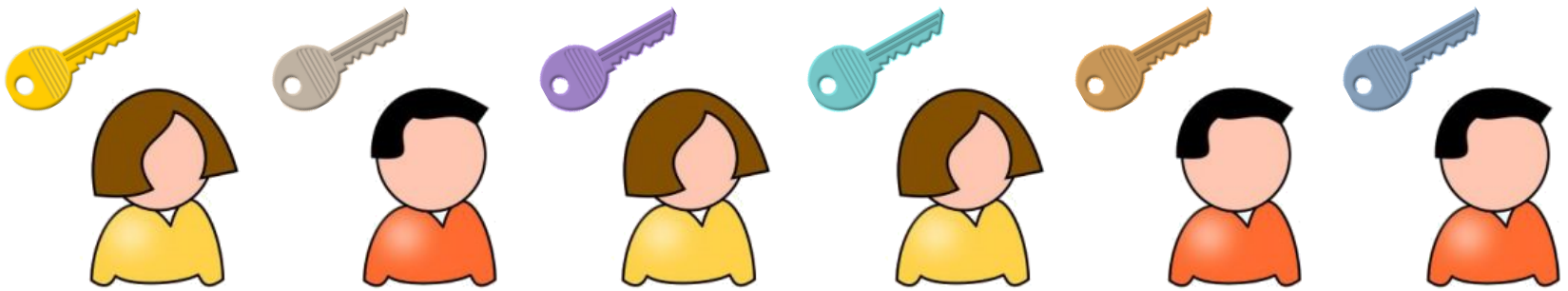
Distributed Broadcast Encryption from Lattices

Jeffrey Champion and David Wu

Broadcast Encryption

[FN93]



Broadcast Encryption

[FN93]

message m

$S = \{1,3,6\}$

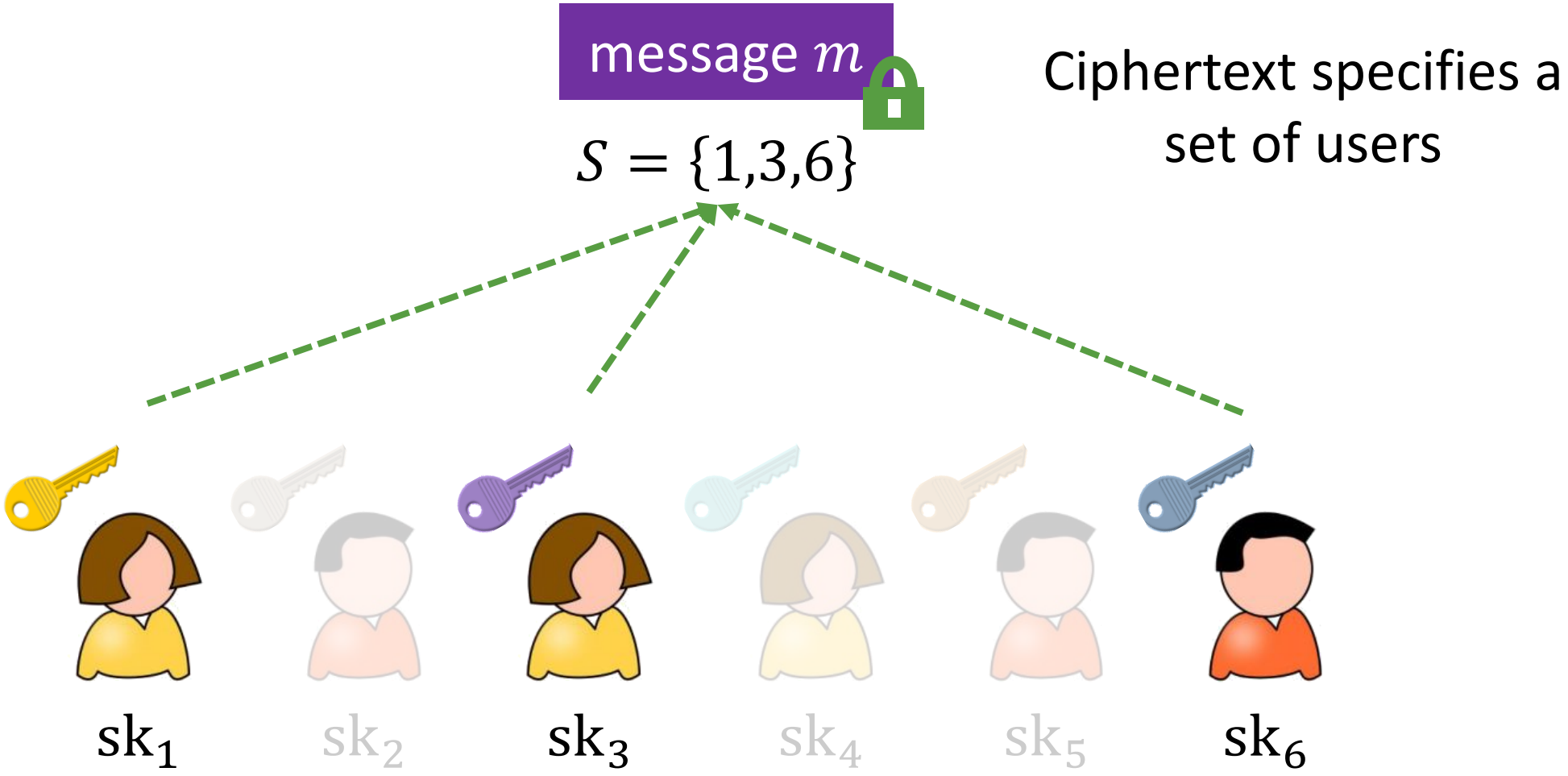
Ciphertext specifies a set of users

sk_1 sk_2 sk_3 sk_4 sk_5 sk_6

Broadcast Encryption

[FN93]

Functionality: Users in the set can decrypt

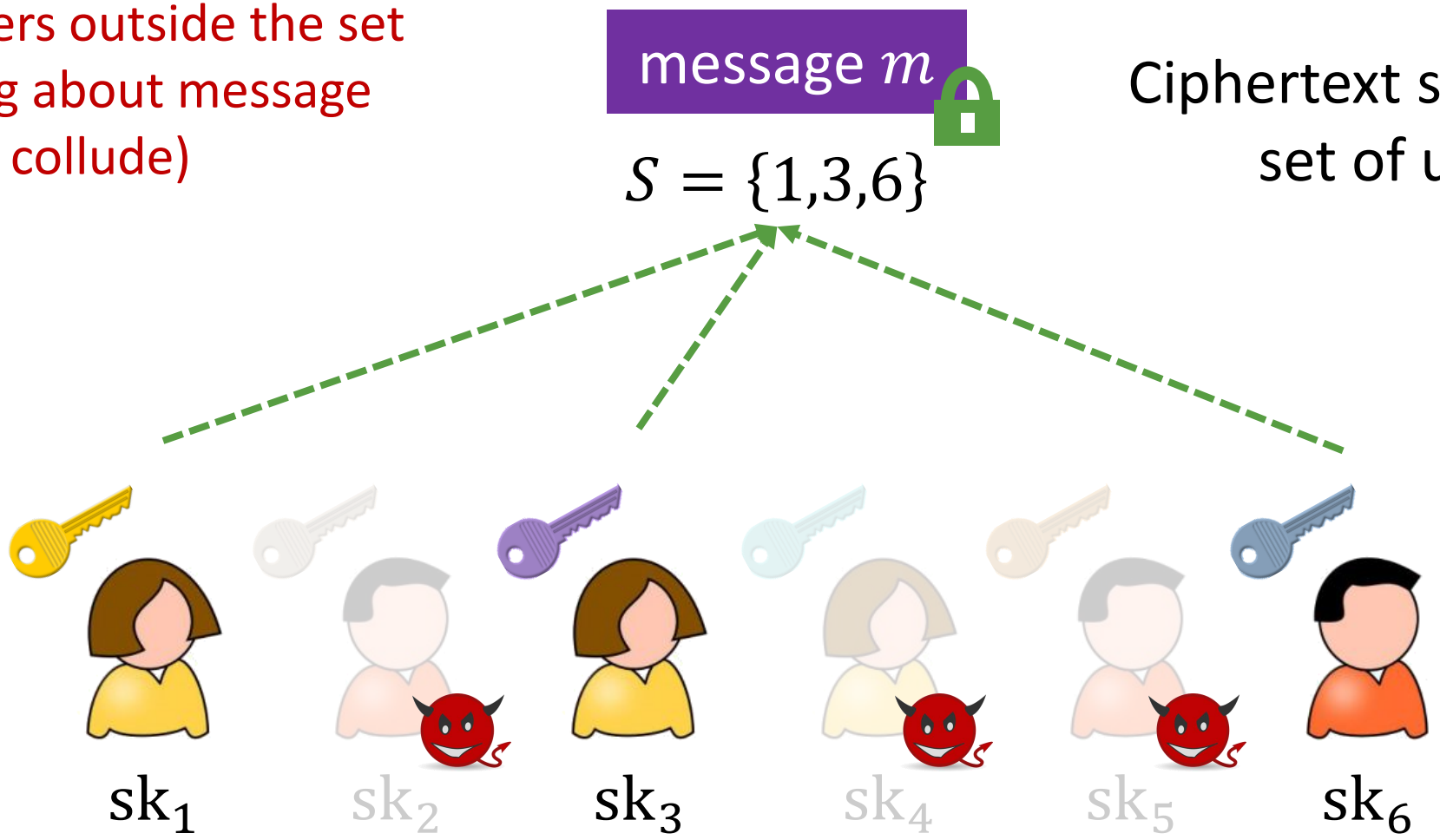


Broadcast Encryption

[FN93]

Functionality: Users in the set can decrypt

Security: Users outside the set learn nothing about message (even if they collude)



Broadcast Encryption

[FN93]

Functionality: Users in the set can decrypt

Security: Users outside the set learn nothing about message (even if they collude)

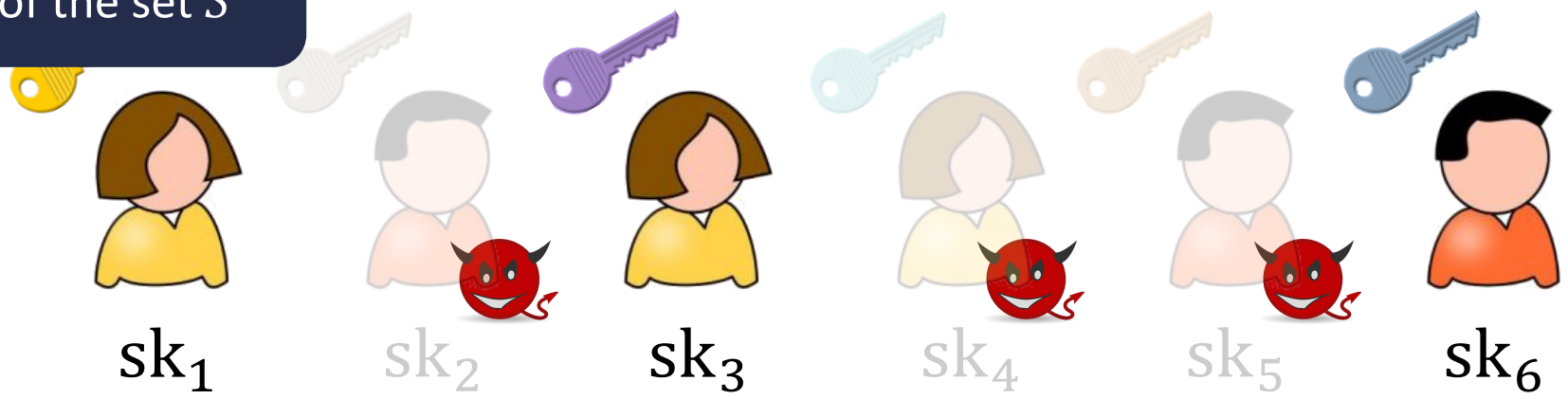
Efficiency: $|\mathrm{ct}| = |m| + \mathrm{poly}(\lambda, \log|S|)$

message m

$S = \{1,3,6\}$

Ciphertext specifies a set of users

Note: decryption requires knowledge of the set S

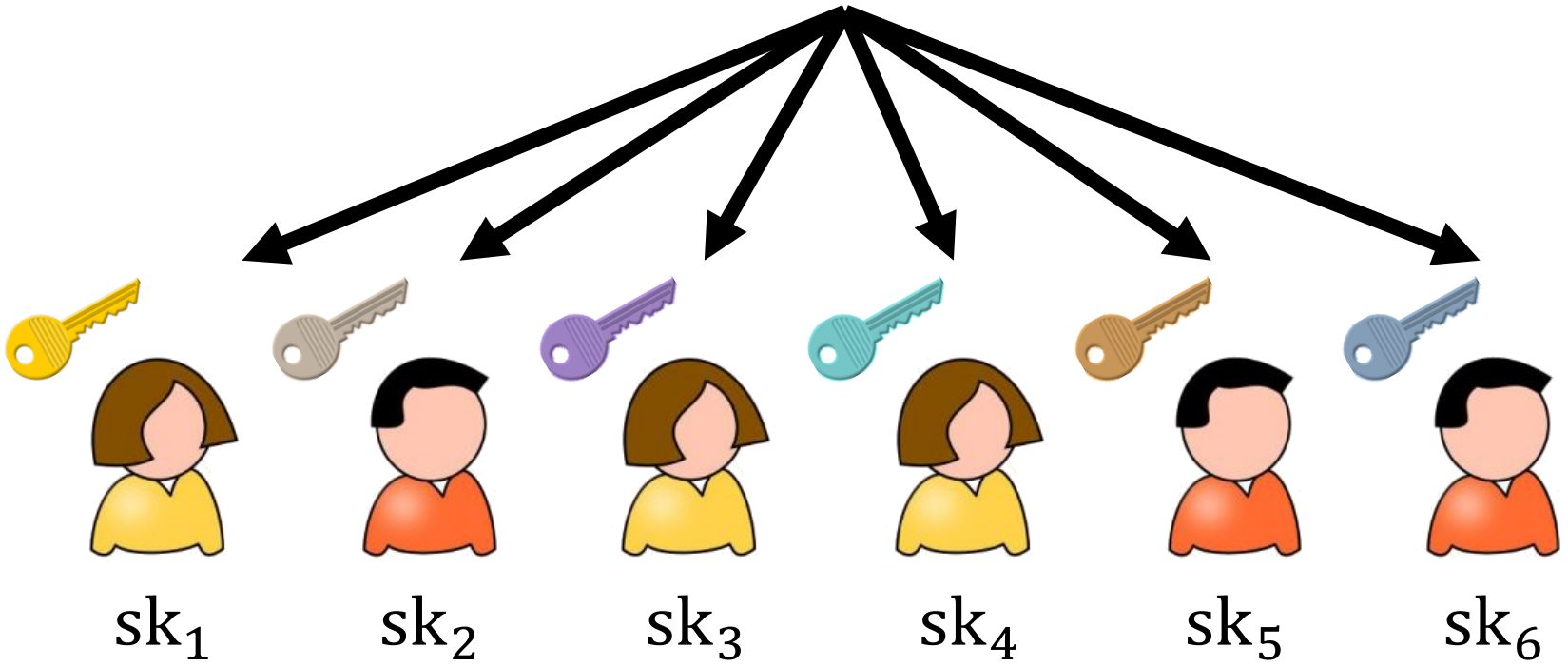


Broadcast Encryption

[FN93]

