Threshold Fully Homomorphic Encryption from LWE

Challenges and Perspectives

Katharina Boudgoust

CNRS, Univ Montpellier, LIRMM, France

Context

☞ In asymmetric cryptography there is a public key and a secret key. The secret key is used for a **critical operation** and thus needs to be protected.

- 🔒 Encryption: secret key allows to decrypt ciphertexts
- ✎ Signature: secret key allows to sign messages

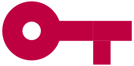
Context

☞ In asymmetric cryptography there is a public key and a secret key. The secret key is used for a **critical operation** and thus needs to be protected.

- Encryption: secret key allows to decrypt ciphertexts
- Signature: secret key allows to sign messages

☞ The secret key can be seen as a **single point of failure**.

- Someone else learns it: security issue
- I loose it: operability issue



Home › Cryptocurrency › Youtuber Loses $60,000 In Crypto and NFTs After Exposing His Private Key...

Cryptocurrency News

Youtuber Loses $60,000 In Crypto and NFTs After Exposing His Private Key While Live Streaming

By **Newton Gitonga** - September 2, 2023

DARRYN POLLOCK NOV 30, 2017

Infamous Discarded Hard Drive Holding 7,500 Bitcoins Would be Worth $80 Million Today

Cryptonews • Altcoin News • LHV Bank Founder Has Lost Private Key to ETH Stash Worth $470 Million

LHV Bank Founder Has Lost Private Key to ETH Stash Worth $470 Million

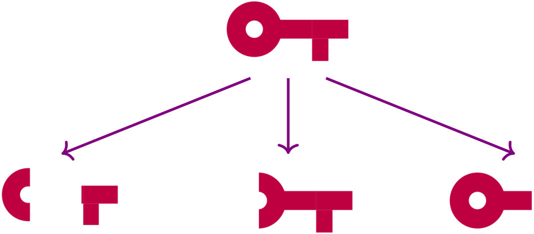
Ruholamin Haqshanas

Last updated: **November 7, 2023 02:36 EST** | 2 min read

Motivation Threshold Cryptography [DF89]

The secret key can be seen as a **single point of failure**.

Idea: divide the secret key into multiple shares



- Better security: multiple secret key shares needed
- Better operability: not necessarily all secret key shares needed

Today: Threshold Fully Homomorphic Encryption

FHE scheme:

- $\mathsf{KGen} \to (\mathsf{pk}, \mathsf{sk})$
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$ $m \in \{0,1\}$
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \widehat{\mathsf{ct}}$ $f : \{0,1\} \times \{0,1\} \to \{0,1\}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to m$

Today: Threshold Fully Homomorphic Encryption

FHE scheme:

- $\mathsf{KGen} \to (\mathsf{pk}, \mathsf{sk})$
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$ $\qquad m \in \{0,1\}$
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \widehat{\mathsf{ct}}$ $\qquad f : \{0,1\} \times \{0,1\} \to \{0,1\}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to m$

Today: Threshold Fully Homomorphic Encryption

t-out-of-n Threshold FHE scheme:

- $\mathsf{KGen} \to (\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$ $\qquad m \in \{0,1\}$
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \widehat{\mathsf{ct}}$ $\qquad f : \{0,1\} \times \{0,1\} \to \{0,1\}$
- $\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct}) \to d_i$
- $\mathsf{Combine}(\{\, d_i \,\}_{i \in S}) \to m$ $\qquad S \subseteq \{\, 1, \ldots, n \,\}$

Today: Threshold Fully Homomorphic Encryption

t-out-of-n Threshold FHE scheme:

- KGen $\rightarrow (\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- $\mathsf{Enc}(\mathsf{pk}, m) \rightarrow \mathsf{ct}$ $m \in \{0, 1\}$
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \rightarrow \widehat{\mathsf{ct}}$ $f : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$
- $\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct}) \rightarrow d_i$
- $\mathsf{Combine}(\{ d_i \}_{i \in S}) \rightarrow m$ $S \subseteq \{ 1, \ldots, n \}$

Properties:

- Correctness t parties can recover the message
- Security less than t parties learn nothing about message

Applications:

- Electronic voting protocols
- Universal thresholdizer [BGG+18]

Overview of Today's Talk

Structure:

- Part 1: *Basic Blueprint of Threshold FHE*
- Part 2: *Suitable Secret Sharings*
- Part 3: *Different Noise Floodings*
- Part 4: *Defining Security*

Overview of Today's Talk

Structure:

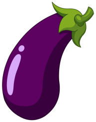
- Part 1: *Basic Blueprint of Threshold FHE*
- Part 2: *Suitable Secret Sharings*
- Part 3: *Different Noise Floodings*
- Part 4: *Defining Security*

This talk: overview
Christian's talk: details

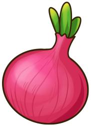
Part 1:

Basic Blueprint

Ingredients for Threshold FHE based on LWE



FHE with nearly linear decryption



Linear secret sharing

FHE from LWE with nearly linear decryption

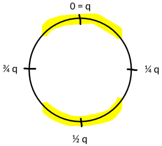
FHE scheme:

