Multi-Key Homomorphic Signatures
Unforgeable under Insider Corruption

Russell W. F. Lai\textsuperscript{1,2}, Raymond K. H. Tai\textsuperscript{2}, Harry W. H. Wong\textsuperscript{2}, Sherman S. M. Chow\textsuperscript{2}

\textsuperscript{1}Friedrich-Alexander University Erlangen-Nuremberg
\textsuperscript{2}Chinese University of Hong Kong
Useful multi-key homomorphic signatures likely require strong assumptions.
We introduce a strong but natural unforgeability notion of (multi-key) homomorphic signatures.
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 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

I signed $m$.  

Alice $\sigma^A_m$ 

Verifier
Homomorphic Signatures

You can evaluate any function on it.
Homomorphic Signatures

Let’s do $f(m)$.

Alice
Evaluator
Verifier

$\sigma^A_{f(m), f}$
Homomorphic Signatures

\[ \sigma^A_{f(m), t} \]

Looks legit.
Unforgeability of Homomorphic Signatures

Alice

\[ \sigma_m^A \]

Adversary

Verifier

I signed \( m \).
Unforgeability of Homomorphic Signatures

You can evaluate any function on it.

Alice → Adversary $\sigma^A_m$ → Verifier
Let’s pretend \( m^* = f(m) \).
Unforgeability of Homomorphic Signatures

Alice

Adversary

Verifier

Smells fishy.

Multi-Key Homomorphic Signatures Unforgeable under Insider Corruption

Russell W. F. Lai
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]

I signed $m_A$.

I signed $m_B$.

$\sigma_{mA}$

$\sigma_{mB}$

Verifier

Evaluator

Alice

Bob
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]

You can evaluate any function on them.

Alice

Bob

Evaluator

Verifier
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]

Let's do $f(m_A, m_B)$.

Alice

Bob

Evaluator

Verifier

$\sigma^A_B f(m_A, m_B), f$
Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]

Looks legit.

Verifier

\[ \sigma_{f(m_A, m_B), f} \]

Evaluator

Alice

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

I signed $m_A$.

I signed $m_B$.

Alice

$\sigma^A_{mA}$

Bob

$\sigma^B_{mB}$

Adversary

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

You can evaluate any function on them.
Unforgeability of Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]

Let's pretend $m^* = f(m_A, m_B)$.

Alice

Bob

Adversary

Verifier

$\sigma_{m^*, A, B}$
Unforgeability of Multi-key Homomorphic Signatures [FMNP, Asiacrypt16]

Smells fishy.

Alice

Bob

Adversary

Verifier

\[ \sigma_{m^A,t}^{A,B} \]
Insider Attack?

I signed $m_A$.

Here is my secret key $sk_B$.

Alice

$\sigma_{mA}$

$\sigma_{mA}^A$

Adversary

Verifier

Bob
Insider Attack?

You can evaluate any function on them.

Let's mess with Alice.

Alice

Bob

Verifer

Adversary

\[ \sigma_{mA} \]

Multi-Key Homomorphic Signatures Unforgeable under Insider Corruption

Russell W. F. Lai

7/16
Let's pretend $m^* = f(m_A, m_B)$.  

Insider Attack?

Alice

Bob

Adversary

Verifier

Let's pretend $m^* = f(m_A, m_B)$. 

Multi-Key Homomorphic Signatures Unforgeable under Insider Corruption

Russell W. F. Lai
Insider Attack?

Alice

Bob

Adversary

Verifier

 Sounds...... legit?

\[ \sigma_{m', f}^{A, B} \]
Unforgeability of (Multi-Key) Homomorphic Signatures under Insider Corruption

- \( \mathcal{A} \) can query sign oracle on \((\text{id}, m)\), which does the following:
  - Generate \((\text{pk}_\text{id}, \text{sk}_\text{id})\) and record id as honest if not done already.
  - Sign \(m\) using \(\text{sk}_\text{id}\) as \(\sigma^\text{id}_m\) and record \(m\) in the set \(M_\text{id}\).
  - Return \((\text{pk}_\text{id}, \sigma^\text{id}_m)\).
Unforgeability of (Multi-Key) Homomorphic Signatures under Insider Corruption

• $\mathcal{A}$ can query sign oracle on $(id, m)$, which does the following:
  • Generate $(pk_{id}, sk_{id})$ and record id as honest if not done already.
  • Sign $m$ using $sk_{id}$ as $\sigma_{id}^m$ and record $m$ in the set $M_{id}$.
  • Return $(pk_{id}, \sigma_{id}^m)$.
• $\mathcal{A}$ produces $(f^*, \{pk_{id_1}^*, \ldots, pk_{id_k}^*\}, m^*, \sigma^*)$. 
Unforgeability of (Multi-Key) Homomorphic Signatures under Insider Corruption

