Tightly Secure Chameleon Hash Functions in the Multi-User Setting and Their Applications*

Xiangyu Liu[1,2], Shengli Liu[1,2,3(✉)], and Dawu Gu[1]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
{xiangyu_liu, slliu, dwgu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Westone Cryptologic Research Center, Beijing 100070, China

Abstract. We define the security notion of (strong) collision resistance for chameleon hash functions in the multi-user setting ((S-)MU-CR security). We also present three constructions, CHF_{dl}, CHF_{rsa} and CHF_{fac}, and prove their tight S-MU-CR security based on the discrete logarithm, RSA and factoring assumptions, respectively. In applications, our tightly S-MU-CR secure chameleon hash functions help us to lift a signature scheme from (weak) unforgeability to strong unforgeability in the multi-user setting, and the security reduction is tightness preserving. Furthermore, they can also be used to construct tightly secure online/offline signatures, chameleon signatures and proxy signatures, etc., in the multi-user setting.

Keywords: Chameleon hash functions · Tight security · Multi-user setting · Signatures.

1 Introduction

Chameleon hash function (CHF) has been studied for decades since its introduction by Krawczyk and Rabin in [22]. Informally, chameleon hash function is a special hash function indexed by a hash key, which is associated with a trapdoor. On the one hand, it has the property of collision resistance, i.e., it is hard to find a collision given the hash key only. On the other hand, one can easily find collisions with the help of the trapdoor. Over the years, various constructions of CHF were proposed [27, 13, 7], and they found wide applications in signatures (SIG), including online/offline signatures [27], chameleon signatures [22], proxy signatures [24, 12], etc.

Tight Security. Generally, the collision resistance of CHF is proved by security reduction. That is, once an adversary finds a collision for CHF with probability ϵ, then another algorithm can be built to make use of the collision to solve some well-known hard problem with success probability ϵ/L. The parameter L is called the security loss factor. If L is a constant, the security reduction is tight. And if L is a polynomial of security parameter λ, the security reduction is loose. With a loose security reduction, the deployments of CHF (and other primitives) have to be equipped with a larger security parameter to compensate the loss factor L. This yields larger elements and slower computations. For instance, if $L \approx 2^{30}$, there will be a great efficiency loss.

* This is the extended version of the short paper published in ACISP 2020. Compared to the proceedings version, this version offers a formal proof of the extended GBSW transform, as well as three further applications of tightly MU-CR secure CHFs in online/offline signatures, chameleon signatures and proxy signatures.

Most constructions of CHF consider single user setting only. In the era of IoT, cryptographic primitives are deployed in systems of multi-users. Hence, it is important for us to consider tight security of CHF in the multi-user setting. With hybrid argument, collision resistance of CHF in single user setting implies collision resistance in the multi-user setting, but with a security loss factor $L = \mu$, where μ is the number of users. In consideration of wide applications of CHF, it is desirable for us to exploit tight collision resistance for CHF.

Related Works. In [22], Krawczyk and Rabin gave two constructions of chameleon hash functions. One is a generic construction from "claw-free" trapdoor permutations [14], and is implemented based on the factoring assumption. The other is based on the discrete logarithm (DL) assumption. Later, numerous constructions of chameleon hash functions are proposed in [2, 3, 27, 13, 19, 7], to name a few.

In [7], Bellare and Ristov proved that chameleon hash functions and Sigma protocols are equivalent, where the strong collision resistance property of chameleon hash functions corresponds to the strong special soundness of Sigma protocols. Due to this equivalence, many new chameleon hash functions CHF_{fs}, CHF_{ms}, CHF_{oka}, CHF_{hs} are obtained from well-studied Sigma protocols [16, 25, 26, 6]. Meanwhile, some variants of chameleon hash functions came up in needs of different applications, like identity-based CHF [2], key-exposure free CHF [3, 21], CHF with ephemeral trapdoors [11], multiple-collision CHF [19], etc.

Chameleon hash functions have found numerous applications in different types of signatures. The first application of CHF is chameleon signatures [22], which provide non-transferability. In [27], Shamir and Tauman defined the "hash-sign-switch" paradigm and gave a generic construction from (traditional) signature to online/offline signature with the help of CHF. Consequently many proxy signatures with variant security requirements are constructed based on CHF [24, 12]. Meanwhile, CHF can also be used to strengthen a (weakly) unforgeable signature to a strong unforgeable one [10, 28].

Most of these constructions consider single user setting only. Though they also work in the multi-user setting, but the price is a great security loss factor μ. As far as we know, there is no research considering tight security of chameleon hash functions in the multi-user setting, and that is exactly the focus of this paper.

Tight (Strong) Multi-User Collision Resistance of CHF. We define the security notion of (strong) multi-user collision resistance ((S-)MU-CR) for chameleon hash functions. In the multi-user setting, each user has its own hash key/trapdoor pair, and each hash key determines a specific chameleon hash function. Informally, (S-)MU-CR security means that after seeing all the hash keys, the adversary cannot find a collision under a specific hash key of its choice.

Over the years, there are lots of proposals of chameleon hash functions, which are tightly secure in single user setting. For example, the chameleon hash function CHF_{claw} from the claw-free permutations [22], CHF_{st} from the factoring assumption by Shamir and Tauman [27], $\mathsf{CHF}_{rsa\text{-}n}$ from the RSA$[n, n]$ assumption [3], CHF_{vsh} from the very smooth hash [13], CHF_{ms} from the Micali-Shamir protocol [7], etc. We believe that it is hard for these CHFs to achieve tight (S-)MU-CR security. Let us take CHF_{st} as an example. Each user has trapdoor (p_i, q_i) and hash key $N_i = p_i q_i$. In the security

reduction, the factoring problem instance N is embedded into a specific $N_j := N$. However, the adversary chooses N_j as its target with probability $1/\mu$. As a result, the security loss factor is at least μ.

Nevertheless, we identify some CHFs, like CHF_{dl} [22], CHF_{rsa} [2] and CHF_{fac} [7], and prove their tight S-MU-CR security based on the discrete logarithm (DL), RSA and factoring assumptions, respectively. Intuitively, the DL problem and RSA problem are random self-reducible. For example, given one DL problem instance (g, g^x), we can create multiple instances (g, g^{x+b_i}), so that the DL problem can be embedded into hash keys of all users. As for CHF_{fac}, we embed the factoring problem instance into the public parameter, which is shared by all users. In this way, no matter which target hash key is chosen by the adversary, the collision can be used to solve the hard problem. That is why tight S-MU-CR security can be achieved.

Applications of Tightly (S-)MU-CR Secure CHF to Signatures. By using our tightly secure chameleon hash functions like CHF_{dl}, CHF_{rsa} and CHF_{fac}, it is possible for us to get a variety of signatures with tight security, see Fig. 1.

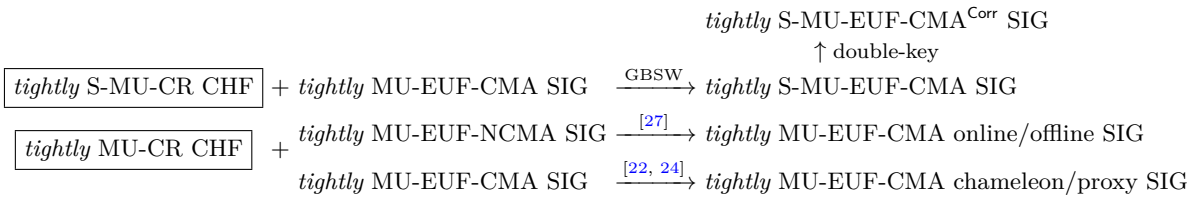

Fig. 1. Applications of tightly (S-)MU-CR secure chameleon hash functions to signatures.

First we extend the GBSW transform [28] to the multi-user setting. With the help of tightly S-MU-CR secure CHFs, the extended GBSW transform lifts a signature scheme from (weak) unforgeability (MU-EUF-CMA) to strong unforgeability (S-MU-EUF-CMA) in the multi-user setting. The resulting S-MU-EUF-CMA secure signature can be used in group signatures [1, 9], CCA secure encryption systems [10], etc. Furthermore, we can cope with corruptions through the "double-key" mechanism [5], and get a tightly S-MU-EUF-CMA$^{\mathsf{Corr}}$ secure signature against adaptive corruptions, which can be used in authenticated key exchange (AKE) protocols [5, 17].

Our tightly MU-CR secure CHFs can further be used in multi-user online/offline signatures [27], chameleon signatures [22] and proxy signatures [24, 12]. All these transforms fall in the "hash-sign-switch" paradigm [27] (see subsec. 5.2), and the security is tightly reduced to the tight security of CHF and signatures in the multi-user setting.

Our Contribution. In conclusion, our contribution is as follows.

1. We define the security notion of *(strong) multi-user collision resistance ((S-)MU-CR)* for chameleon hash functions. Then we present three constructions (CHF_{dl}, CHF_{rsa} and CHF_{fac}) and prove their tight S-MU-CR security based on the discrete logarithm, RSA and factoring assumptions, respectively.

2. We show some applications of tightly (S-)MU-CR secure chameleon hash functions to signatures in the multi-user setting. We first extend the generic GBSW transform to the multi-user setting, resulting in tightly S-MU-EUF-CMA secure signature schemes, which can be further used to construct group signatures and AKE protocols. Meanwhile, the extension can also be applied to online/offline signatures, chameleon signatures and proxy signatures, and help us to get tightly secure signature schemes, which are widely used in grid computing, e-commerce, electronic currency systems, etc.

2 Preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter. For $\mu \in \mathbb{N}$, define $[\mu] := \{1, 2, ..., \mu\}$. Denote by $x := y$ the operation of assigning y to x. Denote by $x \xleftarrow{\$} \mathcal{X}$ the operation of sampling x uniformly at random from a set $\mathcal{X}$. For an algorithm $\mathcal{A}$, denote by $y \leftarrow \mathcal{A}(x)$, the operation of running $\mathcal{A}$ with input x and assigning the output to y. "PPT" is short for probabilistic polynomial-time.

2.1 Chameleon Hash Family

Definition 1 (Chameleon Hash Family). *A chameleon hash family (CHF) consists of four algorithms, namely* $\mathsf{CHF} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Eval}, \mathsf{TdColl})$.

