v+\lambda}$, $f_2 \leftarrow_r \{0,1\}^k$, $c_{2,0} \leftarrow_r H'((\pi_2, Av_2 \oplus f_2), r_{2,0}$, $c_{2,1} \leftarrow_r H'(\pi_2(f_2), r_{2,1})$ and $c_{2,2} \leftarrow_r H'(\pi(f_2 \oplus \xi_2), r_{2,2})$.
- *Compute challenge for h*: Compute $e \leftarrow H((y, h, m), (c_{1,0}, c_{1,1}, c_{1,2}, c_{2,0}, c_{2,1}, c_{2,2}))$. Set $e_2 \leftarrow e - e_1 \bmod 3$.
- *Compute proof for y*:
  - If $e_2{=}0$, set $s_0 \leftarrow_r (\pi_2, Av_2 \oplus f_2, r_{2,0})$, $s_2 \leftarrow_r (\pi_2(f_2), r_{2,1})$, $s_2 \leftarrow_r \perp$.
  - If $e_2{=}1$, set $s_0 \leftarrow_r (\pi_2, Av_2 \oplus f_2, r_{2,0})$, $s_2 \leftarrow_r \perp$, $s_2 \leftarrow_r (\pi_2(f_1 \oplus x_2), r_{2,2})$.
  - If $e_2{=}2$, set $s_0 \leftarrow_r \perp$, $s_2 \leftarrow_r (\pi_2(f_2), r_{2,1})$, $s_2 \leftarrow_r (\pi_2(f_2 \oplus x_2), r_{x,2})$.
- *Generate output*: Return $r' \leftarrow_r ((c_{1,0}, c_{1,1}, c_{1,2}, c_{2,0}, c_{2,1}, c_{2,2}), (e_1, e_2), (s_{1,0}, s_{1,1}, s_{1,2}, s_{2,0}, s_{2,1}, s_{2,2})$.

Construction 4: Instantiation from LPN: Verify and Adapt

finding a collision for the hash function used to instantiate the random oracle and the statement is included when hashing the challenge. [6]

We present the instantiation as Construction 3-4 and note that for notational convenience and readability, we only consider $\ell = 1$. For practical parameters, all proofs need to be simulated or computed ℓ times in parallel, with the challenges

[6] We note that replacing LPN by learning with errors (LWE) and using the commitment scheme and zero-knowledge proofs of Benhamouda et al. [BKLP15] gives an immediate post-quantum instantiation that does not require parallel repetitions, yet requiring assumptions that give rise to public-key encryption.

being computed via a single invocation of H. In the following we denote the symmetric group on k elements (i.e., the set of all permutations on k elements) by $\mathcal{S}_k$. Furthermore, for $A \in \{0,1\}^{m \times n}$, $\mathrm{img}(A)$ we denote the image of the linear function characterized by A, i.e., $\mathrm{img}(A) = \{Ax \mid x \in \{0,1\}^n\}$. Checking whether or not $y \in \mathrm{img}(A)$ can efficiently be done by seeking a solution to the linear system $y = Ax$, e.g., using Gaussian elimination.

Acknowledgements. This work was supported by the European Union H2020 Programme under grant agreement n°830929 (CyberSec4Europe), the H2020 ECSEL Joint Undertaking under grant agreement n°783119 (SECREDAS), and by the Austrian Science Fund (FWF) and netidee SCIENCE under grant agreement P31621-N38 (PROFET).

References

[ACdMT05] Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik. Sanitizable signatures. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005, Proceedings*, volume 3679 of *Lecture Notes in Computer Science*, pages 159–177. Springer, 2005.

[ADK+13] Masayuki Abe, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, volume 7778 of *Lecture Notes in Computer Science*, pages 312–331. Springer, 2013.

[AdM04a] Giuseppe Ateniese and Breno de Medeiros. Identity-based chameleon hash and applications. In Ari Juels, editor, *Financial Cryptography, 8th International Conference, FC 2004, Key West, FL, USA, February 9-12, 2004. Revised Papers*, volume 3110 of *Lecture Notes in Computer Science*, pages 164–180. Springer, 2004.

[AdM04b] Giuseppe Ateniese and Breno de Medeiros. On the key exposure problem in chameleon hashes. In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, volume 3352 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 2004.

