

Adaptive Oblivious Transfer with Access Control from Lattice Assumptions

Benoît Libert^{1,2} San Ling³ **Fabrice Mouhartem**¹
Khoa Nguyen³ Huaxiong Wang³

¹École Normale Supérieure de Lyon, ²CNRS,
³Nanyang Technological University (Singapour)

Cryptography Seminar
University of Oxford
8th of November 2017



NANYANG
TECHNOLOGICAL
UNIVERSITY



UNIVERSITÉ
DE LYON

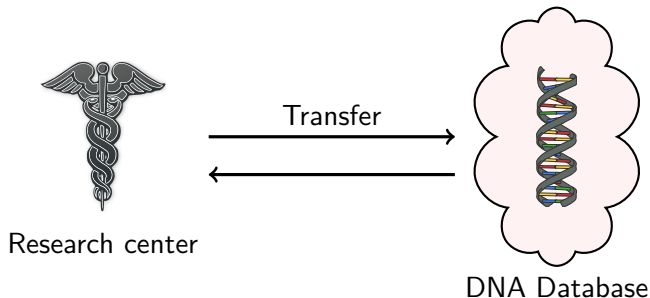
Anonymous Databases



Research center

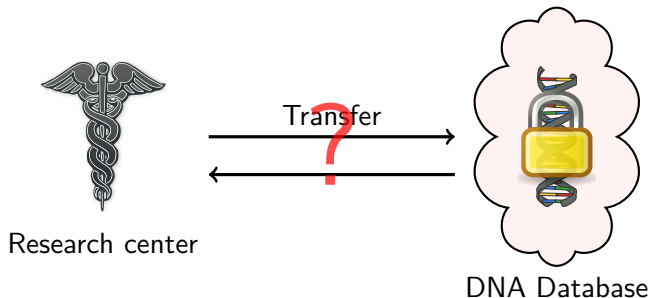
- ▶ DNA storage is expensive

Anonymous Databases



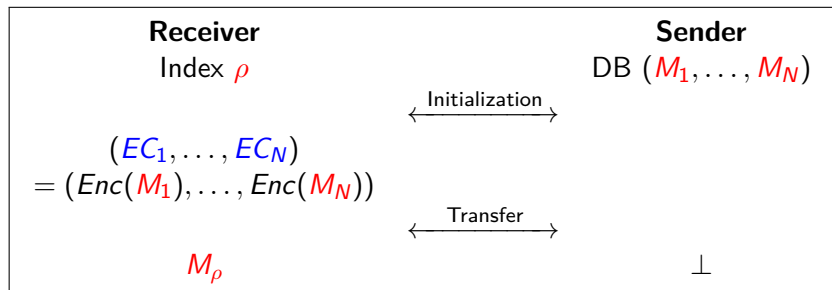
- ▶ DNA storage is expensive
- ▶ DNA Database's requests are sensitive

Anonymous Databases

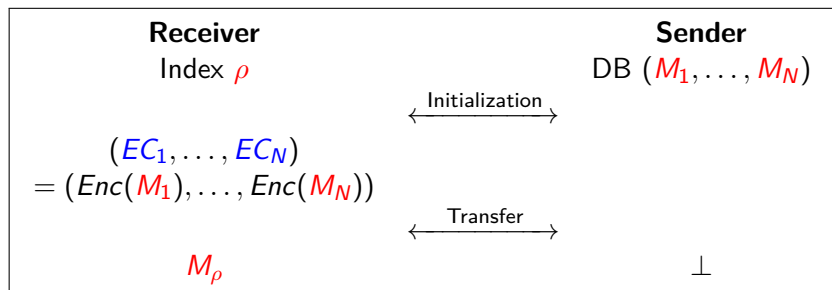


- ▶ DNA storage is expensive
 - ▶ DNA Database's requests are sensitive
- query-anonymous transfers

(Adaptive) Oblivious Transfer (OT) [Chau81]

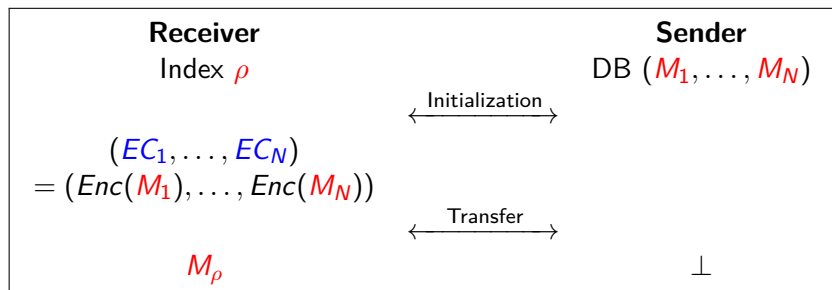


(Adaptive) Oblivious Transfer (OT) [Chau81]



- ▶ Complete building block of cryptography [GMW87]

(Adaptive) Oblivious Transfer (OT) [Chau81]



- ▶ Complete building block of cryptography [GMW87]
- ▶ **Adaptive** OT: receiver adaptively obtains k messages [NP93]
 - Usage: Sensitive DB (DNA, financial data, ...).

