Simple Threshold Fully Homomorphic Encryption from LWE with Polynomial Modulus

NordiCrypt 7th July 2023

Katharina Boudgoust Peter Scholl

Aarhus University

1

Simple Threshold **Fully Homomorphic Encryption** from LWE with Polynomial Modulus

Fully Homomorphic Encryption (FHE)

FHE scheme:

- $\mathsf{KGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$ λ security parameter
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$ $\mathcal{M}$ message space
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \mathsf{ct}'$ $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}') \to m'$

Properties:

- Correctness $\mathsf{Dec}(\mathsf{Eval}(f, \mathsf{Enc}(m_1), \mathsf{Enc}(m_2))) = f(m_1, m_2)$
- Semantic security IND-CPA

Fully Homomorphic Encryption (FHE)

FHE scheme:

- $\mathsf{KGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$ — λ security parameter
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$ — $\mathcal{M}$ message space
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \mathsf{ct}'$ — $f : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}') \to m'$

Properties:

- Correctness — $\mathrm{Dec}(\mathsf{Eval}(f, \mathsf{Enc}(m_1), \mathsf{Enc}(m_2))) = f(m_1, m_2)$
- Semantic security — IND-CPA

Problem:

- sk single point of failure

2 1

Simple **Threshold** Fully Homomorphic Encryption from LWE with Polynomial Modulus

Threshold FHE

t-out-of-n Threshold FHE scheme:

- $\mathsf{KGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \mathsf{ct}'$
- $\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct}') \to d_i$ $\quad i \in \{1, \ldots, n\}$
- $\mathsf{Combine}(\{d_i\}_{i \in S}) \to m'$ $\quad S \subset \{1, \ldots, n\}$

Threshold FHE

t-out-of-n Threshold FHE scheme:

- $\mathsf{KGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \mathsf{ct}'$
- $\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct}') \to d_i$ — $i \in \{1, \ldots, n\}$
- $\mathsf{Combine}(\{d_i\}_{i \in S}) \to m'$ — $S \subset \{1, \ldots, n\}$

Properties:

- Correctness — for $|S| > t$ recover correct message
- Partial decryption security — for $|S| \leq t$ no information is leaked
- Semantic security — FHE is $\mathrm{IND\text{-}CPA}$

Applications:

- Storing sensitive data — Yash's talk
- Electronic voting protocols
- Multiparty computations — Lennart's talk

3 2 1

Simple Threshold Fully Homomorphic Encryption from LWE with Polynomial Modulus

Simple Threshold FHE, First Trial

linear function in
enc randomness and sk

Ingredients:

1) FHE with β nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$
- $\|e_{\mathsf{ct}}\|_\infty \leq \beta$

Simple Threshold FHE, First Trial

linear function in
enc randomness and sk

Ingredients:

1) FHE with β nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$
- $\|e_{\mathsf{ct}}\|_\infty \leq \beta$

2) t-out-of-n linear secret sharing scheme:

- $\mathsf{Share}(\mathsf{sk}) \rightarrow \mathsf{sk}_1, \ldots, \mathsf{sk}_n$
- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}_{i \in S}) = \langle y, \mathsf{sk} \rangle$ $\qquad |S| > t$

Simple Threshold FHE, First Trial

linear function in
enc randomness and sk

Ingredients:

1) FHE with β nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$
- $\|e_{\mathsf{ct}}\|_\infty \leq \beta$

2) t-out-of-n linear secret sharing scheme:

- $\mathsf{Share}(\mathsf{sk}) \to \mathsf{sk}_1, \ldots, \mathsf{sk}_n$
- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}_{i \in S}) = \langle y, \mathsf{sk} \rangle$ $\quad |S| > t$

1st Trial construction:

- KGen: compute $\mathsf{Share}(\mathsf{sk}) \to (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle$
- Combine: compute $\mathsf{Rec}(\{ d_i \})$

Simple Threshold FHE, First Trial

linear function in
enc randomness and sk

Ingredients:

1) FHE with β nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$
- $\|e_{\mathsf{ct}}\|_\infty \leq \beta$

2) t-out-of-n linear secret sharing scheme:

