Non-Interactive CCA2-Secure Threshold Cryptosystems: Achieving Adaptive Security in the Standard Model Without Pairings

Julien Devevey[1] Benoît Libert[2,1] Khoa Nguyen[3] Thomas Peters[4] Moti Yung[5]

ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL), France

CNRS, Laboratoire LIP, France

Nanyang Technological University, SPMS, Singapore

FNRS and Université catholique de Louvain, Belgium

Google and Columbia University, USA

Table of contents

1. Introduction

2. Definitions

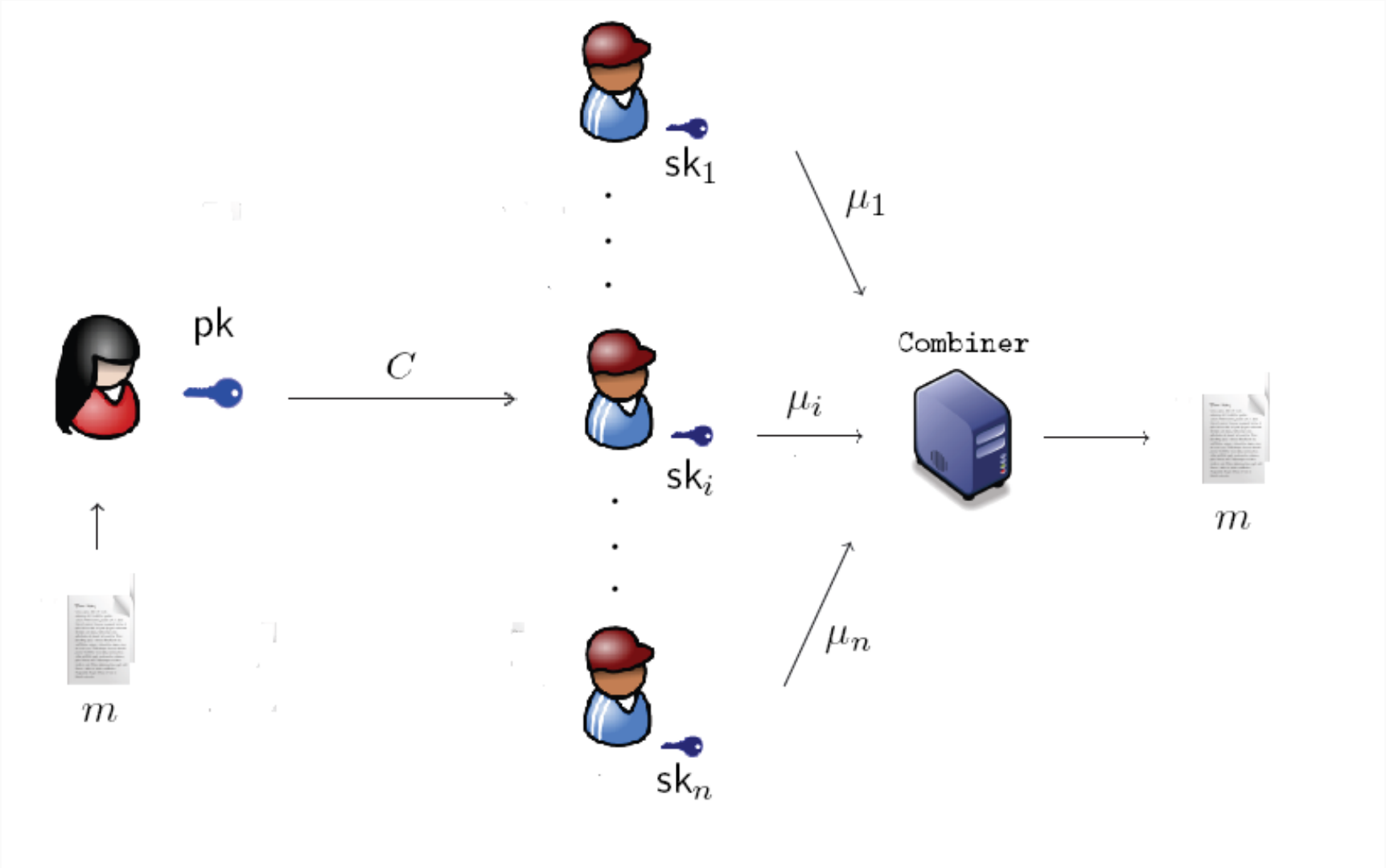
3. Constructions

    Based on Decision Composite Residuosity

    Based on Learning With Errors: Threshold Dual Regev

Introduction

Threshold Cryptography



Our goal

Build a Threshold Public-Key Encryption scheme satisfying:

- Compactness: size of C and pk independent of the number of servers,
- IND-CCA2 security, as in non-threshold PKE,
- … under adaptive corruptions: the adversary can obtain any sk_i, at any time.
- Without using pairings.

Main results and previous works

Construction	Assumption	Adaptive	IND-CCA2	Compactness
[SG98]	CDH/DDH	✗	✓ (ROM)	✓
[FP01]	DDH	✓	✓ (ROM)	✓
[BBH06]	DBDH*	✗	✓	✓
[LY12]	SXDH*	✓	✓	✓
[BGG+18]	FHE (LWE)	✗	✓	✓
This work (1)	LWE & DCR	✓	✓	✓
This work (2)	LWE	✓	✓	✓

*: In a group with pairings.

Ciphertext size:

- Construction (1): About three times the size of a Camenisch-Shoup encryption