- $\mathcal{A}$ can query sign oracle on $(\text{id}, m)$, which does the following:
  - Generate $(pk_{\text{id}}, sk_{\text{id}})$ and record id as honest if not done already.
  - Sign $m$ using $sk_{\text{id}}$ as $\sigma_m^{\text{id}}$ and record $m$ in the set $M_{\text{id}}$.
  - Return $(pk_{\text{id}}, \sigma_m^{\text{id}})$.
- $\mathcal{A}$ produces $(f^*, \{pk_{\text{id}_1}^*, \ldots, pk_{\text{id}_k}^*\}, m^*, \sigma^*)$.
- $\mathcal{A}$ wins if the following hold:
  - $V_f(f^*, \{pk_{\text{id}_1}^*, \ldots, pk_{\text{id}_k}^*\}, m^*, \sigma^*) = 1$.
  - If id is honest, then $pk_{\text{id}}^* = pk_{\text{id}}$.
  - $m^*$ is not in the range of $f^*$, when the inputs of honest id are restricted to those recorded in $M_{\text{id}}$,
    i.e., $m^* \notin \left\{ f^*(m_1, \ldots, m_k) : \begin{cases} m_i \in \mathcal{M} & \text{id}_i \text{ is malicious} \\ m_i \in M_{\text{id}_i} & \text{id}_i \text{ is honest} \end{cases} \right\}$. 

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 $f$. 

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

\( \mathcal{A} \) can query sign oracle on \((id, m)\), which does the following:

- Generate \((pk_{id}, sk_{id})\) and record id as honest if not done already.
- Sign \(m\) using \(sk_{id}\) as \(\sigma_{id}^m\) and record \(m\) in the set \(M_{id}\).
- Return \((pk_{id}, \sigma_{id}^m)\).

\( \mathcal{A} \) produces \((f^*, \{pk_{id_1}^*, \ldots, pk_{id_k}^*\}, m^*, \sigma^*)\).

\( \mathcal{A} \) wins if the following hold:

- \(V(f^*, \{pk_{id_1}^*, \ldots, pk_{id_k}^*\}, m^*, \sigma^*) = 1\).
- If id is honest, then \(pk_{id}^* = pk_{id}\).
- \(m^*\) is not in the range of \(f^*\), when the inputs of honest id are restricted to those recorded in \(M_{id}\), i.e., \(m^* \notin \{f^*(m_1, \ldots, m_k) : \begin{cases} m_i \in \mathcal{M} & \text{id}_i \text{ is malicious} \\ m_i \in M_{id_i} & \text{id}_i \text{ is honest} \end{cases} \}\)

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 $k > 1$

- $f^*(m_1, \ldots, m_k) = \text{MAJORITY}(m_1, \ldots, m_k)$
- $id_k$ malicious
- $id_i$ honest, $M_{id_i} = \{\text{NO}\}$, for all $i = 1, \ldots, k - 1$
- Infeasible to forge $(\text{MAJORITY}, \{pk_{id_1}^*, \ldots, pk_{id_k}^*\}, m^* = \text{YES}, \sigma^*)$
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, $M_{id_i} = \{\text{NO}\}$, for all $i = 1, \ldots, k - 1$
- Infeasible to forge $(\text{MAJORITY}, \{pk_{id_1}^*, \ldots, pk_{id_k}^*\}, m^* = \text{YES}, \sigma^*)$

Example 2: Number of keys $k = 1$

- $C$: Unsatisfiable Boolean circuit
- $f^*(m) = C(m)$
- Infeasible to forge $(C, pk, m^* = 1, \sigma^*)$
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.
- **ARGuments**: 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 $\implies$ Insider Unforgeable Multi-key Homomorphic Signatures.
• 1-key 1-hop Insider Unforgeable Homomorphic Signatures $\implies$ zk-SNARGs

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 $\implies$ 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 $\implies$ 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 $\implies$ 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]

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 $\implies$ 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]

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 $\implies$ 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]

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 $\Pi$
- Digital signature scheme $\Sigma$

Public Parameters: Common reference string for $\Pi$.

Key Generation: Each user generates $(pk, sk)$ for $\Sigma$.

Signing: Sign using $\sigma \leftarrow \Sigma.$Sig$(sk, m)$.

Evaluation: Given $g$, $\{(f_i, pk_i, m_i, \sigma_{im_i}, f_i)\}_{i=1}^k$.

Let $h = g(f_1, ..., f_k)$.

Compute $m = g(m_1, ..., m_k)$.