- KGen $\rightarrow$ (pk, sk)
- Enc(pk, m) $\rightarrow$ ct $\bmod q$
- Eval(pk, f, ct_1, ct_2) $\rightarrow \widehat{\mathsf{ct}}$
- Dec(sk, ct) $\rightarrow m$

q modulus

Nearly linear decryption:

- sk and ct vectors over $\mathbb{Z}_q$
- $\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q = \frac{q}{2} \cdot f(m_1, m_2) + e_{\mathsf{ct}}$
- e_{ct} encryption noise
- $\|e_{\mathsf{ct}}\|_\infty < q/4$



FHE from LWE with nearly linear decryption

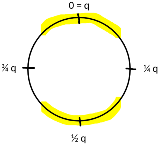
FHE scheme:

- $\mathsf{KGen} \to (\mathsf{pk}, \mathsf{sk})$
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct} \bmod q$
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \mathsf{ct}_2) \to \widehat{\mathsf{ct}}$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to m$

q modulus

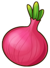
Nearly linear decryption:

- sk and ct vectors over $\mathbb{Z}_q$
- $\langle \mathsf{ct}, \mathsf{sk} \rangle \bmod q = \frac{q}{2} \cdot f(m_1, m_2) + e_{\mathsf{ct}}$
- e_{ct} encryption noise
- $\|e_{\mathsf{ct}}\|_\infty < q/4$



⚠ Damien's talk: decryption failure should be small enough!

Linear secret sharing



t-out-of-n secret sharing:

- $\mathsf{Share}(\mathsf{sk}) \to (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- $\mathsf{Rec}(\{\, \mathsf{sk}_i \,\}_{i \in S}) \to \mathsf{sk}$ $\quad S \subseteq \{\, 1, \ldots, n \,\}$

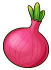
Properties:

- if $|S| < t$ no information about sk leaked
- if $|S| \geq t$ successful reconstruction of sk

Linearity:

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

Blueprint for Threshold FHE, Trial



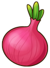
linear secret sharing

$\approx$ linear decrypt

t-out-of-n Threshold FHE scheme:

- $\mathsf{KGen} \to (\mathsf{pk}, \mathsf{sk})$ and $\mathsf{Share}(\mathsf{sk}) \to (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle$
- Combine : comute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

Blueprint for Threshold FHE, Trial



linear secret sharing

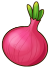
$\approx$ linear decrypt

t-out-of-n Threshold FHE scheme:

- $\mathsf{KGen} \rightarrow (\mathsf{pk}, \mathsf{sk})$ and $\mathsf{Share}(\mathsf{sk}) \rightarrow (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- $\mathsf{PartDec}$: compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle$
- $\mathsf{Combine}$: comute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

$$\mathsf{Rec}(\{ d_i \}_{i \in S}) = \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle \}_i) = \langle \mathsf{ct}, \mathsf{sk} \rangle = \frac{q}{2} \cdot f(m_1, m_2) + \ e_{\mathsf{ct}}$$

Blueprint for Threshold FHE, Trial


linear secret sharing


≈ linear decrypt

t-out-of-n Threshold FHE scheme:

- KGen $\rightarrow$ $(\mathsf{pk}, \mathsf{sk})$ and Share$(\mathsf{sk}) \rightarrow (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle$
- Combine : comute Rec$(\{\, d_i \,\}_{i \in S})$

$$\mathsf{Rec}(\{\, d_i \,\}_{i \in S}) = \mathsf{Rec}(\{\, \langle \mathsf{ct}, \mathsf{sk}_i \rangle \,\}_i) = \langle \mathsf{ct}, \mathsf{sk} \rangle = \frac{q}{2} \cdot f(m_1, m_2) + \boxed{e_{\mathsf{ct}}}$$

⚠ Problem:

- Ciphertext noise e_{ct} depends on sk
- After "enough" partial decryptions, recover sk

Blueprint for Threshold FHE [BD10]

t-out-of-n Threshold FHE scheme:

- KGen $\rightarrow$ (pk, sk) and Share(sk) $\rightarrow$ $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ flooding noise
- Combine : compute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

Blueprint for Threshold FHE [BD10]

t-out-of-n Threshold FHE scheme:

- KGen $\rightarrow (\mathsf{pk}, \mathsf{sk})$ and Share(sk) $\rightarrow (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ flooding noise
- Combine : compute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

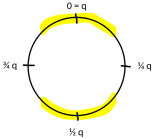
$$\begin{aligned}
\mathsf{Rec}(\{ d_i \}_{i \in S}) &= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}_i) \\
&= \langle \mathsf{ct}, \mathsf{sk} \rangle + \mathsf{Rec}(\{ e_{flood,i} \}_i) \\
&= \frac{q}{2} \cdot f(m_1, m_2) + \ e_{\mathsf{ct}} \ + \ \mathsf{Rec}(\{ e_{flood,i} \}_i)
\end{aligned}$$

Blueprint for Threshold FHE [BD10]

t-out-of-n Threshold FHE scheme:

- $\mathsf{KGen} \to (\mathsf{pk}, \mathsf{sk})$ and $\mathsf{Share}(\mathsf{sk}) \to (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ flooding noise
- Combine : compute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

$$\begin{aligned} \mathsf{Rec}(\{ d_i \}_{i \in S}) &= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}_i) \\ &= \langle \mathsf{ct}, \mathsf{sk} \rangle + \mathsf{Rec}(\{ e_{flood,i} \}_i) \\ &= \frac{q}{2} \cdot f(m_1, m_2) + \underbrace{e_{\mathsf{ct}} + \mathsf{Rec}(\{ e_{flood,i} \}_i)}_{\text{small!}} \end{aligned}$$



Research Directions

- Part 1: Different approach than noise flooding?