- Construction (2): Super-polynomial modulus (but quantum-safe)

Main results and previous works

Construction	Assumption	Adaptive	IND-CCA2	Compactness
[SG98]	CDH/DDH	✗	✓ (ROM)	✓
[FP01]	DDH	✓	✓ (ROM)	✓
[BBH06]	DBDH*	✗	✓	✓
[LY12]	SXDH*	✓	✓	✓
[BGG+18]	FHE (LWE)	✗	✓	✓
This work (1)	LWE & DCR	✓	✓	✓
This work (2)	LWE	✓	✓	✓

*: In a group with pairings.

Ciphertext size:

- Construction (1): About three times the size of a Camenisch-Shoup encryption
- Construction (2): Super-polynomial modulus (but quantum-safe)

Definitions

Threshold Public-Key Encryption

A compact TPKE is a 5-uple (KeyGen, Enc, PartDec, PartVerify, Combine) of algorithms that interact the following way:

Server i	Alice	Bob
	$(\mathsf{pk}, \{\mathsf{sk}_i\}) \leftarrow \mathsf{KeyGen}(1^\lambda, t)$	
	$\longleftarrow \mathsf{sk}_i$	$\mathsf{pk} \longrightarrow$
		$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
		$\longleftarrow \mathsf{ct}$
	$\longleftarrow \mathsf{ct}$	
$m_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$		
	$m_i \longrightarrow$	
	$\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}, m_i)$	
	$m' \leftarrow \mathsf{Combine}(\mathsf{pk}, \{m_j\}_{j \in \mathcal{S}}, \mathsf{ct})$	

Under the condition that $|\mathsf{pk}|, |\mathsf{ct}| = \mathsf{poly}(\lambda)$.

It is correct if $\forall |\mathcal{S}| \geq t, m = m'$ with proba $\geq 1 - \mathsf{negl}(\lambda)$.

