Signatures without Random Oracles

Anonymous Submission

No Institute Given

1 Introduction

Digital Signatures are one of the most important primitives in cryptography. They are used as building blocks in many high level protocols. The security of signatures can be proven in Eeither the Standard Model or the Random Oracle Model.Signatures in the Standard Model are usually less efficient and practical than the ones in the Random Oracle Model.

One of the ways of obtaining an signature is via the Fiat-Shamir transform. In this transform you use an identity scheme and hash the commitment so that the resulting signature is secure in the random oracle model. The problem is that schemes that are secure in the Random Oracle model are not necessarily secure in the real world, therefore it is valuable to have signatures that are secure in the Standard Model. A technique known as OR-Proofs are sometimes used to prove the tight security in this model.

Although there discrete logarithm based signatures in the standard model, most use pairings and there are relatively few pairing free ones. Correlation Interactable Hash functions have been recently used to achieve signatures in the Standard Model.

Correlation Interactable hash functions first introduced by Canetti et al. are hash functions in which it is difficult to find an input that has a fixed non-common relationship with the output of that hash function. A Random Oracle clearly fits into this definition. Unlike Random Oracles, Correlation Interactable hash functions actually exist in the real world.

Our paper offers a generalized construction that uses a transform similar to Fiat-Shamir while borrowing ideas from the OR-proof techniques but doesn't rely on the hash function being a Random Oracle for security. Our scheme instead relies on Correlation Interactable hash functions.

1.1 Contributions

We first introduce a Dual Tag Identification Scheme in Section 3. This Tag Based Identification Sheme has an additional verification called the Alternative Verification which unlike the "normal" verification uses the secret key to verify if a transcript could have been created by an honest prover. Then we introduce an unforgeability notion called ??. This unforgeability notion implies that an adversary cannot come up with a new transcript that verifies with the alternative verification after seeing instances of the transcript. It is important to note that

this Identification Scheme is not Honest Verifier Zero Knowledge in opposed to many common schemes.

In Section 4, we introduce our signature scheme which runs the identification scheme two times and aggregates their commitments similar to an OR proof. We then go on to show that if the underlying Identification Scheme has the desirable properties then the resulting Signature Scheme is secure under the Random Message Attack which can be extended to be secure against the Chosen Message Attack using Chameleon Hashing.

In Section 5 and 6 we show an instantiations of the mentioned Identification Scheme based on the q-SDH and q-DH assumptions respectively. The construction is Section 5 is based on an extension of the EDL Scheme shown by Chevallier-Mames and the construction in the Section 6 is an extension of section 5 using programmable hash functions introduces by Kilts et al. to adapt a well known lemma.

2 Preliminaries

2.1 Signatures

A digital signature scheme consists of three algorithms $(\mathsf{Gen}, \mathsf{Sgn}, \mathsf{Ver})$. A key generation algorithm Gen, a signing algorithm Sgn and a verification algorithm Ver. Gen is a randomised algorithm that produces a random key pair consisting of a public key pk and a secret key sk. The probabilistic signing algorithm Sgn requires a secret key and a message from the message space M and produces a signature σ. Finally, the verification algorithm Ver takes a public key, a message and a signature as input and returns either 0/reject or 1/accept. A signature scheme is called correct, if every signature on a message generated with a secret key is accepted under the corresponding public key.

Definition 1 (RMA security). *A signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sgn}, \mathsf{Ver})$ is said to be (t, q, ϵ)-secure against existential forgery under the random message attack (EUF-RMA), if for all adversary $\mathcal{A}$ running in time at most t we have*

$$\Pr[q\text{-}\mathsf{EUF\text{-}RMA}_{\Pi}(\mathcal{A})] = 1] \leq \epsilon$$

where q-$\mathsf{EUF\text{-}RMA}_{\Pi}$ is depicted in Figure 1.

We say $\mathcal{A}$, (t, q, ϵ)-breaks the EUF-RMA security of the signature if

$$\Pr[q\text{-}\mathsf{EUF\text{-}RMA}_{\Pi}(\mathcal{A}) = 1] > \epsilon$$

2.2 Hash Functions

[**Aysan:** Fix hash function dfinition.] Let $\mathbb{G} = (\mathsf{G}_k)$ be a family of groups, indexed by the security parameter $k \in \mathbb{N}$ (we omit the subscript when the reference to the security parameter is clear, thus write G for G_k).

Game q-EUF-RMA$_{\Pi}(\mathcal{A})$

$(sk, pk) \xleftarrow{\$} \mathsf{Gen}$
$\mathcal{Q} \leftarrow \emptyset$
for $i \in [q]$
  $m_i \xleftarrow{\$} \mathsf{M}$
  $\sigma_i \xleftarrow{\$} \mathsf{Sgn}(sk, m_i)$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup (m_i, \sigma_i)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}(pk, \mathcal{Q})$
if $\mathsf{Ver}(pk, m^*, \sigma^*) = 1 \wedge m^* \notin \{m_1, ..., m_q\}$ **then return** 1
else return 0

Figure 1.

A *group hash function* H over G with input length $l = l(k)$ consists of two efficient algorithms $\mathsf{PHF.Gen}$ and $\mathsf{PHF.Eval}$. The probabilistic algorithm $\kappa \xleftarrow{\$} \mathsf{PHF.Gen}(1^k)$ generates a hash key κ for the security parameter k. Algorithm $\mathsf{PHF.Eval}$ is a deterministic algorithm, taking as input a hash function key κ and $X \in \{0,1\}^l$, and returning $\mathsf{PHF.Eval}(\kappa, X) \in \mathsf{G}$. In the context were κ is clear we write $\mathsf{PHF.Eval}(\kappa, X)$ as $\mathsf{H}(X)$.

Definition 2 (Correlation Interactibilty). *We say an adversary $\mathcal{A}$, $(t, \epsilon)-$breaks the correlation intractability of a hash function $\mathsf{H} = (\mathsf{PHF.Gen}, \mathsf{PHF.Eval})$ with regards to function g if $\mathcal{A}$ runs in time t and*

$$\Pr[x \xleftarrow{\$} \mathcal{A}, \mathsf{PHF.Eval}(\kappa, x) = g(x); \kappa \xleftarrow{\$} \mathsf{PHF.Gen}] \geq \epsilon.$$

We call the hash function $(t, \epsilon)-$correlation intractable if such an adversary does not exist.