Central **trusted** authority generates keys

master secret key



Broadcast Encryption

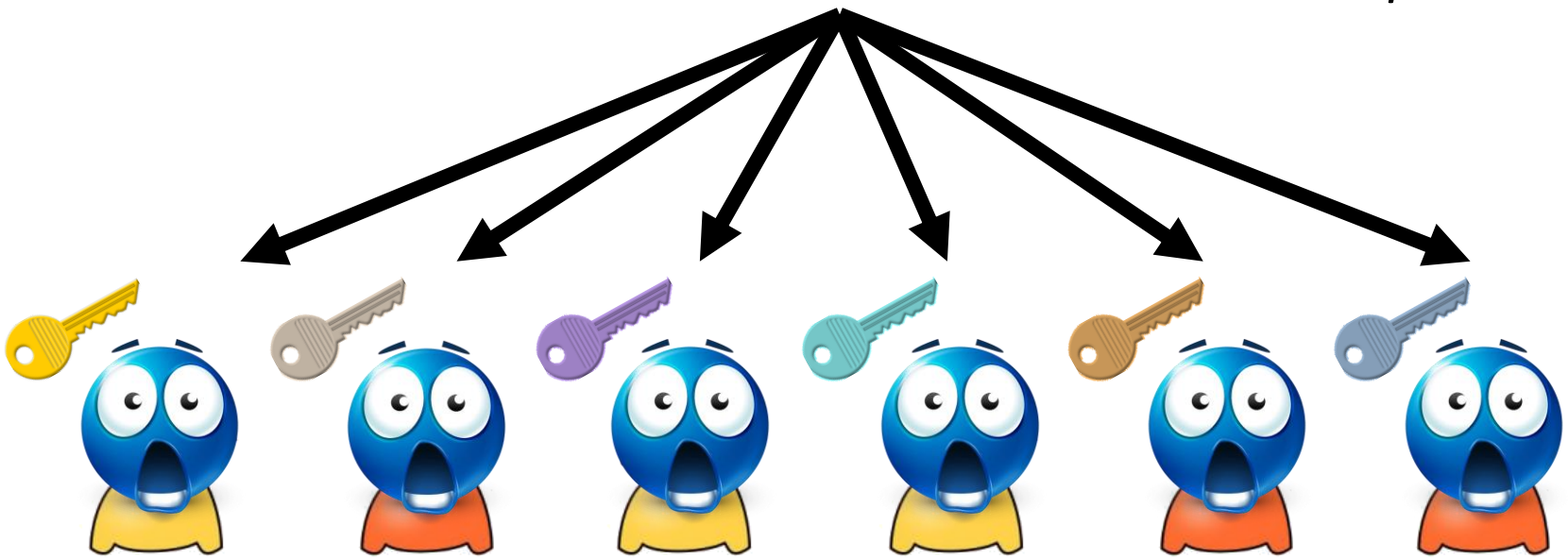
[FN93]

Central **trusted** authority generates keys

Central point of failure

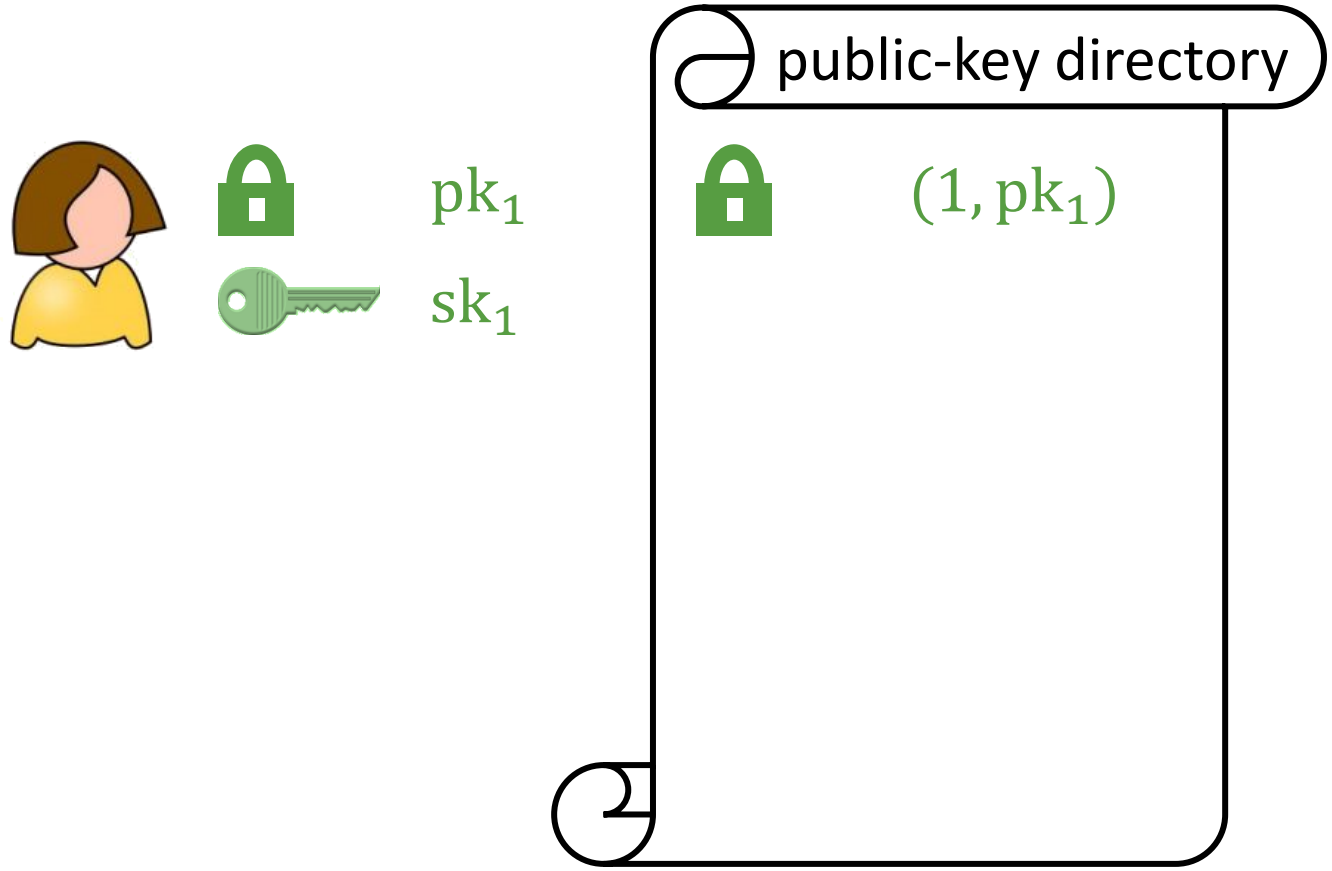
master secret key

What if the key issuer is compromised?



Distributed Broadcast Encryption (DBE)

[BZ14]

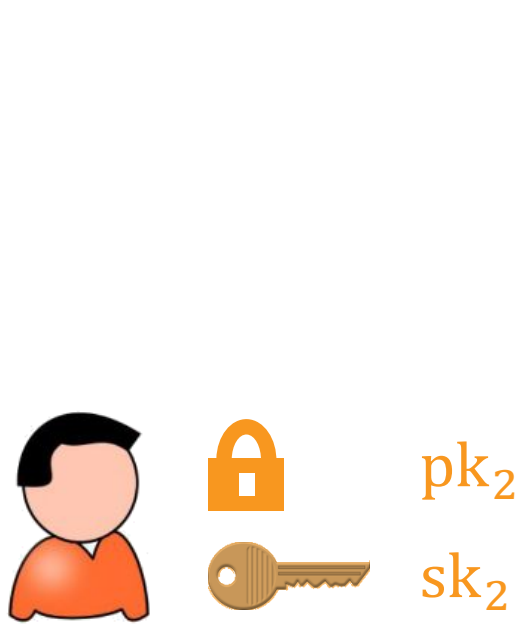


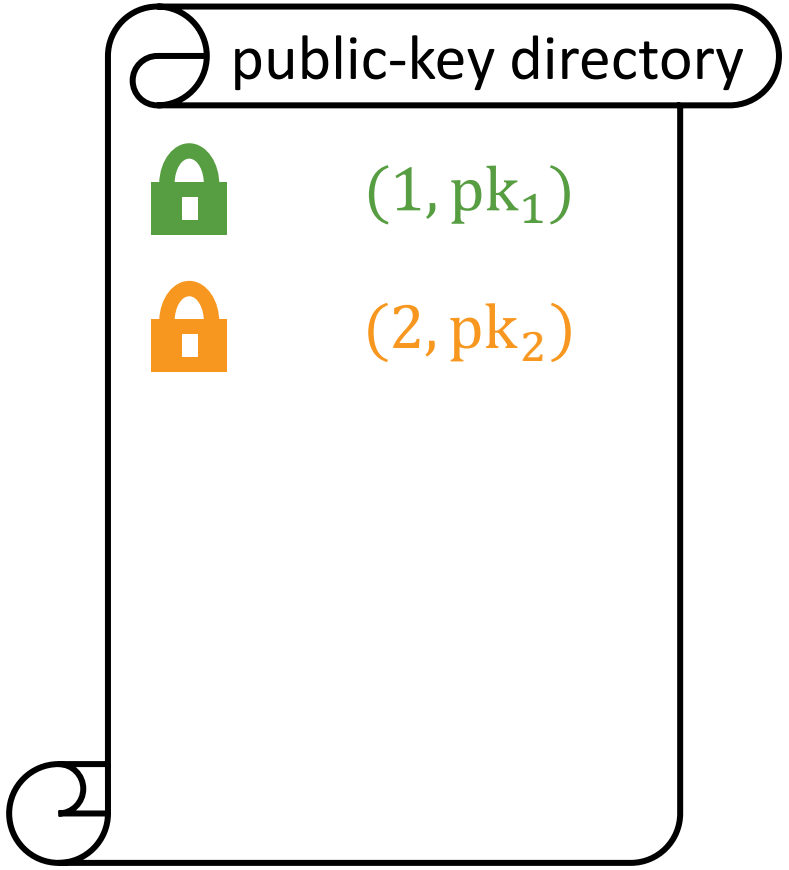
Users generate public/private keys independently (as in public-key encryption)

Broadcast encryption without a central authority

Distributed Broadcast Encryption (DBE)

[BZ14]



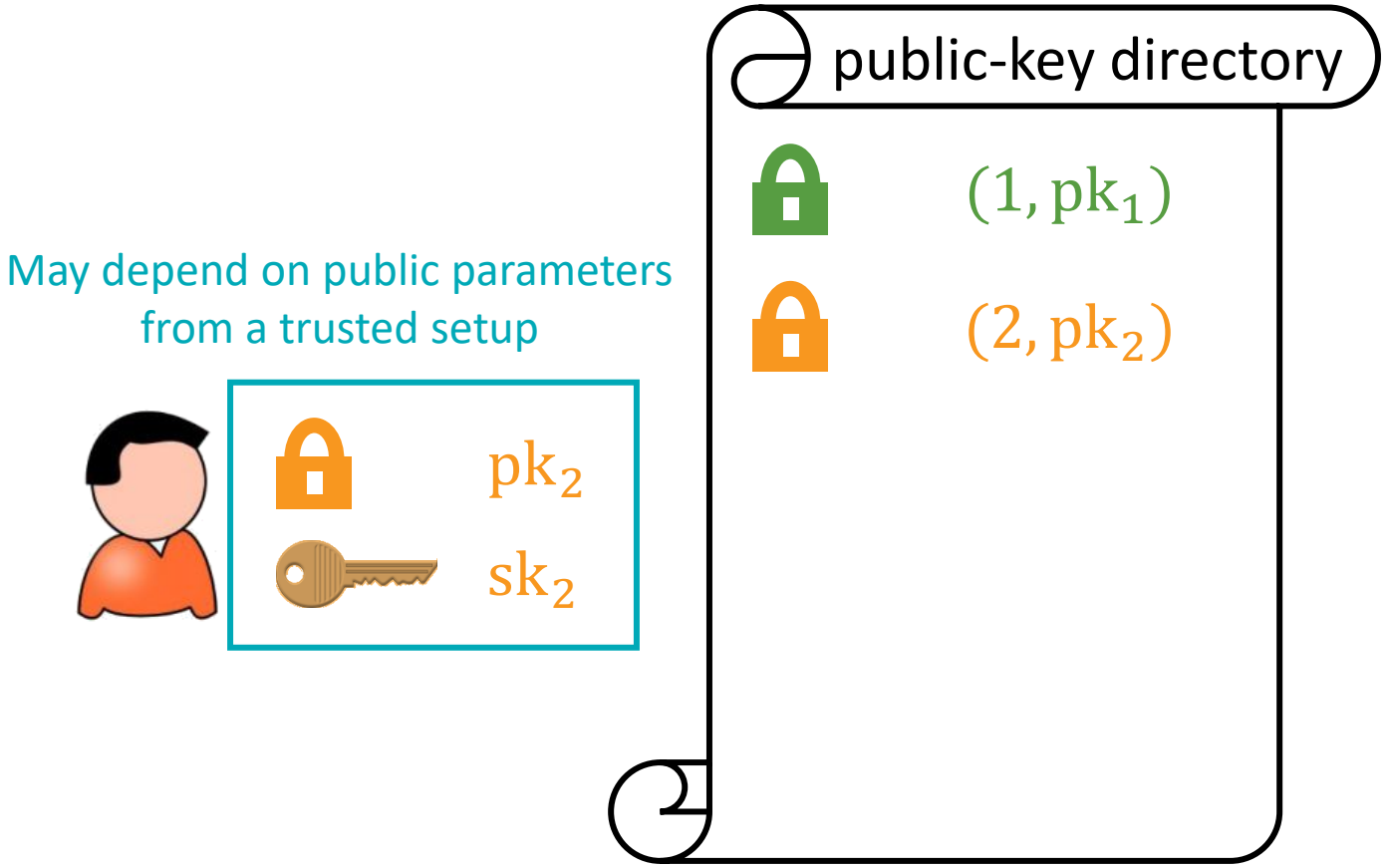


Users generate public/private keys independently (as in public-key encryption)