- $\mathsf{Setup}(1^\lambda)$*: The setup algorithm takes as input the security parameter* 1^λ*, and outputs public parameter* $\mathsf{pp}_{\mathsf{CHF}}$*, which determines the key space* $\mathcal{HK}$*, the trapdoor space* $\mathcal{TD}$*, input domains* $\mathcal{M} \times \mathcal{R}$*, and its range* $\mathcal{Y}$*.*
- $\mathsf{KGen}(\mathsf{pp}_{\mathsf{CHF}})$*: The key generation algorithm takes as input* $\mathsf{pp}_{\mathsf{CHF}}$*, and outputs a hash key* $hk \in \mathcal{HK}$ *along with a trapdoor* $td \in \mathcal{TD}$*. Here* hk *determines a specific chameleon hash function* $H_{hk}(\cdot, \cdot)$ *in the chameleon hash family* $\mathsf{H} = \{H_{hk}(\cdot, \cdot)\}_{hk \in \mathcal{HK}}$*.*
- $\mathsf{Eval}(hk, m, r)$*: The evaluation algorithm takes as input* hk*, message* $m \in \mathcal{M}$ *and randomness* $r \in \mathcal{R}$*, and outputs the hash value* $h = H_{hk}(m, r)$[4]*.*
- $\mathsf{TdColl}(td, m_1, r_1, m_2)$*: The trapdoor collision algorithm takes as input the trapdoor* td*, a message-randomness pair* (m_1, r_1) *and another message* m_2*, and outputs* r_2 *such that* $H_{hk}(m_1, r_1) = H_{hk}(m_2, r_2)$*.*

CHF *is strongly secure if it has the following two properties.*

Strong Collision Resistance (S-CR). *For any PPT adversary* $\mathcal{A}$*, the advantage* $\mathsf{Adv}^{\mathsf{s\text{-}cr}}_{\mathsf{CHF},\mathcal{A}}(\lambda)$ *is negligible, where* $\mathsf{Adv}^{\mathsf{s\text{-}cr}}_{\mathsf{CHF},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{c} \mathsf{pp}_{\mathsf{CHF}} \leftarrow \mathsf{Setup}(1^\lambda); \\ (hk, td) \leftarrow \mathsf{KGen}(\mathsf{pp}_{\mathsf{CHF}}); \\ (m_1, r_1, m_2, r_2) \leftarrow \mathcal{A}(\mathsf{pp}_{\mathsf{CHF}}, hk) \end{array} : \begin{array}{c} H_{hk}(m_1, r_1) = H_{hk}(m_2, r_2) \\ \wedge\ (m_1, r_1) \neq (m_2, r_2) \end{array}\right].$$

Random Trapdoor Collision (RTC). *For every* $(hk, td) \in \mathcal{HK} \times \mathcal{TD}$*,* $m_1, m_2 \in \mathcal{M}$*, if* r_1 *is distributed uniformly over* $\mathcal{R}$*, then* $r_2 := \mathsf{TdColl}(td, m_1, r_1, m_2)$ *enjoys a uniform distribution over* $\mathcal{R}$*.*

[4] We assume that Eval and TdColl take $\mathsf{pp}_{\mathsf{CHF}}$ as an implicit input.

Some definitions of chameleon hash family require uniformity of the hash value $H(m, r)$ (when r is uniform) instead of random trapdoor collision property.

Definition 2 (Uniformity [22, 7]). *A chameleon hash family* $\mathsf{CHF} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Eval}, \mathsf{TdColl})$ *has uniformity property if for every* $(hk, td) \in \mathcal{HK} \times \mathcal{SK}$ *and* $m \in \mathcal{M}$*, if* r *is distributed uniformly over* $\mathcal{R}$*, then* $H_{hk}(m, r)$ *has a uniform distribution over* $\mathcal{Y}$*.*

For a chameleon hash family, uniformity property and random trapdoor collision property characterize different aspects. Uniformity [22, 7, 8] is helpful to define semantic security [2, 3], where an adversary (even all-powerful) cannot get any information about m from its hash value $H_{hk}(m, r)$ if r is chosen uniformly at random. While random trapdoor collision property [27, 13, 28] implies that no one can distinguish whether r is a randomness chosen uniformly at random, or r is the output of the trapdoor collision function.

The following theorem describes some links between uniformity and random trapdoor collision properties.

Theorem 1. *Let* $\mathsf{CHF} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Eval}, \mathsf{TdColl})$ *be a chameleon hash family with input domains* $\mathcal{M} \times \mathcal{R}$ *and range* $\mathcal{Y}$*. If* $|\mathcal{R}| = |\mathcal{Y}|$*, then* CHF *has uniformity property iff it has random trapdoor collision property.*

Proof. Consider a specific key pair (hk, td).

1. Uniformity implies random trapdoor collision.
   For any $m \in \mathcal{M}$, the evaluation function $H_{hk}(m, \cdot)$ defines a mapping from $\mathcal{R}$ to $\mathcal{Y}$. Since CHF has uniformity and $|\mathcal{R}| = |\mathcal{Y}|$, $H_{hk}(m, \cdot)$ must be a bijection. Hence, function $\mathsf{TdColl}(td, m_1, \cdot, m_2)$ is equivalent to $H_{hk}^{-1}(m_2, H_{hk}(m_1, \cdot))$, which is a bijection too. So if r_1 is chosen uniformly at random, $r_2 := \mathsf{TdColl}(td, m_1, r_1, m_2)$ has a uniform distribution over $\mathcal{R}$.
2. Random trapdoor collision implies uniformity.
   We prove it by contradiction. Suppose CHF does not have uniformity property, i.e., there exists $m_1 \in \mathcal{M}$ such that $H_{hk}(m, \cdot)$ is not a bijection from $\mathcal{R}$ to $\mathcal{Y}$. Hence, there must exist $h^* \in \mathcal{Y}$ such that $H_{hk}(m_1, r_1) \neq h^*$ for all r_1. Meanwhile, since $h^* \in \mathcal{Y}$, we can always find (m_2, r_2) such that $H_{hk}(m_2, r_2) = h^*$. Thus, for r_1 chosen uniformly at random, $r_2 := \mathsf{TdColl}(td, m_1, r_1, m_2)$ is not uniformly distributed. □

In the case of $|\mathcal{R}| < |\mathcal{Y}|$, it is impossible for CHF to have uniformity: for any $m \in \mathcal{M}$, the range defined by $H_{hk}(m, \cdot)$ cannot cover $\mathcal{Y}$, consequently the trapdoor collision function cannot work. However, in the case of $|\mathcal{R}| > |\mathcal{Y}|$, it is difficult to analyse the relation between uniformity and random trapdoor collision properties.

2.2 Security of Chameleon Hash Family in the Multi-User Setting

Now we extend the S-CR security of CHF to the multi-user setting by defining *strong multi-user collision resistance* (S-MU-CR) security for CHF.

Definition 3. *A chameleon hash family* $\mathsf{CHF} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Eval}, \mathsf{TdColl})$ *is strongly secure in the multi-user setting if it has RTC property (as defined in Def.1) and strong multi-user collision resistance.*

Strong Multi-User Collision Resistance (S-MU-CR). *For any PPT adversary $\mathcal{A}$, the advantage* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF},\mu,\mathcal{A}}(\lambda)$ *is negligible, where* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF},\mu,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{c}\mathsf{pp}_{\mathsf{CHF}} \leftarrow \mathsf{Setup}(1^\lambda);\\ (hk_i, td_i) \leftarrow \mathsf{KGen}(\mathsf{pp}_{\mathsf{CHF}}) \text{ for } i \in [\mu];\\ (i^*, m_1, r_1, m_2, r_2) \leftarrow \mathcal{A}(\mathsf{pp}_{\mathsf{CHF}}, \{hk_i\}_{i\in[\mu]})\end{array} : \begin{array}{c}H_{hk_{i^*}}(m_1, r_1) = H_{hk_{i^*}}(m_2, r_2)\\ \wedge\ (m_1, r_1) \neq (m_2, r_2)\end{array}\right].$$

If we only require $m_1 \neq m_2$, then S-MU-CR security becomes MU-CR security, and CHF is (non-strongly) secure in the multi-user setting if it has RTC property and MU-CR security.

3 Tightly Secure Chameleon Hash Functions in the Multi-User Setting

In this section, we review some chameleon hash families, CHF_{dl}, CHF_{rsa} and CHF_{fac}, and prove their tight S-MU-CR security and RTC property. Recall that these constructions are originally proposed in [22, 2, 7], but their collision resistance security are proved in the single user setting, and uniformity or semantic security of hash values are considered instead of RTC property.

3.1 Chameleon Hash Family Based on the DL Assumption

Let GGen be a group generation algorithm that outputs a cyclic group $\mathbb{G}$ of prime order q with generator g. In formula, $:= (\mathbb{G}, q, g) \leftarrow \mathsf{GGen}(1^\lambda)$.

Definition 4 (The DL Assumption). *For any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in solving the discrete logarithm (DL) problem is defined as*

$$\mathsf{Adv}^{\mathsf{dl}}_{\mathbb{G},\mathcal{A}}(\lambda) := \Pr[(\mathbb{G}, q, g) \leftarrow \mathsf{GGen}(1^\lambda); x \xleftarrow{\$} \mathbb{Z}_q : \mathcal{A}(\mathbb{G}, q, g, g^x) = x].$$

The DL assumption states that for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{dl}}_{\mathbb{G},\mathcal{A}}(\lambda)$ is negligible.

The construction of CHF_{dl}[5] [22] is shown in Fig. 2.

$\underline{\mathsf{Setup}(1^\lambda)}$: $(\mathbb{G}, q, g) \leftarrow \mathsf{GGen}(1^\lambda)$ Define $\mathcal{M} := \mathbb{Z}_q$, $\mathcal{R} := \mathbb{Z}_q$, $\mathcal{Y} := \mathbb{G}$ Return $\mathsf{pp}_{\mathsf{CHF}} := (\mathbb{G}, q, g, \mathcal{M}, \mathcal{R}, \mathcal{Y})$ $\underline{\mathsf{KGen}(\mathsf{pp}_{\mathsf{CHF}})}$: $x \xleftarrow{\$} \mathbb{Z}_q$; $X := g^x$ Return $(hk := X, td := x)$	$\underline{\mathsf{Eval}(hk, m, r)}$: $h := hk^m \cdot g^r$ Return h $\underline{\mathsf{TdColl}(td, m_1, r_1, m_2)}$: $r_2 := td \cdot (m_1 - m_2) + r_1 \mod q$ Return r_2

Fig. 2. Construction of CHF_{dl}.

[5] Here we compute the hash value with $h = X^m \cdot g^r$ instead of $h = g^m \cdot X^r$ as in [22]. One can easily see that they are essentially the same.

Theorem 2. CHF_{dl} *has tight strong security in the multi-user setting. That is, it has not only the RTC property but also the tight S-MU-CR security from the DL assumption. More precisely, for any PPT adversary* $\mathcal{A}$ *with advantage* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{dl},\mu,\mathcal{A}}(\lambda)$*, there exists a PPT algorithm* $\mathcal{B}$ *against the DL problem such that* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{dl},\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{dl}}_{\mathbb{G},\mathcal{B}}(\lambda)$.

Proof. It is easy to prove the RTC property. For any $td = x$, $m_1, m_2 \in \mathbb{Z}_q$, $r_1 \in \mathbb{Z}_q$, we have $r_2 := x \cdot (m_1 - m_2) + r_1$. Hence, if r_1 is independently chosen from $\mathbb{Z}_q$ uniformly at random, then $r_2 := x \cdot (m_1 - m_2) + r_1$ is uniform over $\mathbb{Z}_q$ as well.

