Chair of Applied Cryptography

ChaAC

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

FACULTY OF ENGINEERING

Multi-Key Homomorphic Signatures Unforgeable under Insider Corruption

Russell W. F. Lai[1,2], Raymond K. H. Tai[2], Harry W. H. Wong[2], Sherman S. M. Chow[2]

[1]Friedrich-Alexander University Erlangen-Nuremberg

[2]Chinese University of Hong Kong

Useful multi-key homomorphic signatures likely require strong assumptions.

Overview

We introduce a strong but natural unforgeability notion of (multi-key) homomorphic signatures.

Overview

We introduce a strong but natural unforgeability notion of (multi-key) homomorphic signatures.

The property is essential for natural applications, e.g., verifiable MPC.

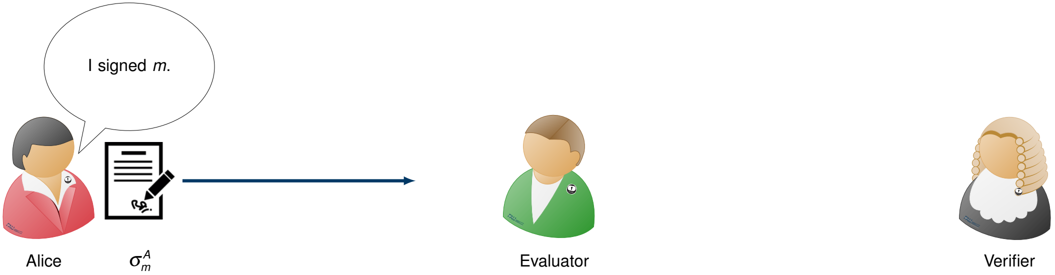
Overview

We introduce a strong but natural unforgeability notion of (multi-key) homomorphic signatures.

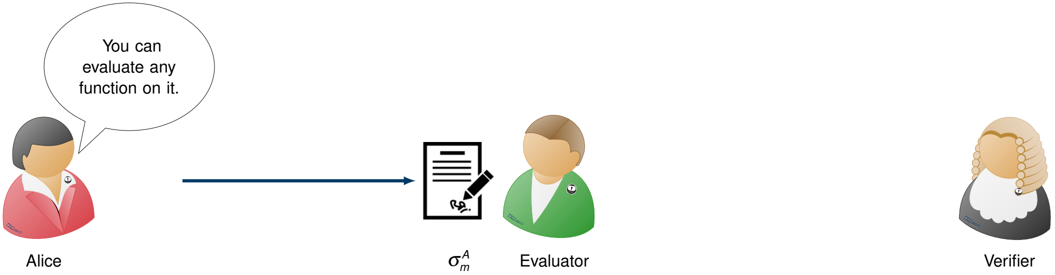
The property is essential for natural applications, e.g., verifiable MPC.

We draw connections of the notion to zk-SNARG/Ks.

