Registered ABE and Adaptively-secure Broadcast Encryption from Succinct LWE

Jeffrey Champion, Yao-Ching Hsieh, David Wu

Unbounded Distributed Broadcast Encryption and Registered ABE from Succinct LWE

Hoeteck Wee and David Wu

Registered ABE and Adaptively-secure Broadcast Encryption from Succinct LWE

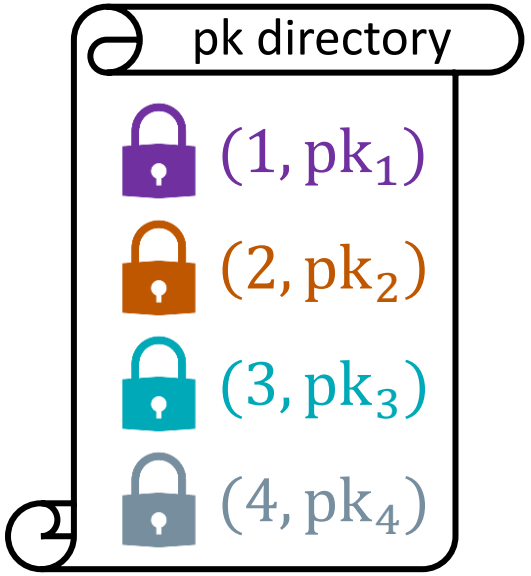
Jeffrey Champion, Yao-Ching Hsieh, David Wu

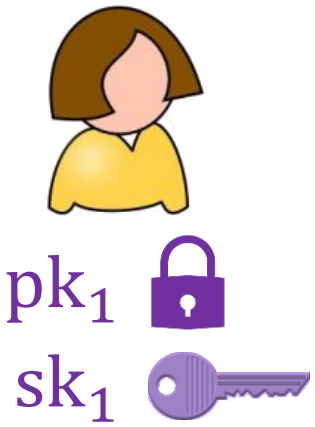
Unbounded Distributed Broadcast Encryption and Registered ABE from Succinct LWE

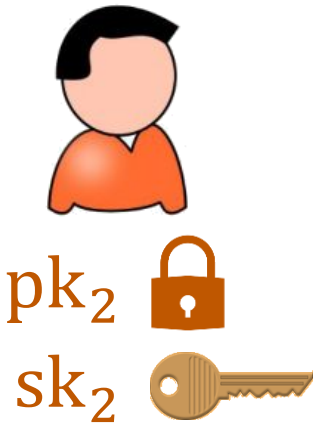
Hoeteck Wee and David Wu

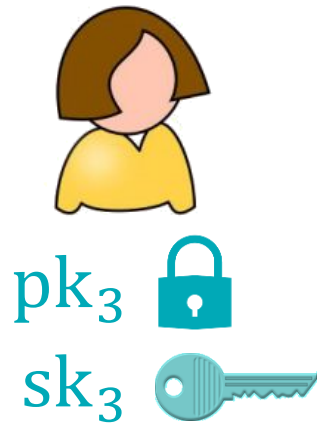
Distributed Broadcast Encryption

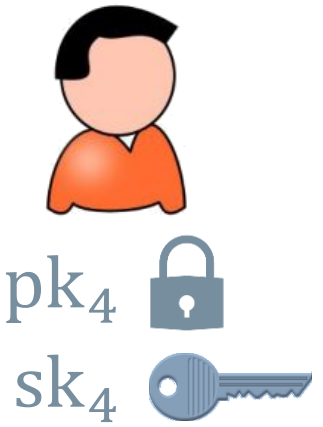
[WQZD10, BZ14]











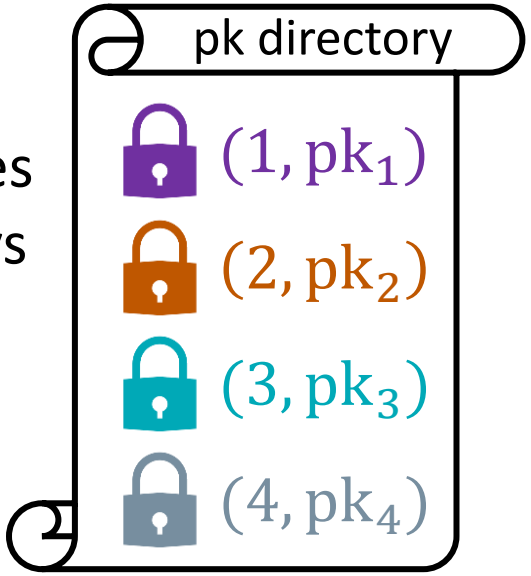
Distributed Broadcast Encryption

[WQZD10, BZ14]

message m

$S = \{1,4\}$

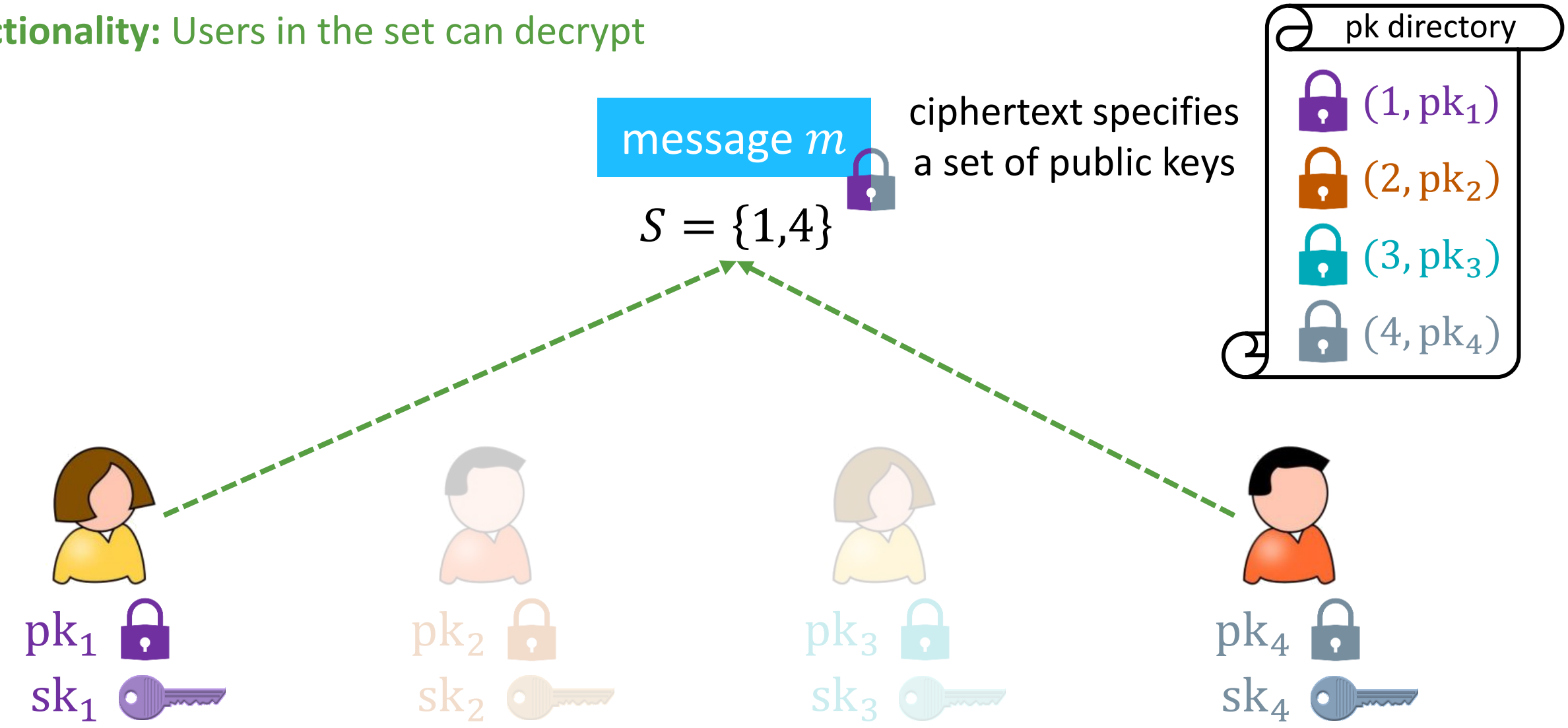
ciphertext specifies a set of public keys



