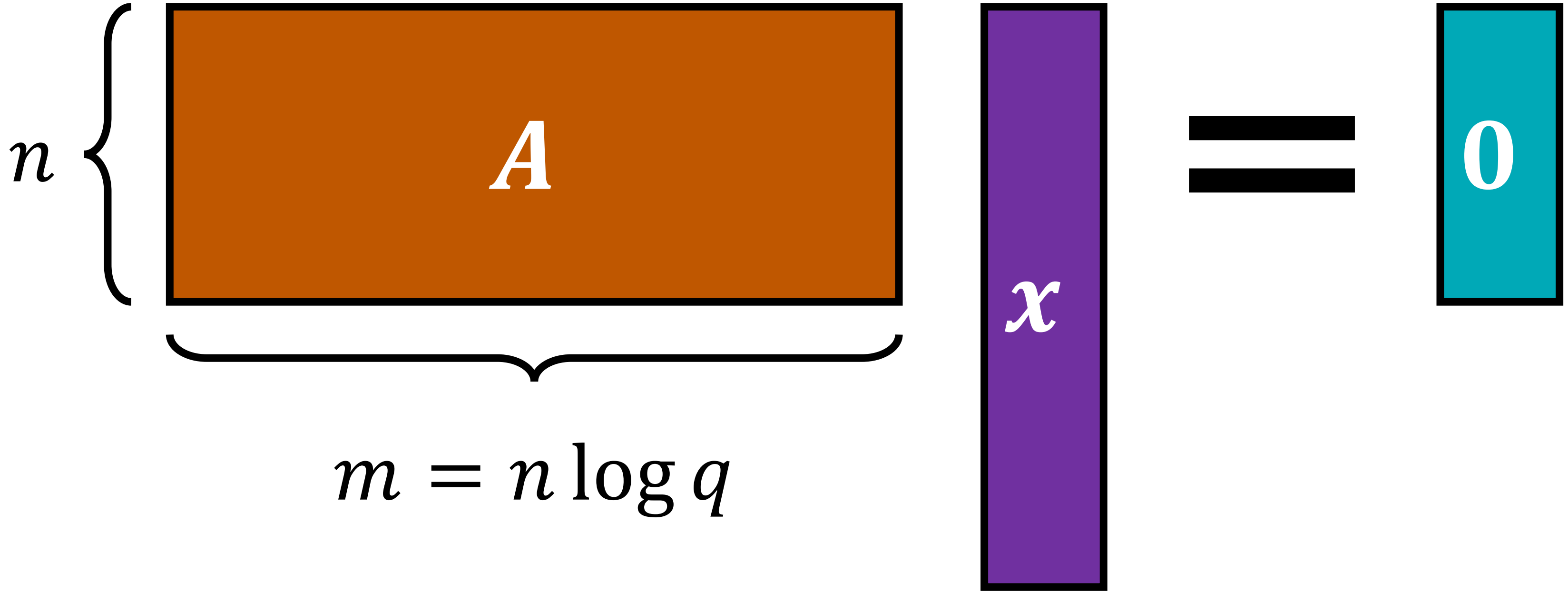
Exotic Lattice Assumptions and How to Tame Them

David Wu

[Images are AI-generated]

Lattice Problems in Cryptography

Short integer solutions (SIS): Given $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$, find $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ [Ajt96]