Part 2:

Suitable Secret Sharings

Recall: Blueprint for Threshold FHE [BD10]

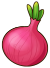
t-out-of-n Threshold FHE scheme:

- KGen → $(\mathsf{pk}, \mathsf{sk})$ and Share(sk) → $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ flooding noise
- Combine : compute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

$$\begin{aligned} \mathsf{Rec}(\{ d_i \}_{i \in S}) &= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}_i) \\ &= \langle \mathsf{ct}, \mathsf{sk} \rangle + \mathsf{Rec}(\{ e_{flood,i} \}_i) \\ &= \frac{q}{2} \cdot f(m_1, m_2) + \underbrace{e_{\mathsf{ct}} + \mathsf{Rec}(\{ e_{flood,i} \}_i)}_{\text{small!}} \end{aligned}$$

Shamir's Secret Sharing over $\mathbb{Z}_q$, Trial

n parties
t threshold



Share(sk):

- sample random polynomial $f(X)$ of degree $< t$ such that $f(0) = \mathsf{sk}$
- output $\mathsf{sk}_i = f(i)$ for $i = 1, \ldots, n$

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

- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{ i \}} \frac{k}{k-i}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

Shamir's Secret Sharing over $\mathbb{Z}_q$, Trial

n parties
t threshold

Share(sk):

- sample random polynomial $f(X)$ of degree $< t$ such that $f(0) = \mathsf{sk}$
- output $\mathsf{sk}_i = f(i)$ for $i = 1, \ldots, n$

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

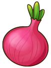
- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{ i \}} \frac{k}{k-i}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

Plug into Threshold FHE:

- PartDec: $d_i = \langle \mathsf{sk}_i, \mathsf{ct} \rangle + e_{flood,i}$
- Combine: $\sum_{i \in S} \lambda_i d_i = \langle \mathsf{sk}, \mathsf{ct} \rangle + \sum_{i \in S} \lambda_i e_{flood,i}$

Shamir's Secret Sharing over $\mathbb{Z}_q$, Trial

n parties
t threshold



Share(sk):

- sample random polynomial $f(X)$ of degree $< t$ such that $f(0) = \mathsf{sk}$
- output $\mathsf{sk}_i = f(i)$ for $i = 1, \ldots, n$

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

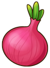
- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{ i \}} \frac{k}{k-i} \in \mathbb{Q}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

Plug into Threshold FHE:

- PartDec: $d_i = \langle \mathsf{sk}_i, \mathsf{ct} \rangle + e_{flood,i}$
- Combine: $\sum_{i \in S} \lambda_i d_i = \langle \mathsf{sk}, \mathsf{ct} \rangle + \sum_{i \in S} \lambda_i e_{flood,i}$

⚠ Problem: Lagrange coefficient λ_i are rationals, not integers

Shamir's Secret Sharing over $\mathbb{Z}_q$, Approaches



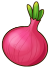
$\mathsf{Rec}(\{\, \mathsf{sk}_i \,\}_{i \in S})$:

- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{\, i \,\}} \frac{k}{k-i}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

Plug in Threshold FHE:

- PartDec: $d_i = \langle \mathsf{sk}_i, \mathsf{ct}' \rangle + e_{flood,i}$
- Combine: $\sum_{i \in S} \lambda_i d_i = \langle \mathsf{sk}, \mathsf{ct}' \rangle + \sum_{i \in S} \lambda_i e_{flood,i}$

Shamir's Secret Sharing over $\mathbb{Z}_q$, Approaches



$\mathsf{Rec}(\{\,\mathsf{sk}_i\,\}_{i \in S})$:

- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{\,i\,\}} \frac{k}{k-i}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

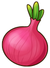
Plug in Threshold FHE:

- PartDec: $d_i = \lambda_i \langle \mathsf{sk}_i, \mathsf{ct}' \rangle + e_{flood,i}$ λ_i depends on S
- Combine: $\sum_{i \in S} \cancel{\lambda_i} d_i = \langle \mathsf{sk}, \mathsf{ct}' \rangle + \sum_{i \in S} \cancel{\lambda_i} e_{flood,i}$

Approaches:

- Move λ_i to PartDec [GKS23, MBH23] ⚠ different model

Shamir's Secret Sharing over $\mathbb{Z}_q$, Approaches



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

- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{ i \}} \frac{k}{k-i} \in \mathbb{Q}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

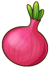
Plug in Threshold FHE:

- PartDec: $d_i = \langle \mathsf{sk}_i, \mathsf{ct}' \rangle + n! \cdot e_{flood,i}$
- Combine: $\sum_{i \in S} \lambda_i d_i = \langle \mathsf{sk}, \mathsf{ct}' \rangle + \sum_{i \in S} (\lambda_i \cdot n!) e_{flood,i}$ $\in \mathbb{Z}$

Approaches:

- Move λ_i to PartDec [GKS23, MBH23] ⚠ different model
- Clearing out denominators, multiply by $n!$ [Sho00, BGG^{+}18] ⚠ $\log q > n \log n$

Shamir's Secret Sharing over $\mathbb{Z}_q$, Approaches



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

- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{ i \}} \frac{k}{k-i} \in \mathbb{Q}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

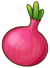
Plug in Threshold FHE:

- PartDec: $d_i = \langle \mathsf{sk}_i, \mathsf{ct}' \rangle + n! \cdot e_{flood,i}$
- Combine: $\sum_{i \in S} \lambda_i d_i = \langle \mathsf{sk}, \mathsf{ct}' \rangle + \sum_{i \in S} (\lambda_i \cdot n!) e_{flood,i}$
  $\in \mathbb{Z}$

Approaches:

- Move λ_i to PartDec [GKS23, MBH23] ⚠ different model
- Clearing out denominators, multiply by $n!$ [Sho00, BGG$^+$18] ⚠ $\log q > n \log n$
- Recursive 2-out-of-3 Shamir secret sharing [CCK23] many shares per party

Shamir's Secret Sharing over $\mathbb{Z}_q$, Approaches



$\mathsf{Rec}(\{\,\mathsf{sk}_i\,\}_{i \in S})$:

- compute Lagrange coefficients $\lambda_i = \prod_{k \in S \setminus \{\,i\,\}} \frac{k}{k-i} \in \mathbb{Q}$
- output $\sum_{i \in S} \lambda_i \mathsf{sk}_i$

Plug in Threshold FHE:

- PartDec: $d_i = \langle \mathsf{sk}_i, \mathsf{ct}' \rangle + n! \cdot e_{flood,i}$
- Combine: $\sum_{i \in S} \lambda_i d_i = \langle \mathsf{sk}, \mathsf{ct}' \rangle + \sum_{i \in S} (\lambda_i \cdot n!) e_{flood,i}$

  $\in \mathbb{Z}$

Approaches:

- Move λ_i to PartDec [GKS23, MBH23] ⚠ different model
- Clearing out denominators, multiply by $n!$ [Sho00, BGG$^+$18] ⚠ $\log q > n \log n$
- Recursive 2-out-of-3 Shamir secret sharing [CCK23] many shares per party

✖ Bit-decomposition of λ_i insecure!

Alternative Approaches for Linear Secret Sharing

- $\{0,1\}$-LSSS [BGG$^+$18] many shares per party
  - from Monotone Boolean formulas
  - Naive secret sharing
  - Replicated secret sharing

- Pseudorandom secret sharing of bounded values over $\mathbb{Z}$ [BD10] requires setup

Research Directions

- Part 1: Different approach than adding noise?
- Part 2: Different approach for linear secret sharing?

Part 3:

Different Noise Floodings

Recall: Blueprint for Threshold FHE [BD10]

t-out-of-n Threshold FHE scheme:

- KGen $\rightarrow$ (pk, sk) and Share(sk) $\rightarrow$ $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ flooding noise
- Combine : compute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

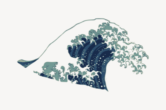
$$\begin{aligned}
\mathsf{Rec}(\{ d_i \}_{i \in S}) &= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}_i) \\
&= \langle \mathsf{ct}, \mathsf{sk} \rangle + \mathsf{Rec}(\{ e_{flood,i} \}_i) \\
&= \frac{q}{2} \cdot f(m_1, m_2) + \underbrace{e_{\mathsf{ct}} + \mathsf{Rec}(\{ e_{flood,i} \}_i)}_{\text{small}}
\end{aligned}$$

Recall: Blueprint for Threshold FHE [BD10]

t-out-of-n Threshold FHE scheme:

- KGen → (pk, sk) and Share(sk) → $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$
- Enc and Eval unchanged
- PartDec : compute $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i}$ flooding noise
- Combine : compute $\mathsf{Rec}(\{ d_i \}_{i \in S})$

$$\begin{aligned} \mathsf{Rec}(\{ d_i \}_{i \in S}) &= \mathsf{Rec}(\{ \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_{flood,i} \}_i) \\ &= \langle \mathsf{ct}, \mathsf{sk} \rangle + \mathsf{Rec}(\{ e_{flood,i} \}_i) \\ &= \frac{q}{2} \cdot f(m_1, m_2) + \underbrace{e_{\mathsf{ct}} + \mathsf{Rec}(\{ e_{flood,i} \}_i)}_{\text{small and no leakage on sk}} \end{aligned}$$



Partial Decryption Security

Two worlds:

- Real: e_{ct} and $e_{flood} := \mathsf{Rec}(\{\, e_{flood,i} \,\}_{i \in S})$
- 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:

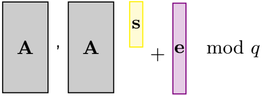
- Real: e_{ct} and $e_{flood} := \mathsf{Rec}(\{\, e_{flood,i} \,\}_{i \in S})$
- Simulated: only e_{flood}

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



Partial Decryption Security

Two worlds:

- Real: e_{ct} and $e_{flood} := \mathsf{Rec}(\{\, e_{flood,i} \,\}_{i \in S})$
- Simulated: only e_{flood}

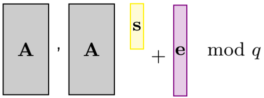
Can be avoided for Gaussian distributions in full threshold PKE setting! [MS23]

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

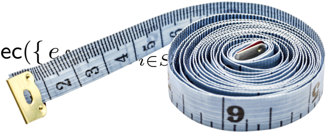


Partial Decryption Security

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

Two worlds:

- Real: e_{ct} and $e_{flood} := \text{Rec}(\{e$ $_{i \in S}$
- Simulated: only e_{flood}

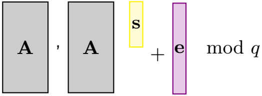


How close are they? [BD10] measures with statistical distance Δ

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

Problem:

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



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 [BS23]: Rényi Divergence

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

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 [BS23]: Rényi Divergence

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

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

$$\begin{array}{llcll} \text{[BD10]:} & P(E) & \leq & \Delta(P,Q) + Q(E) & \text{(additive)} \\ \textbf{Our work}: & P(E)^2 & \leq & \mathsf{RD}(P,Q) \cdot Q(E) & \text{(multiplicative)} \end{array}$$

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

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

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

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

   → from simulation-based to game-based security notion
2) Works well with search problems, not so well with decision problems

Research Directions

- Part 1: Different approach than adding noise?
- Part 2: Different approach for linear secret sharing?
- Part 3: Optimal noise analysis?

Part 4:

Defining Security

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

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

$(\mathsf{pk}, \mathsf{sk})$ — pk

$$\xleftarrow{\quad m_0, m_1 \quad}$$

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

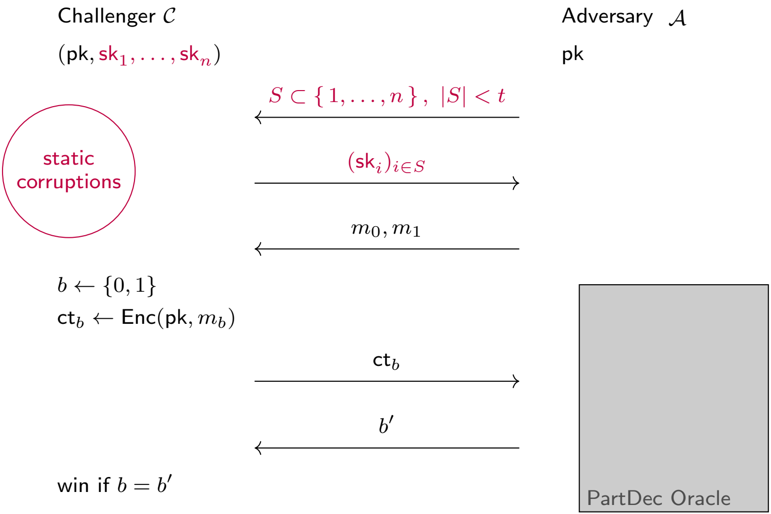
$\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$

