Optimal Tightness for Chain-Based Unique Signatures

Fuchun Guo *Willy Susilo*

(Eurocrypt 2022)

UNIVERSITY OF WOLLONGONG AUSTRALIA

iC² Institute of Cybersecurity and Cryptology

西安电子科技大学

In This Work

We focus on this question:

How to program a tight reduction for a signature scheme when signatures are unique (without using random numbers in signature generations)?

Security Reduction and Tight Reduction

Security Reduction:

- Suppose there exists an adversary who can break a scheme.
- We (prover) reduce breaking the scheme to solving a hard problem
  (Formally speaking, a hard problem is reducible to a cryptographic scheme)

A security reduction is tight if

- An adversary can (t, ϵ)-break the scheme, and
- We (prover) can solve a hard problem with (t', ϵ') such that $\frac{t}{\epsilon} \approx \frac{t'}{\epsilon'}$.

A tight reduction guarantees that breaking a scheme is as difficult as solving a hard problem.

Digital Signature Schemes with Tight Reductions

Digital Signatures are one of the fundamental primitives for integrity

- **KeyGen**: A signer generates a key pair (pk, sk).
- **Sign**: The signer uses sk to generate signature Σ_m on message m.
- **Verify**: Any verifier can use pk to verify (m, Σ_m).

How to construct a signature scheme with a tight reduction receives lots of attention in the literature.

Our community has also invented many intelligent methods for tight security in the literature (See references in our paper).

Digital Signature Schemes with Tight Reductions

It is no longer hard to obtain tight reductions in the following cases.

	EUF-CMA	Non-Interactive Hardness Assumption	General Adversary
Case 1	× (e.g. Key-only attacks)	✓	✓
Case 2	✓	× (e.g. One-more assumption)	✓
Case 3	✓	✓	× (e.g. Algebraic Group Model)

EUF-CMA: existentially unforgeable against chosen-message attacks.

General Adversary: probabilistic polynomial time Turing machine.

Research question is narrowed down to:

How to have a tight reduction under non-interactive assumption in EUF-CMA security model against general adversaries?

Digital Signature Schemes with Tight Reductions

It is also not hard to have a tight reduction under non-interactive assumption in EUF-CMA security model against general adversaries, as long as the signatures are randomized.

	Signature Query		Signature Forgery
EUF-CMA	$m_1 \to m_2 \to m_3 \to \cdots \to m_q$	$\to$	m^*

Randomized Approach for Tight Reduction

- There are at least two valid signatures for each message.
- One is **simulatable** and the others are **reducible**.
- For signature query on m_i, we return a simulatable signature.
- The forged signature is reducible with high probability.

Research question is further narrowed down to:

How to have a tight reduction with these three factors but without using the randomized approach?

Unique Signature Schemes with Tight Reductions

Unique Signatures

- Let $\Sigma(pk, m) = \{\Sigma_m : \Sigma_m \text{ is a valid signature on } m \text{ under } pk\}$
- A signature scheme is a unique signature scheme if $\Sigma(pk, m)$ has only one signature for all pk and m.

Example: BLS Signatures

- $pk = g^{\alpha}$ and $\Sigma_m = H(m)^{\alpha}$ where $H : \{0,1\}^* \to \mathbb{G}$

Randomized Approach cannot be applied to unique signatures.

- There are ~~at least two~~ valid signatures for each message.
- ~~One is simulatable and the others are reducible.~~
- ~~We return a simulatable signature for signature query on m_i.~~
- ~~The forged signature is reducible with high probability.~~

Tight for Unique Signatures: Possible or Impossible?

It seems impossible to have a tight reduction if $\cdots\cdots$

An adversary can

- Choose random messages $m_1, m_2, \cdots, m_q, m_{q+1}$, and
- Make hash queries to RO (if it is in RO model)

before signature queries and signature forgery, such that

- The signature on each m_i is simulatable or reducible, and
- the result (s or r) cannot be switched by the simulator (prover).

Attack in this way:

The adversary randomly picks q messages from $\{m_1, m_2, \cdots, m_{q+1}\}$ for signature queries and then forges Σ_{m^*} on the last message m^*.

- If there are two reducible signatures, the reduction will fail.
- If there is only one, the success probability is $\frac{1}{q+1} < \frac{1}{q}$.

Related Work for Impossible

The impossibility is analyzed via meta-reduction technique.