Threshold Public-Key Encryption

A compact TPKE is a 5-uple (KeyGen, Enc, PartDec, PartVerify, Combine) of algorithms that interact the following way:

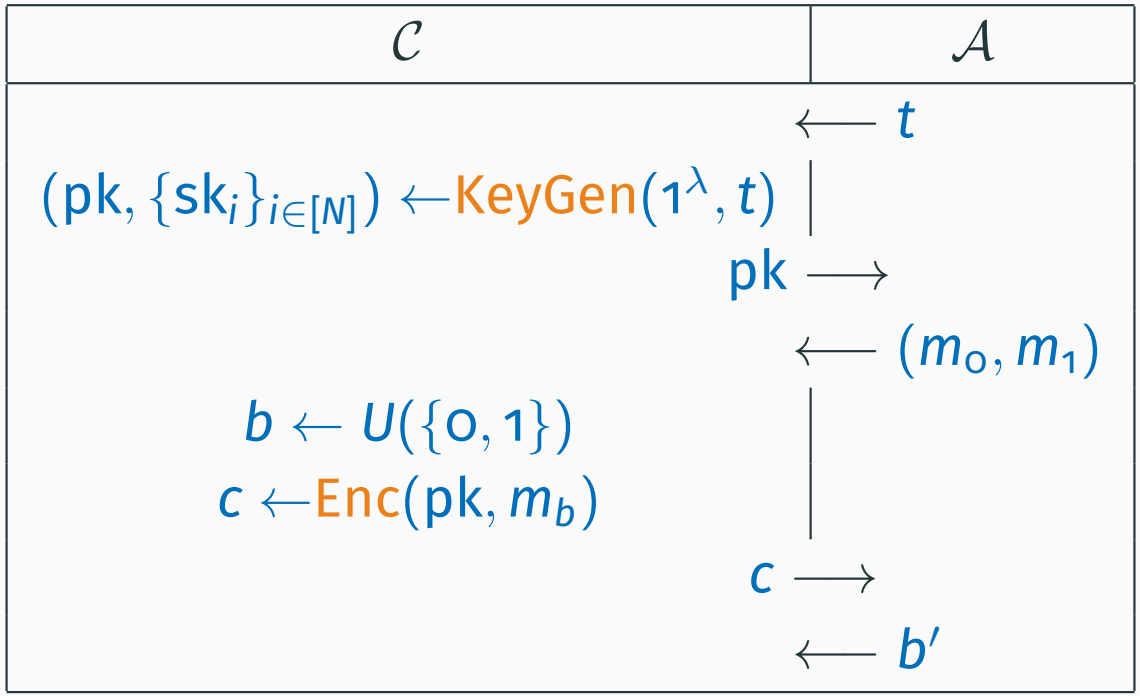
Server i	Alice	Bob
	$(\mathsf{pk}, \{\mathsf{sk}_i\}) \leftarrow \mathsf{KeyGen}(1^\lambda, t)$	
	$\longleftarrow \mathsf{sk}_i$	$\mathsf{pk} \longrightarrow$
		$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
		$\longleftarrow \mathsf{ct}$
	$\longleftarrow \mathsf{ct}$	
$m_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$		
	$m_i \longrightarrow$	
	$\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}, m_i)$	
	$m' \leftarrow \mathsf{Combine}(\mathsf{pk}, \{m_j\}_{j \in \mathcal{S}}, \mathsf{ct})$	

Under the condition that $|\mathsf{pk}|, |\mathsf{ct}| = \mathsf{poly}(\lambda)$.

It is correct if $\forall |\mathcal{S}| \geq t, m = m'$ with proba $\geq 1 - \mathsf{negl}(\lambda)$.