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

$$\xleftarrow{\quad b' \quad}$$

win if $b = b'$

Goal: 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

$S \subset \{1, \ldots, n\},\ |S| < t$ $\longleftarrow$

static corruptions

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

m_0, m_1 $\longleftarrow$

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

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

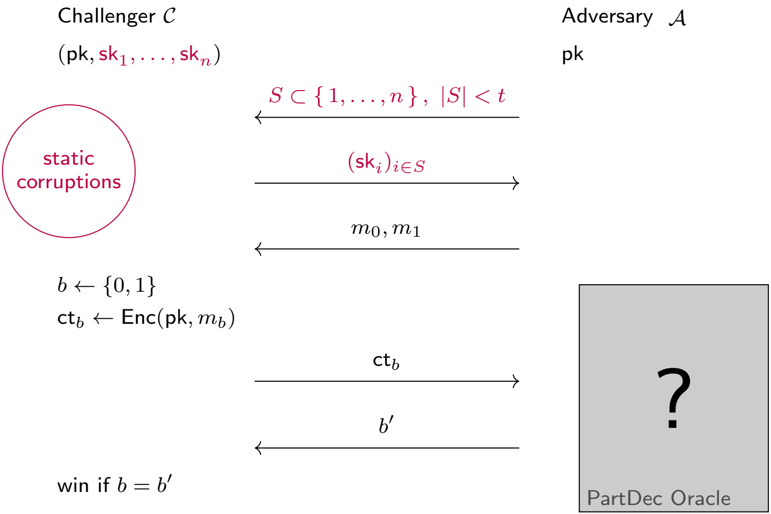
$\longrightarrow$ ct_b

b' $\longleftarrow$

win if $b = b'$

PartDec Oracle

Goal: 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

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

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

$$\xleftarrow{\quad m_0, m_1 \quad}$$

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

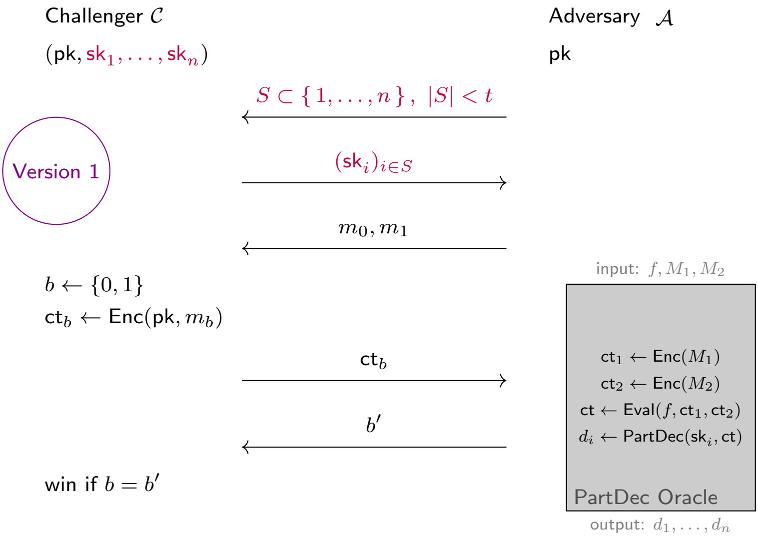
$\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$

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

$$\xleftarrow{\quad b' \quad}$$

win if $b = b'$

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



Challenger $\mathcal{C}$

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

Adversary $\mathcal{A}$

pk

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

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

$\mathsf{ct}_1 \leftarrow \mathsf{Enc}(M_1)$

$\mathsf{ct}_2 \leftarrow \mathsf{Enc}(M_2)$

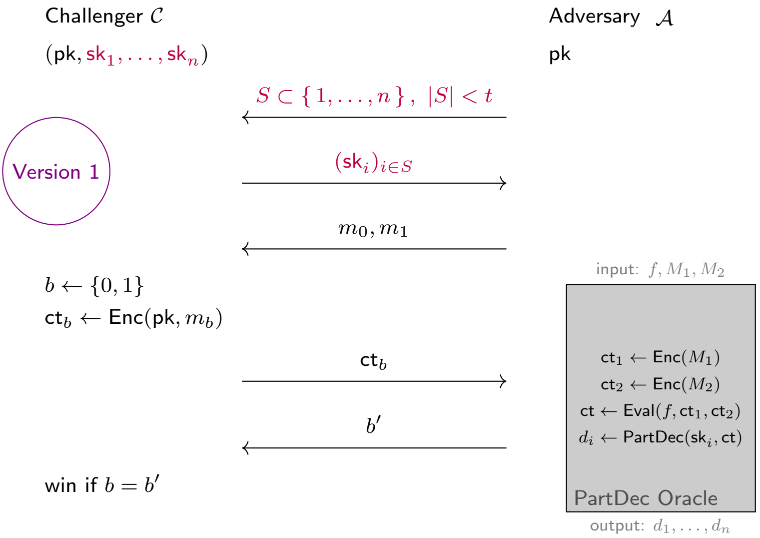
$\mathsf{ct} \leftarrow \mathsf{Eval}(f, \mathsf{ct}_1, \mathsf{ct}_2)$

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

PartDec Oracle

output: $d_1, \ldots, d_n$

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



Challenger $\mathcal{C}$

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

Adversary $\mathcal{A}$

pk

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

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

$\mathsf{ct}_1 \leftarrow \mathsf{Enc}(M_1)$

$\mathsf{ct}_2 \leftarrow \mathsf{Enc}(M_2)$

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

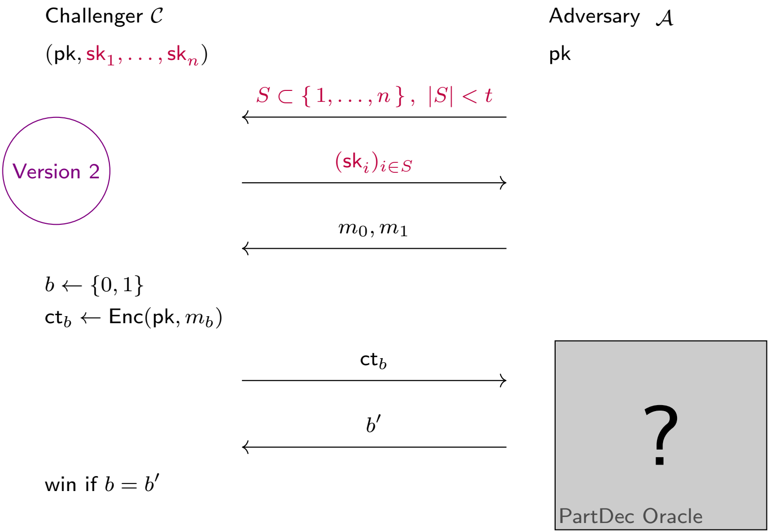
$d_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$