- $\mathsf{Share}(\mathsf{sk}) \to \mathsf{sk}_1, \ldots, \mathsf{sk}_n$
- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}_{i \in S}) = \langle y, \mathsf{sk} \rangle$ $\qquad |S| > t$

1st Trial construction:

- KGen: compute $\mathsf{Share}(\mathsf{sk}) \to (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle$
- Combine: compute $\mathsf{Rec}(\{ d_i \}) = \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle \}) =^{(2)} \langle \mathsf{ct}, \mathsf{sk} \rangle =^{(1)} f(m_1, m_2) + e_{\mathsf{ct}}$

Problem:

- Corrupted parties learn e_{ct}
- After enough decryptions, can recover sk

Simple Threshold FHE, Second Trial

Ingredients:

1) FHE with nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$ — e_{ct} ciphertext noise

2) t-out-of-n linear secret sharing scheme:

- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}) = \langle y, \mathsf{sk} \rangle$

2nd Trial Construction:

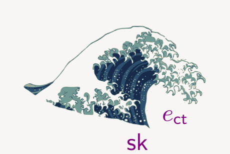
- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ — e_{flood} flooding noise
- Combine: compute $\mathsf{Rec}(\{ d_i \})$

Simple Threshold FHE, Second Trial

Ingredients:

1) FHE with nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$



e_{ct} ciphertext noise

2) t-out-of-n linear secret sharing scheme:

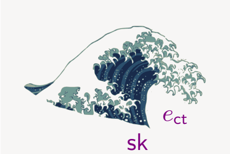
- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}) = \langle y, \mathsf{sk} \rangle$

2nd Trial Construction:

- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$
- Combine: compute $\mathsf{Rec}(\{ d_i \})$

e_{flood} flooding noise

Simple Threshold FHE, Second Trial



Ingredients:

1) FHE with nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$

e_{ct} ciphertext noise

2) t-out-of-n linear secret sharing scheme:

- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}) = \langle y, \mathsf{sk} \rangle$

2nd Trial Construction:

- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$

e_{flood} flooding noise

- Combine: compute $\mathsf{Rec}(\{ d_i \})$
  $= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}) = f(m_1, m_2) + e_{\mathsf{ct}} + \mathsf{Rec}(\{ e_{flood,i} \})$

Problem:

- Is $\mathsf{Rec}(\{ e_{flood,i} \})$ still small? Needed for correctness!
- For Shamir secret sharing: no

Simple Threshold FHE, Second Trial

Ingredients:

1) FHE with nearly linear decryption:

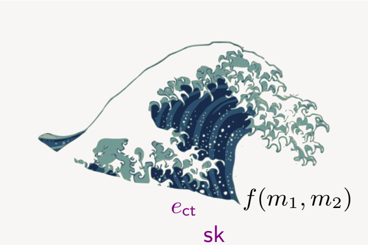
- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$ $\qquad$ e_{ct} ciphertext noise

2) t-out-of-n linear secret sharing scheme:

- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}) = \langle y, \mathsf{sk} \rangle$

2nd Trial Construction:

- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle$ $\qquad$ flooding noise
- Combine: compute $\mathsf{Rec}(\{ d_i \})$
  $= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}) =$



Problem:

- Is $\mathsf{Rec}(\{ e_{flood,i} \})$ still small? Needed for correctness!
- For Shamir secret sharing: no

Simple Threshold FHE, Final [BD10]

Ingredients:

1) FHE with β nearly linear decryption:
   - $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$ — e_{ct} ciphertext noise

2) t-out-of-n linear secret sharing scheme with small reconstruction:
   - $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}) = \langle y, \mathsf{sk} \rangle$
   - if e_i small, then $\mathsf{Rec}(\{ e_i \})$ also small

Construction:

- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ — e_{flood} flooding noise
- Combine: compute $\mathsf{Rec}(\{ d_i \})$

Simple Threshold FHE, Final [BD10]

Ingredients:

1) FHE with β nearly linear decryption:

- $\langle \mathsf{ct}, \mathsf{sk} \rangle = f(m_1, m_2) + e_{\mathsf{ct}}$ $\qquad$ e_{ct} ciphertext noise

2) t-out-of-n linear secret sharing scheme with small reconstruction:

