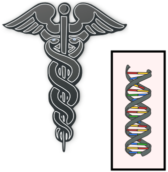
Adaptive Oblivious Transfer with Access Control from Lattice Assumptions

Benoît Libert[1,2] San Ling[3] **Fabrice Mouhartem**[1]
Khoa Nguyen[3] Huaxiong Wang[3]

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

Séminaire de Cryptographie
Université de Rennes
16 Juin 2017

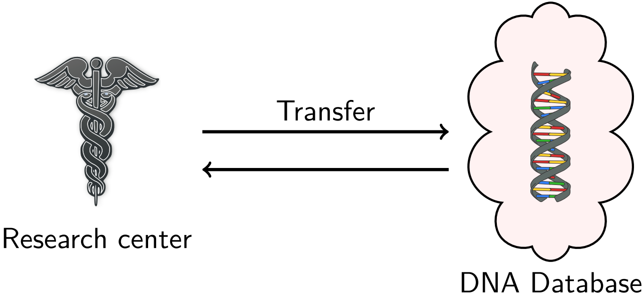
Anonymous Databases



Research center

- DNA storage is expensive

Anonymous Databases



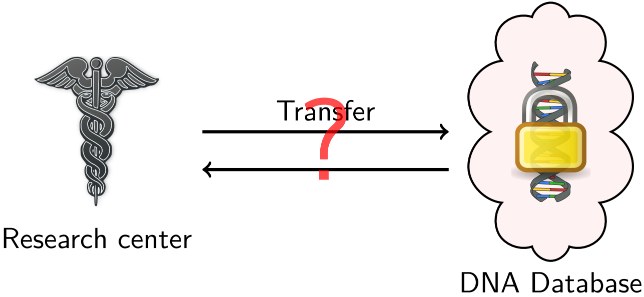
Research center

Transfer

DNA Database

- 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]

Sender		**Receiver**
DB $(M_1, \ldots, M_N)$		Index ρ
	$\xleftrightarrow{\text{Initialization}}$	
		$(EC_1, \ldots, EC_N)$ $= (Enc(M_1), \ldots, Enc(M_N))$
	$\xleftrightarrow{\text{Transfer}}$	
$\perp$		M_ρ

(Adaptive) Oblivious Transfer (OT) [Chau81]

Sender

DB $(M_1, \ldots, M_N)$

Receiver

Index ρ

$$\xleftrightarrow{\text{Initialization}}$$

$$(EC_1, \ldots, EC_N) = (Enc(M_1), \ldots, Enc(M_N))$$

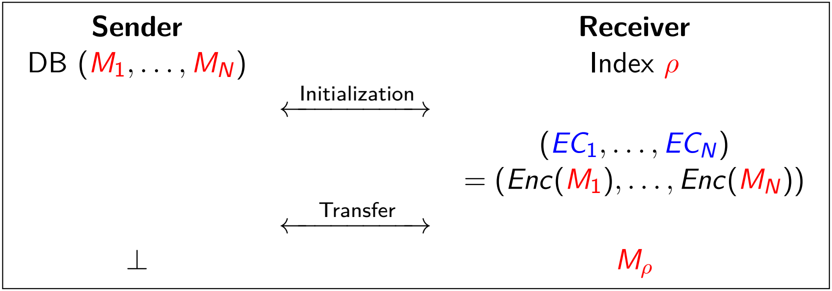
$$\xleftrightarrow{\text{Transfer}}$$

$\perp$

M_ρ

- Complete building block of cryptography [GMW87]

(Adaptive) Oblivious Transfer (OT) [Chau81]



Sender

DB $(M_1, \ldots, M_N)$

Receiver

Index ρ

$\xleftrightarrow{\text{Initialization}}$

$(EC_1, \ldots, EC_N)$
$= (Enc(M_1), \ldots, Enc(M_N))$

$\xleftrightarrow{\text{Transfer}}$

$\perp$

M_ρ

- 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-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.

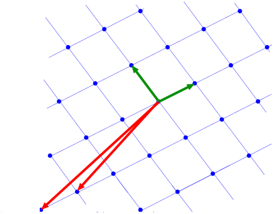
$$\Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i \leq n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \right\}$$

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.

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



Why?

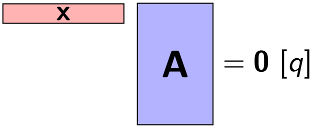
- Simple and asymptotically efficient;
- **Still** conjectured quantum-resistant;
- Connection between average-case and worst-case problems;
- Powerful functionalities (e.g., FHE).