Broadcast encryption without a central authority

Distributed Broadcast Encryption (DBE)

[BZ14]

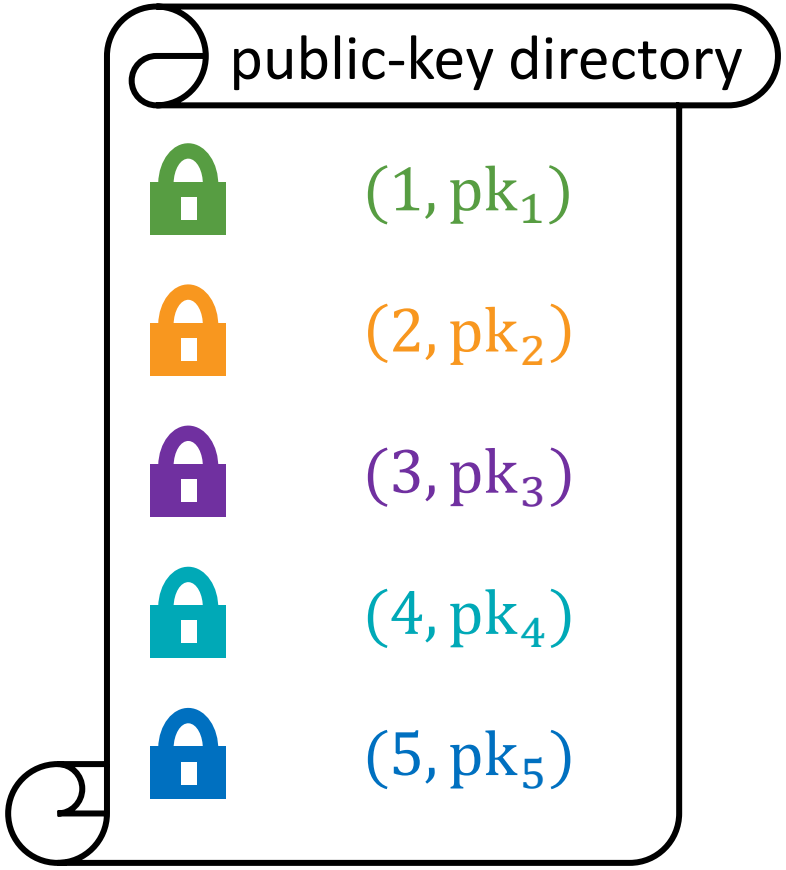


Users generate public/private keys independently (as in public-key encryption)

Broadcast encryption without a central authority

Distributed Broadcast Encryption (DBE)

[BZ14]



public parameters

$$\mathrm{Encrypt}(\mathrm{pp}, \{\mathrm{pk}_i\}_{i \in S}, m) \to \mathrm{ct}$$

$$\mathrm{Decrypt}(\mathrm{pp}, \{\mathrm{pk}_i\}_{i \in S}, \mathrm{sk}, \mathrm{ct}) \to m$$

Efficiency: $|\mathrm{ct}| = |m| + \mathrm{poly}(\lambda, \log|S|)$

Security: Users outside the set learn nothing about message (even if they collude)

Broadcast encryption without a central authority

Constructions of DBE

- Indistinguishability obfuscation (and OWF) [BZ14]
- Witness encryption (and leveled HE) [FWW23]
- Registered attribute-based encryption [FWW23]
- Pairing-based assumptions (BDHE or k-Lin) [KMW23, GKPW24]

Constructions of DBE

- Indistinguishability obfuscation (and OWF) [BZ14]
- Witness encryption (and leveled HE) [FWW23]
- Registered attribute-based encryption [FWW23]
- Pairing-based assumptions (BDHE or k-Lin) [KMW23, GKPW24]

Constructions from lattice assumptions?

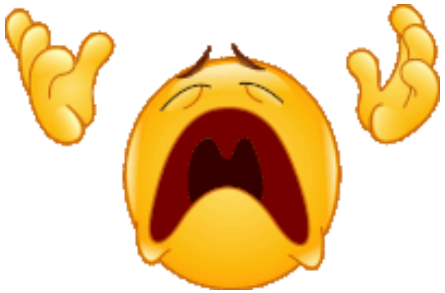
Lattice-Based Distributed Broadcast

LWE?
[Reg05]

Lattice-Based Distributed Broadcast

LWE?
[Reg05]

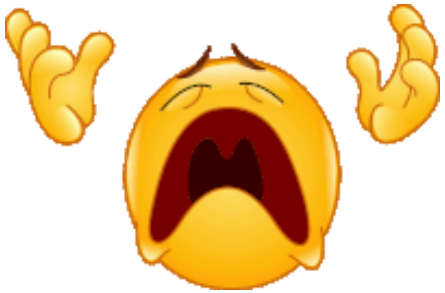
No centralized broadcast after ~20 years



Lattice-Based Distributed Broadcast

LWE?
[Reg05]

No centralized broadcast after ~20 years



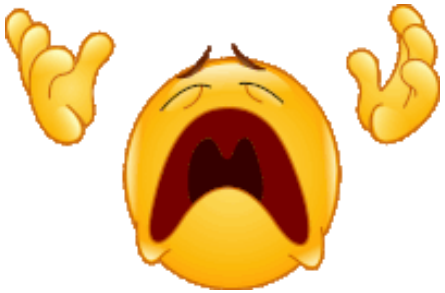
Evasive LWE:
[Tsa22, Wee22]

Public-coin: centralized broadcast [Wee22]
Private-coin: DBE (via WE [Tsa22, VWW22])

Lattice-Based Distributed Broadcast

LWE?
[Reg05]

No centralized broadcast after ~20 years

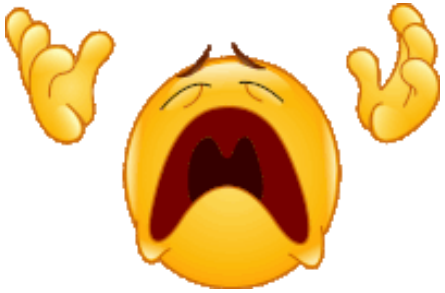


Evasive LWE:
[Tsa22, Wee22]

Public-coin: centralized broadcast [Wee22]

Private-coin: DBE (via WE [Tsa22, VWW22])

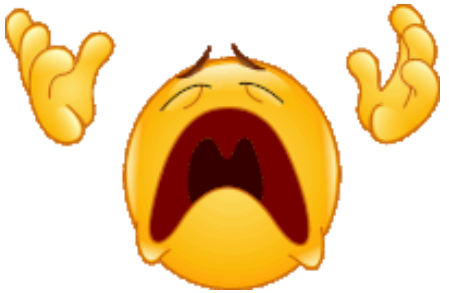
Assumption broken and partly patched [BÜW24]



Lattice-Based Distributed Broadcast

LWE?
[Reg05]

No centralized broadcast after ~20 years

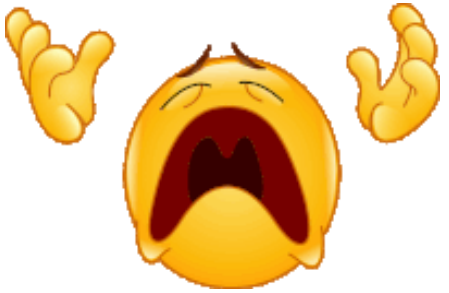


Evasive LWE:
[Tsa22, Wee22]

Public-coin: centralized broadcast [Wee22]

Private-coin: DBE (via WE [Tsa22, VWW22])

Assumption broken and partly patched [BÜW24]



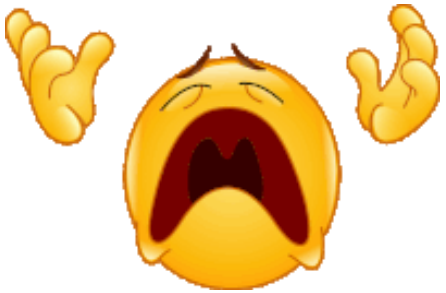
ℓ-succinct LWE:
[Wee24]

Centralized broadcast via succinct ABE [Wee24]

Lattice-Based Distributed Broadcast

LWE?
[Reg05]

No centralized broadcast after ~20 years

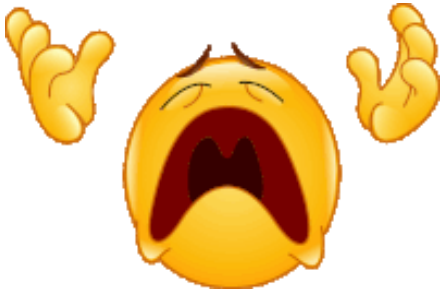


Evasive LWE:
[Tsa22, Wee22]

Public-coin: centralized broadcast [Wee22]

Private-coin: DBE (via WE [Tsa22, VWW22])

Assumption broken and partly patched [BÜW24]



ℓ-succinct LWE:
[Wee24]

Centralized broadcast via succinct ABE [Wee24]

This work: DBE from ℓ-succinct LWE

ℓ-Succinct LWE Assumption

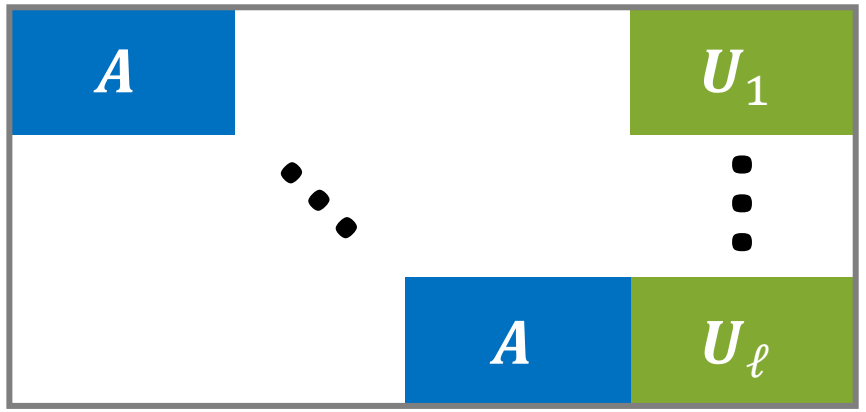
[Wee24]

LWE is hard with respect to $\boldsymbol{A}$ given a "fresh" trapdoor for a related matrix $\boldsymbol{D}_\ell$:

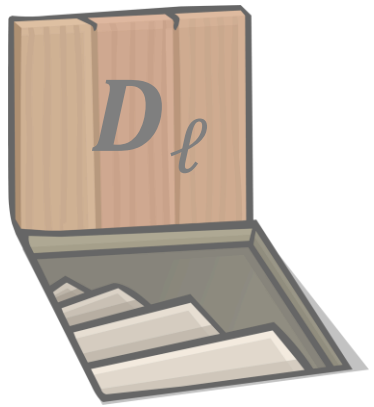
$$\boldsymbol{s}^T \boldsymbol{A} + \boldsymbol{e}^T \approx \boldsymbol{z}^T$$

$\boldsymbol{A}, \boldsymbol{U_i}, \boldsymbol{s}, \boldsymbol{z}$ are uniform

given $\boldsymbol{D}_\ell =$



and



ℓ-Succinct LWE Assumption

[Wee24]

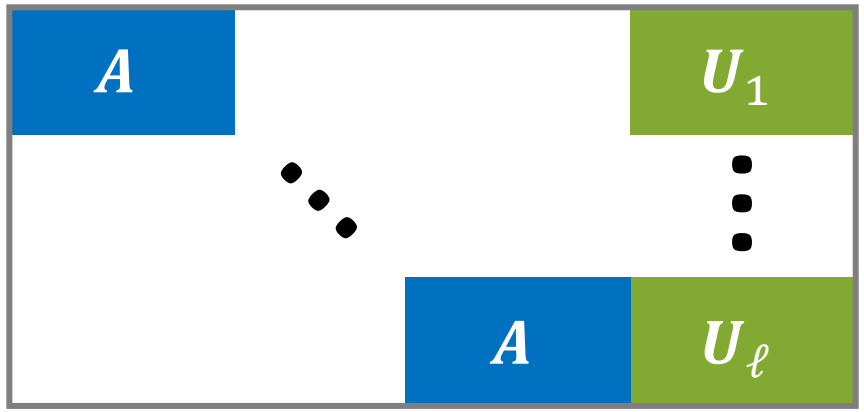
LWE is hard with respect to $\boldsymbol{A}$ given a "fresh" trapdoor for a related matrix $\boldsymbol{D}_\ell$:

$$\boldsymbol{s}^T \boldsymbol{A} + \boldsymbol{e}^T \approx \boldsymbol{z}^T$$

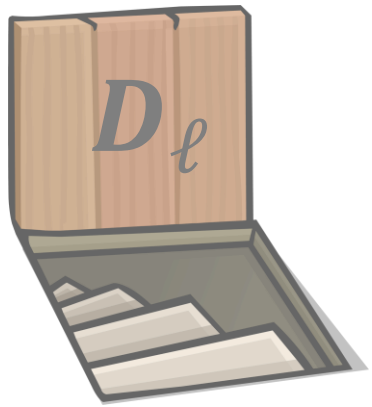
$\boldsymbol{A}, \boldsymbol{U_i}, \boldsymbol{s}, \boldsymbol{z}$ are uniform