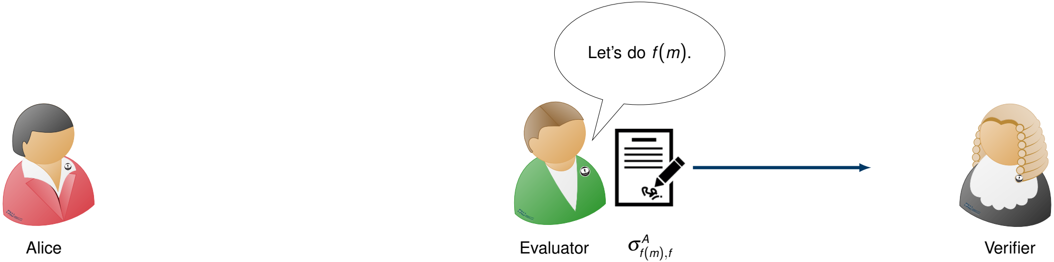
Homomorphic Signatures



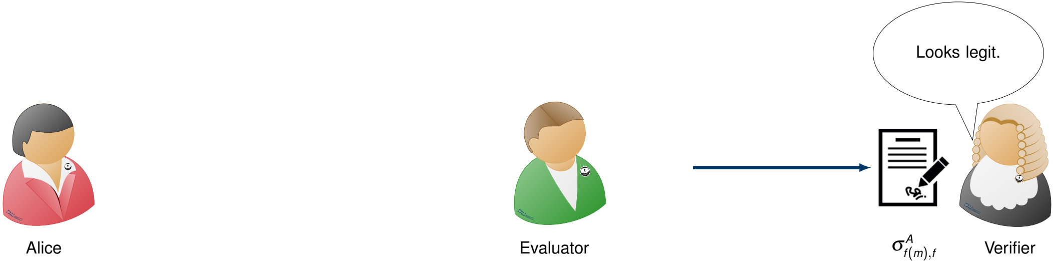
Homomorphic Signatures



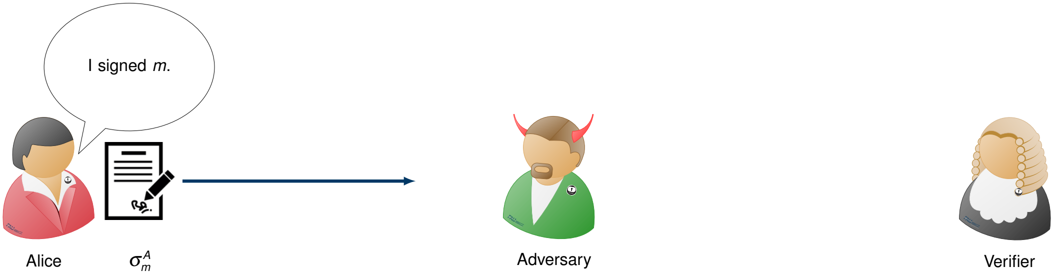
Homomorphic Signatures



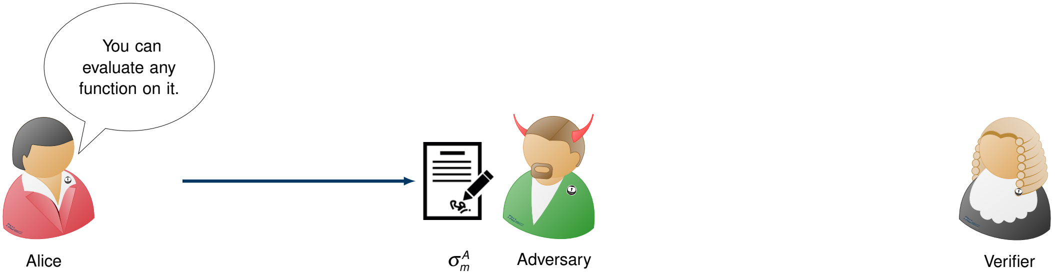
Homomorphic Signatures



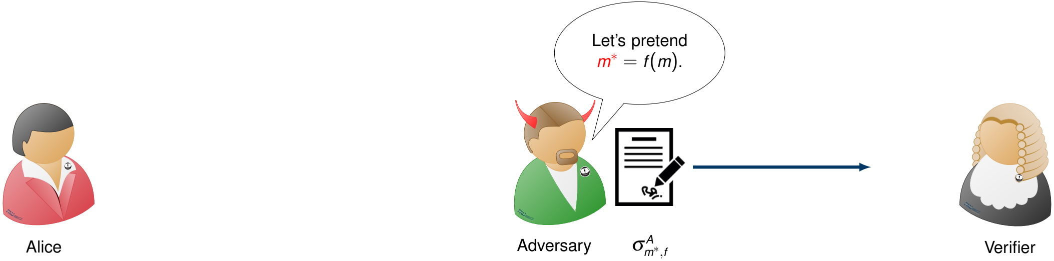
Unforgeability of Homomorphic Signatures



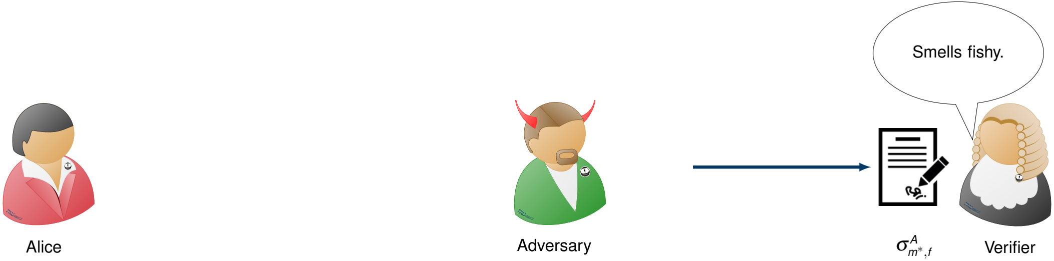
Unforgeability of Homomorphic Signatures



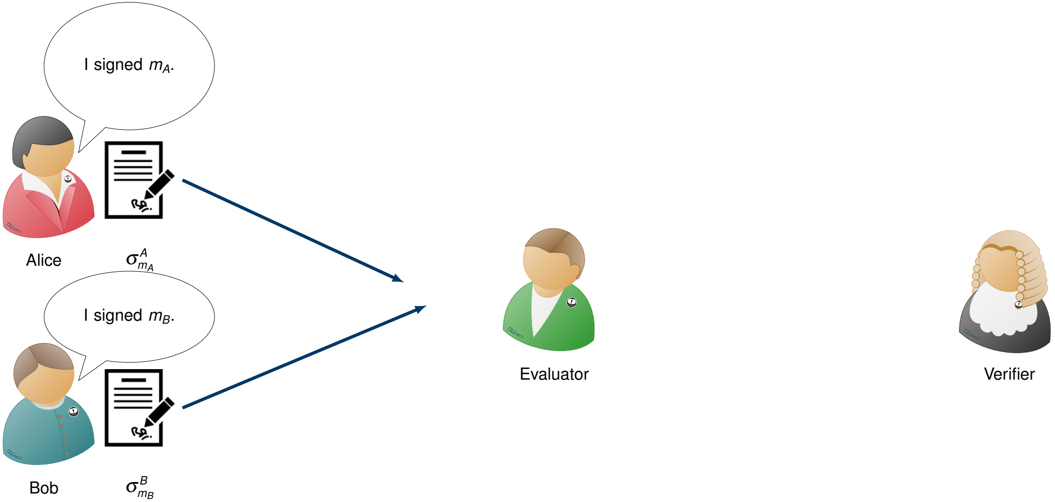
Unforgeability of Homomorphic Signatures



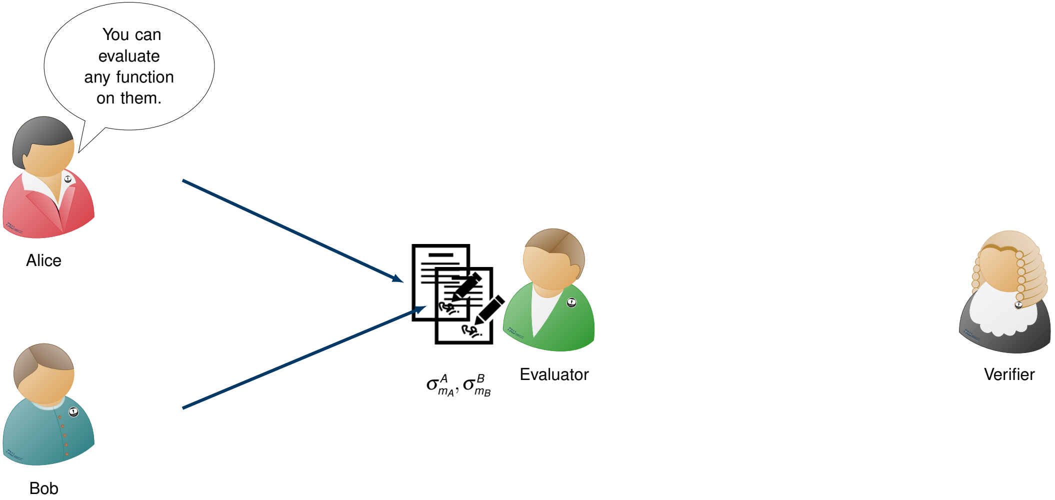
Unforgeability of Homomorphic Signatures