Adaptive IND-CCA2 security for TPKE

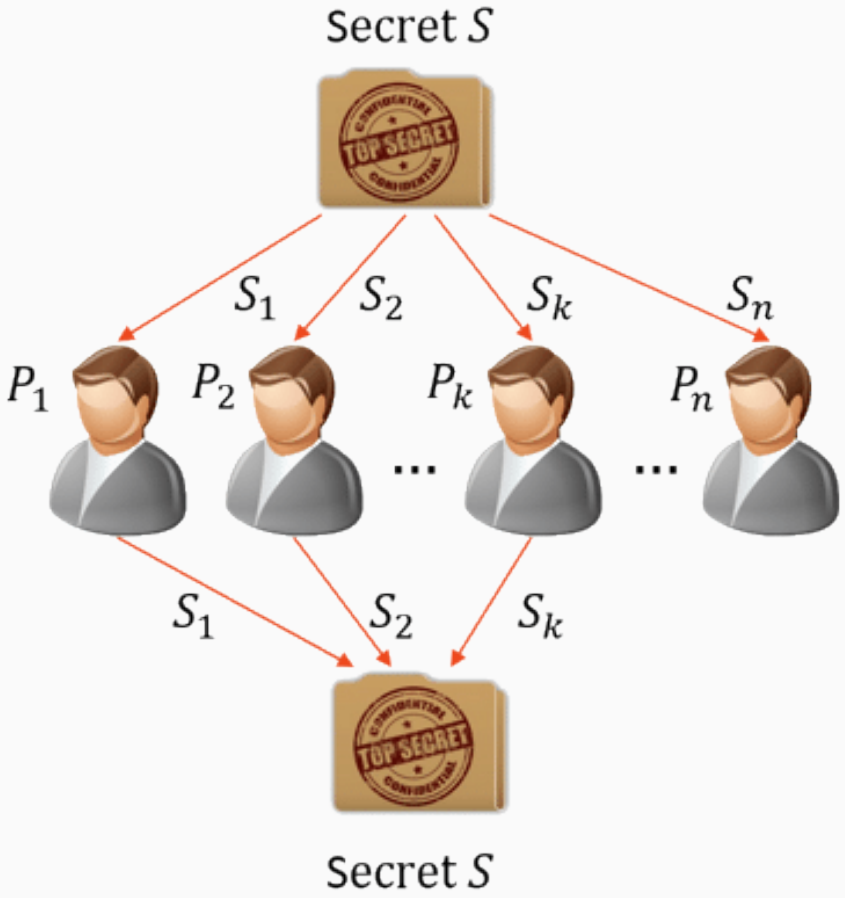
No PPT adversary $\mathcal{A}$ with a $\mathsf{PartDec}(\mathsf{sk}_i, \cdot)$ oracle for any $i \in [\ell]$ has non-negligible advantage:



- $\mathcal{A}$ can obtain any sk_i at any time,
- $\mathcal{A}$ can make partial decryption queries (i, c) for the challenge,

as long as it cannot trivially win. Its advantage is $|\Pr(b = b') - 1/2|$.

Building block: Linear Integer Secret Sharing



Building block: Linear Integer Secret Sharing

LISS (Damgård-Thorbek; PKC'06)

To share an integer $s \in [-2^l, 2^l]$ among parties $[\ell]$, use a matrix $\mathbf{M} \in \mathbb{Z}^{d \times e}$,

- Choose random $\rho_2, \ldots \rho_e$ and define $\vec{\rho} = (s, \rho_2, \ldots, \rho_e)^\top$
- Compute $\vec{s} = (s_1, \ldots, s_d)^\top = \mathbf{M} \cdot \vec{\rho}$
- Give s_i to party $\psi(i)$ for some function $\psi : [d] \to [\ell]$

Shares $\mathbf{s} \in \mathbb{Z}_q^m$ into $(\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell) \in \mathbb{Z}_q^{d_1 \times m} \times \cdots \times \mathbb{Z}_q^{d_\ell \times m}$ such that for any $\mathcal{S}, |\mathcal{S}| \geq t$, there exist $\vec{\lambda}_i \in \{-1, 0, 1\}^{d_i}$ for $i \in \mathcal{S}$ such that:

$$\sum_{i \in \mathcal{S}} \vec{\lambda}_i^\top \cdot \mathsf{sk}_i = \mathbf{s}.$$