Implied by LWE if width of $\boldsymbol{U_i}$ scales with ℓ and by public-coin evasive LWE when $\boldsymbol{U_i}$ are narrow

given $\boldsymbol{D}_\ell =$



and



ℓ-Succinct LWE Assumption

[Wee24]

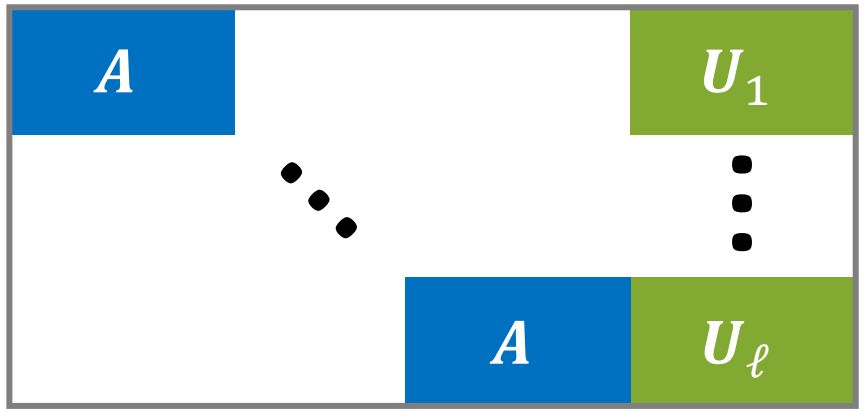
LWE is hard with respect to $\boldsymbol{A}$ given a "fresh" trapdoor for a related matrix $\boldsymbol{D}_\ell$:

$$\boldsymbol{s}^T \boldsymbol{A} + \boldsymbol{e}^T \approx \boldsymbol{z}^T$$

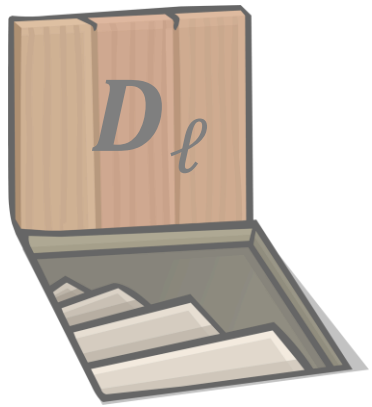
$\boldsymbol{A}, \boldsymbol{U_i}, \boldsymbol{s}, \boldsymbol{z}$ are uniform

Implied by LWE if width of $\boldsymbol{U_i}$ scales with ℓ and by public-coin evasive LWE when $\boldsymbol{U_i}$ are narrow

given $\boldsymbol{D}_\ell =$



and



Falsifiable and instance-independent unlike evasive LWE

ℓ-Succinct LWE Assumption

[Wee24]

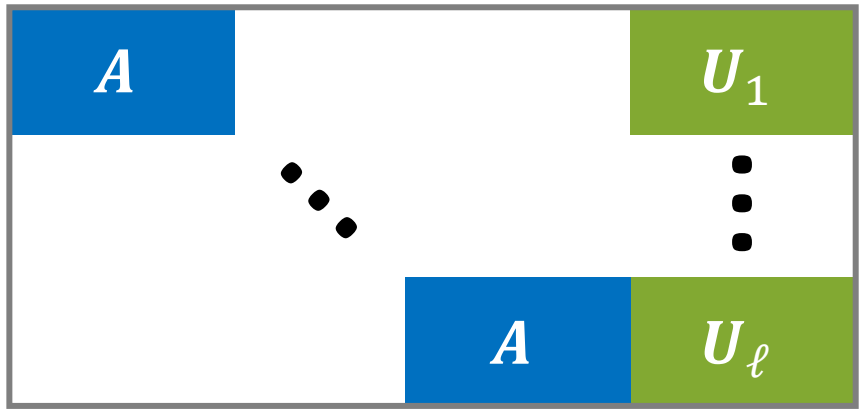
LWE is hard with respect to $\boldsymbol{A}$ given a "fresh" trapdoor for a related matrix $\boldsymbol{D}_\ell$:

$$\boldsymbol{s}^T \boldsymbol{A} + \boldsymbol{e}^T \approx \boldsymbol{z}^T$$

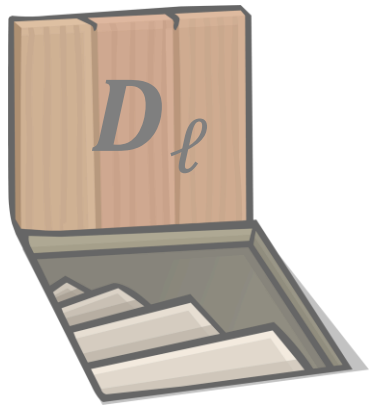
$\boldsymbol{A}, \boldsymbol{U_i}, \boldsymbol{s}, \boldsymbol{z}$ are uniform

Implied by LWE if width of $\boldsymbol{U_i}$ scales with ℓ and by public-coin evasive LWE when $\boldsymbol{U_i}$ are narrow

given $\boldsymbol{D}_\ell =$



and

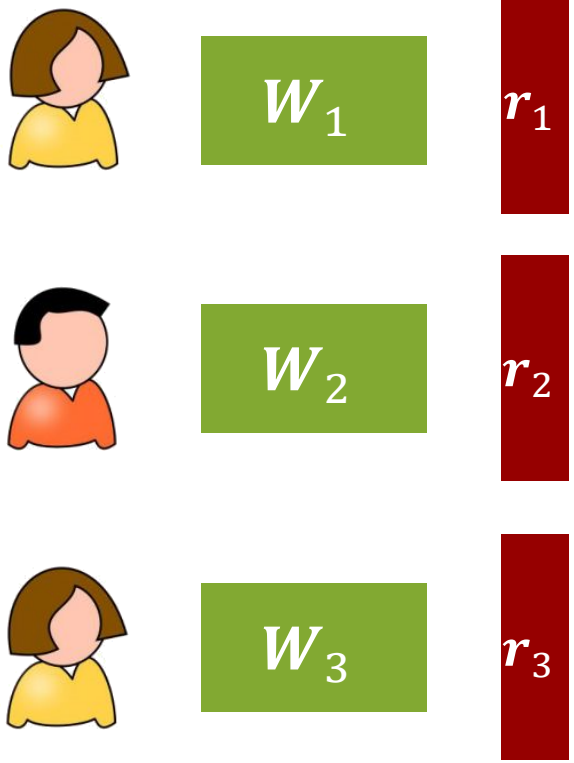


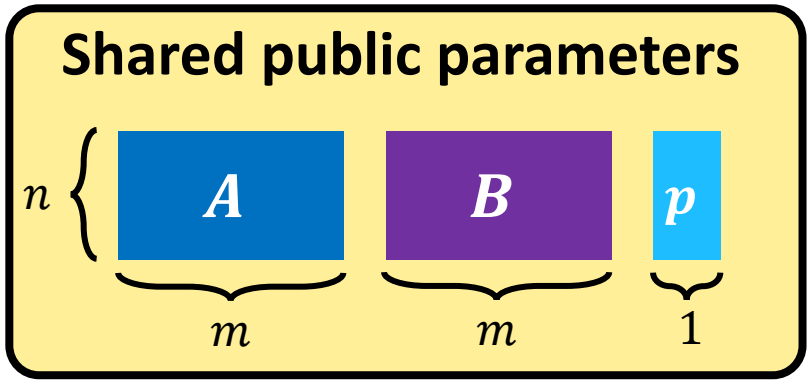
Falsifiable and instance-independent unlike evasive LWE

Also yields ABE with short ciphertexts [Wee24] and functional commitments [WW23]

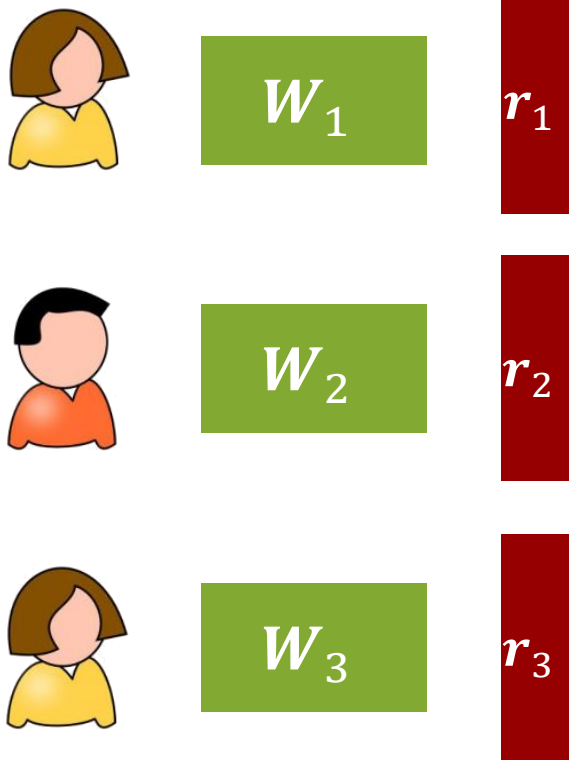
Starting Point: Centralized Broadcast Encryption

Starting Point: Centralized Broadcast Encryption





Starting Point: Centralized Broadcast Encryption



Shared public parameters

n { A (m) B (m) p (1)

Ciphertext encrypting a bit $b \in \{0,1\}$ to a set $S \subseteq [\ell]$:

$s^T A$ $\quad$ $s^T p$ (masks b) — plain public-key encryption terms

$$s^T \left(B + \sum_{i \in S} W_i \right)$$ — broadcast term

Noise omitted

Starting Point: Centralized Broadcast Encryption

Public parameters

$\boldsymbol{A}$ $\boldsymbol{W}_i$

$\boldsymbol{B}$ $\boldsymbol{r}_i$ $i \in [\ell]$

$\boldsymbol{p}$

Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

$$\boldsymbol{s}^{\mathrm{T}} \boldsymbol{A} \qquad \underbrace{\boldsymbol{s}^{\mathrm{T}} \boldsymbol{p}}_{\text{masks } b}$$

$$\boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j \right)$$

Starting Point: Centralized Broadcast Encryption

Public parameters

A W_i B r_i $i \in [\ell]$ p

Goal: user $i \in S$ should be able to uniquely compute s^T p

Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

s^T A $\quad$ s^T p (masks b)

$$s^T \left(B + \sum_{j \in S} W_j \right)$$

Starting Point: Centralized Broadcast Encryption

Public parameters

A W_i B r_i $i \in [\ell]$ p

Goal: user $i \in S$ should be able to uniquely compute $s^T p$

Secret key for user i: should derive a short vector y_S such that

$$A\, y_S = p + B\, r_i + \sum_{j \in S} W_j\, r_i$$

Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

$s^T A$ $\quad$ $s^T p$ (masks b)

$$s^T \left(B + \sum_{j \in S} W_j \right)$$

Starting Point: Centralized Broadcast Encryption

Public parameters

A W_i B r_i $i \in [\ell]$ p

Goal: user $i \in S$ should be able to uniquely compute $s^T p$

Secret key for user i: should derive a short vector y_S such that

$$A\, y_S = p + B\, r_i + \sum_{j \in S} W_j\, r_i$$

Decryption: User i can now compute $s^T p$ by the subtraction

$$s^T A\, y_S - s^T \left(B + \sum_{j \in S} W_j \right) r_i$$

Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

$s^T A$ $\quad$ $\underbrace{s^T p}_{\text{masks } b}$

$$s^T \left(B + \sum_{j \in S} W_j \right)$$

Requires r_i be short when noise is included

Starting Point: Centralized Broadcast Encryption

Public parameters

A W_i B r_i $i \in [\ell]$ p

Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

s^T A s^T p

masks b

$s^T \left(B + \sum_{j \in S} W_j \right)$

Goal: user $i \in S$ should be able to uniquely compute $s^T p$

Secret key for user i: short vectors $y_{j,i}$ such that

$$A\, y_{i,i} = p + B\, r_i + W_i\, r_i$$

$$A\, y_{j,i} = W_j\, r_i \qquad j \in [\ell], j \neq i$$

When $i \in S$, derive $y_S = \sum_{j \in S} y_{j,i}$

Simplifying Secret Keys

Public parameters

A W_i B r_i $i \in [\ell]$ p

Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

s^T A $\quad$ s^T p

masks b

$$s^T \left(B + \sum_{j \in S} W_j \right)$$

Secret key for user i: short vectors $y_{j,i}$ such that

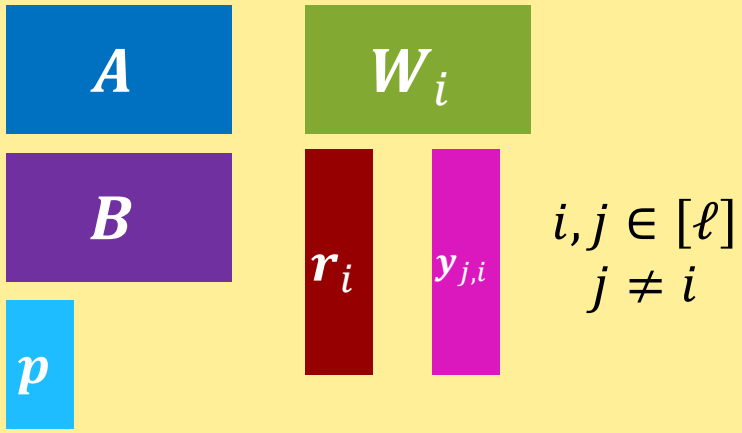
$$A\, y_{i,i} = p + B\, r_i + W_i\, r_i$$

$$A\, y_{j,i} = W_j\, r_i \qquad j \in [\ell], j \neq i$$

Does not map A to p !