Definition 3. *A group hash function $H = (\mathsf{PHF.Gen}, \mathsf{PHF.Eval})$ is a $(m, n, n\gamma, \delta)-$programmable, if there is an efficient trapdoor key generation algorithm $\mathsf{PHF.TrapGen}$ and an efficient trapdoor evaluation algorithm $\mathsf{PHF.TrapEval}$ with the following properties.*

1. *The probabilistic trapdoor generation algorithm $(\kappa, \eta) \xleftarrow{\$} \mathsf{PHF.TrapGen}(1^k, g_1, g_2)$ takes as input group elements $g, h \in \mathsf{G}$, and produces a hash function key κ together with trapdoor information η.*
2. *For all generators $g_1, g_2 \in \mathsf{G}$, the keys $\kappa \xleftarrow{\$} \mathsf{PHF.Gen}(1^k)$ and $\kappa' \xleftarrow{\$} \mathsf{PHF.Gen}(1^k, g_1, g_2)$ are statistically γ-close.*
3. *On input $X \in \{0,1\}^l$ and trapdoor information η, the deterministic trapdoor evaluation algorithm $(a_X, b_X) \leftarrow \mathsf{PHF.TrapEval}(\eta, X)$ produces $a_X, b_X \in \mathbb{Z}$ so that for all $Xin\{0,1\}^l$,*

$$\mathsf{PHF.Eval}(\kappa, X) = g_1^{a_X} g_2^{b_X}.$$

4. *For all* $g_1, g_2 \in \mathsf{G}$*, all* κ *generated by* $\kappa \xleftarrow{\$} \mathsf{PHF.TrapGen}(1^k, g_1, g_2)$*, and all* $X_1, .., X_m$ *in*$\{0,1\}^l$ *and* $Z_1, ..., Z_n \in \{0,1\}^l$ *such that* $X_i \neq Z_j$ *for all* i, j*, we have*

$$\Pr[a_{X_1} = ... = a_{X_m} = 0 \wedge a_{Z_1}, ..., a_{Z_n} \neq 0] \geq \delta$$

where $(a_{X_i}, b_{X_i}) = \mathsf{PHF.TrapEval}(\eta, X_i)$ *and* $(a_{Z_i}, b_{Z_i}) = \mathsf{PHF.TrapGen}(\eta, Z_j)$*, and the probability is taken over the trapdoor* η *produced along with* κ*.*

3 Identification Scheme

Definition 4 (Canonical Tag-based Identification Scheme). *A canonical tag-based identification (tag-ID) scheme is defined as the probabilistic algotithms* $\mathsf{ID} := (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ *where*

- IGen *returns a public key and secret key* (pk, sk)*. We assume that* pk *defines the challenge set* ChSet *and tag space* TgSet*.*
- *The prover algorithm* $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ *is split into two algorithms.* P_1 *takes the secret key* sk *and a tag* τ *from the tag space* M *as the input and returns a commitment* Com *and a state* St*.* P_2 *takes the secret key* sk*, the state* St *and a challenge* C *as an input and returns a response* s*.*
- *The deterministic verifier algorithm* V *takes the public key* pk*, the tag* τ*, the commitment* Com*, the challenge* C *and the response* s *as an input and outputs a decision, 1 (acceptance) or 0 (rejection).*

For correctness we require that for all $k \in \mathbb{N}$*,* $(pk, sk) \in \mathsf{IGen}(1^k)$*, all* $(Com, St) \in \mathsf{P}_1(sk, \tau)$*, all* $C \in \mathsf{ChSet}$ *and all* $s \in \mathsf{P}_2(sk, St, C)$*, we have*

$$\mathsf{V}(pk, Com, C, s) = 1.$$

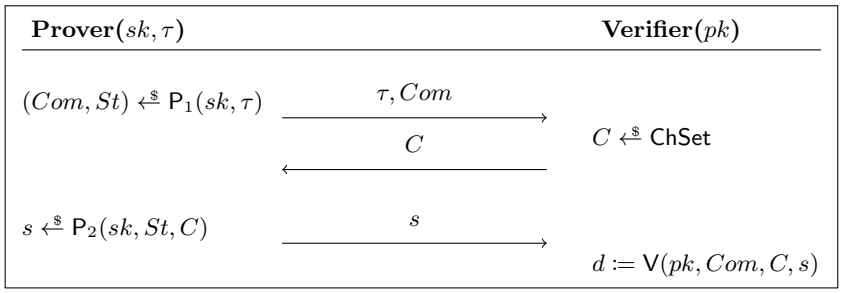

Figure 2. Canonical Tag-based Identification Scheme

Definition 5 (Dual Tag-ID). *A dual canonical tag based identification scheme (dual tag-ID) is a identification scheme* ID*, with an additional algorithm* $\tilde{\mathsf{V}}$ *called*

the alternative verification algorithm that takes the secret key sk, the tag τ, the commitment Com, the challenge C and the response s as an input and outputs a decision, 1 (acceptance) or 0 (rejection).
For the correctness of this scheme in additional to the correctness defined before we require that for all $k \in \mathbb{N}$, $(pk, sk) \in \mathsf{IGen}(1^k)$, all $(Com, St) \in \mathsf{P}_1(sk, \tau)$, all $C \in \mathsf{ChSet}$ and all $s \in \mathsf{P}_2(sk, St, C)$, we have

$$\tilde{\mathsf{V}}(sk, \tau, Com, C, s) = 1.$$

Definition 6 (Existential Unforgeablilty against Passive Attacks). *A dual tag-ID scheme is said to be (t, q, ϵ)-$\mathsf{UF\text{-}PA}$ secure, if for all adversary $\mathcal{A}$ running in time at most t we have*

$$\Pr[q\text{-}\mathsf{PA}_{\mathsf{ID}}(\mathcal{A})] = 1] \leq \epsilon.$$

Unlike most commonly used identification schemes the canonical tag based ID schemes we use are not Honest Verifier Zero Knowledge (HVZK) and instead have some different soundness property.

