Bringing Order to Chaos: The Case of Collision-Resistant Chameleon-Hashes

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

[1] DFINITY, Zurich, Switzerland
david@dfinity.org
[2] TÜV Rheinland i-sec GmbH, Hallbergmoos, Germany
kaispapers@gmail.com
[3] AIT Austrian Institute of Technology, Vienna, Austria
daniel.slamanig@ait.ac.at

Abstract. Chameleon-hash functions, introduced by Krawczyk and Rabin at NDSS 2000, are trapdoor collision-resistant hash-functions parametrized by a public key. If the corresponding secret key is known, arbitrary collisions for the hash function can be efficiently found. Chameleon-hash functions have prominent applications in the design of cryptographic primitives, such as lifting non-adaptively secure signatures to adaptively secure ones. Recently, this primitive also received a lot of attention as a building block in more complex cryptographic applications ranging from editable blockchains to advanced signature and encryption schemes.

We observe that in latter applications various different notions of collision-resistance are used, and it is not always clear if the respective notion does really cover what seems intuitively required by the application. Therefore, we revisit existing collision-resistance notions in the literature, study their relations, and—using the example of the recent redactable blockchain proposals—discuss which practical impact different notions of collision-resistance might have. Moreover, we provide a stronger, and arguably more desirable, notion of collision-resistance than what is known from the literature. Finally, we present a surprisingly simple and efficient black-box construction of chameleon-hash functions achieving this strong notion.

1 Introduction

A chameleon-hash function (CH) is a trapdoor collision-resistant hash-function parameterized by a public key. If the corresponding secret key is known, arbitrary collisions for the hash function, i.e., distinct messages $m \neq m'$ yielding the same hash value h, can be efficiently found. Over the years, they have proven to be a very useful tool in theory, as well as practice. Exemplary, CHs are used to construct on/offline signatures [17, 26, 42], and to generically lift non-adaptively

secure signature schemes to adaptively secure ones (cf. [42]), see e.g., Hohenberger and Waters [35]. If CHs are tightly-secure, they are used to generically construct tightly-secure signatures [12]. Likewise, CHs are used to generically construct strong one-time signatures as shown by Mohassel [39], inspired by a concrete construction from Pedersen commitments by Groth [30]. Zhang [46] shows how to construct IND-CCA secure public-key encryption from tag-based encryption (TBE) or identity-based encryption (IBE) and CHs. Bellare and Ristov made the interesting discovery that chameleon-hashes and Σ-protocols, i.e., three round public-coin honest-verifier zero-knowledge proofs of knowledge, are equivalent [10,11]. CHs are also used to construct sanitizable signatures [3,14,15], i.e., signatures where a designated entity can modify certain parts of a signed message without invalidating the respective signature under controlled conditions. Furthermore, CHs have been used by Steinfeld et al. [44] to extend Schnorr and RSA signatures to the universal designated-verifier setting [43]. Also, different flavors of chameleon-hashing such as (hierarchical) identity-based [5,7] or policy-based chameleon-hash functions [21,41] have been studied.

In a more applied setting, CHs have shown to be valuable to construct integrity measurement and remote attestation mechanisms (denoted chameleon attestation) [2], and are used in vehicular ad-hoc networks (VANETs) [33] or handover authentication in mobile networks [18]. More recently, CHs have been used as a means to rewrite blocks in blockchains by replacing the hash function to chain blocks and/or to hash transactions by chameleon-hashes [4,21], to which we come back in Sect. 5. This brief discussion already shows that chameleon-hashes are used in a wide spectrum of different applications requiring different strength of the respective chameleon-hash. Consequently, authors often introduce some ad-hoc notion of collision-resistance for their applications, or even ignore that applications might require a stronger notion. Subsequently, we briefly discuss the different notions which are most commonly found in the literature.

Formalizing Chameleon-Hashes. The concept of chameleon-hashing dates back to the notion of trapdoor commitments introduced by Brassard et al. [13], and was firstly coined chameleon-hashing by Krawczyk and Rabin [37] with an instantiation based on the well-known trapdoor-commitment scheme by Pedersen [40]. Later, Ateniese and de Medeiros in [6] observed that the initial collision-resistance notion (which we denote W-CollRes) is rather weak (it does not give the adversary access to any collisions), and, more importantly, it is also satisfied by chameleon-hashes suffering from a key-exposure problem. Namely, when seeing a single collision for some hash h, it allows to publicly extract the secret trapdoor. Thus, any further guarantees are lost. While this is a desirable property for the initial use in chameleon signatures [37], and is also sufficient for the lifting compiler to adaptively secure signatures [42] (as no collision is ever revealed), it is too weak for many other applications. The key-exposure freeness definition in [6] is for the specific case of public-coin chameleon-hashing (where verifying the chameleon-hash is essentially re-computing it). To address this, Ateniese et al. [4] introduced a related notion called enhanced collision-resistance (which we denote E-CollRes) for the generalized case of secret-coin chameleon-hashing

(which is the setting that we also consider). The latter notion allows the adversary to see collisions, but it is not allowed to see any collision for the target hash, i.e., the hash corresponding to the collision it computes. Hence, once a single collision for a hash h is seen, an adversary can find arbitrary collisions for that particular hash h. Recently, Khalili et al. [36] have pointed out issues regarding the practicality of the concrete random-oracle model instantiation[4], proposed by Ateniese et al. in [4], and propose alternative constructions in the standard model. In another work Camenisch et al. [15] proposed an alternative collision-resistance notion which allows the adversary to see arbitrary collisions also for the target hash, but not for the target message, i.e., the message used in the collision output by the adversary has never been queried. In other words, once a collision for a message m is seen, an adversary is allowed to find arbitrary other hashes h' with the queried messages. Arguably, this notion seems more realistic as it is better compatible with practical applications (e.g., one can often make the messages unique by appending a tag/nonce), and thus we denote it as standard collision-resistance (or S-CollRes).

Motivation and Contribution. The previous discussion already illustrates that there are many different collision-resistance notions. While this does not necessarily point to an issue, we observe that it is not always clear whether the respective notion does really cover what is required by the respective application. Moreover, it is not clear if the last notion discussed above (S-CollRes) is already the most desirable notion, or, if even stronger notions are achievable, and do have practical relevance. Motivated by these observations, we provide the following contributions:

Relations among Properties. We discuss the different security notions of chameleon-hashes, and rigorously study relations among them. Most importantly, we, for the first time, clarify the picture of existing collision-resistance notions by showing implications, and separations, (cf. Figure 1 for an overview). In the course of showing separations, we also provide a construction of a chameleon-hash satisfying the E-CollRes notion, which clearly demonstrates weaknesses of this notion.

Stronger Notion. We find that the strongest existing collision-resistance notions, i.e., E-CollRes and S-CollRes (which are incomparable), might still be too weak for practical applications, see, e.g., Sect. 5. In particular, even if S-CollRes is satisfied, the hash values might still be malleable leaving space for potential real-world attacks. Consequently, we propose a stronger notion coined full collision-resistance (or F-CollRes for short), which enforces that the adversary cannot (except with negligible probability) output *any* new collisions and covers what one intuitively expects from collision-resistance.

[4] The requirement for an invertible encoding into the group introduces an enormous efficiency penalty, and thus their instantiation is incomplete. Moreover, it is possible that their schemes do meet our stronger definition of full collision-resistance, but we neither prove nor disprove this statement here.

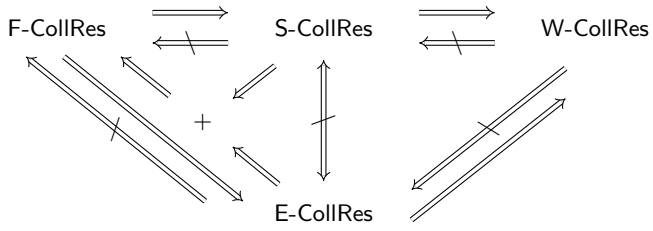

Fig. 1. Relations between CH collision-resistance properties

Black-Box Construction. We present a simple black-box construction of a chameleon-hash function satisfying this strong F-CollRes notion. Considering the complexity of existing constructions in [4, 36] which only achieve the weaker notion of E-CollRes, this is somewhat surprising. To recall, the construction from Ateniese et al. [4] starts from a public-coin chameleon-hash function that satisfies W-CollRes, uses an IND-CPA secure encryption scheme to encrypt the randomness of the chameleon-hash and then uses a true-simulation extractable (tSE) NIZK [25], which is in turn based on a NIZK and an IND-CCA secure public-key encryption scheme, to prove that the ciphertext is an encryption of the randomness. The constructions from Khalili et al. [36], which avoid the aforementioned issues with [4], are based on another new public-coin chameleon-hash function that satisfies W-CollRes and then either uses Groth-Sahai NIZK proofs [32] and the IND-CCA secure Cramer-Shoup encryption scheme [20] or a succinct non-interactive argument of knowledge (SNARK). Both constructions in [36] basically follow the generic template in [4]. In contrast, our black-box construction of a F-CollRes chameleon-hash is constructed from perfectly correct (multi-challenge) IND-CPA secure encryption, e.g., ElGamal encryption, and a simulation-sound extractable non-interactive zero-knowledge proof (SSE-NIZK), e.g., applying the compiler of Faust et al. [27] to a Fiat-Shamir transformed Σ-protocol. The basic idea is that the chameleon-hash is the encryption c of the message m and the randomness of the chameleon-hash is a NIZK proof s.t. either c correctly encrypts m under the pk of CH *or* one knows the secret key sk corresponding to pk. Interestingly, already a perfectly-binding commitment (without any hiding) is sufficient to achieve the F-CollRes notion, but instead a multi-challenge IND-CPA secure encryption scheme as a perfectly-binding commitment is used to additionally achieve the indistinguishability property of the CH, i.e., that fresh and adapted hashes are indistinguishable, a notion that is considered standard for chameleon-hashes.

Applications. We discuss how our stronger notion allows to strengthen the security of existing applications and in particular will discuss what problems may be caused by different notions of collision-resistance within recent applications to redactable blockchains [4, 21]. Here, either the hash function to chain blocks in a blockchain or the hash functions to aggregate transactions within single blocks (usually by means of a Merkle-tree) are replaced by a chameleon-hash function.

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 the output of a probabilistic algorithm A with input x is assigned to a 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. If we want to make the random coins used by an algorithm A explicit, we use the notation $a \leftarrow_r A(x;\xi)$. We write $(a;\xi) \leftarrow_r A(x)$, if we need to access the random coins ξ internally drawn by A. 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. 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 Building Blocks

We now present the building blocks we require. These include key-verifiable multi-challenge IND-CPA (mcIND-CPA) secure public-key encryption schemes Ω, digitall signature schemes Σ, and non-interactive zero-knowledge proofs Π.

Public-Key Encryption Schemes. Subsequently, we define public-key encryption schemes.

Definition 1 (Public-Key Encryption Scheme). *A public-key encryption scheme Ω consists of five algorithms $\{\mathsf{PG}_\Omega, \mathsf{KG}_\Omega, \mathsf{Enc}, \mathsf{Dec}, \mathsf{KVf}_\Omega\}$, such that:*

PG_Ω. *The algorithm PG_Ω outputs the public parameters of the scheme:*

$$\mathsf{pp}_\Omega \leftarrow_r \mathsf{PG}_\Omega(1^\lambda).$$

It is assumed that pp_Ω is an implicit input to all other algorithms.