History

1981 [Chaum](#): introduction

1985 [Even, Goldreich and Lempel](#): extension

History

1981 [Chaum](#): introduction

1985 [Even, Goldreich and Lempel](#): extension

1987 [Goldreich, Micali and Wigderson](#): OT implies MPC

History

1981 [Chaum](#): introduction

1985 [Even, Goldreich and Lempel](#): extension

1987 [Goldreich, Micali and Wigderson](#): OT implies MPC

1993 [Naor and Pinkas](#): **adaptive** OT

2007 [Camenisch, Neven and shelat](#): assisted decryption

2008 [Green and Hohenberger](#): adaptive OT in the UC model

2011 [Green and Hohenberger](#): adaptive OT from pairings

History

1981 [Chaum](#): introduction

1985 [Even, Goldreich and Lempel](#): extension

1987 [Goldreich, Micali and Wigderson](#): OT implies MPC

1993 [Naor and Pinkas](#): **adaptive** OT

2007 [Camenisch, Neven and shelat](#): assisted decryption

2008 [Green and Hohenberger](#): adaptive OT in the UC model

2009 [Camenisch, Dubovitskaya and Neven](#): access control

2011 [Green and Hohenberger](#): adaptive OT from pairings

► From FHE + OPRF, or PIR, or ad-hoc pairing assumptions. . .

No fully simulatable adaptive OT with access control from lattice assumptions

Lattice

A lattice is a discrete subgroup of \mathbb{R}^n . Can be seen as integer linear combinations of a finite set of vectors.

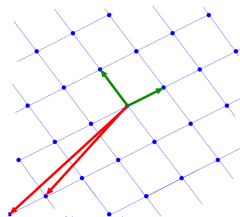
$$\Lambda(\mathbf{b}_1, \dots, \mathbf{b}_n) = \left\{ \sum_{i \leq n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \right\}$$

Lattice-Based Cryptography [Ajt96, Reg05]

Lattice

A lattice is a discrete subgroup of \mathbb{R}^n . Can be seen as integer linear combinations of a finite set of vectors.

$$\Lambda(\mathbf{b}_1, \dots, \mathbf{b}_n) = \left\{ \sum_{i \leq n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \right\}$$



Why?

- ▶ Simple and asymptotically efficient;
- ▶ **Still** conjectured quantum-resistant;
- ▶ Connection between average-case and worst-case problems;
- ▶ Powerful functionalities (e.g., FHE).

→ Finding a short non-zero vector in a lattice is hard.

Hardness Assumptions: SIS and LWE [Ajt96, Reg05]

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

For $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$:

Small Integer Solution

$$\mathbf{x} \mathbf{A} = \mathbf{0} [q]$$

Goal: Given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, find $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ small

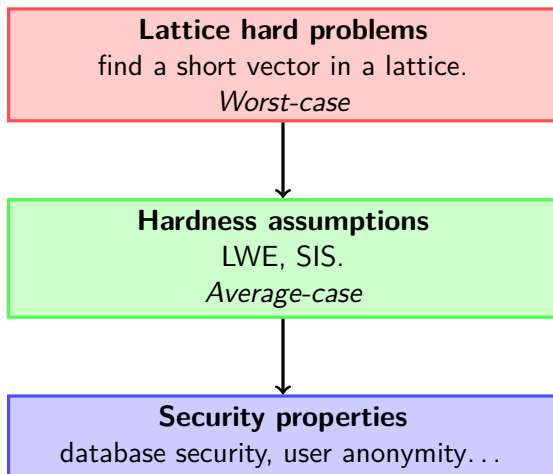
Learning With Errors

$$\left(\begin{array}{c} m \\ \mathbf{A} \end{array}, \mathbf{A} \mathbf{s} + \mathbf{e} \right)$$

$\mathbf{s} \leftarrow \mathbb{Z}_q^n$ \mathbf{e} small error

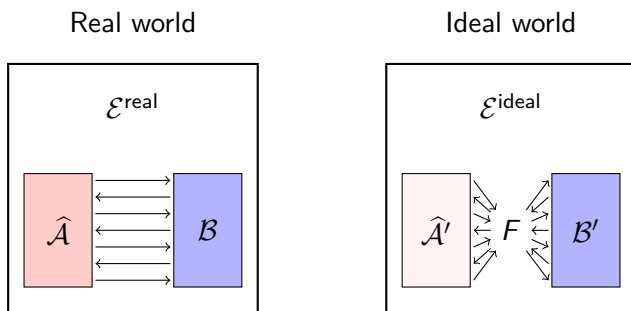
Goal: Given $(\mathbf{A}, \mathbf{A} \mathbf{s} + \mathbf{e})$, find $\mathbf{s} \in \mathbb{Z}_q^n$

