Efficient Sanitizable Signatures without Random Oracles

Russell W. F. Lai [1] Tao Zhang [1] Sherman S. M. Chow [1]
Dominique Schröder [2]

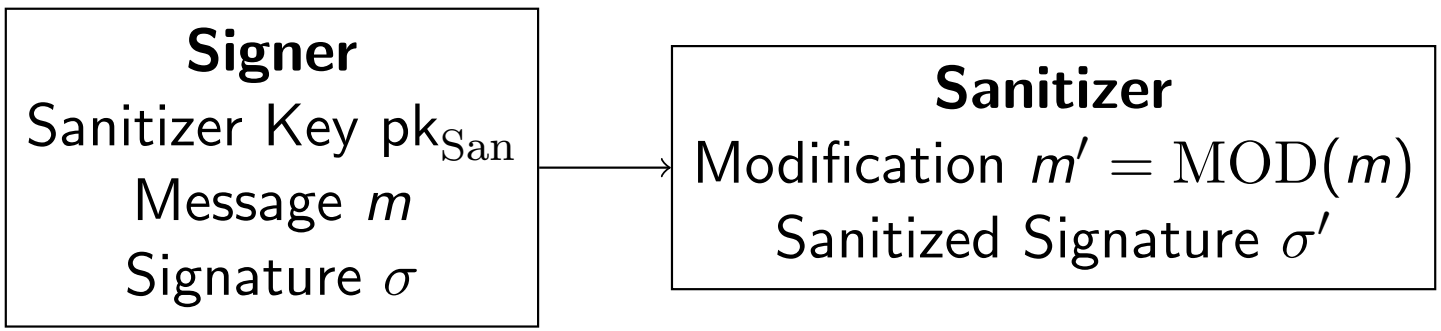
[1]Chinese University of Hong Kong, Hong Kong

[2]Friedrich-Alexander University Erlangen-Nürnberg, Germany

Table of Contents

Table of Contents

What are Sanitizable Signatures?



- MOD: Sanitize sensitive info before releasing to public
- Applications: Outsourced Database, Medical Records, Secure Routing, etc

Regular Signatures

- Signing: $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}_{\mathrm{Sig}}, m)$
- Verification: $b \leftarrow \mathsf{Vf}(\mathsf{pk}_{\mathrm{Sig}}, m, \sigma)$

Sanitizable Signatures

Malleability!

- Signing: $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}_{\mathrm{Sig}}, \mathsf{pk}_{\mathrm{San}}, m, \mathrm{ADM})$
  - ADM: Admissible functions
- Sanitizing: $(m', \sigma') \leftarrow \mathsf{San}(\mathsf{sk}_{\mathrm{San}}, \mathsf{pk}_{\mathrm{Sig}}, m, \sigma, \mathrm{MOD})$
  - MOD: Modification function, $m' = \mathrm{MOD}(m)$
  - $\mathrm{MOD} \in \mathrm{ADM}$ iff $\mathrm{FIX}_{\mathrm{ADM}}(m') = \mathrm{FIX}_{\mathrm{ADM}}(m)$
- Verification: $b \leftarrow \mathsf{Vf}(\mathsf{pk}_{\mathrm{Sig}}, \mathsf{pk}_{\mathrm{San}}, m, \sigma)$

Sanitizable Signatures

Controlled Malleability!