Next we prove that for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{dl},\mu,\mathcal{A}}$ is negligible under the DL assumption. To this end, we construct a PPT algorithm $\mathcal{B}$ against the DL problem. $\mathcal{B}$ gets a group description $= (\mathbb{G}, q, g)$ along with a challenge $(g, X = g^x)$ from its DL challenger.

$\mathcal{B}$ directly sets $\mathsf{pp}_{\mathsf{CHF}} := (\mathbb{G}, q, g, \mathcal{M}, \mathcal{R}, \mathcal{Y})$ with $\mathcal{M} := \mathbb{Z}_q, \mathcal{R} := \mathbb{Z}_q, \mathcal{Y} := \mathbb{G}$. For $i \in [\mu]$, $\mathcal{B}$ samples $b_i \xleftarrow{\$} \mathbb{Z}_q$, and sets $hk_i := X \cdot g^{b_i}$. In this way, $\mathcal{B}$ implicitly sets $td_i := x_i := x + b_i$. Then $\mathcal{B}$ sends $\mathsf{pp}_{\mathsf{CHF}}$ and $\{hk_i\}_{i\in[\mu]}$ to $\mathcal{A}$. Finally $\mathcal{A}$ outputs $(i^*, m_1, r_1, m_2, r_2)$. If $m_1 \neq m_2$, $\mathcal{B}$ outputs $(r_2 - r_1)/(m_1 - m_2) - b_{i^*}$ as its answer to the DL problem.

If $\mathcal{A}$ successfully finds a collision, then $g^{x_{i^*} m_1 + r_1} = g^{x_{i^*} m_2 + r_2}$ and $(m_1, r_1) \neq (m_2, r_2)$. If $m_1 = m_2$, then $r_1 = r_2$. So we must have $m_1 \neq m_2$ and $x_{i^*} = (r_2 - r_1)/(m_1 - m_2)$ when $\mathcal{A}$ succeeds. As a result, $x := x_{i^*} - b_{i^*}$ is the correct answer to the DL problem, and $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{dl},\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{dl}}_{\mathbb{G},\mathcal{B}}(\lambda)$. □

3.2 Chameleon Hash Family Based on the RSA Assumption

Let RSAGen be an algorithm that outputs an RSA tuple (N, p, q, e, d), where p, q are safe primes of bit-length $\lambda/2$, $N = pq$ and $ed \equiv 1 \mod \varphi(N)$. In formula, $(N, p, q, e, d) \leftarrow \mathsf{RSAGen}(1^\lambda)$. Here we limit that e is a prime and $e > 2^{L(\lambda)}$, where $L(\cdot)$ is the challenge length function associated with RSAGen.

Definition 5 (The RSA Assumption). *For any adversary* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in solving the RSA problem is defined as*

$$\mathsf{Adv}^{\mathsf{rsa}}_{N,e,\mathcal{A}}(\lambda) := \Pr[(N, p, q, e, d) \leftarrow \mathsf{RSAGen}(1^\lambda); x \xleftarrow{\$} \mathbb{Z}_N^* : \mathcal{A}(N, e, x^e) = x].$$

The RSA assumption states that for any PPT adversary $\mathcal{A}$*,* $\mathsf{Adv}^{\mathsf{rsa}}_{N,e,\mathcal{A}}(\lambda)$ *is negligible.*

The construction of CHF_{rsa} [2, 7] is shown in Fig. 3.

Theorem 3. CHF_{rsa} *has tight strong security in the multi-user setting. That is, it has not only the RTC property but also the tight S-MU-CR security from the RSA assumption. More precisely, for any PPT adversary* $\mathcal{A}$ *with advantage* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{rsa},\mu,\mathcal{A}}(\lambda)$*, there exists a PPT algorithm* $\mathcal{B}$ *against the RSA problem such that* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{rsa},\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{rsa}}_{N,e,\mathcal{B}}(\lambda)$.

Proof. Recall that $\mathbb{Z}_N^*$ is a multiplicative group. For any fixed $td = x$ and $m_1, m_2 \in \{0,1\}^\ell$, if r_1 is uniform over $\mathbb{Z}_N^*$, then $x^{m_1 - m_2} \cdot r_1$ is also uniform. This gives the RTC property of CHF_{rsa}.

As for the proof of S-MU-CR security, we construct a PPT algorithm $\mathcal{B}$ against the RSA problem as follows. $\mathcal{B}$ gets (N, e) and $X = x^e$ from its challenger, where $x \xleftarrow{\$} \mathbb{Z}_N^*$.

$\underline{\mathsf{Setup}(1^\lambda)}$: $(N,p,q,e,d) \leftarrow \mathsf{RSAGen}(1^\lambda)$ $\ell := L(\lambda)$ Define $\mathcal{M} := \{0,1\}^\ell$, $\mathcal{R} := \mathbb{Z}_N^*$, $\mathcal{Y} := \mathbb{Z}_N^*$ Return $\mathsf{pp}_{\mathsf{CHF}} := (N,e,\mathcal{M},\mathcal{R},\mathcal{Y})$ $\underline{\mathsf{KGen}(\mathsf{pp}_{\mathsf{CHF}})}$: $x \xleftarrow{\$} \mathbb{Z}_N^*$; $X := x^e \mod N$ Return $(hk := X, td := x)$	$\underline{\mathsf{Eval}(hk,m,r)}$: $h := hk^m \cdot r^e \mod N$ Return h $\underline{\mathsf{TdColl}(td,m_1,r_1,m_2)}$: $r_2 := td^{m_1-m_2} \cdot r_1 \mod N$ Return r_2

Fig. 3. Construction of CHF_{rsa}.

The public parameter is set as $\mathsf{pp}_{\mathsf{CHF}} := (N,e,\mathcal{M},\mathcal{R},\mathcal{Y})$ with $\mathcal{M} := \{0,1\}^\ell, \mathcal{R} := \mathbb{Z}_N^*, \mathcal{Y} := \mathbb{Z}_N^*$. For $i \in [\mu]$, $\mathcal{B}$ samples $b_i \xleftarrow{\$} \mathbb{Z}_N^*$ and sets $hk_i := X_i = X \cdot b_i^e$. In this way, $\mathcal{B}$ implicitly sets $td_i := x_i := x \cdot b_i$. Then $\mathcal{B}$ sends $\mathsf{pp}_{\mathsf{CHF}}$ and $\{hk_i\}_{i\in[\mu]}$ to $\mathcal{A}$. Finally $\mathcal{A}$ outputs $(i^*, m_1, r_1, m_2, r_2)$, and $\mathcal{B}$ outputs $(r_2/r_1)^\beta \cdot X_{i^*}^\alpha \cdot b_{i^*}^{-1}$ as its answer to the RSA problem, where $\alpha e + \beta(m_1 - m_2) = 1$.

Suppose $\mathcal{A}$ successfully finds a collision. That is, $H_{X_{i^*}}(m_1,r_1) = H_{X_{i^*}}(m_2,r_2)$, so $(x_{i^*}^{m_1} \cdot r_1)^e = (x_{i^*}^{m_2} \cdot r_2)^e$. Note that $f_e : x \mapsto x^e$ is a bijection over $\mathbb{Z}_N^*$. Hence $x_{i^*}^{m_1} \cdot r_1 = x_{i^*}^{m_2} \cdot r_2$, equivalently $x_{i^*}^{m_1-m_2} = r_2/r_1$. We must have $m_1 \neq m_2$, otherwise $r_1 = r_2$ and $\mathcal{A}$ fails. Let α, β be two integers such that $\alpha e + \beta(m_1 - m_2) = 1$ (α and β can always be found since e is a prime and $e > 2^{L(\lambda)}$), then we get $x_{i^*} = (r_2/r_1)^\beta \cdot X_{i^*}^\alpha$. And $x := x_{i^*}/b_{i^*}$ is the correct answer to the RSA problem. So $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{rsa},\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{rsa}}_{N,e,\mathcal{B}}(\lambda)$. □

3.3 Chameleon Hash Family Based on the Factoring Assumption

Let FacGen be an algorithm that outputs (N,p,q), where p,q are safe primes of bit-length $\lambda/2$ and $N = pq$. In formula, $(N,p,q) \leftarrow \mathsf{FacGen}(1^\lambda)$.

Definition 6 (The Factoring Assumption). *For any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in solving the factoring problem is defined as*

$$\mathsf{Adv}^{\mathsf{fac}}_{N,\mathcal{A}}(\lambda) := \Pr[(N,p,q) \leftarrow \mathsf{FacGen}(1^\lambda) : \mathcal{A}(N) = p \ \vee \ \mathcal{A}(N) = q].$$

The factoring assumption states that for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{fac}}_{N,\mathcal{A}}(\lambda)$ is negligible.

Define $\mathbb{Z}_N^+ := \mathbb{Z}_N^* \cap \{1,...,N/2\}$. For $m \in \{0,1\}^\ell$, denote by m_k the k-th bit of m. The construction of CHF_{fac} [7] is shown in Fig. 4.

Theorem 4. *CHF_{fac} has tight strong security in the multi-user setting. That is, it has not only the RTC property but also the tight S-MU-CR security from the factoring assumption. More precisely, for any PPT adversary $\mathcal{A}$ with advantage $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{fac},\mu,\mathcal{A}}(\lambda)$, there exists a PPT algorithm $\mathcal{B}$ against the factoring problem such that $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{fac},\mu,\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{fac}}_{N,\mathcal{B}}(\lambda)$.*

Proof. For the proof of random trapdoor collision (RTC) property, consider fixed values of $td = (s_1,...,s_\ell)$, $m_1, m_2 \in \{0,1\}^\ell$. The algorithm $\mathsf{TdColl}(td,m_1,r_1,m_2)$ returns $r_2 := \min\{\tau \cdot r_1, N - \tau \cdot r_1\}$, where $\tau := \prod_{k=1}^{\ell} s_k^{m_{1,k}-m_{2,k}}$ is some fixed value in $\mathbb{Z}_N^*$. We just

$\underline{\mathsf{Setup}(1^\lambda)}$:	$\underline{\mathsf{Eval}(hk, m, r)}$:
$(N, p, q) \leftarrow \mathsf{FacGen}(1^\lambda)$	Parse $hk = (u_1, ..., u_\ell)$
$\ell := \mathsf{poly}(\lambda)$	$h := \prod_{k=1}^{\ell} u_k^{m_k} \cdot r^2 \mod N$
Define $\mathcal{M} := \{0,1\}^\ell$, $\mathcal{R} := \mathbb{Z}_N^+$, $\mathcal{Y} := \mathbb{QR}_N$	Return h
Return $\mathsf{pp}_{\mathsf{CHF}} := (N, \mathcal{M}, \mathcal{R}, \mathcal{Y})$	
$\underline{\mathsf{KGen}(\mathsf{pp}_{\mathsf{CHF}})}$:	$\underline{\mathsf{TdColl}(td, m_1, r_1, m_2)}$:
For $k \in [\ell]$:	Parse $td = (s_1, ..., s_\ell)$
$s_k \xleftarrow{\$} \mathbb{Z}_N^*$; $u_k := s_k^2 \mod N$	$r_2 := \prod_{k=1}^{\ell} s_k^{m_{1,k} - m_{2,k}} \cdot r_1$
$hk := (u_1, ..., u_\ell)$; $td := (s_1, ..., s_\ell)$	$r_2 := \min\{r_2, N - r_2\}$
Return (hk, td)	Return r_2

Fig. 4. Construction of CHF_{fac}.

need to prove that the function $f_\tau(r_1) := \min\{\tau \cdot r_1, N - \tau \cdot r_1\}$ is an injection (hence bijection) over $\mathbb{Z}_N^+$. This is justified by the fact that for $r_1, r_1' \in \mathbb{Z}_N^+$ with $r_1 \neq r_1'$, neither $\tau \cdot r_1 \equiv \tau \cdot r_1' \mod N$ nor $\tau(r_1 + r_1') \equiv 0 \mod N$, i.e., no two distinct inputs correspond to the same output. With a bijection $f_\tau : \mathcal{R} \to \mathcal{R}$, $r_2 := f_\tau(r_1)$ is uniformly random as long as r_1 is, and the RTC property follows.

