Distributed Broadcast Encryption from Lattices

David Wu

joint work with Jeffrey Champion

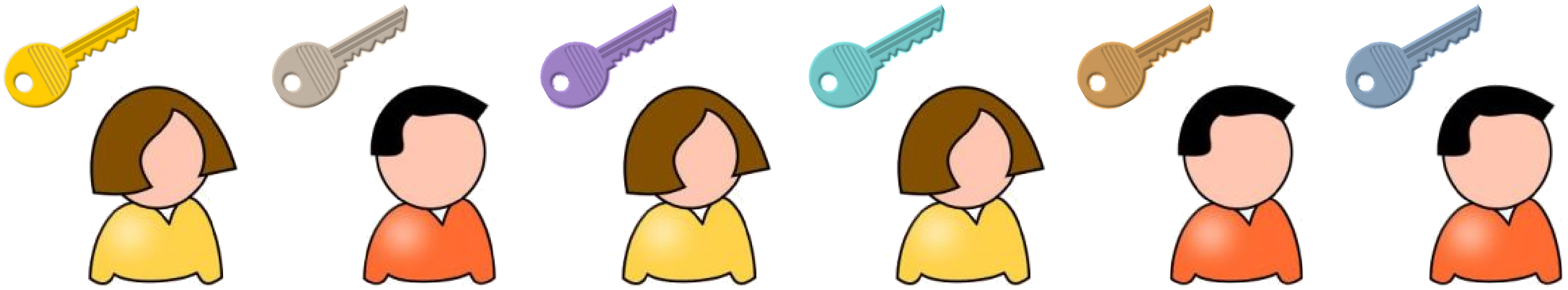
Broadcast Encryption

[FN93]

message m

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

Ciphertext specifies a set of users

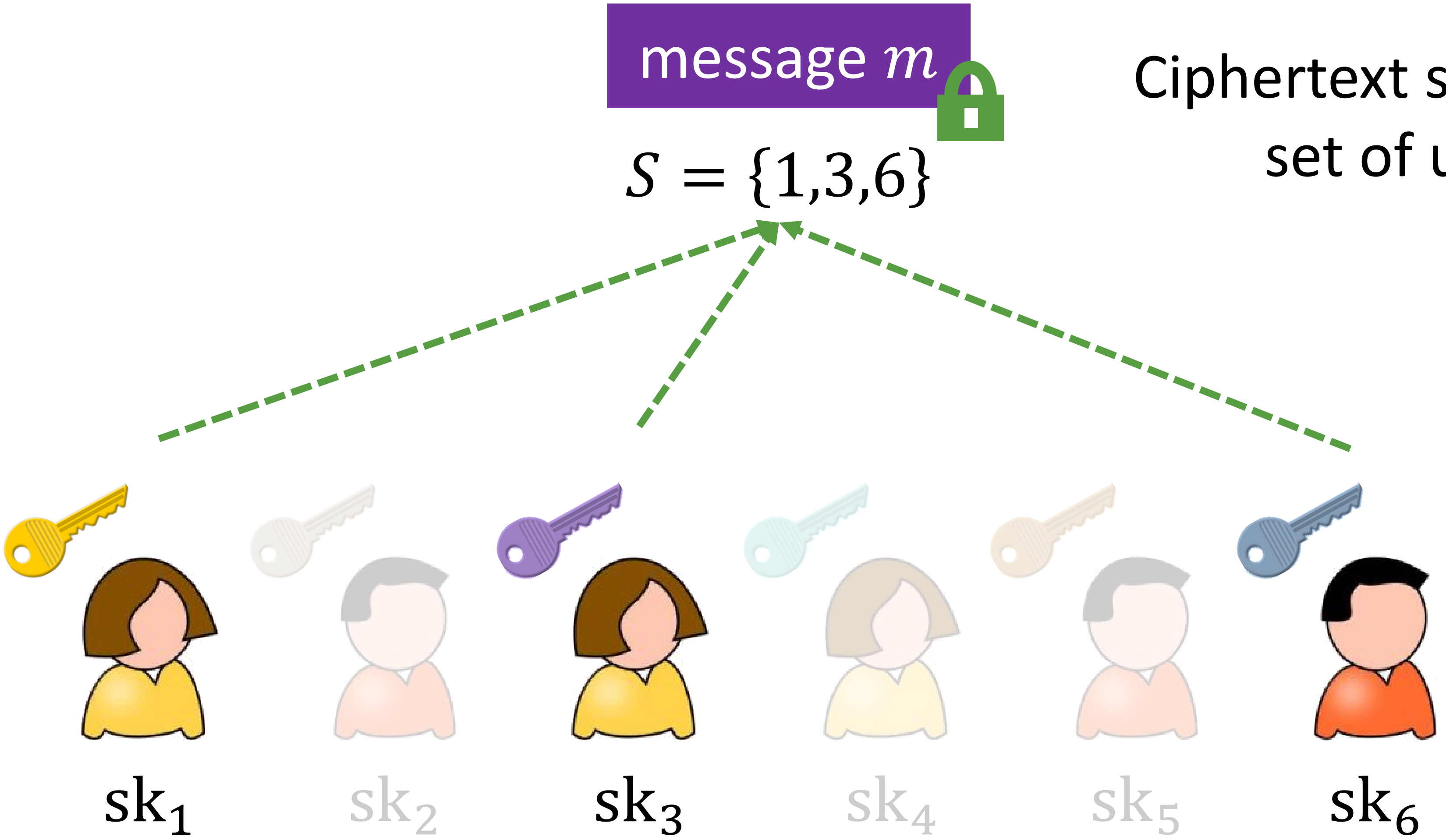


sk_1 sk_2 sk_3 sk_4 sk_5 sk_6

Broadcast Encryption

[FN93]

Functionality: Users in the set can decrypt



Ciphertext specifies a set of users

Broadcast Encryption

[FN93]