Hardness assumptions

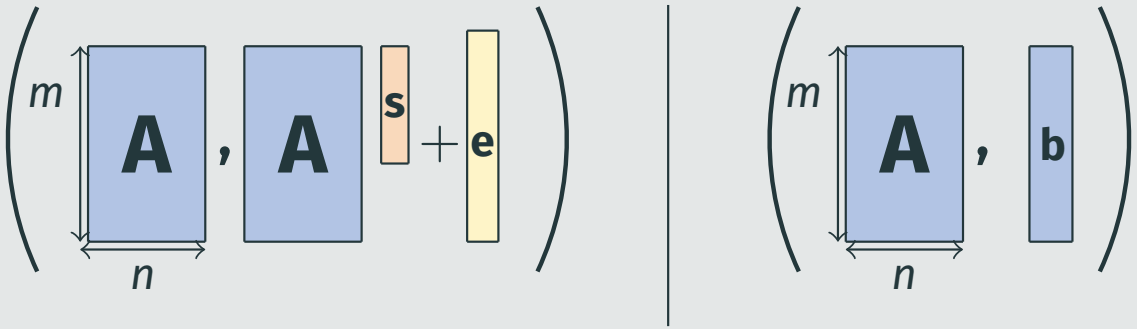
ζ-Decision Composite Residuosity assumption [Pai99, DJ01]

Given $N = pq$ and $\zeta > 1$ for primes p, q. The distributions $\{x = w^{N^\zeta} \bmod N^{\zeta+1} \mid w \leftarrow U(\mathbb{Z}_N^\star)\}$ and $\{x \mid x \leftarrow U(\mathbb{Z}_{N^{\zeta+1}}^\star)\}$ are computationally indistinguishable.

The Learning-With-Errors (LWE) problem (Regev, STOC'05)

Parameters: dimension n, number of samples $m \geq n$, modulus q.

For $\mathbf{A} \hookleftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \hookleftarrow \mathbb{Z}_q^n$ and $\mathbf{e}$ a small error $\approx \alpha q$, distinguish



for uniform $\mathbf{b} \hookleftarrow \mathbb{Z}_q^m$.

Constructions

Construction from DCR+LWE: Intuition

- Pairing-free adaptation of [LY12]
- Exploits the entropy of shared secret keys “à la Cramer-Shoup”; build a DCR-based hash proof system
  (similar to `Camenisch-Shoup; Crypto'03`)
- Ciphertext (C_0, C_1) contains a simulation-sound proof that C_0 is an N^ζ-th residue in $\mathbb{Z}^*_{N^{\zeta+1}}$
- NIZK component instantiated from Fiat-Shamir and CI-hash functions (implied by LWE, cf. `Peikert-Shiehian; Crypto'19`)
- We provide a new construction of one-time simulation-sound (OT-SS) argument from DCR

Based on DCR and LWE

- KeyGen$(1^\lambda, t)$:
  1. Set $N = pq$, where $p, q, \frac{p-1}{2}$ and $\frac{q-1}{2} \geq 2^\lambda$ are primes, and $\zeta \geq 1$.
  2. Generate $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ for a NIZK $\Pi^{OTSS} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ for $\mathcal{L}^{\mathsf{DCR}} := \{x \in \mathbb{Z}^*_{N^{\zeta+1}} \mid \exists w \in \mathbb{Z}^\star_N : x = w^{N^\zeta} \bmod N^{\zeta+1}\}$.
  3. Sample $g_0 \hookleftarrow U(\mathbb{Z}^*_N)$ and set $h = g_0^{4N^\zeta \cdot x} \bmod N^{\zeta+1}$, where $x \hookleftarrow D_{\mathbb{Z},\sigma}$.
  4. LISS: key shares are $\mathsf{sk}_i = \left(\mathbf{M} \cdot \begin{pmatrix} x \\ \rho_1 \\ \vdots \\ \rho_{e-1} \end{pmatrix} \right)_{j \in \psi^{-1}(i)} \in \mathbb{Z}^{d_i}, \forall i \in [\ell]$, where $\rho_j \hookleftarrow D_{\mathbb{Z},\sigma}, \forall j \leq e-1$.