Then we prove $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{fac},\mu,\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{fac}}_{N,\mathcal{B}}(\lambda)$. We construct a PPT algorithm $\mathcal{B}$ against the factoring problem. $\mathcal{B}$ gets N from its own challenger. The public parameter is set as $\mathsf{pp}_{\mathsf{CHF}} := (N, \mathcal{M}, \mathcal{R}, \mathcal{Y})$ with $\mathcal{M} := \{0,1\}^\ell, \mathcal{R} := \mathbb{Z}_N^+, \mathcal{Y} := \mathbb{QR}_N$. For $i \in [\mu], k \in [\ell]$, $\mathcal{B}$ samples $s_{i,k} \xleftarrow{\$} \mathbb{Z}_N^*$ and sets $u_{i,k} := s_{i,k}^2$. In this way, $hk_i = (u_{i,1}, ..., u_{i,\ell})$ and $td_i = (s_{i,1}, ..., s_{i,\ell})$. Then $\mathcal{B}$ sends $\mathsf{pp}_{\mathsf{CHF}}$ and $\{hk_i\}_{i\in[\mu]}$ to $\mathcal{A}$.

If $\mathcal{A}$ finds a collision with output $(i^*, m_1, r_1, m_2, r_2)$, then

$$\prod_{k=1}^{\ell} (u_{i^*,k})^{m_{1,k}} \cdot r_1^2 = \prod_{k=1}^{\ell} (u_{i^*,k})^{m_{2,k}} \cdot r_2^2. \tag{1}$$

We discuss Eq. (1) in two cases.

Case 1. $m_1 = m_2$ but $r_1 \neq r_2$.
In this case, Eq. (1) implies $r_1^2 \equiv r_2^2 \mod N$, i.e., $(r_1 + r_2)(r_1 - r_2) \equiv 0 \mod N$. Note that $r_1, r_2 \in \mathbb{Z}_N^+$ and $r_1 \neq r_2$. Thus, $\mathcal{B}$ can always find a factor of N by outputting $\gcd(r_1 + r_2, N)$.

Case 2. $m_1 \neq m_2$. Then there must exist $z \in [\ell]$ such that $m_{1,z} \neq m_{2,z}$.
We can rewrite Eq. (1) as

$$(u_{i^*,z})^{m_{1,z} - m_{2,z}} = \prod_{k \neq z} (u_{i^*,k})^{m_{2,k} - m_{1,k}} \cdot (r_2/r_1)^2. \tag{2}$$

Equivalently,

$$\left((s_{i^*,z})^{m_{1,z} - m_{2,z}}\right)^2 = \left(\prod_{k \neq z} (s_{i^*,k})^{m_{2,k} - m_{1,k}} \cdot (r_2/r_1)\right)^2. \tag{3}$$

We denote the right part of Eq.(3) by Δ^2. Note that $m_{1,z} - m_{2,z} = \pm 1$.

– If $m_{1,z} - m_{2,z} = 1$, then Eq.(3) is simplified to $(s_{i^*,z})^2 = \Delta^2$, and $\mathcal{B}$ outputs $\gcd(s_{i^*,z} + \Delta, N)$.
– If $m_{1,z} - m_{2,z} = -1$, then Eq.(3) is simplified to $(s_{i^*,z}^{-1})^2 = \Delta^2$, and $\mathcal{B}$ outputs $\gcd(s_{i^*,z}^{-1} + \Delta, N)$.

Recall that $s_{i^*,z}$ is chosen randomly in $\mathbb{Z}_N^*$, and the only information $\mathcal{A}$ gets is $u_{i^*,z} = (s_{i^*,z})^2$. Thus, $s_{i^*,z} \notin \{\Delta, N-\Delta\}$ with probability 1/2, or $s_{i^*,z}^{-1} \notin \{\Delta, N-\Delta\}$ with probability 1/2. In either case, $\mathcal{B}$ successfully factors N with probability 1/2.

In conclusion, $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{CHF}_{fac},\mu,\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{fac}}_{N,\mathcal{B}}(\lambda)$. □

4 Extending Message Space to Arbitrary String

For a chameleon hash family H, we can extend the message space from $\mathcal{M}$ to bit strings of any polynomial length by applying a traditional collision resistant hash function J to the message first. The composition [22] results in a chameleon hash family over $\{0,1\}^*$, and the S-MU-CR security can be tightly reduced to the S-MU-CR security of H and the collision resistance security of J.

Theorem 5. *Let $\mathsf{H} = \{H_{hk}(\cdot,\cdot)\}_{hk\in\mathcal{HK}}$ be a chameleon hash family with tight and strong security and associated with message domain $\mathcal{M}$. Let $J : \{0,1\}^* \to \mathcal{M}$ be a collision resistant hash function. Then $\mathsf{H}' := \{H_{hk}(J(\cdot),\cdot)\}_{hk\in\mathcal{HK}}$ is also a chameleon hash family with tight and strong security but its message space is extended to $\{0,1\}^*$. More precisely, for any PPT adversary $\mathcal{A}$ against the S-MU-CR security of H', there exist PPT algorithms $\mathcal{B}_\mathsf{H}$ against the S-MU-CR security of H, and $\mathcal{B}_J$ against the collision resistance security of J, such that $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{H}',\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{H},\mu,\mathcal{B}_\mathsf{H}}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{J,\mathcal{B}_J}(\lambda)$.*

Proof. The RTC property of H' directly follows from H. As for the S-MU-CR security, let $\{(hk_i, td_i)\}_{i\in[\mu]}$ be the challenge key pairs of H', and $\mathcal{A}$ successfully finds a collision by outputting $(i^*, m_1, r_1, m_2, r_2)$, i.e., $H_{hk_{i^*}}(J(m_1), r_1) = H_{hk_{i^*}}(J(m_2), r_2)$ but $(m_1, r_1) \neq (m_2, r_2)$. We analyse it in two cases.

Case 1. $m_1 \neq m_2 \ \wedge \ J(m_1) = J(m_2)$.
This means a collision of J. Hence, we can construct a PPT algorithm $\mathcal{B}_J$ against J's collision resistance security, and $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{H}',\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{cr}}_{J,\mathcal{B}_J}(\lambda)$.

Case 2. $J(m_1) \neq J(m_2) \ \vee \ r_1 \neq r_2$.
Excluding case 1, either $J(m_1) \neq J(m_2)$ or $r_1 \neq r_2$ holds. Hence, we can construct a PPT algorithm $\mathcal{B}_\mathsf{H}$ against H's S-MU-CR security by outputting $(i^*, J(m_1), r_1, J(m_2), r_2)$, and $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{H}',\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{H},\mu,\mathcal{B}_\mathsf{H}}(\lambda)$. □

5 Applications of Tightly (S-)MU-CR Secure Chameleon Hash Families to Signatures in the Multi-User Setting

By using our tightly S-MU-CR secure chameleon hash families, we can extend the generic GBSW transform [28] to the multi-user setting and achieve tight security, as shown in subsec. 5.1. Meanwhile, in subsec. 5.2, tightly MU-CR secure chameleon hash families can be further applied to online/offline signatures [27], chameleon signatures [22], proxy signatures [24, 12], etc., to achieve tight security in the multi-user setting.

Since all applications in this section are signatures, we first present the definition of signature and its security notions in the multi-user setting.

Definition 7 (Signature Scheme). *A signature (SIG) scheme consists of four algorithms,* $\mathsf{S} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$.

- $\mathsf{Setup}(1^\lambda)$*: The setup algorithm takes as input the security parameter* 1^λ *and outputs public parameter* $\mathsf{pp_S}$*. Note that* $\mathsf{pp_S}$ *is an implicit input of* Sign *and* Ver*.*
- $\mathsf{KGen}(\mathsf{pp_S})$*: The key generation algorithm takes as input* $\mathsf{pp_S}$ *and outputs a verification/signing key pair* (vk, sk)*.*
- $\mathsf{Sign}(sk, m)$*: The signing algorithm takes as input the signing key* sk *and message* m*, and outputs a signature* σ*.*
- $\mathsf{Ver}(vk, m, \sigma)$*: The verification algorithm takes as input the verification key* vk*, a message* m *and a signature* σ*, and outputs a bit 1/0, indicating whether* σ *is a valid signature of* m*.*

Definition 8 (Security of Signature Scheme). *A signature scheme* $\mathsf{S} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ *is existentially unforgeable under chosen message attacks in the multi-user setting (MU-EUF-CMA), if for any PPT adversary* $\mathcal{A}$*, the advantage* $\mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}cma}}_{\mathsf{S},\mu,\mathcal{A}}(\lambda)$ *is negligible, where* $\mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}cma}}_{\mathsf{S},\mu,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{c} \mathsf{pp_S} \leftarrow \mathsf{Setup}(1^\lambda) \\ (vk_i, sk_i) \leftarrow \mathsf{KGen}(\mathsf{pp_S}) \ for\ i \in [\mu] \\ (i^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{SIGN}}(\cdot,\cdot)}(\mathsf{pp_S}, \{vk_i\}_{i\in[\mu]}) \end{array} : \begin{array}{c} i^* \in [\mu] \ \wedge \ (m^*, \cdot) \notin \mathcal{Q}_{i^*} \\ \wedge\ \mathsf{Ver}(vk_i, m^*, \sigma^*) = 1 \end{array}\right]. \tag{4}$$

Here $\mathcal{O}_{SIGN}(\cdot,\cdot)$ *is an oracle that takes* (i, m) *as input, invokes* $\sigma \leftarrow \mathsf{Sign}(sk_i, m)$*, updates* $\mathcal{Q}_i := \mathcal{Q}_i \cup \{(m, \sigma)\}$ *and returns* σ*.*

- *If* $(m^*, \cdot) \notin \mathcal{Q}_{i^*}$ *is replaced by* $(m^*, \sigma^*) \notin \mathcal{Q}_{i^*}$ *in (4),* S *is strongly existentially unforgeable under chosen message attacks in the multi-user setting (S-MU-EUF-CMA), and the advantage of* $\mathcal{A}$ *is denoted by* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}euf\text{-}cma}}_{\mathsf{S},\mu,\mathcal{A}}(\lambda)$*.*
- *If* $\mathcal{A}$ *submits all signing queries* $\{(i, m)\}$ *before it sees* $\mathsf{pp_S}$ *and* $\{vk_i\}_{i\in[\mu]}$ *in (4),* S *is existentially unforgeable under non-adaptive chosen message attacks in the multi-user setting (MU-EUF-NCMA), and the advantage of* $\mathcal{A}$ *is denoted by* $\mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}ncma}}_{\mathsf{S},\mu,\mathcal{A}}(\lambda)$*.*
- *If* $\mu = 1$ *(and* $(m^*, \sigma^*) \notin \mathcal{Q}_{i^*}$*) in (4),* S *is (strongly) existentially unforgeable under chosen message attacks ((S-)EUF-CMA) in the single user setting, and the advantage of* $\mathcal{A}$ *is denoted by* $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{S},\mathcal{A}}(\lambda)$ *(w.r.t.* $\mathsf{Adv}^{\mathsf{s\text{-}euf\text{-}cma}}_{\mathsf{S},\mathcal{A}}(\lambda)$*).*

Signature schemes with tight MU-EUF-CMA security can be found in [20, 4, 17, 29, 15, 18, 30].