Distributed Broadcast Encryption

[WQZD10, BZ14]

Functionality: Users in the set can decrypt

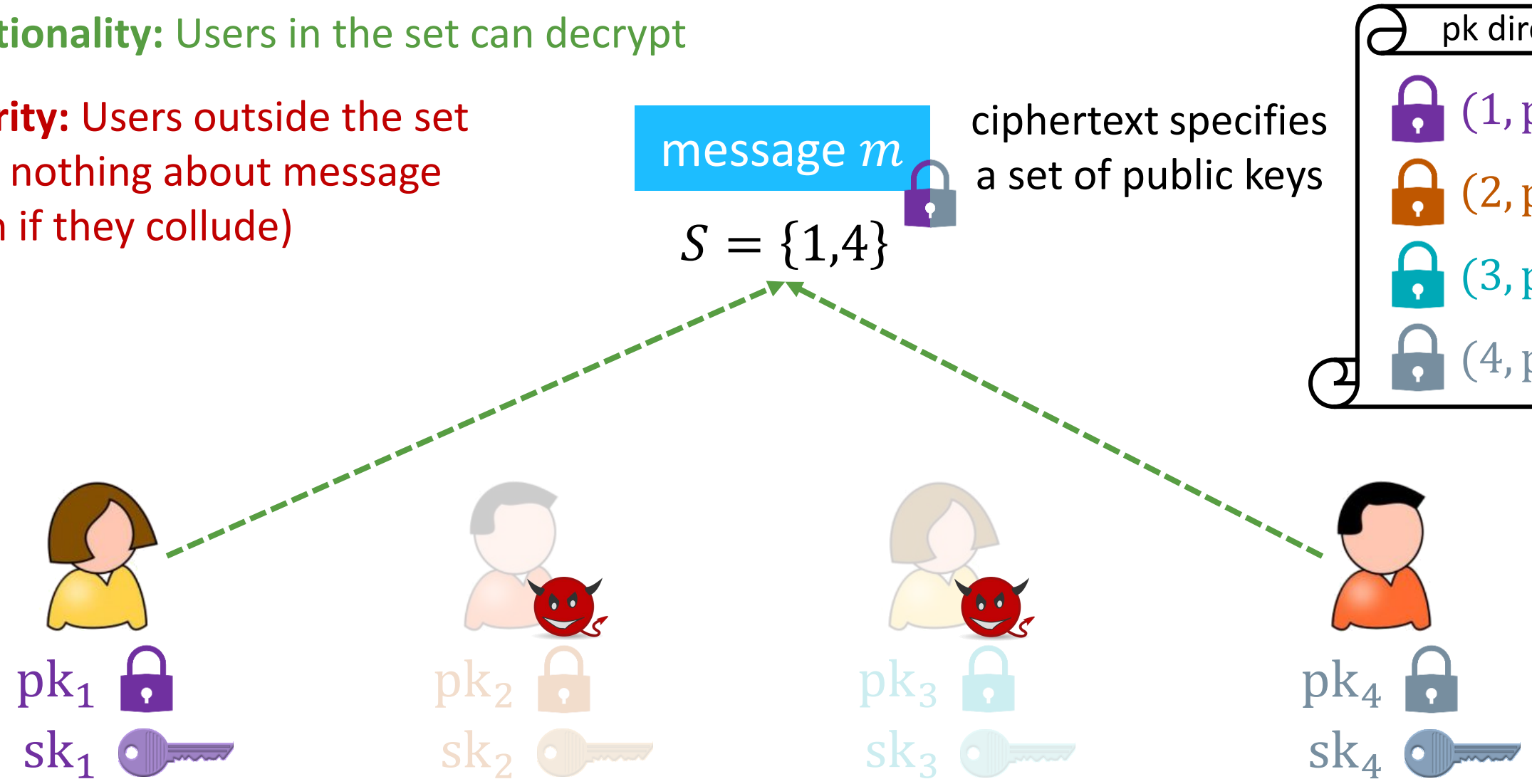


Distributed Broadcast Encryption

[WQZD10, BZ14]

Functionality: Users in the set can decrypt

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



Distributed Broadcast Encryption

[WQZD10, BZ14]

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

ciphertext specifies a set of public keys

$S = \{1,4\}$

pk directory

$(1, \mathrm{pk}_1)$

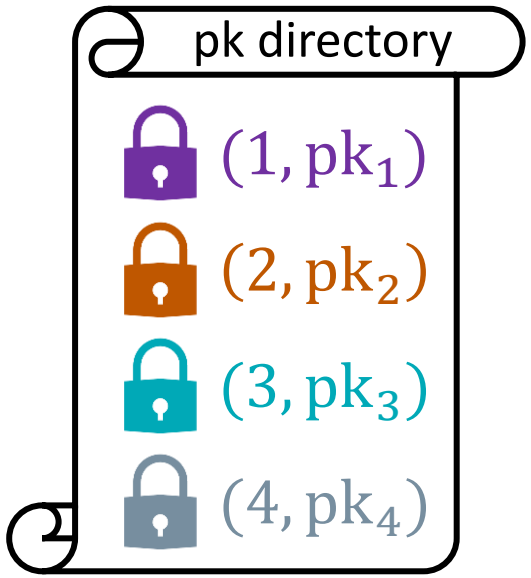
$(2, \mathrm{pk}_2)$

$(3, \mathrm{pk}_3)$

$(4, \mathrm{pk}_4)$

Distributed Broadcast Encryption (DBE)