  Output $\mathsf{pk} = (N, \zeta, g_0, h, \mathsf{crs})$ and $(\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$.

Based on DCR and LWE

- KeyGen$(1^\lambda, t)$:
  1. Set $N = pq$, where $p, q, \frac{p-1}{2}$ and $\frac{q-1}{2} \geq 2^\lambda$ are primes, and $\zeta \geq 1$.
  2. Generate $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ for a NIZK $\Pi^{OTSS} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ for $\mathcal{L}^{\mathsf{DCR}} := \{x \in \mathbb{Z}^*_{N^{\zeta+1}} \mid \exists w \in \mathbb{Z}^\star_N : x = w^{N^\zeta} \bmod N^{\zeta+1}\}$.
  3. Sample $g_0 \leftarrow\!\!\!\rightarrow U(\mathbb{Z}^*_N)$ and set $h = g_0^{4N^\zeta \cdot x} \bmod N^{\zeta+1}$, where $x \leftarrow\!\!\!\rightarrow D_{\mathbb{Z},\sigma}$.
  4. LISS: key shares are $\mathsf{sk}_i = \left(\mathbf{M} \cdot \begin{pmatrix} x \\ \rho_1 \\ \vdots \\ \rho_{e-1} \end{pmatrix} \right)_{j \in \psi^{-1}(i)} \in \mathbb{Z}^{d_i}, \forall i \in [\ell]$, where $\rho_j \leftarrow\!\!\!\rightarrow D_{\mathbb{Z},\sigma}, \forall j \leq e-1$.

  Output $\mathsf{pk} = (N, \zeta, g_0, h, \mathsf{crs})$ and $(\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$.

Based on DCR and LWE

- $\mathsf{KeyGen}(1^\lambda, t)$:
  1. Set $N = pq$, where $p, q, \frac{p-1}{2}$ and $\frac{q-1}{2} \geq 2^\lambda$ are primes, and $\zeta \geq 1$.
  2. Generate $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ for a NIZK $\Pi^{OTSS} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ for $\mathcal{L}^{\mathrm{DCR}} := \{x \in \mathbb{Z}^*_{N^{\zeta+1}} \mid \exists w \in \mathbb{Z}^\star_N : x = w^{N^\zeta} \bmod N^{\zeta+1}\}$.
  3. Sample $g_0 \leftarrow\!\!\!\rightarrow U(\mathbb{Z}^*_N)$ and set $h = g_0^{4N^\zeta \cdot x} \bmod N^{\zeta+1}$, where $x \leftarrow\!\!\!\rightarrow D_{\mathbb{Z},\sigma}$.
  4. LISS: key shares are $\mathsf{sk}_i = \left(\mathbf{M} \cdot \begin{pmatrix} x \\ \rho_1 \\ \vdots \\ \rho_{e-1} \end{pmatrix} \right)_{j \in \psi^{-1}(i)} \in \mathbb{Z}^{d_i}, \forall i \in [\ell]$, where $\rho_j \leftarrow\!\!\!\rightarrow D_{\mathbb{Z},\sigma}, \forall j \leq e-1$.

  Output $\mathsf{pk} = (N, \zeta, g_0, h, \mathsf{crs})$ and $(\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$.

Based on DCR and LWE (2)

- Encrypt(pk, Msg): To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$,
  1. Sample $r \hookleftarrow U(\{0, \ldots, \lfloor N/4 \rfloor\})$.
  2. Compute
     $$C_0 = {g_0}^{2N^\zeta \cdot r} \bmod N^{\zeta+1} \text{ and } C_1 = (1+N)^{\mathsf{Msg}} \cdot h^r \bmod N^{\zeta+1}.$$
  3. Compute $\vec{\pi} \leftarrow \mathsf{P}\left(\mathsf{crs}, C_0, g_0^{2r} \bmod N, \mathsf{lbl}\right)$, a proof that $C_0 \in \mathcal{L}^{\mathrm{DCR}}$ using the label $\mathsf{lbl} = C_1$.
  4. Return $\mathsf{ct} := (C_0, C_1, \vec{\pi})$.