Game q-$\mathsf{PA}_{\mathsf{ID}}(\mathcal{A})$

$(sk, pk) \xleftarrow{\$} \mathsf{IGen}$
$\mathcal{Q} \leftarrow \emptyset$
for $i \in [q]$
  $\tau_i \xleftarrow{\$} \mathsf{M}$
  $(Com_i, St_i) \xleftarrow{\$} \mathsf{P}_1(sk, \tau_i)$
  $C_i \xleftarrow{\$} \mathsf{ChSet}$
  $s_i \xleftarrow{\$} \mathsf{P}_2(sk, St_i, C_i)$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup (\tau_i, Com_i, C_i, s_i)$
$(\tau^*, Com^*, C^*, s^*) \leftarrow \mathcal{A}(pk, \mathcal{Q})$
if $\tau^* \notin \{\tau_1, ..., \tau_q\} \wedge \tilde{\mathsf{V}}(sk, \tau^*, Com^*, C^*, s^*) = 1$
 then return 1
else return 0

Figure 3.

Definition 7 (Uniqueness). *We say the identification scheme $\mathsf{ID} := (\mathsf{IGen}, \mathsf{P}, \mathsf{ChSet}, \mathsf{V})$ is unique if for every $(sk, pk) \in \mathsf{IGen}$ and every $(Com, St) \in \mathsf{P}_1(sk, \tau)$,*

$$\left|\{C \in \mathsf{ChSet} \mid \exists\, s : \mathsf{V}(pk, Com, C, s) = 1 \wedge \tilde{\mathsf{V}}(sk, Com, \mathcal{C}, s) \neq 1\}\right| = 1.$$

This means there exist a (not necessarily polynomial time) function we call the uniqueness function such as f that

$$f(pk, Com) = C.$$

4 Constructions

To construct a signature $\mathsf{Sig}[\mathsf{ID}, \mathsf{H}] := (\mathsf{Gen}, \mathsf{Sgn}, \mathsf{Ver})$ from a cannonical tag-based identification scheme $\mathsf{ID} := (\mathsf{IGen}, \mathsf{P}, \mathsf{V})$ we proceed as in Figure 4.

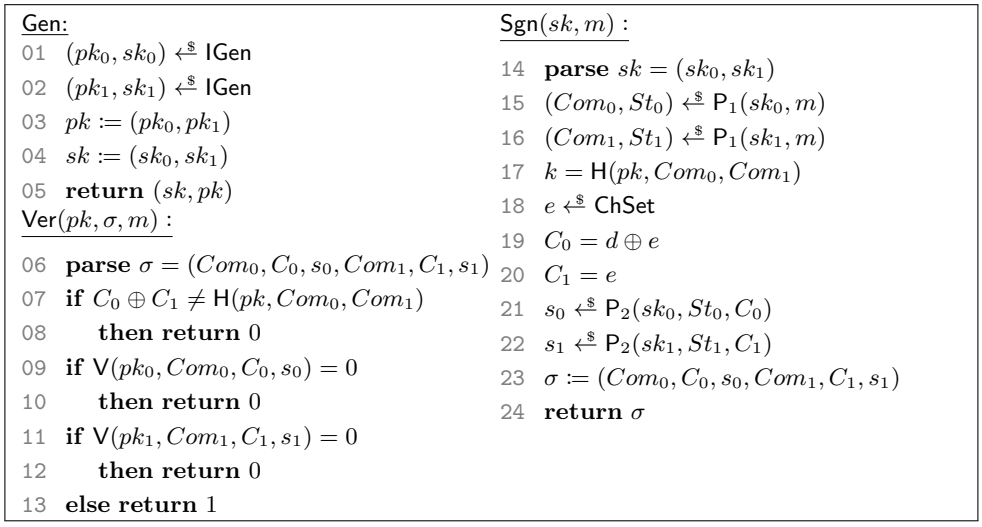

Figure 4. Instantiation 1

4.1 Security

[**Aysan:** Write what we do in this section.]

We say the signature $\sigma = (Com_0, C_0, s_0, Com_1, C_1, s_1)$ is partially valid if $\tilde{\mathsf{V}}(pk_0, Com_0, C_0, s_0) = 1$ or $\tilde{\mathsf{V}}(pk_1, Com_1, C_1, s_1) = 1$ not partially valid if $\tilde{\mathsf{V}}(pk_0, Com_0, C_0, s_0) = 0$ and $\tilde{\mathsf{V}}(pk_1, Com_1, C_1, s_1) = 0$.

Theorem 1. *Let* ID *be a unique dual tag-based ID scheme with the uniqueness function* f *and* H *be a* (t'', ϵ'') *correlation intractable hash function with regards to function* g *where*

$$g :=$$

Suppose there exists a (t, q, ϵ)*-forger* $\mathcal{F}$ *breaking the security of* $\mathsf{Sig}_{\mathsf{ID},\mathsf{H}}$ *against the existential forgery under the random message attack. Then there exists an adversary that* (t', q, ϵ')*-breaks the* $\mathsf{UF\text{-}PA}^{\tilde{\mathsf{V}}}$ *security of* ID *with* $t' \approx t$ *and*

$$\epsilon \leq \epsilon'' + 2\epsilon'$$

Proof. We define the event of Game G_i winning (returning 1) as X_i. Let (m_i, σ_i) denote the i-th random message and its signature. Let (m^*, σ^*) be the forgery output by $\mathcal{F}$.