- Signing: $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}_{\mathrm{Sig}}, \mathsf{pk}_{\mathrm{San}}, m, \mathrm{ADM})$
  - ADM: Admissible functions
- Sanitizing: $(m', \sigma') \leftarrow \mathsf{San}(\mathsf{sk}_{\mathrm{San}}, \mathsf{pk}_{\mathrm{Sig}}, m, \sigma, \mathrm{MOD})$
  - MOD: Modification function, $m' = \mathrm{MOD}(m)$
  - $\mathrm{MOD} \in \mathrm{ADM}$ iff $\mathrm{FIX}_{\mathrm{ADM}}(m') = \mathrm{FIX}_{\mathrm{ADM}}(m)$
- Verification: $b \leftarrow \mathsf{Vf}(\mathsf{pk}_{\mathrm{Sig}}, \mathsf{pk}_{\mathrm{San}}, m, \sigma)$
- Proving: $\pi \leftarrow \mathsf{Prov}(\mathsf{sk}_{\mathrm{Sig}}, \mathsf{pk}_{\mathrm{San}}, m, \sigma)$
  - Signer proves or disproves that m is sanitized
- Judging: $d \leftarrow \mathsf{Jud}(\mathsf{pk}_{\mathrm{Sig}}, \mathsf{pk}_{\mathrm{San}}, m, \sigma, \pi)$
  - Judge whether σ is sanitized ($d = \mathrm{San}$) or not ($d = \mathrm{Sig}$)

Immutability

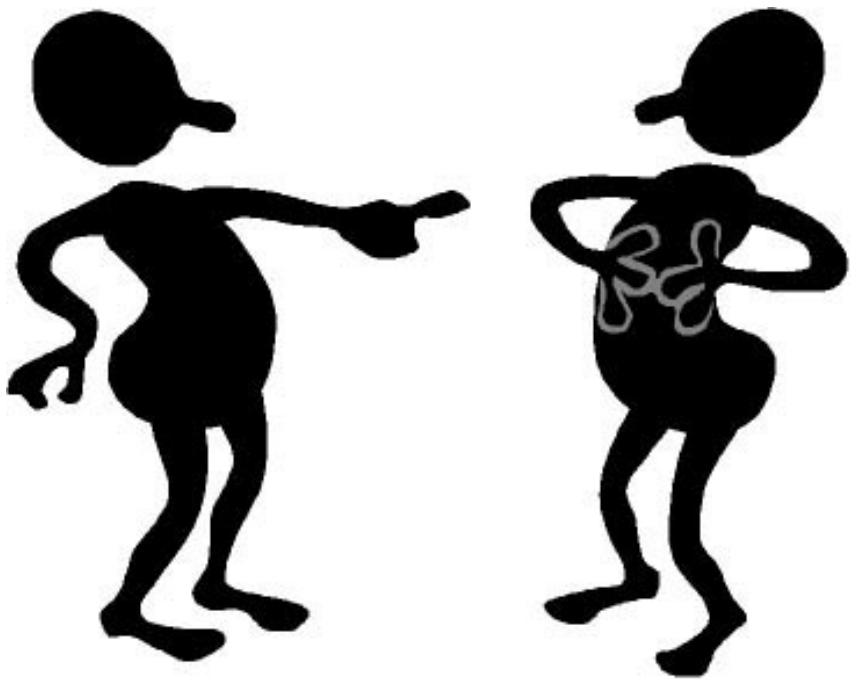
Sanitizer cannot change fixed part.

- $\mathrm{FIX}_{\mathrm{ADM}}(m') = \mathrm{FIX}_{\mathrm{ADM}}(m)$

Accountability

- Signer Accountability: Signer cannot accuse sanitizer.
- Sanitizer Accountability: Sanitizer cannot accuse signer.

"No! It's not me!"



Transparency

Sanitized and fresh signatures are indistinguishable.

"Who signed this?"

$$\begin{array}{c} \mathsf{Sig}(m') \to \sigma \\ \wr\!\wr \\ \mathsf{San}(m, \mathrm{Mod}) \to (\sigma', m') \end{array}$$

Unlinkability (Optional Property)

Sanitized signature from different sources are indistinguishable.