[WQZD10, BZ14]



public parameters

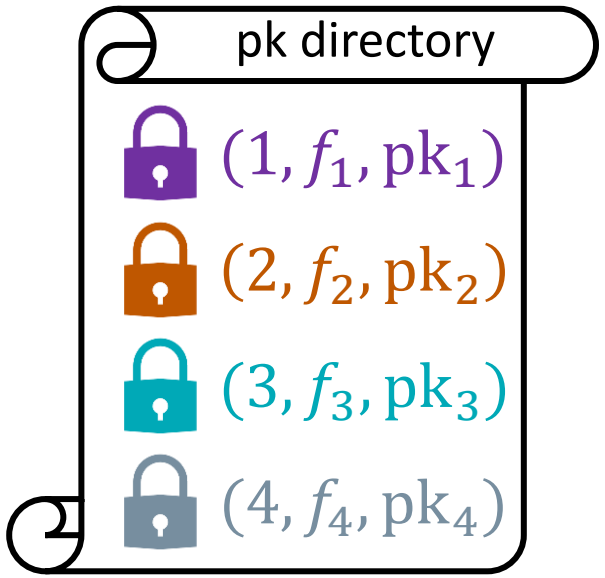
$$\mathrm{KeyGen}(\mathrm{pp}, i) \to (\mathrm{pk}_i, \mathrm{sk}_i)$$

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

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

- Trustless version of traditional broadcast encryption [FN93]
- DBE implies traditional broadcast encryption with a long public key

Distributed Attribute-Based Encryption



public parameters

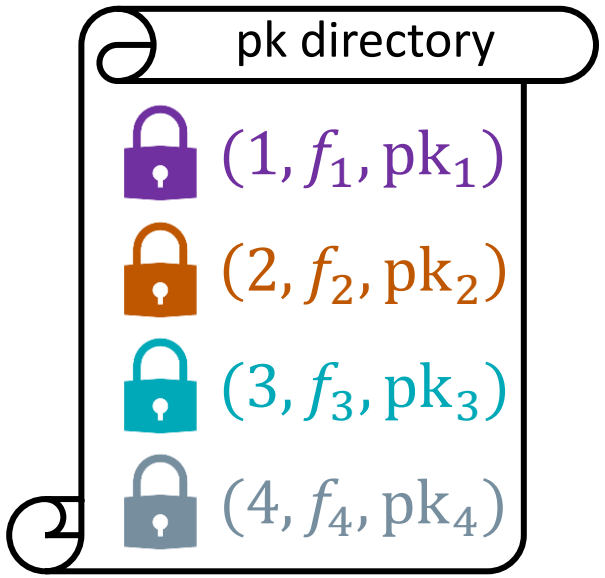
$$\mathrm{KeyGen}(\mathrm{pp}, i, f_i) \to (\mathrm{pk}_i, \mathrm{sk}_i)$$

size $\mathrm{poly}(\lambda, \log N)$

$$\mathrm{Encrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, x, m\right) \to \mathrm{ct}$$

$$\mathrm{Decrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, \mathrm{sk}_i, x, \mathrm{ct}\right) \to m$$

Distributed Attribute-Based Encryption



Functionality: Secret key for f_i recovers m if $f_i(x) = 0$

public parameters

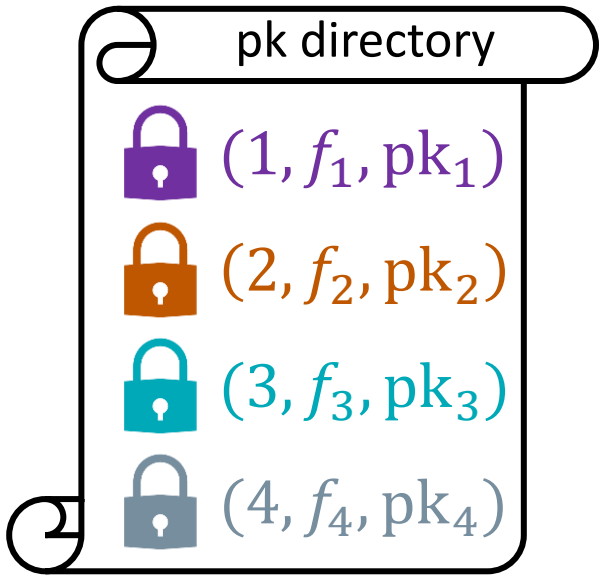
$$\mathrm{KeyGen}(\mathrm{pp}, i, f_i) \to (\mathrm{pk}_i, \mathrm{sk}_i)$$

size $\mathrm{poly}(\lambda, \log N)$

$$\mathrm{Encrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, x, m\right) \to \mathrm{ct}$$

$$\mathrm{Decrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, \mathrm{sk}_i, x, \mathrm{ct}\right) \to m$$

Distributed Attribute-Based Encryption



Functionality: Secret key for f_i recovers m if $f_i(x) = 0$

Security: Users with $f_i(x) = 1$ learn nothing about m

public parameters

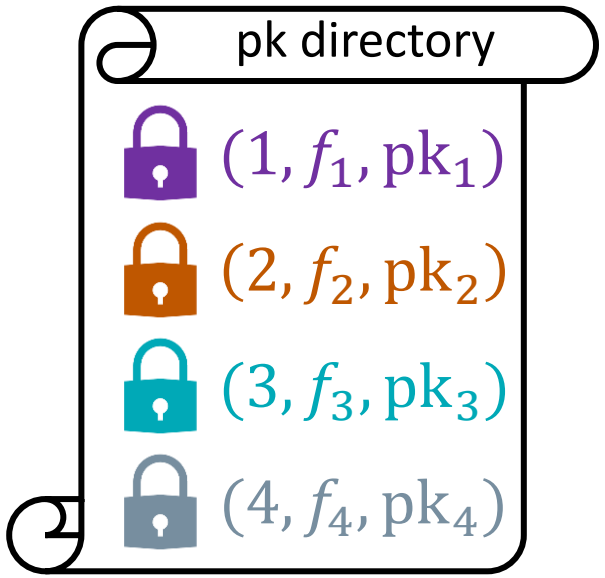
$$\mathrm{KeyGen}(\mathrm{pp}, i, f_i) \to (\mathrm{pk}_i, \mathrm{sk}_i)$$

size $\mathrm{poly}(\lambda, \log N)$

$$\mathrm{Encrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, x, m\right) \to \mathrm{ct}$$

$$\mathrm{Decrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, \mathrm{sk}_i, x, \mathrm{ct}\right) \to m$$

Distributed Attribute-Based Encryption



public parameters

$$\mathrm{KeyGen}(\mathrm{pp}, i, f_i) \to (\mathrm{pk}_i, \mathrm{sk}_i)$$

size $\mathrm{poly}(\lambda, \log N)$

$$\mathrm{Encrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, x, m\right) \to \mathrm{ct}$$

$$\mathrm{Decrypt}\left(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}, \mathrm{sk}_i, x, \mathrm{ct}\right) \to m$$