Functionality: Users in the set can decrypt

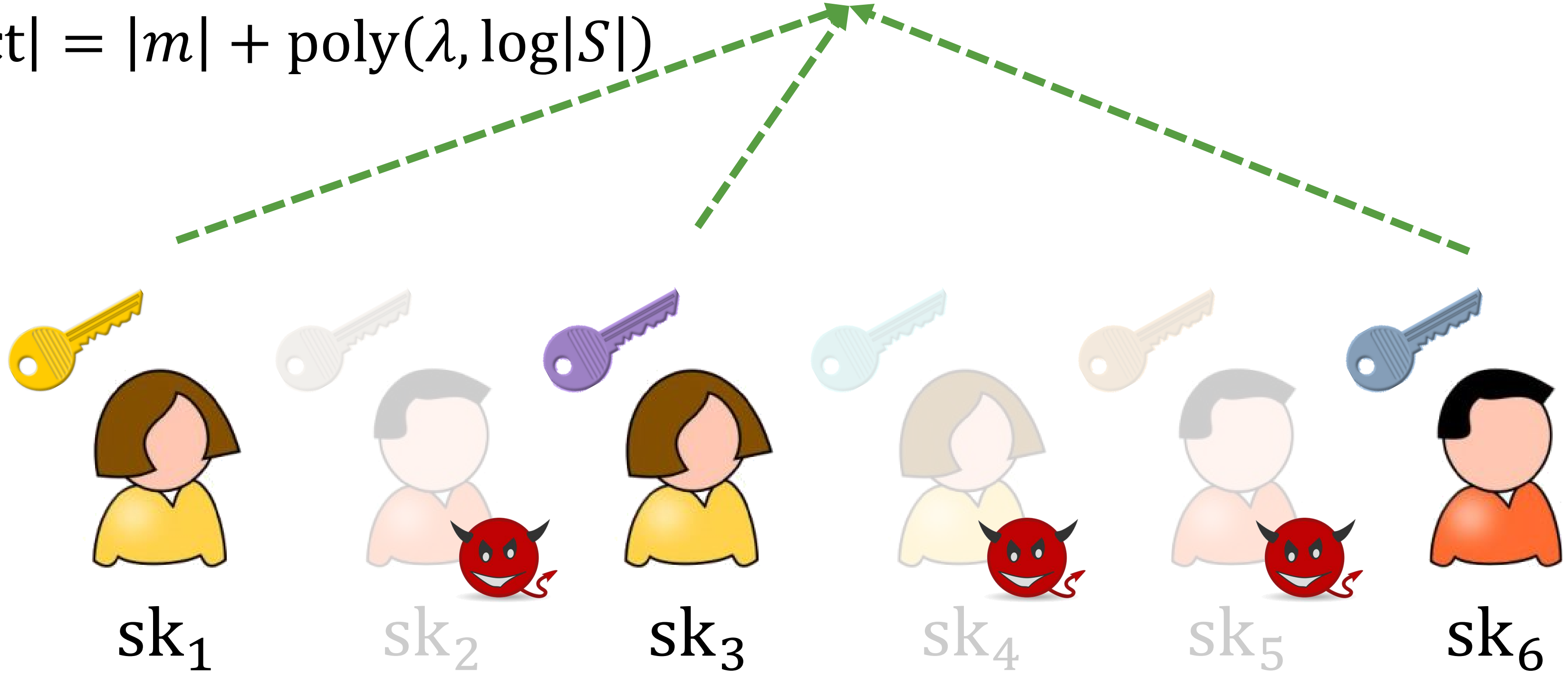
Security: Users outside the set learn nothing about message (even if they collude)

message m

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

Ciphertext specifies a set of users

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



Broadcast Encryption

[FN93]

Functionality: Users in the set can decrypt

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

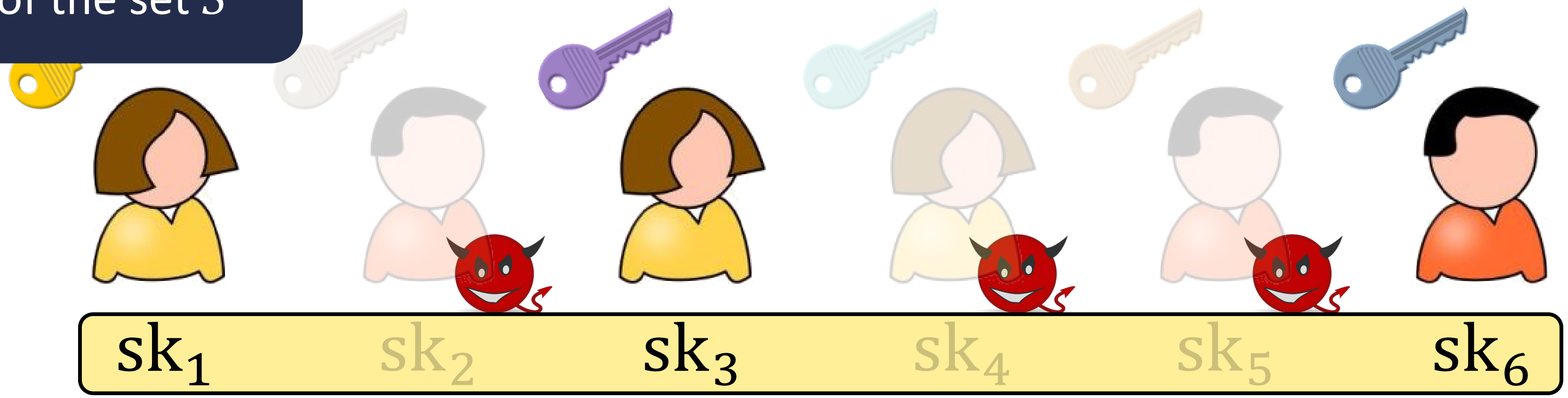
message m

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

Ciphertext specifies a set of users

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

Note: decryption requires knowledge of the set S



Where do the secret keys come from?

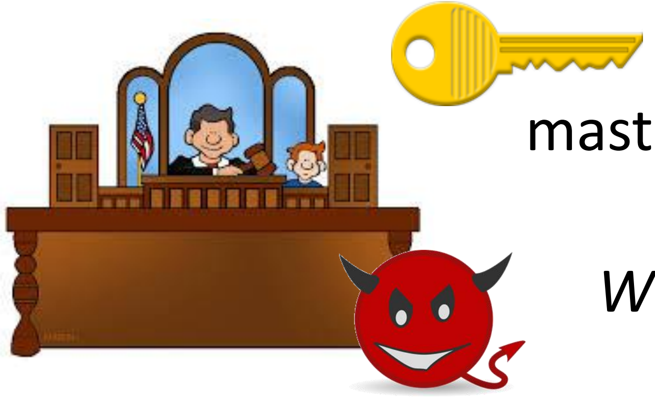
Broadcast Encryption

[FN93]