5.1 Generic Transform for Signatures from MU-EUF-CMA Security to S-MU-EUF-CMA Security

In [28], Steinfeld, Pieprzyk and Wang proposed a generic transform (the GBSW transform), which can invert an EUF-CMA secure signature scheme to a S-EUF-CMA secure signature scheme. The GBSW transform is (security) tightness preserving, but limited only in single user setting. By using our strongly secure chameleon hash families that

have tight S-MU-CR security and RTC property, we are able to extend the GBSW transform to the multi-user setting, which strengthens signature schemes from weak unforgeability (MU-EUF-CMA) to strong unforgeability (S-MU-EUF-CMA) and enjoys a tight security reduction.

The GBSW Transform [28]**.** Let $\mathsf{S} = (\mathsf{S.Setup}, \mathsf{S.KGen}, \mathsf{S.Sign}, \mathsf{S.Ver})$ be a signature scheme with EUF-CMA security, and F, H be two chameleon hash families with strong security (i.e., S-CR security and RTC property). Define the new signature scheme $\mathsf{S}_{\mathsf{GBSW}}$ as follows.

1. $\mathsf{S}_{\mathsf{GBSW}}.\mathsf{Setup}(1^\lambda)$. Invoke $\mathsf{pp}_\mathsf{S} \leftarrow \mathsf{S.Setup}(1^\lambda)$, $\mathsf{pp}_\mathsf{F} \leftarrow \mathsf{F.Setup}(1^\lambda)$, $\mathsf{pp}_\mathsf{H} \leftarrow \mathsf{H.Setup}(1^\lambda)$, and return $\mathsf{pp}_{\mathsf{S}_{\mathsf{GBSW}}} := (\mathsf{pp}_\mathsf{S}, \mathsf{pp}_\mathsf{F}, \mathsf{pp}_\mathsf{H})$.
2. $\mathsf{S}_{\mathsf{GBSW}}.\mathsf{KGen}(\mathsf{pp}_{\mathsf{S}_{\mathsf{GBSW}}})$. Invoke $(vk, sk) \leftarrow \mathsf{S.KGen}(\mathsf{pp}_\mathsf{S})$, $(hk_\mathsf{F}, td_\mathsf{F}) \leftarrow \mathsf{F.KGen}(\mathsf{pp}_\mathsf{F})$, $(hk_\mathsf{H}, td_\mathsf{H}) \leftarrow \mathsf{H.KGen}(\mathsf{pp}_\mathsf{H})$, return $(vk_{\mathsf{S}_{\mathsf{GBSW}}}, sk_{\mathsf{S}_{\mathsf{GBSW}}}) := ((vk, hk_\mathsf{F}, hk_\mathsf{H}), (sk, td_\mathsf{H}, hk_\mathsf{F}, hk_\mathsf{H}))$.
3. $\mathsf{S}_{\mathsf{GBSW}}.\mathsf{Sign}(sk_{\mathsf{S}_{\mathsf{GBSW}}}, m)$.
   (a) Choose random r', s;
   (b) Choose random m', σ', and compute $h := H_{hk_\mathsf{H}}(m'||\sigma', r')$;
   (c) Compute $\bar{m} := F_{hk_\mathsf{F}}(h, s)$ and $\sigma \leftarrow \mathsf{S.Sign}(sk, \bar{m})$;
   (d) Invoke $r \leftarrow \mathsf{H.TdColl}(td_\mathsf{H}, m'||\sigma', r', m||\sigma)$, return $\sigma_{\mathsf{S}_{\mathsf{GBSW}}} := (\sigma, r, s)$.
4. $\mathsf{S}_{\mathsf{GBSW}}.\mathsf{Ver}(vk_{\mathsf{S}_{\mathsf{GBSW}}}, m, \sigma_{\mathsf{S}_{\mathsf{GBSW}}})$.
   (a) Compute $h := H_{hk_\mathsf{H}}(m||\sigma, r)$, $\bar{m} := F_{hk_\mathsf{F}}(h, s)$;
   (b) Return $\mathsf{S.Ver}(vk, \bar{m}, \sigma)$.

The Extended GBSW Transform. The transform is similar to the GBSW transform, except that the building block S is replaced with a signature scheme with MU-EUF-CMA security, and F, H are replaced with chameleon hash families with S-MU-CR security and RTC property.

Theorem 6. *If chameleon hash families* F *and* H *are strongly secure in the multi-user setting,* S *is MU-EUF-CMA secure, then the extended GBSW transform results in a S-MU-EUF-CMA secure signature scheme* $\mathsf{S}_{\mathsf{GBSW}}$ *and the transform is (security) tightness preserving. More precisely, for any PPT adversary* $\mathcal{A}$ *against* $\mathsf{S}_{\mathsf{GBSW}}$*'s strong security with advantage* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}euf\text{-}cma}}_{\mathsf{S}_{\mathsf{GBSW}},\mu,\mathcal{A}}(\lambda)$*, there exist PPT adversaries* $\mathcal{B}_\mathsf{S}, \mathcal{B}_\mathsf{F}$ *and* $\mathcal{B}_\mathsf{H}$*, such that* $\mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}euf\text{-}cma}}_{\mathsf{S}_{\mathsf{GBSW}},\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}cma}}_{\mathsf{S},\mu,\mathcal{B}_\mathsf{S}}(\lambda) + \mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{F},\mu,\mathcal{B}_\mathsf{F}}(\lambda) + \mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{H},\mu,\mathcal{B}_\mathsf{H}}(\lambda)$.

Proof. The proof mainly follows the proof in [28]. Intuitively, if $\mathcal{A}$ succeeds, then either it forges a valid signature on a new message with respect to the signature scheme S, or it finds a collision of F or H. Let $(i^*, m^*, \sigma^*_{\mathsf{GBSW}} = (\sigma^*, r^*, s^*))$ be $\mathcal{A}$'s output and Win be the event that $\mathcal{A}$ wins. Recall that $\mathcal{A}$ wins iff $(m^*, \sigma^*_{\mathsf{GBSW}}) \notin \mathcal{Q}_{i^*} \wedge \mathsf{S}_{\mathsf{GBSW}}.\mathsf{Ver}(vk_{i^*}, m^*, \sigma^*_{\mathsf{GBSW}}) = 1$. Let $h^* = H_{hk_{\mathsf{H},i^*}}(m^*||\sigma^*, r^*)$ and $\bar{m}^* = F_{hk_{\mathsf{F},i^*}}(h^*, s^*)$.

For each signing query (i, m) from $\mathcal{A}$, the challenger returns a signature $\sigma_{\mathsf{GBSW}} = (\sigma, r, s)$ and stores $(m, \sigma_{\mathsf{GBSW}})$ in $\mathcal{Q}_i$. Let $h = H_{hk_{\mathsf{H},i}}(m||\sigma, r)$ and $\bar{m} = F_{hk_{\mathsf{F},i}}(h, s)$ be the internal values in the signing algorithm. Denote by $\bar{\mathcal{Q}}_i$ the set collecting all internal values $\bar{m}$ when answering the signing queries (i, m). We divide the event Win into three cases:

- Win_S: $\bar{m}^* \notin \bar{\mathcal{Q}}_{i^*}$.
- Win_F: There exists $(m, (\sigma, r, s)) \in \mathcal{Q}_{i^*}$ such that $\bar{m} = \bar{m}^*$, but $(h, s) \neq (h^*, s^*)$.

– $\mathsf{Win}_{\mathsf{H}}$: There exists $(m,(\sigma,r,s)) \in \mathcal{Q}_{i^*}$ such that $\bar{m} = \bar{m}^*$ and $(h,s) = (h^*,s^*)$, but $(m,\sigma,r) \neq (m^*,\sigma^*,r^*)$.

Claim 1. $\Pr[\mathsf{Win}_{\mathsf{S}}] \leq \mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}cma}}_{\mathsf{S},\mu,\mathcal{A}}(\lambda)$.

Proof of Claim 1. To prove it, we construct a PPT adversary $\mathcal{B}_{\mathsf{S}}$ against S's MU-EUF-CMA security. $\mathcal{B}_{\mathsf{S}}$ gets pp_{S} and a list of verification keys $\{vk_i\}_{i\in[\mu]}$ from its challenger. Also, the challenger provides $\mathcal{B}_{\mathsf{S}}$ with a signing oracle $\mathcal{O}_{\mathrm{SIGN}}(\cdot,\cdot)$. Then $\mathcal{B}_{\mathsf{S}}$ generates $\mathsf{pp}_{\mathsf{F}}, \mathsf{pp}_{\mathsf{H}}$, $\{(hk_{\mathsf{F},i}, td_{\mathsf{F},i})\}_{i\in[\mu]}$ and $\{(hk_{\mathsf{H},i}, td_{\mathsf{H},i})\}_{i\in[\mu]}$ itself, and sets $\mathsf{pp}_{\mathsf{S}_{\mathsf{GBSW}}} := (\mathsf{pp}_{\mathsf{S}}, \mathsf{pp}_{\mathsf{F}}, \mathsf{pp}_{\mathsf{H}})$, $vk_{\mathsf{GBSW},i} := (vk_i, hk_{\mathsf{F},i}, hk_{\mathsf{H},i})$, $sk_{\mathsf{GBSW},i} := (\cdot, td_{\mathsf{H},i}, hk_{\mathsf{F},i}, hk_{\mathsf{H},i})$. It sends $\mathsf{pp}_{\mathsf{S}_{\mathsf{GBSW}}}$ and the verification key list $\{vk_{\mathsf{GBSW},i}\}_{i\in[\mu]}$ to $\mathcal{A}$ and simulates the strong security experiment.