Based on DCR and LWE (2)

- Encrypt(pk, Msg): To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$,
  1. Sample $r \hookleftarrow U(\{0, \ldots, \lfloor N/4 \rfloor\})$.
  2. Compute
     $$C_0 = {g_0}^{2N^\zeta \cdot r} \bmod N^{\zeta+1} \text{ and } C_1 = (1+N)^{\mathsf{Msg}} \cdot h^r \bmod N^{\zeta+1}.$$
  3. Compute $\vec{\pi} \leftarrow \mathsf{P}\big(\mathsf{crs}, C_0, g_0^{2r} \bmod N, \mathsf{lbl}\big)$, a proof that $C_0 \in \mathcal{L}^{\mathrm{DCR}}$ using the label $\mathsf{lbl} = C_1$.
  4. Return $\mathsf{ct} := (C_0, C_1, \vec{\pi})$.

Based on DCR and LWE (3)

- $\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$: To decrypt, server i does:
  1. If $\mathsf{V}(\mathsf{crs}, C_0, \vec{\pi}, \mathsf{lbl}) = 0$, return $\perp$.
  2. For each $j \in \psi^{-1}(i) = \{j_1, \ldots, j_{d_i}\}$, compute $\mu_{i,j} = C_0^{2 \cdot s_j} \bmod N^{\zeta+1}$ and return

$$\vec{\mu}_i = (\mu_{i,j_1}, \ldots, \mu_{i,j_{d_i}}).$$

Based on DCR and LWE (4)

- $\mathsf{Combine}\big(\mathcal{B} = (\mathcal{S}, |\mathcal{S}| \geq t, \{\vec{\mu}_i\}_{i\in\mathcal{S}}), \mathsf{ct} = (C_0, C_1, \vec{\pi})\big)$: Letting $\mathcal{S} = \{j_1, \ldots, j_t\}$,
  1. LISS: find a reconstruction vector $\vec{\lambda}_{\mathcal{S}} = [\vec{\lambda}_{j_1}^{\top} \mid \ldots \mid \vec{\lambda}_{j_t}^{\top}]^{\top} \in \{-1, 0, 1\}^{d_{\mathcal{S}}}$.
  2. LISS: compute

$$\hat{\mu} \triangleq \prod_{i\in[t]} \prod_{k\in[d_{j_i}]} \mu_{j_i,k}^{\lambda_{j_i,k}} = C_0^{2x} \bmod N^{\zeta+1}.$$

  3. Compute $\hat{C}_1 = C_1/\hat{\mu} \bmod N^{\zeta+1}$ and return $\perp$ if $\hat{C}_1 \not\equiv 1 \pmod{N}$. Otherwise, return $\mathsf{Msg} = (\hat{C}_1 - 1)/N \in \mathbb{Z}_{N^\zeta}$.

Based on DCR and LWE (4)

- $\mathsf{Combine}\big(\mathcal{B} = (\mathcal{S}, |\mathcal{S}| \geq t, \{\vec{\mu}_i\}_{i\in\mathcal{S}}), \mathsf{ct} = (C_0, C_1, \vec{\pi})\big)$: Letting $\mathcal{S} = \{j_1, \ldots, j_t\}$,
  1. LISS: find a reconstruction vector $\vec{\lambda}_{\mathcal{S}} = [\vec{\lambda}_{j_1}^{\top} \mid \ldots \mid \vec{\lambda}_{j_t}^{\top}]^{\top} \in \{-1, 0, 1\}^{d_{\mathcal{S}}}$.
  2. LISS: compute

$$\hat{\mu} \triangleq \prod_{i\in[t]} \prod_{k\in[d_{j_i}]} \mu_{j_i,k}^{\lambda_{j_i,k}} = C_0^{2x} \bmod N^{\zeta+1}.$$

  3. Compute $\hat{C}_1 = C_1/\hat{\mu} \bmod N^{\zeta+1}$ and return $\perp$ if $\hat{C}_1 \not\equiv 1 \pmod{N}$. Otherwise, return $\mathsf{Msg} = (\hat{C}_1 - 1)/N \in \mathbb{Z}_{N^\zeta}$.