- $\mathsf{Rec}(\{ \langle y, \mathsf{sk}_i \rangle \}) = \langle y, \mathsf{sk} \rangle$
- if e_i small, then $\mathsf{Rec}(\{ e_i \})$ also small

Construction:

- PartDec: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ $\qquad$ e_{flood} flooding noise
- Combine: compute $\mathsf{Rec}(\{ d_i \})$
  $= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}) = f(m_1, m_2) + e_{\mathsf{ct}} + \mathsf{Rec}(\{ e_{flood,i} \})$

Partial Decryption Security

Two worlds:

- Real: e_{ct} and e_{flood}
- Simulated: only e_{flood}

How close are they? [BD10] measures with statistical distance Δ

$$\Delta(\mathsf{Real}, \mathsf{Sim}) \leq \Delta(e_{flood} + e_{\mathsf{ct}}, e_{flood}) \leq \mathsf{negl}(\lambda)$$

Partial Decryption Security

Two worlds:

- Real: e_{ct} and e_{flood}
- Simulated: only e_{flood}

How close are they? [BD10] measures with statistical distance Δ

$$\Delta(\mathsf{Real}, \mathsf{Sim}) \leq \Delta(e_{flood} + e_{\mathsf{ct}}, e_{flood}) \leq \mathsf{negl}(\lambda)$$

Problem:

- $\|e_{flood}\|$ needs to be super-polynomially larger than $\|e_{\mathsf{ct}}\|$
- LWE-based constructions: $\|e_{flood}\| \sim$ LWE modulus q and $\|e_{\mathsf{ct}}\| \sim$ LWE noise $\mathbf{e}$, thus super-polynomial modulus-noise ratio
  - Larger parameters
  - Easier problem

$$\mathbf{A}\ ,\ \mathbf{A}\,\mathbf{s} + \mathbf{e} \mod q$$

Partial Decryption Security

Two worlds:

- Real: e_{ct} and e_{flood}
- Simulated: only e_{flood}

Idea: change the measure! [BLR$^+$18]

How close are they? [BD10] measures with statistical distance Δ

$$\Delta(\mathsf{Real}, \mathsf{Sim}) \leq \Delta(e_{flood} + e_{\mathsf{ct}}, e_{flood}) \leq \mathsf{negl}(\lambda)$$

Problem:

- $\|e_{flood}\|$ needs to be super-polynomially larger than $\|e_{\mathsf{ct}}\|$
- LWE-based constructions: $\|e_{flood}\| \sim$ LWE modulus q and $\|e_{\mathsf{ct}}\| \sim$ LWE noise $\mathbf{e}$, thus super-polynomial modulus-noise ratio
  - Larger parameters
  - Easier problem

$$\mathbf{A}, \; \mathbf{A}\,\mathbf{s} + \mathbf{e} \quad \bmod q$$

3 2 1

Simple Threshold Fully Homomorphic Encryption **from LWE with Polynomial Modulus**

4

Improved Noise Flooding via Rényi Divergence 1/2

Let P, Q be discrete probability distributions

In [BD10]: Statistical Distance $\Delta(P,Q) = \frac{1}{2}\sum_{x \in \mathrm{Supp}(P)} |P(x) - Q(x)|$

In our work: Rényi Divergence

$$\mathsf{RD}(P,Q) = \sum_{\substack{x \in \mathrm{Supp}(P) \\ \subset \mathrm{Supp}(Q)}} \frac{P(x)^2}{Q(x)}$$

Improved Noise Flooding via Rényi Divergence 1/2

Let P, Q be discrete probability distributions

In [BD10]: Statistical Distance $\Delta(P,Q) = \frac{1}{2}\sum_{x\in\mathrm{Supp}(P)}|P(x)-Q(x)|$

In our work: Rényi Divergence

$$\mathrm{RD}(P,Q) = \sum_{\substack{x\in\mathrm{Supp}(P)\\ \subset\,\mathrm{Supp}(Q)}} \frac{P(x)^2}{Q(x)}$$

Both fulfill the **probability preservation property** for an event E:

[BD10]:	$P(E)$	$\leq$	$\Delta(P,Q) + Q(E)$	(additive)
Our work:	$P(E)^2$	$\leq$	$\mathrm{RD}(P,Q)\cdot Q(E)$	(multiplicative)