Central **trusted** authority generates keys

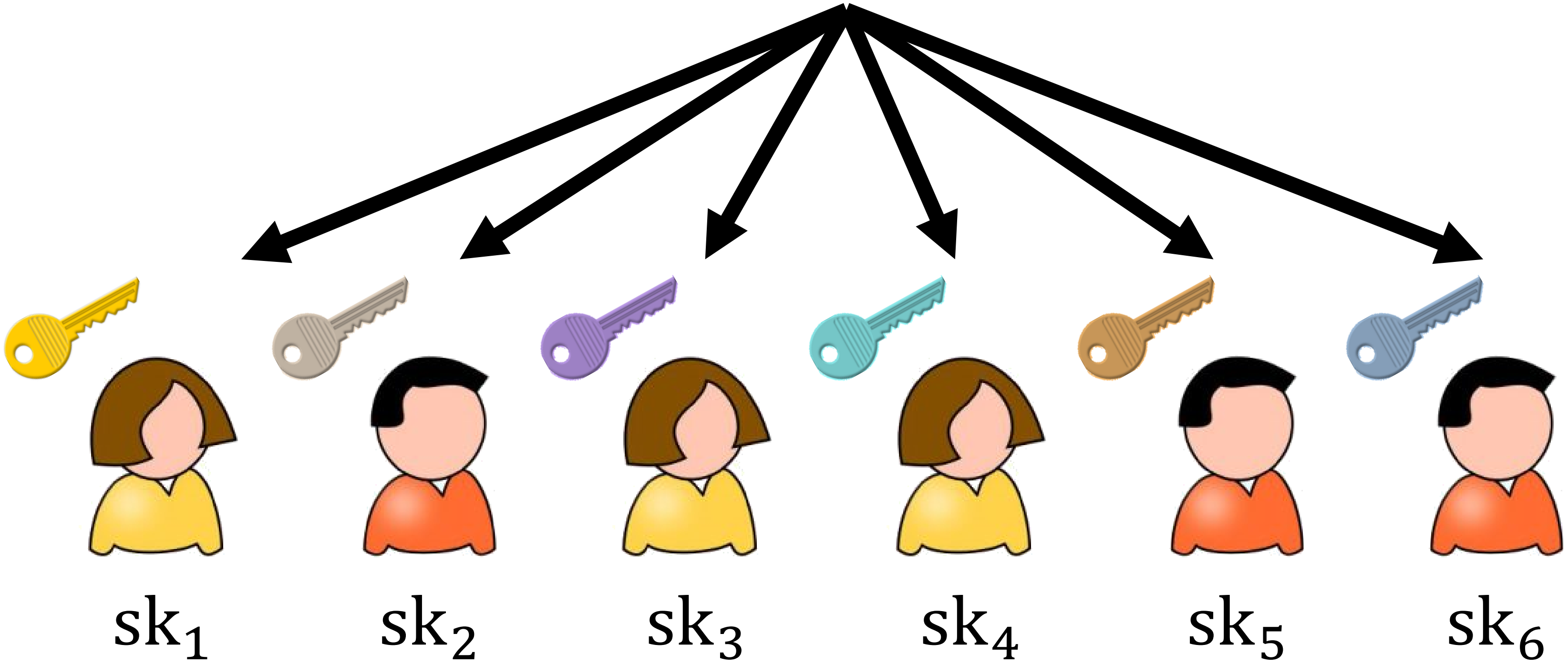
Built-in **key escrow**

Central point of failure



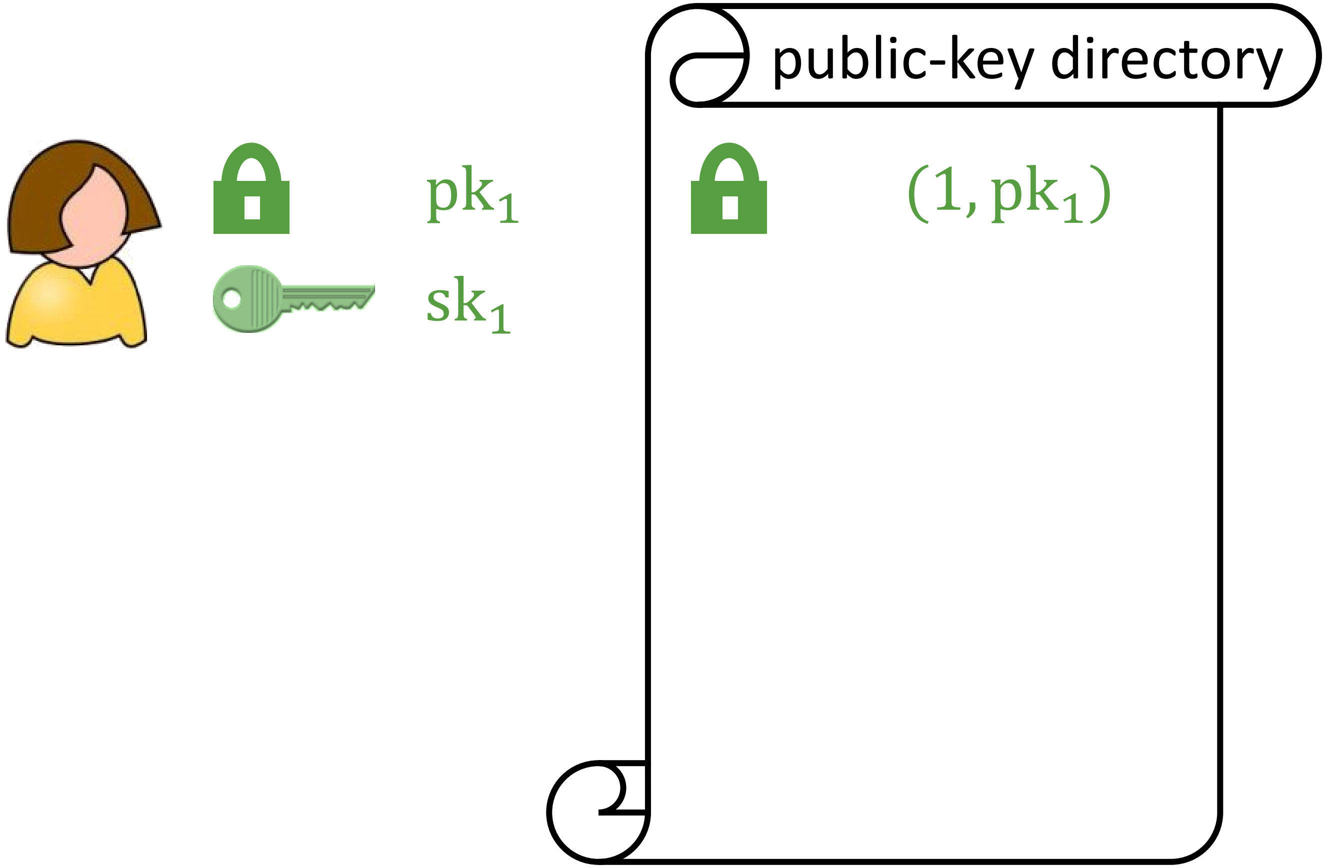
master secret key

What if the key issuer is compromised?



