Non-Malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures

Benoît Libert
ENS, Lyon

Thomas Peters
UCL, Belgium

Marc Joye
Palo Alto, USA

Moti Yung
New York, USA

Eurocrypt - 14th May 2014

This work makes connections between the following primitives:

Linearly-homomorphic structure-preserving signatures

and

Quasi-Adaptive NIZK proof systems

Structure-preserving "malleable" signatures allow building:

Constant-size unbounded simulation-sound proofs

and

Short CCA-secure encryptions in the multi-challenge setting

This work makes connections between the following primitives:

Linearly-homomorphic structure-preserving signatures

and

Quasi-Adaptive NIZK proof systems

Structure-preserving “malleable” signatures allow building:

Constant-size unbounded simulation-sound proofs

and

Short CCA-secure encryptions in the multi-challenge setting

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize. . .

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

. . . from random tuples (with appropriate length)

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize...

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

...from random tuples (with appropriate length)

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize…

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

…from random tuples (with appropriate length)

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize…

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

…from random tuples (with appropriate length)

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize...

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

...from random tuples (with appropriate length)

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize...

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1 ?
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

...from random tuples (with appropriate length)

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize…

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2 ?
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

…from random tuples (with appropriate length)

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize...

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k} \text{ ?}$$

...from random tuples (with appropriate length)

Linear Subspaces: Where ?

CPA Encryptions in Prime Order Group $\mathbb{G}$:

- El-Gamal ciphertext: $(h_1^{r_1}, m \cdot g^{r_1}) \in \mathbb{G}^2$
- BBS ciphertext: $(h_1^{r_1}, h_2^{r_2}, m \cdot g^{r_1+r_2}) \in \mathbb{G}^3$
- Extension: $(h_1^{r_1}, \ldots, h_k^{r_k}, m \cdot g^{r_1+\cdots+r_k}) \in \mathbb{G}^{k+1}$

Decisional Assumptions in $\mathbb{G}$: Hard to recognize...

- DH-tuple: $(h_1^{r_1}, y) = (h_1, g)^{r_1}$ for some r_1
- Linear tuple: $(h_1^{r_1}, h_2^{r_2}, y) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$ for some r_1, r_2
- k-linear tuple: for some $r_1, \ldots, r_k$,

$$(h_1^{r_1}, \ldots, h_k^{r_k}, y) = (h_1, 1, \ldots, 1, g)^{r_1} \cdots (1, \ldots, 1, h_k, g)^{r_k}$$

... from random tuples (with appropriate length)

Proving Membership of Linear Subspace: Why ?

Cramer-Shoup-like CCA Encryptions:

- DDH-based: $(c_1, c_2, c_3, \pi_1) = (h_1^{r_1}, g^{r_1}, m \cdot X_1^{r_1}, \pi_1)$

  allowing private verification for $(c_1, c_2) = (h_1, g)^{r_1}$

- DLIN-based: $(c_1, c_2, c_3, c_4, \pi_2) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot X_1^{r_1} X_2^{r_2}, \pi_2)$

  π_2 is an "proof" that $(c_1, c_2, c_3) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$

$\Longrightarrow$ Need to prove that a vector is in a linear subspace

Publicly Verifiable CCA Encryptions

- Turn the "designated-verifier" proof into a publicly verifiable one

Proving Membership of Linear Subspace: Why ?

Cramer-Shoup-like CCA Encryptions:

- DDH-based: $(c_1, c_2, c_3, \pi_1) = (h_1^{r_1}, g^{r_1}, m \cdot X_1{}^{r_1}, \pi_1)$

  allowing private verification for $(c_1, c_2) = (h_1, g)^{r_1}$

- DLIN-based: $(c_1, c_2, c_3, c_4, \pi_2) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot X_1{}^{r_1} X_2{}^{r_2}, \pi_2)$

  π_2 is an "proof" that $(c_1, c_2, c_3) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$

$\Longrightarrow$ Need to prove that a vector is in a linear subspace

Publicly Verifiable CCA Encryptions

- Turn the "designated-verifier" proof into a publicly verifiable one

Proving Membership of Linear Subspace: Why ?

Cramer-Shoup-like CCA Encryptions:

- DDH-based: $(c_1, c_2, c_3, \pi_1) = (h_1^{r_1}, g^{r_1}, m \cdot X_1{}^{r_1}, \pi_1)$

  allowing private verification for $(c_1, c_2) = (h_1, g)^{r_1}$

- DLIN-based: $(c_1, c_2, c_3, c_4, \pi_2) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot X_1{}^{r_1} X_2{}^{r_2}, \pi_2)$

  π_2 is an "proof" that $(c_1, c_2, c_3) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$

$\Longrightarrow$ Need to prove that a vector is in a linear subspace

Publicly Verifiable CCA Encryptions

- Turn the "designated-verifier" proof into a publicly verifiable one

Proving Membership of Linear Subspace: Why ?

Cramer-Shoup-like CCA Encryptions:

- DDH-based: $(c_1, c_2, c_3, \pi_1) = (h_1^{r_1}, g^{r_1}, m \cdot X_1{}^{r_1}, \pi_1)$
  allowing private verification for $(c_1, c_2) = (h_1, g)^{r_1}$
- DLIN-based: $(c_1, c_2, c_3, c_4, \pi_2) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot X_1{}^{r_1} X_2{}^{r_2}, \pi_2)$
  π_2 is an "proof" that $(c_1, c_2, c_3) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$

$\Longrightarrow$ Need to prove that a vector is in a linear subspace

Publicly Verifiable CCA Encryptions

- Turn the "designated-verifier" proof into a publicly verifiable one

Proving Membership of Linear Subspace: Why ?