- $Q(E)$ negligible $\Rightarrow$ $P(E)$ negligible
- $\Delta(P,Q) \overset{!}{=}$ negligible and $\mathrm{RD}(P,Q) \overset{!}{=}$ **constant**

Improved Noise Flooding via Rényi Divergence 2/2

Two worlds:

- Real: e_{ct} and e_{flood}
- Simulated: only e_{flood}

How close are they?

$$\Delta(\mathsf{Real}, \mathsf{Sim}) \leq \Delta(e_{flood} + e_{\mathsf{ct}}, e_{flood}) \leq \mathsf{negl}(\lambda)$$

$$\mathsf{RD}(\mathsf{Real}, \mathsf{Sim}) \leq \mathsf{RD}(e_{flood} + e_{\mathsf{ct}}, e_{flood}) \leq \mathsf{constant}$$

Advantage:

- $\|e_{flood}\|$ only needs to be polynomially larger than $\|e_{\mathsf{ct}}\|$
- LWE-based constructions: polynomial modulus-noise ratio

Improved Noise Flooding via Rényi Divergence 2/2

Two worlds:

- Real: e_{ct} and e_{flood}
- Simulated: only e_{flood}

How close are they?

$$\Delta(\mathsf{Real}, \mathsf{Sim}) \leq \Delta(e_{flood} + e_{\mathsf{ct}}, e_{flood}) \leq \mathsf{negl}(\lambda)$$

$$\mathsf{RD}(\mathsf{Real}, \mathsf{Sim}) \leq \mathsf{RD}(e_{flood} + e_{\mathsf{ct}}, e_{flood}) \leq \text{constant}$$

Advantage:

- $\|e_{flood}\|$ only needs to be polynomially larger than $\|e_{\mathsf{ct}}\|$
- LWE-based constructions: polynomial modulus-noise ratio

Disadvantage:

1) Rényi divergence depends on the number of issued partial decryptions

   $\rightarrow$ from simulation-based to game-based security notion

2) Works well with search problems, not so well with decision problems

3 2 1

Simple Threshold Fully Homomorphic Encryption
from LWE with Polynomial Modulus

4

5 Challenges on the way

Our Approaches

1) Rényi divergence depends on the number of issued partial decryptions
   - $\rightarrow$ $\mathrm{IND\text{-}CPA}$ security notion for Threshold FHE [JRS17]
   - $\rightarrow$ a priori bound on partial decryption queries

2) Works well with search problems, not so well with decision problems
   - $\rightarrow$ one-way security notion for Threshold FHE
   - $\rightarrow$ lift one-way security to $\mathrm{IND\text{-}CPA}$ security for Threshold FHE
     use Goldreich-Levin hardcore bits

Our Approaches

1) Rényi divergence depends on the number of issued partial decryptions
   $\rightarrow$ $\mathrm{IND\text{-}CPA}$ security notion for Threshold FHE [JRS17]
   $\rightarrow$ a priori bound on partial decryption queries

2) Works well with search problems, not so well with decision problems
   $\rightarrow$ one-way security notion for Threshold FHE
   $\rightarrow$ lift one-way security to $\mathrm{IND\text{-}CPA}$ security for Threshold FHE
   use Goldreich-Levin hardcore bits

Game-Based Security for t-out-of-n Threshold FHE

Challenger $\mathcal{C}$		Adversary $\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk})$		pk
	$\xleftarrow{\quad m_0, m_1 \quad}$	
$b \leftarrow \{0,1\}$ $\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$		
	$\xrightarrow{\quad \mathsf{ct}_b \quad}$	
	$\xleftarrow{\quad b' \quad}$	
win if $b = b'$		

Game-Based Security for t-out-of-n Threshold FHE

Challenger $\mathcal{C}$		Adversary $\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$		pk
	$\xleftarrow{\quad S \subset \{1, \ldots, n\},\ \lvert S \rvert \leq t \quad}$	
static corruptions	$\xrightarrow{\quad (\mathsf{sk}_i)_{i \in S} \quad}$	
	$\xleftarrow{\quad m_0, m_1 \quad}$	
$b \leftarrow \{0,1\}$ $\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$		
	$\xrightarrow{\quad \mathsf{ct}_b \quad}$	PartDec Oracle
	$\xleftarrow{\quad b' \quad}$	
win if $b = b'$		