KG_Ω. *The algorithm KG_Ω outputs the key pair, on input pp_Ω:*

$$(\mathsf{sk}_\Omega, \mathsf{pk}_\Omega) \leftarrow_r \mathsf{KG}_\Omega(\mathsf{pp}_\Omega).$$

Enc. *The algorithm Enc gets as input the public key pk_Ω, and a message $m \in \mathcal{M}$ to encrypt. It outputs a ciphertext c:*

$$c \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, m).$$

$\mathbf{Exp}^{\mathsf{mc\text{-}IND\text{-}CPA}}_{\mathcal{A},\Omega}(\lambda)$:
  $\mathsf{pp}_\Omega \leftarrow_r \mathsf{PG}_\Omega(1^\lambda)$
  $(\mathsf{sk}_\Omega, \mathsf{pk}_\Omega) \leftarrow_r \mathsf{KG}_\Omega(\mathsf{pp}_\Omega)$
  $b \leftarrow_r \{0,1\}$
  $a \leftarrow_r \mathcal{A}^{\mathsf{Enc}'(\mathsf{pk}_\Omega,\cdot,\cdot,b)}(\mathsf{pk}_\Omega)$
    where Enc' on input $\mathsf{pk}_\Omega, m_0, m_1, b$:
      If $m_0 \notin \mathcal{M} \vee m_1 \notin \mathcal{M} \vee |m_0| \neq |m_1|$:
        $c \leftarrow \perp$
      Else:
        $c \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, m_b)$
    return c
  return 1, if $a = b$
  return 0

Fig. 2. Multi-Challenge IND-CPA Security

Dec. *The deterministic algorithm* Dec *outputs a message* $m \in \mathcal{M} \cup \{\perp\}$ *on input* sk_Ω*, and a ciphertext* c*:*

$$m \leftarrow \mathsf{Dec}(\mathsf{sk}_\Omega, c).$$

KVf_Ω. *The deterministic algorithm* KVf_Ω *decides whether a given public key* pk_Ω *corresponds to a given secret key* sk_Ω*:*

$$d \leftarrow \mathsf{KVf}_\Omega(\mathsf{pk}_\Omega, \mathsf{sk}_\Omega).$$

Definition 2 (Correctness). *A public key encryption scheme* Ω *is called correct, if for all security parameters* $\lambda \in \mathbb{N}$*, for all* $\mathsf{pp}_\Omega \leftarrow_r \mathsf{PG}_\Omega(1^\lambda)$*, for all* $(\mathsf{sk}_\Omega, \mathsf{pk}_\Omega) \leftarrow_r \mathsf{KG}_\Omega(\mathsf{pp}_\Omega)$*, for all* $m \in \mathcal{M}$*, for all* $c \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, m)$*, we have that* $m = \mathsf{Dec}(\mathsf{sk}_\Omega, c)$ *and that for all* $\mathsf{sk}_\Omega{}'$ *we have that* $\mathsf{KVf}_\Omega(\mathsf{pk}_\Omega, \mathsf{sk}_\Omega{}') = 1 \implies m = \mathsf{Dec}(\mathsf{sk}_\Omega{}', c)$*.*

Definition 3 (Multi-Challenge IND-CPA Security). *A public-key encryption scheme* Ω *is multi-challenge IND-CPA secure (mcIND-CPA), if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that:*

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

The corresponding experiment is depicted in Figure 2.

Bellare et al. have shown, via a hybrid argument, that mcIND-CPA is equivalent to standard, i.e., "single-message", IND-CPA [8]. We opted for using mcIND-CPA, because it allows writing our proofs down more compactly, improving readability.

Digital Signature Schemes. Subsequently, we define signature schemes.

Definition 4 (Digital Signatures). *A digital signature scheme* Σ *consists of four algorithms* $\{\mathsf{PG}_\Sigma, \mathsf{KG}_\Sigma, \mathsf{Sgn}_\Sigma, \mathsf{Vrf}_\Sigma\}$ *such that:*

PG_Σ. *The algorithm* PG_Σ *outputs the public parameters*

$$\mathsf{pp}_\Sigma \leftarrow_r \mathsf{PG}_\Sigma(1^\lambda).$$

We assume that pp_Σ *contains* 1^λ *and is implicit input to all other algorithms.*

KG_Σ. *The algorithm* KG_Σ *outputs the public and private key of the signer, where* λ *is the security parameter:*

$$(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_r \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma).$$

Sgn_Σ. *The algorithm* Sgn_Σ *gets as input the secret key* sk_Σ *and the message* $m \in \mathcal{M}$ *to sign. It outputs a signature:*

$$\sigma \leftarrow_r \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, m).$$

Vrf_Σ. *The deterministic algorithm* Vrf_Σ *outputs a decision bit* $d \in \{0,1\}$*, indicating if the signature* σ *is valid, w.r.t.* pk_Σ *and* m*:*

$$d \leftarrow \mathsf{Vrf}_\Sigma(\mathsf{pk}_\Sigma, m, \sigma).$$

Definition 5 (Correctness). *A digital signature scheme* Σ *is called correct, if for all security parameters* $\lambda \in \mathbb{N}$*, for all* $\mathsf{pp}_\Sigma \leftarrow_r \mathsf{PG}_\Sigma(1^\lambda)$*, for all* $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_r \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma)$*, for all* $m \in \mathcal{M}$*,* $\mathsf{Vrf}_\Sigma(\mathsf{pk}_\Sigma, m, \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, m)) = 1$ *is true.*

We require existential unforgeability under adaptively chosen message attacks (eUNF-CMA security). In a nutshell, unforgeability requires that an adversary $\mathcal{A}$ cannot (except with negligible probability) come up with a signature for a message m^* for which the adversary did not see any signature before, even if the adversary $\mathcal{A}$ is allowed to adaptively query for signatures on messages of its own choice.

$\mathbf{Exp}^{\mathsf{eUNF\text{-}CMA}}_{\mathcal{A},\Sigma}(\lambda)$
  $\mathsf{pp}_\Sigma \leftarrow_r \mathsf{PG}_\Sigma(1^\lambda)$
  $(\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_r \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma)$
  $\mathcal{Q} \leftarrow \emptyset$
  $(m^*, \sigma^*) \leftarrow_r \mathcal{A}^{\mathsf{Sgn}'_\Sigma(\mathsf{sk}_\Sigma, \cdot)}(\mathsf{pk}_\Sigma)$
    where Sgn'_Σ on input sk_Σ and m:
      $\sigma \leftarrow_r \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, m)$
      set $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
      return σ
  return 1, if $\mathsf{Vrf}_\Sigma(\mathsf{pk}_\Sigma, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathcal{Q}$
  return 0

Fig. 3. Unforgeability

Definition 6 (Unforgeability). *We say a digital signature scheme Σ scheme is unforgeable, if for every PPT adversary $\mathcal{A}$, there exists a negligible function ν such that:*

$$\Pr\left[\mathbf{Exp}^{\mathsf{eUNF\text{-}CMA}}_{\mathcal{A},\Sigma}(\lambda) = 1\right] \leq \nu(\lambda).$$

The corresponding experiment is depicted in Figure 3.

For Construction 1, we require that the size of signatures is independent of the size of the signed messages.

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

PG_Π**.** *The algorithm PG_Π outputs public parameters of the scheme, where λ is the security parameter:*

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

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 8 (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. We define them analogously to the definitions given in [22].

Informally speaking, zero-knowledge says that the receiver of the proof π does not learn anything except the validity of the statement.

Definition 9 (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*

$\mathbf{Exp}^{\mathsf{Zero\text{-}Knowledge}}_{\mathcal{A},\Pi,\mathsf{SIM}}(\lambda)$
  $(\mathsf{crs}_\Pi, \tau) \leftarrow_r \mathsf{SIM}_1(1^\lambda)$
  $b \leftarrow_r \{0,1\}$
  $b^* \leftarrow_r \mathcal{A}^{P_b(\cdot,\cdot)}(\mathsf{crs}_\Pi)$
    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

Fig. 4. Zero-Knowledge

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 4.

Simulation-sound extractability says that every adversary who is able to come up with a proof π^* for a statement must know the witness, even when seeing simulated proofs for adaptively chosen statements potentially not in L. 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

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

Fig. 5. Simulation Sound Extractability

of [30] 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 [1, 25]) suffices.

Definition 10 (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

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

where the corresponding experiment is depicted in Figure 5.

3 Chameleon-Hashes, Revisited

In this section we present the formal framework for chameleon-hashes, their security properties with a special focus on the collision-resistance notion and then show relations and separations between the security properties.

3.1 Framework

We now present the framework for chameleon-hashes. We rely on the most recent comprehensive framework by Camenisch et al. [15], which is, in turn, based upon work done by Ateniese et al. and Brzuska et al. [4, 14].

Definition 11. *A chameleon-hash* CH *is a tuple of five PPT algorithms* (CHPG, CHKG, CHash, $\mathsf{CHCheck}$, $\mathsf{CHAdapt}$)*, such that:*

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

$$\mathsf{pp}_{\mathsf{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda).$$

We assume that $\mathsf{pp}_{\mathsf{ch}}$ *is implicit input to all other algorithms.*

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

$$(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}}).$$

CHash. *The algorithm* CHash *gets as input the public key* $\mathsf{pk}_{\mathsf{ch}}$*, and a message* m *to hash. It outputs a hash* h*, and some randomness* r*:*[5]

$$(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m).$$

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

$$d \leftarrow \mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h).$$

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

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

Definition 12 (Correctness). *A chameleon-hash is called correct, if for all security parameters* $\lambda \in \mathbb{N}$*, for all* $\mathsf{pp}_{\mathsf{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$*, for all* $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})$*, for all* $m \in \mathcal{M}$*, for all* $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)$*, for all* $m' \in \mathcal{M}$*, we have for all* $r' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)$*, that* $1 = \mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h) = \mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m', r', h)$*.*

3.2 Indistinguishability

Indistinguishability requires that the randomness r does not reveal if it was obtained through CHash or $\mathsf{CHAdapt}$. Upon setup, a challenger generates a key pair $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}})$ for CH (along with some public parameters), and draws a bit $b \leftarrow_r \{0, 1\}$. The challenger initializes the adversary with the $\mathsf{pk}_{\mathsf{ch}}$ and gives the adversary access to a $\mathsf{HashOrAdapt}$ oracle, which allows the adversary to submit two messages m, m'. Depending on the bit b, the challenger then either hashes m' directly ($b = 0$), of first hashes m, and then adapts m to m' ($b = 1$). The resulting hash/randomness pair (h, r) (or (h', r'') resp.) is the oracle's output to the adversary. The adversary's objective is to guess the bit b. Note that all keys are generated honestly and the adversary gets access to a collision-finding oracle $\mathsf{CHAdapt}$ for arbitrary hashes, meaning that the adversary may also input hashes generated by the $\mathsf{HashOrAdapt}$-oracle. We stress that there may be scenarios where indistinguishability is not required or even hindering.

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

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

where the corresponding experiment is depicted in Figure 6.

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

$\mathbf{Exp}^{\mathsf{Ind}}_{\mathcal{A},\mathsf{CH}}(\lambda)$
  $\mathsf{pp}_{\mathsf{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$
  $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})$
  $b \leftarrow_r \{0,1\}$
  $a \leftarrow_r \mathcal{A}^{\mathsf{HashOrAdapt}(\mathsf{sk}_{\mathsf{ch}},\cdot,\cdot,b),\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}},\mathsf{pk}_{\mathsf{ch}},\cdot,\cdot,\cdot)}(\mathsf{pk}_{\mathsf{ch}})$
    where $\mathsf{HashOrAdapt}$ on input $\mathsf{sk}_{\mathsf{ch}}, m, m', b$:
      $(h, r) \leftarrow \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m')$
      $(h', r') \leftarrow \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)$
      $r'' \leftarrow \mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r', h')$
      If $r = \bot \ \vee \ r'' = \bot$, return $\bot$
      if $b = 0$:
        return (h, r)
      if $b = 1$:
        return (h', r'')
  return 1, if $a = b$
  return 0

Fig. 6. CH Indistinguishability

Samelin and Slamanig recently introduced *full* indistinguishability [41], which, in turn, generalizes the notion of *strong* indistinguishability by Derler et al. [21]. In their notion, the adversary is even allowed to generate the keys which are used for hashing and adapting (in the strong version, the adversary only knows all keys, but cannot generate them).

We do neither consider full nor strong indistinguishability as fundamental for chameleon-hashes, but examine these notions to achieve a more complete picture of the relations. The formal definitions of full and strong indistinguishability are given in Appendix A, where we also prove that full indistinguishability is strictly stronger than strong indistinguishability, which, in turn, is strictly stronger than indistinguishability.