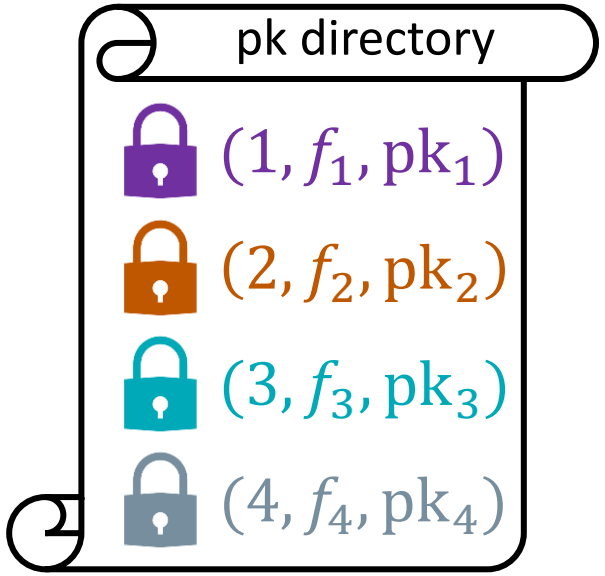
Functionality: Secret key for f_i recovers m if $f_i(x) = 0$

Security: Users with $f_i(x) = 1$ learn nothing about m

User set is fixed but encryption/decryption time scales at least linearly with N!

Registered ABE

[GHMR18, HLWW23]



$$\mathrm{KeyGen}(\mathrm{pp}, i, f_i) \to (\mathrm{pk}_i, \mathrm{sk}_i)$$

$$\mathrm{Aggregate}\big(\mathrm{pp}, \{\mathrm{pk}_i\}_{i\in[N]}\big) \to (\mathrm{mpk}, \{\mathrm{hint}_i\}_{i\in[N]})$$

size $\mathrm{poly}(\lambda, \log N)$

size $\mathrm{poly}(\lambda, \log N)$

$$\mathrm{Encrypt}(\mathrm{mpk}, x, m) \to \mathrm{ct}$$

$$\mathrm{Decrypt}(\mathrm{hint}_i, \mathrm{sk}_i, x, \mathrm{ct}) \to m$$

ABE without a central authority

Prior Work

- **Registered Attribute-Based Encryption:**
  - Indistinguishability obfuscation [HLWW23]
  - Witness encryption [FWW23]
  - Pairings [HLWW23,ZZGQ23,GLWW24,AT24]]: require a structured setup, a bounded number of users, and limited to NC1 circuits
- **Distributed Broadcast Encryption:**
  - Pairings [KMW23,GKPW24]: require a structured setup and bounded number of users
  - Succinct LWE [CW24]: requires a structured setup, bounded number of users, and limited to selective security

Prior Work

- **Registered Attribute-Based Encryption:**
  - Indistinguishability obfuscation [HLWW23]
  - Witness encryption [FWW23]
  - Pairings [HLWW23,ZZGQ23,GLWW24,AT24]]: require a structured setup, a bounded number of users, and limited to NC1 circuits
- **Distributed Broadcast Encryption:**
  - Pairings [KMW23,GKPW24]: require a structured setup and bounded number of users
  - Succinct LWE [CW24]: requires a structured setup, bounded number of users, and limited to selective security

Can we do more from lattice assumptions?

Our Results (Distributed Broadcast Encryption)

- [CHW25]: Adaptively-secure DBE for N users from succinct LWE in the random oracle model
  - **Public parameter size:** N^2
  - **User public key size:** N
  - **Master public key, helper key, and secret key size:** $O(1)$
  - **Ciphertext size:** $O(1)$

First adaptive scheme from a falsifiable lattice assumption!

- [WW25]: Selectively-secure DBE for unbounded users from succinct LWE
  - **Public parameter size:** $O(1)$ transparent setup from decomposed LWE
  - **User public key size:** $O(1)$
  - **Master public key, helper key, and secret key size:** $O(1)$
  - **Ciphertext size:** $O(1)$

First scheme with unbounded users or transparent setup without iO/WE!

$\mathrm{poly}(\lambda, \log N)$ factors omitted

Our Results (Registered ABE)

- [CHW25]: Registered ABE for input length ℓ depth d circuits and N users from succinct LWE in the random oracle model
  - **Public parameter size:** $N^2 + \ell^2$
  - **User public key size:** N
  - **Master public key, helper key, and secret key size:** $O(1)$
  - **Ciphertext size:** $O(1)$

First scheme for circuits without iO/WE!

- [WW25]: Registered ABE for input length ℓ depth d circuits and unbounded users from succinct LWE in the random oracle model
  - **Public parameter size:** $O(1)$ transparent setup from decomposed LWE
  - **User public key size:** $O(1)$
  - **Master public key, helper key, and secret key size:** $O(1)$
  - **Ciphertext size:** $O(1)$

First scheme with unbounded users or transparent setup without iO/WE!

$\mathrm{poly}(\lambda, d, \log N)$ factors omitted

Our Results (Registered ABE)

- [CHW25]: Registered ABE for input length ℓ depth d circuits and N users from succinct LWE in the random oracle model
  - **Public parameter size:** $N^2 + \ell^2$
  - **User public key size:** N
  - **Master public key, helper key, and secret key size:** $O(1)$
  - **Ciphertext size:** $\boxed{O(1)}$ optimal up to the $\mathrm{poly}(d)$ factor!

- [WW25]: Registered ABE for input length ℓ depth d circuits and unbounded users from succinct LWE in the random oracle model
  - **Public parameter size:** $O(1)$ transparent setup from decomposed LWE
  - **User public key size:** $O(1)$
  - **Master public key, helper key, and secret key size:** $O(1)$
  - **Ciphertext size:** $\boxed{O(1)}$ optimal up to the $\mathrm{poly}(d)$ factor!

$\mathrm{poly}(\lambda, d, \log N)$ factors omitted

Level of efficiency only known from iO previously!

Key Challenges

1. **Functionality:** want users to generate public keys that can be *succinctly aggregated* and secret keys that facilitate correctness

2. **Security:** need to simulate a challenge ciphertext where the public keys are *adversarially chosen*

Key Challenges

1. **Functionality:** want users to generate public keys that can be *succinctly aggregated* and secret keys that facilitate correctness

2. **Security:** need to simulate a challenge ciphertext where the public keys are *adversarially chosen*

In the lattice setting, ciphertexts
often have components of the form:

What if M is chosen adversarially?