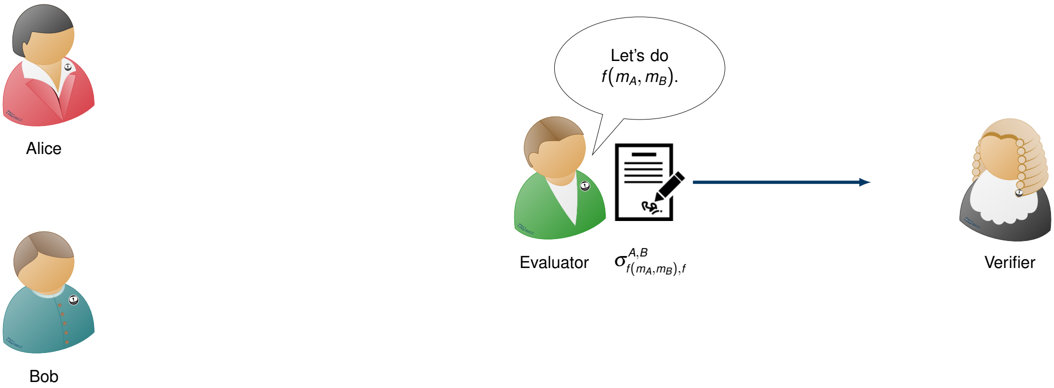
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



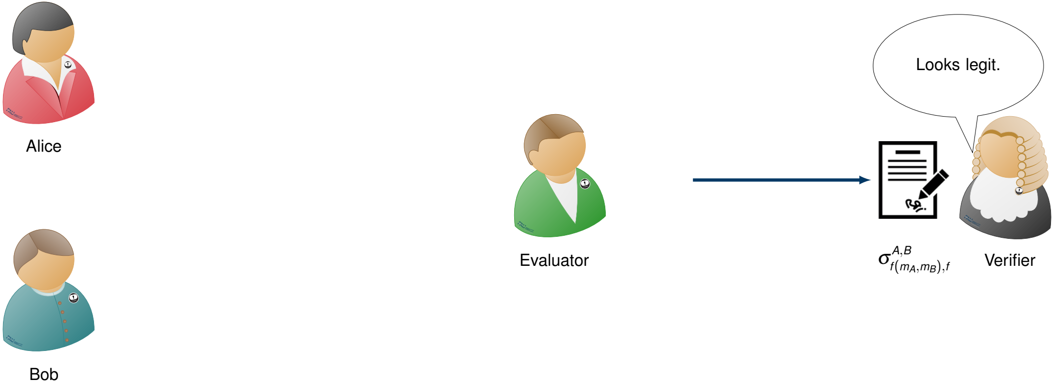
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



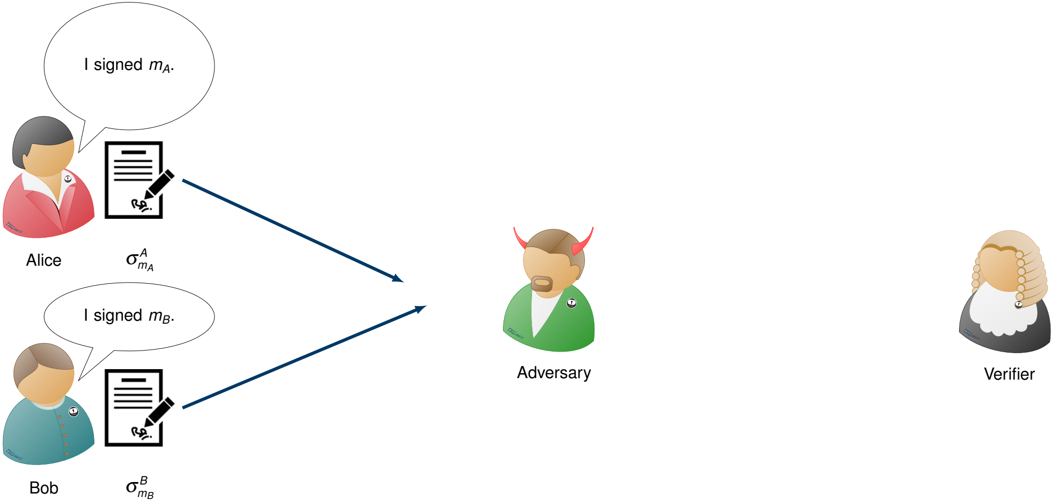
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