→ 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 $\boxed{\mathbf{A}} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$:

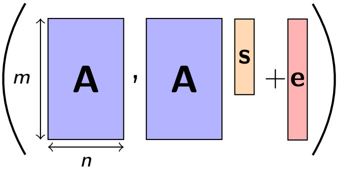
Small Integer Solution



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

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

Learning With Errors

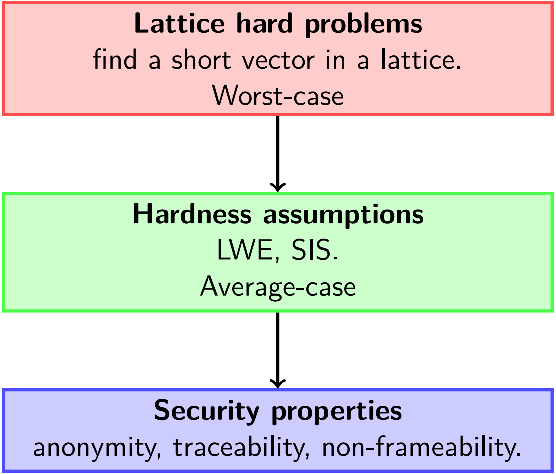


$$\left(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}\right)$$

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

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

Provable Security



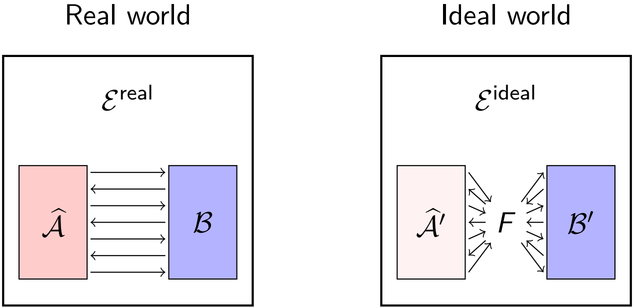
Lattice hard problems
find a short vector in a lattice.
Worst-case

Hardness assumptions
LWE, SIS.
Average-case

Security properties
anonymity, traceability, non-frameability.

Full Simulation Model [Can01]

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



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

- Strictly stronger security model than indistinguishability-based one

Outline

Introduction

Building Blocks

Adaptive Oblivious Transfer with Access Control

Assisted Decryption Technique [CNs07]

Transfer_i

Sender		**Receiver**
		Sample μ at random
		$C_i \leftarrow \text{Rerand}(EC_{\rho_i}, \mu)$
		$= \text{Enc}(M_{\rho_i} \oplus \mu)$
	$\xleftarrow{C_i}$	

Assisted Decryption Technique [CNs07]

Transfer$_i$

Sender		**Receiver**
		Sample μ at random
		$C_i \leftarrow \text{Rerand}(EC_{\rho_i}, \mu)$
		$= \text{Enc}(M_{\rho_i} \oplus \mu)$
	$\xleftarrow{C_i}$	
$\widetilde{M}_i \leftarrow \text{Dec}(C_i)$		
	$\xrightarrow{\widetilde{M}_i}$	
		$M_{\rho_i} = \widetilde{M}_i \oplus \mu$

Assisted Decryption Technique [CNs07]

Transfer$_i$

Sender		**Receiver**
		Sample μ at random
		$C_i \leftarrow \mathrm{Rerand}(EC_{\rho_i}, \mu)$
		$= \mathrm{Enc}(M_{\rho_i} \oplus \mu)$
	$\xleftarrow{C_i}$	
$\widetilde{M}_i \leftarrow \mathrm{Dec}(C_i)$		
	$\xrightarrow{\widetilde{M}_i}$	
		$M_{\rho_i} = \widetilde{M}_i \oplus \mu$

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

Regev's Encryption [Reg05]

Keygen:

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

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

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

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

Regev's Encryption [Reg05]

Keygen:

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

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

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

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

+ ZK proofs

Smudging [AJL+12]

Problem

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

Smudging [AJL+12]

Problem

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

Attack scenario:

- $\mathrm{DB} = (M_0, M_1)$
- 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

Smudging [AJL+12]

Problem

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

Attack scenario:

- $\mathrm{DB} = (M_0, M_1)$
- 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: $(\mathbf{a}', \mathbf{b}') = (\mathbf{a} + \boxed{\mathbf{F}}\ \mathbf{e}, \mathbf{b} + \boxed{\mathbf{P}}^T \mathbf{e} + \mu \lfloor \frac{q}{2} \rfloor + \nu)$

Access Control [CDN09]

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. $AP_i(x) = 1$.