Distributed Broadcast Encryption

[BZ14]

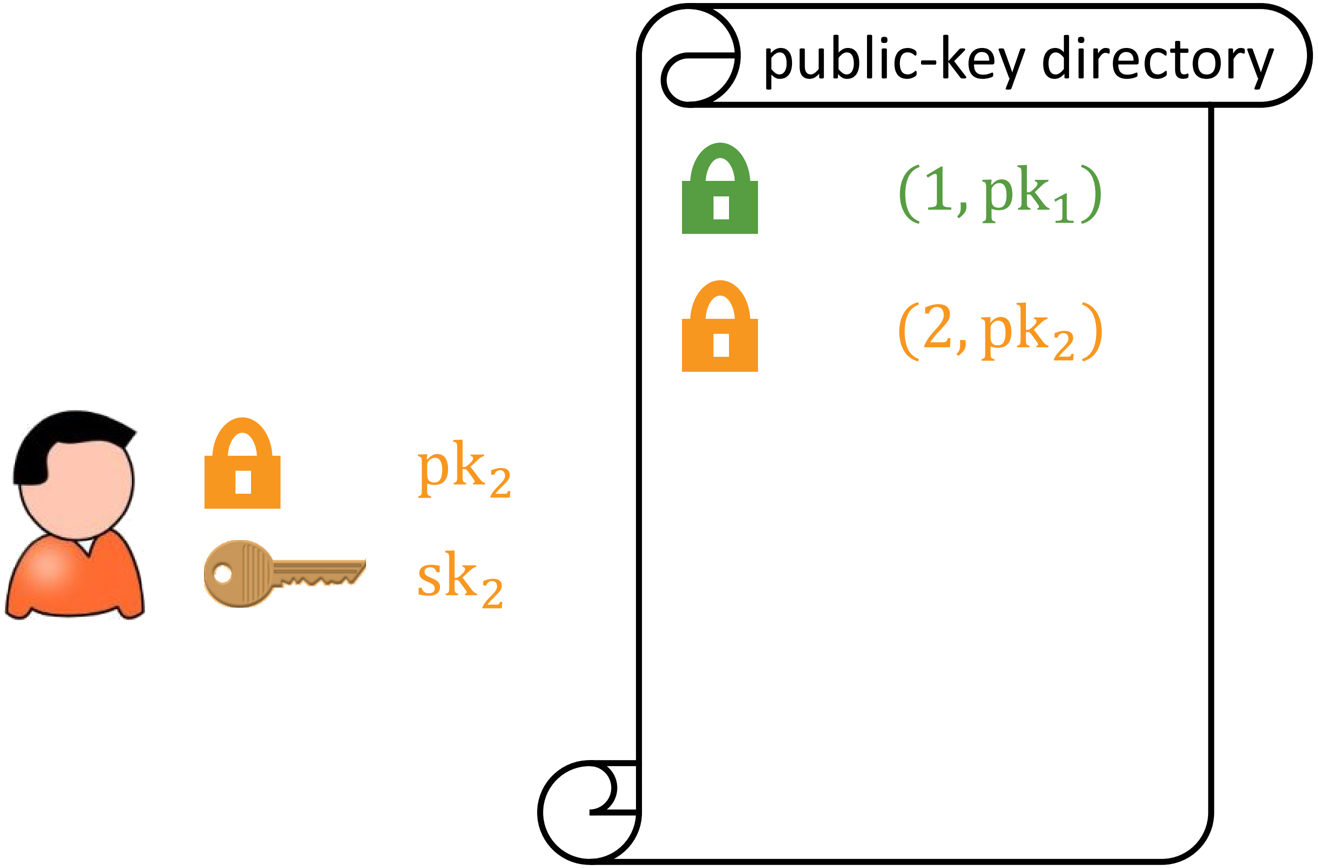


Users generate public/private keys independently (as in public-key encryption)

Broadcast encryption without a central authority

Distributed Broadcast Encryption

[BZ14]

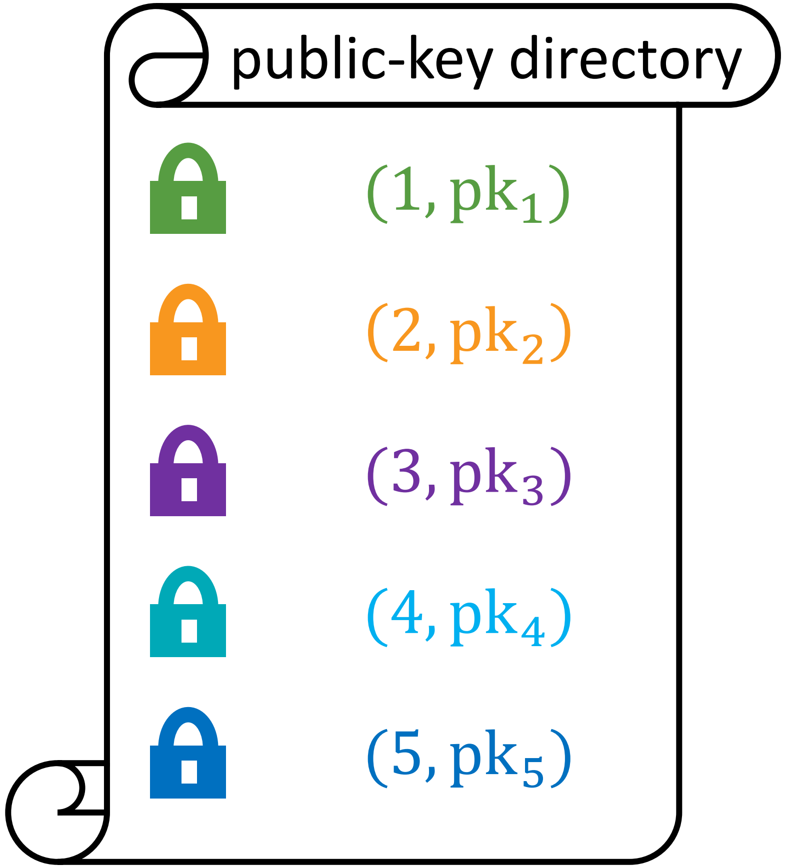


Users generate public/private keys independently (as in public-key encryption)

Broadcast encryption without a central authority

Distributed Broadcast Encryption

[BZ14]