[AMVA17] Giuseppe Ateniese, Bernardo Magri, Daniele Venturi, and Ewerton R. Andrade. Redactable blockchain - or - rewriting history in bitcoin and friends. In *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017*, pages 111–126. IEEE, 2017.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[BCD+17] Michael Till Beck, Jan Camenisch, David Derler, Stephan Krenn, Henrich C. Pöhls, Kai Samelin, and Daniel Slamanig. Practical strongly invisible and strongly accountable sanitizable signatures. In Josef Pieprzyk and Suriadi Suriadi, editors, *Information Security and Privacy - 22nd*

Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3-5, 2017, Proceedings, Part I, volume 10342 of *Lecture Notes in Computer Science*, pages 437–452. Springer, 2017.

[BDD+11] Feng Bao, Robert H. Deng, Xuhua Ding, Junzuo Lai, and Yunlei Zhao. Hierarchical identity-based chameleon hash and its applications. In Javier López and Gene Tsudik, editors, *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*, volume 6715 of *Lecture Notes in Computer Science*, pages 201–219, 2011.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

[BFF+09] Christina Brzuska, Marc Fischlin, Tobias Freudenreich, Anja Lehmann, Marcus Page, Jakob Schelbert, Dominique Schröder, and Florian Volk. Security of sanitizable signatures revisited. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, volume 5443 of *Lecture Notes in Computer Science*, pages 317–336. Springer, 2009.

[BKKP15] Olivier Blazy, Saqib A. Kakvi, Eike Kiltz, and Jiaxin Pan. Tightly-secure signatures from chameleon hash functions. In Jonathan Katz, editor, *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, volume 9020 of *Lecture Notes in Computer Science*, pages 256–279. Springer, 2015.

[BKLP15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part I*, volume 9326 of *Lecture Notes in Computer Science*, pages 305–325. Springer, 2015.

[Blu81] Manuel Blum. Coin flipping by telephone. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981*, pages 11–15. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No 82-04, 1981.

[BM19] Alexandra Boldyreva and Daniele Micciancio, editors. *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*. Springer, 2019.

[BR08] Mihir Bellare and Todor Ristov. Hash functions from sigma protocols and improvements to VSH. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Aus-*

tralia, December 7-11, 2008. Proceedings, volume 5350 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2008.

[BR14] Mihir Bellare and Todor Ristov. A characterization of chameleon hash functions and new, efficient designs. *J. Cryptology*, 27(4):799–823, 2014.

[CDK^{+}17] Jan Camenisch, David Derler, Stephan Krenn, Henrich C. Pöhls, Kai Samelin, and Daniel Slamanig. Chameleon-hashes with ephemeral trapdoors - and applications to invisible sanitizable signatures. In Serge Fehr, editor, *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part II*, volume 10175 of *Lecture Notes in Computer Science*, pages 152–182. Springer, 2017.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010.

[CZSM07] Xiaofeng Chen, Fangguo Zhang, Willy Susilo, and Yi Mu. Efficient generic on-line/off-line signatures without key exposure. In Katz and Yung [KY07], pages 18–30.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. *IACR Cryptol. ePrint Arch.*, 2020:282, 2020.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In Boldyreva and Micciancio [BM19], pages 356–383.

[DHLW10] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 613–631. Springer, 2010.

[DS18] David Derler and Daniel Slamanig. Highly-efficient fully-anonymous dynamic group signatures. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, *Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018*, pages 551–565. ACM, 2018.

[DS19] David Derler and Daniel Slamanig. Key-homomorphic signatures: definitions and applications to multiparty signatures and non-interactive zero-knowledge. *Des. Codes Cryptogr.*, 87(6):1373–1413, 2019.

[DSS20] David Derler, Kai Samelin, and Daniel Slamanig. Bringing order to chaos: The case of collision-resistant chameleon-hashes. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 462–492. Springer, 2020.

[DSSS19] David Derler, Kai Samelin, Daniel Slamanig, and Christoph Striecks. Fine-grained and controlled rewriting in blockchains: Chameleon-hashing gone attribute-based. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019.