"Which is the original?"

$$\begin{array}{c} \mathsf{San}(m_1, \mathrm{Mod}_1) \to (\sigma_1', m') \\ \wr\!\wr \\ \mathsf{San}(m_2, \mathrm{Mod}_2) \to (\sigma_2', m') \end{array}$$

Existing Schemes and Challenges

Selected schemes satisfying the above security properties:

Scheme	Technique	Security	Efficiency	Assumption	Model
[ACdT05] (ESORICS) [BFF+09] (PKC)	Chameleon Hash	Basic	High	Static	ROM
[BFLS10] (PKC) + [FY05]	Special Group Signatures	Unlinkable	Low	GGM	ROM
[BFLS10] (PKC) + [Gro07] (w/ specific modification)	Special Group Signatures	Unlinkable	Moderate	GGM	Standard
[FKM+16] (PKC)	Signatures w/ Rerandomizable Keys	Unlinkable	High	Static	ROM
Construction I	(New) Re-randomizable Tagging	Basic	High	Static	Standard
Construction II	Accountable Ring Signatures	Unlinkable	Moderate	q-type	CRS

- Achieve all security notions
  (Challenging even w/o unlinkability)
- Avoid idealized models
  (random oracle model (ROM), generic group model (GGM))
- Achieve reasonable efficiency

Contribution

Scheme	Technique	Security	Efficiency	Assumption	Model
Construction I	(New) Re-randomizable Tagging	Basic	High	Static	Standard
Construction II	Accountable Ring Signatures	Unlinkable	Moderate	q-type	CRS

- Two Constructions without Random Oracles
- Re-randomizable Tagging (New):
  - Capture accountability of sanitizable signatures
  - Double Trapdoor Chameleon Hash +
  - Lattice Trapdoor Functions
- Accountable Ring Signatures:
  - Constant signature size (w.r.t. ring size)
  - Constant-size Ring Signatures +
  - Structure Preserving Encryption

Table of Contents

Intuition

Core of Sanitizable Signatures besides Signing?

- A cryptographic object associated to a message
- Once created by the signer
  $\rightarrow$ Can be changed / re-randomized by sanitizer many times
- Signer can (dis)prove authorship

Modular Approach

- Formulate re-randomizable tag to capture these properties!
- Re-randomizable Tag + Signatures $\rightarrow$ Sanitizable Signatures

Basic Construction from Re-randomizable Tagging

$$\sigma = (\sigma_{\mathrm{FIX}}, \tau)$$
$$m_{\mathrm{FIX}} = (\mathrm{FIX}_{\mathrm{ADM}}(m), \mathsf{pk}_{\mathrm{San}})$$

- σ_{FIX}: Signature of m_{FIX} under signer's secret key $\mathsf{sk}_{\mathrm{Sig}}$
- τ: Tag of m with "re-randomizable" property (sanitizable)

Re-randomizable Tagging Scheme

$\sigma = (\sigma_{\mathrm{FIX}}, \tau)$

τ: Re-randomizable Tag of m

- Captures accountability and transparency

Re-randomizable Tagging Scheme

$\sigma = (\sigma_{\mathrm{FIX}}, \tau)$

τ: Re-randomizable Tag of m

- Captures accountability and transparency
- Transparency:
  - Sanitizer can re-tag τ for m to τ' for m' many times
  - Fresh and re-randomized tags look the same to public

Re-randomizable Tagging Scheme

$\sigma = (\sigma_{\mathrm{FIX}}, \tau)$

τ: Re-randomizable Tag of m

- Captures accountability and transparency
- Transparency:
  - Sanitizer can re-tag τ for m to τ' for m' many times
  - Fresh and re-randomized tags look the same to public
- Accountability:
  - Fresh tags embed secrets
  - Re-randomization destroys embedded secret
  - Signer proves authorship by revealing embedded secret

Ingredients

- Pseudorandom Function
- Pseudorandom Generator
- Digital Signatures

Ingredients

- Pseudorandom Function
- Pseudorandom Generator
- Digital Signatures
- Extractable Public Key Encryption
  $m \leftarrow \mathsf{Ext}(\mathsf{pk}, c, r)$, $r =$ encryption randomness