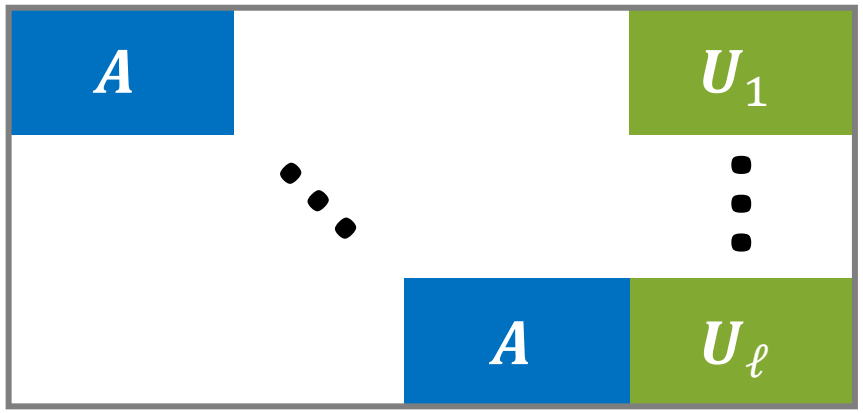
Succinct LWE Family of Assumptions

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

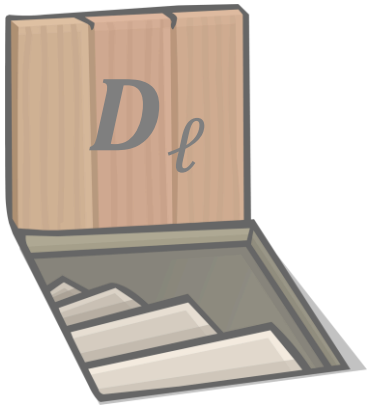
$$\boldsymbol{s}^T \boldsymbol{A} + \boldsymbol{e}^T \approx \boldsymbol{z}^T$$

$\boldsymbol{s}, \boldsymbol{z}$ are uniform, $\boldsymbol{e}$ is Gaussian

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



and

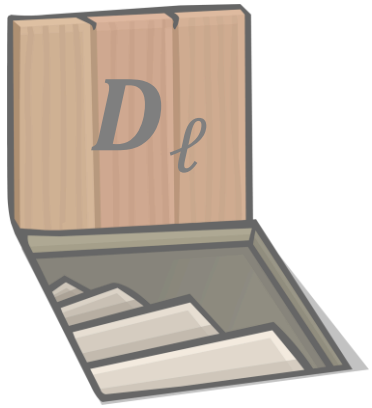


Succinct LWE Family of Assumptions

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

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

$\boldsymbol{s}, \boldsymbol{z}$ are uniform, $\boldsymbol{e}$ is Gaussian

given $\boldsymbol{D}_\ell = \begin{bmatrix} \boldsymbol{A} & & \boldsymbol{U}_1 \\ & \ddots & \vdots \\ & \boldsymbol{A} & \boldsymbol{U}_\ell \end{bmatrix}$ and 

BASIS ⇒ Succinct ⇒ Decomposed

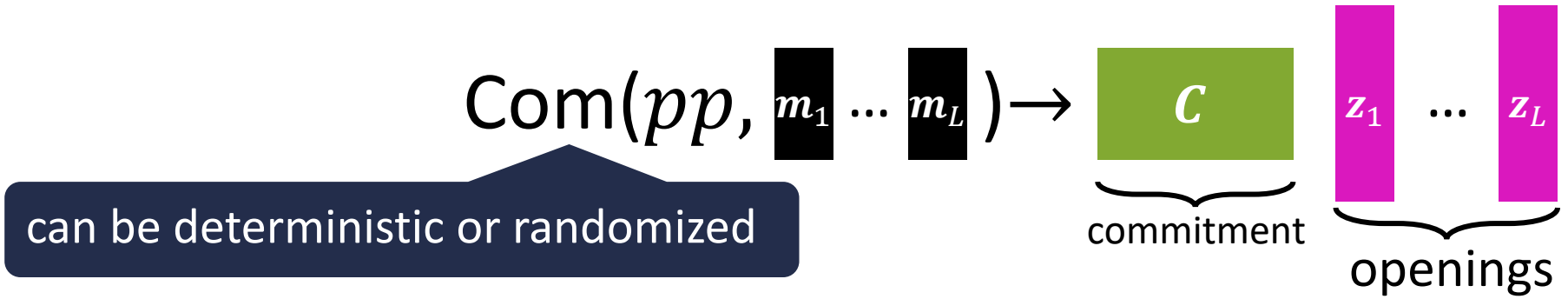
Instantiations

- BASIS [WW23]: $\boldsymbol{A}$ is uniform, $\boldsymbol{U}_i$ is structured
- Succinct LWE [Wee24]: $\boldsymbol{A}$ is uniform, $\boldsymbol{U}_i$ is uniform
- Decomposed LWE [AMR25]: $\boldsymbol{A}$ is structured, $\boldsymbol{U}_i$ is uniform, trapdoor is *public*

Matrix Commitments

[Wee25, adapted]

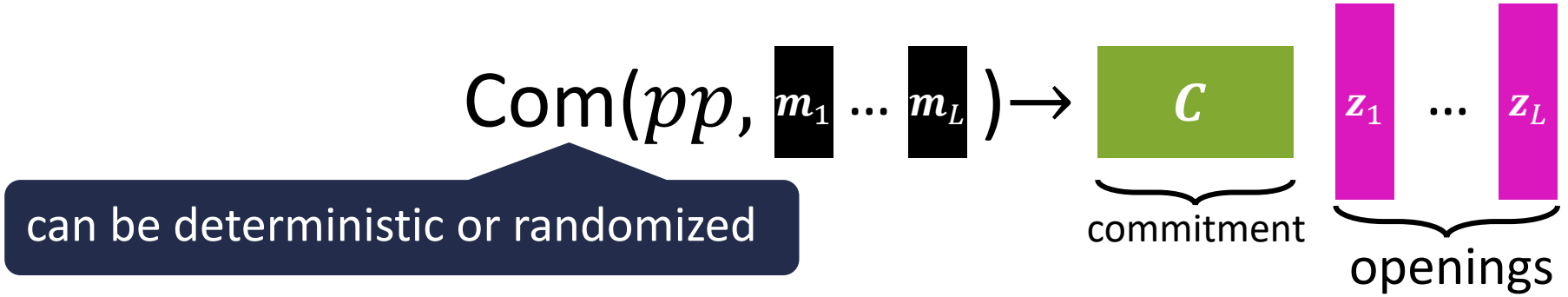
"vector commitment to vectors"



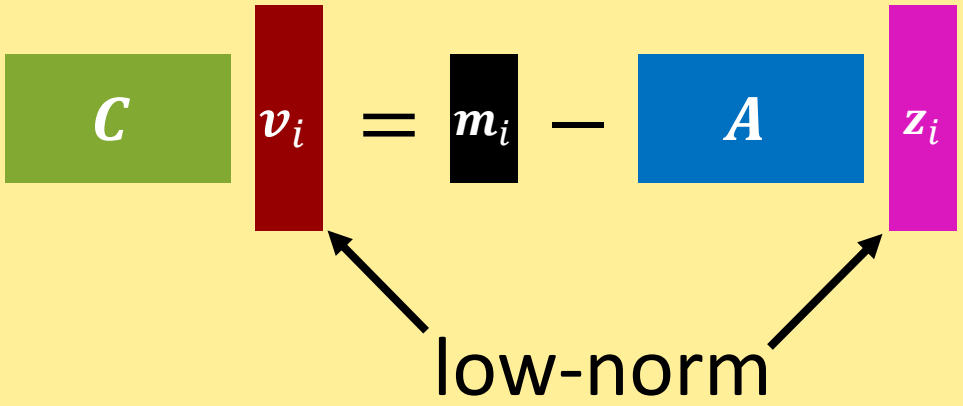
Matrix Commitments

[Wee25, adapted]