Provable Security



Full Simulation Model [\[Can01\]](#)

For any cheating $\hat{\mathcal{A}}$, there exists $\hat{\mathcal{A}}'$ s.t.



$$\text{View}(\mathcal{E}^{\text{real}}) \approx_s \text{View}(\mathcal{E}^{\text{ideal}})$$

- ▶ Strictly stronger security model than indistinguishability-based one

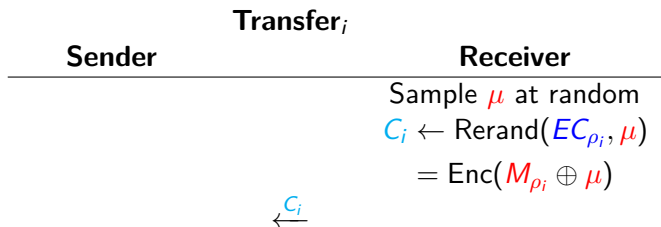
Outline

Introduction

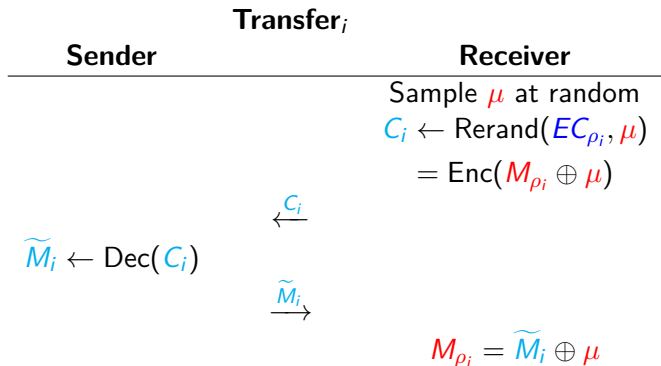
Building Blocks

Adaptive Oblivious Transfer with Access Control

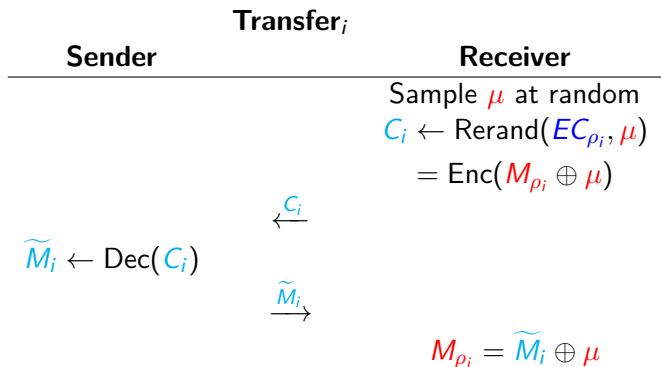
Assisted Decryption Technique [CNs07]



Assisted Decryption Technique [CNs07]



Assisted Decryption Technique [CNs07]



+ **Zero-knowledge proofs** compatible with the
PKE (Keygen, Enc, Dec, Rerand)

Regev's Encryption [Reg05]

Keygen:

Secret key: $\mathbf{S} \leftarrow \chi^{n \times t}$

Public key: (\mathbf{F}, \mathbf{P}) s.t. $\mathbf{F} \leftarrow U(\mathbb{Z}_q^{n \times m})$, $\mathbf{P} = \mathbf{F}^T \mathbf{S} + \mathbf{E}$

with $\mathbf{E} \leftarrow \chi^{m \times t}$

Encryption: $(\mathbf{a}, \mathbf{b}) = (\mathbf{a}, \mathbf{S}^T \mathbf{a} + \mathbf{x} + M \lfloor \frac{q}{2} \rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$

Decryption: $M = \lfloor (\mathbf{b} - \mathbf{S}^T \cdot \mathbf{a}) / (\frac{q}{2}) \rfloor$

Rerand: $(\mathbf{a}', \mathbf{b}') = (\mathbf{a} + \mathbf{F} \mathbf{e}, \mathbf{b} + \mathbf{P}^T \mathbf{e} + \mu \lfloor \frac{q}{2} \rfloor) = \text{Enc}(M \oplus \mu)$

Regev's Encryption [Reg05]

Keygen:

Secret key: $\mathbf{S} \leftarrow \chi^{n \times t}$

Public key: (\mathbf{F}, \mathbf{P}) s.t. $\mathbf{F} \leftarrow U(\mathbb{Z}_q^{n \times m})$, $\mathbf{P} = \mathbf{F}^T \mathbf{S} + \mathbf{E}$

with $\mathbf{E} \leftarrow \chi^{m \times t}$

Encryption: $(\mathbf{a}, \mathbf{b}) = (\mathbf{a}, \mathbf{S}^T \mathbf{a} + \mathbf{x} + M \lfloor \frac{q}{2} \rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$

Decryption: $M = \lfloor (\mathbf{b} - \mathbf{S}^T \cdot \mathbf{a}) / (\frac{q}{2}) \rfloor$

Rerand: $(\mathbf{a}', \mathbf{b}') = (\mathbf{a} + \mathbf{F} \mathbf{e}, \mathbf{b} + \mathbf{P}^T \mathbf{e} + \mu \lfloor \frac{q}{2} \rfloor) = \text{Enc}(M \oplus \mu)$

+ ZK proofs

Problem

The Sender only proves bounded noise x for Regev encryption.