[EGM96] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *J. Cryptology*, 9(1):35–67, 1996.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the fiat-shamir transform. In Steven D. Galbraith and Mridul Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2012.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology - ASIACRYPT 2006, 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3-7, 2006, Proceedings*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459. Springer, 2006.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2008.

[HW09] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 654–670. Springer, 2009.

[HZM$^+$20] Ke Huang, Xiaosong Zhang, Yi Mu, Fatemeh Rezaeibagha, Xiaofen Wang, Jingwei Li, Qi Xia, and Jing Qin. EVA: efficient versatile auditing scheme for iot-based datamarket in jointcloud. *IEEE Internet Things J.*, 7(2):882–892, 2020.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysz-tof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China,* *December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 663–680. Springer, 2012.

[KPSS18] Stephan Krenn, Henrich C. Pöhls, Kai Samelin, and Daniel Slamanig. Chameleon-hashes with dual long-term trapdoors and their applications. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2018 - 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7-9, 2018, Proceedings*, volume 10831 of *Lecture Notes in Computer Science*, pages 11–32. Springer, 2018.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2000, San Diego, California, USA*, pages 143–154. The Internet Society, 2000.

[KY07] Jonathan Katz and Moti Yung, editors. *Applied Cryptography and Network Security, 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007, Proceedings*, volume 4521 of *Lecture Notes in Computer Science*. Springer, 2007.

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In Boldyreva and Micciancio [BM19], pages 326–355.

[Moh10] Payman Mohassel. One-time signatures and chameleon hash functions. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, volume 6544 of *Lecture Notes in Computer Science*, pages 302–319. Springer, 2010.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[Pie12] Krzysztof Pietrzak. Cryptography from learning parity with noise. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM 2012: Theory and Practice of Computer Science - 38th Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 21-27, 2012. Proceedings*, volume 7147 of *Lecture Notes in Computer Science*, pages 99–114. Springer, 2012.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 543–553. IEEE Computer Society, 1999.

[SBWP03] Ron Steinfeld, Laurence Bull, Huaxiong Wang, and Josef Pieprzyk. Universal designated-verifier signatures. In Chi-Sung Laih, editor, *Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 - December 4, 2003, Proceedings*, volume 2894 of *Lecture Notes in Computer Science*, pages 523–542. Springer, 2003.

[SS20] Kai Samelin and Daniel Slamanig. Policy-based sanitizable signatures. In Stanislaw Jarecki, editor, *Topics in Cryptology - CT-RSA 2020 - The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA,* *USA, February 24-28, 2020, Proceedings*, volume 12006 of *Lecture Notes in Computer Science*, pages 538–563. Springer, 2020.

[ST01] Adi Shamir and Yael Tauman. Improved online/offline signature schemes. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 355–367. Springer, 2001.

[Ste93] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 1993.

[Zha07] Rui Zhang. Tweaking TBE/IBE to PKE transforms with chameleon hash functions. In Katz and Yung [KY07], pages 323–339.

A Additional Property of Chameleon-Hashes

We now present some additional security notions and relations.

A.1 Full Indistinguishability

Full indistinguishability requires that a randomness r does not reveal whether it was generated using CHash or $\mathsf{CHAdapt}$, even if the adversary $\mathcal{A}$ controls all values, but the public parameters [SS20].[7]

```
$\mathbf{Exp}^{\mathsf{F\text{-}Indistinguishability}}_{\mathcal{A},\mathsf{CH}}(\lambda)$
  $\mathsf{pp_{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$
  $b \leftarrow_r \{0,1\}$
  $b^* \leftarrow_r \mathcal{A}^{\mathsf{HashOrAdapt}(\cdot,\cdot,\cdot,\cdot,b)}(\mathsf{pp_{ch}})$
    oracle $\mathsf{HashOrAdapt}$ on input $\mathsf{pk_{ch}}, \mathsf{sk_{ch}}, m, m', b$:
      $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, m')$
      $(h', r') \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, m)$
      $r'' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r', h')$
      return $\perp$, if $r'' = \perp \lor r' = \perp \lor r = \perp$
      if $b = 0$, return $(h, r)$
      if $b = 1$, return $(h', r'')$
  return 1, if $b^* = b$
  return 0
```

Fig. 5: Full Indistinguishability

Definition 15 (Full Indistinguishability). *A chameleon-hash* CH *offers full indistinguishability, if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that*

$$\left|\Pr[\mathbf{Exp}^{\mathsf{F\text{-}Indistinguishability}}_{\mathcal{A},\mathsf{CH}}(\lambda) = 1] - 1/2\right| \leq \nu(\lambda).$$

The corresponding experiments are depicted in Figure 5.