Ingredients

- Pseudorandom Function
- Pseudorandom Generator
- Digital Signatures
- Extractable Public Key Encryption
  $m \leftarrow \mathsf{Ext}(\mathsf{pk}, c, r)$, $r =$ encryption randomness
- Chameleon Hash Function:
  $\mathit{hash} \leftarrow \mathsf{CH}(\mathsf{pk}, \mathit{message}, \mathit{randomness})$
  $\mathit{randomness}' \leftarrow \mathsf{CH}^{-1}(\mathsf{sk}, \mathit{message}, \mathit{randomness}, \mathit{message}')$

Ingredients

- Pseudorandom Function
- Pseudorandom Generator
- Digital Signatures
- Extractable Public Key Encryption
  $m \leftarrow \mathsf{Ext}(\mathsf{pk}, c, r)$, $r =$ encryption randomness
- Chameleon Hash Function:
  $\mathit{hash} \leftarrow \mathsf{CH}(\mathsf{pk}, \mathit{message}, \mathit{randomness})$
  $\mathit{randomness}' \leftarrow \mathsf{CH}^{-1}(\mathsf{sk}, \mathit{message}, \mathit{randomness}, \mathit{message}')$
- "Special" Tag-based Trapdoor Function:
  $\mathit{image} \leftarrow \mathsf{TDF}(\mathsf{pk}, \mathit{tag}, \mathit{pre\text{-}image})$
  $\mathit{pre\text{-}image} \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}, \mathit{tag}', \mathit{image})$

Ingredients

- Pseudorandom Function
- Pseudorandom Generator
- Digital Signatures
- Extractable Public K
  $m \leftarrow \mathsf{Ext}(\mathsf{pk}, c, r)$, r
- Chameleon Hash Function:
  $hash \leftarrow \mathsf{CH}(\mathsf{pk}, message, randomness)$
  $randomness' \leftarrow \mathsf{CH}^{-1}(\mathsf{sk}, message, randomness, message')$
- "Special" Tag-based Trapdoor Function:
  $image \leftarrow \mathsf{TDF}(\mathsf{pk}, tag, pre\text{-}image)$
  $pre\text{-}image \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}, tag', image)$

We mistakenly removed this in proceeding version!

Ingredients

- Chameleon Hash Function:
  - Given Collision $\Longrightarrow$ Recover Secret Key

Ingredients

- Chameleon Hash Function:
  - Given Collision $\Longrightarrow$ Recover Secret Key
- "Special" Tag-based Trapdoor Function:
  - Collision-Resistant even if given access to inversion oracle
  - *i.e.*, CR under Selective-Tag Adaptive-Image Attack
  - Realizable from lattice-based trapdoor functions

Intuition

- Tagging:
  - Randomness ρ_1, ρ_2
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $\tau := (\rho_1, \rho_2)$
  - (y binds all sanitized signatures from same source)

Intuition

- Tagging:
  - Randomness ρ_1, ρ_2
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $\tau := (\rho_1, \rho_2)$
  - (y binds all sanitized signatures from same source)
- Re-Tagging:
  - Randomness ρ'_1
  - $\mu' \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m'; \rho'_1)$
  - $\rho'_2 \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}_{\mathrm{San}}, \mu', y)$
  - $\tau' := (\rho'_1, \rho'_2)$
  - (Check: $y = \mathsf{TDF}(\mu, \rho_2) = \mathsf{TDF}(\mu', \rho'_2)$)

Construction

- Tagging:
  - Randomness ρ_1, ρ_2
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $\tau := (\rho_1, \rho_2)$
- Re-Tagging:
  - Randomness ρ_1'
  - $\mu' \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m'; \rho_1')$
  - $\rho_2' \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}_{\mathrm{San}}, \mu', y)$
  - $\tau := (\rho_1', \rho_2')$