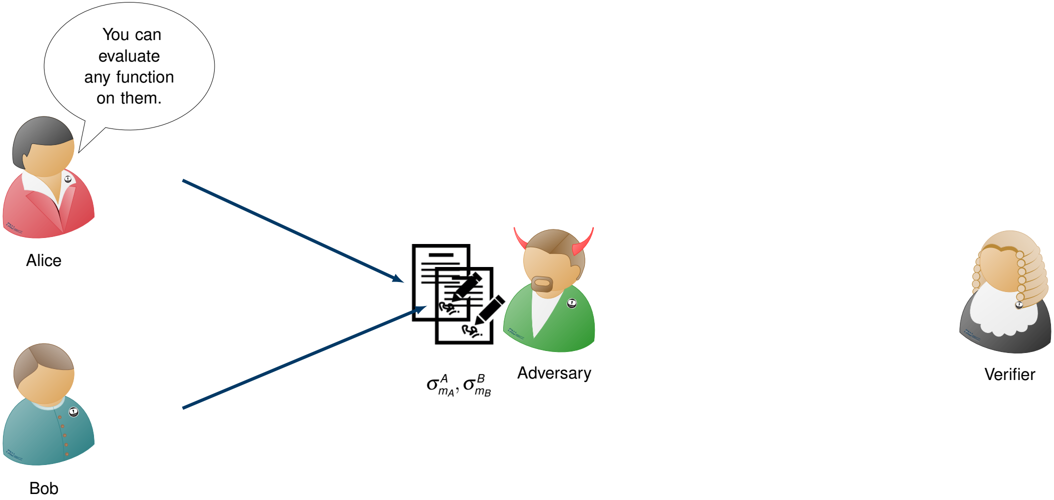
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



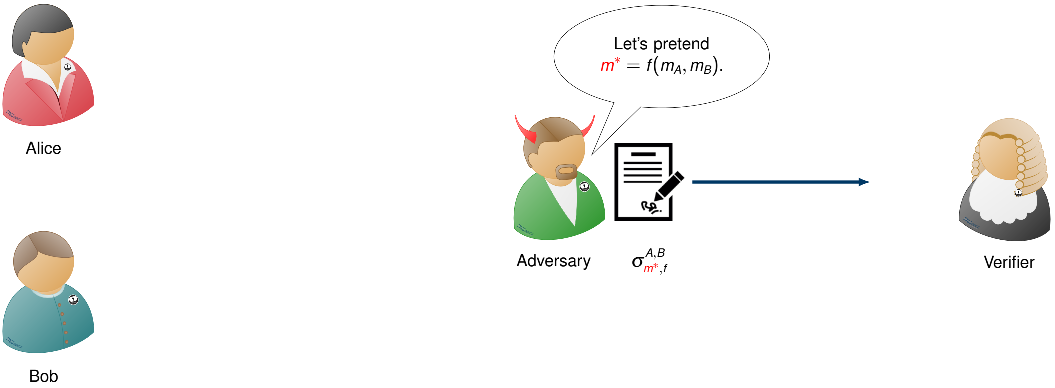
Unforgeability of Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



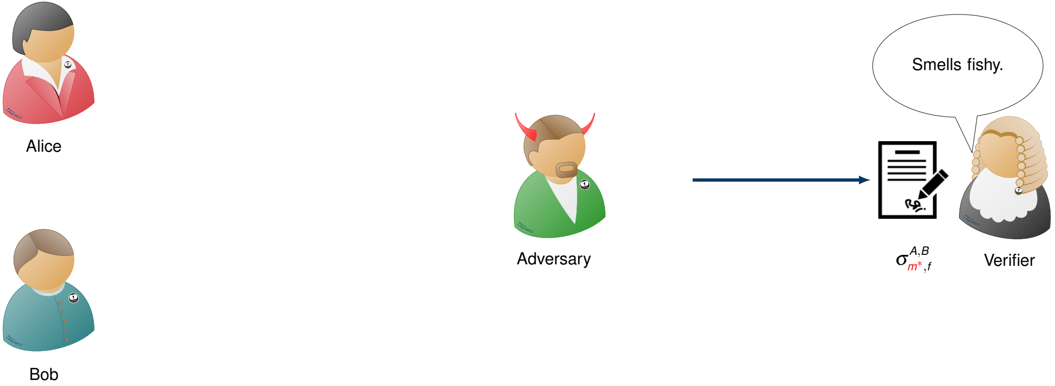
Unforgeability of Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



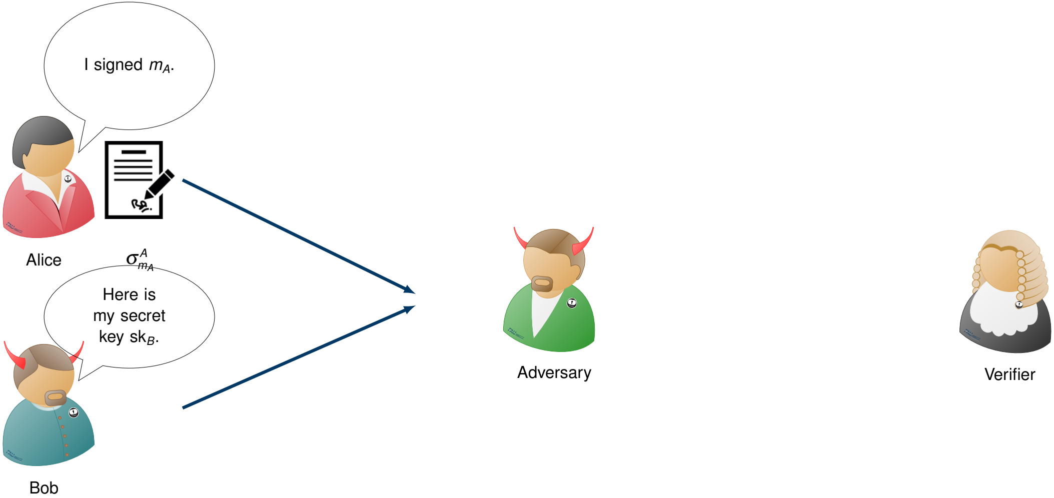
Unforgeability of Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