public parameters

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

Can encrypt a message m to any set of public keys

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

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

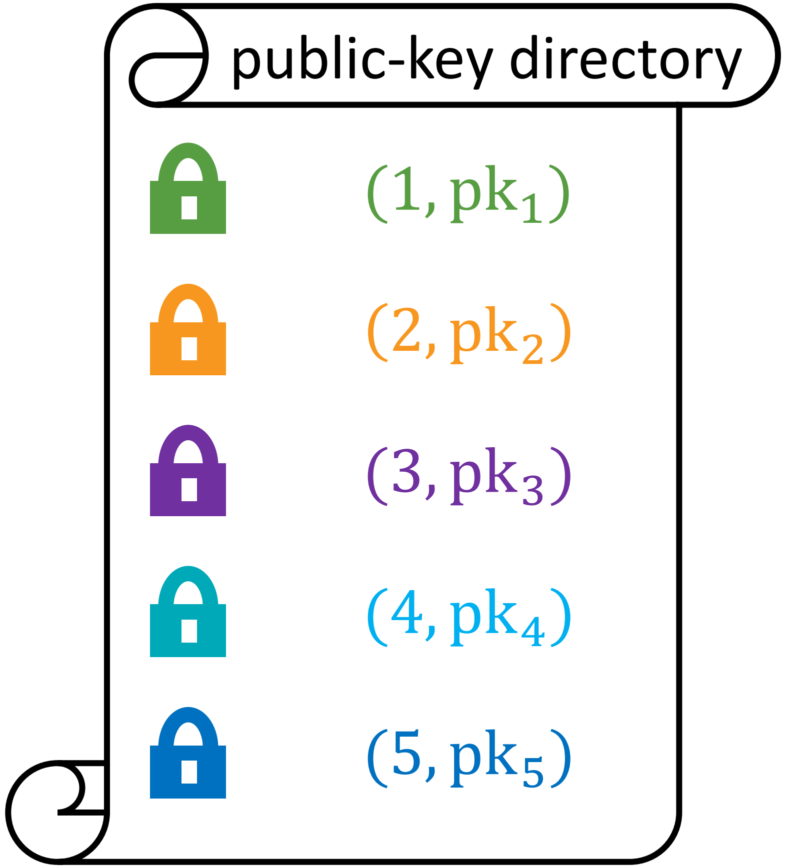
Any secret key associated with broadcast set can decrypt

Decryption does require knowledge of public keys in broadcast set

Broadcast encryption without a central authority

Distributed Broadcast Encryption

[BZ14]



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

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

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

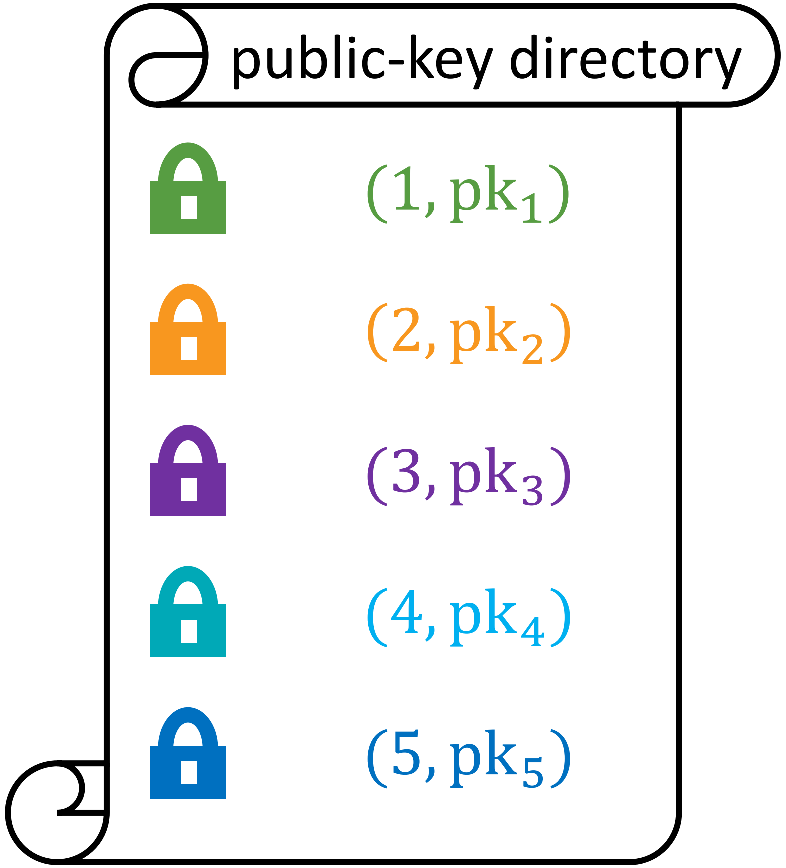
<u>Constructions of Distributed Broadcast Encryption</u>

- 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?

Broadcast encryption without a central authority

Lattice-Based Distributed Broadcast



Lattice-based **centralized broadcast** encryption currently known from

- Lattice-based (no explicit assumption) [BV22]
- Public-coin evasive LWE [Wee22]
- ℓ-succinct LWE [Wee24]

These schemes construct a succinct ciphertext-policy ABE

For **distributed broadcast**, only lattice instantiation goes through witness encryption [FWW23]

- Requires **private-coin** evasive LWE [Tsa22, VWW22]

This work: distributed broadcast encryption from ℓ-succinct LWE

ℓ-Succinct LWE Assumption

[Wee24]

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

$$\boxed{\begin{array}{c}\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}\\ \boldsymbol{U}_i \leftarrow \mathbb{Z}_q^{n\times m}\end{array}}$$

$$\underbrace{\left[\begin{array}{ccc|c}\boldsymbol{A} & & & \boldsymbol{U}_1\\ & \ddots & & \vdots\\ & & \boldsymbol{A} & \boldsymbol{U}_\ell\end{array}\right]}_{\boldsymbol{D}_\ell}\boldsymbol{T} = \begin{bmatrix}\boldsymbol{G} & & \\ & \ddots & \\ & & \boldsymbol{G}\end{bmatrix}$$

$$\boxed{\boldsymbol{G} = \boldsymbol{I}_n \otimes \left[1,2,\dots,2^{\lfloor \log q\rfloor}\right]}$$

$$\left(\boldsymbol{A}, \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}\right) \approx \left(\boldsymbol{A}, \boldsymbol{z}^{\mathrm{T}}\right) \quad \text{given } \boldsymbol{U}_1, \dots, \boldsymbol{U}_\ell, \boldsymbol{T}$$

Falsifiable!

$$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{U}_i \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^{n}, \boldsymbol{e} \leftarrow \chi^{m}, \boldsymbol{z} \leftarrow \mathbb{Z}_q^{m}$$

ℓ-Succinct LWE Assumption

[Wee24]

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

$$(\boldsymbol{A}, \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}) \approx (\boldsymbol{A}, \boldsymbol{z}^{\mathrm{T}}) \text{ given } \boldsymbol{D}_\ell = [\boldsymbol{I}_\ell \otimes \boldsymbol{A} \mid \boldsymbol{U}] \text{ and trapdoor for } \boldsymbol{D}_\ell$$

Special cases that is implied by LWE:

- $\ell = 1$
- if $\boldsymbol{U}$ is very wide (i.e., if $\boldsymbol{U} \in \mathbb{Z}_q^{\ell n \times \ell m}$)

Applications typically require large ℓ and narrow $\boldsymbol{U}$ (e.g., $\boldsymbol{U} \in \mathbb{Z}_q^{\ell n \times m}$)

- Falsifiable, instance-independent assumption, implied by public-coin evasive LWE + LWE
- Trapdoor useful for compression: CP-ABE with short ciphertexts [Wee24], functional commitments for circuits [WW23]

Starting Point: Centralized Broadcast Encryption

Previous lattice-based broadcast encryption all constructed a CP-ABE scheme

We take a more direct approach (similar to earlier pairing-based approaches)