3.3 Collision-Resistance

In this section we revisit existing collision-resistance notions, introduce a stronger and more desirable notion of collision-resistance dubbed *full collision-resistance* (or F-CollRes for short) and discuss how these notions differ. The main idea behind collision-resistance in general is to argue that an adversary that has no access to the secret key $\mathsf{sk}_{\mathsf{ch}}$ cannot find any collisions, i.e,. pairs (m, r) and (m', r') and hash value h s.t. $\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h) = \mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m', r', h) = 1$. In the weakest case, the adversary has no access to any other collisions, whereas in stronger notions the adversary is explicitly allowed to obtain collisions for arbitrary hashes via a $\mathsf{CHAdapt}'$ oracle (we indicate these by using boxes). We present all the different notions in Figure 7, where we indicate the differences in the winning conditions by using boxes. In all the experiments the challenger generates a key pair $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}})$ honestly (along with some public parameters) and the adversary is then initialized with $\mathsf{pk}_{\mathsf{ch}}$. We now discuss the

Fig. 7. The $\mathbf{Exp}^{\mathsf{X\text{-}CollRes}}_{\mathcal{A},\mathsf{CH}}$ experiment with $\mathsf{X} \in \{\mathsf{W}, \mathsf{E}, \mathsf{S}, \mathsf{F}\}$.

differences of the single collision resistance notions, where in the weakest case the adversary has no access to an $\mathsf{CHAdapt}'$ oracle (which allows the adversary to adaptively ask for collisions with messages and hashes of its own choice), but in all other cases the adversary does. To vertically align the experiments, we insert boxes for lines which are missing in one experiment but are present in the other.

Weak Collision-Resistance (W-CollRes) [37]**.** The adversary $\mathcal{A}$ wins, if it can come up with a collision for the given public key.

Enhanced Collision-Resistance (E-CollRes) [4]**.** The adversary gets access to a collision-finding oracle $\mathsf{CHAdapt}'$, which outputs a collision for adversarially chosen hashes, but also keeps track of each queried *hash* h using the list $\mathcal{Q}$. The adversary wins, if it comes up with a collision for the given public key for an adverserially chosen hash h^* never input to $\mathsf{CHAdapt}'$.

Standard Collision-Resistance (S-CollRes) [15]. 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 *messages* m and m', using the list $\mathcal{Q}$. The adversary wins, if it comes up with a collision for the given public key for an adversarially chosen h^* for which the message m^* output by the adversary was never queried to the collision-finding oracle.

Full Collision-Resistance (F-CollRes). 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 *hash/message pair* (h, m) and (h, m'), using the list $\mathcal{Q}$. The adversary wins, if it comes up with a hash/message *pair* (h^*, m^*), for the given public key, never queried to or output from the collision-finding oracle.[6]

Now, we formally define security with respect to all the collision-resistance notions.

Definition 14 (X Collision-Resistance). *A chameleon-hash* CH *offers* X *collision-resistance with* $\mathsf{X} \in \{\mathsf{W}, \mathsf{E}, \mathsf{S}, \mathsf{F}\}$*, if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that*

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

where the corresponding experiment is depicted in Figure 7.

Discussion of the Notions. W-CollRes is the notion introduced in the first work on chameleon-hashes by Krawczyk and Rabin [37] and essentially represents the binding notion of a trapdoor-commitment scheme. Note that due to not giving access to a collision-finding oracle it gives no guarantees whatsoever if the adversary sees a single collision for any hash computed for the given public key.[7] The E-CollRes notion has been introduced by Ateniese et al. [4] and we note that there exists a definition in the setting of public-coin chameleon hashes, i.e., where the CHCheck algorithm simply re-runs the CHash, which is called key-exposure freeness [6, 16]. It captures requirements similar to the ones captured by E-CollRes, but it is not directly comparable as we are considering the more general secret-coin setting. We note that the E-CollRes notion allows the adversary to come up with arbitrary collisions for hashes it has seen a collision for. The S-CollRes notion has been introduced by Camenisch et al. [15], and it captures all of the intuitive requirements of real-world applications of chameleon-hashes. Yet, it still allows the hash itself to be malleable which might still be problematic in certain applications. Finally, our new F-CollRes notion enforces that the adversary cannot (except with negligible probability) output any new collisions and seems to be the most desirable notion for collision-resistance.

[6] In the case (h'^*, m'^*) is the new hash/message pair, simply switch names.

[7] A slightly stronger notion has been proposed by Zhang in [46] where the adversary sees a hash on a random message and is then given a single collision on a message of its choice. We do not cover this notion here as it seems to be tailored to the specific applications in [46] and all notions stronger than W-CollRes considered here cover more general cases.

3.4 Uniqueness

Camenisch et al. [15] 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]

$\mathbf{Exp}^{\mathsf{Uniqueness}}_{\mathcal{A},\mathsf{CH}}(\lambda)$
$\mathsf{pp}_{\mathsf{ch}} \leftarrow_r \mathsf{CHPG}(1^\lambda)$
$(\mathsf{pk}^*, m^*, r^*, r'^*, h^*) \leftarrow_r \mathcal{A}(\mathsf{pp}_{\mathsf{ch}})$
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. 8. Uniqueness

Definition 15 (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 8.

We do not consider uniqueness as a fundamental property, as there are only very few applications requiring this notion [15, 41]. However, to obtain a more complete picture with respect to the relations of the security properties, we also investigate uniqueness.

3.5 Relationships between Properties

Below we show relations and separations between the security properties of chameleon-hashes.

Collision-Resistance Properties. We start by analyzing how the various collision-resistance notions are related.

Theorem 1. *Standard collision-resistance is strictly stronger than weak collision-resistance.*

Proof. We first prove that standard collision-resistance implies weak collision-resistance and then give a counterexample showing that the other direction of the implication does not hold.

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

S-CollRes $\implies$ W-CollRes: Assume $\mathcal{A}$ to be an adversary who breaks weak collision-resistance. We now construct an adversary $\mathcal{B}$ which breaks standard collision-resistance. In particular, $\mathcal{B}$ proceeds as follows. It receives $\mathsf{pp}_{\mathsf{ch}}$ and $\mathsf{pk}_{\mathsf{ch}}$ from its own challenger, and uses both to initialize $\mathcal{A}$. Whenever $\mathcal{A}$ outputs a winning tuple $(m^*, r^*, m'^*, r'^*, h^*)$, $\mathcal{B}$ returns that tuple to its own challenger. As the collision-finding oracle was never queried, that tuple also makes $\mathcal{B}$ win the standard collision-resistance game with the same probability $\mathcal{A}$ wins the weak collision-resistance game.

W-CollRes $\not\implies$ S-CollRes: The CH by Krawczyk and Rabin [37] provides a counterexample: it is weakly collision-resistant, but does not offer standard collision-resistance. Observe that it is possible to trivially extract the secret key from a collision. That collision is obtained from the collision-finding oracle in the standard collision-resistance game (cf. Appendix B.1 for more details).

□

Theorem 2. *Enhanced collision-resistance is strictly stronger than weak collision-resistance.*

Proof. The proof is identical to the one of Theorem 1. □

Theorem 3. *Full collision-resistance is strictly stronger than standard collision-resistance.*

Proof. We first prove that full collision-resistance implies standard collision-resistance and then give a counterexample showing that the other direction of the implication does not hold.

F-CollRes $\implies$ S-CollRes: Assume $\mathcal{A}$ to be an adversary who breaks standard collision-resistance. Now we construct an adversary $\mathcal{B}$ which breaks full collision-resistance. In particular, $\mathcal{B}$ proceeds as follows. It receives $\mathsf{pp}_{\mathsf{ch}}$ and $\mathsf{pk}_{\mathsf{ch}}$ from its own challenger, and uses both to initialize $\mathcal{A}$. All queries to the collision-finding oracle are relayed to $\mathcal{B}$'s own oracle. Whenever $\mathcal{A}$ outputs a winning tuple $(m^*, r^*, m'^*, r'^*, h^*)$, $\mathcal{B}$ returns that tuple to its own challenger. As $m^* \neq m'^*$ must be true, and m^* was never queried to $\mathcal{A}$'s collision-finding oracle, this also means that (h^*, m^*) was never queried to $\mathcal{B}$'s oracle, thus meeting the winning condition.

S-CollRes $\not\implies$ F-CollRes: The scheme by Camenisch et al. [15] provides a counterexample: it offers standard collision-resistance, but does not offer full collision-resistance. In particular, their construction is re-randomizable (cf. Appendix B.2 for more details). In more detail, to show that this construction is not fully collision-resistant, consider the following strategy: Receive $\mathsf{pk}_{\mathsf{ch}} = (N, H)$ and $\mathsf{pp}_{\mathsf{ch}} = e$. Compute $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)$, with m random. Then, ask for an adaption (h, r, m) to (h, r', m'), for some random $m' \neq m$. Then, compute $h^* \leftarrow h2^e \mod N$, $r_1^* \leftarrow 2r \mod N$, and $r_2^* \leftarrow 2r' \mod N$. Because no collision for h^* was computed, this construction cannot be fully collision-resistant. Note, this works, as $H(m)(2r)^e \equiv h2^e \pmod N$ for any input. Also note that the attack above also breaks enhanced collision-resistance (we will later use this to derive a corollary). □

Theorem 4. *Full collision-resistance is strictly stronger than enhanced collision-resistance.*

Before we provide the proof of Theorem 4, we provide a novel construction of a chameleon-hash satisfying the E-CollRes notion that is used to separate the notions F-CollRes and E-CollRes.

Construction. Our CH presented below provides E-CollRes, but allows to efficiently find arbitrary collisions for a given hash, once a single collision was seen. However, it is not possible to find collisions for any other hash. The main idea is to encrypt a message m using a mcIND-CPA secure encryption scheme Ω and use the ciphertext as the hash. The randomness r of the chameleon-hash is the public key $\mathsf{pk}_\Omega{}'$ of a freshly sampled key-pair $(\mathsf{sk}_\Omega{}', \mathsf{pk}_\Omega{}')$ of Ω, the encryption c' of a signature σ under $\mathsf{pk}_\Omega{}'$ and a SSE NIZK π for the following language:

$$L := \{(\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m) \mid \exists\ (\sigma, \xi)\ : \\ h = \mathsf{Enc}(\mathsf{pk}_\Omega, m;\ \xi)\ \vee\ \mathsf{Vrf}_\Sigma(\mathsf{pk}_\Sigma, h, \sigma) = 1\}. \tag{1}$$

Informally, this language requires the prover to show that it either knows the randomness ξ attesting that h is a well-formed encryption of m, *or* a valid signature σ for h. The basic idea of the construction is that when computing a hash, the witness ξ is used. The randomness includes an encryption of the signature (initially one on 0) under the public key $\mathsf{pk}_\Omega{}'$. Note that the trick is that for adaption one computes a signature σ for h, uses σ as a witness, and includes an encryption of σ under $\mathsf{pk}_\Omega{}'$ in the randomness. Clearly, now seeing a single collision allows to compute arbitrary collisions for the hash h.

This CH can be instantiated by instantiating Σ as structure-preserving signatures (SPS) in type-III bilinear groups (assuming SXDH), e.g., Groth's SPS [31]. Thus, Ω can be ElGamal [29] in one of the base-groups. The algorithm KVf_Ω is simply checking whether $g^{\mathsf{sk}_\Omega} = g^x = \mathsf{pk}_\Omega$, while for Π, a suitable instantiation is a Fiat-Shamir transformed Σ-protocol in the random-oracle model [28], which also works very well with ElGamal encryption and Groth's signature scheme.

We defer the proof of Construction 1 to the Appendix C. We are now ready to present the proof of Theorem 4.

Proof. We first prove that full collision-resistance implies enhanced collision-resistance and then give a counterexample showing that the other direction of the implication does not hold.