$\underline{\mathsf{G}_0 - \mathsf{G}_4}$

$b \xleftarrow{\$} \{0,1\}$ // $\mathsf{G}_2 - \mathsf{G}_3$
$\mathtt{BAD}_2 \leftarrow \mathbf{true}$ // $\mathsf{G}_2 - \mathsf{G}_3$
$(pk_0, sk_0) \xleftarrow{\$} \mathsf{IGen}$
$(pk_1, sk_1) \xleftarrow{\$} \mathsf{IGen}$
$pk \coloneqq (pk_0, pk_1)$
$sk \coloneqq (sk_0, sk_1)$
for $i \in [q]$
  $m_i \xleftarrow{\$} \mathsf{M}$
  $(Com_{0,i}, St_{0,i}) \xleftarrow{\$} \mathsf{P}_1(sk_0, m_i)$
  $(Com_{1,i}, St_{1,i}) \xleftarrow{\$} \mathsf{P}_1(sk_1, m_i)$
  $k_i = \mathsf{H}(pk, Com_{0,i}, Com_{1,i})$
  $e_i \xleftarrow{\$} \mathsf{ChSet}$
  $C_{0,i} = k_i \oplus e_i$ // $\mathsf{G}_0 - \mathsf{G}_3$
  $C_{1,i} = e_i$ // $\mathsf{G}_0 - \mathsf{G}_3$
  $C_{b,i} = e_i$ // G_4
  $C_{1-b,i} = k_i \oplus e_i$ // G_4
  $s_{0,i} \xleftarrow{\$} \mathsf{P}_2(sk_0, St_{0,i}, C_{0,i})$
  $s_{1,i} \xleftarrow{\$} \mathsf{P}_2(sk_1, St_{1,i}, C_{1,i})$
  $\sigma_i \coloneqq (Com_{0,i}, C_{0,i}, s_{0,i}, Com_{1,i}, C_{1,i}, s_{1,i})$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup (m_i, \sigma_i)$
$(m*, \sigma^*) \leftarrow \mathcal{A}(pk, \mathcal{Q})$
parse $\sigma^* = (Com_0^*, C_0^*, s_0^*, Com_1^*, C_1^*, s_1^*)$
if $\tilde{\mathsf{V}}(pk_0, Com_0^*, C_0^*, s_0^*) = 0 \wedge \tilde{\mathsf{V}}(pk_0, Com_0^*, C_0^*, s_0^*) = 0$ // $\mathsf{G}_1 - \mathsf{G}_3$
 then $\mathtt{BAD}_1 \leftarrow \mathbf{true}$; abort // $\mathsf{G}_1 - \mathsf{G}_3$
if $\tilde{\mathsf{V}}(pk_b, Com_b^*, C_b^*, s_b^*) = 0$ // $\mathsf{G}_2 - \mathsf{G}_3$
 then $\mathtt{BAD}_2 \leftarrow \mathbf{true}$; // $\mathsf{G}_2 - \mathsf{G}_2$
  abort // G_3
if $C_0^* + C_1^* \neq \mathsf{H}(pk, Com_0^*, Com_1^*)$
  then return 0
if $\mathsf{V}(pk_0, Com_0, C_0, s_0) = 0 \vee \mathsf{V}(pk_1, Com_1, C_1, s_1) = 0$
  then return 0
if $m^* \notin \{m_1, .., m_q\}$
  then return 0
else return 1

Figure 5.

Game 0. We define Game 0 as the existential unforgeability experiment with forger $\mathcal{F}$ on the signature scheme$\mathsf{Sig}_{\mathsf{ID},\mathsf{H}}$ as shown in Figure 5. By definition, we have

$$\Pr[X_0] = \epsilon.$$

Game 1. In G_1 we check if the signature is partially valid or not and set $\mathtt{BAD}_1$ to **true** and abort if it isn't. Which according to Lemma 1 and H being (t'', ϵ'') correlation intractable happens with at most ϵ'' probability and so we have

$$| \Pr[X_1] - \Pr[X_0] | \leq \epsilon''.$$

Game 2. In G_2 we pick a random bit b in the beginning of the game and after getting the forged signature σ^* which we parse as

$$\sigma^* = (Com_0^*, C_0^*, s_0^*, Com_1^*, C_1^*, s_1^*),$$

we check whether $\tilde{\mathsf{V}}(pk_b, Com_b^*, C_b^*, s_b^*)$ is zero and set the tag $\mathtt{BAD}_2$ to **true** if it is. Since this change is only internal to the game

$$\Pr[X_1] = \Pr[X_2].$$

Game 3. In G_3 we abort if $\mathtt{BAD}_2$ that we defined in the last game is set to **true**. Since the game would have already aborted if the forged signature was not partially valid signature and b was chosen randomly in the beginning, we have $\Pr[\mathtt{BAD}_2] \leq \frac{1}{2}$, which implies

$$\Pr[X_3] = \Pr[X_2 \wedge \neg \mathtt{BAD}_2] \geq \frac{1}{2} \Pr[X_2].$$

Game 4. Game G_4 is exactly like G_3 except instead of always choosing $C_{0,i}$ randomly from the ChSet and then calculating $C_{1,i}$ accordingly, we choose $C_{b,i}$ first and then calculate $C_{1-b,i}$. Since the distribution of $(C_{0,i}, C1, i)$ does not change we have

$$\Pr[X_4] = \Pr[X_3].$$

We point out that in this game we can choose $(m_i, Com_{b,i}, C_{i,b}, s_{b,i})$ first and then calculate $(Com_{1-b,i}, C_{1-b,i}, s_{1-b,i})$ and thus the signature σ_i accordingly.

Now adversary $\mathcal{A}$ simulates game G_4. The $\mathcal{A}$ receives pk_b and $(\tau_i, Com_{b,i}, C_{b,i}, s_{b,i})$ from the alternative impersonation game and proceeds to run IGen to obtain pk_{1-b} and calculate signatures on message $m_i := \tau_i$. As pointed out before it is possible to calculate σ_i according to $(\tau_i, Com_{b,i}, C_{b,i}, s_{b,i})$.
It remains to show how $\mathcal{A}$ can break the alternative impersonation from the forged signature $\sigma^* = (Com_0^*, C_0^*, s_0^*, Com_1^*, C_1^*, s_1^*)$ on message m^* output by $\mathcal{F}$. We know that $\tilde{\mathsf{V}}(sk_b, m^*, Com_b^*, C_b^*, s_b^*) = 1$ (by game 2). So $\mathcal{A}$ can win the alternative impersonation game by outputting $(m^*, Com_b^*, C_b^*, s_b^*)$. ■