Problem

The Sender only proves bounded noise \mathbf{x} for Regev encryption.

Attack scenario:

- ▶ $DB = (M_0, M_1)$
 - ▶ Sender encrypts M_0 with noise \mathbf{x}_0 and M_1 with noise \mathbf{x}_1 s.t.
 $\|\mathbf{x}_0\| \ll \|\mathbf{x}_1\| \leq B_\chi$
 - ▶ Upon receiving $(\mathbf{c}_0, \mathbf{c}_1)$, decryption leaks $\|\mathbf{x}_i + \mathbf{e}\|$
- \Rightarrow Sender can break Receiver messages' anonymity

Problem

The Sender only proves bounded noise \mathbf{x} for Regev encryption.

Attack scenario:

- ▶ $DB = (M_0, M_1)$
- ▶ Sender encrypts M_0 with noise \mathbf{x}_0 and M_1 with noise \mathbf{x}_1 s.t.
 $\|\mathbf{x}_0\| \ll \|\mathbf{x}_1\| \leq B_\chi$
- ▶ Upon receiving $(\mathbf{c}_0, \mathbf{c}_1)$, decryption leaks $\|\mathbf{x}_i + \mathbf{e}\|$

\Rightarrow Sender can break Receiver messages' anonymity

Solution: Flood the noise with ν s.t. $\|\nu\| \gg B_\chi$.

Rerand: $(\mathbf{a}', \mathbf{b}') = (\mathbf{a} + \mathbf{F} \mathbf{e}, \mathbf{b} + \mathbf{P}^T \mathbf{e} + \mu \lfloor \frac{q}{2} \rfloor + \nu)$

AC-OT

Encrypted database consists in $(EC_i, AP_i)_{i=1}^N$.

Receiver can retrieve message M_i iff it possesses cert_x for some $x \in \{0, 1\}^*$ s.t. $AP_i(x) = 1$.

AC-OT

Encrypted database consists in $(EC_i, AP_i)_{i=1}^N$.

Receiver can retrieve message M_i iff it possesses cert_x for some $x \in \{0, 1\}^*$ s.t. $AP_i(x) = 1$.

[ACDN13]: access policy made of conjunctions: $x_1 \wedge \dots \wedge x_\ell$.

Disjunctions possible through replication.

[ZAW+10]: use CP-ABE to handle NC^1 access policies.

AC-OT

Encrypted database consists in $(EC_i, AP_i)_{i=1}^N$.

Receiver can retrieve message M_i iff it possesses cert_x for some $x \in \{0, 1\}^*$ s.t. $AP_i(x) = 1$.

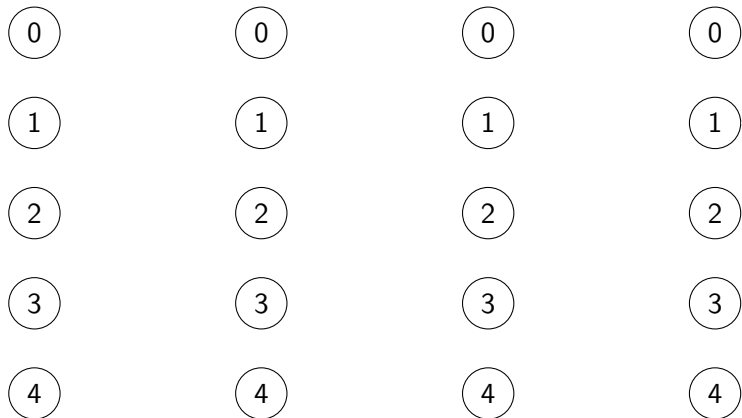
[ACDN13]: access policy made of conjunctions: $x_1 \wedge \dots \wedge x_\ell$.

Disjunctions possible through replication.