- (Eurocrypt 02) Coron $\rightarrow$ Unique signatures
- (PKC 12) Hofheinz-Jager-Knapp $\rightarrow$ Re-randomizable signatures
- (Eurocrypt 16) Bader-Jager-Li-Schäge $\rightarrow$ Lower bound and more
- (TCC 18) Morgan-Pass $\rightarrow$ Bounded-round interactive assumption

They showed that the maximum success probability of security reduction is $\frac{1}{q}$ for q signature queries in EUF-CMA.

See our paper to find more impossibility analysis of tight reductions for other cryptographic primitives: specific signature schemes, encryption, key exchange, MACs, PRFs, and verifiable random functions.

Related Work for Possible: Chain-Based Construction

Guo-Chen-Susilo-Lai-Yang-Mu: "Optimal Security Reductions for Unique Signatures: Bypassing Impossibilities with a Counterexample". Crypto 2017: 517-547

KeyGen: $pk = g^{\alpha}$ and $sk = \alpha$.

Sign: The signature on m is composed of n block signatures:

$$(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_n) = \left(H(m|\Sigma_m^0)^{\alpha}, H(m|\Sigma_m^1)^{\alpha}, H(m|\Sigma_m^2)^{\alpha}, \cdots, H(m|\Sigma_m^{n-1})^{\alpha}\right),$$

where $\Sigma_m^i = (\sigma_1, \sigma_2, \cdots, \sigma_i)$, $\Sigma_m^0 = (\ )$, and $\Sigma_m = \Sigma_m^n$.

Verify: Accepts Σ_m if $e(\sigma_{i+1}, g) = e(H(m|\Sigma_m^i), h)$ for all $i \in [0, n-1]$.

The meaning of **Chain** (blockchain)

- Block signatures Σ_m^0 are treated as message and signed to obtain Σ_m^1
- Block signatures Σ_m^1 are treated as message and signed to obtain Σ_m^2
- $\cdots$

Related Work for Possible: Chain-Based Construction

Guo-Chen-Susilo-Lai-Yang-Mu: "Optimal Security Reductions for Unique Signatures: Bypassing Impossibilities with a Counterexample". Crypto 2017: 517-547

$$(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_n) = \left(H(m|\Sigma_m^0)^\alpha,\ H(m|\Sigma_m^1)^\alpha,\ H(m|\Sigma_m^2)^\alpha, \cdots, H(m|\Sigma_m^{n-1})^\alpha\right),$$

where $\Sigma_m^i = (\sigma_1, \sigma_2, \cdots, \sigma_i)$, $\Sigma_m^0 = (\ )$, and $\Sigma_m = \Sigma_m^n$.

In their security reduction, an adversary can still

- Choose random messages $m_1, m_2, \cdots, m_q, m_{q+1}$, and
- Make hash queries to RO (if it is in RO model)

before signature queries and signature forgery, such that

- The signature on each m_i must be either simulatable or reducible.

But simulator has already solved hard problem from hash queries.

Related Work for Possible: Chain-Based Construction

$$(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_n) = \left(H(m|\Sigma_m^0)^\alpha,\ H(m|\Sigma_m^1)^\alpha,\ H(m|\Sigma_m^2)^\alpha, \cdots, H(m|\Sigma_m^{n-1})^\alpha\right),$$

where $\Sigma_m^i = (\sigma_1, \sigma_2, \cdots, \sigma_i)$, $\Sigma_m^0 = (\ )$, and $\Sigma_m = \Sigma_m^n$.

Random Oracle Model: Each signature requires n different queries to RO

Type-0 Query	Type-1 Query	Type-2 Query	$\cdots$	Type-$(n-1)$ Query
$m\|\Sigma_m^0$	$m\|\Sigma_m^1$	$m\|\Sigma_m^2$	$\cdots$	$m\|\Sigma_m^{(n-1)}$

Before signature query on m,

- $\mathcal{A}$ should make Type-0 first, Type-1, Type-2, $\cdots$ (sequential).
- $\mathcal{A}$ needs to compute and submit σ_i for Type-i query (except Type-0)

The prover can embed hard problem solution in σ_i.