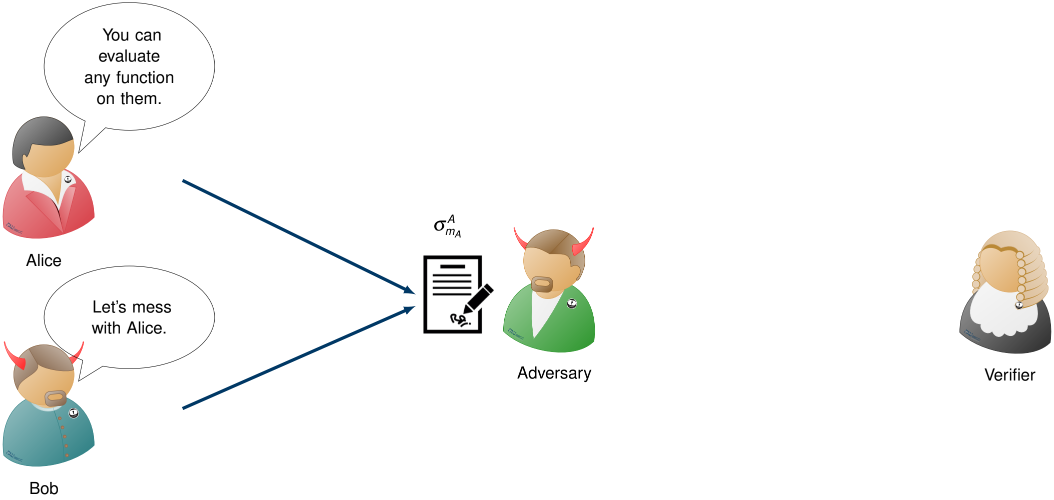
Unforgeability of Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]



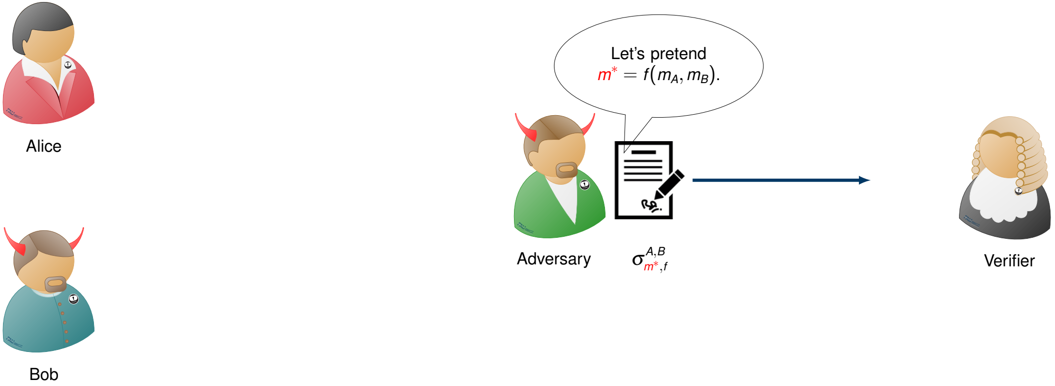
Insider Attack?



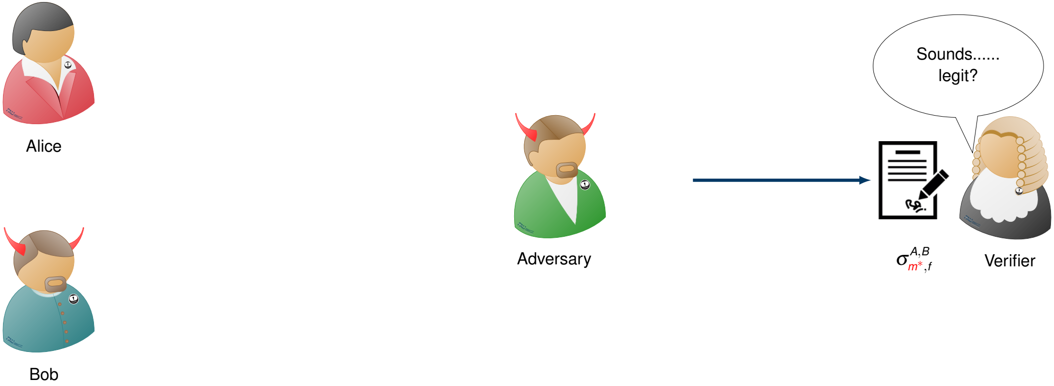
Insider Attack?



Insider Attack?



Insider Attack?



Unforgeability of (Multi-Key) Homomorphic Signatures under Insider Corruption

- $\mathcal{A}$ can query sign oracle on (id, m), which does the following:
  - Generate $(\mathsf{pk}_{\mathrm{id}}, \mathsf{sk}_{\mathrm{id}})$ and record id as honest if not done already.
  - Sign m using $\mathsf{sk}_{\mathrm{id}}$ as σ_m^{id} and record m in the set M_{id}.
  - Return $(\mathsf{pk}_{\mathrm{id}}, \sigma_m^{\mathrm{id}})$.

Unforgeability of (Multi-Key) Homomorphic Signatures under Insider Corruption

- $\mathcal{A}$ can query sign oracle on (id, m), which does the following:
  - Generate $(\mathsf{pk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id}})$ and record id as honest if not done already.
  - Sign m using $\mathsf{sk}_{\mathsf{id}}$ as σ_m^{id} and record m in the set M_{id}.
  - Return $(\mathsf{pk}_{\mathsf{id}}, \sigma_m^{\mathsf{id}})$.
- $\mathcal{A}$ produces $(f^*, \{\mathsf{pk}^*_{\mathsf{id}_1}, \ldots, \mathsf{pk}^*_{\mathsf{id}_k}\}, m^*, \sigma^*)$.