Lemma 1. *Let $\mathcal{F}$ be a forger that (t, q, ϵ)-breaks the RMA security of the signature such that the forged signature it outputs is not partially valid. Then there exists adversary $\mathcal{A}$ that (t'', ϵ'')-breaks the correlation intractability of the hash function H with $t \approx t''$ and*

$$\epsilon'' \geq \epsilon.$$

Proof. The correlation intractability adversary $\mathcal{A}$ runs the unforgeability experiment by running IGen twice and obtaining two pairs of keys we name (sk_0, pk_0) and (sk_1, pk_1). The adversary now return $pk := (pk_0, pk_1)$ to $\mathcal{F}$ as the public key and also chooses random messages $m_1, ..., m_q$ and signs them with the secret key $sk := (sk_0, sk_1)$ to obtain the signatures $\sigma_1, ..., \sigma_q$ and returns the (m_i, σ_i) pairs to $\mathcal{F}$.
Eventually, $\mathcal{F}$ returns a message and signature pair (m^*, σ^*), from which $\mathcal{A}$ extracts the solution that breaks the hash intractability as follows.
First $\mathcal{A}$ parses σ^* as $(Com_0^*, C_0^*, s_0^*, Com_1^*, C_1^*, s_1^*)$. We assume that the forged signature is valid and since it is not partially valid, due to the uniqueness of the identification scheme we can write

$$C_0^* = f(pk_0, Com_0^*)$$
$$C_1^* = f(pk_1, Com_1^*).$$

Since we have assumed the forged signature is valid

$$\mathsf{H}(pk, Com_0^*, Com_1^*) = C_0^* + C_1^* = f(pk_0, Com_0^*) + f(pk_1, Com_1^*) = g(pk, Com_0^*, Com_1^*)$$

must hold. So $\mathcal{A}$ can (t, ϵ') break the g-correlation intractability of H where g is defined as

$$g(pk = (pk_0, pk_1), Com_0, Com_1) = f(pk_0, Com_0) + f(pk_1, Com_1).$$

Adversary $\mathcal{A}$ succeeds at giving a solution that breaks the correlation intractability of H whenever $\mathcal{F}$ succeeds at forging a valid signature so

$$\epsilon'' \geq \epsilon.$$

■

5 Instantiation of dual tag ID from q-SDH

Throughout thus section let $\mathsf{par} := (p, \mathsf{G})$ be a set of system parameters, where G is a cyclic group of prime order p.

Definition 8 (q-SDH Assumption). *We say an adversary $\mathcal{A}$ breaks the q-strong Diffie Hellman (q-SDH)assumption if it's running time is bounded by t and*

$$\Pr[(s, g^{\frac{1}{s+x}}) \xleftarrow{\$} \mathcal{A}(g, g^x, ..., g^{x^q})] \geq \epsilon,$$

where $g \xleftarrow{\$} \mathsf{G}$ and $x \xleftarrow{\$} \mathbb{Z}_p^*$.

Lemma 2. *Let f be a polynomial*

$$f(X) = \prod_{i=1}^{i=q} (X + \tau_i)$$

for some $\tau_i \in \mathbb{Z}_p$. Given $\{g^{x_i}\}_{i=0,...,q}$, let us define $g_1 = g^{\theta f(x)}$. For any τ_i where $i \in [1, q]$ it is easy to compute $g_1^{\frac{1}{x+\tau_i}}$ and given $g_1^{\frac{1}{x+\tau}}$ where $\tau \notin \{\tau_1, ..., \tau_q\}$, one can easily compute $g^{\frac{1}{x+\tau}}$.

Proof. Reference. ■

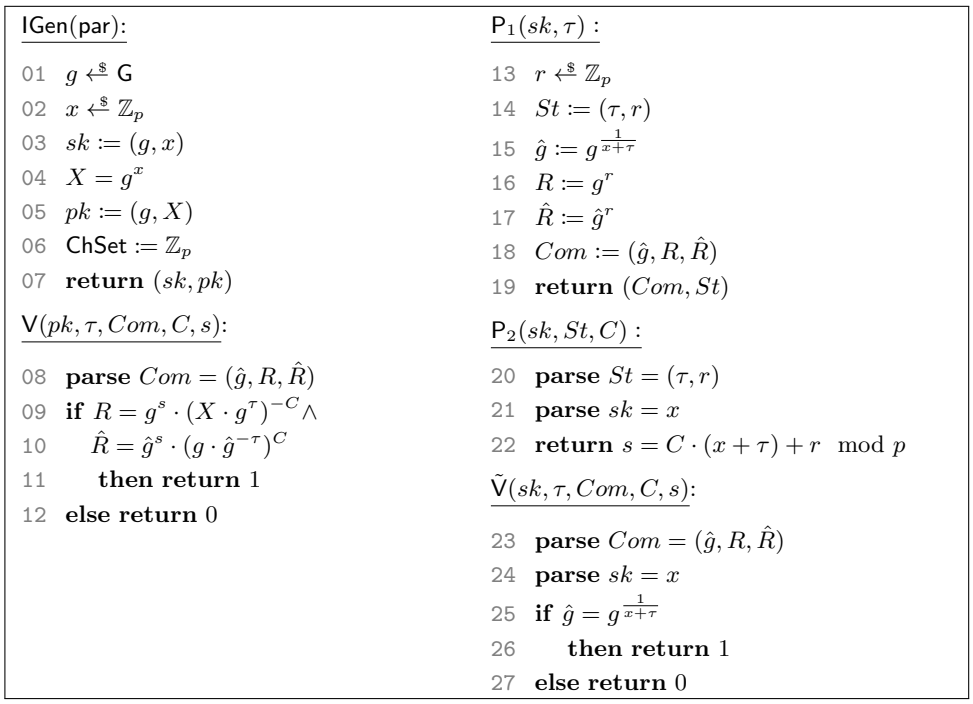

Figure 6. Instantiation 1

We describe the identification scheme $\mathsf{ID}_{q-\mathrm{SDH}} \coloneqq (\mathsf{IGen}, \mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2), \mathsf{ChSet}, \mathsf{V})$ and it's alternative verification $\tilde{\mathsf{V}}$ as depicted in figure 4.