- CDH Problem: Given g, g^a, g^b, compute g^{ab}.
- Set $\alpha = a$ and $H(m|\Sigma_m^{i-1}) = g^b$. (respond type-$(i-1)$ query with g^b)
- We have $\sigma_i = H(m|\Sigma_m^{i-1})^\alpha = g^{ab}$. (obtain problem solution from type-i)

Crypto 17: How to Achieve Log Tightness

	Example						
Type-(n-1)			$m_3 \vert \Sigma_{m_3}^{n-1}$			$m^* \vert \Sigma_{m^*}^{n-1}$	Q_{n-1}
⋮			⋮			⋮	⋮
Type-2	$m_1 \vert \Sigma_{m_1}^{2}$		$m_3 \vert \Sigma_{m_3}^{2}$			$m^* \vert \Sigma_{m^*}^{2}$	Q_2
Type-1	$m_1 \vert \Sigma_{m_1}^{1}$	$m_2 \vert \Sigma_{m_2}^{1}$	$m_3 \vert \Sigma_{m_3}^{1}$		$m_q \vert \Sigma_{m_q}^{1}$	$m^* \vert \Sigma_{m^*}^{1}$	Q_1
Type-0	$m_1 \vert \Sigma_{m_1}^{0}$	$m_2 \vert \Sigma_{m_2}^{0}$	$m_3 \vert \Sigma_{m_3}^{0}$	$\cdots$	$m_q \vert \Sigma_{m_q}^{0}$	$m^* \vert \Sigma_{m^*}^{0}$	Q_0
	$m_1 \rightarrow$	$m_2 \rightarrow$	$m_3 \rightarrow$	$\cdots$	$m_q \rightarrow$	m^*	

- $\mathcal{A}$ can make type-0, type-1, $\cdots$, type-k_i queries of m_i before its signature query for any adaptive integer $k_i \in [0, n-1]$.
- $\mathcal{A}$ must make all n queries of m^* for signature forgery.
- Let Q_i be the number of all type-i queries generated by $\mathcal{A}$.

The authors in Crypto 17 proved that there exists $i^* \in [0, n-1]$ such that

$$Q_{i^*} \leq q_H^{1-\frac{i^*}{n}} \wedge Q_{i^*+1} \geq q_H^{1-\frac{i^*+1}{n}} : \quad \frac{q_H^{1-\frac{i^*+1}{n}}}{q_H^{1-\frac{i^*}{n}}} = \frac{1}{q_H^{1/n}}$$

Crypto 17: How to Achieve Log Tightness

	Example						
Type-(n-1)			$m_3 \vert \Sigma_{m_3}^{n-1}$			$m^* \vert \Sigma_{m^*}^{n-1}$	Q_{n-1}
⋮			⋮			⋮	⋮
Type-2	$m_1 \vert \Sigma_{m_1}^{2}$		$m_3 \vert \Sigma_{m_3}^{2}$			$m^* \vert \Sigma_{m^*}^{2}$	Q_2
Type-1	$m_1 \vert \Sigma_{m_1}^{1}$	$m_2 \vert \Sigma_{m_2}^{1}$	$m_3 \vert \Sigma_{m_3}^{1}$		$m_q \vert \Sigma_{m_q}^{1}$	$m^* \vert \Sigma_{m^*}^{1}$	Q_1
Type-0	$m_1 \vert \Sigma_{m_1}^{0}$	$m_2 \vert \Sigma_{m_2}^{0}$	$m_3 \vert \Sigma_{m_3}^{0}$	$\cdots$	$m_q \vert \Sigma_{m_q}^{0}$	$m^* \vert \Sigma_{m^*}^{0}$	Q_0
	$m_1 \rightarrow$	$m_2 \rightarrow$	$m_3 \rightarrow$	$\cdots$	$m_q \rightarrow$	m^*	

The authors in Crypto 17 proved that there exists $i^* \in [0, n-1]$ such that

$$Q_{i^*} \leq q_H^{1-\frac{i^*}{n}} \wedge Q_{i^*+1} \geq q_H^{1-\frac{i^*+1}{n}} : \frac{\text{Number of type-}(i^*+1)}{\text{Number of type-}i^*} \geq \frac{q_H^{1-\frac{i^*+1}{n}}}{q_H^{1-\frac{i^*}{n}}} = \frac{1}{q_H^{1/n}}$$

- One of type-i^* queries is responded with g^b.
- The CDH solution g^{ab} will appear in one of type-(i^*+1) queries.
- The success probability is $\frac{1}{n \cdot q_H^{1/n}}$ (guess i^* correctly with $\frac{1}{n}$)

Question: How to program a tight reduction when signatures are unique?

