Updatable Signatures and Message Authentication Codes

Valerio Cini[‡], Sebastian Ramacher[‡], Daniel Slamanig[‡], Christoph Striecks[‡], Erkan Tairi[§]

PKC 2021, May 10

‡ AIT Austrian Institute of Technology

§ Technische Universität Wien

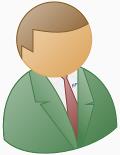
Motivation

- Rotate keys and update signatures/MACs to the new key (using a compact token),
- Previous work on Updatable Encryption (e.g., [Bon+13] and [LT18]),
- Equally important in context of signatures and MACs to follow good key management practices (e.g., key-rotation in software distribution).

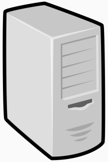
Our Framework

Definition

epoch $\boldsymbol{e}$

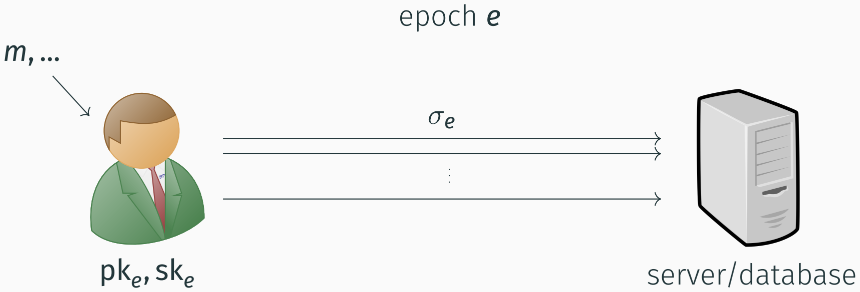


$\mathsf{pk}_e, \mathsf{sk}_e$



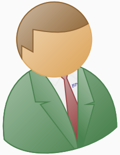
server/database

Definition

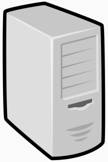

epoch e
$m, \ldots$
σ_e
$\mathsf{pk}_e, \mathsf{sk}_e$
server/database

Definition

epoch $\boldsymbol{e}$



$\mathsf{pk}_e, \mathsf{sk}_e$

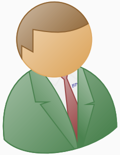


σ_e

server/database

Definition

epoch e

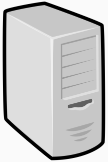


~~$\mathsf{pk}_e, \mathsf{sk}_e$~~

pk_e, sk_e

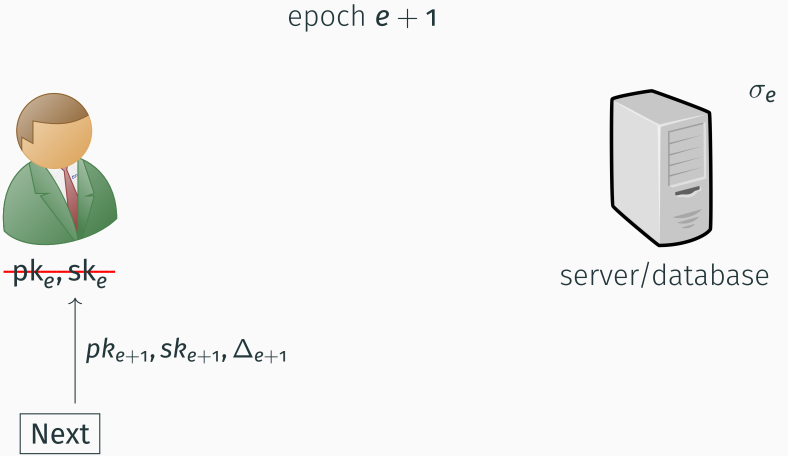
Next

σ_e



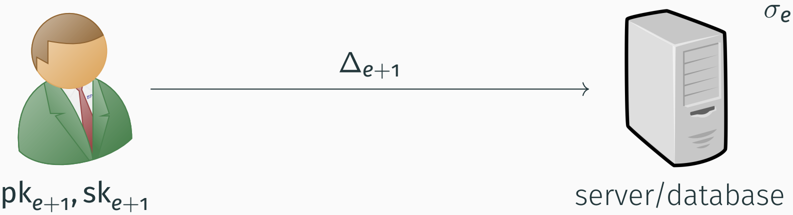
server/database

Definition


epoch $e+1$
σ_e
pk_e, sk_e
server/database
$pk_{e+1}, sk_{e+1}, \Delta_{e+1}$
Next

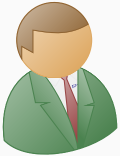
Definition

epoch $e + 1$

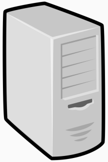


Definition

epoch $e+1$



$\mathsf{pk}_{e+1}, \mathsf{sk}_{e+1}$



server/database

σ_e → σ_{e+1}

Update

Security

We introduced two security notions:

- existential unforgeability under chosen-message attack (UX-EUF-CMA),
- unlinkable updates under chosen-message attack (UX-UU-CMA),

for $\mathsf{X} \in \{\text{MAC}, \text{S}\}$.

Leakage Profile [LT18]

We use the concept of a leakage profile originally defined, for updatable encryption, in [LT18], to capture key, token, and signature "leakage" that cannot be directly captured via oracles.

- Key-update inferences,
- Token inferences,
- Signature-update inferences,

Example of Leakage

epoch:	$e-5$	$e-4$	$e-3$	$e-2$	$e-1$	e	$e+1$	$e+2$	$e+3$	$e+4$
keys:	k_{e-5}	**k_{e-4}**	k_{e-3}	**k_{e-2}**	k_{e-1}	k_e	**k_{e+1}**	k_{e+2}	k_{e+3}	k_{e+4}
tokens:	**Δ_{e-4}**	Δ_{e-3}	**Δ_{e-2}**	Δ_{e-1}	**Δ_e**	Δ_{e+1}	**Δ_{e+2}**	Δ_{e+3}	Δ_{e+4}	**Δ_{e+5}**
signature:	σ_{e-5}	**σ_{e-4}**	σ_{e-3}	σ_{e-2}	σ_{e-1}	σ_e	**σ_{e+1}**	σ_{e+2}	**σ_{e+3}**	σ_{e+4}

Figure 1: Example of directly obtained (green) and inferable information (blue) for UX schemes.