Simplifying Secret Keys

Public parameters



Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

$$s^T \left(B + \sum_{j \in S} W_j \right)$$

Secret key for user i: short vector $y_{i,i}$ such that

$$A \, y_{i,i} = p + B \, r_i + W_i \, r_i$$

Simplifying Secret Keys

Public parameters

A W_i B r_i $y_{j,i}$ $i, j \in [\ell]$ $j \neq i$ p

Secret key for user i: short vector $y_{i,i}$ such that

$$A \, y_{i,i} = p + B \, r_i + W_i \, r_i$$

Encryption of $b \in \{0,1\}$ to $S \subseteq [\ell]$:

$s^T A$ $\quad$ $s^T p$ (masks b)

$$s^T \left(B + \sum_{j \in S} W_j \right)$$

This is a **centralized** broadcast encryption scheme

Sampling $y_{i,j}$ requires knowledge of the trapdoor for A

Distributed Key Generation

Challenge: No one can know a trapdoor for A

Distributed Key Generation

Challenge: No one can know a trapdoor for $\boldsymbol{A}$

Approach: User i will generate $\boldsymbol{W}_i$ and short $\boldsymbol{y}_{i,j}$ given public parameters

Distributed Key Generation

Challenge: No one can know a trapdoor for A

Approach: User i will generate W_i and short $y_{i,j}$ given public parameters

Secret key for user i: short vector $y_{i,i}$ such that

$$A \, y_{i,i} = p + B \, r_i + W_i \, r_i$$

"Cross-term" for distinct users i and j: short vector $y_{i,j}$ such that

$$A \, y_{i,j} = W_i \, r_j$$

Distributed Key Generation

Challenge: No one can know a trapdoor for A

Approach: User i will generate W_i and short $y_{i,j}$ given public parameters

Public parameters

A B p r_i $i \in [\ell]$

Public key for user i:

W_i $y_{i,j}$ $j \in [\ell]$, $j \neq i$

Secret key for user i: short vector $y_{i,i}$ such that

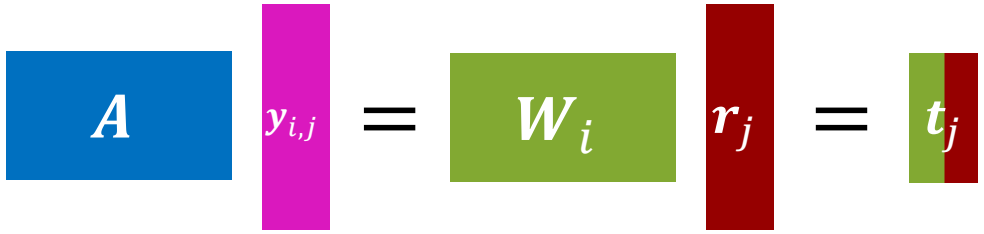
$$A \, y_{i,i} = p + B \, r_i + W_i \, r_i$$

"Cross-term" for distinct users i and j: short vector $y_{i,j}$ such that

$$A \, y_{i,j} = W_i \, r_j$$

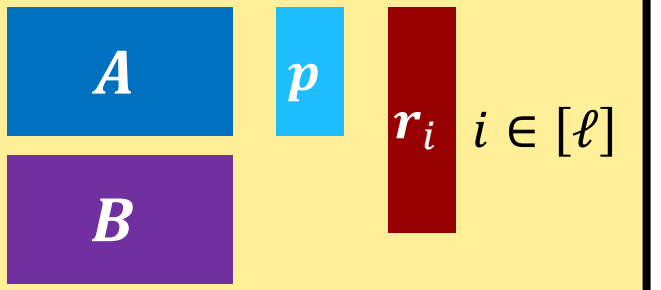
Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A



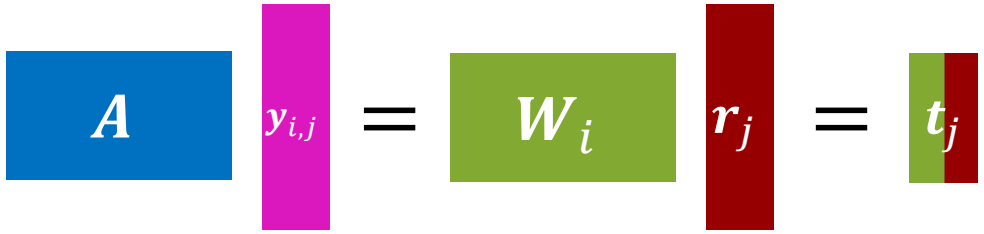
Removed $p + Br_i$ for simplicity

Public parameters



Distributed Key Generation

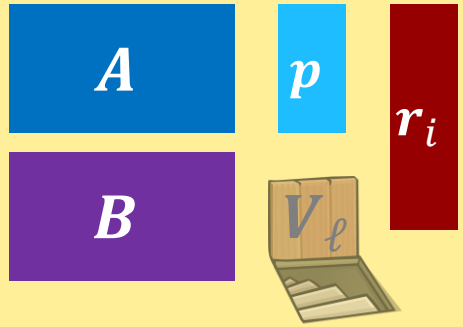
Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A



Removed $p + Br_i$ for simplicity

Approach: Use a random trapdoor for a matrix V_ℓ related to A

Public parameters



Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

$$A \; y_{i,j} = W_i \; r_j = t_j$$

Removed $p + Br_i$ for simplicity

Approach: Use a random trapdoor for a matrix V_ℓ related to A

$$\begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} \begin{bmatrix} y_{i,1} \\ \vdots \\ y_{i,\ell} \end{bmatrix} = \begin{bmatrix} t_1 \\ \vdots \\ t_\ell \end{bmatrix}$$

V_ℓ

Public parameters

A B p r_i $i \in [\ell]$

Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

$$A \; y_{i,j} = W_i \; r_j = t_j$$

Removed $p + Br_i$ for simplicity

Public parameters

A B p r_i $i \in [\ell]$

Approach: Use a random trapdoor for a matrix V_ℓ related to A

$$\begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} \begin{bmatrix} y_{i,1} \\ \vdots \\ y_{i,\ell} \end{bmatrix} = \begin{bmatrix} t_1 \\ \vdots \\ t_\ell \end{bmatrix}$$

V_ℓ

Trapdoor for V_ℓ used to sample short solutions to this equation

Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

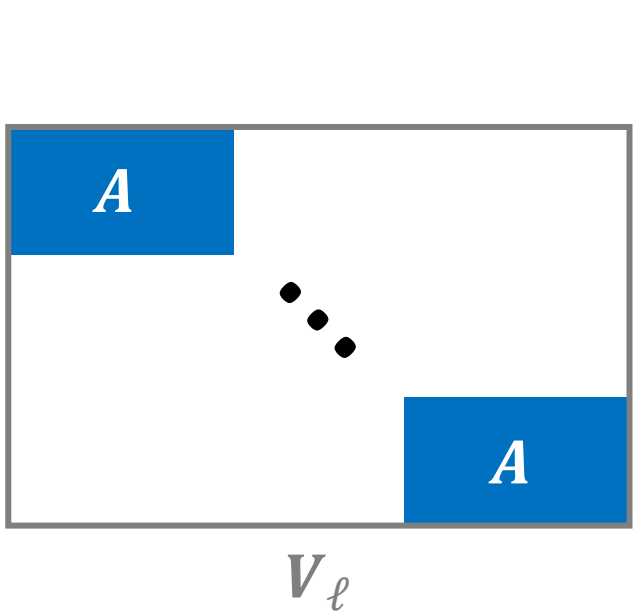
$$A \quad y_{i,j} = W_i \quad r_j = t_j$$

Removed $p + Br_i$ for simplicity

Public parameters

A B p V_ℓ r_i $i \in [\ell]$

Approach: Use a random trapdoor for a matrix V_ℓ related to A



$$\begin{bmatrix} y_{i,1} \\ \vdots \\ y_{i,\ell} \end{bmatrix} = \begin{bmatrix} t_1 \\ \vdots \\ t_\ell \end{bmatrix}$$

Each block could be **any** vector h!

Trapdoor for V_ℓ leaks trapdoor for A!

Trapdoor for V_ℓ used to sample short solutions to this equation

Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

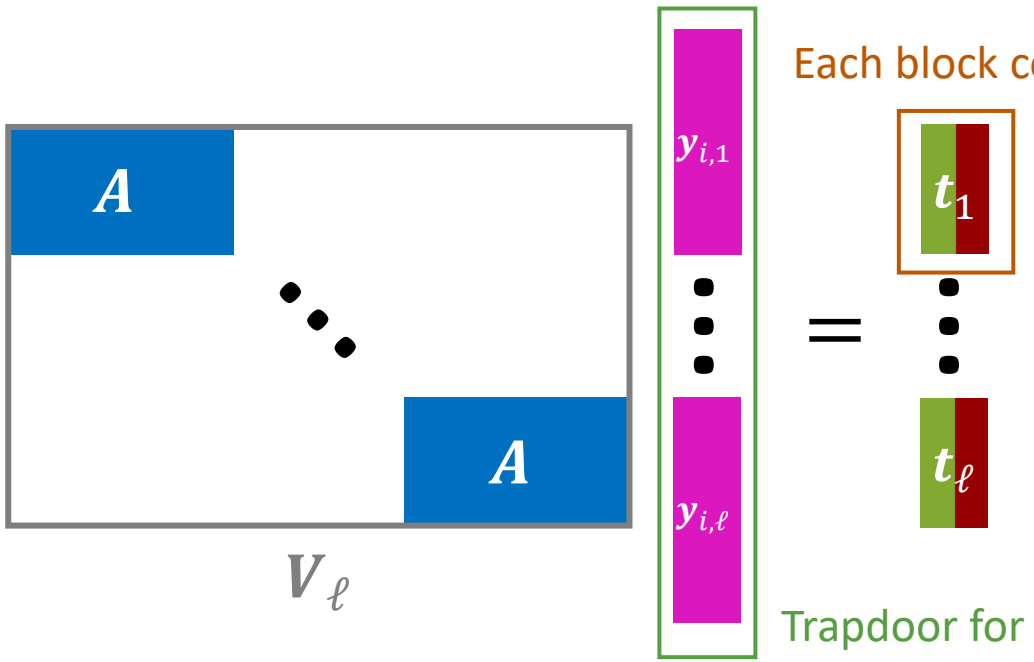
$$A \; y_{i,j} = W_i \; r_j = t_j$$

Removed $p + Br_i$ for simplicity

Public parameters

A B p V_ℓ r_i $i \in [\ell]$

Approach: Use a random trapdoor for a matrix V_ℓ related to A



Each block could be **any** vector h!

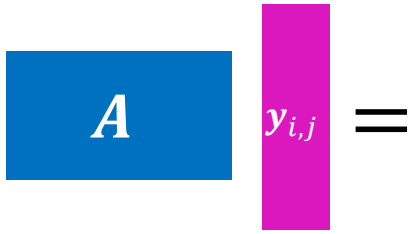
Trapdoor for V_ℓ used to sample short solutions to this equation

Trapdoor for V_ℓ leaks trapdoor for A!

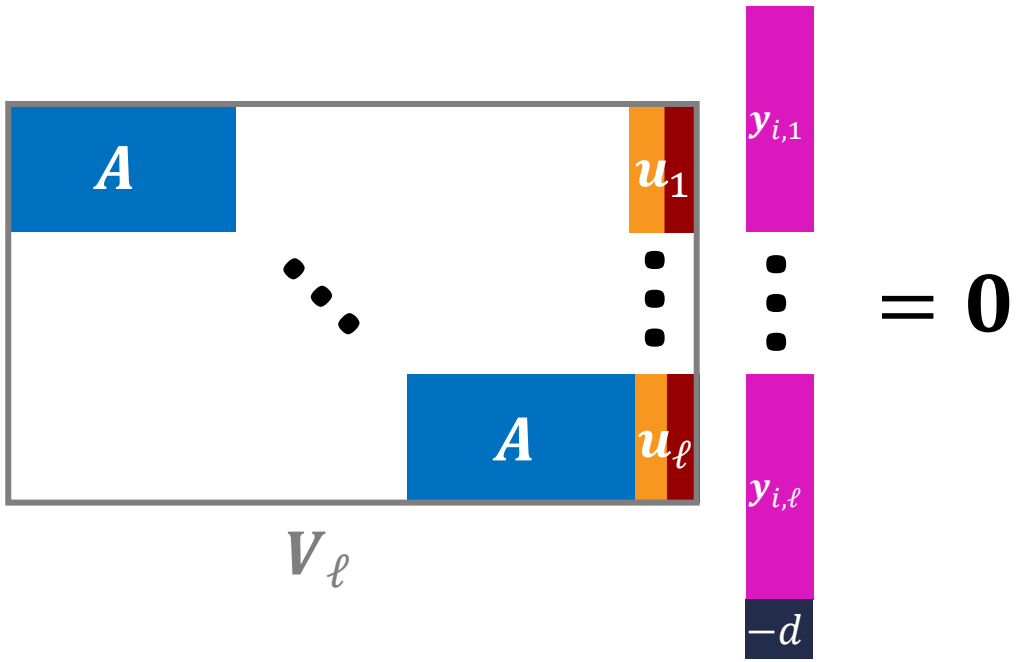
Want $Ay_{i,j} = W_i r_j + h$ for any choice of h on right