KeyGen: $pk = g^{\alpha}$ and $sk = \alpha$.

Sign: The signature on m is

$$(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_n) = \left(H(m|\Sigma_m^0)^{\alpha}, H(m|\Sigma_m^1)^{\alpha}, H(m|\Sigma_m^2)^{\alpha}, \cdots, H(m|\Sigma_m^{n-1})^{\alpha}\right),$$

where $\Sigma_m^i = (\sigma_1, \sigma_2, \cdots, \sigma_i)$, $\Sigma_m^0 = (\ )$, and $\Sigma_m = \Sigma_m^n$.

Verify: Accepts Σ_m if $e(\sigma_{i+1}, g) = e(H(m|\Sigma_m^i), h)$ for all $i \in [0, n-1]$.

- The proof for chain-based BLS scheme is the only known one.
- The reduction loss is $n \cdot q_H^{1/n} \geq \log(q_H)$.
- **Contribution 1:** Show that the optimal loss is $q^{1/n}$ (q signature queries).
- **Contribution 2:** Show how to obtain such an optimal reduction.

Chain-Based BLS Scheme: Our Observation (1/4)

KeyGen: The key pair is $pk = h = g^{\alpha}$ and $sk = \alpha$.
Sign: The signature on m is
$(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_n) = \left(H(m|\Sigma_m^0)^{\alpha}, H(m|\Sigma_m^1)^{\alpha}, H(m|\Sigma_m^2)^{\alpha}, \cdots, H(m|\Sigma_m^{n-1})^{\alpha}\right)$,
where $\Sigma_m^i = (\sigma_1, \sigma_2, \cdots, \sigma_i)$, $\Sigma_m^0 = (\,)$, and $\Sigma_m = \Sigma_m^n$.

Proof. Given g, g^a, g^b, the CDH problem is to compute g^{ab}.

- Set $h = g^a$, namely $\alpha = a$.
- Non-uniformly choose an integer $c \in [0, n-1]$ for each message m.
- Plan (but could change) to set the RO response to type-i query as

$$H(m|\Sigma_m^i) = \begin{cases} g^{x_i} & i \neq c \quad (\text{Normal query without } g^b \text{ in response}) \\ g^{b+x_i} & i = c \quad (\text{Challenge query with } g^b \text{ in response}) \end{cases}$$

- $\mathcal{A}$ makes type-0, type-1, $\cdots$, type-k of m before signature query.
  - If $k < c$, we can simulate signature by setting type-c to a normal query.
  - If $k = c$, abort (cannot simulate the signature)
  - If $k > c$, hard problem solution has appeared in type-$(c+1)$ query.

Chain-Based BLS Scheme: Our Observation (2/4)

KeyGen: The key pair is $pk = h = g^{\alpha}$ and $sk = \alpha$.
Sign: The signature on m is
$(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_n) = \left(H(m|\Sigma_m^0)^{\alpha}, H(m|\Sigma_m^1)^{\alpha}, H(m|\Sigma_m^2)^{\alpha}, \cdots, H(m|\Sigma_m^{n-1})^{\alpha}\right)$,
where $\Sigma_m^i = (\sigma_1, \sigma_2, \cdots, \sigma_i)$, $\Sigma_m^0 = (\,)$, and $\Sigma_m = \Sigma_m^n$.

Proof. $\mathcal{A}$ makes type-0, type-1, $\cdots$, type-k before signature query on m.

- If $k < c$, we can simulate signature by setting all to normal queries.
- If $k = c$, abort.
- If $k > c$, hard problem solution has appeared in hash query.

Suppose that $\mathcal{A}$ makes one signature query before signature forgery.

Something BIG here:

$$\Pr = \text{solve before query} + \text{neither success nor abort} \cdot \text{solve from forgery}$$
$$= P_{S_1} + (1 - P_{F_1} - P_{S_1}) \cdot P_{S^*}$$

- P_{S_1} : solve problem before signature query.
- P_{F_1} : fail due to signature query.
- P_{S^*} : solve problem from forged signature.

Chain-Based BLS Scheme: Our Observation (3/4)

The most important page in our presentation!

Suppose that $\mathcal{A}$ makes one signature query before signature forgery.