Example of Leakage

epoch:	$e-5$	$e-4$	$e-3$	$e-2$	$e-1$	e	$e+1$	$e+2$	$e+3$	$e+4$
keys:	k_{e-5}	k_{e-4}	k_{e-3}	k_{e-2}	k_{e-1}	k_e	k_{e+1}	k_{e+2}	k_{e+3}	k_{e+4}
tokens:	Δ_{e-4}	Δ_{e-3}	Δ_{e-2}	Δ_{e-1}	Δ_e	Δ_{e+1}	Δ_{e+2}	Δ_{e+3}	Δ_{e+4}	Δ_{e+5}
signature:	σ_{e-5}	σ_{e-4}	σ_{e-3}	σ_{e-2}	σ_{e-1}	σ_e	σ_{e+1}	σ_{e+2}	σ_{e+3}	σ_{e+4}

Figure 1: Example of directly obtained (green) and inferable information (blue) for UX schemes.

Example of Leakage

epoch:	$e-5$	$e-4$	$e-3$	$e-2$	$e-1$	e	$e+1$	$e+2$	$e+3$	$e+4$
keys:	k_{e-5}	k_{e-4}	k_{e-3}	k_{e-2}	k_{e-1}	k_e	k_{e+1}	k_{e+2}	k_{e+3}	k_{e+4}
tokens:	Δ_{e-4}	Δ_{e-3}	Δ_{e-2}	Δ_{e-1}	Δ_e	Δ_{e+1}	Δ_{e+2}	Δ_{e+3}	Δ_{e+4}	Δ_{e+5}
signature:	σ_{e-5}	σ_{e-4}	σ_{e-3}	σ_{e-2}	σ_{e-1}	σ_e	σ_{e+1}	σ_{e+2}	σ_{e+3}	σ_{e+4}

Figure 1: Example of directly obtained (green) and inferable information (blue) for UX schemes.

Example of Leakage

epoch:	$e-5$	$e-4$	$e-3$	$e-2$	$e-1$	e	$e+1$	$e+2$	$e+3$	$e+4$
keys:	k_{e-5}	k_{e-4}	k_{e-3}	k_{e-2}	k_{e-1}	k_e	k_{e+1}	k_{e+2}	k_{e+3}	k_{e+4}
tokens:	Δ_{e-4}	Δ_{e-3}	Δ_{e-2}	Δ_{e-1}	Δ_e	Δ_{e+1}	Δ_{e+2}	Δ_{e+3}	Δ_{e+4}	Δ_{e+5}
signature:	σ_{e-5}	σ_{e-4}	σ_{e-3}	σ_{e-2}	σ_{e-1}	σ_e	σ_{e+1}	σ_{e+2}	σ_{e+3}	σ_{e+4}

Figure 1: Example of directly obtained (green) and inferable information (blue) for UX schemes.

Example of Leakage

epoch:	$e-5$	$e-4$	$e-3$	$e-2$	$e-1$	e	$e+1$	$e+2$	$e+3$	$e+4$
keys:	k_{e-5}	k_{e-4}	k_{e-3}	k_{e-2}	k_{e-1}	k_e	k_{e+1}	k_{e+2}	k_{e+3}	k_{e+4}
tokens:	Δ_{e-4}	Δ_{e-3}	Δ_{e-2}	Δ_{e-1}	Δ_e	Δ_{e+1}	Δ_{e+2}	Δ_{e+3}	Δ_{e+4}	Δ_{e+5}
signature:	σ_{e-5}	σ_{e-4}	σ_{e-3}	σ_{e-2}	σ_{e-1}	σ_e	σ_{e+1}	σ_{e+2}	σ_{e+3}	σ_{e+4}

Figure 1: Example of directly obtained (green) and inferable information (blue) for UX schemes.

Constructions

Outline

- US from Key-Homomorphic Signatures [DS19],

Outline

- US from Key-Homomorphic Signatures [DS19],
- Lattice-based candidate US construction [GPV08],

Outline

- US from Key-Homomorphic Signatures [DS19],
- Lattice-based candidate US construction [GPV08],
- UMAC from “almost” key-homomorphic PRFs [Bon+13],

Outline

- US from Key-Homomorphic Signatures [DS19],
- Lattice-based candidate US construction [GPV08],
- UMAC from "almost" key-homomorphic PRFs [Bon+13],
- Security Proof Ideas.

Key-Homomorphic Signatures [DS19] (1/2)

Definition (Secret Key to Public Key Homomorphism [DS19])

Let Σ be a signature scheme, where secret and public key elements live in groups $(\mathbb{H}, +)$ and $(\mathbb{E}, \cdot)$ respectively. A Secret Key to Public Key Homomorphism is a map $\mu : \mathbb{H} \to \mathbb{E}$, such that:

- $\mu(sk + sk') = \mu(sk) \cdot \mu(sk')$ for all $sk, sk' \in \mathbb{H}$,
- $pk = \mu(sk)$ for all $(sk, pk) \leftarrow \mathsf{KeyGen}(\lambda)$.

Key-Homomorphic Signatures [DS19] (1/2)

Definition (Secret Key to Public Key Homomorphism [DS19])

Let Σ be a signature scheme, where secret and public key elements live in groups $(\mathbb{H}, +)$ and $(\mathbb{E}, \cdot)$ respectively. A Secret Key to Public Key Homomorphism is a map $\mu : \mathbb{H} \to \mathbb{E}$, such that:

- $\mu(sk + sk') = \mu(sk) \cdot \mu(sk')$ for all $sk, sk' \in \mathbb{H}$,
- $pk = \mu(sk)$ for all $(sk, pk) \leftarrow \mathsf{KeyGen}(\lambda)$.