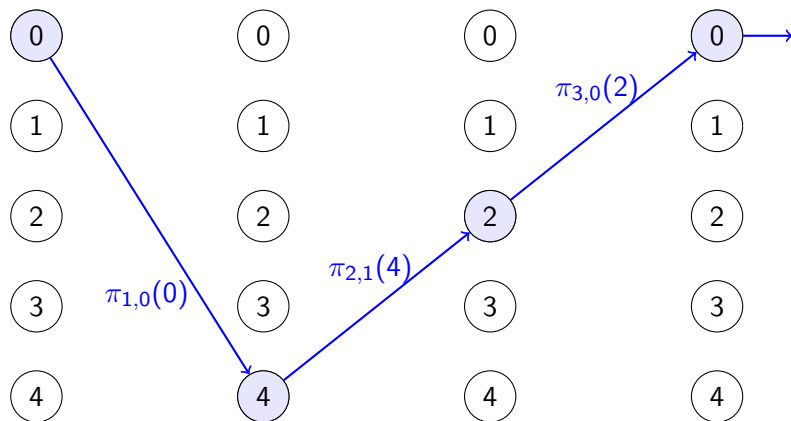
[ZAW+10]: use CP-ABE to handle NC^1 access policies.

Here: access policy made of **branching program** (BP).

Branching Programs

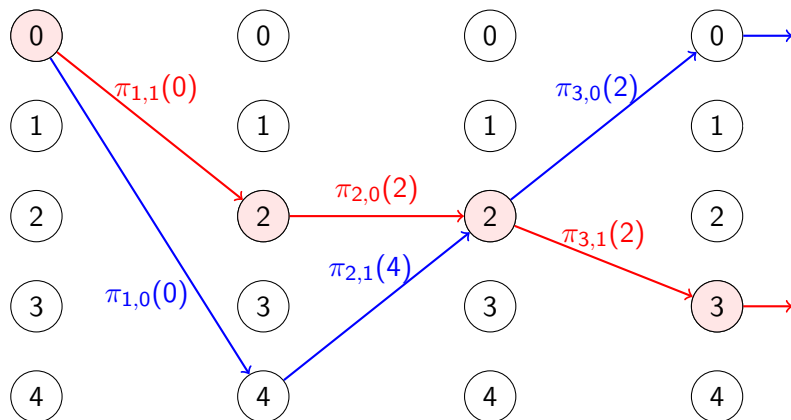


Branching Programs



$x = 010$: accepted

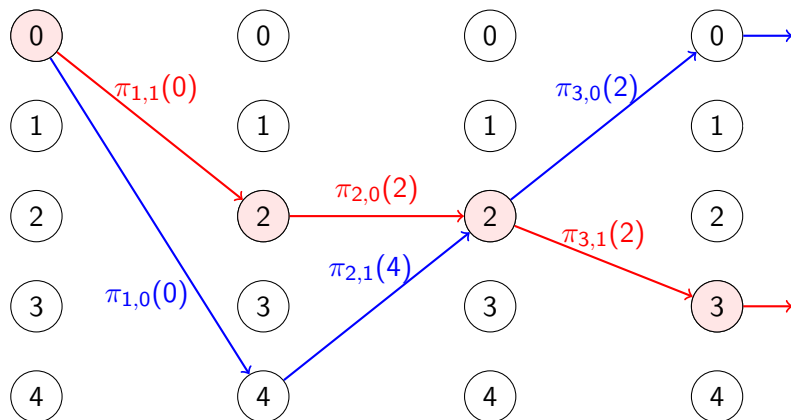
Branching Programs



$x = 010$: accepted

$y = 101$: rejected

Branching Programs




$x = 010$: accepted
 $y = 101$: rejected

[Barr86]: polynomially-long BP
are equivalent to NC^1

Branching Programs

$$\{(EC_1, BP_1), (EC_2, BP_2), \dots, (EC_N, BP_N)\}.$$




$(x, cert_x)$

Goal: Prove knowledge of $cert_x = \text{Sign}(x)$ s.t. $\exists i : BP_i(x) = 1$.

Branching Programs

$$\{(EC_1, BP_1), (EC_2, BP_2), \dots, (EC_N, BP_N)\}.$$



$(x, cert_x)$

Goal: Prove knowledge of $cert_x = \text{Sign}(x)$ s.t. $\exists i : BP_i(x) = 1$.

This work: Make BP's statements work with Stern's proofs [Ste93]

Branching Programs

Encoding of a branching program:

$$\mathbf{z}_{BP} = (d_{1,1}, \dots, d_{1,\delta_\kappa}, \dots, d_{L,1}, \dots, d_{L,\delta_\kappa}, \pi_{1,0}(0), \dots, \pi_{1,0}(4), \pi_{1,1}(0), \dots, \pi_{1,1}(4), \dots, \pi_{L,0}(0), \dots, \pi_{L,0}(4), \pi_{L,1}(0), \dots, \pi_{L,1}(4)) \in [0, 4]^\zeta$$

$d_{\theta,1}, \dots, d_{\theta,\delta_\kappa}$: bit representation of $\text{var}(\theta)$

Step-by-step evaluation:

$$\eta_\theta = \pi_{\theta, \mathbf{x}_{\text{var}(\theta)}}(\eta_{\theta-1}) = \pi_{\theta,0}(\eta_{\theta-1}) \cdot \bar{\mathbf{x}}_{\text{var}(\theta)} + \pi_{\theta,1}(\eta_{\theta-1}) \cdot \mathbf{x}_{\text{var}(\theta)}$$