Unforgeability of (Multi-Key) Homomorphic Signatures under Insider Corruption

- $\mathcal{A}$ can query sign oracle on (id, m), which does the following:
  - Generate $(\mathsf{pk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id}})$ and record id as honest if not done already.
  - Sign m using $\mathsf{sk}_{\mathsf{id}}$ as σ_m^{id} and record m in the set M_{id}.
  - Return $(\mathsf{pk}_{\mathsf{id}}, \sigma_m^{\mathsf{id}})$.
- $\mathcal{A}$ produces $(f^*, \{\mathsf{pk}^*_{\mathsf{id}_1}, \ldots, \mathsf{pk}^*_{\mathsf{id}_k}\}, m^*, \sigma^*)$.
- $\mathcal{A}$ wins if the following hold:
  - $\mathsf{Vf}(f^*, \{\mathsf{pk}^*_{\mathsf{id}_1}, \ldots, \mathsf{pk}^*_{\mathsf{id}_k}\}, m^*, \sigma^*) = 1$.
  - If id is honest, then $\mathsf{pk}^*_{\mathsf{id}} = \mathsf{pk}_{\mathsf{id}}$.
  - m^* is not in the range of f^*, when the inputs of honest id are restricted to those recorded in M_{id},

    $$\text{i.e., } m^* \notin \left\{ f^*(m_1, \ldots, m_k) : \begin{cases} m_i \in \mathcal{M} & \mathsf{id}_i \text{ is malicious} \\ m_i \in M_{\mathsf{id}_i} & \mathsf{id}_i \text{ is honest} \end{cases} \right\}$$

Unforgeability of (Multi-Key) Homomorphic Signatures under Insider Corruption

- $\mathcal{A}$ can query sign oracle on (id, m), which does the following:
  - Generate $(\mathsf{pk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id}})$ and record id as honest if not done already.
  - Sign m using $\mathsf{sk}_{\mathsf{id}}$ as σ_m^{id} and record m in the set M_{id}.
  - Return $(\mathsf{pk}_{\mathsf{id}}, \sigma_m^{\mathsf{id}})$.
- $\mathcal{A}$ produces $(f^*, \{\mathsf{pk}^*_{\mathsf{id}_1}, \ldots, \mathsf{pk}^*_{\mathsf{id}_k}\}, m^*, \sigma^*)$.
- $\mathcal{A}$ wins if the following hold:
  - $\mathsf{Vf}(f^*, \{\mathsf{pk}^*_{\mathsf{id}_1}, \ldots, \mathsf{pk}^*_{\mathsf{id}_k}\}, m^*, \sigma^*) = 1$.
  - If id is honest, then $\mathsf{pk}^*_{\mathsf{id}} = \mathsf{pk}_{\mathsf{id}}$.
  - m^* is not in the range of f^*, when the inputs of honest id are restricted to those recorded in M_{id},

    i.e., $m^* \notin \left\{ f^*(m_1, \ldots, m_k) : \begin{cases} m_i \in \mathcal{M} & \mathsf{id}_i \text{ is malicious} \\ m_i \in M_{\mathsf{id}_i} & \mathsf{id}_i \text{ is honest} \end{cases} \right\}$

Remark

- The definition still makes sense even with one key, *i.e.*, $k = 1$.
- It means that even the signer cannot produce $\sigma_{m,f}$ for m not in the range of f.

Why is the notion meaningful?

Example 1: Number of keys $k > 1$

- $f^*(m_1, \ldots, m_k) = \mathsf{MAJORITY}(m_1, \ldots, m_k)$
- id_k malicious
- id_i honest, $\mathsf{M}_{\mathsf{id}_i} = \{\mathrm{NO}\}$, for all $i = 1, \ldots, k-1$
- Infeasible to forge $(\mathsf{MAJORITY}, \{\mathsf{pk}^*_{\mathsf{id}_1}, \ldots, \mathsf{pk}^*_{\mathsf{id}_k}\}, m^* = \mathrm{YES}, \sigma^*)$

Why is the notion meaningful?

Example 1: Number of keys $k > 1$

- $f^*(m_1, \ldots, m_k) = \text{MAJORITY}(m_1, \ldots, m_k)$
- id_k malicious
- id_i honest, $\text{M}_{\text{id}_i} = \{\text{NO}\}$, for all $i = 1, \ldots, k-1$
- Infeasible to forge $(\text{MAJORITY}, \{\text{pk}^*_{\text{id}_1}, \ldots, \text{pk}^*_{\text{id}_k}\}, m^* = \text{YES}, \sigma^*)$

Example 2: Number of keys $k = 1$

- C: Unsatisfiable Boolean circuit
- $f^*(m) = C(m)$
- Infeasible to forge $(C, \text{pk}, m^* = 1, \sigma^*)$

Other Properties of (Multi-key) Homomorphic Signatures

(Weakly) Context-Hiding $\sigma_{f(m),f}$ reveals nothing about m.

Succinctness Size of $\sigma_{f(m),f}$ is independent of the size of m and f.

Preliminary: zk-(O-)SNARG/Ks

Argument systems which allow a prover to prove to the verifier:

There exists a witness w such that the relation $R(x, w) = 1$ holds for the statement x.

zero-knowledge	: Proofs reveal nothing about witnesses.
Oracle	: Sound even if the prover has access to certain (*e.g.*, signing) oracles.
Succinct	: Proof size is independent of witness size.
Non-Interactive	: The prover only sends 1 message to the verifier.
ARGguments	: The system is computationally sound.
ARguments of Knowledge	: There exists an extractor which extracts witnesses from provers.

Roadmap

- zk-(O-)SNARKs + Signatures $\implies$ Insider Unforgeable Multi-key Homomorphic Signatures.