F-CollRes $\implies$ E-CollRes: Assume $\mathcal{A}$ to be an adversary who breaks the enhanced collision-resistance. We can then construct an adversary $\mathcal{B}$ which breaks the full collision-resistance. In particular, $\mathcal{B}$ proceeds as follows. It receives pp_ch and pk_ch from its own challenger, and uses both to initialize $\mathcal{A}$. All queries to the collision-finding oracle are relayed to $\mathcal{B}$'s own oracle. Whenever $\mathcal{A}$ outputs a winning tuple $(m^*, r^*, m'^*, r'^*, h^*)$, $\mathcal{B}$ returns that tuple to its own challenger. As $m^* \neq m'^*$ must be true, and h^* was never queried to $\mathcal{A}$'s collision-finding oracle, this also means that (h^*, m^*) was never queried to $\mathcal{B}$'s oracle, thus meeting the winning condition.

$\underline{\mathsf{CHPG}(1^\lambda)}$: Fix a public-key encryption scheme Ω, a signature scheme Σ, and a compatible NIZK proof system for language L in (1). Return $\mathsf{pp_{ch}} = (\mathsf{pp}_\Omega, \mathsf{pp}_\Sigma, \mathsf{crs}_\Pi)$, where

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

$\underline{\mathsf{CHKG}(\mathsf{pp_{ch}})}$: Return $(\mathsf{sk_{ch}}, \mathsf{pk_{ch}}) = ((\mathsf{sk}_\Omega, \mathsf{sk}_\Sigma), (\mathsf{pp_{ch}}, \mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, \sigma_0))$, where

$$(\mathsf{sk}_\Omega, \mathsf{pk}_\Omega) \leftarrow_r \mathsf{KG}_\Omega(\mathsf{pp}_\Omega), (\mathsf{sk}_\Sigma, \mathsf{pk}_\Sigma) \leftarrow_r \mathsf{KG}_\Sigma(\mathsf{pp}_\Sigma), \text{ and } \sigma_0 \leftarrow_r \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, 0).$$

0 is considered some special invalid hash value for CH.

$\underline{\mathsf{CHash}(\mathsf{pk_{ch}}, m)}$: Parse $\mathsf{pk_{ch}}$ as $((\mathsf{pp}_\Omega, \mathsf{crs}_\Pi), \mathsf{pk}_\Omega)$, and return $(h, r) = (c, (\pi, c', \mathsf{pk}_\Omega{}'))$, where

$$(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, m), (\mathsf{sk}_\Omega{}', \mathsf{pk}_\Omega{}') \leftarrow_r \mathsf{KG}_\Omega(\mathsf{pp}_\Omega), c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega{}', \sigma_0), \text{ and}$$
$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, c, m), (\bot, \xi))$$

$\underline{\mathsf{CHCheck}(\mathsf{pk_{ch}}, m, r, h)}$: Parse $\mathsf{pk_{ch}}$ as $((\mathsf{pp}_\Omega, \mathsf{crs}_\Pi), \mathsf{pk}_\Omega)$ and r as $(\pi, c', \mathsf{pk}_\Omega{}')$, and return 1 if the following holds, and 0 otherwise:

$$m \in \mathcal{M} \ \wedge\ \mathsf{Vfy}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m), \pi) = 1.$$

$\underline{\mathsf{CHAdapt}(\mathsf{sk_{ch}}, m, m', r, h)}$: Parse $\mathsf{sk_{ch}}$ as sk_Ω. Verify that $m' \in \mathcal{M}$, $\mathsf{CHCheck}(\mathsf{pk_{ch}}, m, r, h) = 1$, and return $\bot$ if not. Otherwise, return $r' = (\pi', c'', \mathsf{pk}_\Omega{}')$, where

$$\sigma \leftarrow_r \mathsf{Sgn}_\Sigma(\mathsf{sk}_\Sigma, h), c'' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega{}', \sigma), \text{ and}$$
$$\pi' \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m'), (\sigma, \bot)).$$

Construction 1: Enhanced Collision-Resistant Chameleon-Hash

E-CollRes $\not\Longrightarrow$ F-CollRes: The scheme presented in Construction 1 gives a counterexample: it allows finding arbitrarily many collisions for a given hash h, if it sees a single one, but for no other $h' \neq h$. In more detail, to show that this construction is not fully collision-resistant, consider the following strategy. Receive $\mathsf{pk_{ch}} = (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma)$ and $\mathsf{pp_{ch}} = (\mathsf{pp}_\Omega, \mathsf{crs}_\Pi, \mathsf{pp}_\Sigma)$. Compute $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk_{ch}}, m)$, with m random. Also store the secret key $\mathsf{sk}_\Omega{}'$. Then, ask for an adaption (h, r, m) to (h, r', m'), where $r' = (\pi, c'', \mathsf{pk}_\Omega{}')$, for some random m'. Then, compute $\sigma \leftarrow \mathsf{Dec}(\mathsf{sk}_\Omega{}', c'')$. Then arbitrary collisions for h are generated by executing $\mathsf{CHAdapt}$ in a similar way the owner of $\mathsf{pk_{ch}}$ does for finding collisions, due to the knowledge of σ for h. Because such collisions can only be generated for already seen collisions w.r.t. h, enhanced collision-resistance holds, but full collision-resistance does not. Also note that standard collision-resistance does not hold for Construction 1 for the same reason (we will later use this to derive a corollary). □

Theorem 5. *Enhanced collision-resistance and standard collision-resistance together imply full collision-resistance.*

Proof. The theorem above is proven using a sequence of games.

Game 0: The original full collision-resistance game.

Game 1: As Game 0, we abort, if the adversary $\mathcal{A}$ outputs $(m^*, r^*, m'^*, r'^*, h^*)$ such that the winning conditions are met, but h^* was never queried to the collision-finding oracle.

Transition - Game 0 → Game 1: If this is the case, we build an adversary $\mathcal{B}$ which breaks the enhanced collision-resistance of the underlying scheme. Namely, $\mathcal{B}$ receives $\mathsf{pk}_{\mathsf{ch}}$ and uses it to initialize $\mathcal{A}$. Every adaption query by $\mathcal{A}$ is answered by $\mathcal{B}$ using its own oracle. Once $\mathcal{A}$ outputs $(m^*, r^*, m'^*, r'^*, h^*)$, $\mathcal{B}$ returns $(m^*, r^*, m'^*, r'^*, h^*)$ to its own challenger. As h^* was never seen, $\mathcal{B}$ wins its own game. $|\Pr[S_0] - \Pr[S_1]| \leq \nu_{\mathsf{enh\text{-}collres}}(\lambda)$ follows.

Game 2: As Game 1, we abort, if the adversary $\mathcal{A}$ outputs $(m^*, r^*, m'^*, r'^*, h^*)$ such that the winning conditions are met, but m^* was never queried to the collision-finding oracle.

Transition - Game 1 → Game 2: If this is the case, we build an adversary $\mathcal{B}$ which breaks the standard collision-resistance of the underlying scheme. Namely, $\mathcal{B}$ receives $\mathsf{pk}_{\mathsf{ch}}$ and uses it to initialize $\mathcal{A}$. Every adaption query by $\mathcal{A}$ is answered by $\mathcal{B}$ using its own oracle. Once $\mathcal{A}$ outputs $(m^*, r^*, m'^*, r'^*, h^*)$, $\mathcal{B}$ returns $(m^*, r^*, m'^*, r'^*, h^*)$ to its own challenger. As m^* was never seen, $\mathcal{B}$ wins its own game. $|\Pr[S_1] - \Pr[S_2]| \leq \nu_{\mathsf{st\text{-}collres}}(\lambda)$ follows.

In Game 2, the adversary can no longer win the full collision-resistance game. This proves the theorem. □

The corollary below follows from the constructions used in the proofs of Theorem 3 and Theorem 4, which provide standard collision-resistance but not enhanced collision-resistance, and vice versa.

Corollary 1. *Standard collision-resistance and enhanced collision-resistance are independent.*

Additional Separations. We now prove some additional separations. We note that indistinguishability is strictly weaker than full indistinguishability (as formally shown in Appendix A).

Theorem 6. *Even full indistinguishability and uniqueness together do not imply weak collision-resistance.*

Proof. Assume the following contrived construction of a chameleon-hash: $\mathsf{CHPG}'(1^\lambda) := \emptyset$, $\mathsf{CHKG}'(\mathsf{pp}_{\mathsf{ch}}) := \emptyset$, $\mathsf{CHash}'(\mathsf{pk}_{\mathsf{ch}}, m) := (\emptyset, \emptyset)$, $\mathsf{CHCheck}'(\mathsf{pk}_{\mathsf{ch}}, m, r, h) :=$ **if** $h = \emptyset \ \wedge \ \mathsf{pk}_{\mathsf{ch}} = \emptyset \ \wedge \ r = \emptyset$ **then** 1 **else** 0, $\mathsf{CHAdapt}'(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h) :=$ **if** $\mathsf{CHCheck}'(\mathsf{pk}_{\mathsf{ch}}, m, r, h) = 1$ **then** $\emptyset$ **else** $\perp$. Clearly, this construction is fully indistinguishable and unique. Finding collisions, however, is a trivial task. □

Theorem 7. *Even full collision-resistance and uniqueness together do not imply indistinguishability.*

Proof. Assume $\mathsf{CH} := (\mathsf{CHPG}, \mathsf{CHKG}, \mathsf{CHash}, \mathsf{CHCheck}, \mathsf{CHAdapt})$ to be a fully collision-resistant, 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 m to the hash. CH' is defined as: $\mathsf{CHPG}'(1^\lambda) := \mathsf{CHPG}($

1^λ), $\mathsf{CHKG}'(\mathsf{pp}_{\mathsf{ch}}) := \mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})$, $\mathsf{CHash}'(\mathsf{pk}_{\mathsf{ch}}, m) := ((h, m), r)$ **where** $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, (m, m))$, and also $\mathsf{CHCheck}'(\mathsf{pk}_{\mathsf{ch}}, m, r, h) := \mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, (m, \hat{m}), r, h')$ **where** $h' = (h, \hat{m})$, and $\mathsf{CHAdapt}'(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h') := (\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, (m, \hat{m}), (m', \hat{m}), r', h))$ **where** $h' = (h, \hat{m})$. Clearly, CH' is still fully collision-resistant and unique, but looking at the appended messages allows deciding whether an adaption has occurred. □

Theorem 8. *Even full collision-resistance and full indistinguishability together do not imply uniqueness.*

Proof. Assume $\mathsf{CH} := (\mathsf{CHPG}, \mathsf{CHKG}, \mathsf{CHash}, \mathsf{CHCheck}, \mathsf{CHAdapt})$ to be a fully collision-resistant, 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: $\mathsf{CHPG}'(1^\lambda) := \mathsf{CHPG}(1^\lambda)$, $\mathsf{CHKG}'(\mathsf{pp}_{\mathsf{ch}}) := \mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})$, $\mathsf{CHash}'(\mathsf{pk}_{\mathsf{ch}}, m) := (h, (r, 0))$ **where** $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)$, $\mathsf{CHCheck}'(\mathsf{pk}_{\mathsf{ch}}, m, r, h) := \mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r', h)$ **where** $r = (r', \cdot)$, $\mathsf{CHAdapt}'(\mathsf{sk}_{\mathsf{ch}}, m, m', r', h) := (\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r', h), 0)$ **where** $r = (r', \cdot)$. Clearly, CH' is still fully collision-resistant and fully indistinguishable, but changing the bit in the randomness r is trivial, breaking uniqueness trivially. □

4 Fully Collision-Resistant Chameleon-Hashes

We are now ready to present our black-box construction of fully collision-resistant chameleon-hashes.

4.1 Construction

The main idea of our construction is to encrypt a message m using an mcIND-CPA secure encryption scheme and use the ciphertext as the hash, i.e., it is very close to our "contrived" construction providing enhanced collision-resistance given in Construction 1. However, it has some important, and subtle, differences.