- First, we have

$$\Pr = P_{S_1} + (1 - P_{F_1} - P_{S_1}) \cdot P_{S^*}.$$

- If $P_{S^*} \leq \frac{1}{2}$ (close and not more than), we have

$$\begin{aligned} \Pr &= P_{S_1} + (1 - P_{F_1} - P_{S_1}) \cdot P_{S^*} \\ &\geq 2 \cdot P_{S^*} \cdot P_{S_1} + (1 - P_{F_1} - P_{S_1}) \cdot P_{S^*} = (1 - P_{F_1} + P_{S_1}) \cdot P_{S^*} \end{aligned}$$

- If we can achieve $P_{F_1} - P_{S_1} = \frac{1}{q}$ (constant and small), we have

$$\Pr \geq (1 - P_{F_1} + P_{S_1}) \cdot P_{S^*} = \left(1 - \frac{1}{q}\right) \cdot P_{S^*}$$

Meaning: After q signature queries, the success probability is

$$\left(1 - \frac{1}{q}\right)^q \cdot P_{S^*} \approx \frac{P_{S^*}}{e} : \quad e \approx 2.7$$

Chain-Based BLS Scheme: Our Observation (4/4)

How to achieve $P_{S^*} \approx \frac{1}{2}$ and $P_{F_i} - P_{S_i} = \frac{1}{q}$?

- Consider Geometric Progression (GP)

$$\left(\frac{1}{2^{n+1}}, \frac{1}{2^n}, \frac{1}{2^{n-1}}, \cdots, \frac{1}{2^{j+1}}, \frac{1}{2^j}, \cdots, \frac{1}{2^2}\right)$$

$$P_{S_i} = \frac{1}{2^{n+1}} + \frac{1}{2^n} + \frac{1}{2^{n-1}} + \cdots + \frac{1}{2^{j+1}}, \quad P_{F_i} = \frac{1}{2^j}$$

- No matter what j is, we have

$$P_{F_i} - P_{S_i} = \frac{1}{2^j} - \left(\frac{1}{2^{n+1}} + \frac{1}{2^n} + \frac{1}{2^{n-1}} + \cdots + \frac{1}{2^{j+1}}\right) = \frac{1}{2^{n+1}} \to \frac{1}{q}$$

- We have

$$\frac{1}{2^{n+1}} + \frac{1}{2^n} + \frac{1}{2^{n-1}} + \cdots + \frac{1}{2^j} + \cdots + \frac{1}{2^2} = \frac{1}{2} - \frac{1}{2^{n+1}} \approx \frac{1}{2} \to P_{S^*}$$

Chain-Based BLS Scheme: Our Proof (1/2)

Proof. Given g, g^a, g^b, set $h = g^a$, namely $\alpha = a$. Then

For each message, g^b will be embedded in response to type-c query before its signature query

	Type-0	Type-1	Type-2	$\cdots$	Type-$(n-1)$
	$H(m\|\Sigma_m^0)$	$H(m\|\Sigma_m^1)$	$H(m\|\Sigma_m^2)$	$\cdots$	$H(m\|\Sigma_m^{n-1})$
with probability	$\frac{1}{2^{n+1}}$	$\frac{1}{2^n}$	$\frac{1}{2^{n-1}}$	$\cdots$	$\frac{1}{2^2}$

- For each message m, a specific c will be non-uniformly chosen.
- Suppose g^b is (will be) embedded in the response to type-c query.
- $\mathcal{A}$ makes type-0, type-1, $\cdots$,type-k of m before signature query.
  - If $k < c$, we can simulate Σ_m by setting type-c to a normal query.
  - If $k = c$, abort (cannot simulate the signature)
  - If $k > c$, hard problem solution g^{ab} has appeared in type-$(c+1)$ query.

Chain-Based BLS Scheme: Our Proof (2/2)

- For m_i, an integer $c_i \in [0, n-1]$ will be non-uniformly chosen.
- $\mathcal{A}$ makes type-0, type-1, $\cdots$,type-k_i of m_i before signature query.
- P_{F_i}: when g^b is embedded in response to type-k_i.
- P_{S_i}: when g^b is embedded in response to any type-c_i with $c_i < k_i$.
- k_i is adaptively chosen by $\mathcal{A}$, but no matter what k_i is, we have

$$P_{F_i} - P_{S_i} = \frac{1}{2^{n-k_i+1}} - \left(\frac{1}{2^{n+1}} + \frac{1}{2^n} + \frac{1}{2^{n-1}} + \cdots + \frac{1}{2^{n-k_i+2}}\right) = \frac{1}{2^{n+1}}.$$

$\mathcal{A}$ must make all n types of queries of m^*.

$$P_{S^*} = \frac{1}{2^{n+1}} + \frac{1}{2^n} + \frac{1}{2^{n-1}} + \cdots + \frac{1}{2^j} + \cdots + \frac{1}{2^2} = \frac{1}{2} - \frac{1}{2^{n+1}} \approx \frac{1}{2}$$

This completes the overview of our optimal reduction. □

Optimal Loss of Chain-Based BLS Scheme

A chain-based unique signature is composed of n BLS signatures.