[7] Lifting this definition to also cover those parameters is straightforward.

A.2 Uniqueness

Camenisch et al. [CDK+17] defined a property called uniqueness. Uniqueness requires that for each hash/message pair, exactly one randomness can be found, even if the adversary $\mathcal{A}$ controls all values, but the public parameters.[8]

```
$\mathbf{Exp}^{\mathsf{Uniqueness}}_{\mathcal{A},\mathsf{CH}}(\lambda)$
  $\mathsf{pp_{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$
  $(\mathsf{pk}^*, m^*, r^*, r'^*, h^*) \leftarrow_r \mathcal{A}(\mathsf{pp_{ch}})$
  return 1, if $\mathsf{CHCheck}(\mathsf{pk}^*, m^*, r^*, h^*) = \mathsf{CHCheck}(\mathsf{pk}^*, m^*, r'^*, h^*) = 1 \ \wedge\ r^* \neq r'^*$
  return 0
```

Fig. 6: Uniqueness

Definition 16 (Uniqueness). *A chameleon-hash* CH *is unique, if for any PPT adversary $\mathcal{A}$ there exists a negligible function ν such that*

$$\Pr[\mathbf{Exp}^{\mathsf{Uniqueness}}_{\mathcal{A},\mathsf{CH}}(\lambda) = 1] \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 6.

The relations between randomness unforgeability and uniqueness are depicted in Figure 7.

$$\textsf{Full Collision-Resistance} \Longrightarrow \left(\textsf{Uniqueness} \Longrightarrow \textsf{Randomness-Unforgeability}\right)$$

Fig. 7: Relations between CH uniqueness properties

Theorem 5. *If* CH *is fully collision-resistant, then uniqueness is strictly stronger than randomness unforgeability.*

Proof. We first prove that uniqueness implies randomness unforgeability (assuming full collision-resistance), and then give a counterexample showing that the other direction of the implication does not hold, even if we assume full indistinguishability and full collision-resistance.

F-CollRes $\Longrightarrow$ (Uniqueness $\Longrightarrow$ Rand-Unforg)**:** Assume $\mathcal{A}$ to be an adversary who wins the randomness unforgeability game with some probability (non-negligibly) larger than 0. Consider the output (m^*, h^*, r^*). If m^* is fresh (thus was never input/output to the hash or adaption oracles), a standard reduction shows that the scheme is not fully collision-resistant. The proof

[8] Lifting this definition to also cover those parameters is straightforward.

essentially follows along the same lines as the proof for Theorem 4, and is therefore omitted. Thus, we can now assume that m^* is not "fresh". Next, we construct an adversary $\mathcal{B}$ which wins the uniqueness game. In particular, $\mathcal{B}$ proceeds as follows. It receives $\mathsf{pp_{ch}}$ from its own challenger, generates $(\mathsf{sk_{ch}}, \mathsf{pk_{ch}})$ honestly, and uses $\mathsf{pp_{ch}}$ and $\mathsf{pk_{ch}}$ to initialize $\mathcal{A}$. All queries to the collision-finding oracle are answered by $\mathcal{B}$ directly, while hashing is done honestly. Whenever $\mathcal{A}$ outputs (m^*, h^*, r^*), $\mathcal{B}$ outputs $(\mathsf{pk_{ch}}, m^*, r^*, r'^*, h^*)$ as its own forgery, where r'^* was generated by the reduction itself in on of the oracles. Hence, $\mathcal{B}$'s winning probability equals the one of $\mathcal{A}$, as the simulation is perfect.