Cramer-Shoup-like CCA Encryptions:

- DDH-based: $(c_1, c_2, c_3, \pi_1) = (h_1^{r_1}, g^{r_1}, m \cdot X_1{}^{r_1}, \pi_1)$

  allowing private verification for $(c_1, c_2) = (h_1, g)^{r_1}$

- DLIN-based: $(c_1, c_2, c_3, c_4, \pi_2) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot X_1{}^{r_1} X_2{}^{r_2}, \pi_2)$

  π_2 is an "proof" that $(c_1, c_2, c_3) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$

$\Longrightarrow$ Need to prove that a vector is in a linear subspace

Publicly Verifiable CCA Encryptions

- Turn the "designated-verifier" proof into a publicly verifiable one

Proving Membership of Linear Subspace: Why ?

Cramer-Shoup-like CCA Encryptions:

- DDH-based: $(c_1, c_2, c_3, \pi_1) = (h_1^{r_1}, g^{r_1}, m \cdot {X_1}^{r_1}, \pi_1)$

  allowing private verification for $(c_1, c_2) = (h_1, g)^{r_1}$

- DLIN-based: $(c_1, c_2, c_3, c_4, \pi_2) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot {X_1}^{r_1}{X_2}^{r_2}, \pi_2)$

  π_2 is an "proof" that $(c_1, c_2, c_3) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$

$\Longrightarrow$ Need to prove that a vector is in a linear subspace

Publicly Verifiable CCA Encryptions

- Turn the "designated-verifier" proof into a publicly verifiable one

Proving Membership of Linear Subspace: Why ?

Cramer-Shoup-like CCA Encryptions:

- DDH-based: $(c_1, c_2, c_3, \pi_1) = (h_1^{r_1}, g^{r_1}, m \cdot X_1{}^{r_1}, \pi_1)$

  allowing private verification for $(c_1, c_2) = (h_1, g)^{r_1}$

- DLIN-based: $(c_1, c_2, c_3, c_4, \pi_2) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot X_1{}^{r_1} X_2{}^{r_2}, \pi_2)$

  π_2 is an "proof" that $(c_1, c_2, c_3) = (h_1, 1, g)^{r_1} \cdot (1, h_2, g)^{r_2}$

$\Longrightarrow$ Need to prove that a vector is in a linear subspace

Publicly Verifiable CCA Encryptions

- Turn the "designated-verifier" proof into a publicly verifiable one

Simulation-sound NIZK proofs

Simulation-Soundness [Sahai, FOCS'99]

Informally, adversary cannot prove false statements, even after having seen simulated proofs for false statements

Motivation for **unbounded** simulation-soundness:

Chosen-ciphertext security in the multi-challenge setting (not implied by single-challenge-CCA in, e.g., KDM security)

Simulation-sound NIZK proofs

Simulation-Soundness [Sahai, FOCS'99]

Informally, adversary cannot prove false statements, even after having seen simulated proofs for false statements

Motivation for **unbounded** simulation-soundness:

Chosen-ciphertext security in the multi-challenge setting (not implied by single-challenge-CCA in, e.g., KDM security)

Proofs of Linear Subspace Membership: Prior Work

Problem: prove that a vector $\vec{v} \in \mathbb{G}^n$ belongs to the row space of

$$\left(G_{ij}\right)_{\substack{1 \le i \le t \\ 1 \le j \le n}} = \begin{pmatrix} G_{11} & \dots & G_{1n} \\ \vdots & \ddots & \vdots \\ G_{t1} & \dots & G_{tn} \end{pmatrix} \in \mathbb{G}^{t \times n} \qquad \text{with} \quad t < n$$

Existing solutions:

- Sigma protocols (via Fiat-Shamir): proofs of length $\Theta(t)$
- Groth-Sahai (Eurocrypt'08): requires $\Theta(n+t)$ elements of $\mathbb{G}$
- Jutla-Roy (Asiacrypt'13): QA-NIZK proofs of $\Theta(n-t)$ elements of $\mathbb{G}$

Our goal: getting competitive with random-oracle-based solutions

Proofs of Linear Subspace Membership: Prior Work

Problem: prove that a vector $\vec{v} \in \mathbb{G}^n$ belongs to the row space of

$$(G_{ij})_{\substack{1 \leq i \leq t \\ 1 \leq j \leq n}} = \begin{pmatrix} G_{11} & \dots & G_{1n} \\ \vdots & \ddots & \vdots \\ G_{t1} & \dots & G_{tn} \end{pmatrix} \in \mathbb{G}^{t \times n} \qquad \text{with} \quad t < n$$

Existing solutions:

- Sigma protocols (via Fiat-Shamir): proofs of length $\Theta(t)$
- Groth-Sahai (Eurocrypt'08): requires $\Theta(n+t)$ elements of $\mathbb{G}$
- Jutla-Roy (Asiacrypt'13): QA-NIZK proofs of $\Theta(n-t)$ elements of $\mathbb{G}$

Our goal: getting competitive with random-oracle-based solutions

...*with Simulation-Soundness: Prior Work*

Problem: prove that vector $\vec{v} \in \mathbb{G}^n$ belongs to the row space of $(G_{ij})_{\substack{1 \leq i \leq t \\ 1 \leq j \leq n}}$

Solutions with one-time simulation-soundness:

- Groth (Asiacrypt'06), Katz-Vaikuntanathan (TCC'11):
  - Use OR proofs (quadratic equations)
  - Proofs of size $\Theta(t + n)$
- Jutla-Roy (PKC'12), Libert-Yung (TCC'12): proofs of size $\Theta(t + n)$

Our goal:

- Still getting competitive with random-oracle-based solutions
- Avoiding the $\Theta(n)$ overhead

. . . with Simulation-Soundness: Prior Work

Unbounded simulation-soundness for linear subspace membership:

- Groth (Asiacrypt'06): based on OR proofs

  "A set S of PPEs is satisfiable"
  "I know a valid signature on a one-time VK"

- Camenisch-Chandran-Shoup (Eurocrypt'09): based on OR proofs

  "A set S of PPEs is satisfiable"
  "a CCA-encrypted value solves a hard problem"