- We can prove it with reduction loss $4 \cdot q^{1/n}$.
- It is constant and small when $n = \log(q)$ (namely, $\frac{1}{2^{n+1}} \approx \frac{1}{q}$)

Next, we show that the reduction loss must be at least $q^{1/n}$.

The framework of meta reduction by Coron (Eurocrypt 2002)

- We construct a special hypothetical adversary $\mathcal{A}$ that can break with probability $\epsilon_{\mathcal{A}}$. When interacting with $\mathcal{A}$, $\mathcal{R}$ would break the hardness assumption with probability $\epsilon_{\mathcal{R}}$.
- We simulate $\mathcal{A}$ via rewinding $\mathcal{R}$. If we can efficiently simulate $\mathcal{A}$ except with error probability ϵ_E, $\mathcal{R}$ would break the hardness assumption with probability $\epsilon_{\mathcal{R}} - \epsilon_E$. $\quad (\epsilon_E = \frac{\epsilon_{\mathcal{A}}}{q^{1/n}})$
- The meta-reduction shows that $\epsilon_{\mathcal{R}} \leq \epsilon_E$. Otherwise, we can run $\mathcal{R}$ as an oracle to break the hardness assumption.

Challenge: How to construct $\mathcal{A}$ and simulate $\mathcal{A}$ with error probability $\epsilon_E = \frac{\epsilon_{\mathcal{A}}}{q^{1/n}}$

The Special Hypothetical Adversary

The adversary makes the following queries if $\mathcal{R}$ doesn't abort.

$$\mathcal{T}_0(\mathcal{M}_0) \to \mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1) \to \mathcal{T}_1(\mathcal{M}_1) \to \mathcal{S}(\mathcal{M}_1 \setminus \mathcal{M}_2) \to \cdots \to \mathcal{S}(\mathcal{M}_{n-1} \setminus \mathcal{M}_n) \to \mathcal{T}_n(\mathcal{M}_n)$$

- Message sets $\mathcal{M}_0 \supset \mathcal{M}_1 \supset \mathcal{M}_2 \supset \mathcal{M}_3 \supset \cdots \supset \mathcal{M}_n$
- $\mathcal{M}_i$ has $q^{1-\frac{i}{n}}$ messages. (We have $|\mathcal{M}_0| = q$ and $|\mathcal{M}_n| = 1$)
- $\mathcal{T}_i(\mathcal{M}_i)$ is the set of type-i queries of all messages in $\mathcal{M}_i$.
- $\mathcal{S}(\mathcal{M}_i \setminus \mathcal{M}_{i+1})$ is the set of signature queries on $\mathcal{M}_i \setminus \mathcal{M}_{i+1}$.

The last type-n query is $m^*|\Sigma^n_{m^*}$. If the adversary can make such a query, it can forge a signature on m^* because of no signature query.

Example: $\mathcal{T}_0(\mathcal{M}_0) \to \mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1) \to \mathcal{T}_1(\mathcal{M}_1)$

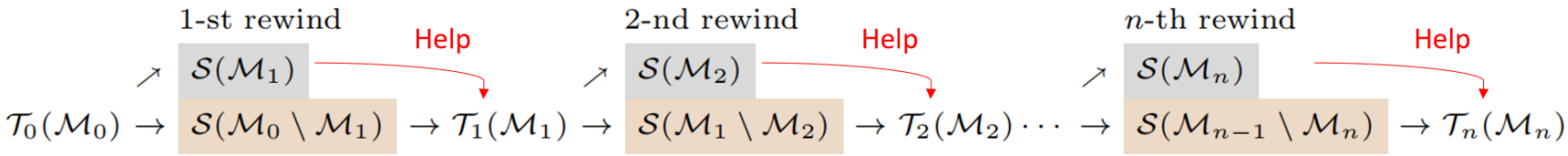
- Make all type-0 queries of messages $\{m_1, m_2, \cdots, m_q\}$.
- Query signatures on messages in $\mathcal{M}_0 \setminus \mathcal{M}_1$ with number $q - q^{1-\frac{1}{n}}$.
- Generate all type-1 queries $(q^{1-\frac{1}{n}})$ of messages in $\mathcal{M}_1 : m|\Sigma^1_m$.