Roadmap

- zk-(O-)SNARKs + Signatures $\Longrightarrow$ Insider Unforgeable Multi-key Homomorphic Signatures.
- 1-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ zk-SNARGs

Roadmap

- zk-(O-)SNARKs + Signatures $\implies$ Insider Unforgeable Multi-key Homomorphic Signatures.
- 1-key 1-hop Insider Unforgeable Homomorphic Signatures $\implies$ zk-SNARGs
- 2-key 1-hop Insider Unforgeable Homomorphic Signatures $\implies$ Functional Signatures
- Functional Signatures $\implies$ zk-SNARGs [Boyle-Goldwasser-Ivan, PKC14]

Roadmap

- zk-(O-)SNARKs + Signatures $\Longrightarrow$ Insider Unforgeable Multi-key Homomorphic Signatures.
- 1-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ zk-SNARGs
- 2-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ Functional Signatures
- Functional Signatures $\Longrightarrow$ zk-SNARGs [Boyle-Goldwasser-Ivan, PKC14]

Roadmap

- zk-(O-)SNARKs + Signatures $\Longrightarrow$ Insider Unforgeable Multi-key Homomorphic Signatures.
- 1-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ zk-SNARGs
- 2-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ Functional Signatures
- Functional Signatures $\Longrightarrow$ zk-SNARGs [Boyle-Goldwasser-Ivan, PKC14]

Theorem (Gentry-Wichs, STOC11)

No SNARGs can be proven adaptive sound via a black-box reduction from any falsifiable assumption.

Roadmap

- zk-(O-)SNARKs + Signatures $\Longrightarrow$ Insider Unforgeable Multi-key Homomorphic Signatures.
- 1-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ zk-SNARGs
- 2-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ Functional Signatures
- Functional Signatures $\Longrightarrow$ zk-SNARGs [Boyle-Goldwasser-Ivan, PKC14]

Theorem (Gentry-Wichs, STOC11)

No SNARGs can be proven adaptive sound via a black-box reduction from any falsifiable assumption.

Corollary

Homomorphic signatures cannot be proven unforgeable under insider corruption via a black-box reduction from any falsifiable assumption.

Roadmap

- zk-(O-)SNARKs + Signatures $\Longrightarrow$ Insider Unforgeable Multi-key Homomorphic Signatures.
- 1-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ zk-SNARGs
- 2-key 1-hop Insider Unforgeable Homomorphic Signatures $\Longrightarrow$ Functional Signatures
- Functional Signatures $\Longrightarrow$ zk-SNARGs [Boyle-Goldwasser-Ivan, PKC14]

Theorem (Gentry-Wichs, STOC11)

No SNARGs can be proven adaptive sound via a black-box reduction from any falsifiable assumption.

Corollary

Homomorphic signatures cannot be proven unforgeable under insider corruption via a black-box reduction from any falsifiable assumption (assuming messages can depend on public parameters).

Construction of Homomorphic Signatures

Ingredients

- zk-(O-)SNARK Π
- Digital signature scheme Σ

Construction of Homomorphic Signatures

- Public Parameters: Common reference string for Π.

Ingredients

- zk-(O-)SNARK Π
- Digital signature scheme Σ

Construction of Homomorphic Signatures

- Public Parameters: Common reference string for Π.
- Key Generation: Each user generates $(\mathsf{pk}, \mathsf{sk})$ for Σ.

Ingredients

- zk-(O-)SNARK Π
- Digital signature scheme Σ

Construction of Homomorphic Signatures

- Public Parameters: Common reference string for Π.
- Key Generation: Each user generates $(\mathsf{pk}, \mathsf{sk})$ for Σ.
- Signing: Sign using $\sigma \leftarrow \Sigma.\mathsf{Sig}(\mathsf{sk}, m)$.

Ingredients

- zk-(O-)SNARK Π
- Digital signature scheme Σ

Construction of Homomorphic Signatures

- Public Parameters: Common reference string for Π.
- Key Generation: Each user generates $(\mathrm{pk}, \mathrm{sk})$ for Σ.
- Signing: Sign using $\sigma \leftarrow \Sigma.\mathrm{Sig}(\mathrm{sk}, m)$.
- Evaluation: Given g, $\{(f_i, \mathrm{pk}_i, m_i, \sigma^i_{m_i,f_i})\}_{i=1}^k$,
  - Let $h = g(f_1, \ldots, f_k)$.
  - Compute $m = g(m_1, \ldots, m_k)$.

Ingredients

- zk-(O-)SNARK Π
- Digital signature scheme Σ

Construction of Homomorphic Signatures

- Public Parameters: Common reference string for Π.
- Key Generation: Each user generates $(\mathsf{pk}, \mathsf{sk})$ for Σ.
- Signing: Sign using $\sigma \leftarrow \Sigma.\mathsf{Sig}(\mathsf{sk}, m)$.
- Evaluation: Given g, $\{(f_i, \mathsf{pk}_i, m_i, \sigma^i_{m_i,f_i})\}_{i=1}^k$,
  - Let $h = g(f_1, \ldots, f_k)$.
  - Compute $m = g(m_1, \ldots, m_k)$.
  - Produce a zk-SNARK proof for the following statement:

$$\text{“}I\ know\ g\ and\ \{(f_i, m_i, \sigma^i_{m_i,f_i})\}_{i=1}^k\ such\ that$$
$$h = g(f_1, \ldots, f_k),$$
$$m = g(m_1, \ldots, m_k),\ and$$
$$for\ i \in [k],\ \sigma^i_{m_i,f_i}\ is\ valid\ under\ \mathsf{pk}_i.\text{”}$$

Ingredients

- zk-(O-)SNARK Π
- Digital signature scheme Σ

Construction of Homomorphic Signatures

- Public Parameters: Common reference string for Π.
- Key Generation: Each user generates $(\mathsf{pk}, \mathsf{sk})$ for Σ.
- Signing: Sign using $\sigma \leftarrow \Sigma.\mathsf{Sig}(\mathsf{sk}, m)$.
- Evaluation: Given g, $\{(f_i, \mathsf{pk}_i, m_i, \sigma^i_{m_i,f_i})\}_{i=1}^k$,
  - Let $h = g(f_1, \ldots, f_k)$.
  - Compute $m = g(m_1, \ldots, m_k)$.
  - Produce a zk-SNARK proof for the following statement:

$$\text{“}\textit{I know } g \textit{ and } \{(f_i, m_i, \sigma^i_{m_i,f_i})\}_{i=1}^k \textit{ such that}$$
$$h = g(f_1, \ldots, f_k),$$
$$m = g(m_1, \ldots, m_k), \textit{ and}$$
$$\textit{for } i \in [k],\ \sigma^i_{m_i,f_i} \textit{ is valid under } \mathsf{pk}_i.\text{”}$$

- Verification: If signature is fresh, use verification of Σ. If signature is evaluated, use verification of Π.