Construction

- Tagging:
  - Randomness r_1, r_2
  - $\rho_1 \leftarrow \mathsf{PRG}(r_1)$, $\rho_2 \leftarrow \mathsf{PRG}(r_2)$
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $\tau := (\rho_1, \rho_2)$
- Re-Tagging:
  - Randomness ρ_1'
  - $\mu' \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m'; \rho_1')$
  - $\rho_2' \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}_{\mathrm{San}}, \mu', y)$
  - $\tau := (\rho_1', \rho_2')$

Construction

- Tagging:
  - Randomness q_1, q_2
  - $r_1 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_1)$, $r_2 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_2)$
  - $\rho_1 \leftarrow \mathsf{PRG}(r_1)$, $\rho_2 \leftarrow \mathsf{PRG}(r_2)$
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $\tau := (\rho_1, \rho_2, q_1, q_2)$
- Re-Tagging:
  - Randomness ρ_1'
  - $\mu' \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m'; \rho_1')$
  - $\rho_2' \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}_{\mathrm{San}}, \mu', y)$
  - $\tau := (\rho_1', \rho_2', q_1, q_2)$

Construction

- Tagging:
  - Randomness q_1, q_2, q_3
  - $r_1 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_1)$, $r_2 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_2)$, $r_3 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_3)$
  - $\rho_1 \leftarrow \mathsf{PRG}(r_1)$, $\rho_2 \leftarrow \mathsf{PRG}(r_2)$
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $c \leftarrow \mathsf{Enc}(\mathsf{pk}_{\mathrm{San}}, m; r_3)$
  - $\tau := (\rho_1, \rho_2, q_1, q_2, q_3, c)$
- Re-Tagging:
  - Randomness ρ_1'
  - $\mu' \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m'; \rho_1')$
  - $\rho_2' \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}_{\mathrm{San}}, \mu', y)$
  - $\tau := (\rho_1', \rho_2', q_1, q_2, q_3, c)$

Construction

- Tagging:
  - Randomness q_1, q_2, q_3
  - $r_1 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_1)$, $r_2 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_2)$, $r_3 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_3)$
  - $\rho_1 \leftarrow \mathsf{PRG}(r_1)$, $\rho_2 \leftarrow \mathsf{PRG}(r_2)$
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $c \leftarrow \mathsf{Enc}(\mathsf{pk}_{\mathrm{San}}, m; r_3)$
  - $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}_{\mathrm{Sig}}, (\mathsf{pk}_{\mathrm{San}}, y, q_1, q_2, q_3, c))$
  - $\tau := (\rho_1, \rho_2, q_1, q_2, q_3, c, \sigma)$
- Re-Tagging:
  - Randomness ρ_1'
  - $\mu' \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m'; \rho_1')$
  - $\rho_2' \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}_{\mathrm{San}}, \mu', y)$
  - $\tau := (\rho_1', \rho_2', q_1, q_2, q_3, c, \sigma)$

Construction

- Tagging:
  - Randomness q_1, q_2, q_3
  - $r_1 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_1)$, $r_2 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_2)$, $r_3 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_3)$
  - $\rho_1 \leftarrow \mathsf{PRG}(r_1)$, $\rho_2 \leftarrow \mathsf{PRG}(r_2)$
  - $\mu \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m; \rho_1)$
  - $y \leftarrow \mathsf{TDF}(\mathsf{pk}_{\mathrm{San}}, \mu, \rho_2)$
  - $c \leftarrow \mathsf{Enc}(\mathsf{pk}_{\mathrm{San}}, m; r_3)$
  - $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}_{\mathrm{Sig}}, (\mathsf{pk}_{\mathrm{San}}, y, q_1, q_2, q_3, c))$
  - $\tau := (\rho_1, \rho_2, q_1, q_2, q_3, c, \sigma)$
  - Take-home: (m, ρ_1, ρ_2) can be recovered from (q_1, q_2, q_3, c)
- Re-Tagging:
  - Randomness ρ_1'
  - $\mu' \leftarrow \mathsf{CH}(\mathsf{pk}_{\mathrm{San}}, m'; \rho_1')$
  - $\rho_2' \leftarrow \mathsf{TDF}^{-1}(\mathsf{sk}_{\mathrm{San}}, \mu', y)$
  - $\tau := (\rho_1', \rho_2', q_1, q_2, q_3, c, \sigma)$