PartDec Oracle

output: $d_1, \ldots, d_n$

Very weak: queries to PartDec oracle are independent of ct_b

IND-CPA Security for t-out-of-n Threshold FHE [JRS17, BS23, CCP$^+$24]



Challenger $\mathcal{C}$

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

Adversary $\mathcal{A}$

pk

Version 2

$$S \subset \{1, \ldots, n\}, \ |S| < t \quad \longleftarrow$$

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

$$m_0, m_1 \quad \longleftarrow$$

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

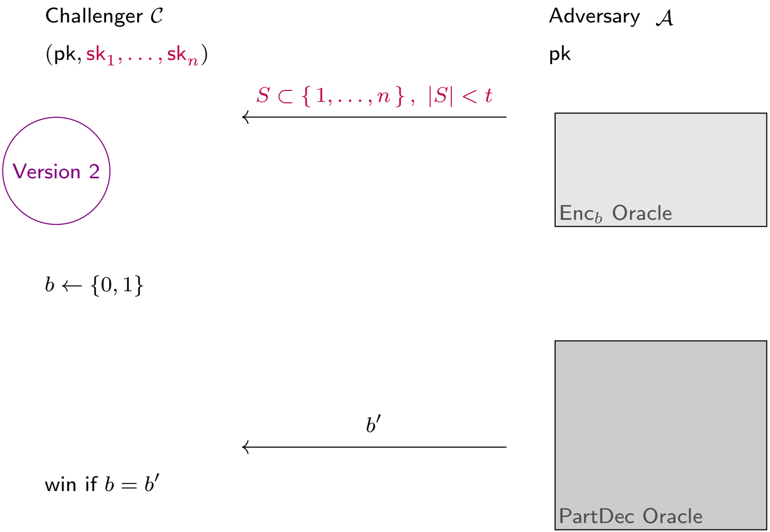
$\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$

$$\mathsf{ct}_b \quad \longrightarrow$$

$$b' \quad \longleftarrow$$

win if $b = b'$

IND-CPA Security for t-out-of-n Threshold FHE [JRS17, BS23, CCP$^+$24]



Challenger $\mathcal{C}$

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

Adversary $\mathcal{A}$

pk

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

Version 2

Enc_b Oracle

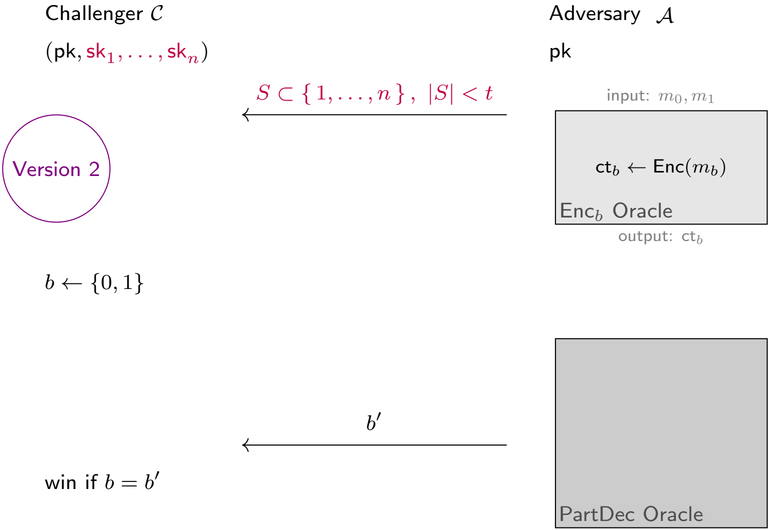
$b \leftarrow \{0, 1\}$

PartDec Oracle

b'

win if $b = b'$

IND-CPA Security for t-out-of-n Threshold FHE [JRS17, BS23, CCP$^+$24]



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

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

input: m_0, m_1

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

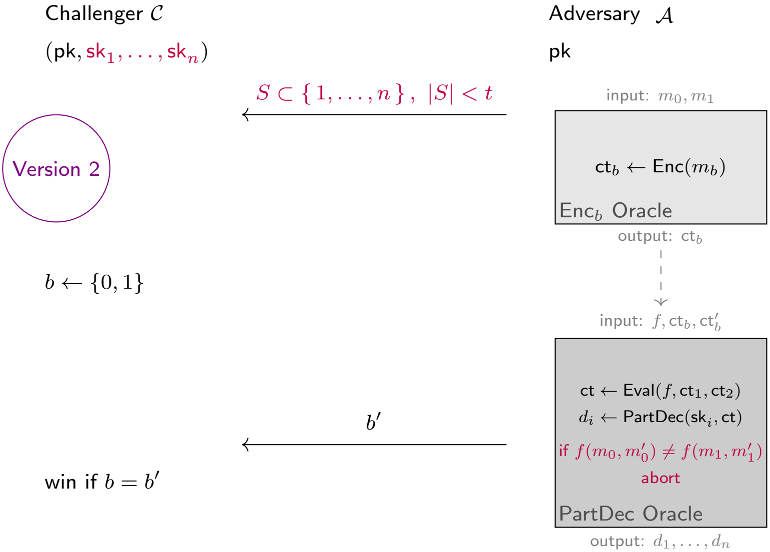
Enc_b Oracle

output: ct_b

PartDec Oracle

b'