Difficulty: Proving disjunctions of PPEs requires quadratic equations; Cost depends on the number of variables and pairings per equation

Our goal: avoiding OR proofs, CCA-secure encryption, quadratic equations

Our Contributions

Pairing-based Proofs for Linear Subspace Membership

- Secure in the standard model under DLIN (*resp.* k-LIN)
- QA-NIZK arguments in 3 elements (*resp.* $k+1$) of $\mathbb{G}$!
- **Unbounded** simulation-sound proofs in 15 $\mathbb{G}$-elements

Publicly Verifiable Threshold Encryption Schemes

- Fully secure (threshold) keyed-homomorphic encryption
- Adaptively secure non-interactive CCA-secure scheme with ciphertexts in $\mathbb{G}^8$ (*resp.* $\mathbb{G}^{2k+4}$)

Our Contributions

Pairing-based Proofs for Linear Subspace Membership

- Secure in the standard model under DLIN (*resp.* k-LIN)
- QA-NIZK arguments in 3 elements (*resp.* $k+1$) of $\mathbb{G}$!
- **Unbounded** simulation-sound proofs in 15 $\mathbb{G}$-elements

Publicly Verifiable Threshold Encryption Schemes

- Fully secure (threshold) keyed-homomorphic encryption
- Adaptively secure non-interactive CCA-secure scheme with ciphertexts in $\mathbb{G}^8$ (*resp.* $\mathbb{G}^{2k+4}$)

Our Contributions

Pairing-based Proofs for Linear Subspace Membership

- Secure in the standard model under DLIN (*resp.* k-LIN)
- QA-NIZK arguments in 3 elements (*resp.* $k+1$) of $\mathbb{G}$!
- **Unbounded** simulation-sound proofs in 15 $\mathbb{G}$-elements

Publicly Verifiable Threshold Encryption Schemes

- Fully secure (threshold) keyed-homomorphic encryption
- Adaptively secure non-interactive CCA-secure scheme with ciphertexts in $\mathbb{G}^8$ (*resp.* $\mathbb{G}^{2k+4}$)

Structure-Preserving Realization Π_{LH}

Signature on $\vec{v} = (m_1, \ldots, m_n) \in \mathbb{G}^n$ is given by (z, r, u) verifying

$$1_T = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, m_i)$$

$$1_T = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^{n} e(h_i, m_i)$$

Authentication of $\mathbb{F}_p$-linear subspaces of $\mathbb{G}^n$

Given signatures (z_j, r_j, u_j) on $\vec{v}_j = (m_{j,1}, \ldots, m_{j,n})$ allows computing

for any $(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t$

Tool: Linearly Homomorphic Signatures

Structure-Preserving Realization Π_{LH}

Signature on $\vec{v} = (m_1, \ldots, m_n) \in \mathbb{G}^n$ is given by (z, r, u) verifying

$$1_T = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, m_i)$$

$$1_T = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^{n} e(h_i, m_i)$$

Authentication of $\mathbb{F}_p$-linear subspaces of $\mathbb{G}^n$

Given signatures (z_j, r_j, u_j) on $\vec{v}_j = (m_{j,1}, \ldots, m_{j,n})$ allows computing

$$(z', r', u') \longleftarrow \text{SignDerive}(\vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$$

for any $(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t$

Tool: Linearly Homomorphic Signatures

Structure-Preserving Realization Π_{LH}

Signature on $\vec{v} = (m_1, \ldots, m_n) \in \mathbb{G}^n$ is given by (z, r, u) verifying

$$1_T = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, m_i)$$

$$1_T = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^{n} e(h_i, m_i)$$

Authentication of $\mathbb{F}_p$-linear subspaces of $\mathbb{G}^n$

Given signatures (z_j, r_j, u_j) on $\vec{v}_j = (m_{j,1}, \ldots, m_{j,n})$ allows computing

$$(z', r', u') \longleftarrow \mathsf{SignDerive}(\vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$$

for any $(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t$

Our Basic Quasi-Adaptive NIZK Proof (1)

Fixed Common Reference String

- Noted $\Gamma \leftarrow K_0(\lambda)$: pairing description $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$

Language for linear subspace membership

$$\mathcal{L}_{lin}(\Gamma) = \left\{ \vec{v} \in \mathbb{G}^n \mid \exists (\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v_1}^{\alpha_1} \cdots \vec{v_t}^{\alpha_t} \right\}$$

- Then we have $\vec{v} \in \mathcal{L}_{lin}$ if and only if $\vec{v} \in \text{span}\langle \vec{v}_1, \ldots, \vec{v}_t \rangle$

Language-Dependent CRS

- Noted $\phi \leftarrow K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$: generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\Gamma)$ of Π_{LH}, $\sigma_j \leftarrow \mathsf{Sign}_{LH}(\mathsf{sk}, \vec{v}_j)$ for each $j = 1$ to t, and return $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$

Our Basic Quasi-Adaptive NIZK Proof (1)

Fixed Common Reference String

- Noted $\Gamma \leftarrow K_0(\lambda)$: pairing description $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$

Language for linear subspace membership

$$\mathcal{L}_{lin}(\Gamma) = \left\{ \vec{v} \in \mathbb{G}^n \,|\, \exists(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v}_1^{\,\alpha_1} \cdots \vec{v}_t^{\,\alpha_t} \right\}$$

- Then we have $\vec{v} \in \mathcal{L}_{lin}$ if and only if $\vec{v} \in \mathrm{span}\langle \vec{v}_1, \ldots, \vec{v}_t \rangle$

Language-Dependent CRS

- Noted $\phi \leftarrow K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$: generate $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{KeyGen}(\Gamma)$ of Π_{LH}, $\sigma_j \leftarrow \mathrm{Sign}_{LH}(\mathrm{sk}, \vec{v}_j)$ for each $j = 1$ to t, and return $\phi = (\mathrm{pk}, \{\sigma_j\}_{j=1}^t)$

Our Basic Quasi-Adaptive NIZK Proof (1)