$\mathsf{Rand\text{-}Unforg} \not\Rightarrow \mathsf{Uniqueness}$: Assume $\mathsf{CH} := (\mathsf{CHPG}, \mathsf{CHKG}, \mathsf{CHash}, \mathsf{CHCheck}, \mathsf{CHAdapt})$ to be a fully collision-resistant, randomness unforgeable, unique, and fully indistinguishable chameleon-hash. Let $\mathsf{CH}' := (\mathsf{CHPG}', \mathsf{CHKG}', \mathsf{CHash}', \mathsf{CHCheck}', \mathsf{CHAdapt}')$ be a chameleon-hash which internally uses CH but appends a random bit to each r. In particular let CH' be defined as follows: $\mathsf{CHPG}'(1^\lambda) := \mathsf{CHPG}(1^\lambda)$, $\mathsf{CHKG}'(\mathsf{pp_{ch}}) := \mathsf{CHKG}(\mathsf{pp_{ch}})$, $\mathsf{CHash}'(\mathsf{pk_{ch}}, m) := (h, (r, 0))$ **where** $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, (m, 0))$, $\mathsf{CHCheck}'(\mathsf{pk_{ch}}, m, r, h) := \mathsf{CHCheck}(\mathsf{pk_{ch}}, (m, r''), r', h)$ **where** $r = (r', r'')$, and $\mathsf{CHAdapt}'(\mathsf{sk_{ch}}, m, m', r', h) := (\mathsf{CHAdapt}(\mathsf{sk_{ch}}, (m, r''), (m, r''), r', h), r'')$ **where** $r = (r', r'')$. Clearly, CH' is still fully collision-resistant, randomness unforgeable, and fully indistinguishable, but changing the bit in the randomness r is trivial (for any non-outsider), breaking uniqueness unconditionally. □

B Proof of Theorem 4: Randomness Unforgeability

Before we prove the theorem, we introduce an intermediate security notion for Γ which we name opening hiding and prove a lemma which says that this security property directly follows from hiding and binding. We introduce this intermediate notion because it will make our proof more compact.

Essentially, this property says that an adversary cannot find any opening for a commitment it did not create by itself, even if it can adaptively query for new commitments on messages of its own choice.

In more detail, the challenger generates the public parameters pp_Γ honestly. The adversary is then initialized with pp_Γ and gets access to an oracle Commit'_Γ which the adversary can use to obtain commitments on messages of its own choice. Thus, we define a multi-challenge version. Again, this makes our proofs more readable. The generated commitments (and messages) are stored in a set $\mathcal{Q}$. The adversary wins, if it can generate any opening O^* (along with some message m^*) for a commitment generated by the Commit'_Γ oracle (checked via $\mathcal{Q}$) which makes Open_Γ verify correctly.

Definition 17 (Opening Hiding). *A non-interactive commitment scheme Γ is opening hiding, if for any PPT adversary $\mathcal{A}$, there exists a negligible functions ν such that*

$$\Pr[\mathbf{Exp}^{\mathsf{OH}}_{\mathcal{A},\Gamma}(\lambda) = 1] \leq \nu(\lambda),$$

where the corresponding experiment is depicted in Figure 8.

```
$\mathbf{Exp}^{\mathsf{OH}}_{\mathcal{A},\Gamma}(\lambda)$
  $\mathsf{pp}_\Gamma \leftarrow_r \mathsf{ParGen}_\Gamma(1^\lambda)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(C^*, O^*, m^*) \leftarrow_r \mathcal{A}^{\mathsf{Commit}'_\Gamma(\mathsf{pp}_\Gamma, \cdot)}(\mathsf{pp}_\Gamma)$
    where $\mathsf{Commit}'_\Gamma$ on input $\mathsf{pp}_\Gamma$, and $m$:
      $(C, O) \leftarrow_r \mathsf{Commit}_\Gamma(\mathsf{pp}_\Gamma, m)$
      return $\perp$, if $C = \perp$
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(C, O, m)\}$
      return $C$
  return 1, if $\mathsf{Open}_\Gamma(\mathsf{pp}_\Gamma, C^*, O^*, m^*) = 1 \ \wedge \ (C^*, \cdot, \cdot) \in \mathcal{Q}$
  return 0
```

Fig. 8: Opening Hiding

Lemma 1. *If Γ is binding and hiding, then Γ offers opening hiding.*

Proof (of Lemma 1). To prove opening hiding, we use a sequence of games.