$\boldsymbol{W}_1, \boldsymbol{r}_1$

$\boldsymbol{W}_2, \boldsymbol{r}_2$

$\boldsymbol{W}_3, \boldsymbol{r}_3$

Each user associated with **public** matrix $\boldsymbol{W}_i \in \mathbb{Z}_q^{n \times m}$ and vector $\boldsymbol{r}_i \in \mathbb{Z}_q^m$

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}$ where $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{Z}_q^{n \times m}$ and $\boldsymbol{p} \in \mathbb{Z}_q^n$

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

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A}$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} + \sum_{i \in S} \boldsymbol{W}_i \right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

Noise terms not shown

Starting Point: Centralized Broadcast Encryption

$$\boldsymbol{c}_1^{\mathrm{T}}\mathrm{sk}_i - \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} - \sum_{j \in S\setminus\{i\}} \boldsymbol{s}^{\mathrm{T}}\boldsymbol{W}_j\boldsymbol{r}_i$$

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}$ and $(\boldsymbol{W}_1, \boldsymbol{r}_1), \ldots, (\boldsymbol{W}_\ell, \boldsymbol{r}_\ell)$

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

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \xrightarrow{\text{multiply by } \mathrm{sk}_i} \boldsymbol{c}_1^{\mathrm{T}}\mathrm{sk}_i \approx \boldsymbol{s}^{\mathrm{T}}(\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i\boldsymbol{r}_i)$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j\right) \xrightarrow{\text{multiply by } \boldsymbol{r}_i} \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B}\boldsymbol{r}_i + \sum_{j \in S} \boldsymbol{W}_j\boldsymbol{r}_i\right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

This requires $\boldsymbol{r}_i$ be short

Goal: user $i \in S$ should be able to recover μ

Secret key for user $\boldsymbol{i}$: short vector that recodes from $\boldsymbol{A}$ to $\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i\boldsymbol{r}_i$

$$\mathrm{sk}_i \leftarrow \boldsymbol{A}^{-1}(\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i\boldsymbol{r}_i)$$

sk_i is a (short) preimage of $\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i\boldsymbol{r}_i$

Starting Point: Centralized Broadcast Encryption

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}$ and $(\boldsymbol{W}_1, \boldsymbol{r}_1), \ldots, (\boldsymbol{W}_\ell, \boldsymbol{r}_\ell)$

$$\boldsymbol{c}_1^{\mathrm{T}} \mathrm{sk}_i - \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p} - \sum_{j \in S \setminus \{i\}} \boldsymbol{s}^{\mathrm{T}} \boldsymbol{W}_j \boldsymbol{r}_i$$

Need a way to remove the cross terms $\boldsymbol{W}_j \boldsymbol{r}_i$

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

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A} \xrightarrow{\text{multiply by } \mathrm{sk}_i} \boldsymbol{c}_1^{\mathrm{T}} \mathrm{sk}_i \approx \boldsymbol{s}^{\mathrm{T}} (\boldsymbol{p} + \boldsymbol{B} \boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i)$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j \right) \xrightarrow{\text{multiply by } \boldsymbol{r}_i} \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} \boldsymbol{r}_i + \sum_{j \in S} \boldsymbol{W}_j \boldsymbol{r}_i \right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

This requires $\boldsymbol{r}_i$ be short

Goal: user $i \in S$ should be able to recover μ

Secret key for user $\boldsymbol{i}$: short vector that recodes from $\boldsymbol{A}$ to $\boldsymbol{p} + \boldsymbol{B} \boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i$

$$\mathrm{sk}_i \leftarrow \boldsymbol{A}^{-1} (\boldsymbol{p} + \boldsymbol{B} \boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i)$$

sk_i is a (short) preimage of $\boldsymbol{p} + \boldsymbol{B} \boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i$

Starting Point: Centralized Broadcast Encryption

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}$ and $(\boldsymbol{W}_1, \boldsymbol{r}_1), \ldots, (\boldsymbol{W}_\ell, \boldsymbol{r}_\ell)$ and $\boxed{\boldsymbol{A}^{-1}(\boldsymbol{W}_i \boldsymbol{r}_j)}$

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

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A} \xrightarrow{\text{multiply by } \mathrm{sk}_i} \boldsymbol{c}_1^{\mathrm{T}} \mathrm{sk}_i \approx \boldsymbol{s}^{\mathrm{T}} (\boldsymbol{p} + \boldsymbol{B} \boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i)$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j \right) \xrightarrow{\text{multiply by } \boldsymbol{r}_i} \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} \boldsymbol{r}_i + \sum_{j \in S} \boldsymbol{W}_j \boldsymbol{r}_i \right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

Decryption:

Suffices to recover μ from c_3

$$\boxed{\boldsymbol{c}_1^{\mathrm{T}} \mathrm{sk}_i - \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p} - \sum_{j \in S \setminus \{i\}} \boldsymbol{s}^{\mathrm{T}} \boldsymbol{W}_j \boldsymbol{r}_i} \Rightarrow \boxed{\boldsymbol{c}_1^{\mathrm{T}} \mathrm{sk}_i + \boldsymbol{c}_1^{\mathrm{T}} \sum_{j \in S \setminus \{i\}} \boldsymbol{A}^{-1}(\boldsymbol{W}_j \boldsymbol{r}_i) - \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p}}$$

Starting Point: Centralized Broadcast Encryption

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}$ and $(\boldsymbol{W}_1, \boldsymbol{r}_1), \dots, (\boldsymbol{W}_\ell, \boldsymbol{r}_\ell)$ and $\boldsymbol{A}^{-1}(\boldsymbol{W}_i \boldsymbol{r}_j)$

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

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A} \xrightarrow{\text{multiply by } \mathrm{sk}_i} \boldsymbol{c}_1^{\mathrm{T}} \mathrm{sk}_i \approx \boldsymbol{s}^{\mathrm{T}} (\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i)$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j \right) \xrightarrow{\text{multiply by } \boldsymbol{r}_i} \boldsymbol{c}_2^{\mathrm{T}} \boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B}\boldsymbol{r}_i + \sum_{j \in S} \boldsymbol{W}_j \boldsymbol{r}_i \right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

This is a **centralized** broadcast encryption scheme

Sampling cross-terms $\boldsymbol{A}^{-1}(\boldsymbol{W}_i \boldsymbol{r}_j)$ and secret keys $\mathrm{sk}_i \leftarrow \boldsymbol{A}^{-1}(\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i)$ require knowledge of the trapdoor for $\boldsymbol{A}$

Distributing the Setup

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

Approach: Each user will choose their own $\boldsymbol{W}_i$, everything else will be in the public parameters