Example: DL setting $(\mathbb{G}, p, g)$

$$sk \leftarrow \mathbb{Z}_p, pk = g^{sk} \qquad \mu : \begin{cases} \mathbb{Z}_p \to \mathbb{G} \\ k \mapsto g^k \end{cases}$$

Definition (Key-Homomorphic Signatures [DS19])

A signature scheme is called key-homomorphic, if it provides a secret key to public key homomorphism and an additional PPT algorithm **Adapt**, such that for all $\Delta \in \mathbb{H}$ and all $(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$, all messages $M \in \mathcal{M}$ and all σ with $\mathsf{Ver}(pk, M, \sigma) = 1$ and $(pk', \sigma') \leftarrow \mathsf{Adapt}(\mathsf{pk}, M, \sigma, \Delta)$, it holds that

$$\Pr[\mathsf{Ver}(pk', M, \sigma') = 1] = 1 \quad \wedge \quad pk' = \mu(\Delta) \cdot pk.$$

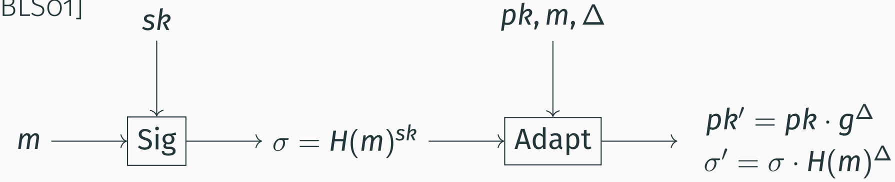
Key-Homomorphic Signatures [DS19] (2/2)

Definition (Key-Homomorphic Signatures [DS19])

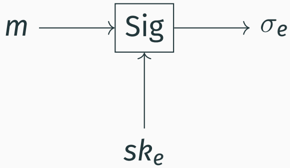
A signature scheme is called key-homomorphic, if it provides a secret key to public key homomorphism and an additional PPT algorithm **Adapt**, such that for all $\Delta \in \mathbb{H}$ and all $(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$, all messages $M \in \mathcal{M}$ and all σ with $\mathsf{Ver}(pk, M, \sigma) = 1$ and $(pk', \sigma') \leftarrow \mathsf{Adapt}(\mathsf{pk}, M, \sigma, \Delta)$, it holds that

$$\Pr[\mathsf{Ver}(pk', M, \sigma') = 1] = 1 \quad \wedge \quad pk' = \mu(\Delta) \cdot pk.$$

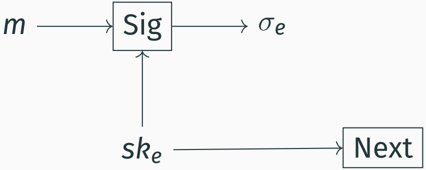
Example: [BLS01]



KH-based construction

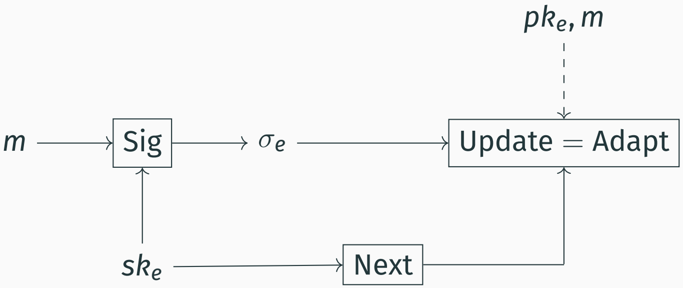


KH-based construction



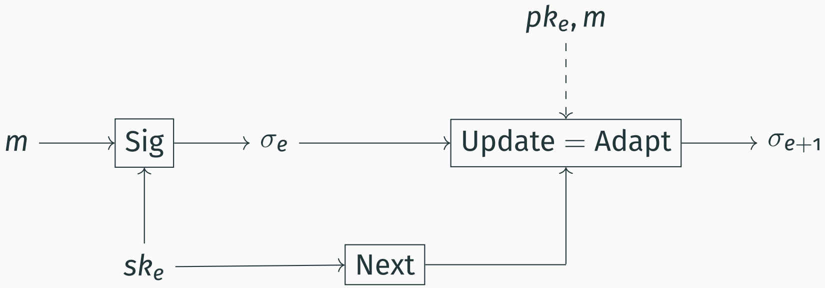
$\Delta_{e+1} \leftarrow \mathbb{H};$ $pk_{e+1} := pk_e \cdot \mu(\Delta_{e+1})$

$sk_{e+1} := sk_e + \Delta_{e+1}$



$$\Delta_{e+1} \leftarrow \mathbb{H}; \quad \begin{aligned} pk_{e+1} &:= pk_e \cdot \mu(\Delta_{e+1}) \\ sk_{e+1} &:= sk_e + \Delta_{e+1} \end{aligned}$$

KH-based construction



$$\Delta_{e+1} \leftarrow \mathbb{H}; \quad \begin{array}{l} pk_{e+1} := pk_e \cdot \mu(\Delta_{e+1}) \\ sk_{e+1} := sk_e + \Delta_{e+1} \end{array}$$

Lattice-based candidate construction (1/2)

We start from the well-known GPV signature scheme of Gentry et al. [GPV08].

Lattice-based candidate construction (1/2)

We start from the well-known GPV signature scheme of Gentry et al. [GPV08].

Real: pk_e

Lattice-based candidate construction (1/2)

We start from the well-known GPV signature scheme of Gentry et al. [GPV08].

Real: $pk_e \quad = \quad H(m)$

Lattice-based candidate construction (1/2)

We start from the well-known GPV signature scheme of Gentry et al. [GPV08].

Real: $$pk_e \cdot \sigma_e = H(m)$$

Lattice-based candidate construction (1/2)

We start from the well-known GPV signature scheme of Gentry et al. [GPV08].

Simulation: pk_e

Lattice-based candidate construction (1/2)

We start from the well-known GPV signature scheme of Gentry et al. [GPV08].

Simulation: $pk_e \cdot \sigma_e$

Lattice-based candidate construction (1/2)

We start from the well-known GPV signature scheme of Gentry et al. [GPV08].