Game 0: The original opening hiding game.

Game 1: As Game 0, but we abort if $(C^*, \cdot, m^*) \notin \mathcal{Q}$. Let this event be E_1.

Transition - Game 0 → Game 1: Game 0 and Game 1 proceed identically unless E_1 happens, i.e., we have that $|\Pr[S_0] - \Pr[S_1]| \leq \Pr[E_1]$. To show that the games are indistinguishable, we present a reduction which breaks binding of the underlying commitment with $\Pr[E_1]$. In particular, we replace the ParGen_Γ algorithm with an algorithm ParGen'_Γ that obtains pp_Γ from a binding challenger $\mathcal{C}^{\mathsf{binding}}$:

$\underline{\mathsf{ParGen}'_\Gamma(1^\lambda)}$:

$$\mathsf{pp}_\Gamma \leftarrow_r \mathsf{ParGen}_\Gamma(1^\lambda) \rightsquigarrow \boxed{\mathsf{pp}_\Gamma \leftarrow_r \mathcal{C}^{\mathsf{binding}}}.$$

In case E_1 happens, the reduction can directly return (C^*, O^*, O, m^*, m) for some $(C^*, O, m) \in \mathcal{Q}$ as its own forgery. Thus, we have that $|\Pr[S_0] - \Pr[S_1]| \leq \nu_{\mathsf{binding}}(\lambda)$.

Game 2: As Game 1, but we modify the Commit_Γ algorithm used inside the Commit'_Γ oracle as follows:

$\underline{\mathsf{Commit}''_\Gamma(\mathsf{pp}_\Gamma, m)}$:

$$(C, O) \leftarrow_r \ldots \rightsquigarrow \boxed{\text{choose } m' \neq m \text{ and set } (C, O) \leftarrow_r \mathsf{Commit}_\Gamma(\mathsf{pp}_\Gamma, m')}.$$

Transition - Game 1 → Game 2: We show that the probability to distinguish between Game 1 and Game 2 is negligible by presenting a reduction that uses a hiding challenger $\mathcal{C}^{\mathsf{hiding}}$ to interpolate between Game 1 and Game 2. In particular, we further modify the algorithms used inside the game as follows:

$\underline{\mathsf{ParGen}''_{\Gamma}(1^\lambda)}$:

$$\mathsf{pp}_\Gamma \leftarrow_r \mathsf{ParGen}_\Gamma(1^\lambda) \rightsquigarrow \boxed{\mathsf{pp}_\Gamma \leftarrow_r \mathcal{C}^{\mathsf{hiding}}}.$$

$\underline{\mathsf{Commit}'''_{\Gamma}(\mathsf{pp}_\Gamma, m)}$:

$$(C, O) \leftarrow_r \ldots \rightsquigarrow \boxed{\text{choose } m' \neq m \text{ and set } (C, \bot) \leftarrow_r \mathcal{C}^{\mathsf{hiding}}.\mathsf{Commit}'_\Gamma(m, m')}.$$

Depending on the challenger's bit we either simulate Game 1 or Game 2 and we have that we have that $|\Pr[S_1] - \Pr[S_2]| \leq \nu_{\mathsf{hiding}}(\lambda)$.

Reduction to binding: Now we are in a game where a reduction to binding is straightforward. We have already established in the previous game changes that the adversary can only return a tuple (C^*, O^*, m^*) where $(C^*, \cdot, m^*) \in \mathcal{Q}$ and that C^* is not a commitment to m^*. Hence we can use the message m' and the opening from the respective oracle call together with (C^*, O^*, m^*) to break binding. □

Proof (of Theorem 4). To prove randomness unforgeability, we use a sequence of games.

Game 0: The original randomness unforgeability game.