Theorem 2. *Suppose that there exists a (t, q, ϵ)-forger $\mathcal{F}$ breaking the $\mathsf{UF\text{-}PA}^{\tilde{\mathsf{V}}}$ of the ID_{q-SDH} identification scheme. Then there exists an adversary $\mathcal{A}$ that $(t', q+1, \epsilon')-$ breaks the $q+1$-SDH assumption with $t \approx t'$ and $\epsilon' \geq \epsilon$.*

Proof. The q-SDH adversary $\mathcal{A}$ receives $d_0, ..., d_q$ as inputs where $d_i = g^{x^i}$ and simulates the q-SDH experiment as follows

Key Generation:

Adversary $\mathcal{A}$ first chooses random $\tau_1, ..., \tau_q$ from $\mathbb{Z}_p$. Now $\mathcal{A}$ defines $g_1 = g^{\theta f(x)}$ as in Lemma 2. $\mathcal{A}$ can also calculates $X = g_1^x = g^{xf(x)}$ similarly since $Xf(X)$ has a degree equal to $q+1$.
Adversary $\mathcal{A}$ returns (g_1, X) as the public key to $\mathcal{F}$. This is indistinguishable from the normal key generation for $\mathbb{F}$ since g_1 is randomly distributed in G and X is correctly computed.

Transcript Generation:

Now adversary $\mathcal{A}$ must compute (Com_i, C_i, s_i) for τ_i.

According to Lemma 2 $\mathcal{A}$ can compute $\hat{g}_i = g_1^{\frac{1}{x+\tau_i}}$.
To complete the computation $\mathcal{A}$ chooses $C_i, s_i \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$R = g_1^{s_i} \cdot (X \cdot g_1^{\tau})^{-C_i}$$

$$\hat{R} = \hat{g}_i^{s} \cdot (g \cdot \hat{g}^{-\tau})_i^{C}.$$

Now $\mathcal{A}$ returns $(Com_i = (\hat{g}_i, R, \hat{R}), C_i, s_i)$ to $\mathcal{F}$ and this is indistinguishable from the normal transcript generation for $\mathcal{F}$ since if we define r to be $r = s - Cx$ then $R = g_1^x$ and $\hat{R} = \hat{g}^x$. Additionally since s and C are uniformly distributed in $\mathbb{Z}_p$ so is r.

Breaking the $q+1$-SDH:

Eventually forger $\mathcal{F}$ returns a forgery $(\tau^*, Com^*, C^*, s^*)$ we assume that $\mathcal{F}$ wins the game and thus $\tau^* \notin \{\tau_1, ..., \tau_q\}$ and $\tilde{\mathsf{V}}(sk, \tau^*, Com^*, C^*, s^*) = 1$ which means if we parse Com^* as $(\hat{g}^*, R^*, \hat{R}^*)$

$$\hat{g}^* = g_1^{\frac{1}{x+\tau^*}} = g^{\frac{\theta f(x)}{x+\tau^*}}$$

will hold.
According to Lemma 2, $\mathcal{A}$ can compute $g^{\frac{1}{x+\tau^*}}$ and ultimately return the pair $(\tau^*, g^{\frac{1}{x+\tau^*}})$ as the solution to the $q+1-$SDH problem.

■

6 Instantiation of dual tag ID from q-DH

We describe the identification scheme as in figure 7. Throughout this section we will write $D(\tau)$ shorthand for $\mathsf{PHF.Eval}(\kappa, \tau)$ and $d(\tau)$ shorthand for the function computing $(a, b) \leftarrow \mathsf{PHF.TrapEval}(\eta, \tau)$ and returning $ax + b$.

Definition 9 (q-DH Assumption). *We say an adversary $\mathcal{A}$ breaks the q-Diffie Hellman (q-DH) assumption if it's running time is bounded by t and*

$$\Pr[g^{\frac{1}{x}} \xleftarrow{\$} \mathcal{A}(g, g^x, ..., g^{x^q})] \geq \epsilon,$$

where $g \xleftarrow{\$} \mathsf{G}$ and $x \xleftarrow{\$} \mathbb{Z}_p^*$. We require that the q-DH assumption holds meaning that no adversary can (t, ϵ) break the q-DH problem for a polynomial t and a non-negligible ϵ.

Lemma 3. *Let PHF be a (m, n, λ, δ)-programmable hash function and*

$$(\kappa, \eta) \xleftarrow{\$} \mathsf{PHF.Gen}(g, X)$$

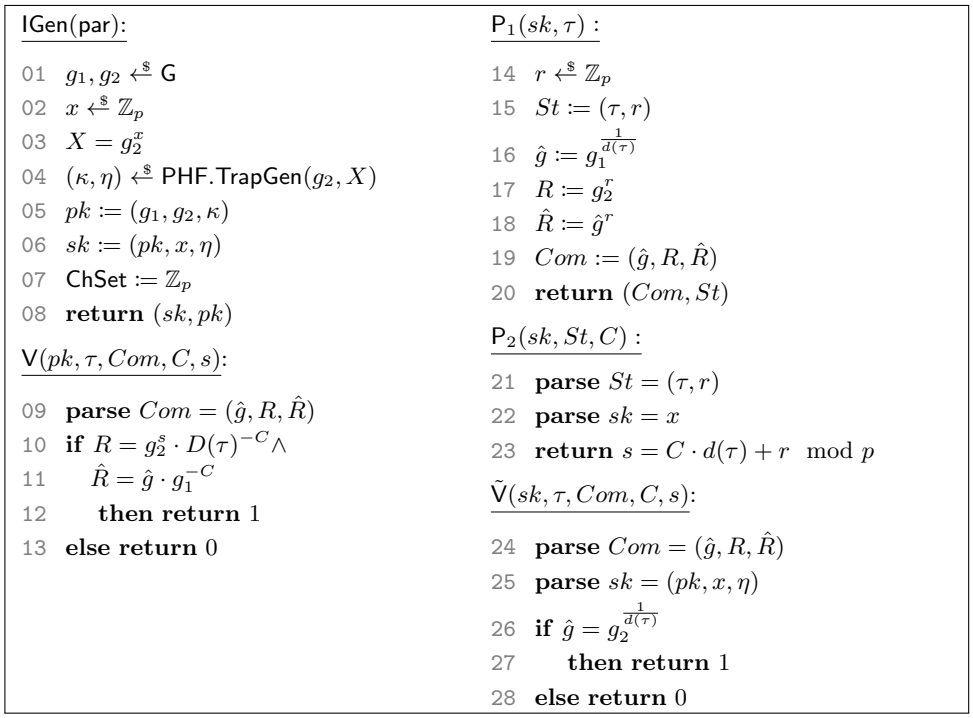

Figure 7. Instantiation 1

and $\tau_1, ..., \tau_q \in \mathbb{Z}_p$ *be such that* $(k_i, l_i) \xleftarrow{\$} \mathsf{PHF.Eval}(\eta, \tau_i)$. *We define the polynomial* f *as*