Proving correct evaluation

Naively: prove each step $\rightarrow O(L \cdot \kappa)$

Branching Programs

Encoding of a branching program:

$$\mathbf{z}_{BP} = (d_{1,1}, \dots, d_{1,\delta_\kappa}, \dots, d_{L,1}, \dots, d_{L,\delta_\kappa}, \pi_{1,0}(0), \dots, \pi_{1,0}(4), \pi_{1,1}(0), \dots, \pi_{1,1}(4), \dots, \pi_{L,0}(0), \dots, \pi_{L,0}(4), \pi_{L,1}(0), \dots, \pi_{L,1}(4)) \in [0, 4]^\zeta$$

$d_{\theta,1}, \dots, d_{\theta,\delta_\kappa}$: bit representation of $\text{var}(\theta)$

Step-by-step evaluation:

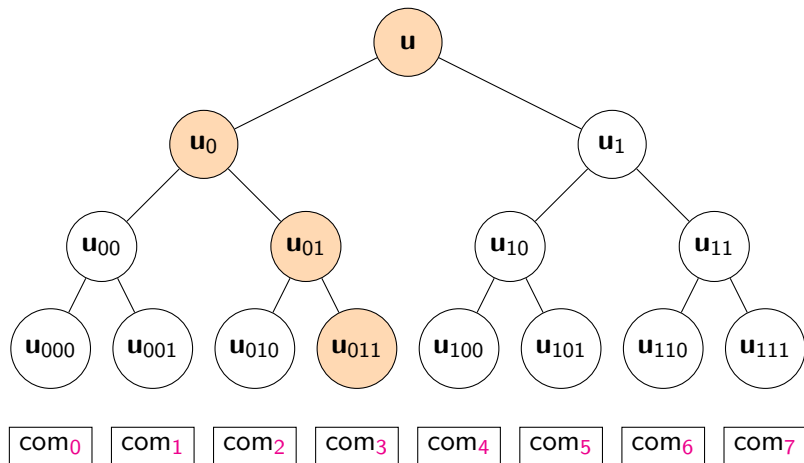
$$\eta_\theta = \pi_{\theta, \mathbf{x}_{\text{var}(\theta)}}(\eta_{\theta-1}) = \pi_{\theta,0}(\eta_{\theta-1}) \cdot \bar{\mathbf{x}}_{\text{var}(\theta)} + \pi_{\theta,1}(\eta_{\theta-1}) \cdot \mathbf{x}_{\text{var}(\theta)}$$

Proving correct evaluation

Naively: prove each step $\rightarrow O(L \cdot \kappa)$

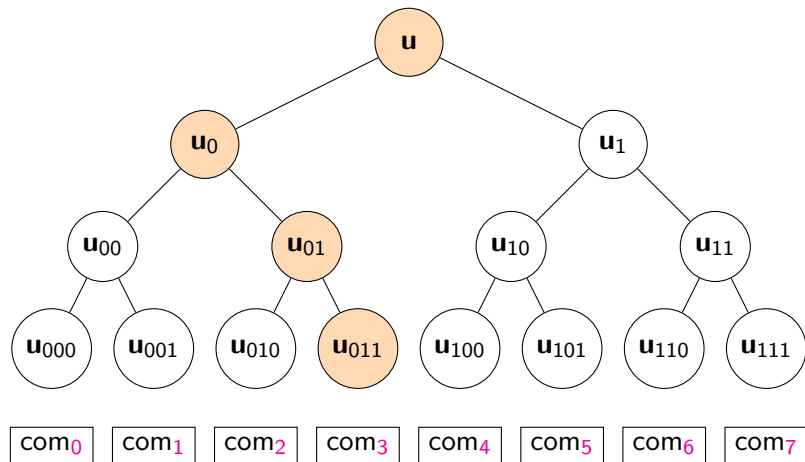
Our idea: use binary-search $\rightarrow O(L \cdot \log(\kappa))$

Branching Programs



$$com_i = Com(x_i)$$

Branching Programs



$com_i = Com(x_i) \rightarrow$ correct binary search \Rightarrow knowledge of $x_{var(\theta)}$

Commitments

Digital equivalent of a sealed box.



e.g., Pedersen Commitment

$$pk = (g, h) \leftarrow \mathbb{G}^2$$

$$com = g^m \cdot h^r$$

$$open = (m, r)$$

Commitments

Digital equivalent of a sealed box.



e.g., Pedersen Commitment

$$pk = (g, h) \leftarrow \mathbb{G}^2$$

$$com = g^m \cdot h^r$$

$$open = (m, r)$$

Properties

Commitments provide

- ▶ **Binding** property: once sealed, a value cannot be changed
- ▶ **Hiding** property: nobody is able tell what is inside the box without the key

Zero-Knowledge (ZK) Proofs

Definition

A ZK proof is an interactive protocol between prover P and verifier V that verifies:

Completeness: Correctness of the protocol.