Distributed Key Generation

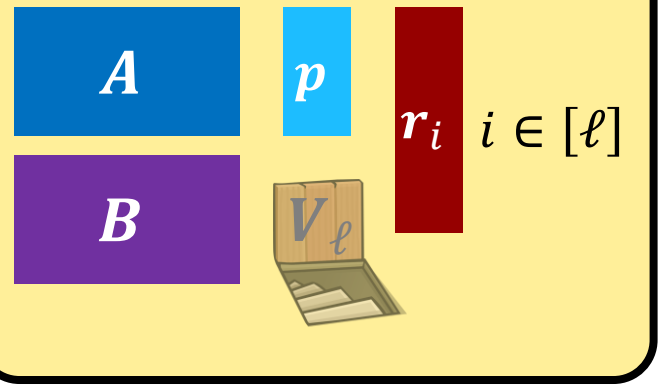
Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A



Approach: Use a random trapdoor for a matrix V_ℓ related to A

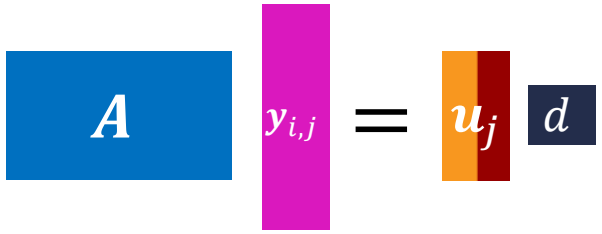


Public parameters

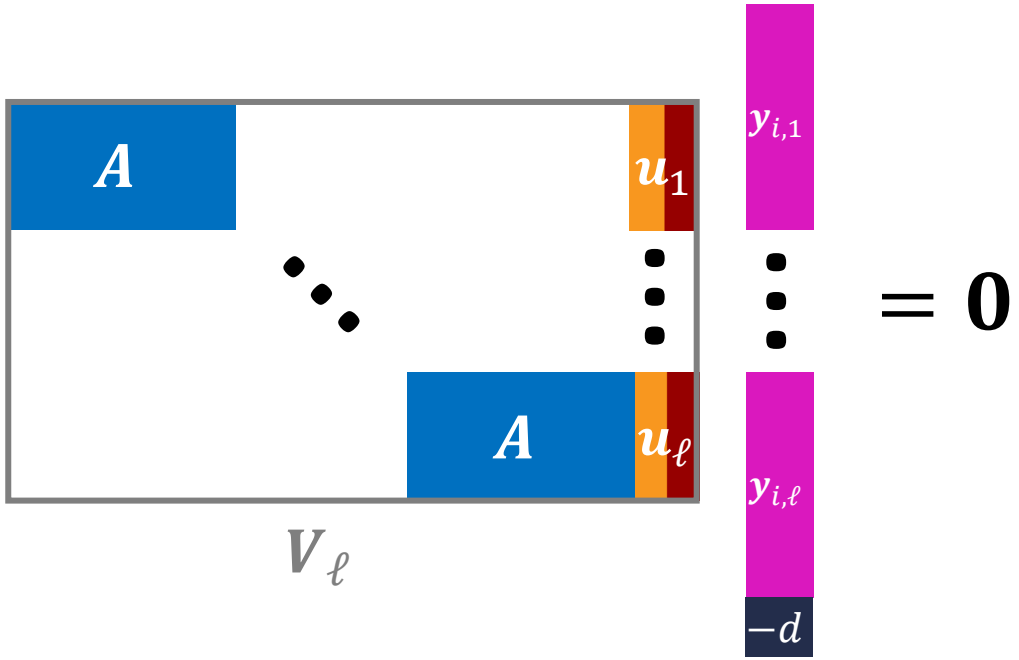


Distributed Key Generation

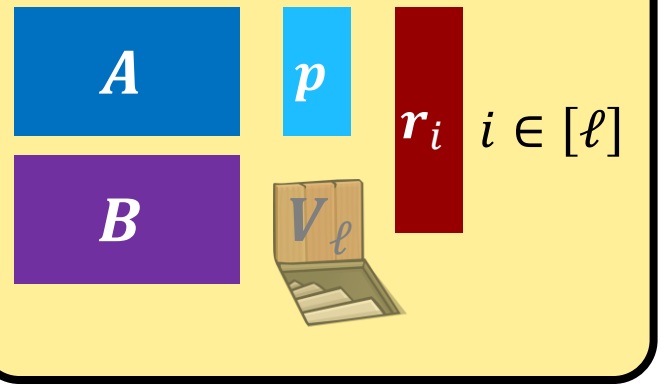
Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A



Approach: Use a random trapdoor for a matrix V_ℓ related to A

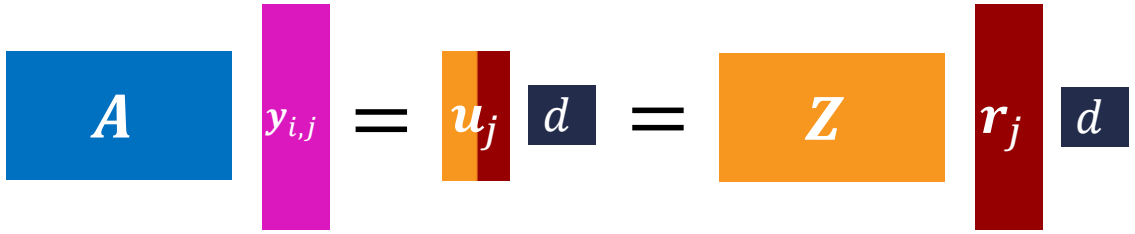


Public parameters

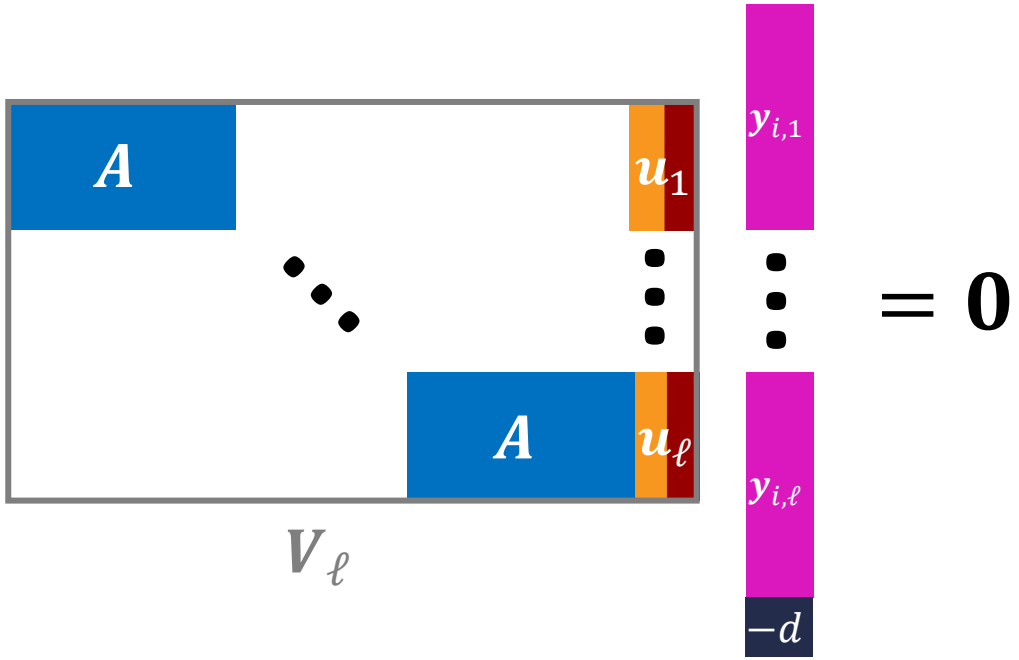


Distributed Key Generation

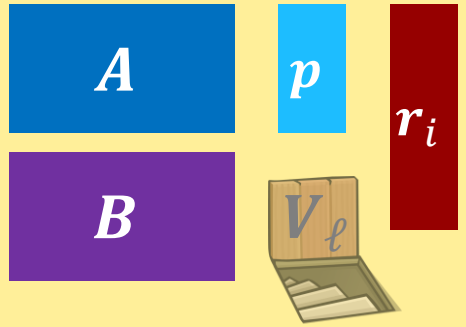
Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A



Approach: Use a random trapdoor for a matrix V_ℓ related to A



Public parameters



Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

$$A \; y_{i,j} = u_j \; d = Z \; r_j \; d = Z \; d \; r_j$$

Public parameters

A B p V_ℓ r_i $i \in [\ell]$

Approach: Use a random trapdoor for a matrix V_ℓ related to A

$$\begin{bmatrix} A & & & u_1 \\ & \ddots & & \vdots \\ & & A & u_\ell \end{bmatrix} \begin{bmatrix} y_{i,1} \\ \vdots \\ y_{i,\ell} \\ -d \end{bmatrix} = 0$$

V_ℓ

Distributed Key Generation

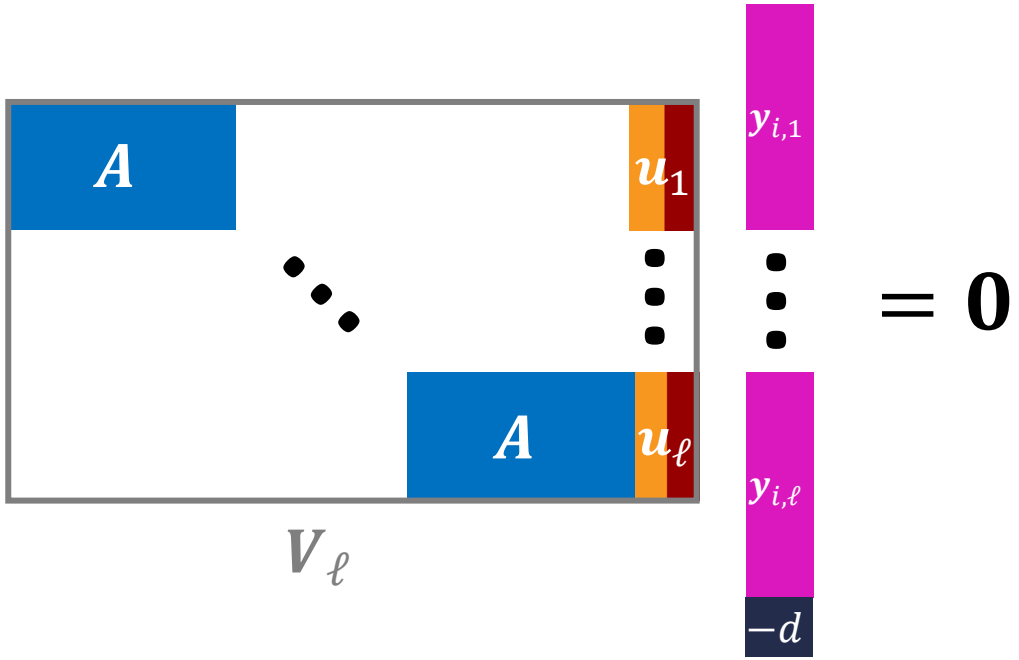
Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

$$A \quad y_{i,j} = u_j \; d = Z \; r_j \; d = Z \; d \; r_j$$

Public parameters

A B Z p V_ℓ r_i $i \in [\ell]$

Approach: Use a random trapdoor for a matrix V_ℓ related to A



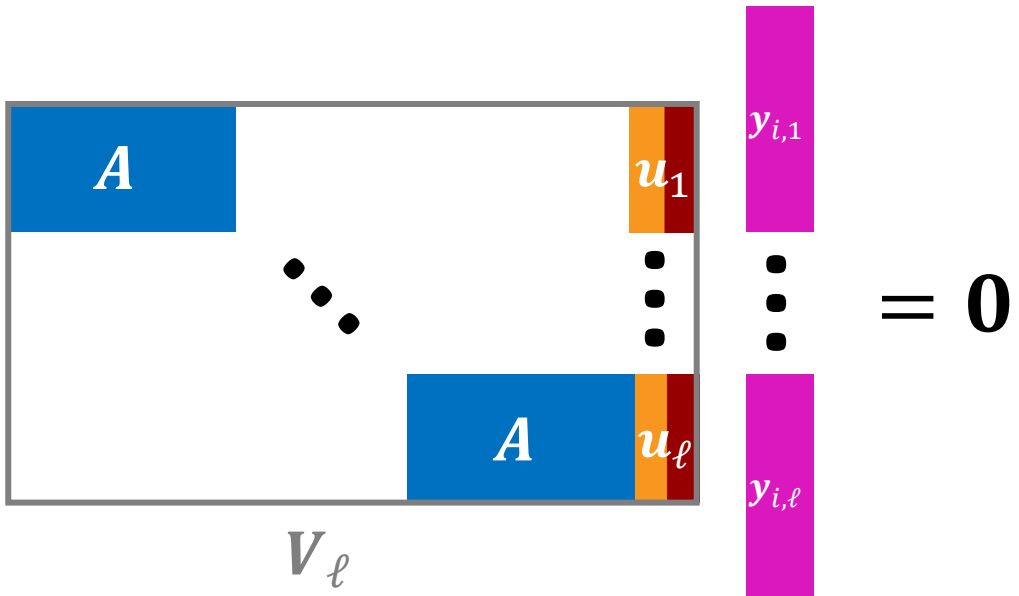
Set $W_i = Z \; d$

Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

$$A \; y_{i,j} = u_j \; d = Z \; r_j \; d = Z \; d \; r_j$$

Approach: Use a random trapdoor for a matrix V_ℓ related to A



$= 0$

Public parameters

A B Z p V_ℓ r_i $i \in [\ell]$

Set $W_i = Z \; d$

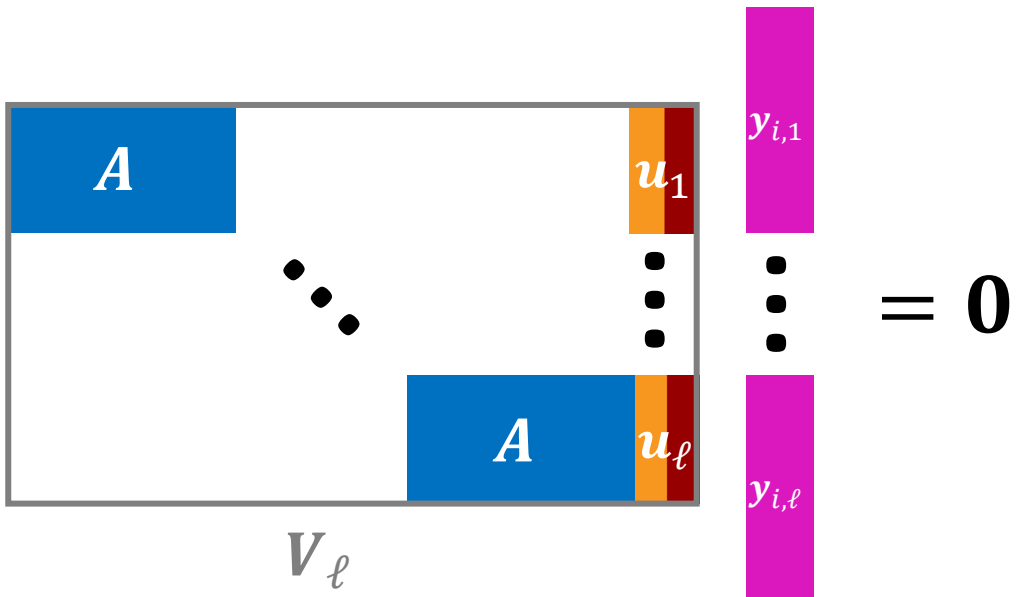
Small number of possible W_i ⇒ high chance of collisions!