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell$

$\boldsymbol{W}_1$

$\boldsymbol{W}_2$

$\boldsymbol{W}_3$

For correctness, each user also needs to generate a secret key and cross-terms

Cross term: $\forall i \neq j : \boldsymbol{A}\boldsymbol{y}_{i,j} = \boldsymbol{W}_i \boldsymbol{r}_j$

Secret key: $\boldsymbol{A}\boldsymbol{y}_{i,i} = \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i$

But user i does **not** have a trapdoor for $\boldsymbol{A}$…

Consider first a simpler problem:

Sample $\boldsymbol{W}_i$ together with short $\boldsymbol{y}_{ij}$ such that for all $j \in [\ell]$: $\boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$

Distributing the Setup

Sample $\boldsymbol{W}_i$ together with short $\boldsymbol{y}_{ij}$ such that for all $j \in [\ell]$: $\boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$

$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$

$\boldsymbol{B} \leftarrow \mathbb{Z}_q^{n \times m}$

$\boldsymbol{p}$

$\boldsymbol{r}_1 \cdots \boldsymbol{r}_\ell$

$\boldsymbol{Z}_1 \leftarrow \mathbb{Z}_q^{n \times m}$

$\vdots$

$\boldsymbol{Z}_k \leftarrow \mathbb{Z}_q^{n \times m}$

$\forall t \in [k], j \in [\ell]$:

$$\boldsymbol{u}_{tj} \leftarrow \boldsymbol{A}^{-1}(\boldsymbol{Z}_t \boldsymbol{r}_j)$$

Public parameters

Sample $d \leftarrow \{0,1\}^k$

$$\boldsymbol{W}_i = \sum_{t \in [k]} d_t \boldsymbol{Z}_t$$

Then $\boldsymbol{A} \cdot \underbrace{\sum_{t \in [k]} d_t \boldsymbol{u}_{tj}}_{\boldsymbol{y}_{ij}} = \sum_{t \in [k]} d_t \boldsymbol{Z}_t \boldsymbol{r}_j = \boldsymbol{W}_i \boldsymbol{r}_j$

Public parameters contain “pre-sampled” public keys, and a user key is a random combination of the pre-sampled keys

A More General View

Sample $\boldsymbol{W}_i$ together with short $\boldsymbol{y}_{ij}$ such that for all $j \in [\ell]$: $\boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$

Approach can be described more compactly as sampling a solution to the linear system

$$\left[\begin{array}{ccc|ccc} \boldsymbol{A} & & & -\boldsymbol{Z}_1\boldsymbol{r}_1 & \cdots & -\boldsymbol{Z}_k\boldsymbol{r}_1 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}_1\boldsymbol{r}_\ell & \cdots & -\boldsymbol{Z}_k\boldsymbol{r}_\ell \end{array}\right] \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ d_1 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

Then, for all $j \in [\ell]$:

$$\boldsymbol{A}\boldsymbol{y}_{ij} - \sum_{t \in [k]} d_t \boldsymbol{Z}_t \boldsymbol{r}_j = 0 \quad \Longrightarrow \quad \boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j \qquad\qquad \boldsymbol{W}_i = \sum_{t \in [k]} d_t \boldsymbol{Z}_t$$

A More General View

Sample $\boldsymbol{W}_i$ together with short $\boldsymbol{y}_{ij}$ such that for all $j \in [\ell]$: $\quad \boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i\boldsymbol{r}_j$

Approach can be described more compactly as sampling a solution to the linear system

$$\left[\begin{array}{ccc|ccc} \boldsymbol{A} & & & -\boldsymbol{Z}_1\boldsymbol{r}_1 & \cdots & -\boldsymbol{Z}_k\boldsymbol{r}_1 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}_1\boldsymbol{r}_\ell & \cdots & -\boldsymbol{Z}_k\boldsymbol{r}_\ell \end{array}\right] \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ d_1 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

More compactly: $\boldsymbol{Z} = [\boldsymbol{Z}_1 \mid \boldsymbol{Z}_2 \mid \cdots \mid \boldsymbol{Z}_k]$

$$\left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right] \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \mathbf{0} \quad \Longrightarrow \quad \begin{aligned} \boldsymbol{A}\boldsymbol{y}_{ij} &= \boldsymbol{Z}\big(\boldsymbol{I} \otimes \boldsymbol{r}_j\big)\boldsymbol{d} = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})\boldsymbol{r}_j \\ \boldsymbol{W}_i &= \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I}) \end{aligned}$$

Distributing the Setup

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

Approach: Each user will choose their own $\boldsymbol{W}_i$, everything else will be in the public parameters

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$\boldsymbol{W}_1$

$\boldsymbol{W}_2$

$\boldsymbol{W}_3$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

For correctness, each user also needs to generate a secret key and cross-terms

$$\forall i \neq j : \boldsymbol{A}\boldsymbol{y}_{i,j} = \boldsymbol{W}_i \boldsymbol{r}_j$$

$$\boldsymbol{A}\boldsymbol{y}_{i,i} = \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{W}_i \boldsymbol{r}_i$$

$$\left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right] \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{array}{c} \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i \\ \vdots \\ \boldsymbol{0} \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \leftarrow \text{row } i \\ \\ \\ \end{array}$$

Set $\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

Suppose LWE is hard with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx \text{random}$$

Selective security

Adversary

Challenger

$S \subseteq [\ell]$

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

Adversary declares challenge set upfront

How do we simulate the public keys and the challenge ciphertext?

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j\right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

Suppose LWE is hard with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{s}^{\mathrm{T}} \boldsymbol{A} \approx \text{random}$$

How do we simulate the public keys and the challenge ciphertext?

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A}$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j \right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

$\mathrm{pk}_i : \ \boldsymbol{W}_i, \ \{\boldsymbol{y}_{ij}\}_{j \neq i}$ where $\boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$

Can be sampled using trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell \cdot \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i \\ \vdots \\ \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

Suppose LWE is hard with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx \text{random}$$

How do we simulate the public keys and the challenge ciphertext?

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j\right)$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rceil$$

Set $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{r}$

$\mathrm{pk}_i : \ \boldsymbol{W}_i, \ \{\boldsymbol{y}_{ij}\}_{j \neq i}$ where $\boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$

Can be sampled using trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell \cdot \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i \\ \vdots \\ \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

Suppose LWE is hard with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx \text{random}$$

How do we simulate the public keys and the challenge ciphertext?

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B} + \sum_{j \in S} \boldsymbol{W}_j\right)$$

Set $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} - \sum_{j \in S} \boldsymbol{W}_j$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r} + \mu \cdot \lfloor q/2 \rceil$$

Set $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{r}$

$\mathrm{pk}_i :\ \boldsymbol{W}_i,\ \{\boldsymbol{y}_{ij}\}_{j \neq i}$ where $\boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$

Can be sampled using trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell \cdot \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i \\ \vdots \\ \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

Suppose LWE is hard with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{s}^{\mathrm{T}} \boldsymbol{A} \approx \text{random}$$

How do we simulate the public keys and the challenge ciphertext?