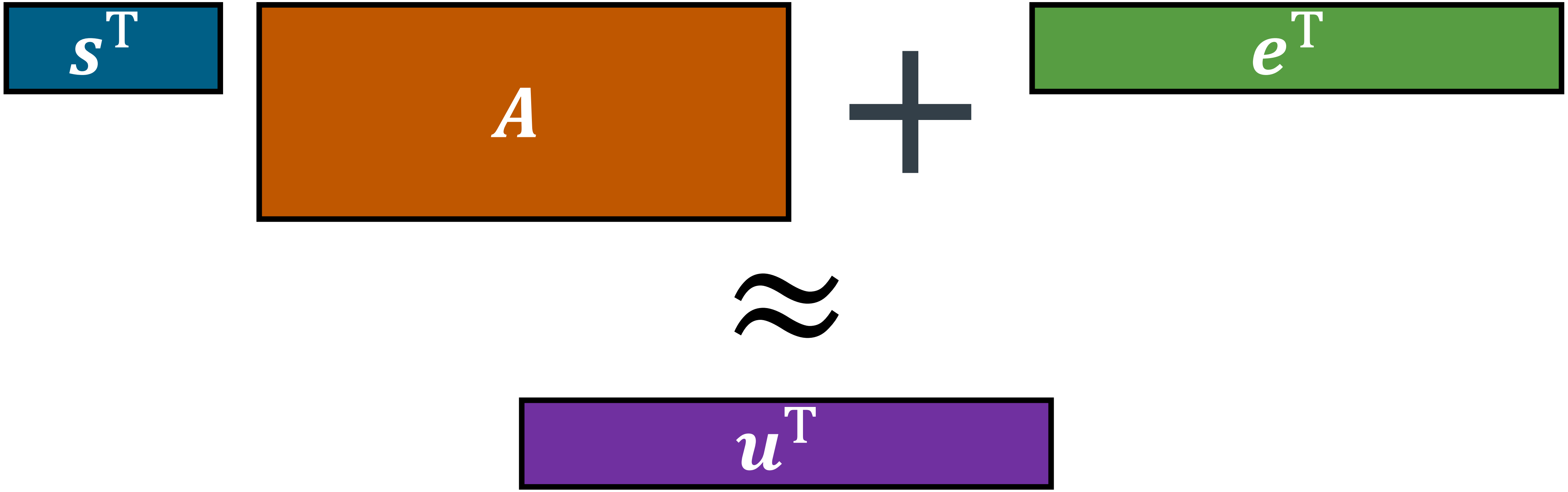


Yields one-way functions, collision-resistant hash functions, digital signatures

Lattice Problems in Cryptography

Short integer solutions (SIS): Given $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}$, find $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ [Ajt96]

Learning with errors (LWE): Distinguish $\left(\boldsymbol{A}, \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}\right)$ from $\left(\boldsymbol{A}, \boldsymbol{u}^{\mathrm{T}}\right)$ [Reg05]



Lattice Problems in Cryptography

Short integer solutions (SIS): Given $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$, find $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ [Ajt96]

Learning with errors (LWE): Distinguish $(\boldsymbol{A}, \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}})$ from $(\boldsymbol{A}, \boldsymbol{u}^{\mathrm{T}})$ [Reg05]

PKE	IBE	FHE	ABE	homomorphic signatures	lockable obfuscation	traitor tracing	NIZK	SNARGs for P
[Reg05]	[GPV08]	[Gen09, BV11]	[GVW13, BGG+14]	[GVW15]	[WZ17, GVW17]	[GKW18]	[PS19]	[CJJ21]

But… *not* everything

Broadcast encryption [BV22]

Witness encryption [GGH15, CVW18]

Indistinguishability obfuscation

[GGH15, Agr19, CHVW19, AP20, BDGM20a, WW21, GP21, BDGM20b, DQVWW21]

However, many **lattice-inspired** approaches

Lattice Problems in Cryptography

Short integer solutions (SIS): Given $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$, find $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \mathbf{0}$ [Ajt96]

Learning with errors (LWE): Distinguish $\left(\boldsymbol{A}, \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}\right)$ from $\left(\boldsymbol{A}, \boldsymbol{u}^{\mathrm{T}}\right)$ [Reg05]

PKE	IBE	FHE	ABE	homomorphic signatures	lockable obfuscation	traitor tracing	NIZK	SNARGs for P
[Reg05]	[GPV08]	[Gen09, BV11]	[GVW13, BGG+14]	[GVW15]	[WZ17, GVW17]	[GKW18]	[PS19]	[CJJ21]

But… *not* everything

Broadcast encryption [BV22]

Witness encryption [GGH15, CVW18]

Indistinguishability obfuscation

[GGH15, Agr19, CHVW19, AP20, BDGM20a, WW21, GP21, BDGM20b, DQVWW21]