Fixed Common Reference String

- Noted $\Gamma \leftarrow K_0(\lambda)$: pairing description $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$

Language for linear subspace membership

$$\mathcal{L}_{lin}(\Gamma) = \left\{ \vec{v} \in \mathbb{G}^n \,|\, \exists(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t} \right\}$$

- Then we have $\vec{v} \in \mathcal{L}_{lin}$ if and only if $\vec{v} \in \mathrm{span}\langle \vec{v}_1, \ldots, \vec{v}_t \rangle$

Language-Dependent CRS

- Noted $\phi \leftarrow K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$: generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\Gamma)$ of Π_{LH}, $\sigma_j \leftarrow \mathsf{Sign}_{LH}(\mathsf{sk}, \vec{v}_j)$ for each $j = 1$ to t, and return $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$

Our Basic Quasi-Adaptive NIZK Proof (1)

Fixed Common Reference String

- Noted $\Gamma \leftarrow K_0(\lambda)$: pairing description $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$

Language for linear subspace membership

$$\mathcal{L}_{lin}(\Gamma) = \left\{ \vec{v} \in \mathbb{G}^n \,|\, \exists(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v}_1^{\,\alpha_1} \cdots \vec{v}_t^{\,\alpha_t} \right\}$$

- Then we have $\vec{v} \in \mathcal{L}_{lin}$ if and only if $\vec{v} \in \mathrm{span}\langle \vec{v}_1, \ldots, \vec{v}_t \rangle$

Language-Dependent CRS

- Noted $\phi \leftarrow K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$: generate $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{KeyGen}(\Gamma)$ of Π_{LH}, $\sigma_j \leftarrow \mathrm{Sign}_{LH}(\mathrm{sk}, \vec{v}_j)$ for each $j = 1$ to t, and return $\phi = (\mathrm{pk}, \{\sigma_j\}_{j=1}^t)$

Our Basic Quasi-Adaptive NIZK Proof (2)

The CRS generation and the language

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $\mathcal{L}_{lin}(\Gamma) = \left\{\vec{v} \in \mathbb{G}^n \mid \exists(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t}\right\}$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$

The Prover P

- $P(\Gamma, \phi, \vec{v}, w)$: for $\vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t}$ where $w = (\alpha_1, \ldots, \alpha_t)$ simply derive and ouptut a signature $\pi = \sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$

The Verifier V

- $V(\Gamma, \phi, \vec{v}, \pi)$: simply return $b = \mathsf{Verify}_{LH}(\mathsf{pk}, \vec{v}, \sigma)$ for $\sigma = \pi$

The signing secret key sk allows simulating proofs

Our Basic Quasi-Adaptive NIZK Proof (2)

The CRS generation and the language

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $\mathcal{L}_{lin}(\Gamma) = \left\{\vec{v} \in \mathbb{G}^n \mid \exists(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t}\right\}$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$

The Prover P

- $P(\Gamma, \phi, \vec{v}, w)$: for $\vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t}$ where $w = (\alpha_1, \ldots, \alpha_t)$ simply derive and ouptut a signature $\pi = \sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$

The Verifier V

- $V(\Gamma, \phi, \vec{v}, \pi)$: simply return $b = \mathsf{Verify}_{LH}(\mathsf{pk}, \vec{v}, \sigma)$ for $\sigma = \pi$

The signing secret key sk allows simulating proofs

Our Basic Quasi-Adaptive NIZK Proof (2)

The CRS generation and the language

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $\mathcal{L}_{lin}(\Gamma) = \left\{\vec{v} \in \mathbb{G}^n \mid \exists(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t}\right\}$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$

The Prover P

- $P(\Gamma, \phi, \vec{v}, w)$: for $\vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t}$ where $w = (\alpha_1, \ldots, \alpha_t)$ simply derive and ouptut a signature $\pi = \sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$

The Verifier V

- $V(\Gamma, \phi, \vec{v}, \pi)$: simply return $b = \mathsf{Verify}_{LH}(\mathsf{pk}, \vec{v}, \sigma)$ for $\sigma = \pi$

The signing secret key sk allows simulating proofs

Our Basic Quasi-Adaptive NIZK Proof (2)

The CRS generation and the language

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $\mathcal{L}_{lin}(\Gamma) = \left\{ \vec{v} \in \mathbb{G}^n \,|\, \exists(\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_p^t : \vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t} \right\}$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$

The Prover P

- $P(\Gamma, \phi, \vec{v}, w)$: for $\vec{v} = \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t}$ where $w = (\alpha_1, \ldots, \alpha_t)$ simply derive and ouptut a signature $\pi = \sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$

The Verifier V

- $V(\Gamma, \phi, \vec{v}, \pi)$: simply return $b = \mathsf{Verify}_{LH}(\mathsf{pk}, \vec{v}, \sigma)$ for $\sigma = \pi$

The signing secret key sk allows simulating proofs

Unbounded Simulation-Sound QA-NIZK

The CRS generation for $\mathcal{L}_{lin}$

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$ with a Groth-Sahai CRS $\vec{f}_1$, $\vec{f}_2$ and $\{\vec{f}_{3,i}\}_{i=0}^L$ and a one-time signature generator $\mathcal{G}$

The Prover P

- Derive a signature $\sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$ on $\vec{v}$
- Generate a one-time key-pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}$
- Groth-Sahai NIWI proof of a valid LH signature $\sigma = (z, r, u)$ where the CRS_{GS} is assembled as $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_{\mathsf{VK}} = \vec{f}_{3,0} \cdot \prod_{i=1}^L \vec{f}_{3,i}^{\,\mathsf{VK}[i]}$
- Output VK and the signed proof π_{GS} using one-time SK

NM-proof of knowledge of a malleable proof of membership

Unbounded Simulation-Sound QA-NIZK

The CRS generation for $\mathcal{L}_{lin}$

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$ with a Groth-Sahai CRS $\vec{f}_1$, $\vec{f}_2$ and $\{\vec{f}_{3,i}\}_{i=0}^L$ and a one-time signature generator $\mathcal{G}$