Namely, the randomness r is a SSE NIZK attesting membership of a tuple containing the public key used for encryption, the hash, as well as the hashed message in the following NP-language:

$$L := \{(\mathsf{pk}_\Omega, h, m) \mid \exists\, (\mathsf{sk}_\Omega, \xi) \;:\; h = \mathsf{Enc}(\mathsf{pk}_\Omega, m;\, \xi) \vee \mathsf{KVf}_\Omega(\mathsf{pk}_\Omega, \mathsf{sk}_\Omega) = 1\}. \quad (2)$$

Informally, this language requires the prover to demonstrate that it either knows the randomness ξ attesting that h is a well-formed encryption of m under the CH key pk_Ω, *or* it knows a secret key sk_Ω corresponding to pk_Ω, instead of encrypting a signature and proving the verification relation. Our construction of a fully collision-resistant CH is presented as Construction 2. We note that compared to Ateniese et al. [4] we cannot use true-simulation extractable NIZKs (tSE-NIZKs) [25] and need SSE NIZKs.

$\underline{\mathsf{CHPG}(1^\lambda)}$: Fix a public-key encryption scheme Ω and a compatible NIZK proof system for language L in (2). Return $\mathsf{pp}_{\mathsf{ch}} = (\mathsf{pp}_\Omega, \mathsf{crs}_\Pi)$, where

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

$\underline{\mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})}$: Return $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) = (\mathsf{sk}_\Omega, (\mathsf{pp}_{\mathsf{ch}}, \mathsf{pk}_\Omega))$, where

$$(\mathsf{sk}_\Omega, \mathsf{pk}_\Omega) \leftarrow_r \mathsf{KG}_\Omega(\mathsf{pp}_\Omega).$$

$\underline{\mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)}$: Parse $\mathsf{pk}_{\mathsf{ch}}$ as $((\mathsf{pp}_\Omega, \mathsf{crs}_\Pi), \mathsf{pk}_\Omega)$, and return $(h, r) = (c, \pi)$, where

$$(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, m), \text{ and } \pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, h, m), (\bot, \xi)).$$

$\underline{\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h)}$: Parse $\mathsf{pk}_{\mathsf{ch}}$ as $((\mathsf{pp}_\Omega, \mathsf{crs}_\Pi), \mathsf{pk}_\Omega)$, and r as π. Return 1, if the following holds, and 0 otherwise:

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

$\underline{\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$: Parse $\mathsf{sk}_{\mathsf{ch}}$ as sk_Ω. Verify whether $m' \in \mathcal{M}$, and $\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h) = 1$. Return $\bot$, if not. Otherwise, return $r' = \pi'$, where

$$\pi' \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, h, m'), (\mathsf{sk}_\Omega, \bot)).$$

Construction 2: Our Construction of a Fully Collision-Resistant CH

4.2 Security

Subsequently, we prove the security of our CH in Construction 2.

Theorem 9. *If Ω is correct and Π is complete, then CH in Construction 2 is correct.*

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

Theorem 10. *If Ω is mcIND-CPA secure, and Π is zero-knowledge, then CH in Construction 2 is indistinguishable.*

In the proof, we use 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 indistinguishability, we use a sequence of games:

Game 0: The original indistinguishability game.

Game 1: As Game 0, but we modify the algorithms CHPG, CHash, and $\mathsf{CHAdapt}$ used inside the game:

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

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

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

$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, h, m), (\bot, \xi)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, h, m))}$$

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

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

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:

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

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

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

$$\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, h, m)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pk}_\Omega, h, m), (\bot, \xi))}.$$

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

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

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 further modify the CHash algorithm as follows:

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

$$(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, m) \rightsquigarrow \boxed{(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, 0)}.$$

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 multi-challenge IND-CPA challenger to interpolate between two consecutive games.

$\underline{\mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})''}$: Return $(\bot, \mathsf{pk}_{\mathsf{ch}}) = (\bot, (\mathsf{pp}_{\mathsf{ch}}, \mathsf{pk}_\Omega))$, where

$$(\mathsf{sk}_\Omega, \mathsf{pk}_\Omega) \leftarrow_r \mathsf{KG}_\Omega(\mathsf{pp}_\Omega) \rightsquigarrow \boxed{\mathsf{pk}_\Omega \leftarrow_r \mathcal{C}^{\text{mc-cpa}}}.$$

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

$$(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, 0) \rightsquigarrow \boxed{(c; \bot) \leftarrow_r \mathcal{C}^{\text{mc-cpa}}.\mathsf{Enc}'(m, 0)}.$$

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

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

Theorem 11. *If Ω is perfectly correct and mcIND-CPA secure, and Π is zero-knowledge as well as simulation-sound extractable, then* CH *in Construction 2 is fully collision-resistant.*

Proof. To prove full collision-resistance, we use a sequence of games.

Game 0: The original full collision-resistance game.

Game 1: As Game 0, but we modify the CHPG and the CHAdapt algorithm as follows:

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

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

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

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

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.

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

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

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

$$\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, h, m')) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pk}_\Omega, h, m'), \mathsf{sk}_\Omega)}.$$

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 further modify the CHPG algorithm 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 CHCheck algorithm, where h was never submitted to the collision-finding oracle before.

Transition - Game 2 → Game 3: This change is conceptual, i.e., $|\Pr[S_2]-\Pr[S_3]| = 0$.

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 $(\mathsf{sk}', \xi') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pk}_\Omega, h^*, m'^*), r'^*)$ 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:

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

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

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

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

In the end we output $((\mathsf{pk}_\Omega, h^*, m'^*), r'^*)$ 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 $(\mathsf{sk}'', \xi'') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pk}_\Omega, h^*, m^*), r^*)$ or $(\mathsf{sk}'', \xi'') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pk}_\Omega, h^*, m), r)$, 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 mcIND-CPA: We are now ready to construct an adversary $\mathcal{B}$ which breaks the mcIND-CPA security of the underlying Ω. Our adversary $\mathcal{B}$ proceeds as follows. It receives pp_Ω and pk_Ω from its own 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 (sk, ξ) as well as (sk'', ξ'') where one attests membership of $(\mathsf{pk}_\Omega, h^*, m'^*)$ in L and one attests membership of $(\mathsf{pk}_\Omega, h^*, m'')$ for some $m'' \neq m'^*$ in L. By the perfect correctness of the encryption scheme, we know that at most one of them can be consistent with the ciphertext contained in h^*, which implies that either sk or sk'' will be the key for the underlying encryption scheme (which of them we figure out by using KVf_Ω). With knowledge of the key, $\mathcal{B}$ trivially breaks the mcIND-CPA security of the underlying Ω by randomly sending two distinct messages to its

own challenger (for encryption), simply decrypting the returned ciphertext, and answering with the correct bit. We have that $\Pr[S_5] \leq \nu_{\mathsf{mc\text{-}cpa}}(\lambda)$. This concludes the proof. □

4.3 Concrete Instantiation

A suitable instantiation for Ω is ElGamal [29]. The algorithm KVf_Ω is simply checking whether $g^{\mathsf{sk}_\Omega} = g^x = \mathsf{pk}_\Omega$. Note that for Π we only need to extract a bounded number of times (i.e., twice). To this end one may use Fiat-Shamir transformed Σ-protocols for DLOG relations in the random-oracle model [28] when additionally applying the compiler by Faust et al. [27]. In particular, Faust et al. show that such proofs are simulation-sound extractable when additionally including the statement x upon hashing in the challenge computation and if the Σ-protocol provides a property called quasi-unique responses. The latter is straightforward for the statements which need to be proven in our context. See, e.g., [23], for a detailed discussion of this transformation.

For the sake of completeness and to demonstrate how efficiently our approach can be instantiated, we provide this concrete instantiation as Construction 3. Therefore, let $(\mathbb{G}, g, q) \leftarrow_r \mathsf{GGen}(1^\lambda)$ be an instance generator which returns a prime-order, and multiplicatively written, group $\mathbb{G}$ where the DDH problem is hard, along with a generator g such that $\langle g \rangle = \mathbb{G}$. Note that an SSE NIZK for the required L in (3) can easily be obtained as an *equality* proof of two discrete logarithms together with an *or* composition of a proof of a discrete logarithm [19] of Fiat-Shamir transformed Σ-protocols discussed above.

$$L \coloneqq \{(y, h, m) \mid \exists\ (x, \xi)\ :\ h = (g^\xi, m \cdot y^\xi)\ \vee\ y = g^x\}. \tag{3}$$

4.4 Comparison

Subsequently, in Table 1 we compare existing constructions of chameleon-hashes providing the W-CollRes, E-CollRes and S-CollRes notions with instantiations of our approach (in the random oracle and standard model) providing the stronger F-CollRes notion. Here E denotes an exponentiation in the respective algebraic structure, "?" denotes that it is unclear how efficient this can be realized due to requirement of an invertible onto mapping into the used group (cf. the discussion in [36]). SM and RO denote the standard and the random oracle model respectively. Furthermore, DDH, SXDH, PKoE, and OM-RSA denote the decisional Diffie-Hellman, the symmetric DDH, the power knowledge of exponent [34], and the one-more RSA inversion [9] assumptions. We also stress that for constructions relying on SXDH, for typical instantiations of type-III bilinear groups, we have that $|\mathbb{G}_2| = 2(|\mathbb{G}_1| - 1) + 1$ (where $|\cdot|$ denotes the size of the representation of a group element). Regarding our construction in the standard model, e.g.,

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

$\underline{\mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})}$: Return $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) = (x, y)$, where $x \leftarrow_r \mathbb{Z}_q$ and $y \leftarrow g^x$.

$\underline{\mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)}$: Parse $\mathsf{pk}_{\mathsf{ch}}$ as y, choose $(\xi, k_1, e_2, s_2) \leftarrow_r \mathbb{Z}_q^4$, set $u_{1,1} \leftarrow g^{k_1}$, $u_{1,2} \leftarrow y^{k_1}$, $u_2 \leftarrow g^{s_2} \cdot y^{-e_2}$, $e \leftarrow H((y, h, m), (u_{1,1}, u_{1,2}, u_2))$ and $e_1 \leftarrow e - e_2 \bmod q$. Then compute $s_1 \leftarrow k_1 + e_1\xi \bmod q$ and finally, return $(h, r) = (c, \pi)$, where

$$c \leftarrow (c_1, c_2) = (g^\xi, m \cdot y^\xi) \text{ , and } \pi \leftarrow (e_1, e_2, s_1, s_2).$$

$\underline{\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h)}$: Parse $\mathsf{pk}_{\mathsf{ch}}$ as y and r as (e_1, e_2, s_1, s_2), and h as (c_1, c_2). Return 1 if the following holds, and 0 otherwise:

$$m \in \mathbb{G} \ \wedge \ e_1 + e_2 = H((y, h, m), (g^{s_1} \cdot c_1^{-e_1}, y^{s_1} \cdot (c_2/m)^{-e_1}, g^{s_2} \cdot y^{-e_2})).$$

$\underline{\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)}$: Parse $\mathsf{sk}_{\mathsf{ch}}$ as x, and h as (c_1, c_2). Verify whether $m' \in \mathbb{G}$, and $\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, r, h) = 1$. Return $\perp$ if not. Otherwise, choose $(k_2, e_1, s_1) \leftarrow_r \mathbb{Z}_q^3$, set $u_{1,1} \leftarrow g^{s_1} \cdot c_1^{-e_1}$, $u_{1,2} \leftarrow y^{s_1} \cdot (c_2/m')^{-e_1}$, $u_2 \leftarrow g^{k_2}$, $e \leftarrow H((y, h, m'), (u_{1,1}, u_{1,2}, u_2))$, 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

$$\pi' \leftarrow (e_1, e_2, s_1, s_2).$$

Construction 3: Concrete instantiation of a Fully Collision-Resistant CH

using SSE NIZKs based on Groth-Sahai NIZKs, one can use the compiler in [22] to efficiently achieve simulation-sound extractability. We, however, note that a naive instantiation of our template in the standard model would still require to include bit-wise proofs of the parts of the witness which are in $\mathbb{Z}_q$, which would, all in all, require a number of group elements in the order of $1k - 2k$ (a very rough estimate; thus we also omit the remaining costs which is indicated by "–" in Table 1). It seems that switching to a variant of ElGamal in the target group (and maybe some other tweaks) would help to work around the requirement of having bit-wise proofs. Optimizing this instantiation is not in the scope of this work and therefore we only give our rough estimates in the table. Finally, we note that we omit comparing our scheme given in Construction 1 as it is contrived and its sole purpose is to prove a separation result.