Access Control [CDN09]

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: $x_1 \wedge \ldots \wedge x_\ell$.
Disjunctions possible through replication.

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

Access Control [CDN09]

AC-OT

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

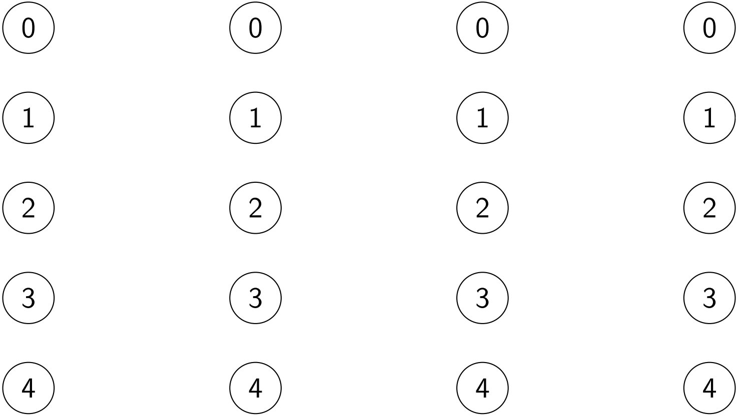
Receiver can retrieve message M_i iff it possesses cert_x for some $x \in \{0,1\}^*$ s.t. $AP_i(x) = 1$.

[ACDN13]: access policy made of conjunctions: $x_1 \wedge \ldots \wedge x_\ell$.
Disjunctions possible through replication.

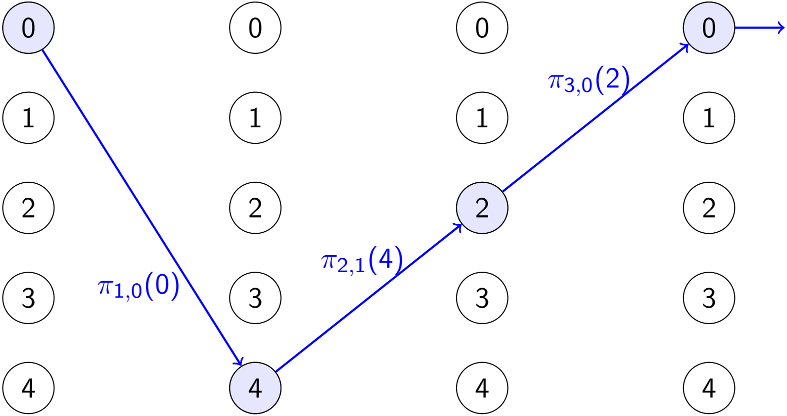
[ZAW+10]: use CP-ABE to handle NC^1 access policies.