Soundness: No cheating prover can convince the verifier.

Zero-Knowledge: Verifier learns nothing but the validity of the statement.

Zero-Knowledge (ZK) Proofs

Definition

A ZK proof is an interactive protocol between prover P and verifier V that verifies:

Completeness: Correctness of the protocol.

Soundness: No cheating prover can convince the verifier.

Zero-Knowledge: Verifier learns nothing but the validity of the statement.

- ▶ Non interactive variant exists: NIZK proofs
- ▶ Random-Oracle Model: transform from ZK to NIZK [FS86]
- ▶ Standard Model: Groth-Sahai for pairing-based crypto [GS08]

Zero-Knowledge Proofs for Lattices

No equivalent of Groth-Sahai proofs!

Reason

Lattices propose less algebraic structure than pairing groups.

Zero-Knowledge Proofs for Lattices

No equivalent of Groth-Sahai proofs!

Reason

Lattices propose less algebraic structure than pairing groups.

Two main proof systems in lattice-based cryptography:

Lyubashevky like [Lyu09]: On Ring-LWE, concise but not expressive.
Algebraic

Stern like [Ste93]: On LWE, heavy but expressive.
Combinatorial

Both are interactive.

Stern's Protocol (Crypto'93)

Stern's protocol is a ZK proof for Syndrome Decoding Problem.

Stern's Protocol (Crypto'93)

Stern's protocol is a ZK proof for Syndrome Decoding Problem.

Syndrome Decoding Problem

Given $\mathbf{P} \in \mathbb{Z}_2^{n \times m}$ and $\mathbf{v} \in \mathbb{Z}_2^n$, find \mathbf{x} s.t. $w(\mathbf{x}) = w$ and

$$\mathbf{P} \mathbf{x} = \mathbf{v} \pmod{2}$$

Stern's Protocol (Crypto'93)

Stern's protocol is a ZK proof for Syndrome Decoding Problem.

Syndrome Decoding Problem

Given $\mathbf{P} \in \mathbb{Z}_2^{n \times m}$ and $\mathbf{v} \in \mathbb{Z}_2^n$, find \mathbf{x} s.t. $w(\mathbf{x}) = w$ and

$$\begin{matrix} & \xrightarrow{m} \\ \begin{matrix} \uparrow n \\ \downarrow \end{matrix} & \mathbf{P} & \mathbf{x} = \mathbf{v} \pmod{2} \end{matrix}$$

[KTX08]: $\text{mod } 2 \rightarrow \text{mod } q$

[LNSW13]: Extends Stern's protocol for SIS and LWE statements

Recent uses of Stern-like protocols in lattice-based crypto:

[LNW15, LLNW16, LLNMW16a, LLNMW16b, LLNW17]

Signature with Efficient Protocols [CL02]

A signature scheme (**Keygen**, **Sign**_{sk}, **Verif**_{vk}) with companion protocols:

- ▶ **Sign** a committed value;
- ▶ Prove possession of a signature in **ZK**.

Signature with Efficient Protocols [CL02]

A signature scheme (**Keygen**, **Sign**_{sk}, **Verif**_{vk}) with companion protocols:

- ▶ **Sign** a committed value;
- ▶ Prove possession of a signature in **ZK**.

Security

- ▶ **Unforgeability**;

Signature with Efficient Protocols [CL02]

A signature scheme (**Keygen**, **Sign**_{sk}, **Verif**_{vk}) with companion protocols:

- ▶ **Sign** a committed value;
- ▶ Prove possession of a signature in **ZK**.

Security

- ▶ **Unforgeability**;
- ▶ **Security** of the two protocols;
- ▶ **Anonymity**.

Signature with Efficient Protocols [CL02]

A signature scheme (**Keygen**, **Sign**_{sk}, **Verif**_{vk}) with companion protocols:

- ▶ **Sign** a committed value;
- ▶ Prove possession of a signature in **ZK**.

Security

- ▶ **Unforgeability**;
- ▶ **Security** of the two protocols;
- ▶ **Anonymity**.

→ many applications for privacy-based protocols

Signature with Efficient Protocols [\[CL02\]](#)

A signature scheme (**Keygen**, **Sign**_{sk}, **Verif**_{vk}) with companion protocols:

- ▶ **Sign** a committed value;
- ▶ Prove possession of a signature in **ZK**.

Security

- ▶ **Unforgeability**;
- ▶ **Security** of the two protocols;
- ▶ **Anonymity**.

→ many applications for privacy-based protocols

We use a SIS-based construction from Asiacrypt'16 [\[LLNMW16\]](#).