Ingredients

- zk-(O-)SNARK Π
- Digital signature scheme Σ

Caution

- On the number of hops of evaluation:
  - ✗ “Poly-hop” evaluation requires “strong” zk-SNARK extractor whose runtime is independent of that of the prover. As far as we know, no candidate construction exists.

Caution

- On the number of hops of evaluation:
  - ✗ "Poly-hop" evaluation requires "strong" zk-SNARK extractor whose runtime is independent of that of the prover. As far as we know, no candidate construction exists.
  - ✓ 1-hop is sufficient for the construction of zk-SNARG.

Caution

- On the number of hops of evaluation:
  - ✗ "Poly-hop" evaluation requires "strong" zk-SNARK extractor whose runtime is independent of that of the prover. As far as we know, no candidate construction exists.
  - ✓ 1-hop is sufficient for the construction of zk-SNARG.
- On the existence of O-SNARKs:
  - ✗ There exists Σ $s.t.$ no candidate construction of O-SNARK satisfies proof of knowledge with respect to the signing oracle of Σ. [Fiore-Nitulescu, TCC16B]

Caution

- On the number of hops of evaluation:
  - ✗ "Poly-hop" evaluation requires "strong" zk-SNARK extractor whose runtime is independent of that of the prover. As far as we know, no candidate construction exists.
  - ✓ 1-hop is sufficient for the construction of zk-SNARG.
- On the existence of O-SNARKs:
  - ✗ There exists Σ $s.t.$ no candidate construction of O-SNARK satisfies proof of knowledge with respect to the signing oracle of Σ. [Fiore-Nitulescu, TCC16B]
  - ✓ Use a Σ which admits an O-SNARK. [Fiore-Nitulescu, TCC16B]

Construction of zk-SNARG

- Ingredients:
  - 1-key 1-hop homomorphic signature Σ unforgeable under insider corruption
  - A circuit g such that $g(x,w) = \begin{cases} x & R(x,w) = 1 \\ \perp & \text{otherwise} \end{cases}$.

Construction of zk-SNARG

- Ingredients:
  - 1-key 1-hop homomorphic signature Σ unforgeable under insider corruption
  - A circuit g such that $g(x, w) = \begin{cases} x & R(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$.
- Common Reference String: Public Parameter of Σ.

Construction of zk-SNARG

- Ingredients:
  - 1-key 1-hop homomorphic signature Σ unforgeable under insider corruption
  - A circuit g such that $g(x, w) = \begin{cases} x & R(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$.
- Common Reference String: Public Parameter of Σ.
- Proving that $R(x, w) = 1$:
  - Generate fresh $(\mathsf{pk}, \mathsf{sk})$ for Σ.
  - Sign x and w using sk.
  - Evaluate g on the signatures and produce $\sigma_{x,g}$.
  - Output $(\mathsf{pk}, \sigma_{x,g})$.

Construction of zk-SNARG

- Ingredients:
  - 1-key 1-hop homomorphic signature Σ unforgeable under insider corruption
  - A circuit g such that $g(x, w) = \begin{cases} x & R(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$.
- Common Reference String: Public Parameter of Σ.
- Proving that $R(x, w) = 1$:
  - Generate fresh $(\mathsf{pk}, \mathsf{sk})$ for Σ.
  - Sign x and w using sk.
  - Evaluate g on the signatures and produce $\sigma_{x,g}$.
  - Output $(\mathsf{pk}, \sigma_{x,g})$.
- Verification of statement x and proof $\pi = (\mathsf{pk}, \sigma)$:
  - Output $\Sigma.\mathsf{Vf}(g, \mathsf{pk}, x, \sigma)$.

Construction of zk-SNARG

- Ingredients:
  - 1-key 1-hop homomorphic signature Σ unforgeable under insider corruption
  - A circuit g such that $g(x, w) = \begin{cases} x & R(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$.
- Common Reference String: Public Parameter of Σ.
- Proving that $R(x, w) = 1$:
  - Generate fresh $(\mathsf{pk}, \mathsf{sk})$ for Σ.
  - Sign x and w using sk.
  - Evaluate g on the signatures and produce $\sigma_{x,g}$.
  - Output $(\mathsf{pk}, \sigma_{x,g})$.
- Verification of statement x and proof $\pi = (\mathsf{pk}, \sigma)$:
  - Output $\Sigma.\mathsf{Vf}(g, \mathsf{pk}, x, \sigma)$.

Soundness

If x^* is a NO instance, then $g(x^*, w) = \perp$ for all w.

Conclusion

(Multi-key) homomorphic signatures unforgeable under insider corruption imply zk-SNARGs, which likely require non-falsifiable assumptions.

Conclusion

(Multi-key) homomorphic signatures unforgeable under insider corruption imply zk-SNARGs, which likely require non-falsifiable assumptions.

Can we construct insider unforgeable homomorphic signatures ...

directly without using zk-SNARKs?

for restricted functionalities (not including g) from standard assumptions?

Conclusion

(Multi-key) homomorphic signatures unforgeable under insider corruption imply zk-SNARGs, which likely require non-falsifiable assumptions.

Can we construct insider unforgeable homomorphic signatures ...

directly without using zk-SNARKs?

for restricted functionalities (not including g) from standard assumptions?

`ia.cr/2016/834`

Russell W. F. Lai
Friedrich-Alexander University Erlangen-Nuremberg
`russell.lai@cs.fau.de`