Game-Based Security for t-out-of-n Threshold FHE

Challenger $\mathcal{C}$		Adversary $\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$		pk
static corruptions	$\xleftarrow{S \subset \{1, \ldots, n\},\ \lvert S \rvert \leq t}$	
	$\xrightarrow{(\mathsf{sk}_i)_{i \in S}}$	
	$\xleftarrow{m_0, m_1}$	
$b \leftarrow \{0, 1\}$ $\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$		
	$\xrightarrow{\mathsf{ct}_b}$	?
	$\xleftarrow{b'}$	PartDec Oracle
win if $b = b'$		

Game-Based Security for t-out-of-n Threshold FHE

Challenger $\mathcal{C}$		Adversary $\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$		pk
	$\xleftarrow{S \subset \{1, \ldots, n\},\ \lvert S \rvert \leq t}$	
Version 1	$\xrightarrow{(\mathsf{sk}_i)_{i \in S}}$	
	$\xleftarrow{m_0, m_1}$	
$b \leftarrow \{0,1\}$ $\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$		
	$\xrightarrow{\mathsf{ct}_b}$	
	$\xleftarrow{b'}$	
win if $b = b'$		

input: f, m_1, m_2

PartDec Oracle

$\mathsf{ct}_1 \leftarrow \mathsf{Enc}(m_1)$

$\mathsf{ct}_2 \leftarrow \mathsf{Enc}(m_2)$

$\mathsf{ct} \leftarrow \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2)$

$d_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$

output: $d_1, \ldots, d_n$

Game-Based Security for t-out-of-n Threshold FHE

Challenger $\mathcal{C}$ | Adversary $\mathcal{A}$

$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ | pk

$\xleftarrow{S \subset \{1, \ldots, n\},\ |S| \leq t}$

Version 1

$\xrightarrow{(\mathsf{sk}_i)_{i \in S}}$

$\xleftarrow{m_0, m_1}$

$b \leftarrow \{0, 1\}$

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$

$\xrightarrow{\mathsf{ct}_b}$

$\xleftarrow{b'}$

win if $b = b'$

input: f, m_1, m_2

$\mathsf{ct}_1 \leftarrow \mathsf{Enc}(m_1)$

$\mathsf{ct}_2 \leftarrow \mathsf{Enc}(m_2)$

$\mathsf{ct} \leftarrow \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2)$

$d_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$

PartDec Oracle

output: $d_1, \ldots, d_n$

Problem: queries to PartDec oracle are independent of ct_b

ℓ-IND-CPA Security for t-out-of-n Threshold FHE [JRS17]

Challenger $\mathcal{C}$		Adversary $\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$		pk
	$\xleftarrow{S \subset \{1, \ldots, n\},\ \lvert S \rvert \leq t}$	
Version 2	$\xrightarrow{(\mathsf{sk}_i)_{i \in S}}$	
	$\xleftarrow{m_0, m_1}$	
$b \leftarrow \{0,1\}$ $\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$		
	$\xrightarrow{\mathsf{ct}_b}$	? PartDec Oracle
	$\xleftarrow{b'}$	
win if $b = b'$		

ℓ-IND-CPA Security for t-out-of-n Threshold FHE [JRS17]

Challenger $\mathcal{C}$		Adversary $\mathcal{A}$
$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$		pk
	$\xleftarrow{S \subset \{1, \ldots, n\},\ \lvert S \rvert \leq t}$	
Version 2		Enc_b Oracle
$b \leftarrow \{0, 1\}$		
	$\xleftarrow{b'}$	PartDec Oracle
win if $b = b'$		

ℓ-IND-CPA Security for t-out-of-n Threshold FHE [JRS17]

Challenger $\mathcal{C}$

$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$

Adversary $\mathcal{A}$

pk

$\xleftarrow{S \subset \{1, \ldots, n\},\ |S| \leq t}$

Version 2

input: m_0, m_1

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(m_b)$

Enc_b Oracle

output: ct_b

$b \leftarrow \{0, 1\}$

PartDec Oracle

$\xleftarrow{b'}$

win if $b = b'$

ℓ-IND-CPA Security for t-out-of-n Threshold FHE [JRS17]