Outline

Introduction

Building Blocks

Adaptive Oblivious Transfer with Access Control

Our Construction

Ingredients

- ▶ The assisted decryption technique
- ▶ A simplification of [LLNMW16]'s signatures as certificates
- ▶ Access control using BP
- ▶ WI proofs *à la* Stern

Our Adaptive Oblivious Transfer Construction

Initialization

Sender side:

1. Generate $(VK_{sig}, SK_{sig}) \leftarrow \Sigma.keygen(1^\lambda)$
2. Compute $((\mathbf{S}, \mathbf{E}), (\mathbf{F}, \mathbf{P})) \leftarrow Regev.keygen(1^\lambda)$
3. Use \mathbf{S} to compute encryptions of $M_i \rightarrow (\mathbf{a}_i, \mathbf{b}_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$
4. Use SK_{sig} to compute signatures of $(\mathbf{a}_i, \mathbf{b}_i) \rightarrow \sigma_i$
5. $EC_i \leftarrow (\sigma_i, (\mathbf{a}_i, \mathbf{b}_i))$

Our Adaptive Oblivious Transfer Construction

Initialization

Sender side:

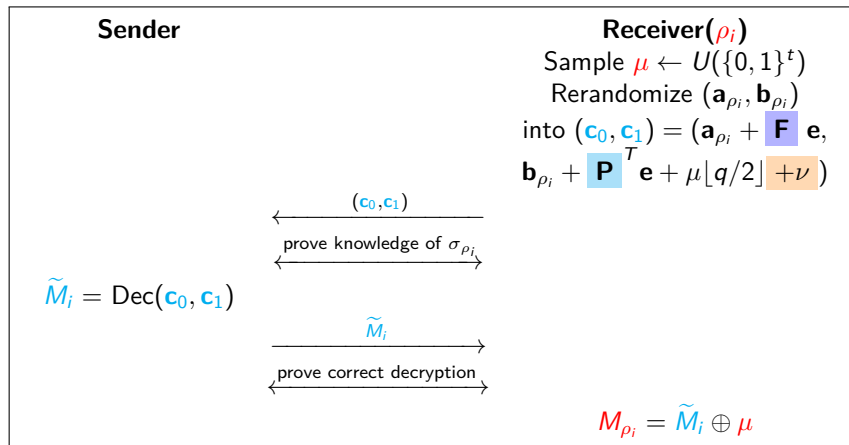
1. Generate $(VK_{sig}, SK_{sig}) \leftarrow \Sigma.keygen(1^\lambda)$
2. Compute $((\mathbf{S}, \mathbf{E}), (\mathbf{F}, \mathbf{P})) \leftarrow Regev.keygen(1^\lambda)$
3. Use \mathbf{S} to compute encryptions of $M_i \rightarrow (\mathbf{a}_i, \mathbf{b}_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$
4. Use SK_{sig} to compute signatures of $(\mathbf{a}_i, \mathbf{b}_i) \rightarrow \sigma_i$
5. $EC_i \leftarrow (\sigma_i, (\mathbf{a}_i, \mathbf{b}_i))$

Sender sends $(VK_{sig}, (\mathbf{F}, \mathbf{P}), EC_i)$ to the Receiver and proves that everything was done correctly.

Sender keeps the previous information and (\mathbf{S}, \mathbf{E}) .

Our Adaptive Oblivious Transfer Construction

Transfer



Our Adaptive Oblivious Transfer Construction

Final steps

- ▶ Access control can be plug in our scheme

Our Adaptive Oblivious Transfer Construction

Final steps

- ▶ Access control can be plug in our scheme
- ▶ Our scheme is proven secure in the standard model
 - In the ROM: optimizations using NIWI proofs [\[FS86\]](#)

Comparison

Adaptive OT with Access Control

Protocol	Initialization Cost	Transfer Cost	Assumptions	Policies	Private Policies	Security
CDN09	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \text{Poly}(\lambda)$	q -type	Conj.	✗	Full-Sim
CDNZ11	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \text{Poly}(\lambda)$	q -type + XDDH	Conj.	✓	Full-Sim
ACDN13	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \text{Poly}(\lambda)$	DLIN + SXDH	Conj.	✗	UC
ZAW+10	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda)$	CP-ABE + q -type	NC ¹	✗	Full-Sim
CDEN12	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda \log N) + \text{Poly}(\lambda)$	CP-ABE + GGM	CNF ⁻	✓	Full-Sim
Ours	$\mathcal{O}(\lambda \cdot N)$	$\widetilde{\mathcal{O}}(\lambda \log N) + \text{Poly}(\lambda)$	LWE + SIS	NC ¹	✗	Full-Sim

CNF⁻ is a restricted version of CNF: $\text{AP}(EC) = \bigwedge_i \bigvee_j x_{i,j}$ where each $x_{i,j}$ belongs to a given category C_i (e.g. {Surgeon, Nurse, Administrative}).

Conclusion

- ▶ First AC-OT in the lattice setting that handles expressive statements (NC^1)
- ▶ Rely on LWE with superpolynomial modulus
- ▶ Proved in the full simulation model

Possible improvements:

- ▶ Remove smudging

Questions?