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A}$$

$$\mathrm{pk}_i : \ \boldsymbol{W}_i, \ \{\boldsymbol{y}_{ij}\}_{j \neq i} \text{ where } \boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A}\boldsymbol{R}$$

Set $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} - \sum_{j \in S} \boldsymbol{W}_j$

Can be sampled using trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell \cdot \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i \\ \vdots \\ \boldsymbol{0} \end{bmatrix}$$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}} \boldsymbol{A}\boldsymbol{r} + \mu \cdot \lfloor q/2 \rceil$$

Set $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{r}$

$$\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

How do we simulate the

There's a **circularity** here!

Public key components $\boldsymbol{W}_i, \boldsymbol{y}_{ij}$ depend on $\boldsymbol{B}$, so we cannot program B to be a function of $\boldsymbol{W}_i$

More generally, reduction algorithm knows the secret keys for all users, so the overall strategy is problematic

$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$

$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{R}$ Set $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} - \sum_{j \in S} \boldsymbol{W}_j$

$c_3 \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r} + \mu \cdot \lfloor q/2 \rceil$ Set $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{r}$

Can be sampled using trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell \cdot \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i \\ \vdots \\ \boldsymbol{0} \end{bmatrix} \longrightarrow \boldsymbol{0}$$

Target $\boldsymbol{0}$ in *all* blocks

$\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

Suppose LWE is hard with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx \text{random}$$

How do we simulate the public keys and the challenge ciphertext?

$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$ pk

Distributions of $\boldsymbol{y}_{ij}$ for $j \neq i$ and of $\boldsymbol{d}$ is statistically indistinguishable to original distribution

$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{R}$ Set $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} - \sum_{j \in S} \boldsymbol{W}_j$

$c_3 \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r} + \mu \cdot \lfloor q/2 \rceil$ Set $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{r}$

Target $\mathbf{0}$ in *all* blocks

$$\boldsymbol{V}_\ell \cdot \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i \\ \vdots \\ \mathbf{0} \end{bmatrix} \longrightarrow \mathbf{0}$$

$$\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$$

Proving Selective Security

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

No more circularity!

- First sample $\boldsymbol{W}_i$ using $\boldsymbol{V}_\ell$
- Then set $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} - \sum_{j \in S} \boldsymbol{W}_j$

How do we simulate the public keys and the challenge ciphertext?

$$\boldsymbol{c}_1^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$$

$$\boldsymbol{c}_2^{\mathrm{T}} \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{R}$$

Set $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} - \sum_{j \in S} \boldsymbol{W}_j$

$$c_3 \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r} + \mu \cdot \lfloor q/2 \rceil$$

Set $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{r}$

$\mathrm{pk}_i : \; \boldsymbol{W}_i, \; \{\boldsymbol{y}_{ij}\}_{j \neq i}$ where $\boldsymbol{A}\boldsymbol{y}_{ij} = \boldsymbol{W}_i \boldsymbol{r}_j$

Can be sampled using trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell \cdot \begin{bmatrix} \boldsymbol{y}_{i1} \\ \vdots \\ \boldsymbol{y}_{i\ell} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}$$

Target $\boldsymbol{0}$ in *all* blocks

$$\boldsymbol{W}_i = \boldsymbol{Z}(\boldsymbol{d} \otimes \boldsymbol{I})$$

Completing the Proof

Public parameters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{p}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell, \boldsymbol{V}_\ell$, trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{V}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_1) \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & -\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_\ell) \end{array}\right]$$

Suppose LWE is hard with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx \text{random}$$

This is not the ℓ-succinct LWE trapdoor!

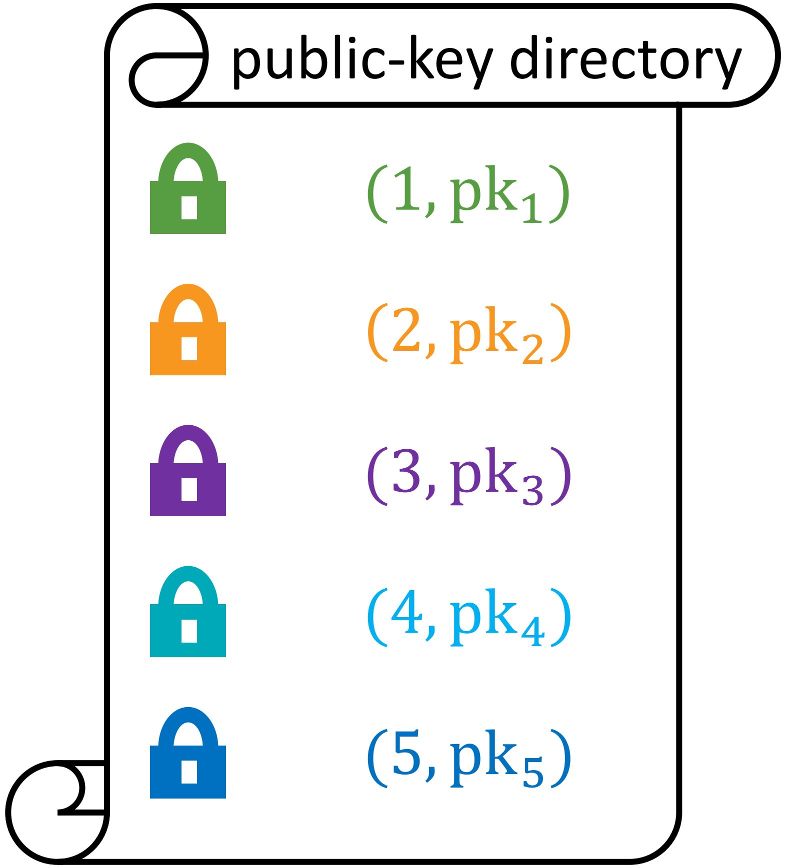
$$\boldsymbol{D}_\ell = \left[\begin{array}{ccc|c} \boldsymbol{A} & & & \boldsymbol{U}_1 \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & \boldsymbol{U}_\ell \end{array}\right]$$

Distribution of $\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_i)$ not independent uniform (given $\boldsymbol{Z}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell$)

Given a trapdoor for $\boldsymbol{D}_{\ell'}$ where $\ell' \geq O(\ell n \log q)$, we can derive $\boldsymbol{Z}, \boldsymbol{r}_1, \dots, \boldsymbol{r}_\ell$ and a trapdoor for the matrix $\boldsymbol{V}_\ell$ (with polynomial loss in parameters)

[see paper for details]

Summary



Broadcast encryption without a central authority

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:

- Scheme with short CRS and public keys
- Proving security from plain LWE
- Cryptanalysis of ℓ-succinct LWE

Thank you!