The Prover P

- Derive a signature $\sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$ on $\vec{v}$
- Generate a one-time key-pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}$
- Groth-Sahai NIWI proof of a valid LH signature $\sigma = (z, r, u)$ where the CRS_{GS} is assembled as $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_{\mathsf{VK}} = \vec{f}_{3,0} \cdot \prod_{i=1}^L \vec{f}_{3,i}^{\,\mathsf{VK[i]}}$
- Output VK and the signed proof π_{GS} using one-time SK

NM-proof of knowledge of a malleable proof of membership

Unbounded Simulation-Sound QA-NIZK

The CRS generation for $\mathcal{L}_{lin}$

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$ with a Groth-Sahai CRS $\vec{f}_1$, $\vec{f}_2$ and $\{\vec{f}_{3,i}\}_{i=0}^L$ and a one-time signature generator $\mathcal{G}$

The Prover P

- Derive a signature $\sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$ on $\vec{v}$
- Generate a one-time key-pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}$
- Groth-Sahai NIWI proof of a valid LH signature $\sigma = (z, r, u)$ where the CRS_{GS} is assembled as $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_{\mathsf{VK}} = \vec{f}_{3,0} \cdot \prod_{i=1}^L \vec{f}_{3,i}^{\,\mathsf{VK}[i]}$
- Output VK and the signed proof π_{GS} using one-time SK

NM-proof of knowledge of a malleable proof of membership

Unbounded Simulation-Sound QA-NIZK

The CRS generation for $\mathcal{L}_{lin}$

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$ with a Groth-Sahai CRS $\vec{f}_1$, $\vec{f}_2$ and $\{\vec{f}_{3,i}\}_{i=0}^L$ and a one-time signature generator $\mathcal{G}$

The Prover P

- Derive a signature $\sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$ on $\vec{v}$
- Generate a one-time key-pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}$
- Groth-Sahai NIWI proof of a valid LH signature $\sigma = (z, r, u)$ where the CRS_{GS} is assembled as $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_{\mathsf{VK}} = \vec{f}_{3,0} \cdot \prod_{i=1}^L \vec{f}_{3,i}^{\mathsf{VK}[i]}$
- Output VK and the signed proof π_{GS} using one-time SK

NM-proof of knowledge of a malleable proof of membership

Unbounded Simulation-Sound QA-NIZK

The CRS generation for $\mathcal{L}_{lin}$

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$ with a Groth-Sahai CRS $\vec{f}_1$, $\vec{f}_2$ and $\{\vec{f}_{3,i}\}_{i=0}^L$ and a one-time signature generator $\mathcal{G}$

The Prover P

- Derive a signature $\sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$ on $\vec{v}$
- Generate a one-time key-pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}$
- Groth-Sahai NIWI proof of a valid LH signature $\sigma = (z, r, u)$ where the CRS_{GS} is assembled as $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_{\mathsf{VK}} = \vec{f}_{3,0} \cdot \prod_{i=1}^L \vec{f}_{3,i}^{\,\mathsf{VK[i]}}$
- Output VK and the signed proof π_{GS} using one-time SK

NM-proof of knowledge of a malleable proof of membership

Unbounded Simulation-Sound QA-NIZK

The CRS generation for $\mathcal{L}_{lin}$

- $K_0(\lambda)$ returns $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g)$ with $p > 2^\lambda$
- $K_1(\Gamma, \vec{v}_1, \ldots, \vec{v}_t)$ returns $\phi = (\mathsf{pk}, \{\sigma_j\}_{j=1}^t)$ with a Groth-Sahai CRS $\vec{f}_1$, $\vec{f}_2$ and $\{\vec{f}_{3,i}\}_{i=0}^L$ and a one-time signature generator $\mathcal{G}$

The Prover P

- Derive a signature $\sigma \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \vec{v}_1^{\alpha_1} \cdots \vec{v}_t^{\alpha_t})$ on $\vec{v}$
- Generate a one-time key-pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}$
- Groth-Sahai NIWI proof of a valid LH signature $\sigma = (z, r, u)$ where the CRS_{GS} is assembled as $\vec{f}_1$, $\vec{f}_2$ and $\vec{f}_{\mathsf{VK}} = \vec{f}_{3,0} \cdot \prod_{i=1}^L \vec{f}_{3,i}^{\,\mathsf{VK}[i]}$
- Output VK and the signed proof π_{GS} using one-time SK

NM-proof of knowledge of a malleable proof of membership

CCA-secure Keyed-Homomorphic Encryption

Keyed homomorphic encryption: (Emura et al., PKC'13)

- Homomorphic evaluation requires an evaluation key SK_h
- System remains CCA1 if SK_h is exposed and CCA2 otherwise!

Intuition of our DLIN-based encryption

- $\mathrm{Enc}_0(m) = (c_1, c_2, c_3, c_4) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot {X_1}^{r_1} {X_2}^{r_2})$

  and set $\vec{c} = (c_1, c_2, c_3)$, $\vec{v}_1 = (h_1, 1, g)$, $\vec{v}_2 = (1, h_2, g)$
- $\pi_{ZK} \leftarrow P_{ZK}(\phi_1, \vec{c})$ is our QA-NIZK proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$
- $\pi_{SS} \leftarrow P_{SS}(\phi_2, \vec{c})$ is our USS QA-proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$

SK_h is the signing secret key related to ϕ_2

CCA-secure Keyed-Homomorphic Encryption

Keyed homomorphic encryption: (Emura et al., PKC'13)

- Homomorphic evaluation requires an evaluation key SK_h
- System remains CCA1 if SK_h is exposed and CCA2 otherwise!