$$f(Y) \coloneqq \prod_{i=1}^{q} (k_i + l_i Y).$$

Given $\{g^{x^i}\}_{i=0,...,q}$, *let us define* $g_1 = g^{\theta f(x)}$. *For any* τ_i *where* $i \in [0, q]$ *it is easy to compute* $g_1^{\frac{1}{d(\tau_i)}}$. *Furthurmore if* $k_i \neq 0$ *for all* $i \in [0, q]$, *given* $g_1^{\frac{1}{d(\tau^*)}}$ *where* $\tau \notin \{\tau_1, ..., \tau_q\}$ *and* $(k^*, l^*) \xleftarrow{\$} \mathsf{PHF.Eval}(\kappa, \tau^*)$ *one can easily compute* $g^{\frac{1}{d(\tau^*)}}$.

Proof. To compute $g_1^{\frac{1}{d(\tau_i)}}$ for $i \in [1, q]$, let f_i be defined as

$$f_i(Y) = \frac{f(Y)}{k_i + l_i Y} = \prod_{j=1, j\neq i}^{q} (k_i + l_i Y).$$

We can write f_i as $f_i(Y) = \sum_{j=0}^{q-1} \beta_j Y^j$ while calculating its coefficient. Now if we denote g^{x^j} by d_j we can calculate

$$\hat{g}_i = \prod_{j=0}^{q-1} d_j^{\theta\beta_j}$$

hence

$$\hat{g}_i = g^{\theta f_i(x)} = g_1^{\frac{f_i(x)}{f(x)}} = g_1^{\frac{1}{d(\tau_i)}}.$$

Now to compute $g^{\frac{1}{d(\tau^*)}}$ by long division we can write $f(Y)$ as

$$f(Y) = (k^* + l^*Y)\alpha(Y) + \beta$$

where the coefficients of $\alpha(Y) = \sum_{i=0}^{q-1} \alpha_i Y^i$ are easily computable. So we can write $\frac{f(Y)}{k^*+l^*Y}$ as $\alpha(Y) + \frac{\beta}{k^*+l^*Y}$ and so we have

$$\hat{g} = g_1^{\frac{1}{d(\tau^*)}} = g^{\frac{\theta f(x)}{d(\tau^*)}} = g^{\frac{\theta f(x)}{k^*+l^*x}} = g^{\theta\cdot(\alpha(x)+\frac{\beta}{k^*+l^*x})}$$

Since $k_i + l_i Y$ divides $f(Y)$ for all $i \in [1, q]$ and $f(Y)$ has a degree of q and $k_i \neq 0$, l^*Y does not divide $f(Y)$ and thus β is non zero and we can compute

$$w \leftarrow \left(\hat{g}^{\frac{1}{\theta}} \cdot \prod_{i=0}^{q-1} d_i^{-\alpha_i}\right)^{\frac{1}{\beta}}$$

Hence, we have computed

$$w = g^{\frac{1}{k^*+l^*x}} = g^{\frac{1}{d(\tau^*)}}.$$

■

Theorem 3. *Suppose that there exists a (t, q, ϵ)-forger $\mathcal{F}$ breaking the $\mathsf{UF\text{-}PA}^{\tilde{\mathsf{V}}}$ of the ID_{q-DH} identification scheme. Then there exists an adversary $\mathcal{A}$ that $(t', q+1, \epsilon')-$ breaks the $q+1$-DH assumption with $t \approx t'$ and $\epsilon' \geq \epsilon$.*

Proof. We define the event of Game G_i winning (returning 1) as X_i. Let τ_i denote the i-th random tag in the alternative impersonation game. Let $(\tau^*, Com^*, C^*, s^*)$ be the forgery output by $\mathcal{F}$.

Game 0. We define Game 0 as the alternative impersonation game and so by definition

$$\Pr[X_0] = \epsilon.$$

Game 1. In this game we choose the tags $\tau_1, ..., \tau_q$ all in the beginning. This does not effect the success probability of the adversary so

$$\Pr[X_1] = \Pr[X_0].$$

Game 2. In this game we compute $(l_i, k_i) \leftarrow \mathsf{PHF.Eval}(\kappa, \tau_i)$ for every τ_i and set $\mathtt{BAD}$ to **true** if for any i, l_i is zero. We also compute $(l^*, k^*) \leftarrow \mathsf{PHF.Eval}(\kappa, \tau^*)$ when the adversary has output the forgery and set $\mathtt{BAD}$ to **true** if l^* is not zero. By the $(1, \mathsf{poly}, \gamma, \delta)-$programmability of D we have

$$\Pr[\neg\mathtt{BAD}] \geq \delta$$

Game 3. This game is exactly like the last game except that we abort the game if $\mathtt{BAD}$ is **true** which means

$$\Pr[X_3] = \Pr[X_2 \wedge \neg\mathsf{abort}] \geq \delta \cdot \Pr[X_2]$$

The q-DH adversary $\mathcal{A}$ receives $d_0, ..., d_q$ as inputs where $d_i = g_2^{x^i}$ and simulates the soundness experiment as follows