Distributed Key Generation

Goal: Generate W_i and $y_{i,j}$ for $j \in [\ell]$ without a trapdoor for A

$$A \; y_{i,j} = u_j \; d = Z \; r_j \; d = Z \; d \; r_j$$

Approach: Use a random trapdoor for a matrix V_ℓ related to A



Public parameters

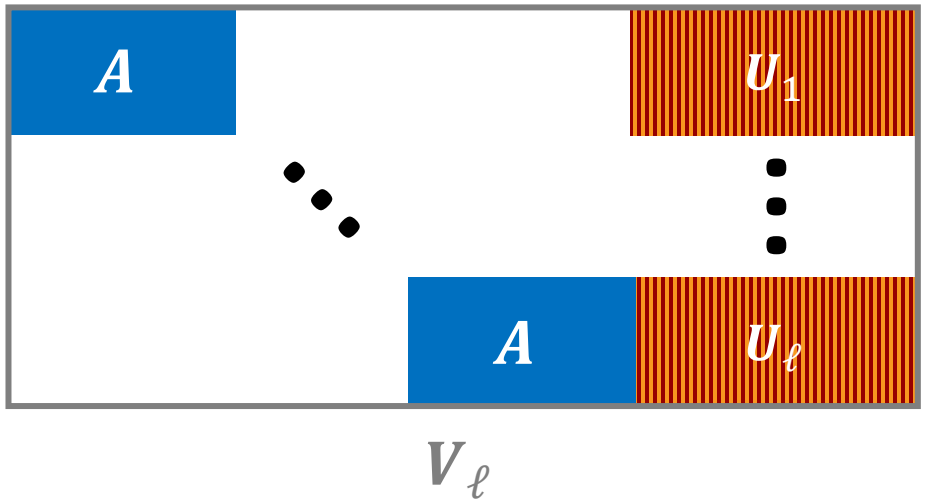
A B Z_τ $\tau \in [k]$ p V_ℓ r_i $i \in [\ell]$

Set $W_i = Z \; d$

Small number of possible W_i ⇒ high chance of collisions!

Solution: generate many independent Z_j, add column in V_ℓ for each, and set $W_i = \sum Z_j \cdot d_j$

Distributed Key Generation



V_ℓ

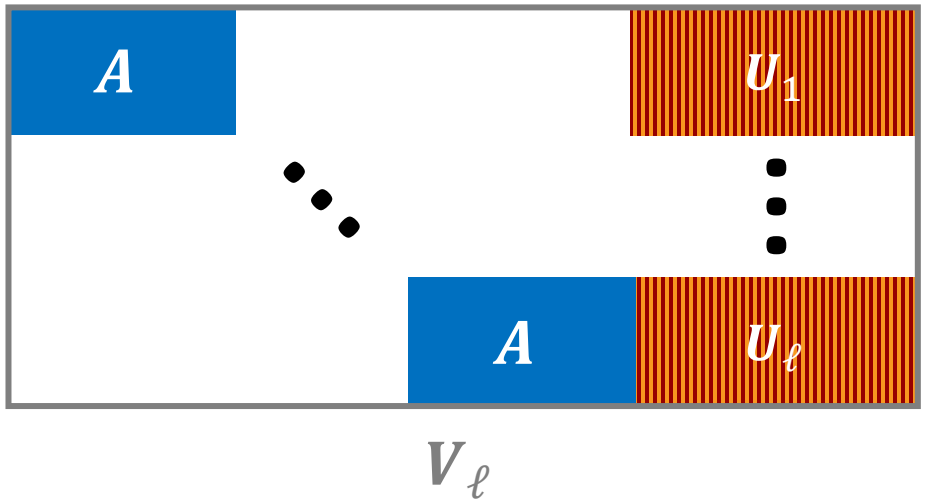
$$\begin{bmatrix} y_{i,1} \\ \vdots \\ y_{i,\ell} \\ -d \end{bmatrix} = 0$$

$$A \; y_{i,j} = U_j \; d$$

$$= \sum_{\tau \in [k]} Z_\tau \; d_\tau \; r_j$$

$$= W_i \; r_j$$

Distributed Key Generation



$$\begin{bmatrix} y_{i,1} \\ \vdots \\ y_{i,\ell} \\ -d \end{bmatrix} = \begin{pmatrix} 0 \\ \vdots \\ p + B\, r_i \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{Block } i$$

$$A\, y_{i,j} = W_i\, r_j$$

$$A\, y_{i,i} = p + B\, r_i + W_i\, r_i$$

Proving Selective Security

Adversary

Challenger

Proving Selective Security

Adversary

$S \subseteq [\ell]$

Challenger

Adversary declares challenge set upfront

Proving Selective Security

Adversary

Challenger

$S \subseteq [\ell]$

$\text{pp}, \{\text{pk}_i\}_{i \in S}, \text{ct}$

Adversary declares challenge set upfront

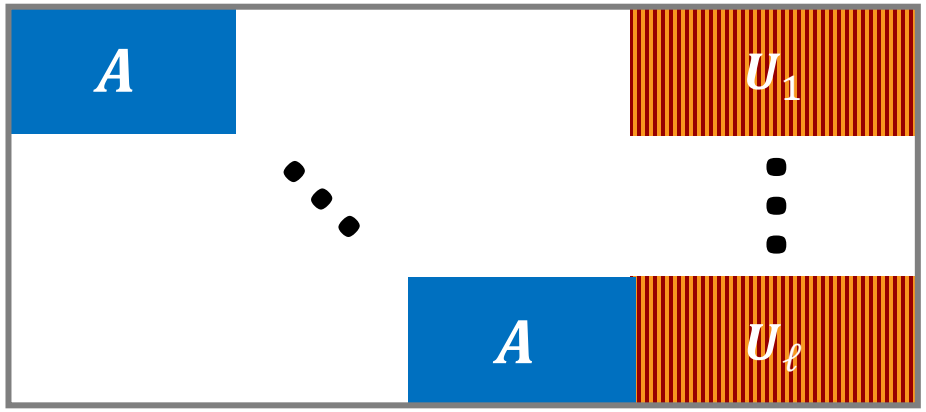
Proving Selective Security

Given $S \subseteq [\ell]$ and an ℓ-succinct LWE tuple, how do we simulate $(\mathrm{pp}, \{\mathrm{pk}_i\}_{i \in S}, \mathrm{ct})$ such that ct is either properly structured or uniform random?

Proving Selective Security

Given $S \subseteq [\ell]$ and an ℓ-succinct LWE tuple, how do we simulate $(\mathrm{pp}, \{\mathrm{pk}_i\}_{i \in S}, \mathrm{ct})$ such that ct is either properly structured or uniform random?

Unlike in $\boldsymbol{V}_\ell$, each $\boldsymbol{U}_i$ is uniform random and narrower in the matrix $\boldsymbol{D}_\ell$ from ℓ-succinct LWE!

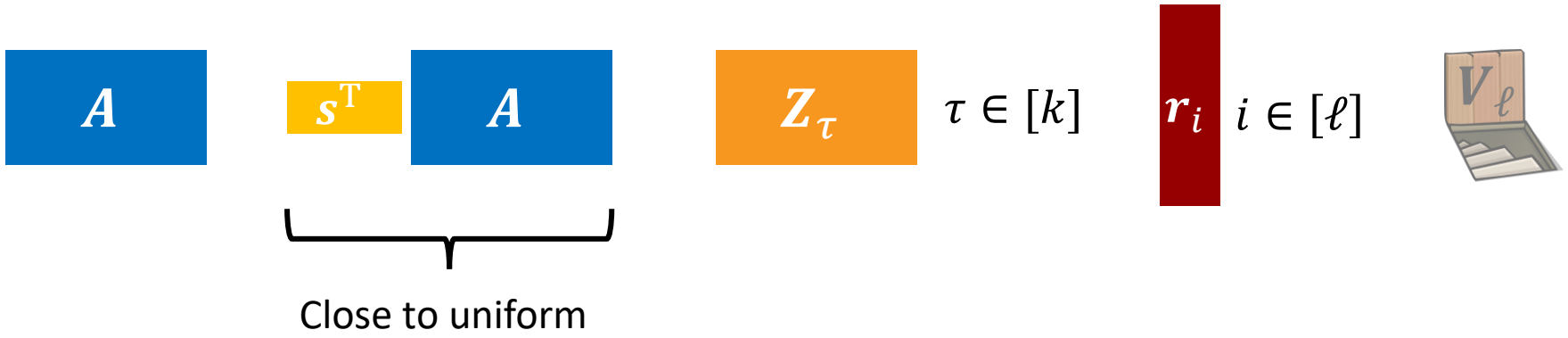


$\boldsymbol{V}_\ell$

Proving Selective Security

Given $S \subseteq [\ell]$ and an ℓ-succinct LWE tuple, how do we simulate $(\mathrm{pp}, \{\mathrm{pk}_i\}_{i \in S}, \mathrm{ct})$ such that ct is either properly structured or uniform random?

Suffices to consider the following tuple if $\boldsymbol{V}_\ell$ is wide enough:



Note that $\boldsymbol{A}, \boldsymbol{Z}_\tau, \boldsymbol{r}_i$ determine $\boldsymbol{V}_\ell$

Proving Selective Security

Given $S \subseteq [\ell]$ and A $s^T A$ Z_τ $\tau \in [k]$ r_i $i \in [\ell]$ 

we need to simulate

Public parameters

A p r_i $i \in [\ell]$

B 

Z_τ $\tau \in [k]$

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$ $j \neq i$

Ciphertext encrypting $b \in \{0,1\}$ to S

$s^T A$ $s^T p$

masks b

$s^T \left(B + \sum_{i \in S} W_i \right)$

Proving Selective Security

Given $S \subseteq [\ell]$ and A $s^T A$ Z_τ $\tau \in [k]$ r_i $i \in [\ell]$ 

we need to additionally simulate

Public parameters

B p

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$, $j \neq i$

Ciphertext encrypting $b \in \{0,1\}$ to S

$$s^T \left(B + \sum_{i \in S} W_i \right) \quad s^T p$$

Proving Selective Security

Given $S \subseteq [\ell]$ and A $s^T A$ Z_τ $\tau \in [k]$ r_i $i \in [\ell]$ 

we need to additionally simulate

Public parameters

B p

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$, $j \neq i$

Ciphertext encrypting $b \in \{0,1\}$ to S

$$s^T \left(B + \sum_{i \in S} W_i \right) \quad s^T p$$

Suppose we sample binary matrices h, H and set

$$B = A H - \sum_{i \in S} W_i$$

$$p = A h$$

Proving Selective Security

Given $S \subseteq [\ell]$ and A $s^T A$ Z_τ $\tau \in [k]$ r_i $i \in [\ell]$ 

we need to additionally simulate

Public parameters

B p

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$ $j \neq i$

Ciphertext encrypting $b \in \{0,1\}$ to S

s^T A H s^T A h

Suppose we sample binary matrices h, H and set

$$B = A H - \sum_{i \in S} W_i$$

$$p = A h$$

Proving Selective Security

Given $S \subseteq [\ell]$ and A $s^T A$ Z_τ $\tau \in [k]$ r_i $i \in [\ell]$ 

we need to additionally simulate

Public parameters

B p

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$ $j \neq i$

Ciphertext encrypting $b \in \{0,1\}$ to S

s^T A H s^T A h

Suppose we sample binary matrices h, H and set

$$B = A H - \sum_{i \in S} W_i$$

$$p = A h$$

If $s^T A$ is switched for uniform, the ciphertext is uniform!