However, many **lattice-inspired** approaches

Most schemes did not have a concrete hardness assumption or were based on a hardness assumption that was subsequently broken (in the most general setting)

Lattice Problems in Cryptography

This talk: new lattice assumptions that enable these advanced applications and moves the field of lattice-based cryptography forward

Hope: over time, will be able to reduce to the standard lattice problems

Very successful in the area of bilinear maps: many new assumptions (e.g., composite-order, q-type, etc.), but can now do most things from k-Lin

But… *not* everything

- Broadcast encryption [BV22]
- Witness encryption [GGH15, CVW18]
- Indistinguishability obfuscation [GGH15, Agr19, CHVW19, AP20, BDGM20a, WW21, GP21, BDGM20b, DQVWW21]

However, many **lattice-inspired** approaches

Most schemes did not have a concrete hardness assumption or were based on a hardness assumption that was subsequently broken (in the most general setting)

Evasive LWE

Evasive LWE [Wee22, Tsa22]:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \text{aux}) \leftarrow \text{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \text{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \text{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \text{aux}$

$\boldsymbol{A}^{-1}(\boldsymbol{P})$ is a short (Gaussian) preimage of $\boldsymbol{P}$: namely $\boldsymbol{A} \cdot \boldsymbol{A}^{-1}(\boldsymbol{P}) = \boldsymbol{P}$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}$$

(will suppress noise terms for simplicity)

Can also restrict the class of samplers
(will discuss more later)

Evasive LWE

Evasive LWE [Wee22, Tsa22]:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^\mathrm{T}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^\mathrm{T}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Adversary in the post-condition can always compute

$$\boldsymbol{s}^\mathrm{T}\boldsymbol{A} \cdot \boldsymbol{A}^{-1}(\boldsymbol{P}) \approx \boldsymbol{s}^\mathrm{T}\boldsymbol{P}$$

This must look indistinguishable from $\boldsymbol{u}^\mathrm{T} \cdot \boldsymbol{A}^{-1}(\boldsymbol{P}) \equiv \mathrm{uniform}$ (pre-condition)

Heuristic is that $\boldsymbol{s}^\mathrm{T}\boldsymbol{A}$ and $\boldsymbol{A}^{-1}(\boldsymbol{P})$ only leaks $\boldsymbol{s}^\mathrm{T}\boldsymbol{P}$ and nothing more

Pre-condition captures "zeroizing" attacks on earlier lattice-based schemes (e.g., auxiliary input reveals a short vector $\boldsymbol{v}$ where $\boldsymbol{P}\boldsymbol{v} = \boldsymbol{0}$)

Evasive LWE

Evasive LWE [Wee22, Tsa22]:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^\mathrm{T}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^\mathrm{T}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Example 1:

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

Pre-condition follows by LWE

Post-condition also follows by LWE

Sample Gaussian $\boldsymbol{R} \in \mathbb{Z}_q^{m \times \ell}$ and set $\boldsymbol{P} = \boldsymbol{AR}$ (statistically close to uniform)

Evasive LWE

Evasive LWE [Wee22, Tsa22]:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$			
	if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
	then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Example 2:

Suppose $\boldsymbol{P} = [\boldsymbol{U} \mid \boldsymbol{U}]$ where $\boldsymbol{U} \in \mathbb{Z}_q^{n\times m}$

Pre-condition is false

Evasive LWE provides no guarantees (post-condition is also false for sufficiently-wide $\boldsymbol{U}$; $\boldsymbol{A}^{-1}([\,\boldsymbol{U} \mid \boldsymbol{U}\,])$ yields a trapdoor for $\boldsymbol{A}$)

Evasive LWE

Evasive LWE [Wee22, Tsa22]:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Public-coin evasive LWE: aux is the random coins to Samp

Private-coin evasive LWE: secret randomness used in Samp

Many different variants (e.g., whether $\boldsymbol{A}, \boldsymbol{P}$ are available to the distinguisher)