"vector commitment to vectors"



For all $i \in [L]$:



pp describes $\boldsymbol{A}$ and $\boldsymbol{v}_i$ for $i \in [L]$

Matrix Commitments

[Wee25, adapted]

"vector commitment to vectors"

$\mathrm{Com}(pp, \boldsymbol{m}_1 \ldots \boldsymbol{m}_L) \to \boldsymbol{C}, \boldsymbol{z}_1 \ldots \boldsymbol{z}_L$

can be deterministic or randomized

$\boldsymbol{C}$: commitment

$\boldsymbol{z}_1 \ldots \boldsymbol{z}_L$: openings

For all $i \in [L]$:

$$\boldsymbol{C}\,\boldsymbol{v}_i = \boldsymbol{m}_i - \boldsymbol{A}\,\boldsymbol{z}_i$$

low-norm

pp describes $\boldsymbol{A}$ and $\boldsymbol{v}_i$ for $i \in [L]$

Security: $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}} \approx \boldsymbol{z}^{\mathrm{T}}$ given pp

DBE: The Basic Approach

DBE: The Basic Approach

- Each user will sample a (dual-Regev) key pair and publish the public key

DBE: The Basic Approach

- Each user will sample a (dual-Regev) key pair and publish the public key

- Encryption does the following:
  1. Merkle hashes the public keys of users in the set (via the matrix commitment)
  2. Encrypts the message with respect to the hash using dual-Regev

DBE: The Basic Approach

- Each user will sample a (dual-Regev) key pair and publish the public key

- Encryption does the following:
  1. Merkle hashes the public keys of users in the set (via the matrix commitment)
  2. Encrypts the message with respect to the hash using dual-Regev

- To decrypt, user $i \in S$ can use the local opening at position i to derive an encryption of the message under their public key and decrypt using their secret key

DBE: The Basic Approach

- Each user will sample a (dual-Regev) key pair and publish the public key

- Encryption does the following:
  1. Merkle hashes the public keys of users in the set (via the matrix commitment)
  2. Encrypts the message with respect to the hash using dual-Regev

- To decrypt, user $i \in S$ can use the local opening at position i to derive an encryption of the message under their public key and decrypt using their secret key

Can view this as an algebraic version of the witness encryption approach in [FWW23]

DBE For Unbounded Users [WW25]

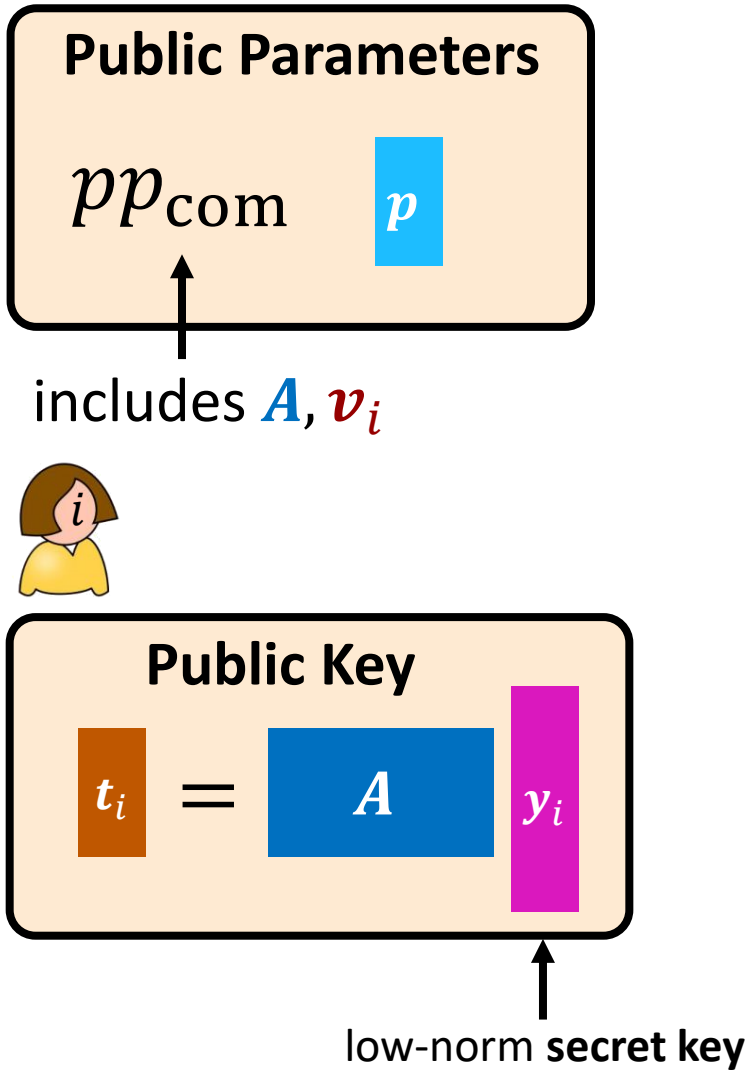
Public Parameters

pp_{com} p

includes A, v_i

deterministic commitment [Wee25] has fixed-size public parameters!

DBE For Unbounded Users [WW25]



DBE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

includes A, v_i

Public Key

$t_i = A \; y_i$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to S :

Noise omitted

s^T p

masks b

DBE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

includes A, v_i

Public Key

$t_i = A\ y_i$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to S :

Noise omitted

$s^T\ C_S$ $\quad$ $s^T\ p$

masks b

commit to $p + t_j$ at position j
when $j \in S$ and $\mathbf{0}$ otherwise

can efficiently commit to exponential number of vectors if poly many are nonzero

DBE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

includes A, v_i

Public Key

$t_i = A\ y_i$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to S :

$s^T\ A$

$s^T\ C_S$

commit to $p + t_j$ at position j when $j \in S$ and $\mathbf{0}$ otherwise

$s^T\ p$

masks b

Noise omitted

DBE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

includes A, v_i

Public Key

$t_i = A\ y_i$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to S :

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

masks b

Noise omitted

For all $i \in S$: $C_S v_i = p + t_i - A z_i$

public and low-norm

DBE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

includes A, v_i

Public Key

$t_i = A\ y_i$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to S :

$s^T\ A$ $s^T\ C_S$ $s^T\ p$

Noise omitted

masks b

For all $i \in S$: $C_S v_i = p + t_i - Az_i$

$$s^T C_S v_i \approx s^T p + s^T t_i - s^T A z_i$$

DBE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

includes A, v_i

only user i can compute $s^{\mathrm{T}} t_i \approx s^{\mathrm{T}} A y_i$

Public Key

$t_i = A \; y_i$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to S :