Simulation: $$pk_e \cdot \sigma_e = H(m)$$

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Next :

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Next : pk_{e+1}

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Next : $pk_{e+1} = pk_e$

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Next : $pk_{e+1} \cdot \Delta_{e+1} = pk_e$

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Update :

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Update :

σ_e

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Update : $\Delta_{e+1} \cdot \sigma_e$

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Update : $$\underbrace{\Delta_{e+1} \cdot \sigma_e}_{\sigma_{e+1}}$$

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

Ver :

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

$$\text{Ver}: \quad pk_{e+1} \cdot \Delta_{e+1} \cdot \sigma_e \stackrel{?}{=} H(m)$$

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

$$\text{Ver} : \underbrace{pk_{e+1} \cdot \Delta_{e+1}}_{pk_e} \cdot \sigma_e \stackrel{?}{=} H(m)$$

Lattice-based candidate construction (2/2)

By using methods inspired by the lattice-based proxy re-signature approach of Fan and Liu [FL19], we obtain a candidate lattice-based US signature.

$$\text{Ver} : \underbrace{pk_{e+1} \cdot \Delta_{e+1}}_{pk_e} \cdot \sigma_e \overset{\checkmark}{=} H(m)$$

UMAC from (almost) key-homomorphic PRFs

Let F be a secure PRF, then we can construct a canonical MAC from it

UMAC from (almost) key-homomorphic PRFs

Let F be a secure PRF, then we can construct a canonical MAC from it

$$\mathsf{Sig}\colon m \longrightarrow \boxed{F(k,\cdot)} \longrightarrow \sigma$$

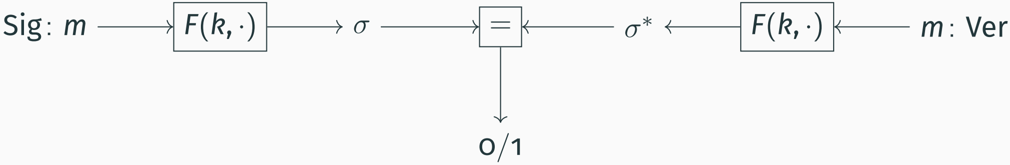
UMAC from (almost) key-homomorphic PRFs

Let F be a secure PRF, then we can construct a canonical MAC from it

$$\mathsf{Sig}: m \longrightarrow \boxed{F(k,\cdot)} \longrightarrow \sigma \qquad\qquad \sigma^* \longleftarrow \boxed{F(k,\cdot)} \longleftarrow m: \mathsf{Ver}$$

UMAC from (almost) key-homomorphic PRFs

Let F be a secure PRF, then we can construct a canonical MAC from it



UMAC from (almost) key-homomorphic PRFs

Definition (Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $F: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a key-homomorphic PRF if F is a secure PRF and for every key $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x)$$

UMAC from (almost) key-homomorphic PRFs

Definition (Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $F\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a key-homomorphic PRF if F is a secure PRF and for every key $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x)$$

$$\text{Sig}\colon m \longrightarrow \boxed{F(k_1, \cdot)} \longrightarrow \sigma_1$$

UMAC from (almost) key-homomorphic PRFs

Definition (Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $F\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a key-homomorphic PRF if F is a secure PRF and for every key $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x)$$

$$\text{Sig}\colon\ m \longrightarrow \boxed{F(k_1, \cdot)} \longrightarrow \sigma_1$$

$$\text{Update}\colon\ m \longrightarrow \boxed{F(\Delta_2, \cdot)}$$

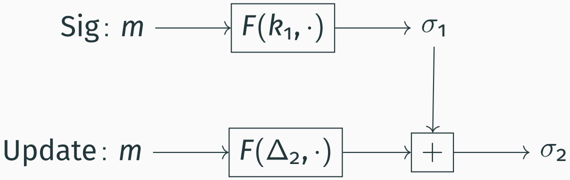
$$\Delta_2 = k_2 \oplus -k_1$$

UMAC from (almost) key-homomorphic PRFs

Definition (Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $\boldsymbol{F}\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a key-homomorphic PRF if $\boldsymbol{F}$ is a secure PRF and for every key $\boldsymbol{k_1}, \boldsymbol{k_2} \in \mathcal{K}$ and every input $\boldsymbol{x} \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x)$$



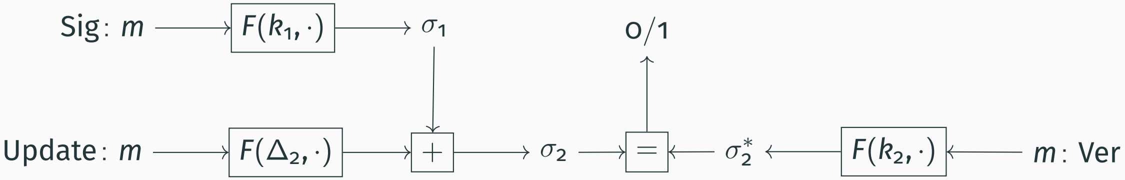
$\Delta_2 = k_2 \oplus -k_1$ $\quad\quad$ $\sigma_2 = F(k_2, m)$

UMAC from (almost) key-homomorphic PRFs

Definition (Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $F \colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a key-homomorphic PRF if F is a secure PRF and for every key $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x)$$



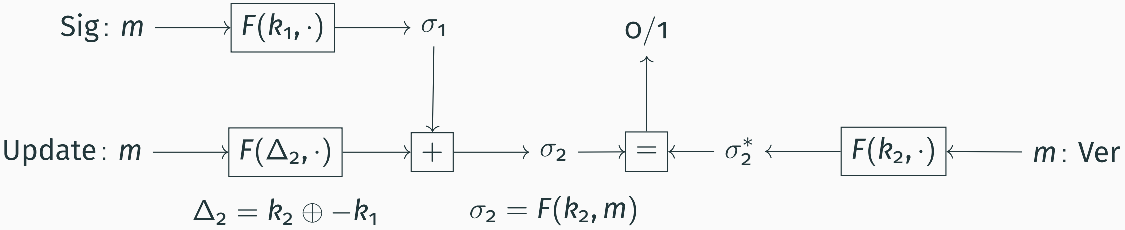
$\Delta_2 = k_2 \oplus -k_1$ $\sigma_2 = F(k_2, m)$

UMAC from (almost) key-homomorphic PRFs

Definition (Almost Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $F\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is an almost key-homomorphic PRF if F is a secure PRF and for every key $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x) \ + e$$

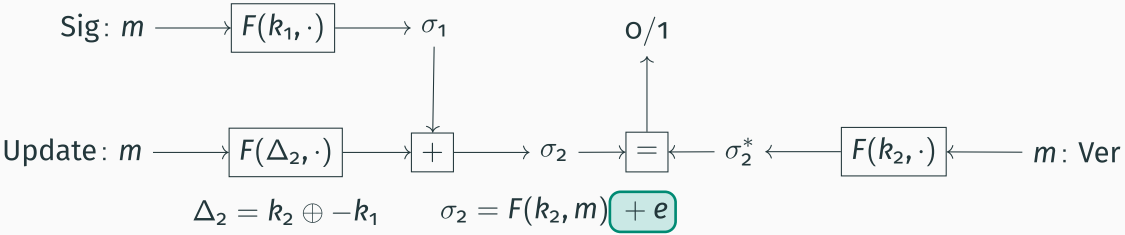


UMAC from (almost) key-homomorphic PRFs

Definition (Almost Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $F \colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is an almost key-homomorphic PRF if F is a secure PRF and for every key $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x) \; + e$$