Intuition of our DLIN-based encryption

- $\mathsf{Enc}_0(m) = (c_1, c_2, c_3, c_4) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot X_1{}^{r_1} X_2{}^{r_2})$ and set $\vec{c} = (c_1, c_2, c_3)$, $\vec{v}_1 = (h_1, 1, g)$, $\vec{v}_2 = (1, h_2, g)$
- $\pi_{ZK} \leftarrow P_{ZK}(\phi_1, \vec{c})$ is our QA-NIZK proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$
- $\pi_{SS} \leftarrow P_{SS}(\phi_2, \vec{c})$ is our USS QA-proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$

SK_h is the signing secret key related to ϕ_2

CCA-secure Keyed-Homomorphic Encryption

Keyed homomorphic encryption: (Emura et al., PKC'13)

- Homomorphic evaluation requires an evaluation key SK_h
- System remains CCA1 if SK_h is exposed and CCA2 otherwise!

Intuition of our DLIN-based encryption

- $\mathsf{Enc}_0(m) = (c_1, c_2, c_3, c_4) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot {X_1}^{r_1} {X_2}^{r_2})$ and set $\vec{c} = (c_1, c_2, c_3)$, $\vec{v}_1 = (h_1, 1, g)$, $\vec{v}_2 = (1, h_2, g)$
- $\pi_{ZK} \leftarrow P_{ZK}(\phi_1, \vec{c})$ is our QA-NIZK proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$
- $\pi_{SS} \leftarrow P_{SS}(\phi_2, \vec{c})$ is our USS QA-proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$

SK_h is the signing secret key related to ϕ_2

CCA-secure Keyed-Homomorphic Encryption

Keyed homomorphic encryption: (Emura et al., PKC'13)

- Homomorphic evaluation requires an evaluation key SK_h
- System remains CCA1 if SK_h is exposed and CCA2 otherwise!

Intuition of our DLIN-based encryption

- $\mathrm{Enc}_0(m) = (c_1, c_2, c_3, c_4) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot {X_1}^{r_1} {X_2}^{r_2})$
  and set $\vec{c} = (c_1, c_2, c_3)$, $\vec{v}_1 = (h_1, 1, g)$, $\vec{v}_2 = (1, h_2, g)$
- $\pi_{ZK} \leftarrow P_{ZK}(\phi_1, \vec{c})$ is our QA-NIZK proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$
- $\pi_{SS} \leftarrow P_{SS}(\phi_2, \vec{c})$ is our USS QA-proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$

SK_h is the signing secret key related to ϕ_2

CCA-secure Keyed-Homomorphic Encryption

Keyed homomorphic encryption: (Emura et al., PKC'13)

- Homomorphic evaluation requires an evaluation key SK_h
- System remains CCA1 if SK_h is exposed and CCA2 otherwise!

Intuition of our DLIN-based encryption

- $\mathsf{Enc}_0(m) = (c_1, c_2, c_3, c_4) = (h_1^{r_1}, h_2^{r_2}, g^{r_1+r_2}, m \cdot {X_1}^{r_1} {X_2}^{r_2})$
  and set $\vec{c} = (c_1, c_2, c_3)$, $\vec{v}_1 = (h_1, 1, g)$, $\vec{v}_2 = (1, h_2, g)$
- $\pi_{ZK} \leftarrow P_{ZK}(\phi_1, \vec{c})$ is our QA-NIZK proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$
- $\pi_{SS} \leftarrow P_{SS}(\phi_2, \vec{c})$ is our USS QA-proof that $\vec{c} \in \mathcal{L}_{lin}(\vec{v}_1, \vec{v}_2)$

SK_h is the signing secret key related to ϕ_2

Comparison between proof systems for linear subspaces

For n equations and t variables, under DLIN:

Proof systems	CRS size	Proof length	Verification
Groth-Sahai [EC'08]	6	$3t + 2n$	$3n(t+3)$
Jutla-Roy [AC'13]	$4t(n-t)+3$	$2(n-t)$	2(n-t)(t+2)
Jutla-Roy RSS ...+[PKC'12]	$4t(n+1-t)+3$	$2(n+1-t)+1$	$2(n+1-t)(t+2)$
Groth-Sahai USS [EC'09]	18	$6t+2n+52$	$O(tn)$
New basic QA-NIZK proofs	$2n+3t+4$	3	$2n+2$
New RSS QA-NIZK proofs	$4n+8t+6$	4	$2n+6$
New USS QA-NIZK proofs	$2n+3t+3L+10$	20	$2n+30$

L : length of a hashed one-time verifying key

Sizes are measured in terms of group elements

Conclusion

Our results:

- Standard-model QA-NIZK proofs can be shorter than RO-based proofs, even under standard assumptions
- $O(1)$-size proofs of linear subspace membership (regardless of subspace dimensions) improved by Jutla-Roy [ePrint Report 2013/670]
- Unbounded simulation-soundness with $O(1)$ elements per proof (without quadratic equations or CCA-secure encryptions)

Applications

- CCA2-secure keyed homomorphic encryption with publicly verifiable ciphertexts (and threshold decryption)
- Publicly verifiable threshold adaptively-secure CCA cryptosystems

Conclusion

Our results:

- Standard-model QA-NIZK proofs can be shorter than RO-based proofs, even under standard assumptions
- $O(1)$-size proofs of linear subspace membership (regardless of subspace dimensions) improved by Jutla-Roy [ePrint Report 2013/670]
- Unbounded simulation-soundness with $O(1)$ elements per proof (without quadratic equations or CCA-secure encryptions)

Applications

- CCA2-secure keyed homomorphic encryption with publicly verifiable ciphertexts (and threshold decryption)
- Publicly verifiable threshold adaptively-secure CCA cryptosystems

Conclusion

Our results:

- Standard-model QA-NIZK proofs can be shorter than RO-based proofs, even under standard assumptions
- $O(1)$-size proofs of linear subspace membership (regardless of subspace dimensions) improved by Jutla-Roy [ePrint Report 2013/670]
- Unbounded simulation-soundness with $O(1)$ elements per proof (without quadratic equations or CCA-secure encryptions)