When $\mathcal{A}$ asks a signing query (i,m), $\mathcal{B}_{\mathsf{S}}$ responses as follows.

(a) Choose random r', s;
(b) Choose random m', σ', and compute $h := H_{hk_{\mathsf{H},i}}(m'||\sigma', r')$;
(c) Compute $\bar{m} := F_{hk_{\mathsf{F},i}}(h,s)$, and query its own signing oracle to get $\sigma \leftarrow \mathcal{O}_{\mathrm{SIGN}}(i,\bar{m})$;
(d) Invoke $r \leftarrow \mathsf{H.TdColl}(td_{\mathsf{H},i}, m'||\sigma', r', m||\sigma)$, return $\sigma_{\mathsf{S}_{\mathsf{GBSW}}} := (\sigma,r,s)$ to $\mathcal{A}$.

Finally $\mathcal{A}$ outputs $(i^*, m^*, \sigma^*_{\mathsf{GBSW}} = (\sigma^*, r^*, s^*))$. If $\mathsf{Win}_{\mathsf{S}}$ happens, then $\mathcal{B}_{\mathsf{S}}$ outputs $(i^*, \bar{m}^*, \sigma^*)$ as its forgery.

It is easy to see that $\mathcal{B}_{\mathsf{S}}$ simulates the experiment perfectly. $\mathsf{Win}_{\mathsf{S}}$ implies $\bar{m}^* \notin \bar{\mathcal{Q}}_{i^*}$ and $\mathsf{S.Ver}(vk_{i^*}, \bar{m}^*, \sigma^*) = 1$. In other words, $\mathcal{B}_{\mathsf{S}}$ has never asked $\mathcal{O}_{\mathrm{SIGN}}(i^*, \bar{m}^*)$, and $(\bar{m}^*, \sigma^*)$ is a valid pair. Therefore, $\mathcal{B}_{\mathsf{S}}$ breaks S's MU-EUF-CMA security if $\mathsf{Win}_{\mathsf{S}}$ happens. ∎

Claim 2. $\Pr[\mathsf{Win}_{\mathsf{F}}] \leq \mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{F},\mu,\mathcal{B}_{\mathsf{F}}}(\lambda)$.

Proof of Claim 2. We construct a PPT adversary $\mathcal{B}_{\mathsf{F}}$ against F's S-MU-CR security. $\mathcal{B}_{\mathsf{F}}$ gets pp_{F} and μ hash keys $\{hk_{\mathsf{F},i}\}_{i\in[\mu]}$ from its own challenger. Recall that in $\mathsf{S}_{\mathsf{GBSW}}$, $sk_{\mathsf{GBSW},i} = (sk_i, td_{\mathsf{H},i}, hk_{\mathsf{F},i}, hk_{\mathsf{H},i})$, and we do not use $td_{\mathsf{F},i}$ in the signing algorithm at all. That is, $\mathcal{B}_{\mathsf{F}}$ can simulate S and H itself and embed $\{hk_{\mathsf{F},i}\}_{i\in[\mu]}$ into the verification key list. Obviously, $\mathcal{B}_{\mathsf{F}}$'s simulation is perfect.

If $\mathsf{Win}_{\mathsf{F}}$ happens, then $\mathcal{B}_{\mathsf{F}}$ finds a collision of F by outputting (i^*, h, s, h^*, s^*). Hence, $\Pr[\mathsf{Win}_{\mathsf{F}}] \leq \mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{F},\mu,\mathcal{B}_{\mathsf{F}}}(\lambda)$. ∎

Claim 3. $\Pr[\mathsf{Win}_{\mathsf{H}}] \leq \mathsf{Adv}^{\mathsf{s\text{-}mu\text{-}cr}}_{\mathsf{H},\mu,\mathcal{B}_{\mathsf{H}}}(\lambda)$.

Proof of Claim 3. To this end, we construct a PPT adversary $\mathcal{B}_{\mathsf{F}}$ against H's S-MU-CR security. $\mathcal{B}_{\mathsf{H}}$ gets pp_{H} and μ hash keys $\{hk_{\mathsf{H},i}\}_{i\in[\mu]}$ from its own challenger. Then it simulates S and F itself, sets $\mathsf{pp}_{\mathsf{S}_{\mathsf{GBSW}}} := (\mathsf{pp}_{\mathsf{S}}, \mathsf{pp}_{\mathsf{F}}, \mathsf{pp}_{\mathsf{H}})$, $hk_{\mathsf{GBSW},i} := (vk_i, hk_{\mathsf{F},i}, hk_{\mathsf{H},i})$, $sk_{\mathsf{GBSW},i} := (sk_i, td_{\mathsf{F},i}, hk_{\mathsf{F},i}, hk_{\mathsf{H},i})$, and sends $\mathsf{pp}_{\mathsf{S}_{\mathsf{GBSW}}}$ and the verification key list to $\mathcal{A}$.

When $\mathcal{A}$ asks a signing query (i,m), $\mathcal{B}_{\mathsf{S}}$ uses trapdoors of F to generate signatures. In detail, $\mathcal{B}$ responses as follows.

(a) Choose random r, s', random h', compute $\bar{m} := F_{hk_{\mathsf{F},i}}(h', s')$;
(b) Invoke $\sigma \leftarrow \mathsf{S.Sign}(sk_i, \bar{m})$;
(c) Compute $h := H_{hk_{\mathsf{H},i}}(m||\sigma, r)$;
(d) Invoke $s \leftarrow \mathsf{F.TdColl}(td_{\mathsf{F},i}, h', s', h)$, return $\sigma_{\mathsf{S}_{\mathsf{GBSW}}} := (\sigma,r,s)$ to $\mathcal{A}$.

Finally $\mathcal{A}$ outputs $(i^*, m^*, \sigma^*_{\mathsf{GBSW}} = (\sigma^*, r^*, s^*))$. If $\mathsf{Win}_{\mathsf{H}}$ happens, i.e., there exists $(m, (\sigma, r, s)) \in \mathcal{Q}_{i^*}$ such that $h = h^*$ and $(m, \sigma, r) \neq (m^*, \sigma^*, r^*)$, then $\mathcal{B}_{\mathsf{H}}$ finds a collision $H_{hk_{\mathsf{H},i^*}}(m||\sigma, r) = H_{hk_{\mathsf{H},i^*}}(m^*||\sigma^*, r^*)$ and outputs $(i^*, m||\sigma, r, m^*||\sigma^*, r^*)$.

It remains to prove that $\mathcal{B}_{\mathsf{H}}$'s simulation is perfect. This is done by the random trapdoor collision properties of F and H. Consider a specific signature $\sigma_{\mathsf{GBSW}} = (\sigma, r, s)$.

- In the real experiment, r is computed by trapdoor collision function $\mathsf{TdColl}(td_{\mathsf{H}}, \cdot, r', \cdot)$ for random r', and s is chosen uniformly at random.
- In $\mathcal{B}_{\mathsf{H}}$'s simulation, r is chosen uniformly at random, and s is computed by trapdoor collision function $\mathsf{TdColl}(td_{\mathsf{F}}, \cdot, s', \cdot)$ for random s'.

Since F and H have random trapdoor collision properties, $\mathcal{B}_{\mathsf{H}}$'s simulation is identical to the real experiment. ▮

Theorem 6 follows from Claims 1, 2 and 3. □

Extension to SIG against Adaptive Corruptions. In some applications, the adversary may corrupt some users and get their signing keys. Security in this case is formalized by the notion MU-EUF-CMA$^{\mathsf{Corr}}$ security [4]. MU-EUF-CMA$^{\mathsf{Corr}}$ security requires signatures of those uncorrupted users to be unforgeable. S-MU-EUF-CMA$^{\mathsf{Corr}}$ security for SIG can be defined similarly. Via standard hybrid argument we know, EUF-CMA security implies MU-EUF-CMA$^{\mathsf{Corr}}$ security naturally, but with a security loss factor μ. As suggested in [17], S-MU-EUF-CMA$^{\mathsf{Corr}}$ secure SIG can be used to achieve stronger security of matching conversation for authenticated key exchange (AKE).

Note that we cannot transform a tightly MU-EUF-CMA$^{\mathsf{Corr}}$ secure SIG to a strong one with S-MU-EUF-CMA$^{\mathsf{Corr}}$ security, through the extended GBSW transform (i.e., using tightly S-MU-CR secure chameleon hash families). The reason is simple: in the security reduction to H's S-MU-CR security, $\mathcal{B}_{\mathsf{H}}$ has no knowledge of the trapdoor of H (which is a part of signing key), hence it cannot answer corruption queries.

Though the extended GBSW transform does not work, we can resort to the "double-key" mechanism [5, 17, 15] to convert any *tightly* S-MU-EUF-CMA secure S to *tightly* S-MU-EUF-CMA$^{\mathsf{Corr}}$ secure S'.

The "double-key" mechanism works as follows. Each user generates two pairs of keys, (vk_1, sk_1) and (vk_2, sk_2) by the key generation algorithm of S, and take (vk_1, vk_2) as the verification key and (b, sk_b) as the signing key of S'. Here b is an independent and random bit. The signing algorithm of S' uses sk_b and signing algorithm of S to generate σ, and the signature does not contain σ, but a zero knowledge proof π, which proves that either $\mathsf{S.Ver}(vk_1, m, \sigma) = 1$ or $\mathsf{S.Ver}(vk_2, m, \sigma) = 1$. The verification algorithm outputs 1 if the proof π is correct. In the security proof from a MU-EUF-CMA secure signature to a MU-EUF-CMA$^{\mathsf{Corr}}$ secure signature, the reduction algorithm $\mathcal{B}$ can generate a key pair itself, and embed another key pair, whose sk is unknown to $\mathcal{B}$, to the final verification/signing key pair. In this way, $\mathcal{B}$ can answer corruption queries from $\mathcal{A}$ perfectly. Meanwhile, the verification algorithm does not leak the information about b due to the usage of or-proof. So $\mathcal{B}$ can convert $\mathcal{A}$'s ability against MU-EUF-CMA$^{\mathsf{Corr}}$ security to its ability against MU-EUF-CMA security in a tight way.

According to Theorem 6, with the help of CHF_{dl} or CHF_{rsa} or CHF_{fac} in Section 3, we can convert any *tightly* MU-EUF-CMA secure SIG to *tightly* S-MU-EUF-CMA

secure SIG and then *tightly* S-MU-EUF-CMA$^{\mathsf{Corr}}$ secure SIG. Note that we have S-MU-EUF-CMA secure SIGs, then we can use the "double-key" mechanism [5, 17, 15] to cope with corruption queries, and get a S-MU-EUF-CMA$^{\mathsf{Corr}}$ secure SIG, as shown in Fig. 1.