- See [BÜW24] for a systematic treatment

Applications of Evasive LWE

Public-coin evasive LWE

- (Optimal) broadcast encryption [Wee22]
- Multi-authority ABE [WWW22, CLW24]
- ABE for unbounded-depth circuits [HLL23]
- ABE for DFA and log-space Turing machines [HLL24]

Different schemes have somewhat different formulations of the assumption, but similar principles

Private-coin evasive LWE

- Witness encryption [Tsa22, VWW22]
- Multi-input ABE [ARYY23]
- Witness PRFs (and designated-verifier SNARGs) for UP [MPV24]
- ABE for Turing machines [AKY24]
- Universal computational extractors [CM24]
- Pseudorandom obfuscation, succinct witness encryption [BDJMMPV24]
- Registered ABE for circuits [ZZCGQ25]

Cryptanalysis of Evasive LWE

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda)$, $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Public-coin evasive LWE

No counter-examples to date (for the standard version where $\boldsymbol{A}, \boldsymbol{P}$ are public)

Private-coin evasive LWE

Obfuscation-based counter-example [Wee22, VWW23, BÜW24]

aux contains an obfuscated program with a trapdoor for $\boldsymbol{P}$ that is used to distinguish $\left(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}, \boldsymbol{A}^{-1}(\boldsymbol{P})\right)$ from $\left(\mathrm{random}, \boldsymbol{A}^{-1}(\boldsymbol{P})\right)$

Cryptanalysis of Evasive LWE

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Public-coin evasive LWE

No counter-examples to date (for the standard version where $\boldsymbol{A}, \boldsymbol{P}$ are public)

Private-coin evasive LWE

Obfuscation-based counter-example [Wee22, VWW23, BÜW24]:

Explicit counter-examples to several families of evasive LWE [BÜW24]

- Gives distributions where pre-condition holds under LWE, but post-condition is false (no auxiliary input!)

Cryptanalysis of Evasive LWE

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \text{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx \text{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Public-coin evasive LWE

No counter-examples to date (for the standa

Private-coin evasive LWE

Obfuscation-based counter-example [Wee22,

Explicit counter-examples to several families

Gives distributions where pre-condition holds und

Suppose $\boldsymbol{P}$ is not given out in pre-condition

Let $\boldsymbol{P} = [\,\boldsymbol{P}_1 \mid \boldsymbol{P}_2\,]$ where $\boldsymbol{P}_2 = \begin{bmatrix} \boldsymbol{u}^{\mathrm{T}} \\ \boldsymbol{R} \end{bmatrix}$ where $\boldsymbol{P}_1\boldsymbol{u} = \boldsymbol{0}$, $\boldsymbol{u}$ is short, and $\boldsymbol{P}_1, \boldsymbol{R}$ uniform

Pre-condition holds under LWE (when $\boldsymbol{P}$ is hidden)

Post-condition is false:

- Recode $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$ to $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{P}_1$
- Use $\boldsymbol{A}, \boldsymbol{A}^{-1}(\boldsymbol{P})$ to obtain $\boldsymbol{u}$
- Check if $\boldsymbol{s}^T\boldsymbol{P}_1\boldsymbol{u} \approx 0$

[BÜW24] counter-example

Cryptanalysis of Evasive LWE

For all efficient samplers Samp and taking $(\boldsymbol{P}, \text{aux}) \leftarrow \text{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \text{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \text{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \text{aux}$

Public-coin evasive LWE

No counter-examples to date (for the standard version where $\boldsymbol{A}, \boldsymbol{P}$ are public)

Private-coin evasive LWE

Obfuscation-based counter-example [Wee22, VWW23, BÜW24]:

Explicit counter-examples to several families of evasive LWE [BÜW24]

- Gives distributions where pre-condition holds under LWE, but post-condition is false (no auxiliary input!)
- Counter-examples apply to original formulation of evasive LWE families from [Tsa22, VWW22, ARYY23], but assumptions can be patched (and security proofs recovered)

Cryptanalysis of Evasive LWE

For all efficient samplers Samp and taking $(\boldsymbol{P}, \text{aux}) \leftarrow \text{Samp}(1^\lambda)$, $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \text{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \text{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \text{aux}$

Public-coin evasive LWE

No counter-examples to date (for the standard version where $\boldsymbol{A}, \boldsymbol{P}$ are public)

Private-coin evasive LWE

Obfuscation-based counter-example [Wee22, VWW23, BÜW24]:

Explicit counter-examples to several families of evasive LWE [BÜW24]

Implies pseudorandom obfuscation for all PRFs (impossible object) [BDJMMPV24]

Useful heuristic, but tread carefully!