Applications

- CCA2-secure keyed homomorphic encryption with publicly verifiable ciphertexts (and threshold decryption)
- Publicly verifiable threshold adaptively-secure CCA cryptosystems

Conclusion

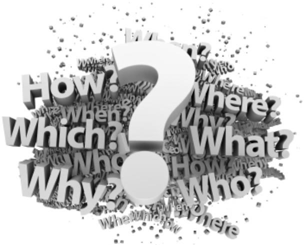
Our results:

- Standard-model QA-NIZK proofs can be shorter than RO-based proofs, even under standard assumptions
- $O(1)$-size proofs of linear subspace membership (regardless of subspace dimensions) improved by Jutla-Roy [ePrint Report 2013/670]
- Unbounded simulation-soundness with $O(1)$ elements per proof (without quadratic equations or CCA-secure encryptions)

Applications

- CCA2-secure keyed homomorphic encryption with publicly verifiable ciphertexts (and threshold decryption)
- Publicly verifiable threshold adaptively-secure CCA cryptosystems

Thank you!



Questions?

Verifiable Encryption ?

- BBS ciphertext of $M \in \mathbb{G}$ has the form $(f^r, h^s, M \cdot g^{r+s})$

  CPA secure under the DLIN assumption:

  $$(f^r, h^s, Z) \in span\langle (f, 1, g), (1, h, g) \rangle ?$$

- DLIN-based CCA encryption "*à la Cramer-Shoup*"

  $$(C_1, C_2, C_3, C_0) := (f^r, h^s, g^{r+s}, M \cdot X^r Y^s)$$

  with a proof that indeed (C_1, C_2, C_3) is a linear tuple

Idea: Replace the proof with a linear signature

. . . but only gives a homomorphic CCA-1 cryptosystem

Verifiable Encryption ?

- BBS ciphertext of $M \in \mathbb{G}$ has the form $(f^r, h^s, M \cdot g^{r+s})$

  CPA secure under the DLIN assumption:

$$(f^r, h^s, Z) \in span\langle (f, 1, g), (1, h, g) \rangle ?$$

- DLIN-based CCA encryption "*à la Cramer-Shoup*"

$$(C_1, C_2, C_3, C_0) := (f^r, h^s, g^{r+s}, M \cdot X^r Y^s)$$

  with a proof that indeed (C_1, C_2, C_3) is a linear tuple

Idea: Replace the proof with a linear signature

. . . but only gives a homomorphic CCA-1 cryptosystem

Verifiable Encryption ?

- BBS ciphertext of $M \in \mathbb{G}$ has the form $(f^r, h^s, M \cdot g^{r+s})$

  CPA secure under the DLIN assumption:

  $$(f^r, h^s, Z) \in span\langle(f, 1, g), (1, h, g)\rangle?$$

- DLIN-based CCA encryption "*à la Cramer-Shoup*"

  $$(C_1, C_2, C_3, C_0) := (f^r, h^s, g^{r+s}, M \cdot X^r Y^s)$$

  with a proof that indeed (C_1, C_2, C_3) is a linear tuple

Idea: Replace the proof with a linear signature

. . . but only gives a homomorphic CCA-1 cryptosystem

Verifiable Encryption ?

- BBS ciphertext of $M \in \mathbb{G}$ has the form $(f^r, h^s, M \cdot g^{r+s})$

  CPA secure under the DLIN assumption:

  $$(f^r, h^s, Z) \in span\langle(f, 1, g), (1, h, g)\rangle?$$

- DLIN-based CCA encryption "*à la Cramer-Shoup*"

  $$(C_1, C_2, C_3, C_0) := (f^r, h^s, g^{r+s}, M \cdot X^r Y^s)$$

  with a proof that indeed (C_1, C_2, C_3) is a linear tuple

Idea: Replace the proof with a linear signature

. . . but only gives a homomorphic CCA-1 cryptosystem

Verifiable Encryption ?

- BBS ciphertext of $M \in \mathbb{G}$ has the form $(f^r, h^s, M \cdot g^{r+s})$

  CPA secure under the DLIN assumption:

  $$(f^r, h^s, Z) \in span\langle (f, 1, g), (1, h, g) \rangle ?$$

- DLIN-based CCA encryption "*à la Cramer-Shoup*"

  $$(C_1, C_2, C_3, C_0) := (f^r, h^s, g^{r+s}, M \cdot X^r Y^s)$$

  with a proof that indeed (C_1, C_2, C_3) is a linear tuple

Idea: Replace the proof with a linear signature

. . . but only gives a homomorphic CCA-1 cryptosystem

Adaptively Secure CCA Threshold Encryption

Keep $(C_0, C_1, C_2, C_3) = (M \cdot X_1^{r_1} X_2^{r_2}, f^{r_1}, h^{r_2}, g^{r_1+r_2})$ and add

- SPHF: $C_4 = (Y_1 Y_2^{\alpha})^{r_1} (W_1 W_2^{\alpha})^{r_2}$ where $\alpha \leftarrow H(C_0, C_1, C_2, C_3)$
- LH signature on $(C_1, C_2, C_3, C_4, C_1^{\alpha}, C_2^{\alpha}, C_3^{\alpha})$ using those on

$$\vec{f_1} = (f, 1, g, Y_1, 1, 1, 1) \qquad \vec{h_1} = (1, h, g, W_1, 1, 1, 1)$$

$$\vec{f_2} = (1, 1, 1, Y_2, f, 1, g) \qquad \vec{h_1} = (1, 1, 1, W_2, 1, h, g)$$

$\Longrightarrow$ This results in ciphertext composed of 8 $\mathbb{G}$-elements

Generalization:

- Relatively-Sound Quasi-Adaptive NIZK (for linear subspace)

Adaptively Secure CCA Threshold Encryption

Keep $(C_0, C_1, C_2, C_3) = (M \cdot X_1^{r_1} X_2^{r_2}, f^{r_1}, h^{r_2}, g^{r_1+r_2})$ and add