$s^{\mathrm{T}} A$ $s^{\mathrm{T}} C_S$ $s^{\mathrm{T}} p$

masks b

Noise omitted

For all $i \in S$: $C_S v_i = p + t_i - A z_i$

$$s^{\mathrm{T}} C_S v_i \approx s^{\mathrm{T}} p + s^{\mathrm{T}} t_i - s^{\mathrm{T}} A z_i$$

$$\underbrace{s^{\mathrm{T}} C_S v_i + s^{\mathrm{T}} A z_i}_{\text{public}} \approx \underbrace{s^{\mathrm{T}} p + s^{\mathrm{T}} t_i}_{\text{dual-Regev ciphertext under } \mathrm{pk}_i}$$

Extending to Registered ABE [CHW25]

- Each user will sample a (dual-Regev) key pair and publish the public key combined with a function encoding

- Aggregation/Encryption does the following:
  1. Merkle hashes the public keys of all registered users (via the matrix commitment) and publishes the hash as the master public key
  2. Encrypts the message with respect to the hash using dual-Regev and provides a policy check that depends on the attribute

- To decrypt, registered user i can use the local opening at position i to derive an encryption of the message under their combined public key and decrypt using their secret key iff the policy check passes

Lattice-based Homomorphic Computation

[GSW13,BGGHNSVV14]

Given any input $\boldsymbol{x} \in \{0,1\}^{\ell}$, function $f: \{0,1\}^{\ell} \to \{0,1\}$, and matrix $\boldsymbol{B}$ of appropriate size, there exists a low-norm matrix $\boldsymbol{H}_{\boldsymbol{B},f,x}$ such that the following holds:

Lattice-based Homomorphic Computation

[GSW13,BGGHNSVV14]

Given any input $\boldsymbol{x} \in \{0,1\}^{\ell}$, function $f: \{0,1\}^{\ell} \to \{0,1\}$, and matrix $\boldsymbol{B}$ of appropriate size, there exists a low-norm matrix $\boldsymbol{H}_{\boldsymbol{B},f,x}$ such that the following holds:

$$\boldsymbol{B} - \boldsymbol{x}^T \otimes \boldsymbol{G} \quad \boldsymbol{H}_{\boldsymbol{B},f,x} \quad = \quad \boldsymbol{B}_f \quad - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

public gadget matrix

encoding of $\boldsymbol{x}$ with respect to $\boldsymbol{B}$

encoding of $f(\boldsymbol{x})$ with respect to $\boldsymbol{B}_f$

Lattice-based Homomorphic Computation

[GSW13,BGGHNSVV14]

Given any input $\boldsymbol{x} \in \{0,1\}^{\ell}$, function $f: \{0,1\}^{\ell} \to \{0,1\}$, and matrix $\boldsymbol{B}$ of appropriate size, there exists a low-norm matrix $\boldsymbol{H}_{\boldsymbol{B},f,x}$ such that the following holds:

$$\boldsymbol{B} - \boldsymbol{x}^T \otimes \boldsymbol{G} \quad \boldsymbol{H}_{\boldsymbol{B},f,x} \quad = \quad \boldsymbol{B}_f \quad - f(x) \cdot \boldsymbol{G}$$

Given additionally a vector $\boldsymbol{d}$, there exists a low-norm vector $\boldsymbol{h}_{\boldsymbol{B},f,x,\boldsymbol{d}} = \boldsymbol{H}_{\boldsymbol{B},f,x} \boldsymbol{G}^{-1}(\boldsymbol{d})$ such that:

$$\boldsymbol{B} - \boldsymbol{x}^T \otimes \boldsymbol{G} \quad \boldsymbol{h} \quad = \quad \boldsymbol{b}_{f,\boldsymbol{d}} \quad - f(x) \cdot \boldsymbol{d}$$

Registered ABE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

Public Key

$t_i = A\ y_i$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$s^T\ A$

$s^T\ C_S$

commit to $p + t_j$ at position j when $j \in S$ and $\mathbf{0}$ otherwise

$s^T\ p$

masks b

Registered ABE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

master public key

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

s^{T} A $\quad$ s^{T} C_{all} $\quad$ s^{T} p

masks b

commit to $p + t_j$ at position $j \in [N]$

Public Key

$t_i = A \; y_i$

low-norm **secret key**

Registered ABE For Unbounded Users [WW25]

Public Parameters

pp_{com} p

B

Public Key

$t_i = A \; y_i$

low-norm **secret key**

master public key

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

s^T A

s^T C_{all}

s^T p

masks b

s^T $B - x^T \otimes G$

commit to $p + t_j$ at position $j \in [N]$

grows with ℓ but can be compressed via [Wee25]

Registered ABE For Unbounded Users [WW25]

can be derived from random oracle

Pub

pp_{com} p d_i

B $i \in [N]$

Also includes NIZK PoK of y_i

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

s^T A

s^T C_{all}

s^T p

masks b

s^T $B - x^T \otimes G$

commit to $p + t_j$ at position $j \in [N]$

Registered ABE For Unbounded Users [WW25]

Public Parameters

pp_{com} p d_i $i \in [N]$

B

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Noise omitted

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

$s^T A$

$s^T C_{all}$

$s^T p$

masks b

$s^T (B - x^T \otimes G)$

commit to $p + t_j$ at position $j \in [N]$

Recall:

$(B - x^T \otimes G)\, h = b_{f,d} - f(x) \cdot d$

$h = h_{B,f,x,d}$ is low-norm

Registered ABE For Unbounded Users [WW25]

Public Parameters

pp_{com} p d_i $i \in [N]$

B

Noise omitted

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

s^{T} A

s^{T} C_{all}

s^{T} p

masks b

s^{T} $B - x^T \otimes G$

commit to $p + t_j$ at position $j \in [N]$

$s^{\mathrm{T}} t_i \approx s^{\mathrm{T}} A y_i + s^{\mathrm{T}} (B - x^T \otimes G) h$
if and only if $f_i(x) = 0$

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Recall:

$B - x^T \otimes G$ h $= b_{f,d} - f(x) \cdot d$

$h = h_{B,f,x,d}$ is low-norm

Registered ABE For Unbounded Users [WW25]

Public Parameters

pp_{com} p d_i

B $i \in [N]$

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

$s^{\mathrm{T}} A$ $s^{\mathrm{T}} C_{all}$ $s^{\mathrm{T}} p$

$s^{\mathrm{T}} (B - x^T \otimes G)$

masks b

Noise omitted

$s^{\mathrm{T}} t_i \approx s^{\mathrm{T}} A y_i + s^{\mathrm{T}} (B - x^T \otimes G) h$
if and only if $f_i(x) = 0$