5 Application: Redactable Blockchains

While one of the major goals of blockchains is their immutability and in particular their use as an immutable append-only log, recently, starting with the work of Ateniense et al. [4], there has been an increasing interest in blockchains that allow some controlled after-the-fact modification of their content. This is motivated by illegal content that was shown to be included into the Bitcoin blockchain [38], which represents a significant challenge for law enforcement agencies [45], as well as legislations like the European General Data Protection Regulation (GDPR)

Scheme	CR	$\lvert h\rvert$	$\lvert h\rvert_{\text{bit}}$	$\lvert r\rvert$	$\lvert r\rvert_{\text{bit}}$	CHash	CHAdapt	Ass.	Model
[37]	W	$1\mathbb{G}$	256	$1\mathbb{Z}_q$	256	$2\text{E}_{\mathbb{G}}$	$0\text{E}_{\mathbb{G}}$	DLOG	SM
[4] (1)	E	$1\mathbb{G}$	256	$12\mathbb{G}+7\mathbb{Z}_q$	4876	$17\text{E}_{\mathbb{G}}$	?	DDH	ROM
[4] (2)	E	$1\mathbb{G}_1$	382	$6\mathbb{G}_1+13\mathbb{G}_2$	12211	$51\text{E}_{\mathbb{G}_1}$	?	SXDH	SM
[36] (1)	E	$1\mathbb{G}_1$	382	$9\mathbb{G}_1+4\mathbb{G}_2$	6490	$25\text{E}_{\mathbb{G}_1}$	$1\text{E}_{\mathbb{Z}_q}$	SXDH	SM
[36] (2)	E	$1\mathbb{G}_1$	382	$3\mathbb{G}_1$	1164	$6\text{E}_{\mathbb{G}_1}$	$1\text{E}_{\mathbb{Z}_q}$	PKoE	SM
[15]	S	$1\mathbb{Z}_N$	3072	$1\mathbb{Z}_N$	3072	$1\text{E}_{\mathbb{Z}_N}$	$1\text{E}_{\mathbb{Z}_N}$	OM-RSA	ROM
Ours	F	$2\mathbb{G}$	514	$4\mathbb{Z}_q$	1024	$6\text{E}_{\mathbb{G}}$	$5\text{E}_{\mathbb{G}}$	DDH	ROM
Ours	F	$2\mathbb{G}_1$	764	$\approx$ 1-2k $\mathbb{G}_{1/2}$	-	-	-	SXDH	SM

Table 1. Comparison of different chameleon-hash functions. $\lvert\cdot\rvert_{\text{bit}}$ refers to the bit size of the respective value which is currently believed to provide 128 bit security. We use 256bit elliptic curves for standard known order groups ($\lvert\mathbb{G}\rvert = 257, \lvert\mathbb{Z}_q\rvert = 256$), 3072bit RSA modulus for the RSA setting ($\lvert\mathbb{Z}_N\rvert = 3072$), and 381bit BLS12 curves for the SXDH setting ($\lvert\mathbb{G}_1\rvert = 382, \lvert\mathbb{G}_2\rvert = 763, \lvert\mathbb{Z}_q\rvert = 256$).

and the associated "right to be forgotten". Solutions to this problem may either be for the permissioned- or permissionless-blockchain setting and cryptographic in nature [4, 21, 41] or non-cryptographic, where in the latter case it is based on the consensus layer of the blockchain [24].

We are considering the former and focus on block-level rewriting (change entire blocks) of blockchains instead of transaction-level rewriting (change single transactions within a block) in a permissionless setting (such as Bitcoin), as this illustrates the problem with much wider implications. In the following we are using the notation used in [4], and describe a block as triple of the form $B = \langle s, x, \texttt{ctr}\rangle$, where $s \in \{0,1\}^\lambda$, $x \in \{0,1\}^*$ and $\texttt{ctr} \in \mathbb{N}$ and a block is valid if

$$\textbf{validblock}_q^D(B) := (H(\texttt{ctr}, G(s,x)) < D) \ \wedge \ (\texttt{ctr} < q) = 1.$$

Here, $H : \{0,1\}^* \to \{0,1\}^{2\lambda}$ and $G : \{0,1\}^* \to \{0,1\}^{2\lambda}$ are collision-resistant hash functions, and the parameters $D \in \mathbb{N}$ and $q \in \mathbb{N}$ are the difficulty level of the block and the maximum number of hash queries that a user is allowed to make in any given round of the protocol, respectively. The chaining of blocks is now done by requiring that when attaching a (valid) block $B' = \langle s', x', \texttt{ctr'}\rangle$ we have that $s' = H(\texttt{ctr}, G(s,x))$. Now to make blocks redactable, one changes the description of blocks to $B = \langle s, x, \texttt{ctr}, (h,r)\rangle$ where the new component is a chameleon-hash (h,r) and the validation predicate changes to

$$\begin{aligned}\textbf{validblock}_q^D(B) :=& (H(\texttt{ctr}, h) < D) \ \wedge \ \mathsf{CHCheck}(\mathsf{pk_{ch}}, (s,x), r, h) = 1 \ \wedge \\ & (\texttt{ctr} < q) = 1.\end{aligned}$$

Chaining is now done by requiring that when attaching a (valid) block $B' = \langle s', x', \texttt{ctr'}\rangle$ we have that $s' = H(\texttt{ctr}, h)$. Observe that now computing a collision in the chameleon-hash gives very much power as it basically allows to rewrite the entire history of the blockchain.

Ateniese et al. in [4] discuss different ways to control this power to actually compute collisions (i.e., run CHAdapt) where 1) either $\mathsf{sk_{ch}}$ may be available to some fully trusted single party only, or 2) $\mathsf{sk_{ch}}$ is generated using a multi-party

computation (MPC) protocol and CHAdapt is also performed in a distributed way by some set of parties. We will discuss the implications of different collision-resistance notions to this setting, which is independent of which of these two approaches is going to be used.

We recall that Ateniese et al. [4], who introduced this application, rely on E-CollRes and Derler et al. in more recent work in [21] rely on S-CollRes. Now, note that in such a permissionless setting as discussed above, where everybody is allowed to participate, it is reasonable to assume that an adversary sees the collisions computed for any blocks over some time in the system (as they will be broadcasted). Now let us discuss the single notions:

Weak Collision-Resistance (W-CollRes)**.** A chameleon-hash providing this notion of collision-resistance provides absolutely no guarantees, as after seeing a single collision all guarantees are lost. A prime example is the Pedersen CH due to Krawczyk and Rabin [37] (cf. Appendix B.1), where a single seen collision exposes the secret key $\mathsf{sk}_{\mathsf{ch}}$ to everybody. Clearly, this has significant consequences in the above scenario as then everybody can arbitrarily alter the blockchain.

Enhanced Collision-Resistance (E-CollRes)**.** Recall that an adversary when attacking some hash h^* must have never input h^* to CHAdapt$'$. Now, this means that if an adversary targets a specific hash and then happens to see a collision for this hash (for some reason), suddenly all guarantees are lost and arbitrary collisions could be computed. Note that our construction in Sect. 3.5 clearly demonstrates potential problems with CHs only satisfying this notion. This still represents a significant problem with this application.

Standard Collision-Resistance (S-CollRes)**.** Recall, that an adversary is only restricted to not query message m^* (which is associated to the computed collision h^*) was never queried to the collision-finding oracle. While this still might be problematic in the redactable blockchain setting, messages can very likely be made unique by perpending a large enough random tag/nonce (note that in this could easily be done in the block format of e.g., the Bitcoin block structure). So, this notion seems suitable if the aforementioned constrained may, under certain circumstances, be guaranteed to be met, but is far away from being ideal.

Full Collision-Resistance (F-CollRes)**.** We recall that, here, only the collision (h^*, m^*) was not generated by the collision-finding oracle, but there is no other restriction whatsoever. Consequently, this collision-resistance notion seems the "right" notion as no issues on higher levels need to be considered and very strong guarantees are already provided by the notion itself.

Acknowledgements. This work was supported by the EU's Horizon 2020 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

1. Abe, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Tagged one-time signatures: Tight security and optimal tag size. In: PKC. pp. 312–331 (2013)
2. Alsouri, S., Dagdelen, Ö., Katzenbeisser, S.: Group-based attestation: Enhancing privacy and management in remote attestation. In: Trust. pp. 63–77 (2010)
3. Ateniese, G., Chou, D.H., de Medeiros, B., Tsudik, G.: Sanitizable signatures. In: ESORICS. pp. 159–177 (2005)
4. Ateniese, G., Magri, B., Venturi, D., Andrade, E.R.: Redactable blockchain - or - rewriting history in bitcoin and friends. In: EuroS&P. pp. 111–126 (2017)
5. Ateniese, G., de Medeiros, B.: Identity-based chameleon hash and applications. In: FC. pp. 164–180 (2004)
6. Ateniese, G., de Medeiros, B.: On the key exposure problem in chameleon hashes. In: SCN. pp. 165–179 (2004)
7. Bao, F., Deng, R.H., Ding, X., Lai, J., Zhao, Y.: Hierarchical identity-based chameleon hash and its applications. In: ACNS. pp. 201–219 (2011)
8. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Eurocrypt. pp. 259–274 (2000)
9. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. J. Cryptology 16(3), 185–215 (2003)
10. Bellare, M., Ristov, T.: Hash functions from sigma protocols and improvements to VSH. In: Asiacrypt. pp. 125–142 (2008)
11. Bellare, M., Ristov, T.: A characterization of chameleon hash functions and new, efficient designs. J. Cryptology 27(4), 799–823 (2014)
12. Blazy, O., Kakvi, S.A., Kiltz, E., Pan, J.: Tightly-secure signatures from chameleon hash functions. In: PKC. pp. 256–279 (2015)
13. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. 37(2), 156–189 (1988)
14. Brzuska, C., Fischlin, M., Freudenreich, T., Lehmann, A., Page, M., Schelbert, J., Schröder, D., Volk, F.: Security of sanitizable signatures revisited. In: PKC. pp. 317–336 (2009)
15. Camenisch, J., Derler, D., Krenn, S., Pöhls, H.C., Samelin, K., Slamanig, D.: Chameleon-hashes with ephemeral trapdoors - and applications to invisible sanitizable signatures. In: PKC. pp. 152–182 (2017)
16. Chen, X., Zhang, F., Kim, K.: Chameleon hashing without key exposure. In: ISC. pp. 87–98 (2004)
17. Chen, X., Zhang, F., Susilo, W., Mu, Y.: Efficient generic on-line/off-line signatures without key exposure. In: ACNS. pp. 18–30 (2007)
18. Choi, J., Jung, S.: A handover authentication using credentials based on chameleon hashing. IEEE Communications Letters 14(1), 54–56 (2010)
19. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Crypto. pp. 174–187 (1994)
20. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Crypto. pp. 13–25 (1998)
21. Derler, D., Samelin, K., Slamanig, D., Striecks, C.: Fine-grained and controlled rewriting in blockchains: Chameleon-hashing gone attribute-based. In: NDSS (2019)
22. Derler, D., Slamanig, D.: Key-homomorphic signatures: definitions and applications to multiparty signatures and non-interactive zero-knowledge. Des. Codes Cryptogr. 87(6), 1373–1413 (2019)