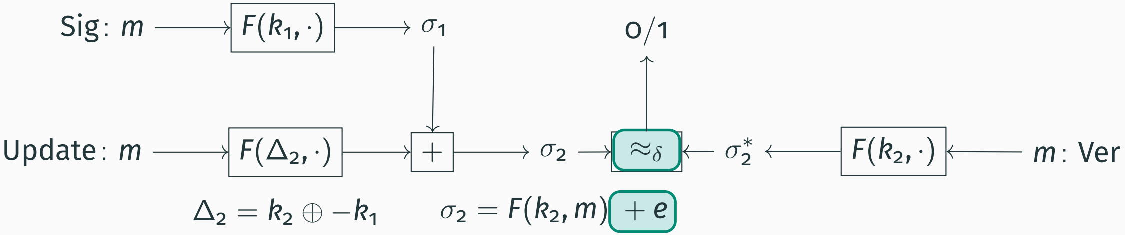


UMAC from (almost) key-homomorphic PRFs

Definition (Almost Key-Homomorphic PRFs [Bon+13])

Let $(\mathcal{K}, \oplus), (\mathcal{Y}, +)$ be groups. Then, a keyed function $F: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is an almost key-homomorphic PRF if F is a secure PRF and for every key $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) + F(k_2, x) = F(k_1 \oplus k_2, x) \; + e$$



Security Proof

- Reduce UX-EUF-CMA to EUF-CMA of X for $\mathsf{X} \in \{\text{MAC}, \text{S}\}$

Security Proof

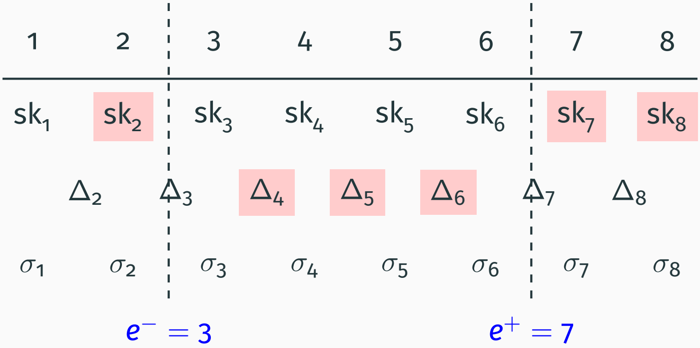
- Reduce UX-EUF-CMA to EUF-CMA of X for X $\in$ {MAC, S}
- Key insulation technique of Klooß et al. [KLR19] (i.e., region $[e^-, e^+[$):
  - No key inside the insulated region is corrupted
  - Tokens “on” the borders of the insulated region are not corrupted
  - All tokens inside the insulated region are corrupted

Security Proof

1	2	3	4	5	6	7	8
sk_1	sk_2	sk_3	sk_4	sk_5	sk_6	sk_7	sk_8
	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	Δ_8
σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8

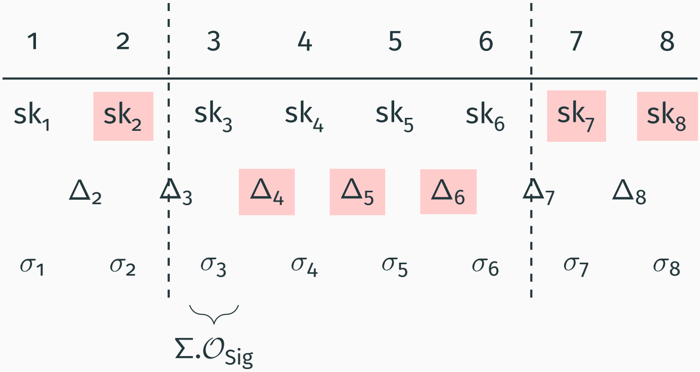
- Reduce U**X**-EUF-CMA to EUF-CMA of **X** for **X** $\in$ {MAC, S}
- Key insulation technique of Klooß et al. [KLR19] (i.e., region $[\boldsymbol{e}^-, \boldsymbol{e}^+[$):
  - No key inside the insulated region is corrupted
  - Tokens "on" the borders of the insulated region are not corrupted
  - All tokens inside the insulated region are corrupted

Security Proof



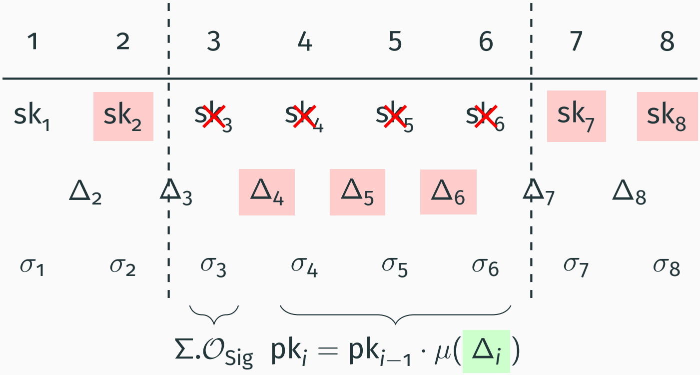
- Reduce UX-EUF-CMA to EUF-CMA of X for X $\in$ {MAC, S}
- Key insulation technique of Klooß et al. [KLR19] (i.e., region $[e^-, e^+[$):
  - No key inside the insulated region is corrupted
  - Tokens “on” the borders of the insulated region are not corrupted
  - All tokens inside the insulated region are corrupted

Security Proof



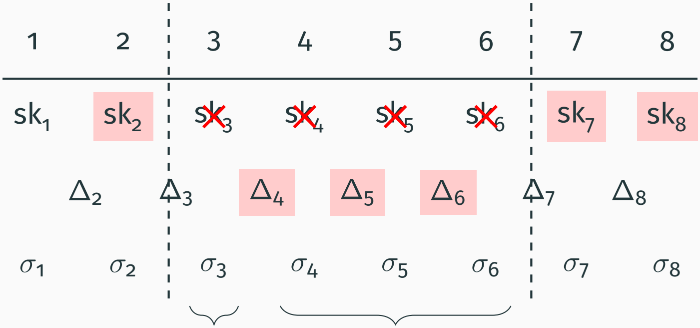
- Associate the EUF-CMA challenger of Σ to an epoch within region (e.g., to $\boldsymbol{e}^-$)