$\underline{\mathsf{G}_0 - \mathsf{G}_3}$

$\mathtt{BAD} \leftarrow \textbf{false}$ // $\mathsf{G}_2 - \mathsf{G}_3$
for $i \in [q]$ // $\mathsf{G}_1 - \mathsf{G}_3$
  $\tau_i \xleftarrow{\$} \mathbb{Z}_p$ // $\mathsf{G}_1 - \mathsf{G}_3$
$g_1, g_2 \xleftarrow{\$} \mathsf{G}$
$x \xleftarrow{\$} \mathbb{Z}_p$
$X = g_2^x$
$(\kappa, \eta) \xleftarrow{\$} \mathsf{PHF.TrapGen}(g_2, X)$
$pk \coloneqq (g_1, g_2, \kappa)$
$sk \coloneqq (pk, x, \eta)$
$\mathcal{Q} \leftarrow \emptyset$
for $i \in [q]$
  $\tau_i \xleftarrow{\$} \mathbb{Z}_p$ // G_0
  $r_i \xleftarrow{\$} \mathbb{Z}_p$
  $St_i \coloneqq (\tau_i, r_i)$
  $(l_i, k_i) \leftarrow \mathsf{PHF.Eval}(\kappa, \tau_i)$ // $\mathsf{G}_2 - \mathsf{G}_3$
  if $l_i = 0$ // $\mathsf{G}_2 - \mathsf{G}_3$
    then $\mathtt{BAD} \leftarrow \textbf{true}$ // $\mathsf{G}_2 - \mathsf{G}_3$
  $\hat{g}_i \coloneqq g_1^{\frac{1}{d(\tau_i)}}$
  $R_i \coloneqq g_2^{r_i}$
  $\hat{R}_i \coloneqq \hat{g}^{r_i}$
  $Com_i \coloneqq (\hat{g}_i, R_i, \hat{R}_i)$
  $C_i \xleftarrow{\$} \mathbb{Z}_p$
  $s_i = C_i \cdot d(\tau_i) + r_i$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup (\tau_i, Com_i, C_i, s_i)$
$(\tau^*, Com^*, C^*, s^*) \leftarrow \mathcal{A}(pk, \mathcal{Q})$
  $(l^*, k^*) \leftarrow \mathsf{PHF.Eval}(\kappa, \tau^*)$ // $\mathsf{G}_2 - \mathsf{G}_3$
  if $l^* \neq 0$ // $\mathsf{G}_2 - \mathsf{G}_3$
    then $\mathtt{BAD} \leftarrow \textbf{true}$ // $\mathsf{G}_2 - \mathsf{G}_3$
  if $\mathtt{BAD}$ // G_3
    then abort // G_3
if $\tau^* \notin \{\tau_1, ..., \tau_q\} \wedge \tilde{\mathsf{V}}(sk, \tau^*, Com^*, C^*, s^*) = 1$
 then return 1
else return 0

Figure 8.

Key Generation:

Adversary $\mathcal{A}$ first chooses random $\tau_1, ..., \tau_q$ from $\mathbb{Z}_p$.

Key Generation:

The adversary $\mathcal{A}$ first chooses random $\tau_1, ..., \tau_q$ from $\mathbb{Z}_p$. Now $\mathcal{A}$ has g_2 as d_0 and sets

$$X \coloneqq d_1 = g_2^x.$$

Now $\mathcal{A}$ runs run the following

$$(\kappa, \eta) \xleftarrow{\$} \mathsf{PHF.Gen}(g_2, X)$$

just like the original key generation algorithm. Then using η, $\mathcal{A}$ runs

$$(k_i, l_i) \xleftarrow{\$} \mathsf{PHF.Eval}(\eta, \tau_i)$$

for every $i \in [1, q]$.
$\mathcal{A}$ computes g_1 as in Lemma 3, such that

$$g_1 = g_2^{\theta f(x)}.$$

$\mathcal{A}$ can also calculates $X = g_1^x = g_2^{xf(x)}$ similarly since $Yf(Y)$ has a degree equal to $q + 1$.
Adversary $\mathcal{A}$ returns (g_1, g_2, κ) as the public key to $\mathcal{F}$. This is indistinguishable from the normal key generation for $\mathbb{F}$ since g_1 is randomly distributed in G.

Transcript Generation:

Now adversary $\mathcal{A}$ has compute (Com_i, C_i, s_i) for all τ_i where $i \in [1, q]$.
According to Lemma 2 $\mathcal{A}$ can compute $\hat{g}_i = g_1^{\frac{1}{d(\tau_i)}}$.
To complete the computation $\mathcal{A}$ chooses $C_i, s_i \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$R = g_2^{s_i} \cdot D(\tau)^{-C_i}$$

$$\hat{R} = \hat{g}_i \cdot g_1^{-C_i}.$$

Now $\mathcal{A}$ returns $(Com_i = (\hat{g}_i, R, \hat{R}), C_i, s_i)$ to $\mathcal{F}$ and this is indistinguishable from the normal transcript generation for $\mathcal{F}$ since if we define r to be $r = s - Cx$ then $R = g_1^x$ and $\hat{R} = \hat{g}^x$ and also since s and C are uniformly distributed in $\mathbb{Z}_p$ so is r. ■

Breaking the $q + 1$-DH:

Eventually forger $\mathcal{F}$ returns a forgery $(\tau^*, Com^*, C^*, s^*)$ we assume that $\mathcal{F}$ wins the game and thus $\tau^* \notin \{\tau_1, ..., \tau_q\}$ and $\tilde{\mathsf{V}}(sk, \tau^*, Com^*, C^*, s^*) = 1$ which means if we parse Com^* as $(\hat{g}^*, R^*, \hat{R}^*)$

$$\hat{g}^* = g_1^{\frac{1}{d(\tau^*)}}$$

will hold.
According to Lemma 3, $\mathcal{A}$ can now compute

$$g^{\frac{1}{d(\tau^*)}} = g^{\frac{1}{k^*+l^*x}} \overset{(*)}{=} g^{\frac{1}{l^*x}}$$

where

$$w := (k^*, l^*) \xleftarrow{\$} \mathsf{PHF.Eval}(\kappa, \tau^*)$$

and $(*)$ uses that $k^* = 0$ by Game 3.
Ultimately, $\mathcal{A}$ can compute $w^{l^*} = g^{\frac{1}{x}}$ and return it as the solution to the $q+1-$DH problem.