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

For all $i \in [N]$: $C_{all} v_i = p + t_i - A z_i$

$$s^{\mathrm{T}} C_{all} v_i \approx s^{\mathrm{T}} p + s^{\mathrm{T}} t_i - s^{\mathrm{T}} A z_i$$

$$\underbrace{s^{\mathrm{T}} C_{all} v_i + s^{\mathrm{T}} A z_i}_{\text{public}} \approx \underbrace{s^{\mathrm{T}} p + s^{\mathrm{T}} t_i}_{\text{dual-Regev ciphertext under } \mathrm{pk}_i}$$

Registered ABE (Selective) Security

[HLWW23]

Adversary

x

pp

Challenger

Registered ABE (Selective) Security

[HLWW23]

Adversary

Challenger

x

pp

i, f

pk_i

repeats throughout game

Registered ABE (Selective) Security

[HLWW23]

Adversary

Challenger

x

pp

i, f

pk_i

repeats throughout game

$\{\text{pk}_i, f_i\}_{i \in [N]}$

if key is adversarial, f_i should not decrypt

Registered ABE (Selective) Security

[HLWW23]

Adversary

Challenger

x

pp

i, f

pk_i

repeats throughout game

$\{\mathrm{pk}_i, f_i\}_{i \in [N]}$

challenge bit $b \in \{0,1\}$ sampled randomly

$\mathrm{Encrypt}(\mathrm{mpk}, x, b)$

computed via Aggregate

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i

B $i \in [N]$

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Noise omitted

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

s^T A

s^T C_{all}

s^T p

s^T $B - x^T \otimes G$

commit to $p + t_j$ at position $j \in [N]$

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i

B $i \in [N]$

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

given by matrix commitment security

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$s^T A$ $s^T C_{all}$ $s^T p$

$s^T (B - x^T \otimes G)$

commit to $p + t_j$ at position $j \in [N]$

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i

B $i \in [N]$

Public Key

$t_i = A\,y_i + b_{f,d_i}$

low-norm **secret key**

given by matrix commitment security

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

s^T A

s^T $B - x^T \otimes G$

s^T C_{all}

Noise omitted

s^T p

can set $p = Ak$ for low-norm k and $B = AK + x^T \otimes G$ for low-norm K

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i

B $i \in [N]$

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

s^T A

s^T C_{all}

s^T p

s^T $B - x^T \otimes G$

commits vectors t_j chosen by the adversary!

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i $i \in [N]$

B

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$s^T A$ $s^T \left(C_{all} + C_0 \right)$ $s^T p$

$s^T (B - x^T \otimes G)$

Key idea: add a randomized commitment to $\mathbf{0}$ at encryption time to remove adversarial control

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i $i \in [N]$

B

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$s^T A$

$s^T \left(C_{all} + C_0 \right)$

$s^T p$

$s^T (B - x^T \otimes G)$

Key idea: add a randomized commitment to $\mathbf{0}$ at encryption time to remove adversarial control

Problem: must provide openings to ensure correctness!

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i $i \in [N]$

B

Public Key

$t_i = A\, y_i + b_{f,d_i}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$s^T A$ $s^T \left(C_{all} + C_0 \right)$ $s^T p$

$s^T (B - x^T \otimes G)$

Key idea: add a randomized commitment to $\mathbf{0}$ at encryption time to remove adversarial control

Problem: must provide openings to ensure correctness!

Solution: random oracle derives C_0 and $z_{0,i}$ from aggregation query

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i $i \in [N]$

B

Public Key

$t_i = A y_i + b_{f,d_i}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$s^T A$ $s^T (C_{all} + C_0)$ $s^T p$

$s^T (B - x^T \otimes G)$

RO derives randomized commitment C_0 and openings $z_{0,i}$ from aggregation query

Additional proof notes:

- Randomness of C_0 is public so commitment must be *explainable* [AWY20,LW22] to program $C_0 = AK_0 - C_{all}$ indistinguishably

Proving Security [CHW25]

Public Parameters

pp_{com} p d_i $i \in [N]$

B

Public Key

$t_i = A\,y_i + b_{f,d_i}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$s^T A$ $s^T (C_{all} + C_0)$ $s^T p$

$s^T (B - x^T \otimes G)$

RO derives randomized commitment C_0 and openings $z_{0,i}$ from aggregation query

Additional proof notes:

- Randomness of C_0 is public so commitment must be *explainable* [AWY20,LW22] to program $C_0 = AK_0 - C_{all}$ indistinguishably
- Openings $z_{0,i}$ must be sufficiently high-norm; idea is to find a low-norm "opening" $z'_{0,i}$ such that $C_0 v_i = d_i - A z'_{0,i}$, program $d_i = A k_i$ where $\|k_i\| = \|z_{0,i}\|$, and set $z_{0,i} = z'_{0,i} - k_i$ as the opening to $\mathbf{0}$

Proving Security [CHW25]

Public Parameters

pp_{com} $\boldsymbol{p}$ $\boldsymbol{d_i}$ $i \in [N]$

$\boldsymbol{B}$

Public Key

$\boldsymbol{t_i} = \boldsymbol{A}\ \boldsymbol{y_i} + \boldsymbol{b_{f,d_i}}$

low-norm **secret key**

Encryption of $b \in \{0,1\}$ to $x \in \{0,1\}^{\ell}$:

Noise omitted

$\boldsymbol{s}^{\mathrm{T}}\ \boldsymbol{A}$ $\quad$ $\boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{C_{all}} + \boldsymbol{C_0}\right)$ $\quad$ $\boldsymbol{s}^{\mathrm{T}}\ \boldsymbol{p}$

$\boldsymbol{s}^{\mathrm{T}}\ \boldsymbol{B} - \boldsymbol{x}^{T} \otimes \boldsymbol{G}$

RO derives randomized commitment $\boldsymbol{C_0}$ and openings $\boldsymbol{z_{0,i}}$ from aggregation query

Additional proof notes:

- Randomness of $\boldsymbol{C_0}$ is public so commitment must be *explainable* [AWY20,LW22] to program $\boldsymbol{C_0} = \boldsymbol{AK_0} - \boldsymbol{C_{all}}$ indistinguishably
- Openings $\boldsymbol{z_{0,i}}$ must be sufficiently high-norm; idea is to find a low-norm "opening" $\boldsymbol{z'_{0,i}}$ such that $\boldsymbol{C_0 v_i} = \boldsymbol{d_i} - \boldsymbol{A z'_{0,i}}$, program $\boldsymbol{d_i} = \boldsymbol{A k_i}$ where $\|\boldsymbol{k_i}\| = \|\boldsymbol{z_{0,i}}\|$, and set $\boldsymbol{z_{0,i}} = \boldsymbol{z'_{0,i}} - \boldsymbol{k_i}$ as the opening to $\mathbf{0}$

Techniques also work for adaptively-secure DBE in the ROM

Open Problems

- Proving security from plain LWE
- Removing the random oracle from registered ABE or adaptively-secure DBE
- Cryptanalysis and more applications of succinct/decomposed LWE

Thanks for listening!

https://eprint.iacr.org/2025/044

https://eprint.iacr.org/2025/1039