Security Proof



- Associate the EUF-CMA challenger of Σ to an epoch within region (e.g., to $\boldsymbol{e}^{-}$)
- Set keys for each epoch within the insulated region (using random $\Delta_i \leftarrow_{\$} \mathcal{T}$)

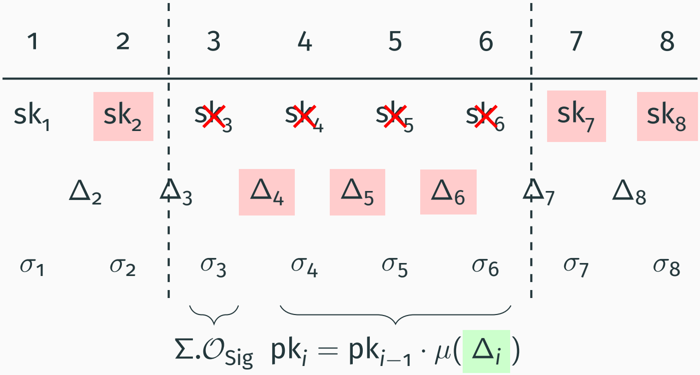
Security Proof



$\Sigma.\mathcal{O}_{\mathsf{Sig}}$ $\mathsf{pk}_i = \mathsf{pk}_{i-1} \cdot \mu(\Delta_i)$

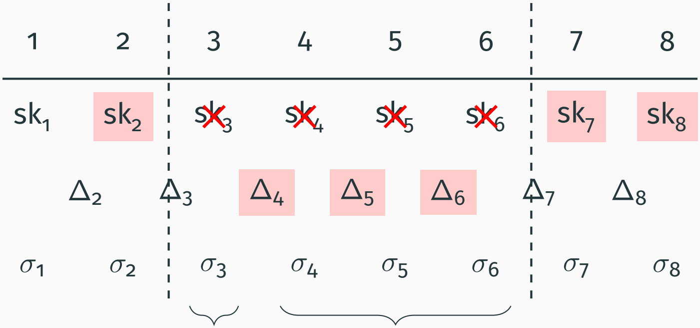
- Associate the EUF-CMA challenger of Σ to an epoch within region (e.g., to $\boldsymbol{e}^-$)
- Set keys for each epoch within the insulated region (using random $\Delta_i \leftarrow_\$ \mathcal{T}$)
- Use the EUF-CMA challenger of Σ and **Σ.Adapt** algorithm to answer queries

Security Proof



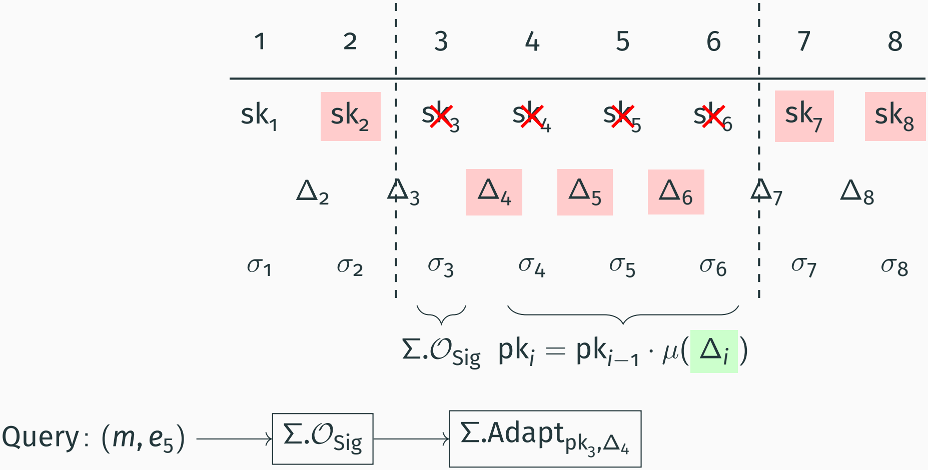
Query: (m, e_5)

Security Proof

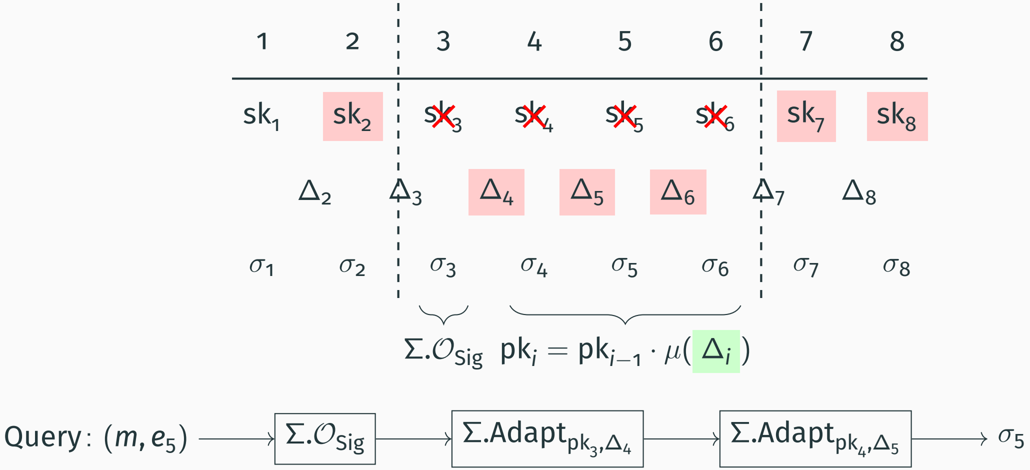


$\Sigma.\mathcal{O}_{\mathsf{Sig}}$ $\quad \mathsf{pk}_i = \mathsf{pk}_{i-1} \cdot \mu(\Delta_i)$