IND-CPA Security for t-out-of-n Threshold FHE [JRS17, BS23, CCP+24]



Challenger $\mathcal{C}$

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

Adversary $\mathcal{A}$

pk

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

Version 2

input: m_0, m_1

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

Enc_b Oracle

output: ct_b

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

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

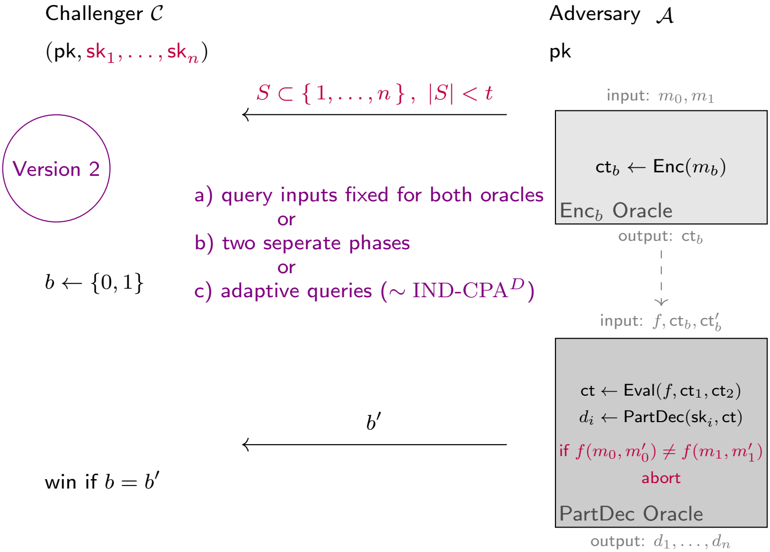
PartDec Oracle

output: $d_1, \ldots, d_n$

b'

win if $b = b'$

IND-CPA Security for t-out-of-n Threshold FHE [JRS17, BS23, CCP$^+$24]



Challenger $\mathcal{C}$

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

Adversary $\mathcal{A}$

pk

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

Version 2

input: m_0, m_1

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

Enc_b Oracle

output: ct_b

a) query inputs fixed for both oracles

or

b) two seperate phases

or

c) adaptive queries ($\sim$ IND-CPA^D)

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

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: $d_1, \ldots, d_n$

b'

win if $b = b'$

Research Directions

- Part 1: Different approach than adding noise?
- Part 2: Different approach for linear secret sharing?
- Part 3: Different noise analysis?
- Part 4: Best efficiency-security trade-off?

Wrap-Up

⚑ Hopefully you have now a rough idea:

- Part 1: *What the blueprint of ThFHE is!*
- Part 2: *What suitable secret sharings are!*
- Part 3: *How to use flooding noise!*
- Part 4: *How to define security!*
- **What research directions there are :-)**

Any questions or interested in my research?

💬 Reach out to me today & during EC24

✉ Write me an e-mail

Wrap-Up

Hopefully you have now a rough idea:

- Part 1: *What the blueprint of ThFHE is!*
- Part 2: *What suitable secret sharings are!*
- Part 3: *How to use flooding noise!*
- Part 4: *How to define security!*
- **What research directions there are :-)**

Thanks!

Any questions or interested in my research?

- Reach out to me today & during EC24
- ✉ Write me an e-mail

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.

Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai.
Threshold cryptosystems from threshold fully homomorphic encryption.
In *CRYPTO (1)*, volume 10991 of *Lecture Notes in Computer Science*, pages 565–596. Springer, 2018.

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.

Katharina Boudgoust and Peter Scholl.
Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus.
IACR Cryptol. ePrint Arch., page 16, 2023.

Jung Hee Cheon, Wonhee Cho, and Jiseung Kim.
Improved universal thresholdizer from threshold fully homomorphic encryption.
IACR Cryptol. ePrint Arch., page 545, 2023.

Jung Hee Cheon, Hyeongmin Choe, Alain Passelègue, Damien Stehlé, and Elias Suvanto.
Attacks against the INDCPA-D security of exact FHE schemes.
IACR Cryptol. ePrint Arch., page 127, 2024.

Yvo Desmedt and Yair Frankel.
Threshold cryptosystems.
In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 1989.

Kamil Doruk Gür, Jonathan Katz, and Tjerand Silde.
Two-round threshold lattice signatures from threshold homomorphic encryption.
IACR Cryptol. ePrint Arch., page 1318, 2023.

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

Christian Mouchet, Elliott Bertrand, and Jean-Pierre Hubaux.
An efficient threshold access-structure for rlwe-based multiparty homomorphic encryption.
J. Cryptol., 36(2):10, 2023.

Daniele Micciancio and Adam Suhl.
Simulation-secure threshold PKE from LWE with polynomial modulus.
IACR Cryptol. ePrint Arch., page 1728, 2023.

Victor Shoup.
Practical threshold signatures.
In *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 207–220. Springer, 2000.