23. Derler, D., Slamanig, D.: Highly-efficient fully-anonymous dynamic group signatures. In: AsiaCCS. pp. 551–565 (2018)
24. Deuber, D., Magri, B., Thyagarajan, S.A.K.: Redactable blockchain in the permissionless setting. In: IEEE S&P. pp. 124–138 (2019)
25. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. In: Asiacrypt. pp. 613–631 (2010)
26. Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. J. Cryptology 9(1), 35–67 (1996)
27. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the fiat-shamir transform. In: Indocrypt. pp. 60–79 (2012)
28. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Crypto. pp. 186–194 (1986)
29. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Crypto. pp. 10–18 (1984)
30. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Asiacrypt. pp. 444–459 (2006)
31. Groth, J.: Efficient fully structure-preserving signatures for large messages. In: Asiacrypt. pp. 239–259 (2015)
32. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Eurocrypt. pp. 415–432 (2008)
33. Guo, S., Zeng, D., Xiang, Y.: Chameleon hashing for secure and privacy-preserving vehicular communications. IEEE Trans. Parallel Distrib. Syst. 25(11) (2014)
34. Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Crypto. pp. 408–423 (1998)
35. Hohenberger, S., Waters, B.: Short and stateless signatures from the RSA assumption. In: Crypto. pp. 654–670 (2009)
36. Khalili, M., Dakhilalian, M., Susilo, W.: Efficient chameleon hash functions in the enhanced collision resistant model. Inf. Sci. 510, 155–164 (2020)
37. Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS. pp. 143–154 (2000)
38. Matzutt, R., Hiller, J., Henze, M., Ziegeldorf, J.H., Müllmann, D., Hohlfeld, O., Wehrle, K.: A quantitative analysis of the impact of arbitrary blockchain content on bitcoin. In: FC. pp. 420–438 (2018)
39. Mohassel, P.: One-time signatures and chameleon hash functions. In: SAC. pp. 302–319 (2010)
40. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Crypto. pp. 129–140 (1991)
41. Samelin, K., Slamanig, D.: Policy-based sanitizable signatures. In: CT-RSA. pp. 538–563 (2020)
42. Shamir, A., Tauman, Y.: Improved online/offline signature schemes. In: Crypto. pp. 355–367 (2001)
43. Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J.: Universal designated-verifier signatures. In: Asiacrypt. pp. 523–542 (2003)
44. Steinfeld, R., Wang, H., Pieprzyk, J.: Efficient extension of standard schnorr/rsa signatures into universal designated-verifier signatures. In: PKC. pp. 86–100 (2004)
45. Tziakouris, G.: Cryptocurrencies - A forensic challenge or opportunity for law enforcement? an INTERPOL perspective. IEEE S&P 16(4) (2018)
46. Zhang, R.: Tweaking TBE/IBE to PKE transforms with chameleon hash functions. In: ACNS. pp. 323–339 (2007)

A Additional Property of Chameleon-Hashes

We now present additional indistinguishability notions of CHs: strong indistinguishability and full indistinguishability, respectively. They are not directly in scope of this paper, but may be useful in other contexts. We present the respective security games in Figure 9. We highlight differences by using boxes, and missing lines using boxes.

Strong Indistinguishability (S-Ind). Strong indistinguishability requires that a randomness r does not reveal whether it was generated using CHash or CHAdapt, even if the adversary $\mathcal{A}$ knows all secret keys.

Full Indistinguishability (F-Ind). Full indistinguishability requires that a randomness r does not reveal whether it was generated using CHash or CHAdapt, even if the adversary $\mathcal{A}$ controls all values, but the public parameters.[9]

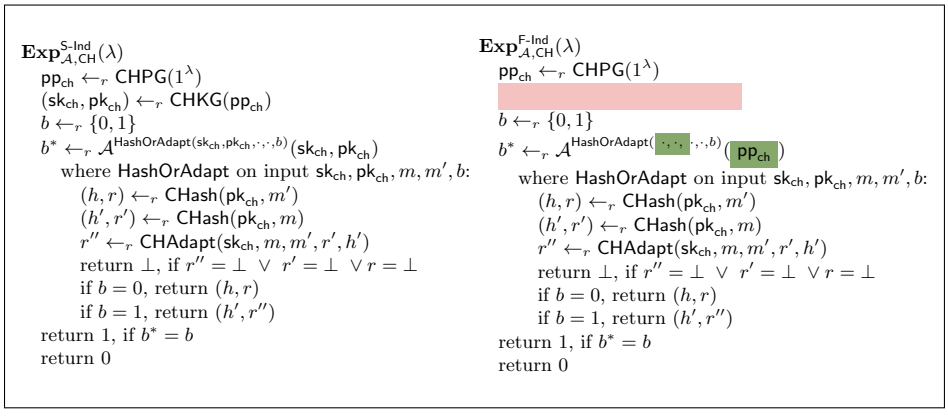

Fig. 9. The $\mathbf{Exp}^{\mathsf{X\text{-}Ind}}_{\mathcal{A},\mathsf{CH}}$ experiment with $\mathsf{X} \in \{\mathsf{S}, \mathsf{F}\}$.

Definition 16 (X Indistinguishability). *A chameleon-hash* CH *offers* X *indistinguishability with* $\mathsf{X} \in \{\mathsf{S}, \mathsf{F}\}$*, if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that*

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

The corresponding experiments are depicted in Figure 9.

Relations Between Indistinguishability Notions. We formally prove that full indistinguishability is strictly stronger than strong indistinguishability, which, in turn, is strictly stronger than indistinguishability (cf. Figure 10 for an overview).

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

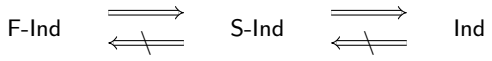

Fig. 10. Relations between CH indistinguishability properties

Theorem 12. *Full Indistinguishability is strictly stronger than Strong Indistinguishability.*

Proof. We first prove that full indistinguishability implies strong indistinguishability and then give a counterexample showing that the other direction of the implication does not hold.

F-Ind $\implies$ S-Ind**:** Assume $\mathcal{A}$ to be an adversary who wins the full indistinguishability game with some probability (non-negligibly) larger than $1/2$. Now, we construct an adversary $\mathcal{B}$ which wins the strong indistinguishability game with the same probability. In particular, $\mathcal{B}$ proceeds as follows. It receives $\mathsf{pp}_{\mathsf{ch}}$ from its own challenger, generates $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}})$ honestly, and uses $\mathsf{pp}_{\mathsf{ch}}$, $\mathsf{sk}_{\mathsf{ch}}$, and $\mathsf{pk}_{\mathsf{ch}}$ to initialize $\mathcal{A}$. All queries to the collision-finding oracle are answered by querying $\mathcal{B}$'s own oracle with the honestly generated keys. Whenever $\mathcal{A}$ outputs a bit a, $\mathcal{B}$ returns that bit to its own challenger. As the simulation is perfect, $\mathcal{B}$'s winning probability equals the one of $\mathcal{A}$.

S-Ind $\not\implies$ F-Ind**:** Let $\mathsf{CH} \coloneqq (\mathsf{CHPG}, \mathsf{CHKG}, \mathsf{CHash}, \mathsf{CHCheck}, \mathsf{CHAdapt})$ be a fully indistinguishable chameleon-hash. We define a chameleon-hash $\mathsf{CH}' \coloneqq (\mathsf{CHPG}', \mathsf{CHKG}', \mathsf{CHash}', \mathsf{CHCheck}', \mathsf{CHAdapt}')$, which internally uses CH, as follows: $\mathsf{CHPG}'(1^\lambda) \coloneqq \mathsf{CHPG}(1^\lambda)$, $\mathsf{CHKG}'(\mathsf{pp}_{\mathsf{ch}}) \coloneqq (\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}})$ **where** $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}'_{\mathsf{ch}}) \leftarrow_r \mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})$, and $\mathsf{pk}_{\mathsf{ch}} = (\mathsf{pk}'_{\mathsf{ch}}, 0)$, $\mathsf{CHash}'(\mathsf{pk}_{\mathsf{ch}}, m) \coloneqq (h, (r, 0))$, where $(h, r) \leftarrow_r \mathsf{CHash}(\mathsf{pk}'_{\mathsf{ch}}, m)$, $\mathsf{CHCheck}'(\mathsf{pk}_{\mathsf{ch}}, m, r', h) \coloneqq \mathsf{CHCheck}(\mathsf{pk}'_{\mathsf{ch}}, m, r, h)$, where $r' = (r, \cdot)$, and $\mathsf{pk}_{\mathsf{ch}} = (\mathsf{pk}'_{\mathsf{ch}}, \cdot)$, $\mathsf{CHAdapt}'(\mathsf{sk}_{\mathsf{ch}}, m, m', r', h) \coloneqq$ **if** $\mathsf{pk}_{\mathsf{ch}} = (\mathsf{pk}'_{\mathsf{ch}}, 0)$ **then** $(r'', 0)$ where $r'' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)$ and $r' = (r, \cdot)$ **else** $(r'', 1)$ where $r'' \leftarrow_r \mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', r, h)$ and $r' = (r, \cdot)$. Clearly, if all parties generate their keys honestly (and thus a 0 is appended to the public key), the last bit appended to the randomness is always switched to 0 after adaption, is never appended at hashing, and is independent of the message hashed. If, however, the adversary can generate the keys, it can generate a $\mathsf{pk}_{\mathsf{ch}}$ with an appended 1, thus making adaption append a 1, while hashing still appends a 0. This breaks full indistinguishability. □

Theorem 13. *Strong Indistinguishability is strictly stronger than Indistinguishability.*

Proof. We first prove that full indistinguishability implies indistinguishability and then give a counterexample showing that the other direction of the implication does not hold.

S-Ind $\implies$ Ind**:** Assume $\mathcal{A}$ to be an adversary who wins the indistinguishability game with non-negligible probability. Using $\mathcal{A}$ we construct an adversary $\mathcal{B}$ which wins the strong indistinguishability game with the same probability: $\mathcal{B}$

receives $\mathsf{pp}_{\mathsf{ch}}$ from its own challenger, receiving $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}})$, and uses $\mathsf{pp}_{\mathsf{ch}}$ and $\mathsf{pk}_{\mathsf{ch}}$ to initialize $\mathcal{A}$. All queries to the collision-finding oracle are answered by querying $\mathcal{B}$'s own oracle with the received keys. Whenever $\mathcal{A}$ outputs a bit a, $\mathcal{B}$ returns that bit to its own challenger. As the simulation is perfect, $\mathcal{B}$'s winning probability equals the one of $\mathcal{A}$.

$\mathsf{Ind} \not\Rightarrow \mathsf{S\text{-}Ind}$**:** Our scheme given in Construction 2 provides a suitable counterexample. In particular, due to the used encryption, knowledge of the secret key allows extracting the original message. In more detail, to show that this construction is not strongly indistinguishable, consider the following strategy. The key pair $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}})$ is generated by the challenger, but (according to the game) known to the adversary. Obtain a challenge tuple $(h, r) \leftarrow_r \mathsf{HashOrAdapt}(\mathsf{pk}_{\mathsf{ch}}, \mathsf{sk}_{\mathsf{ch}}, m, m')$, where $m \neq m'$ are random messages. Then, let $m'' \leftarrow \mathsf{Dec}(\mathsf{sk}_{\mathsf{ch}}, h)$. If $m = m''$, return 0. Otherwise, return 1. Clearly, this strategy always allows learning the challenger's bit. □

B Existing Constructions of Chameleon-Hashes

For the sake of completeness we provide now examples of chameleon-hashes providing the W-CollRes and S-CollRes notions, respectively.

B.1 Instantiation of a Weakly Collision-Resistant CH

We recall the initial CH construction by Krawczyk and Rabin [37] in Construction 4. Note that a collision-resistant hash-function is applied to the message

$\underline{\mathsf{CHPG}(1^\lambda)}$: Outputs the public parameters $(\mathbb{G}, g, q, H)$, where $(\mathbb{G}, g, q) \leftarrow \mathsf{GGen}(1^\lambda)$ and $H : \{0,1\}^* \to \mathbb{Z}_q^*$ is a hash function chosen uniformly at random from a family of collision resistant hash functions.