The Simulated Adversary

Challenge of Simulation: How to simulate $\mathcal{A}$ without knowing secret key?

$$\mathcal{T}_0(\mathcal{M}_0) \to \cdots \to \mathcal{T}_1(\mathcal{M}_1) \to \cdots \to \mathcal{T}_2(\mathcal{M}_2) \to \cdots \to \mathcal{T}_3(\mathcal{M}_3) \to \cdots \to \mathcal{T}_n(\mathcal{M}_n)$$

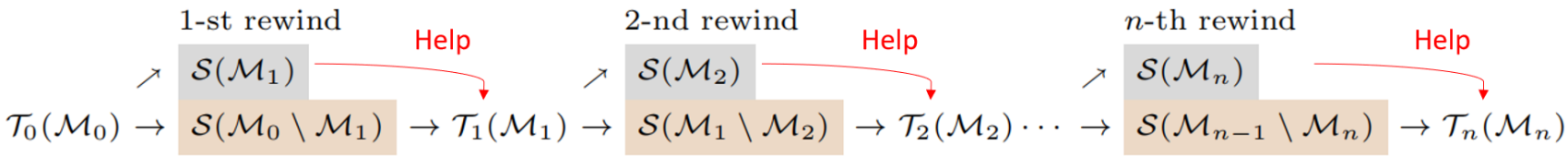
Use rewinding to simulate the adversary:



- It requires n times of rewinding to simulate $\mathcal{A}$.
- At the state after receiving responses to hash queries $\mathcal{T}_0(\mathcal{M}_0)$:
  - Before the rewind: $\mathcal{S}(\mathcal{M}_1)$
  - After the rewind: $\mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1) \to \mathcal{T}_1(\mathcal{M}_1)$

  Use signatures from $\mathcal{S}(\mathcal{M}_1)$ to simulate $\mathcal{T}_1(\mathcal{M}_1)$

The Error Probability (1/2)



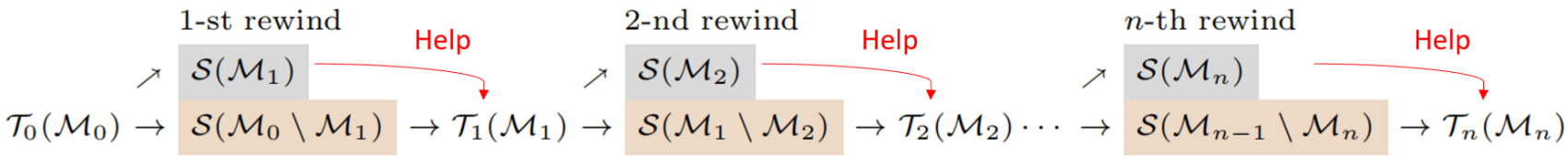
The error happens when there exists an integer $i^{\#} \in [1, n]$ such that

- Before the $i^{\#}$-th rewind: $\mathcal{R}$ cannot respond to queries $\mathcal{S}(\mathcal{M}_{i^{\#}})$ (the simulated adversary does not have signatures on $\mathcal{M}_{i^{\#}}$), but
- After the $i^{\#}$-th rewind: $\mathcal{R}$ can respond to queries $\mathcal{S}(\mathcal{M}_{i^{\#}-1} \setminus \mathcal{M}_{i^{\#}})$ (the simulated adversary has to continue queries $\mathcal{T}_{i^{\#}}(\mathcal{M}_{i^{\#}})$).

Taking $i^{\#} = 1$ as the example:

The error probability is equal to: $\mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1) = 1_{true} \ \wedge \ \mathcal{S}(\mathcal{M}_1) = 0_{false}$

The Error Probability (2/2)



The error probability is equal to: $\mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1) = 1_{true} \wedge \mathcal{S}(\mathcal{M}_1) = 0_{false}$

At the state after receiving responses to hash queries $\mathcal{T}_0(\mathcal{M}_0)$, the signatures $H(m|\Sigma_m^0)^\alpha$ must be either simulatable or reducible.