Game 1: As Game 0, but we alter CHash', $\mathsf{CHAdapt}'$, and the winning conditions as follows:

$\underline{\mathsf{CHash}''(1^\lambda)}$:

$$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(h, r)\} \rightsquigarrow \boxed{\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(h, r, m)\}}.$$

$\underline{\mathsf{CHAdapt}''(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$:

$$\exists (h, \cdot) \in \mathcal{Q} \rightsquigarrow \boxed{\exists (h, \cdot, \cdot) \in \mathcal{Q}} \text{ and}$$
$$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(h, r')\} \rightsquigarrow \boxed{\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(h, r', m')\}}$$

$\underline{\textbf{Winning Conditions}}$:

$$(h^*, \cdot) \in \mathcal{Q} \ \wedge \ (h^*, r'^*) \notin \mathcal{Q} \rightsquigarrow \boxed{(h^*, \cdot, \cdot) \in \mathcal{Q} \ \wedge \ (h^*, r'^*, \cdot) \notin \mathcal{Q}}.$$

Transition - Game 0 → Game 1: This change is conceptual and we have that $|\Pr[S_0] - \Pr[S_1]| = 0$.

Game 2: As Game 1, but we abort, if the adversary makes a query (m, m', r, h) to $\mathsf{CHAdapt}''$, for which $(h, \cdot, \cdot) \in \mathcal{Q} \ \wedge \ (h, \cdot, m) \notin \mathcal{Q} \ \wedge \ \mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h) = 1$.

Transition - Game 1 → Game 2: We bound the probability for an adversary to detect this game change by presenting a reduction which, in case of an abort, wins a full collision-resistance game presented by the challenger $\mathcal{C}^{\mathsf{ch\text{-}fcoll\text{-}res}}$. In particular, consider the following changes:

$\underline{\mathsf{CHPG}'(1^\lambda)}$:

$$\mathsf{crs}_\Pi \leftarrow_r \mathsf{PG}_\Pi(1^\lambda) \rightsquigarrow \boxed{\mathsf{crs}_\Pi \leftarrow_r \mathcal{C}^{\mathsf{ch\text{-}fcoll\text{-}res}}}.$$

$\underline{\mathsf{CHAdapt}'''(\mathsf{pk}_{\mathsf{ch}}, m)}$:

$$r' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h) \rightsquigarrow \boxed{r' \leftarrow_r \mathcal{C}^{\mathsf{ch\text{-}fcoll\text{-}res}}.\mathsf{CHAdapt}'(m, m', r, h)}.$$

Clearly, the simulation is perfect, while the above query let's the reduction win the full collision-resistance game. We have that $|\Pr[S_2] - \Pr[S_3]| \leq \nu_{\mathsf{ch\text{-}fcoll\text{-}res}}(\lambda)$.

Game 3: As Game 2, but we replace the CHPG' algorithm with an algorithm CHPG'' and modify the $\mathsf{CHAdapt}''$ and CHash'' oracles as follows:

$\underline{\mathsf{CHPG}'''(1^\lambda)}$:

$$\mathsf{crs}_\Pi \leftarrow_r \mathsf{PG}_\Pi(1^\lambda) \rightsquigarrow \boxed{(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda)}.$$

$\underline{\mathsf{CHash}'''(\mathsf{pk}_{\mathsf{ch}}, m)}$: In CHash:

$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pp}_\Gamma, h, m, y), (\bot, O)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m, y)).}$$

$\underline{\mathsf{CHAdapt}'''(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$: In $\mathsf{CHAdapt}$:

$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pp}_\Gamma, h, m, y), (x, \bot)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m, y)).}$$

Transition - Game 2 → Game 3: We bound the probability for an adversary to detect this game change by presenting a hybrid game, which, depending on a zero-knowledge challenger $\mathcal{C}^{\mathsf{zk}}$, either produces the distribution in Game 2 or Game 3, respectively. In particular, assume that we use the following algorithms:

$\underline{\mathsf{CHPG}'''(1^\lambda)}$:

$$(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda) \rightsquigarrow \boxed{\mathsf{crs}_\Pi \leftarrow_r \mathcal{C}^{\mathsf{zk}}}.$$

$\underline{\mathsf{CHash}''''(\mathsf{pk}_{\mathsf{ch}}, m)}$: In CHash:

$$\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m, y)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pp}_\Gamma, h, m, y), (O, \bot))}.$$

$\underline{\mathsf{CHAdapt}''''(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$: In $\mathsf{CHAdapt}$:

$$\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m', y)) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pp}_\Gamma, h, m', y), (\bot, x))}.$$

Clearly, if the challenger's internal bit is 0 we simulate the distribution in Game 2, whereas we simulate the distribution in Game 3 otherwise. We have that $|\Pr[S_2] - \Pr[S_3]| \leq \nu_{\mathsf{zk}}(\lambda)$.