Construction

- Proving:
  - $\tau := (\rho_1, \rho_2, q_1, q_2, q_3, c, \sigma)$
  - $r_1 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_1)$, $r_2 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_2)$, $r_3 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_3)$
  - Proof $\pi := (r_1, r_2, r_3)$

Construction

- Proving:
  - $\tau := (\rho_1, \rho_2, q_1, q_2, q_3, c, \sigma)$
  - $r_1 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_1)$, $r_2 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_2)$, $r_3 \leftarrow \mathsf{PRF}(\mathsf{sk}_{\mathrm{Sig}}, q_3)$
  - Proof $\pi := (r_1, r_2, r_3)$
- Judging:
  - Given (m, ρ_1, ρ_2) from tag
  - Compute (m', ρ_1', ρ_2') from $\pi := (r_1, r_2, r_3)$
  - If (m, ρ_1, ρ_2) and (m', ρ_1', ρ_2') evaluate to the same y
    - Sanitizer (Collision only possible when given trapdoor)
  - Otherwise
    - Signer

Table of Contents

Preliminary: Group Signatures

- Setup: Generates $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{osk})$
- Join/Issue: User join the group and obtain usk
- Sign: User signs anonymously
  (signature does not reveal signer)
- Verify by gpk
- Open and Judge: Opener reveals signer of a signature

Recall Construction from Group Signatures [BFLS10]

- Signing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX}
  - Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
  - Group Signature σ_{FULL} for m (w/ signer's key)

Recall Construction from Group Signatures [BFLS10]

- Signing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX}
  - Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
  - Group Signature σ_{FULL} for m (w/ signer's key)
- Sanitizing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX} (unchanged)
  - Group Signature σ'_{FULL} for $m' = \mathrm{MOD}(m)$ (w/ sanitizer's key)

Recall Construction from Group Signatures [BFLS10]

- Signing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX}
  - Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
  - Group Signature σ_{FULL} for m (w/ signer's key)
- Sanitizing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX} (unchanged)
  - Group Signature σ'_{FULL} for $m' = \mathrm{MOD}(m)$ (w/ sanitizer's key)
- Proving: Opening of Group Signatures

Recall Construction from Group Signatures [BFLS10]

- Signing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX}
  - Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
  - Group Signature σ_{FULL} for m (w/ signer's key)
- Sanitizing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX} (unchanged)
  - Group Signature σ'_{FULL} for $m' = \mathrm{MOD}(m)$ (w/ sanitizer's key)
- Proving: Opening of Group Signatures

Unlinkability

From anonymity of group signatures

Recall Construction from Group Signatures [BFLS10]

Signing:

- Regular Deterministic Signature σ_{FIX} for m_{FIX}
- Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
- Group Signature σ_{FULL} for m (w/ signer's key)

Recall Construction from Group Signatures [BFLS10]

Signing:

- Regular Deterministic Signature σ_{FIX} for m_{FIX}
- Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
- Group Signature σ_{FULL} for m (w/ signer's key)

Problem:

- Group creation after user key generation!
- Not satisfied by most schemes

Recall Construction from Group Signatures [BFLS10]

Signing:

- Regular Deterministic Signature σ_{FIX} for m_{FIX}
- Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
- Group Signature σ_{FULL} for m (w/ signer's key)

Problem:

- Group creation after user key generation!
- Not satisfied by most schemes

Our Solution: Accountable Ring Signatures

Accountable Ring Signatures [XY04]