- SPHF: $C_4 = (Y_1 Y_2^{\alpha})^{r_1} (W_1 W_2^{\alpha})^{r_2}$ where $\alpha \leftarrow H(C_0, C_1, C_2, C_3)$
- LH signature on $(C_1, C_2, C_3, C_4, C_1^{\alpha}, C_2^{\alpha}, C_3^{\alpha})$ using those on

$$\vec{f_1} = (f, 1, g, Y_1, 1, 1, 1) \qquad \vec{h_1} = (1, h, g, W_1, 1, 1, 1)$$
$$\vec{f_2} = (1, 1, 1, Y_2, f, 1, g) \qquad \vec{h_1} = (1, 1, 1, W_2, 1, h, g)$$

$\Longrightarrow$ This results in ciphertext composed of 8 $\mathbb{G}$-elements

Generalization:

- Relatively-Sound Quasi-Adaptive NIZK (for linear subspace)

Adaptively Secure CCA Threshold Encryption

Keep $(C_0, C_1, C_2, C_3) = (M \cdot X_1^{r_1} X_2^{r_2}, f^{r_1}, h^{r_2}, g^{r_1+r_2})$ and add

- SPHF: $C_4 = (Y_1 Y_2^{\alpha})^{r_1} (W_1 W_2^{\alpha})^{r_2}$ where $\alpha \leftarrow H(C_0, C_1, C_2, C_3)$
- LH signature on $(C_1, C_2, C_3, C_4, C_1^{\alpha}, C_2^{\alpha}, C_3^{\alpha})$ using those on

$$\vec{f_1} = (f, 1, g, Y_1, 1, 1, 1) \qquad \vec{h_1} = (1, h, g, W_1, 1, 1, 1)$$
$$\vec{f_2} = (1, 1, 1, Y_2, f, 1, g) \qquad \vec{h_1} = (1, 1, 1, W_2, 1, h, g)$$

$\Longrightarrow$ This results in ciphertext composed of 8 $\mathbb{G}$-elements

Generalization:

- Relatively-Sound Quasi-Adaptive NIZK (for linear subspace)

Adaptively Secure CCA Threshold Encryption

Keep $(C_0, C_1, C_2, C_3) = (M \cdot X_1^{r_1} X_2^{r_2}, f^{r_1}, h^{r_2}, g^{r_1+r_2})$ and add

- SPHF: $C_4 = (Y_1 Y_2^{\alpha})^{r_1} (W_1 W_2^{\alpha})^{r_2}$ where $\alpha \leftarrow H(C_0, C_1, C_2, C_3)$
- LH signature on $(C_1, C_2, C_3, C_4, C_1^{\alpha}, C_2^{\alpha}, C_3^{\alpha})$ using those on

$$\vec{f_1} = (f, 1, g, Y_1, 1, 1, 1) \qquad \vec{h_1} = (1, h, g, W_1, 1, 1, 1)$$
$$\vec{f_2} = (1, 1, 1, Y_2, f, 1, g) \qquad \vec{h_1} = (1, 1, 1, W_2, 1, h, g)$$

$\Longrightarrow$ This results in ciphertext composed of 8 $\mathbb{G}$-elements

Generalization:

- Relatively-Sound Quasi-Adaptive NIZK (for linear subspace)

Adaptively Secure CCA Threshold Encryption

Keep $(C_0, C_1, C_2, C_3) = (M \cdot X_1^{r_1} X_2^{r_2}, f^{r_1}, h^{r_2}, g^{r_1+r_2})$ and add

- SPHF: $C_4 = (Y_1 Y_2^{\alpha})^{r_1} (W_1 W_2^{\alpha})^{r_2}$ where $\alpha \leftarrow H(C_0, C_1, C_2, C_3)$
- LH signature on $(C_1, C_2, C_3, C_4, C_1^{\alpha}, C_2^{\alpha}, C_3^{\alpha})$ using those on

$$\vec{f_1} = (f, 1, g, Y_1, 1, 1, 1) \qquad \vec{h_1} = (1, h, g, W_1, 1, 1, 1)$$
$$\vec{f_2} = (1, 1, 1, Y_2, f, 1, g) \qquad \vec{h_1} = (1, 1, 1, W_2, 1, h, g)$$

$\Longrightarrow$ This results in ciphertext composed of 8 $\mathbb{G}$-elements

Generalization:

- Relatively-Sound Quasi-Adaptive NIZK (for linear subspace)

Construction

Hardness Assumptions

Our schemes provide security under the following assumptions

1. The **Simultaneous Double Pairing Problem (SDP)**: given $(g_z, h_z, g_r, h_r) \in \mathbb{G}^4$ it is hard to compute $(z, r, u) \in \mathbb{G}^3$, with $z \neq 1_{\mathbb{G}}$, verifying

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r)$$
$$1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_r, u)$$

2. The **Decision Linear Problem (DLIN)**: given $(g, g^a, g^b, g^{ac}, g^{bd}, g^{\eta}) \in \mathbb{G}^6$, decide whether $\eta = c + d$ or $\eta \in_R \mathbb{Z}_p$

Known result: DLIN implies SDP

Construction

Hardness Assumptions

Our schemes provide security under the following assumptions

1. The **Simultaneous Double Pairing Problem (SDP)**: given $(g_z, h_z, g_r, h_r) \in \mathbb{G}^4$ it is hard to compute $(z, r, u) \in \mathbb{G}^3$, with $z \neq 1_{\mathbb{G}}$, verifying

$$\begin{aligned} 1_{\mathbb{G}_T} &= e(g_z, z) \cdot e(g_r, r) \\ 1_{\mathbb{G}_T} &= e(h_z, z) \cdot e(h_r, u) \end{aligned}$$

2. The **Decision Linear Problem (DLIN)**: given $(g, g^a, g^b, g^{ac}, g^{bd}, g^{\eta}) \in \mathbb{G}^6$, decide whether $\eta = c + d$ or $\eta \in_R \mathbb{Z}_p$

Known result: DLIN implies SDP