Challenger $\mathcal{C}$

$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$

Version 2

$b \leftarrow \{0,1\}$

win if $b = b'$

Adversary $\mathcal{A}$

pk

$\xleftarrow{S \subset \{1, \ldots, n\},\ |S| \leq t}$

input: m_0, m_1

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(m_b)$

Enc$_b$ Oracle

output: ct_b

input: $f, \mathsf{ct}_b, \mathsf{ct}'_b$

$\mathsf{ct} \leftarrow \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2)$

$d_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$

if $f(m_0, m'_0) \neq f(m_1, m'_1)$

abort

PartDec Oracle

output: $d_1, \ldots, d_n$

$\xleftarrow{b'}$

ℓ-IND-CPA Security for t-out-of-n Threshold FHE [JRS17]

Challenger $\mathcal{C}$

$(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$

Version 2

$\sim$ input-indistinguishability for MPC

$b \leftarrow \{0,1\}$

win if $b = b'$

Adversary $\mathcal{A}$

pk

$\xleftarrow{S \subset \{1, \ldots, n\},\ |S| \leq t}$

input: m_0, m_1

Enc$_b$ Oracle

$\mathsf{ct}_b \leftarrow \mathsf{Enc}(m_b)$

output: ct_b

input: $f, \mathsf{ct}_b, \mathsf{ct}'_b$

PartDec Oracle

$\mathsf{ct} \leftarrow \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2)$

$d_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$

if $f(m_0, m'_0) \neq f(m_1, m'_1)$

abort

output: $d_1, \ldots, d_n$

at most ℓ queries

$\xleftarrow{b'}$

Our Approaches

1) Rényi divergence depends on the number of issued partial decryptions
   - $\rightarrow$ use a $\mathrm{IND\text{-}CPA}$ security notion for Threshold FHE [JRS17]
   - $\rightarrow$ add a priori bound on partial decryption queries

2) Works well with search problems, not so well with decision problems
   - $\rightarrow$ define a one-way security notion for Threshold FHE
   - $\rightarrow$ show how to lift one-way security to $\mathrm{IND\text{-}CPA}$ security for Threshold FHE
     use Goldreich-Levin hardcore bits

next time :)

Conclusion

Wrapping Up

My intuition on Rényi divergence:

- In search problems always beneficial
- In decision problems depends on your setting

Related Works:

- [CSS+22] Use Rényi divergence directly for $\mathrm{IND\text{-}CPA}$, but much weaker model
- [LMSS22] Use Rényi differential privacy for $\mathrm{IND\text{-}CPA}$, but much worse parameters

Open Problems:

- Adaptive corruptions $\rightarrow$ Master project Michael & Magdalena
- Alternative noise flooding?

Wrapping Up

My intuition on Rényi divergence:

- In search problems always beneficial
- In decision problems depends on your setting

Related Works:

- [CSS$^+$22] Use Rényi divergence directly for $\mathrm{IND\text{-}CPA}$, but much weaker model
- [LMSS22] Use Rényi differential privacy for $\mathrm{IND\text{-}CPA}$, but much worse parameters

Open Problems:

- Adaptive corruptions $\rightarrow$ Master project Michael & Magdalena
- Alternative noise flooding?

Thank you.

Rikke Bendlin and Ivan Damgård.
Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems.
In *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 201–218. Springer, 2010.

Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld.
Improved security proofs in lattice-based cryptography: Using the rényi divergence rather than the statistical distance.
J. Cryptol., 31(2):610–640, 2018.

Siddhartha Chowdhury, Sayani Sinha, Animesh Singh, Shubham Mishra, Chandan Chaudhary, Sikhar Patranabis, Pratyay Mukherjee, Ayantika Chatterjee, and Debdeep Mukhopadhyay.
Efficient threshold FHE with application to real-time systems.
IACR Cryptol. ePrint Arch., page 1625, 2022.

Aayush Jain, Peter M. R. Rasmussen, and Amit Sahai.
Threshold fully homomorphic encryption.
IACR Cryptol. ePrint Arch., page 257, 2017.

Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell.
Securing approximate homomorphic encryption using differential privacy.
In *CRYPTO (1)*, volume 13507 of *Lecture Notes in Computer Science*, pages 560–589. Springer, 2022.