Proving Selective Security

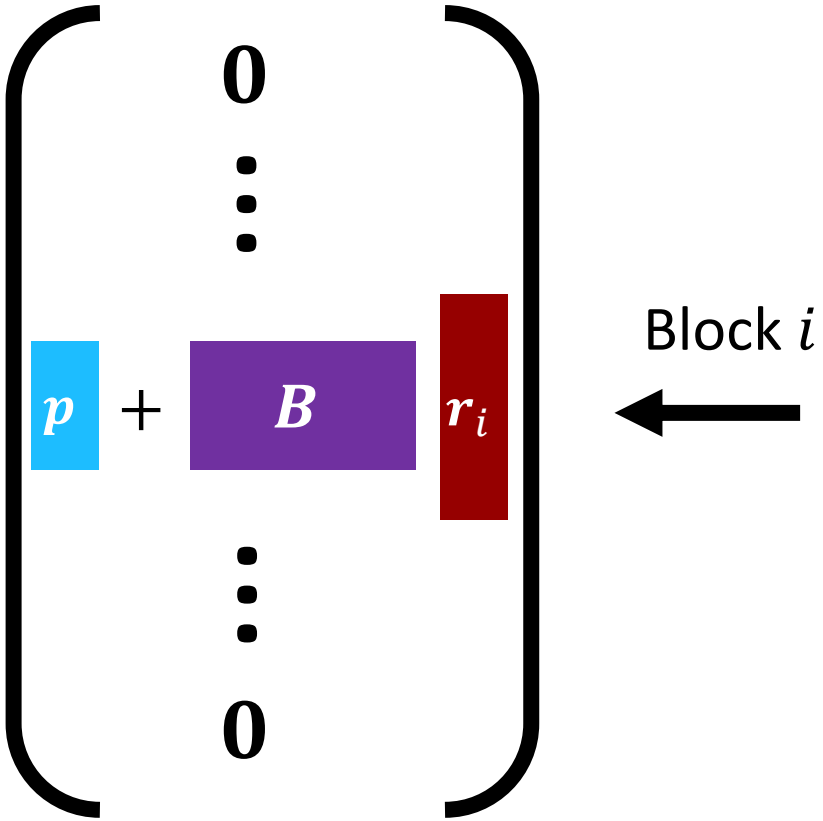
We want the relation

Proving Selective Security

We want the relation

$$B = A\,H - \sum_{i \in S} W_i$$

But… to sample W_i we solve a linear equation involving

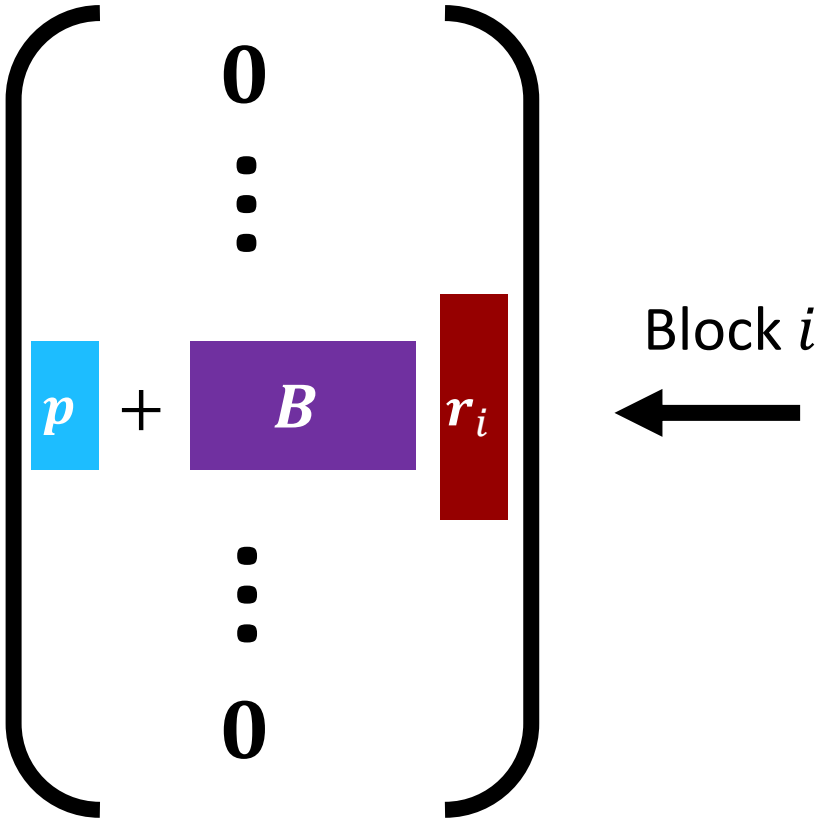


Proving Selective Security

We want the relation

$$B = A\,H - \sum_{i \in S} W_i$$

But… to sample W_i we solve a linear equation involving



This is a circularity!

Proving Selective Security

We want the relation

$$B = A\,H - \sum_{i \in S} W_i$$

Observation: we don't need to simulate $y_{i,i}$

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$, $j \neq i$

But… to sample W_i we solve a linear equation involving

$$\begin{pmatrix} 0 \\ \vdots \\ p + B \; r_i \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{Block } i$$

Proving Selective Security

We want the relation

$$B = A\,H - \sum_{i \in S} W_i$$

Observation: we don't need to simulate $y_{i,i}$

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$, $j \neq i$

Distribution of $y_{i,j}$ is close to one where we instead target $\mathbf{0}$

But… to sample W_i we solve a linear equation involving

$$\begin{pmatrix} \mathbf{0} \\ \vdots \\ p + B\,r_i \\ \vdots \\ \mathbf{0} \end{pmatrix} \leftarrow \text{Block } i$$

Proving Selective Security

We want the relation

$$B = A\,H - \sum_{i \in S} W_i$$

Observation: we don't need to simulate $y_{i,i}$

Public key for user $i \in S$

W_i $y_{i,j}$ $j \in [\ell]$, $j \neq i$

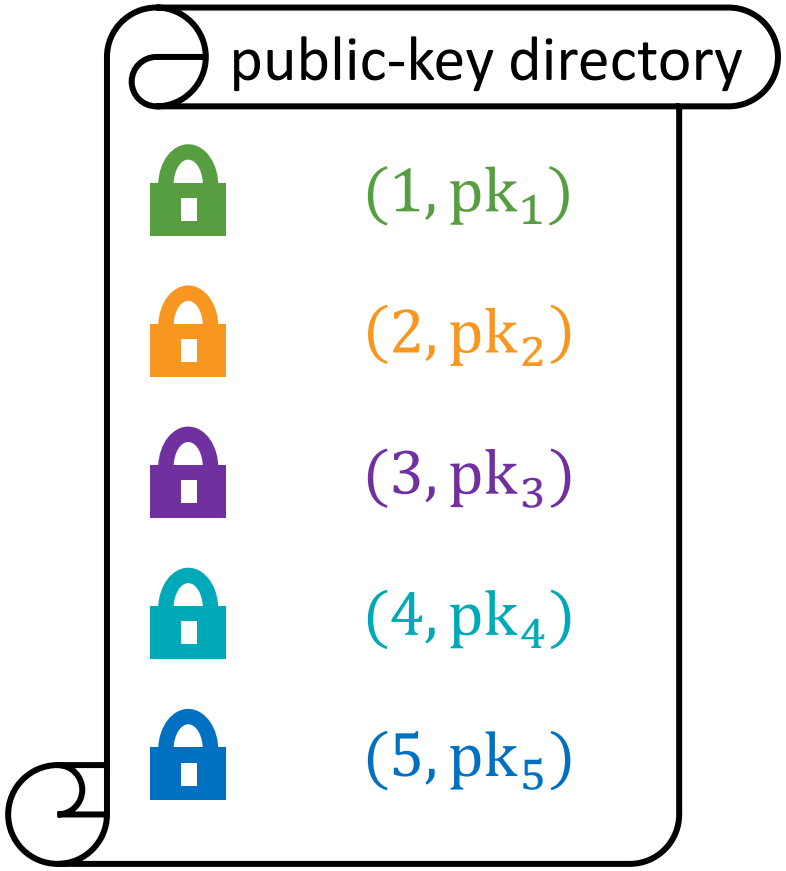
Distribution of $y_{i,j}$ is close to one where we instead target $\mathbf{0}$

But… to sample W_i we solve a linear equation involving

$$\begin{pmatrix} \mathbf{0} \\ \vdots \\ p + B\,r_i \\ \vdots \\ \mathbf{0} \end{pmatrix} \leftarrow \text{Block } i$$

Can now sample public keys before B!

Summary



Broadcast encryption without a central authority

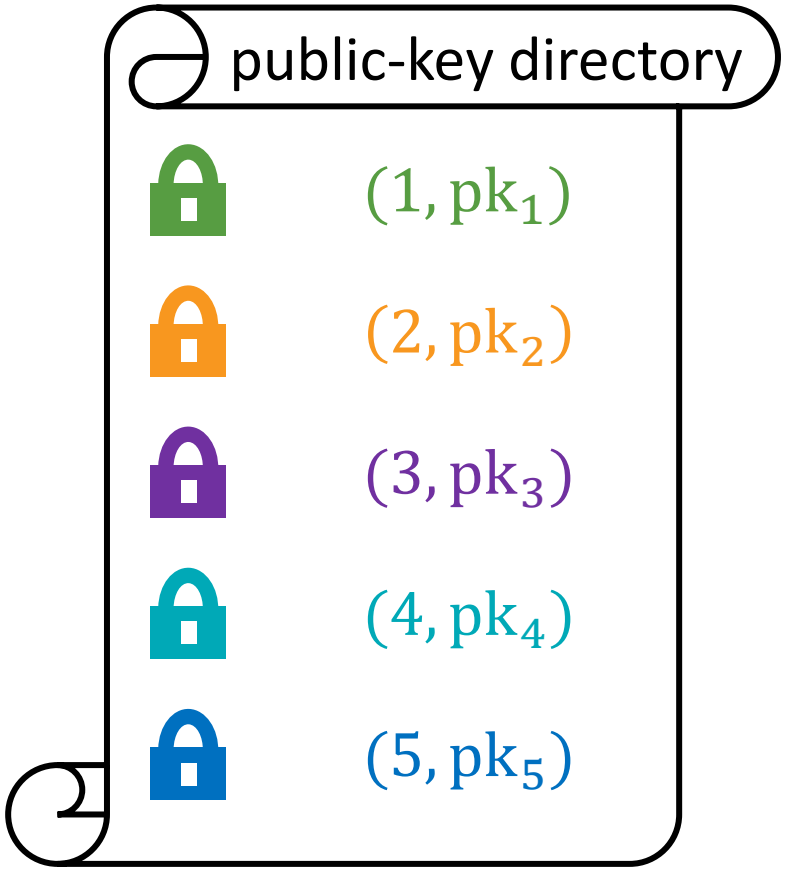
Selectively-secure distributed broadcast encryption for ℓ users from ℓ'-succinct LWE where $\ell' \geq \ell \cdot O(\lambda \log \ell)$

Public parameter size: $\ell^2 \cdot \mathrm{poly}(\lambda, \log \ell)$

User public key size: $\ell \cdot \mathrm{poly}(\lambda, \log \ell)$

Ciphertext size: $\mathrm{poly}(\lambda, \log \ell)$

Summary



Broadcast encryption without a central authority

Selectively-secure distributed broadcast encryption for ℓ users from ℓ'-succinct LWE where $\ell' \geq \ell \cdot O(\lambda \log \ell)$

Public parameter size: $\ell^2 \cdot \mathrm{poly}(\lambda, \log \ell)$

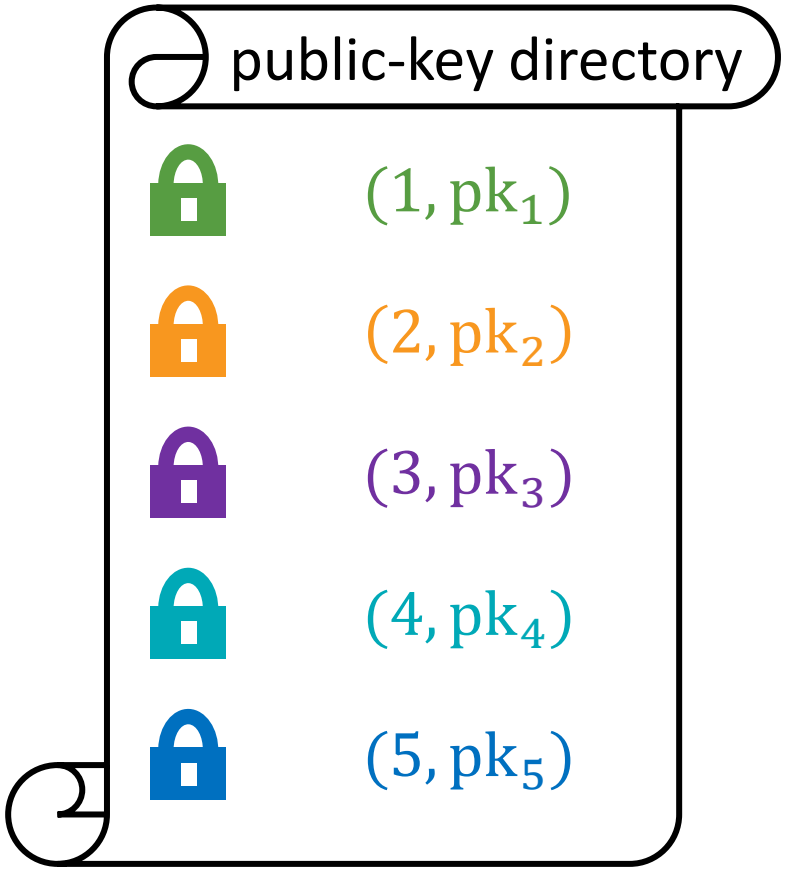
User public key size: $\ell \cdot \mathrm{poly}(\lambda, \log \ell)$

Ciphertext size: $\mathrm{poly}(\lambda, \log \ell)$

Open problems:

- Proving security from plain LWE
- Cryptanalysis and more applications of ℓ-succinct LWE

Summary



Broadcast encryption without a central authority

Selectively-secure distributed broadcast encryption for ℓ users from ℓ'-succinct LWE where $\ell' \geq \ell \cdot O(\lambda \log \ell)$

Public parameter size: $\ell^2 \cdot \mathrm{poly}(\lambda, \log \ell)$

User public key size: $\ell \cdot \mathrm{poly}(\lambda, \log \ell)$

Ciphertext size: $\mathrm{poly}(\lambda, \log \ell)$

Open problems:

- Proving security from plain LWE
- Cryptanalysis and more applications of ℓ-succinct LWE

Upcoming work: registered ABE for circuits from ℓ-succinct LWE in the random oracle model

- Requires simulating challenge ciphertexts w.r.t *malicious* keys
- Techniques generalize to obtain *adaptively-secure* DBE

Thanks for listening!

https://eprint.iacr.org/2024/1417