$\underline{\mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})}$: Parse $\mathsf{pp}_{\mathsf{ch}}$ as $(\mathbb{G}, g, q, H)$ and return $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) \leftarrow (x, g^x)$, where

$$x \leftarrow_r \mathbb{Z}_q^*.$$

$\underline{\mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)}$: Return (h, r), where

$$r \leftarrow_r \mathbb{Z}_q^*, \text{ and } h \leftarrow g^{H(m)}\mathsf{pk}_{\mathsf{ch}}^r.$$

$\underline{\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, h, r)}$: Return 1 if the following holds, and 0 otherwise:

$$h = g^{H(m)}\mathsf{pk}_{\mathsf{ch}}^r.$$

$\underline{\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', h, r)}$: Output $\perp$, if $\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, h, r) \neq 1$. Otherwise return r', where

$$r' \leftarrow \tfrac{H(m)+xr-H(m')}{x}.$$

Construction 4: DL-based chameleon-hash

prior to chameleon-hashing to extend the domain, which is a standard technique. Seeing a collision (if not stemming from the collision-resistant hash-function) allows to extract the $\mathsf{sk}_{\mathsf{ch}}$ by computing $x \leftarrow {}^{(H(m)-H(m'))}/_{(r'-r)} \bmod q$.

B.2 Instantiation of a Standard Collision-Resistant CH

We recall a construction from [15] in Construction 5. Before we do so, we recall some background on the setup the scheme requires: Let $(N, p, q, e, d) \leftarrow_r \mathsf{RSAKG}(1^\lambda)$ be an instance generator which returns an RSA modulus $N = pq$, where p and q are distinct primes, $e > 1$ is an integer co-prime to $\varphi(n)$, and $de \equiv 1 \bmod \varphi(n)$. The scheme requires that RSAKG always outputs moduli of the same bit-length, based on λ, and that the one-more RSA assumption holds [9].

$\underline{\mathsf{CHPG}(1^\lambda)}$: Output the public parameters $\mathsf{pp}_{\mathsf{ch}} \leftarrow (1^\lambda, e)$, where e is prime and $e > N'$ with $N' = \max_r\{N \in \mathbb{N} : (N, \cdot, \cdot, \cdot, \cdot) \leftarrow_r \mathsf{RSAKG}(1^\lambda; r)\}$.
$\underline{\mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})}$: Run $(N, p, q, \cdot, \cdot) \leftarrow_r \mathsf{RSAKG}(1^\lambda)$, choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_N^*$ (modeled as a random-oracle), compute d s.t. $ed \equiv 1 \mod \varphi(N)$, set $\mathsf{sk}_{\mathsf{ch}} \leftarrow_r d$, $\mathsf{pk}_{\mathsf{ch}} \leftarrow_r (N, H)$, and return $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}})$.
$\underline{\mathsf{CHash}(\mathsf{pk}_{\mathsf{ch}}, m)}$: Parse $\mathsf{pk}_{\mathsf{ch}} = (N, H)$ and a message m, choose $r \leftarrow_r \mathbb{Z}_N^*$, compute $h \leftarrow H(m)r^e \mod N$, and output (h, r).
$\underline{\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, h, r)}$: Parse $\mathsf{pk}_{\mathsf{ch}} = (N, H)$, compute $h' \leftarrow H(m)r^e \mod N$, and output 1 if $h' = h$ and 0 otherwise.
$\underline{\mathsf{CHAdapt}(\mathsf{sk}_{\mathsf{ch}}, m, m', h, r)}$: Output $\perp$, if $\mathsf{CHCheck}(\mathsf{pk}_{\mathsf{ch}}, m, h, r) \neq 1$. Otherwise, let $x \leftarrow H(m)$, $x' \leftarrow H(m')$, $y \leftarrow xr^e \mod N$ and return $r' \leftarrow (y(x'^{-1}))^d \mod N$.

Construction 5: RSA-based Chameleon-Hash

C Proof of Our Enhanced Collision-Resistant CH

Subsequently, we use frameboxes and $\rightsquigarrow$ to highlight the changes we make in the algorithms throughout a sequence of games (and we only show the changes).

Theorem 14. *If Ω, Σ, and Π are correct, then Construction 1 is correct.*

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

Theorem 15. *If Ω is mcIND-CPA secure and Π is zero-knowledge, then Construction 1 is indistinguishable.*

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

Game 0: The original indistinguishability game.

Game 1: As Game 0, but we modify the algorithms CHPG, CHash, and $\mathsf{CHAdapt}$ used within the game as follows:

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

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

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

$$\pi \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m), (\bot, \xi)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m))}$$

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

$$\pi' \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m'), (\sigma, \bot)) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m'))}.$$

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 the following changes:

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

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

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

$$\pi \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m)) \rightsquigarrow \boxed{\pi \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m), (\bot, \xi))}.$$

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

$$\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m')) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m'), (\sigma, \bot))}.$$

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 further modify the CHash algorithm as follows:

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

$$(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, m) \rightsquigarrow \boxed{(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_\Omega, 0)}.$$

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 mcIND-CPA challenger to interpolate between two consecutive games:

$\underline{\mathsf{CHKG}(\mathsf{pp}_{\mathsf{ch}})'}$: Return $(\mathsf{sk}_{\mathsf{ch}}, \mathsf{pk}_{\mathsf{ch}}) = ((\perp, \mathsf{sk}_{\Sigma}), (\mathsf{pp}_{\mathsf{ch}}, \mathsf{pk}_{\Omega}, \mathsf{pk}_{\Sigma}, \sigma_0))$, where

$$(\mathsf{sk}_{\Omega}, \mathsf{pk}_{\Omega}) \leftarrow_r \mathsf{KG}_{\Omega}(\mathsf{pp}_{\Omega}) \rightsquigarrow \boxed{\mathsf{pk}_{\Omega} \leftarrow_r \mathcal{C}^{\text{mc-cpa}}},$$

$$(\mathsf{sk}_{\Sigma}, \mathsf{pk}_{\Sigma}) \leftarrow_r \mathsf{KG}_{\Sigma}(\mathsf{pp}_{\Sigma}), \text{ and } \sigma_0 \leftarrow_r \mathsf{Sgn}_{\Sigma}(\mathsf{sk}_{\Sigma}, 0).$$

0 is considered some special invalid hash value for CH.

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

$$(c; \xi) \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}, 0) \rightsquigarrow \boxed{(c; \perp) \leftarrow_r \mathcal{C}^{\text{mc-cpa}}.\mathsf{LoR}(m, 0)}.$$

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

Game 3_i ($1 \leq i \leq q$): As Game 3_{i-1} (resp. Game 2 if $i = 0$) but we modify the $\mathsf{HashOrAdapt}$ as follows. We let q be an upper bound on the queries to the $\mathsf{HashOrAdapt}$ oracle. Up to query number i, we do the following:

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

$$c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', \sigma_0) \rightsquigarrow \boxed{c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', 0)}.$$

and in $\mathsf{CHAdapt}$

$$c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', \sigma) \rightsquigarrow \boxed{c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', 0)}.$$

For every query after query i we simulate $\mathsf{HashOrAdapt}$ as in Game 2.

Transition - Game $3_i \rightarrow$ Game 3_{i+1} (resp. Game $2 \rightarrow 3_1$): We bound the probability for an adversary to distinguish between two consecutive games by introducing a hybrid game which interpolates between to subsequent games. Then, up to query number $i - 1$, we do the following:

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

$$c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', \sigma_0) \rightsquigarrow \boxed{c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', 0)}.$$

and in $\mathsf{CHAdapt}$

$$c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', \sigma) \rightsquigarrow \boxed{c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', 0)}.$$

In query number i we do the following:

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

$$(\mathsf{sk}_{\Omega}{}', \mathsf{pk}_{\Omega}{}') \leftarrow_r \mathsf{KG}_{\Omega}(\mathsf{pp}_{\Omega}) \rightsquigarrow \boxed{(\perp, \mathsf{pk}_{\Omega}{}') \leftarrow_r \mathcal{C}^{\text{mc-cpa}}}.$$

In CHash

$$c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', 0) \rightsquigarrow \boxed{c' \leftarrow_r \mathcal{C}^{\text{mc-cpa}}.\mathsf{Enc}'(\sigma_0, 0)}.$$

and in $\mathsf{CHAdapt}$

$$c' \leftarrow_r \mathsf{Enc}(\mathsf{pk}_{\Omega}{}', 0) \rightsquigarrow \boxed{c' \leftarrow_r \mathcal{C}^{\text{mc-cpa}}.\mathsf{Enc}'(\sigma, 0)}.$$

For every query after query i we simulate $\mathsf{HashOrAdapt}$ as in Game 2. Now, depending on the challenger's bit, we either simulate Game i or Game $i+1$. Thus we have that $|\Pr[S_2] - \Pr[S_{3_q}]| \leq q \cdot \nu_{\mathsf{mc\text{-}cpa}}(\lambda)$, where q is the overall number of queries to $\mathsf{HashOrAdapt}$.[10]

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

Theorem 16. *If Ω is perfectly correct, Σ is unforgeable, and Π is zero-knowledge as well as simulation-sound extractable, then Construction 1 provides enhanced collision-resistance.*

Proof. To prove enhanced collision-resistance, we use a sequence of games.

Game 0: The original enhanced collision-resistance game.
Game 1: As Game 0, but we modify the CHPG and the $\mathsf{CHAdapt}$ as follows:

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

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

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

$$\pi' \leftarrow_r \mathsf{Prf}_\Pi(\mathsf{crs}_\Pi, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m'), (\sigma, \perp)) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m'))}.$$

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.

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

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

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

$$\pi' \leftarrow_r \mathsf{SIM}_2(\mathsf{crs}_\Pi, \tau, (\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m')) \rightsquigarrow \boxed{\pi' \leftarrow_r \mathcal{C}^{\mathsf{zk}}.P_b((\mathsf{pk}_\Omega, \mathsf{pk}_\Sigma, h, m'), \sigma)}.$$

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 further modify the CHPG algorithm 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, $|\Pr[S_1] - \Pr[S_2]| = 0$.

[10] Note, if unrolled, using the bounds of Bellare et al. [8], $|\Pr[S_2] - \Pr[S_{3_q}]| \leq 2q \cdot \nu_{\mathsf{cpa}}(\lambda)$ follows.

Game 3: As Game 2, but we keep a list $\mathcal{Q}$ of all hashes h previously submitted to the collision-finding oracle which are accepted by the $\mathsf{CHCheck}$ algorithm.

Transition - Game 2 → Game 3: This change is conceptual and thus we have $|\Pr[S_2] - \Pr[S_3]| = 0$.

Game 4: As Game 3, but for every valid collision $(m^*, r^*, m'^*, r'^*, h^*)$ output by the adversary we observe that either (h^*, m^*, r^*) or (h^*, m'^*, r'^*) must be a "fresh" collision, i.e., $h^* \notin \mathcal{Q}$. We assume, without loss of generality, that (m'^*, r'^*) is the "fresh" collision. We run $(\mathsf{sk}', \sigma') \leftarrow_r \mathcal{E}_2(\mathsf{crs}_\Pi, \zeta, (\mathsf{pk}_\Omega, h^*, m'^*), r'^*)$ 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:

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

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

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

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

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

Reduction to eUNF-CMA: We are now ready to construct an adversary $\mathcal{B}$ which breaks the unforgeability of the underlying Σ. Our adversary $\mathcal{B}$ proceeds as follows. It receives pp_Σ and pk_Σ from its own challenger. To generate σ_0, $\mathcal{B}$ simply queries its signature oracle to obtain it on the message 0. It embeds them straightforwardly inside $\mathsf{pp}_{\mathsf{ch}}$ and $\mathsf{pk}_{\mathsf{ch}}$ to initialize $\mathcal{A}$. For adaption, a new signature σ' must be generated and encrypted. Those signatures are also obtained by querying the signature oracle. Now we know that we have extracted two witnesses (sk, σ) as well as $(\mathsf{sk}'', \sigma'')$ where one attests membership of $(\mathsf{pk}_\Sigma, h^*, m'^*)$ in L, and one attests membership of $(\mathsf{pk}_\Sigma, h^*, m'')$ for some fresh h^* in L. By the perfect correctness of the signature scheme, we know that at most one of them must be signature for h^*. However, as the signature was never queried, (h^*, σ) (or (h^*, σ'') resp.) must be a validating signature, breaking the unforgeability of the used Σ. Now, we have that $\Pr[S_4] \leq \nu_{\mathsf{eunf\text{-}cma}}(\lambda)$. This concludes the proof. □