5.2 Online/Offline Signatures, Chameleon Signatures and Proxy Signatures

The first application of chameleon hash functions was chameleon signatures [22]. Later, chameleon hash functions were further used to construct efficient online/offline signatures [27], proxy signatures [24, 12], etc. Most of them consider single user setting only. In this subsection, we show that when the building blocks of chameleon hash families are replaced by our tightly MU-CR secure CHFs, tightly MU-EUF-CMA secure signatures can be converted to tightly secure online/offline, chameleon, proxy signatures in the multi-user setting.

Most of these applications in signatures fall in the "hash-sign-switch" paradigm [27]. That is, there are two key pairs in the system, (vk, sk) for a signature scheme S, and (hk, td) for a chameleon hash family H. In the first phase, a signature σ is generated by $\sigma \leftarrow \mathsf{S.Sign}(sk, H_{hk}(m', r'))$, where m' and r' are chosen randomly. Next in the second phase, given the real message m to be signed, a collision is found by $r \leftarrow \mathsf{H.TdColl}(td, m', r', m)$, and the final signature is (σ, r). One could then verify a message-signature pair by $\mathsf{S.Ver}(vk, H_{hk}(m, r), \sigma)$. Which party generates and keeps (hk, td) depends on the needs of applications. Here we take the generic online/offline transform (called the ST transform) in [27] as example, and prove its tight security in the multi-user setting.

Construction of [27]. Let $\mathsf{S} = (\mathsf{S.Setup}, \mathsf{S.KGen}, \mathsf{S.Sign}, \mathsf{S.Ver})$ be a signature scheme, and $\mathsf{H} = (\mathsf{H.Setup}, \mathsf{H.KGen}, \mathsf{H.Eval}, \mathsf{H.TdColl})$ be a chameleon hash family. Define the online/offline signature $\mathsf{S_{ST}}$ as follows.

1. $\mathsf{S_{ST}.Setup}(1^\lambda)$. Invoke $\mathsf{pp_S} \leftarrow \mathsf{S.Setup}(1^\lambda)$, $\mathsf{pp_H} \leftarrow \mathsf{H.Setup}(1^\lambda)$, and return $\mathsf{pp_{S_{ST}}} := (\mathsf{pp_S}, \mathsf{pp_H})$.
2. $\mathsf{S_{ST}.KGen}(\mathsf{pp_{S_{ST}}})$. Invoke $(vk, sk) \leftarrow \mathsf{S.KGen}(\mathsf{pp_S})$, $(hk, td) \leftarrow \mathsf{H.KGen}(\mathsf{pp_H})$, and return $(vk_{\mathsf{S_{ST}}}, sk_{\mathsf{S_{ST}}}) := ((vk, hk), (sk, td, hk))$.
3. $\mathsf{S_{ST}.Sign}(sk_{\mathsf{S_{ST}}}, m)$.
   - Offline phase (no message given).
     (a) Choose random m', r', and compute $h := H_{hk}(m', r')$;
     (b) Invoke $\sigma \leftarrow \mathsf{S.Sign}(sk, h)$ and store $((m', r'), h, \sigma)$.
   - Online phase (given message m).
     (a) Retrieve $((m', r'), h, \sigma)$ from the memory;
     (b) Invoke $r \leftarrow \mathsf{H.TdColl}(td, m', r', m)$, return $\sigma_{\mathsf{S_{ST}}} := (\sigma, r)$.
4. $\mathsf{S_{ST}.Ver}(vk_{\mathsf{S_{GBSW}}}, m, \sigma_{\mathsf{S_{ST}}})$. Parse $\sigma_{\mathsf{S_{ST}}} = (\sigma, r)$ and return $\mathsf{S.Ver}(vk, H_{hk}(m, r), \sigma)$.

Theorem 7. *If* S *is MU-EUF-NCMA secure and* H *is MU-CR secure, then the online/offline signature scheme* $\mathsf{S_{ST}}$ *is MU-EUF-CMA secure. More precisely, for any PPT adversary* $\mathcal{A}$ *with advantage* $\mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}cma}}_{\mathsf{S_{ST}},\mu,\mathcal{A}}(\lambda)$*, there exist PPT adversaries* $\mathcal{B}_\mathsf{S}$ *and* $\mathcal{B}_\mathsf{H}$*, such that* $\mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}cma}}_{\mathsf{S_{ST}},\mu,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{mu\text{-}euf\text{-}ncma}}_{\mathsf{S},\mu,\mathcal{B}_\mathsf{S}}(\lambda) + \mathsf{Adv}^{\mathsf{mu\text{-}cr}}_{\mathsf{H},\mu,\mathcal{B}_\mathsf{H}}(\lambda)$.

Proof Sketch. The proof is similar to that in [27] and we give a sketch here. Consider the MU-EUF-CMA security experiment played between a challenger and an adversary $\mathcal{A}$. For every signing query (i, m) from $\mathcal{A}$, let (σ, r) be the return signature and $h := H_{hk_i}(m, r)$. Denote by $\mathcal{L}_i$ the set collecting all internal values (m, r, h) when answering the signing queries. Let $(i^*, m^*, \sigma^*_{\mathsf{ST}} = (\sigma^*, r^*))$ be $\mathcal{A}$'s output and Win be the event that $\mathcal{A}$ wins. Win happens iff $\mathsf{S.Ver}(vk_{i^*}, H_{hk_{i^*}}(m^*, r^*), \sigma^*) = 1$ and $m^* \notin \mathcal{Q}_{i^*}$. We divide it into two cases.

- **Case 1.** There exists $(m, r, h) \in \mathcal{L}_{i^*}$ s.t. $h = H_{hk_{i^*}}(m^*, r^*)$.
- **Case 2.** For all $(m, r, h) \in \mathcal{L}_{i^*}$, $h \neq H_{hk_{i^*}}(m^*, r^*)$.

Case 1 implies a collision of H, hence we can construct a PPT algorithm $\mathcal{B}_{\mathsf{H}}$ against H's MU-CR security.

Case 2 implies that $\mathcal{A}$ forges a valid signature σ^* for $H_{hk_{i^*}}(m^*, r^*)$ w.r.t. S, and $H_{hk_{i^*}}(m^*, r^*)$ is different from all values of h signed by $\mathsf{S.Sign}$, hence we can construct a PPT algorithm $\mathcal{B}_{\mathsf{S}}$ against S's MU-EUF-NCMA security. $\mathcal{B}_{\mathsf{S}}$ generates μ pairs (hk_i, td_i) via $\mathsf{H.KGen}$, and chooses Q (maximum number of signing queries) random (m', r') and computes their hash values as the messages to be signed. It sends those hash values to its own challenger and gets $\{vk_i\}_{i\in[\mu]}$ and Q signatures. Since $\mathcal{B}_{\mathsf{S}}$ has trapdoors of H, it can do the computation of online phase itself. So $\mathcal{B}_{\mathsf{S}}$ is able to answer signing queries and simulate the MU-EUF-CMA security experiment for $\mathcal{A}$ perfectly.

Since our building blocks H and S are tightly secure, and the security reduction is tightness preserving, the resulting online/offline signature S_{ST} is tightly MU-EUF-CMA secure. □

Tightly Secure Chameleon Signatures. Traditional signature schemes allow any party checking the validity of (m, σ). In some applications, m may contain sensitive information. Thus, it is desirable for the signer to generate a non-transferable signature to the recipient. To this end, Krawczyk and Rabin [22] introduced the concept of chameleon signature to achieve non-transferability, which prevents the recipient of the signature from disclosing the contents of the signed information to any third party without the signer's consent. They also proposed a construction of chameleon signature from the chameleon hash family, which is quite similar to online/offline signature. In the construction, the first phase (see the beginning of subsec. 5.2 for the two phases) was done by the original signer and the second phase was done by the recipient. The security of chameleon signature was proved in the single user setting in [22].

We can adapt the construction of chameleon signatures in [22] to the multi-user setting. There are μ users, $P_1, ..., P_\mu$, all parties share the same public parameter of signature scheme S and chameleon hash family H, and each party P_i generates a key pair (vk_i, sk_i) of S and a key pair (hk_i, td_i) of H. When P_i needs to sign m for P_j, it chooses random r, generates a signature σ for $H_{hk_j}(m, r)||P_j$[6], and sends (σ, r) to P_j. P_j can verify whether (σ, r) is valid by checking whether $\mathsf{S.Ver}(vk_i, H_{hk_j}(m, r)||P_j, \sigma) = 1$.

Analysis of Non-Transferability. It is infeasible for P_j to convince a third party the validity of $(m, (\sigma, r))$. This is because P_j has the trapdoor td_j, and it is easy for

[6] To prevent P_i denying σ later, the identity of P_j is added.

P_j to find a collision such that $H_{hk_j}(m,r) = H_{hk_j}(m',r')$. Hence, P_j may declare that $(m,(\sigma,r))$ is a valid message-signature pair, where σ is an outdated signature signed for some distinct message m'.

Besides non-transferability, chameleon signatures are required to have other properties [22].

- *Unforgeability.* No third party can produce a valid message-signature pair $(m,(\sigma,r))$ under vk_i and hk_j, except P_i and P_j (P_j can generate new valid $(m,(\sigma,r))$ pairs with σ generated by P_i previously). Similar to Theorem 7, the unforgeability in the multi-user setting can be tightly reduced to the MU-EUF-CMA security of S and MU-CR security of H.
- *Denial.* In case of dispute, if a judge presents a message-signature triple $(m,(\sigma,r))$ to P_i, then P_i is able to convince the judge to reject it. If $\mathsf{S.Ver}(vk_i, H_{hk_j}(m,r)||P_j,\sigma) \neq 1$, obviously, the judge will reject it. If $\mathsf{S.Ver}(vk_i, H_{hk_j}(m,r)||P_j,\sigma) = 1$ but P_i did not ever sign m, then according to unforgeability, P_i must have signed another message m' resulting in $(m',(\sigma,r'))$ from which P_j adapted $(m,(\sigma,r))$. In this case, P_j simply presents the judge the previous $(m',(\sigma,r'))$ with $m' \neq m$ for denial, and the judge will be convinced if $H_{hk_j}(m',r') = H_{hk_j}(m,r))$ and $m \neq m'$. The denial property is guaranteed by the MU-CR security of chameleon hash and the unforgeability of the chameleon signature.
- *Non-repudiation.* If P_i did sign $(m,(\sigma,r))$ previously, then it cannot convince a judge to reject it. In this case, P_i cannot repudiate $(m,(\sigma,r))$ if $\mathsf{S.Ver}(vk_i, H_{hk_j}(m,r)||P_j,\sigma) = 1$, since it cannot offer a hash collision under P_j's hash key hk_j. Similarly, the non-repudiation property can be reduced to the tight MU-CR security of H.
- *Exposure free.* P_i can deny a pair $(m,(\sigma,r))$ not produced by itself, without exposing any other message actually it signed. As discussed above, P_i can find a collision (m_1,r_1,m_2,r_2) under $H_{hk_j}(\cdot,\cdot)$, hence it can (partly) compute td_j and find arbitrary collisions (as shown below, all our CHFs in Section 3 achieve this property).

As a result, we have the following corollary.