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

Branching Programs

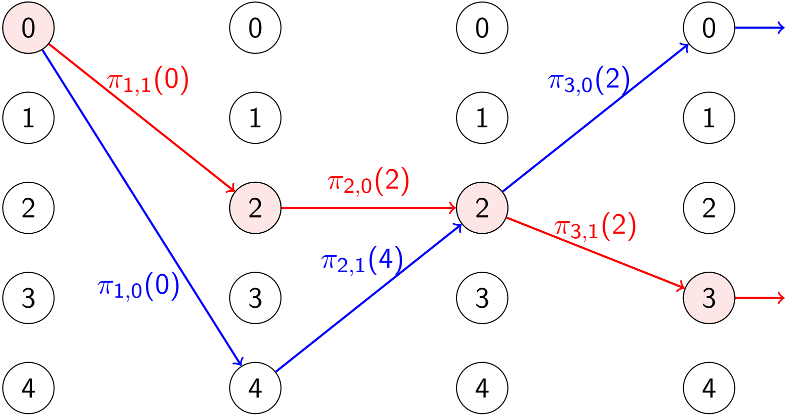


Branching Programs



$x = 010$: accepted

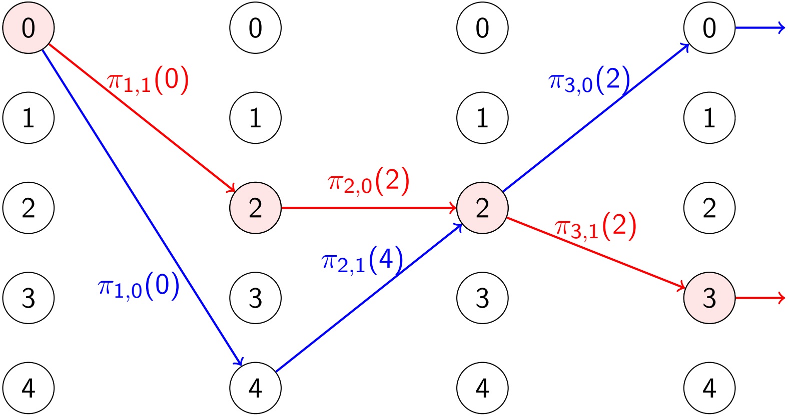
Branching Programs



$x = 010$: accepted

$y = 101$: rejected

Branching Programs



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

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

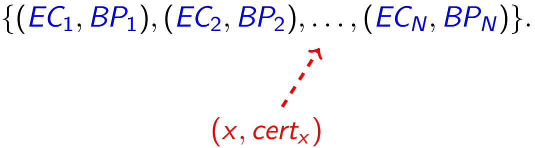
Branching Programs

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

$$(x, cert_x)$$

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

Branching Programs



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

$(x, cert_x)$

Goal: Prove knowledge of $cert_x = Sign(x)$ 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:

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

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

Step-by-step evaluation:

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

Proving correct evaluation

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

Branching Programs

Encoding of a branching program:

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

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

Step-by-step evaluation:

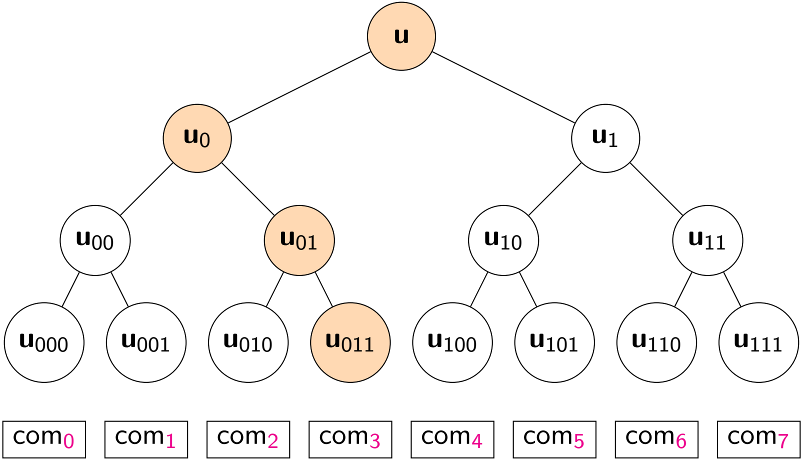
$$\eta_\theta = \pi_{\theta, x_{\mathrm{var}(\theta)}}(\eta_{\theta-1}) = \pi_{\theta,0}(\eta_{\theta-1}) \cdot \bar{x}_{\mathrm{var}(\theta)} + \pi_{\theta,1}(\eta_{\theta-1}) \cdot x_{\mathrm{var}(\theta)}$$

Proving correct evaluation

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

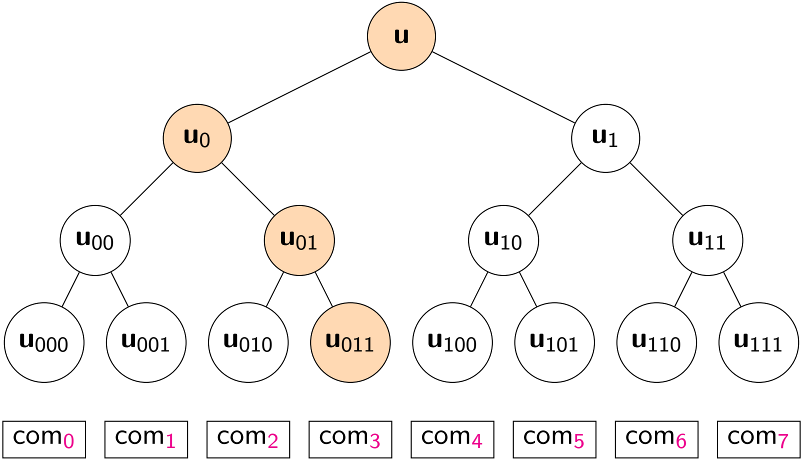
Our idea: use binary-search → $O(L \cdot \log(\kappa))$