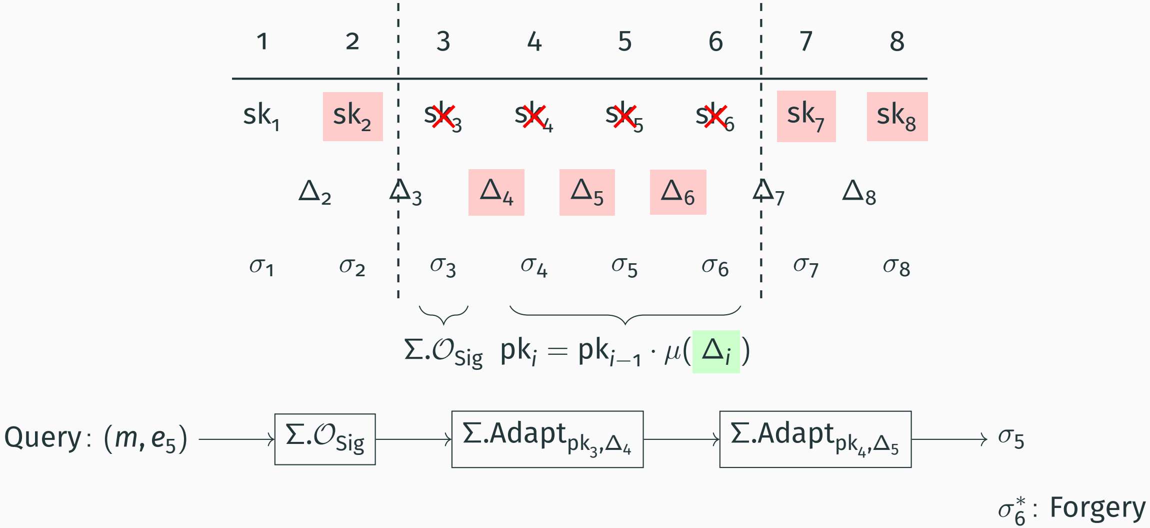
Query: $(m, e_5) \longrightarrow$ $\boxed{\Sigma.\mathcal{O}_{\mathsf{Sig}}}$

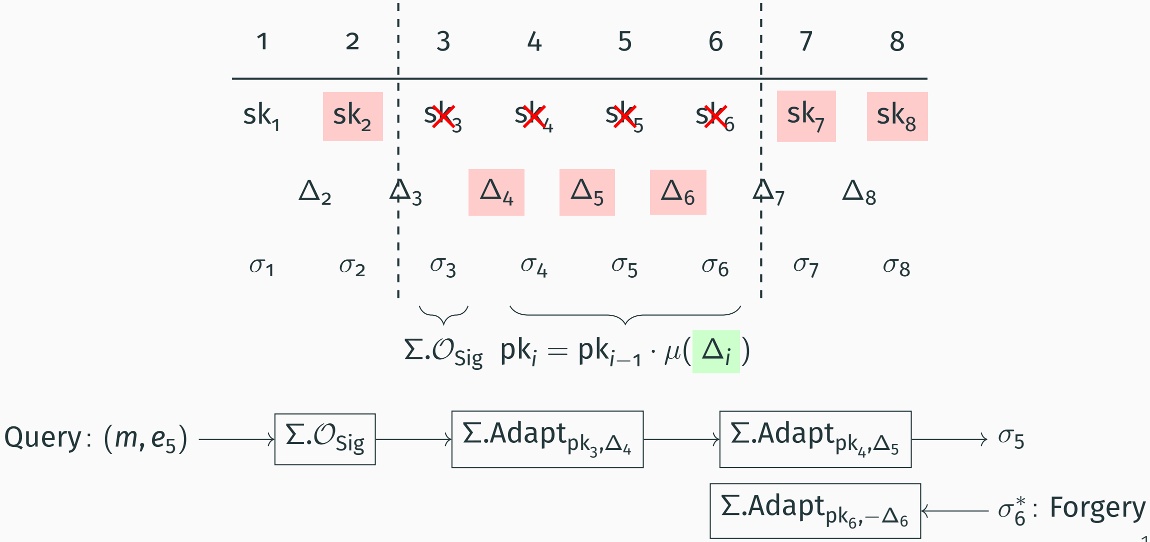

1 2 3 4 5 6 7 8
sk1 sk2 sk3 sk4 sk5 sk6 sk7 sk8
Δ2 Δ3 Δ4 Δ5 Δ6 Δ7 Δ8
σ1 σ2 σ3 σ4 σ5 σ6 σ7 σ8
Σ.OSig
pki = pki−1 · μ(Δi)
Query: (m, e5)
Σ.OSig
Σ.Adaptpk3,Δ4