Security

Theorem

The scheme is CCA2 secure under adaptive corruptions, assuming that: (i) DCR holds; (ii) The NIZK argument is one-time simulation-sound.

Security

Theorem

The scheme is CCA2 secure under adaptive corruptions, assuming that: (i) DCR holds; (ii) The NIZK argument is one-time simulation-sound.

- OTSS: Seeing one simulated proof for any statement does not help prove a non-trivial false statement.
- We give a one-time simulation sound Π^{OTSS} for $\mathcal{L}^{\mathsf{DCR}}$ under the DCR and LWE assumption.

  (shorter public parameters; improves an unbounded SS construction [LNPY20])
- Security proof exploits the entropy of secret keys (sampled from a discrete Gaussian) and the properties of a LISS (similarly to `Libert-Stehlé-Titiu; TCC'18`).

Security

Theorem

The scheme is CCA2 secure under adaptive corruptions, assuming that: (i) DCR holds; (ii) The NIZK argument is one-time simulation-sound.

Proof idea.

- DCR allows moving to a game that encrypts using the secret key x
- Message hidden by $x \bmod N^\zeta$
- Conditionally on $\mathcal{A}$'s view, $x \in \mathbb{Z}$ is Gaussian in a shift of $p'q' \cdot \mathbb{Z}$

  $\Rightarrow$ The distribution of $x \bmod N^\zeta$ is statistically close to $U(\mathbb{Z}_{N^\zeta})$.

Construction from LWE: Threshold Dual Regev

- Exploits the entropy of secret $\mathbf{R} \in \mathbb{Z}^{m \times L}$ conditionally on public keys $\mathbf{U} = \mathbf{A} \cdot \mathbf{R} \in \mathbb{Z}_q^{n \times L}$
- Shares each column of $\mathbf{R} \in \mathbb{Z}^{m \times L}$ using a LISS scheme
- Uses noise flooding in partial decryption shares
- Security proof follows idea from distributed PRFs (`Libert-Stehlé-Titiu; TCC'18`)
- Uses a simulation-sound argument that ciphertext components are of the form $(\mathbf{c}_0, \mathbf{c}_1)^\top = \mathbf{B} \cdot \mathbf{s} + \mathbf{e} \bmod q$ (`Libert et al.; Asiacrypt'20`)
- **Open problem:** avoid noise flooding; use a polynomial modulus while keeping compact ciphertexts

Construction from LWE: Threshold Dual Regev

- Exploits the entropy of secret $\mathbf{R} \in \mathbb{Z}^{m \times L}$ conditionally on public keys $\mathbf{U} = \mathbf{A} \cdot \mathbf{R} \in \mathbb{Z}_q^{n \times L}$
- Shares each column of $\mathbf{R} \in \mathbb{Z}^{m \times L}$ using a LISS scheme
- Uses noise flooding in partial decryption shares
- Security proof follows idea from distributed PRFs (`Libert-Stehlé-Titiu; TCC'18`)
- Uses a simulation-sound argument that ciphertext components are of the form $(\mathbf{c}_0, \mathbf{c}_1)^\top = \mathbf{B} \cdot \mathbf{s} + \mathbf{e} \bmod q$ (`Libert et al.; Asiacrypt'20`)
- **Open problem:** avoid noise flooding; use a polynomial modulus while keeping compact ciphertexts

Thank you for your attention!