Corollary 1. *Combining a tightly MU-CR secure chameleon hash family with a tightly MU-EUF-CMA secure signature scheme yields a tightly secure chameleon signature scheme in the multi-user setting.*

Analysis of Exposure Free Property. We show chameleon signatures from CHF_{dl}, CHF_{rsa} and CHF_{fac} achieve the exposure free property [22]. Note that if one can compute the trapdoor of CHF from a collision, then it can find arbitrary collisions, and the exposure free property follows.

Let us consider a specific key pair (hk,td).

In CHF_{dl}, a collision (m_1,r_1,m_2,r_2) implies that $g^{xm_1+r_1} = g^{xm_2+r_2}$, equivalently $xm_1+r_1 = xm_2+r_2$. If $m_1 = m_2$, then $r_1 = r_2$. So we must have $m_1 \neq m_2$, and one can compute the trapdoor $x := (r_2-r_1)/(m_1-m_2)$.

In CHF_{rsa}, a collision (m_1,r_1,m_2,r_2) implies that $(x^{m_1}r_1)^e = (x^{m_2}r_2)^e$, equivalently $x^{m_1-m_2} = r_2/r_1$. We must have $m_1 \neq m_2$ (otherwise $r_1 = r_2$). Let α,β be two integers such that $\alpha e + \beta(m_1-m_2) = 1$ (α and β can always be found since e is a prime and $e > 2^{L(\lambda)}$), then one can compute the trapdoor $x := (r_2/r_1)^\beta \cdot X^\alpha$.

The analysis of CHF_{fac} is more complex. Let (m_1, r_1, m_2, r_2) be a collision such that $\prod_{k=1}^{\ell}(u_k)^{m_{1,k}} \cdot r_1^2 = \prod_{k=1}^{\ell}(u_k)^{m_{2,k}} \cdot r_2^2$. We divide it in two cases.

Case 1. $m_1 = m_2$ but $r_1 \neq r_2$. In this case, $r_1^2 \equiv r_2^2 \mod N$, equivalently $(r_1 + r_2)(r_1 - r_2) = 0$. Since $r_1, r_2 \in \mathbb{Z}_N^+$ and $r_1 \neq r_2$, anyone can find a factor of N by computing $\gcd(r_1 + r_2, N)$, and then find the (possible) trapdoors[7]. These trapdoors are sufficient to find new collisions.

Case 2. $m_1 \neq m_2$. Define the set $\mathcal{S} := \{k \mid m_{1,k} \neq m_{2,k}\}$. Thus we have $(\prod_{k\in\mathcal{S}} s_k^{m_{1,k}-m_{2,k}})^2 = (r_2/r_1)^2$. We stress that this equation is sufficient to achieve the exposure free property in chameleon signatures. One can choose random $\tilde{m}_1$, $\tilde{m}_2$ such that $\{k \mid \tilde{m}_{1,k} \neq \tilde{m}_{2,k}\} = \mathcal{S}$, and $(\tilde{m}_1, r_1, \tilde{m}_2, r_2)$ is a collision too.

Tightly Secure Proxy Signatures. Proxy signature, proposed by Mambo, Usuda and Okamoto [23], is a variant of signature scheme, where an original signer can delegate its signing ability to a proxy (usually by a warrant), and enable the proxy to sign messages on behalf of it. Many proxy signatures are based on chameleon hash functions [24, 12], where the warrant is a signature for the chameleon hash value $H_{hk}(m', r')$.

Chameleon signature naturally implies a proxy signature[8]. Suppose a proxy signature system containing μ users, where all members share the same public parameter of signature scheme S and chameleon hash family H. P_i generates its own key pair (vk_i, sk_i) of S and (hk_i, td_i) of H. Any party P_i can adaptively delegate its signing ability to a proxy P_j as follows: P_j computes $h := H_{hk_j}(m', r')$ for random (m', r'), and sends h to P_i. Then P_i generates a signature σ for h as a warrant and sends it back to P_j. Whenever there is a real message m to sign for P_j, it uses its trapdoor td_j to find a collision randomness r such that $H_{hk_j}(m, r) = H_{hk_j}(m', r')$. Finally, the proxy signature is of the form (σ, r).

If H is CR secure, we can prove the security of the proxy signature system above, but lose a factor μ (given EUF-CMA secure SIG S) in the security reduction. However, with the help of tightly MU-CR secure H, we can get a tightly secure proxy signature system as long as the underlying building block S is tightly MU-EUF-CMA secure. The analysis is similar to chameleon signatures, and we have the following corollary.

Corollary 2. *Combining a tightly MU-CR secure chameleon hash family with a tightly MU-EUF-CMA secure signature scheme yields a tightly secure proxy signature scheme in the multi-user setting.*

Acknowledgments. This work is supported by National Natural Science Foundation of China (No. 61925207), Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), and Major Project of the Ministry of Industry and Information Technology of China (2018-36).

References

[1] Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A practical and provably secure coalition-resistant group signature scheme. In: Advances in Cryptology - CRYPTO 2000, 20th Annual

[7] For a hash key u_k, there exist four possible s_k such that $u_k = s_k^2$.

[8] The detailed security requirements of proxy signature depend on the needs of the scenarios. We consider one-time proxy signatures [24] here.

International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings. pp. 255–270 (2000), https://doi.org/10.1007/3-540-44598-6_16
[2] Ateniese, G., de Medeiros, B.: Identity-based chameleon hash and applications. In: Financial Cryptography, 8th International Conference, FC 2004, Key West, FL, USA, February 9-12, 2004. Revised Papers. pp. 164–180 (2004), https://doi.org/10.1007/978-3-540-27809-2_19
[3] Ateniese, G., de Medeiros, B.: On the key exposure problem in chameleon hashes. In: Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers. pp. 165–179 (2004), https://doi.org/10.1007/978-3-540-30598-9_12
[4] Bader, C.: Efficient signatures with tight real world security in the random-oracle model. In: Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings. pp. 370–383 (2014), https://doi.org/10.1007/978-3-319-12280-9_24
[5] Bader, C., Hofheinz, D., Jager, T., Kiltz, E., Li, Y.: Tightly-secure authenticated key exchange. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I. pp. 629–658 (2015), https://doi.org/10.1007/978-3-662-46494-6_26
[6] Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. In: Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings. pp. 268–286 (2004), https://doi.org/10.1007/978-3-540-24676-3_17
[7] Bellare, M., Ristov, T.: A characterization of chameleon hash functions and new, efficient designs. J. Cryptology 27(4), 799–823 (2014), https://doi.org/10.1007/s00145-013-9155-8
[8] Blazy, O., Kakvi, S.A., Kiltz, E., Pan, J.: Tightly-secure signatures from chameleon hash functions. In: Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings. pp. 256–279 (2015), https://doi.org/10.1007/978-3-662-46447-2_12
[9] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. pp. 41–55 (2004), https://doi.org/10.1007/978-3-540-28628-8_3
[10] Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational diffie-hellman. In: Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, April 24-26, 2006, Proceedings. pp. 229–240 (2006), https://doi.org/10.1007/11745853_15
[11] 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: Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part II. pp. 152–182 (2017), https://doi.org/10.1007/978-3-662-54388-7_6
[12] Chandrasekhar, S., Chakrabarti, S., Singhal, M., Calvert, K.L.: Efficient proxy signatures based on trapdoor hash functions. IET Information Security 4(4), 322–332 (2010), https://doi.org/10.1049/iet-ifs.2009.0204
[13] Contini, S., Lenstra, A.K., Steinfeld, R.: Vsh, an efficient and provable collision-resistant hash function. In: Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings. pp. 165–182 (2006), https://doi.org/10.1007/11761679_11
[14] Damgård, I.: Collision free hash functions and public key signature schemes. In: Advances in Cryptology - EUROCRYPT '87, Workshop on the Theory and Application of of Cryptographic Techniques, Amsterdam, The Netherlands, April 13-15, 1987, Proceedings. pp. 203–216 (1987), https://doi.org/10.1007/3-540-39118-5_19
[15] Diemert, D., Gellert, K., Jager, T., Lyu, L.: More efficient digital signatures with tight multi-user security. In: Garay, J.A. (ed.) Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12711, pp. 1–31. Springer (2021), https://doi.org/10.1007/978-3-030-75248-4_1

[16] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. pp. 186–194 (1986), https://doi.org/10.1007/3-540-47721-7_12

[17] Gjøsteen, K., Jager, T.: Practical and tightly-secure digital signatures and authenticated key exchange. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. pp. 95–125 (2018), https://doi.org/10.1007/978-3-319-96881-0_4

[18] Han, S., Jager, T., Kiltz, E., Liu, S., Pan, J., Riepel, D., Schäge, S.: Authenticated key exchange and signatures with tight security in the standard model. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 12828, pp. 670–700. Springer (2021), https://doi.org/10.1007/978-3-030-84259-8_23

[19] Harn, L., Hsin, W., Lin, C.: Efficient on-line/off-line signature schemes based on multiple-collision trapdoor hash families. Comput. J. 53(9), 1478–1484 (2010), https://doi.org/10.1093/comjnl/bxp044

[20] Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. Des. Codes Cryptogr. 80(1), 29–61 (2016), https://doi.org/10.1007/s10623-015-0062-x

[21] Khalili, M., Dakhilalian, M., Susilo, W.: Efficient chameleon hash functions in the enhanced collision resistant model. Inf. Sci. 510, 155–164 (2020), https://doi.org/10.1016/j.ins.2019.09.001

[22] Krawczyk, H., Rabin, T.: Chameleon hashing and signatures. IACR Cryptology ePrint Archive 1998, 10 (1998), http://eprint.iacr.org/1998/010

[23] Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: CCS '96, Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, March 14-16, 1996. pp. 48–57 (1996), https://doi.org/10.1145/238168.238185

[24] Mehta, M., Harn, L.: Efficient one-time proxy signatures. IEE Proceedings - Communications 152(2), 129–133 (April 2005)

[25] Micali, S., Shamir, A.: An improvement of the fiat-shamir identification and signature scheme. In: Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings. pp. 244–247 (1988), https://doi.org/10.1007/0-387-34799-2_18

[26] Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. pp. 31–53 (1992), https://doi.org/10.1007/3-540-48071-4_3

[27] Shamir, A., Tauman, Y.: Improved online/offline signature schemes. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. pp. 355–367 (2001), https://doi.org/10.1007/3-540-44647-8_21

[28] Steinfeld, R., Pieprzyk, J., Wang, H.: How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In: Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings. pp. 357–371 (2007), https://doi.org/10.1007/11967668_23

[29] Zhang, X., Liu, S., Gu, D., Liu, J.K.: A generic construction of tightly secure signatures in the multi-user setting. Theor. Comput. Sci. 775, 32–52 (2019), https://doi.org/10.1016/j.tcs.2018.12.012

[30] Zhang, X., Liu, S., Pan, J., Gu, D.: Tightly secure signature schemes from the LWE and subset sum assumptions. Theor. Comput. Sci. 795, 326–344 (2019), https://doi.org/10.1016/j.tcs.2019.07.015