Ring Signatures:
Signature is anonymous within ring of users

- User Key Generation
- Ring-Sign
- Verify

Accountable Ring Signatures [XY04]

Accountable Ring Signatures:
Signature is anonymous within ring of users
Except when opener reveals the signer

- Opener Key Generation
- User Key Generation
- Ring-Sign
- Verify
- Open
- Judge

History of Accountable Ring Signatures

- [XY04]: Introduced, Informal

History of Accountable Ring Signatures

- [XY04]: Introduced, Informal
- [BCC+15]: Formalized
  - Generic construction
  - DDH-based instantiation:
    - Logarithmic signature size (w.r.t. ring size)
    - ROM

History of Accountable Ring Signatures

- [XY04]: Introduced, Informal
- [BCC+15]: Formalized
  - Generic construction
  - DDH-based instantiation:
    - Logarithmic signature size (w.r.t. ring size)
    - ROM
- [BCC+16]: Implies Fully Dynamic Group Signatures

History of Accountable Ring Signatures

- [XY04]: Introduced, Informal
- [BCC+15]: Formalized
  - Generic construction
  - DDH-based instantiation:
    - Logarithmic signature size (w.r.t. ring size)
    - ROM
- [BCC+16]: Implies Fully Dynamic Group Signatures
- Ours: Constant-size accountable ring signatures in CRS
  - Constant-size ring signatures [BDR15] (pairing-based, GS proof) +
  - Structure preserving encryption [CHK+11]

Unlinkability from Accountable Ring Signatures

Recall: Construction from Group Signatures [BFLS10]

- Signing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX}
  - Create a group $G := \{\mathrm{Sig}, \mathrm{San}\}$
  - Group Signature σ_{FULL} for m (w/ signer's key)
- Sanitizing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX} (unchanged)
  - Group Signature σ'_{FULL} for $m' = \mathrm{MOD}(m)$ (w/ sanitizer's key)
- Proving: Opening of Group Signatures

Unlinkability from Accountable Ring Signatures

Construction from Accountable Ring Signatures

- Signing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX}
  - Create a ring $R := \{\mathrm{Sig}, \mathrm{San}\}$
  - Ring Signature σ_{FULL} for m (w/ signer's key)
- Sanitizing:
  - Regular Deterministic Signature σ_{FIX} for m_{FIX} (unchanged)
  - Ring Signature σ'_{FULL} for $m' = \mathrm{MOD}(m)$ (w/ sanitizer's key)
- Proving: Opening of Ring Signatures

Table of Contents

Concluding Remark

- Basic SanSig in Standard Model
  - From Re-randomizable Tagging
    - From Double Trapdoor Chameleon Hash + "Special" Tag-based Trapdoor Functions
- Unlinkable SanSig in CRS Model
  - From Accountable Ring Signatures
    - From Constant-size Ring Signatures + Structure Preserving Encryption
- Better understanding of what constitutes SanSig
- New notions/primitives may find independent interest

Concluding Remark

- We apologize for the mistakes in proceeding version
- Corrected full version and slides available online
  - `personal.ie.cuhk.edu.hk/~wflai`
- Contact: `russell@ie.cuhk.edu.hk`

Table of Contents

References I

Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik.

Sanitizable signatures.

In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *ESORICS 2005*, volume 3679 of *LNCS*, pages 159–177. Springer, Heidelberg, September 2005.

Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit.

Short accountable ring signatures based on DDH.

In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *ESORICS 2015, Part I*, volume 9326 of *LNCS*, pages 243–265. Springer, Heidelberg, September 2015.

Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, and Jens Groth.

Foundations of fully dynamic group signatures.

In *ACNS 2016*, pages 117–136, 2016.

References II

Priyanka Bose, Dipanjan Das, and Chandrasekaran Pandu Rangan.

Constant size ring signature without random oracle.