Branching Programs



$\mathsf{com}_i = \mathsf{Com}(x_i)$

Branching Programs



$\mathsf{com}_i = \mathsf{Com}(x_i)$ → correct binary search ⇒ knowledge of $x_{\mathrm{var}(\theta)}$

Commitments

Digital equivalent of a sealed box.

e.g., Pedersen Commitment
$pk = (g, h) \hookleftarrow \mathbb{G}^2$
$com = g^m \cdot h^r$
$open = (m, r)$

Commitments

Digital equivalent of a sealed box.

e.g., Pedersen Commitment

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

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

$open = (m, r)$

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.

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)

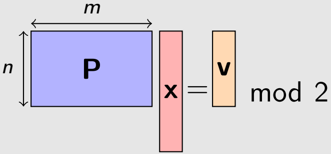
Stern's protocol is a ZK proof for Syndrome Decoding Problem.

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. $\mathrm{w}(\mathbf{x}) = w$ and

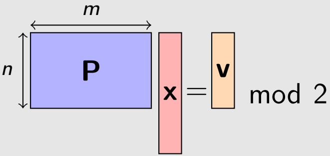


Stern's Protocol (Crypto'93)

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. $\mathrm{w}(\mathbf{x}) = w$ and



[KTX08]: mod 2 → mod q

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

Recent uses of Stern-like protocols in lattice-based crypto:
[LNW15, LLNW16, LLN**M**W16]

Signature with Efficient Protocols [CL02]

A signature scheme $(\mathbf{Keygen}, \mathbf{Sign}_{\mathsf{sk}}, \mathbf{Verif}_{\mathsf{vk}})$ with companion protocols:

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

Signature with Efficient Protocols [CL02]

A signature scheme $(\mathbf{Keygen}, \mathbf{Sign}_{\mathsf{sk}}, \mathbf{Verif}_{\mathsf{vk}})$ with companion protocols:

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

Security

- Unforgeability;

Signature with Efficient Protocols [CL02]

A signature scheme ($\textbf{Keygen}, \textbf{Sign}_{\mathsf{sk}}, \textbf{Verif}_{\mathsf{vk}}$) 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 $(\mathbf{Keygen}, \mathbf{Sign}_{\mathsf{sk}}, \mathbf{Verif}_{\mathsf{vk}})$ 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 $(\mathbf{Keygen}, \mathbf{Sign}_{\mathsf{sk}}, \mathbf{Verif}_{\mathsf{vk}})$ 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 [LLN**M**W16].

Outline

Introduction

Building Blocks

Adaptive Oblivious Transfer with Access Control

Our Construction

Ingredients

- The assisted decryption technique
- A simplification of [LLN**M**W16]'s signatures as certificates
- Access control using BP
- WI proofs *à la* Stern

Our Adaptive Oblivious Transfer Construction

Initialization

Sender side:

1. Generate $(\mathrm{VK}_{sig}, \mathrm{SK}_{sig}) \leftarrow \Sigma.keygen(1^\lambda)$
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 $(\mathsf{VK}_{sig}, \mathsf{SK}_{sig}) \leftarrow \Sigma.keygen(1^\lambda)$
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 $(\mathsf{VK}_{sig}, (\mathbf{F}, \mathbf{P}), EC_i)$ 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

Sender

Receiver(ρ_i)

Sample $\mu \leftarrow U(\{0,1\}^t)$
Rerandomize $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i})$
into $(\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{a}_{\rho_i} + \mathbf{F}\,\mathbf{e},$
$\mathbf{b}_{\rho_i} + \mathbf{P}^T\mathbf{e} + \mu\lfloor q/2 \rfloor + \nu)$

$$\xleftarrow{\quad (\mathbf{c}_0, \mathbf{c}_1) \quad}$$

$$\xleftrightarrow{\text{prove knowledge of } \sigma_{\rho_i}}$$

$\widetilde{M}_i = \mathrm{Dec}(\mathbf{c}_0, \mathbf{c}_1)$

$$\xrightarrow{\quad \widetilde{M}_i \quad}$$

$$\xleftrightarrow{\text{prove correct decryption}}$$

$M_{\rho_i} = \widetilde{M}_i \oplus \mu$

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 $x_{i,\cdot}$ 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?