Security Proof

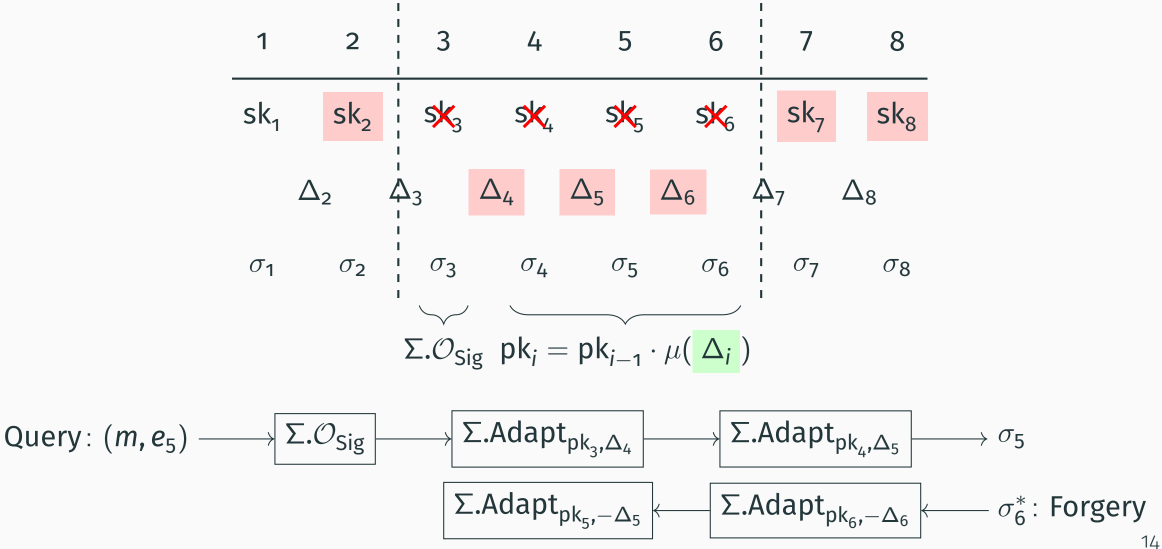


Security Proof

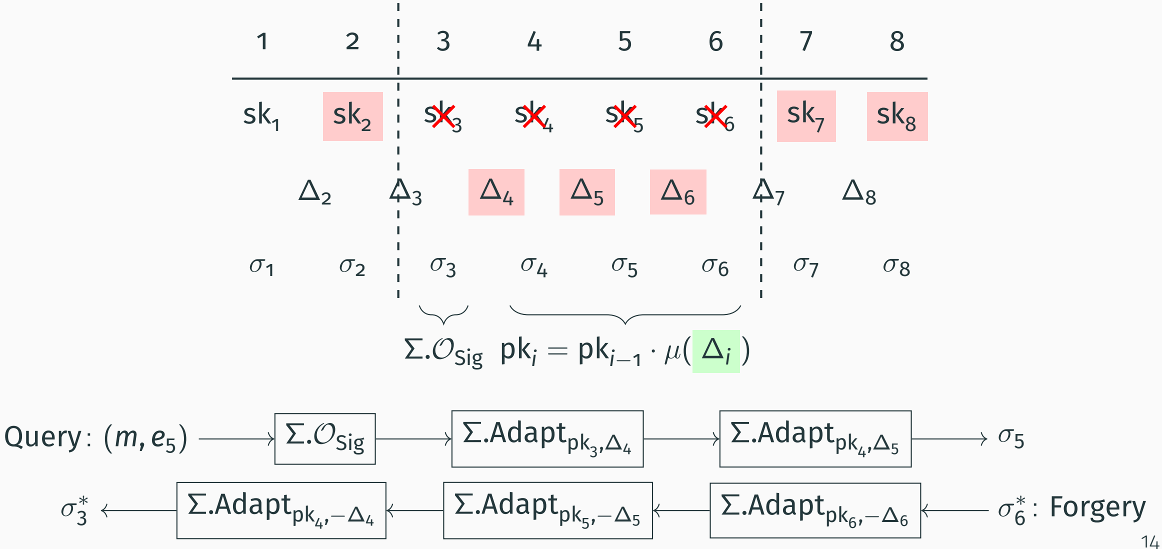



1 2 3 4 5 6 7 8
sk1 sk2 sk3 sk4 sk5 sk6 sk7 sk8
Δ2 Δ3 Δ4 Δ5 Δ6 Δ7 Δ8
σ1 σ2 σ3 σ4 σ5 σ6 σ7 σ8
Σ.OSig
pki = pki−1 · μ(Δi)
Query: (m, e5)
Σ.OSig
Σ.Adaptpk3,Δ4
Σ.Adaptpk4,Δ5
σ5
Σ.Adaptpk6,−Δ6
σ6*: Forgery

Security Proof



Security Proof



Overview and Instantiations

Updatable Signatures

Table 1: Overview of updatable signature schemes.

Scheme	Assumption	Model	UU-CMA	MD/MI	UB
BLS	co-CDH	RO	✓	MI	✓
BLS	co-CDH	RO	✓	MD	✓
PS	P-LRSW	GGM	✓	MI	✓
PS	P-LRSW	GGM	✓	MD	✓
Waters	co-CDH	SM	✓	MD	✓
GPV[1]	SIS	RO	✗	MI	T

[1]Provides US-EUF-CMA security only in a weakened model.

Table 2: Overview of updatable MAC schemes.

Scheme	Assumption	Model	UU-CMA	MD/MI	UB
BLMR (NPR) [Bon+13]	DDH	RO	✓	MD	✓
NPR	DDH	RO	✓	MI	✓
BEKS [Bon+20]	RLWE	RO	✓	MD	T
Kim [Kim20]	LWE	SM	✓	MD	T

Conclusion and Open Questions

Conclusion

- New cryptographic primitives, UMAC and US

Conclusion

- New cryptographic primitives, UMAC and US
- Generic constructions from KH-PRF and KH-Sig

Conclusion

- New cryptographic primitives, UMAC and US
- Generic constructions from KH-PRF and KH-Sig
- Message independent constructions

Conclusion

- New cryptographic primitives, UMAC and US
- Generic constructions from KH-PRF and KH-Sig
- Message independent constructions
- Post-quantum instantiations from lattices

Open Questions

- Construction of lattice-based US with full security?
- Concrete bounds for UMAC from almost KH-PRFs?

Thank you for your attention!

(full version of the paper available on ePrint: ia.cr/2021/365)

@cini_valerio @erkantairi

Supported by: FWF Der Wissenschaftsfonds. netidee SCIENCE KRAKEN SECREDAS Digital Privacy Safety & Security Cyber Security for Europe

References

[BLS01] D. Boneh, B. Lynn, and H. Shacham. “Short signatures from the Weil pairing”. In: *International conference on the theory and application of cryptology and information security*. Springer. 2001, pp. 514–532.

[Bon+13] D. Boneh et al. “Key homomorphic PRFs and their applications”. In: *Annual Cryptology Conference*. Springer. 2013, pp. 410–428.

[Bon+20] D. Boneh et al. “Improving speed and security in updatable encryption schemes”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2020, pp. 559–589.

[DS19] D. Derler and D. Slamanig. “Key-homomorphic signatures: definitions and applications to multiparty signatures and non-interactive zero-knowledge”. In: *Designs, Codes and Cryptography* 87.6 (2019), pp. 1373–1413.

[FL19] X. Fan and F.-H. Liu. “Proxy re-encryption and re-signatures from lattices”. In: *International Conference on Applied Cryptography and Network Security*. Springer. 2019, pp. 363–382.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 197–206.

[Kim20] S. Kim. "Key-homomorphic pseudorandom functions from LWE with small modulus". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2020, pp. 576–607.

[KLR19] M. Klooß, A. Lehmann, and A. Rupp. "(R)CCA Secure Updatable Encryption with Integrity Protection". In: *Advances in Cryptology – EUROCRYPT 2019*. Springer. 2019, pp. 68–99.

[LT18] A. Lehmann and B. Tackmann. "Updatable encryption with post-compromise security". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2018, pp. 685–716.