In Ernest Foo and Douglas Stebila, editors, *ACISP 15*, volume 9144 of *LNCS*, pages 230–247. Springer, Heidelberg, June / July 2015.

Christina Brzuska, Marc Fischlin, Tobias Freudenreich, Anja Lehmann, Marcus Page, Jakob Schelbert, Dominique Schröder, and Florian Volk.

Security of sanitizable signatures revisited.

In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009*, volume 5443 of *LNCS*, pages 317–336. Springer, Heidelberg, March 2009.

Christina Brzuska, Marc Fischlin, Anja Lehmann, and Dominique Schröder.

Unlinkability of sanitizable signatures.

In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 444–461. Springer, Heidelberg, May 2010.

References III

Jan Camenisch, Kristiyan Haralambiev, Markulf Kohlweiss, Jorn Lapon, and Vincent Naessens.

Structure preserving CCA secure encryption and applications.

In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 89–106. Springer, Heidelberg, December 2011.

Nils Fleischhacker, Johannes Krupp, Giulio Malavolta, Jonas Schneider, Dominique Schröder, and Mark Simkin.

Efficient unlinkable sanitizable signatures from signatures with re-randomizable keys.

In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 301–330. Springer, Heidelberg, March 2016.

References IV

Jun Furukawa and Shoko Yonezawa.

Group signatures with separate and distributed authorities.

In Carlo Blundo and Stelvio Cimato, editors, *SCN 04*, volume 3352 of *LNCS*, pages 77–90. Springer, Heidelberg, September 2005.

Jens Groth.

Fully anonymous group signatures without random oracles.

In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, Heidelberg, December 2007.

Shouhuai Xu and Moti Yung.

Accountable ring signatures: A smart card approach.

In Jean-Jacques Quisquater, Pierre Paradinas, Yves Deswarte, and Anas Abou El Kalam, editors, *CARDIS*, volume 153 of *IFIP*, pages 271–286. Kluwer/Springer, 2004.

Table of Contents

Application I - Medical Record

- **Doctor** signs medical record to **Client**
- Medical record =
  (Name, Medical Advice, Address, Disease, ...)
- **Client** wants to apply sick leave
- Traditional approach:
  - **Client** sends signed medical record to **Company**
  - **Company** learns extra personal information!
    e.g., address, disease, ...
- Sanitizable signatures:
  - Sanitized medical record = (Name, Medical Advice, xxx, ...)
  - **Client** sends sanitized medical record to **Company**
  - **Company** learns **Doctor**'s medial advice but nothing else

Application II - Outsourced Database

- **Data Owner** signs data to **Server**
- **Client** queries **Server** for specific entries
- Naive approach:
  - **Server** sends all signed data to **Client**
  - **Client** learns extra data (which it may not be entitled to)
- Sanitizable signatures:
  - **Server** sends specific entries to **Client**
  - **Client** is convinced specific entries are from **Data Owner**

CR under Selectve-Tag Adaptive-Image Attack

- $\mathcal{A}$ outputs Q tags $\mu_1, \ldots, \mu_Q$
- $\mathcal{C}$ generates $(\mathsf{pk}, \mathsf{sk})$, gives pk to $\mathcal{A}$
- $\mathcal{A}$ adaptively chooses images y_i
- $\mathcal{C}$ responds with $\mathsf{TDF}^{-1}(\mathsf{sk}, \mu_i, y_i)$
- $\mathcal{A}$ outputs $(\mu_1^*, \rho_1^*), (\mu_2^*, \rho_2^*)$ such that
  - $\mathsf{TDF}(\mathsf{pk}, \mu_1^*, \rho_1^*) = \mathsf{TDF}(\mathsf{pk}, \mu_2^*, \rho_2^*)$
  - $(\mu_1^*, \rho_1^*) \neq (\mu_2^*, \rho_2^*)$
  - $\mu_1^*, \mu_2^* \notin (\mu_1, \ldots, \mu_Q)$