Game 4: As Game 3, but we replace the CHPG''' algorithm with an algorithm CHPG'''' which works as follows:

$\underline{\mathsf{CHPG}''''(1^\lambda)}$:

$$(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda) \rightsquigarrow \boxed{(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_r \mathcal{E}_1(1^\lambda)}.$$

Transition - Game 3 → Game 4: Under simulation-sound extractability, Game 3 and Game 4 are indistinguishable. That is, $|\Pr[S_3] - \Pr[S_4]| = 0$.

Game 5: As Game 4, but for every valid collision (m^*, h^*, r^*) output by the adversary we observe that (m^*, r^*) must be a "fresh" collision w.r.t. h^* (while m^* was either input or output to the adaption oracle), i.e., one that was never seen by the collision-finding oracle. We run $(x', O') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pp}_\Gamma, h^*, m^*, y), r^*)$ and abort if the extraction fails. We call this event E_1.

Transition - Game 4 → Game 5: Game 4 and Game 5 proceed identically, unless E_1 occurs. Assume, towards contradiction, that event E_1 occurs with non-negligible probability. We now construct an adversary $\mathcal{B}$ which breaks the simulation-sound extractability property of the NIZK proof-system with non-negligible probability. We engage with a simulation-sound extractability challenger $\mathcal{C}^{\mathsf{sse}}$ and modify the algorithms as follows:

$\underline{\mathsf{CHPG}''''''(1^\lambda)}$:

$$(\mathsf{crs}_\Pi, \tau, \zeta) \leftarrow_r \mathcal{E}_1(1^\lambda) \rightsquigarrow \boxed{\mathsf{crs}_\Pi \leftarrow_r \mathcal{C}^{\mathsf{sse}}}.$$

$\underline{\mathsf{CHash}''''''(\mathsf{pk}_{\mathsf{ch}}, m)}$: In CHash

$$\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m, y)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathcal{C}^{\mathsf{sse}}.\mathsf{SIM}((\mathsf{pp}_\Gamma, h, m', y))}.$$

$\underline{\mathsf{CHAdapt}''''''''''(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$:

$$\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pp}_\Gamma, h, m', y)) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{sse}}.\mathsf{SIM}((\mathsf{pp}_\Gamma, h, m', y))}.$$

In the end we output $((\mathsf{pp}_\Gamma, h^*, m^*, y), r^*)$ to the challenger. This shows that we have $|\Pr[S_4] - \Pr[S_5]| \leq \nu_{\mathsf{sse}}(\lambda)$.

Reduction to Opening Hiding and One-Wayness: We are now ready to construct an adversary $\mathcal{B}$ which breaks the used one-way function or the binding property of the underlying Γ. Our adversary $\mathcal{B}$ proceeds as follows. It receives pp_Γ from its opening hiding challenger, as well as, f and y from a one-way challenger. It embeds them straightforwardly as $\mathsf{pp}_{\mathsf{ch}}$ and $\mathsf{pk}_{\mathsf{ch}}$ to initialize $\mathcal{A}$. It further (conceptually) modifies the CHash algorithm as follows:

$\underline{\mathsf{CHash}'''''''(\mathsf{pk}_{\mathsf{ch}}, m)}$: In CHash:

$$(C, O) \leftarrow_r \mathsf{Commit}_\Gamma(\mathsf{pp}_\Gamma, m) \rightsquigarrow \boxed{(C, \perp) \leftarrow_r \mathcal{C}^{\mathsf{OpeningHiding}}.\mathsf{Commit}'_\Gamma(m)}.$$

We now need to consider the following cases: we have extracted $(x', \perp)$ or $(\perp, O')$. In the first case, we have that $f(x') = y$ by SSE. Thus, we can directly return x' to the one-way challenger. In the other case, we know that O' is a fresh opening and we can return (m^*, h^*, O'). A union bound gives us that $\Pr[S_5] \leq \nu_{\mathsf{owf}}(\lambda) + \nu_{\mathsf{OH}}(\lambda)$. This, together with Lemma 1, concludes the proof. □