Produce a zk-SNARK proof for the following statement: “I know $g$ and $\{(f_i, m_i, \sigma_{im_i}, f_i)\}_{i=1}^k$ such that $h = g(f_1, ..., f_k)$, $m = g(m_1, ..., m_k)$, and for $i \in [k]$, $\sigma_{im_i}, f_i$ is valid under $pk_i$.”

Verification: If signature is fresh, use verification of $\Sigma$. If signature is evaluated, use verification of $\Pi$. 
Construction of Homomorphic Signatures

- Public Parameters: Common reference string for $\Pi$.

**Ingredients**

- zk-(O-)SNARK $\Pi$
- Digital signature scheme $\Sigma$

• 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:
  "I know $g$ and $\{(f_i, m_i, \sigma_i m_i, f_i)\}_{k_i=1}$ 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 $pk_i$."
Construction of Homomorphic Signatures

- Public Parameters: Common reference string for $\Pi$.
- Key Generation: Each user generates $(pk, sk)$ for $\Sigma$.

Ingredients

- zk-(O-)SNARK $\Pi$
- Digital signature scheme $\Sigma$
Construction of Homomorphic Signatures

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

Ingredients

- zk-(O-)SNARK $\Pi$
- Digital signature scheme $\Sigma$
Construction of Homomorphic Signatures

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

Ingredients

- zk-(O-)SNARK $\Pi$
- Digital signature scheme $\Sigma$
Construction of Homomorphic Signatures

- Public Parameters: Common reference string for $\Pi$.
- Key Generation: Each user generates $(pk, sk)$ for $\Sigma$.
- Signing: Sign using $\sigma \leftarrow \Sigma . \text{Sig}(sk, m)$.
- Evaluation: Given $g$, $\{(f_i, pk_i, m_i, \sigma_{m_i, f_i}^i)\}_{i=1}^k$,
  - 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:
    
    "I know $g$ and $\{(f_i, m_i, \sigma_{m_i, f_i}^i)\}_{i=1}^k$ such that
    
    $h = g(f_1, \ldots, f_k)$,
    
    $m = g(m_1, \ldots, m_k)$, and
    
    for $i \in [k]$, $\sigma_{m_i, f_i}^i$ is valid under $pk_i$.”

- Verification: If signature is fresh, use verification of $\Sigma$. If signature is evaluated, use verification of $\Pi$.

Ingredients

- zk-(O-)SNARK $\Pi$
- Digital signature scheme $\Sigma$
Construction of Homomorphic Signatures

- Public Parameters: Common reference string for $\Pi$.
- Key Generation: Each user generates $(pk, sk)$ for $\Sigma$.
- Signing: Sign using $\sigma \leftarrow \Sigma.\text{Sig}(sk, m)$.
- Evaluation: Given $g, \{(f_i, pk_i, m_i, \sigma_{m_i,f_i}^i)\}_{i=1}^k$,
  - 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:
    
    "I know $g$ and $\{(f_i, m_i, \sigma_{m_i,f_i}^i)\}_{i=1}^k$ such that
    
    $h = g(f_1, \ldots, f_k)$,
    
    $m = g(m_1, \ldots, m_k)$, and
    
    for $i \in [k]$, $\sigma_{m_i,f_i}^i$ is valid under $pk_i$."

- Verification: If signature is fresh, use verification of $\Sigma$. If signature is evaluated, use verification of $\Pi$.

Ingredients

- zk-(O-)SNARK $\Pi$
- Digital signature scheme $\Sigma$
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 $\Sigma$ s.t. no candidate construction of O-SNARK satisfies proof of knowledge with respect to the signing oracle of $\Sigma$. [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 $\Sigma$ s.t. no candidate construction of O-SNARK satisfies proof of knowledge with respect to the signing oracle of $\Sigma$. [Fiore-Nitulescu, TCC16B]
  - ✔ Use a $\Sigma$ which admits an O-SNARK. [Fiore-Nitulescu, TCC16B]
Construction of zk-SNARG

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

Soundness

If $x^*$ is a NO instance, then $g(x^*, w) = \bot$ for all $w$. 
Construction of zk-SNARG

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

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

Multi-Key Homomorphic Signatures Unforgeable under Insider Corruption
Russell W. F. Lai
Construction of zk-SNARG

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

• Common Reference String: Public Parameter of $\Sigma$.

• Proving that $R(x, w) = 1$:
  • Generate fresh (pk, sk) for $\Sigma$.
  • Sign $x$ and $w$ using sk.
  • Evaluate $g$ on the signatures and produce $\sigma_{x,g}$.
  • Output (pk, $\sigma_{x,g}$).
Construction of zk-SNARG

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

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

Multi-Key Homomorphic Signatures Unforgeable under Insider Corruption

Russell W. F. Lai
Construction of zk-SNARG

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

Soundness

If $x^*$ is a NO instance, then $g(x^*, w) = \bot$ 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?
(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