Let the number of reducible signatures in $\mathcal{M}_0$ be N.

- If $N{=}0$, no error because $\mathcal{S}(\mathcal{M}_1) = 1_{true}$
- If $1{\leq}N{\leq}|\mathcal{M}_1|$, then $\Pr[\textbf{error}] \leq \Pr[\mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1) = 1_{true}] \leq \frac{1}{q^{1/n}}$.
  ($\mathcal{R}$ can respond to $\mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1)$)
- If $N{>}|\mathcal{M}_1|$, we must have $\mathcal{S}(\mathcal{M}_0 \setminus \mathcal{M}_1){\neq}1_{true}$.
  (meaning that it is aborted by $\mathcal{R}$ and no error)

Based on no error or $\frac{1}{q^{1/n}}$ and $\epsilon_{\mathcal{A}}$, the final error probability will be $\frac{\epsilon_{\mathcal{A}}}{q^{1/n}}$.

Conclusion

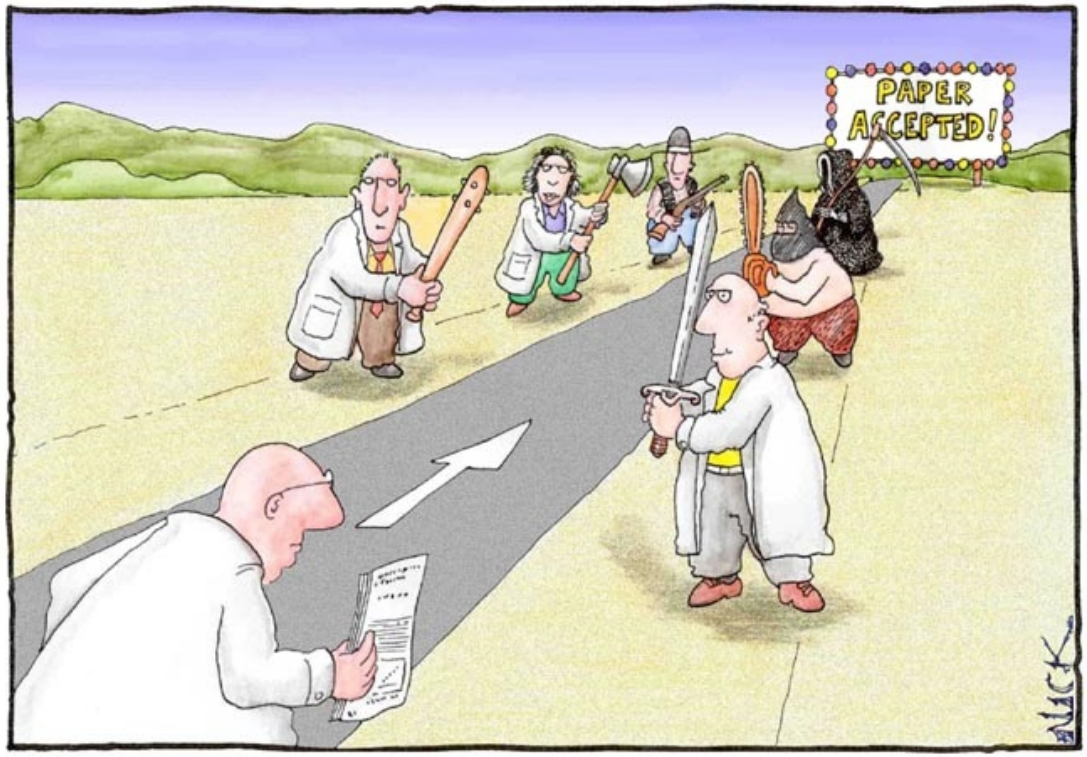
How to program a tight reduction when signatures are unique?

- It is non-trivial under any non-interactive hardness assumption in EUF-CMA security model against general adversaries.
- The only known tight reduction for chain-based construction (BLS) has reduction loss $n \cdot q_H^{1/n}$. (Crypto 2017)

- We prove that the optimal loss is $q^{1/n}$.
- We show how to obtain such an optimal reduction.

Acknowledgement

We would like to thank Tibor Jager for insightful discussions on the first version of this work in 2020.



We would also like to thank the anonymous reviewers ↑ from Eurocrypt 2021, Crypto 2021, and Eurocrypt 2022 for their constructive comments.