Beyond Evasive LWE

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda)$, $\boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Evasive LWE assumption is non-falsifiable (challenging for cryptanalysis)

Specific assumption (i.e., distribution of samplers) is scheme-dependent (i.e., instance-dependent)

Overreliance on post-condition leads to "*super-selective*" security for constructions

Better: identify a single easy-to-state, falsifiable assumption that suffices for applications [today]

Even better: get these applications from plain LWE [not today…]

Beyond Evasive LWE

Common approach:

LWE (or SIS) is hard given some hint
(e.g., trapdoor for a related matrix, short preimages of specific targets)

Examples:

- k-R-ISIS [ACLMT22]
- twin k-R-ISIS [BCFL23]
- BASIS (basis augmented SIS) [WW23]
- PRISIS [FN23]

Constructions of functional commitments and succinct non-interactive arguments (SNARGs) for NP

"SIS with Hints Zoo" (maintained by Martin Albrecht): https://malb.io/sis-with-hints.html

This talk: ℓ-succinct LWE [Wee24]; terms in the assumption have the "least" structure

Implies succinct ABE [Wee24], functional commitments [WW23], distributed broadcast encryption [CW24], registered ABE [CHW25]

Succinct LWE

ℓ-Succinct LWE [Wee24]:

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

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

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

$$\boxed{\boldsymbol{G} = \boldsymbol{I}_n \otimes \left[1,2,\ldots,2^{\lceil \log q\rceil - 1}\right]}$$

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

Falsifiable!

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

Succinct LWE

ℓ-Succinct LWE [Wee24]:

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

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

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

$$\boxed{\boldsymbol{G} = \boldsymbol{I}_n \otimes [1, 2, \dots, 2^{\lceil \log q \rceil - 1}]}$$

Another view of the trapdoor:

LWE is hard with respect to $\boldsymbol{A}$ given many samples of the form

$$\left(\boldsymbol{r}_j \leftarrow \chi^m, \boldsymbol{A}^{-1}(\boldsymbol{W}_1 \boldsymbol{r}_j), \dots, \boldsymbol{A}^{-1}(\boldsymbol{W}_\ell \boldsymbol{r}_j)\right)$$

Succinct LWE

ℓ-Succinct LWE [Wee24]:

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

$$\boldsymbol{D}_\ell = \left[\begin{array}{ccc|c} A & & & W_1 \\ & \ddots & & \vdots \\ & & A & W_\ell \end{array}\right]$$

Two axis for hardness:

LWE (when $\mathrm{width}(\boldsymbol{W}) \geq O(n \log q)$)

Open!

problem should get easier

$\ell = 1$

Baby step: $\mathrm{poly}(\ell)$ speed-up over solving LWE

number of blocks ℓ

Succinct LWE

ℓ-Succinct LWE [Wee24]:

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

$$\boldsymbol{D}_\ell = \left[\begin{array}{ccc|c} A & & & W_1 \\ & \ddots & & \vdots \\ & & A & W_\ell \end{array}\right]$$

Two axis for hardness:

Succinct LWE

ℓ-Succinct LWE [Wee24]:

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

Special cases where it is implied by LWE:

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

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

Two types of applications (so far) using the trapdoor:

- **Compression:** functional commitments [WW23], succinct ABE [Wee24]
- **Distributed key-generation:** distributed broadcast encryption [CW24], registered ABE [CHW25]

Homomorphic Computation using Lattices

[GSW13, BGGHNSVV14]

Encodes a vector $\boldsymbol{x} \in \{0,1\}^\ell$ with respect to matrix $\boldsymbol{B} = [\boldsymbol{B}_1 \mid \cdots \mid \boldsymbol{B}_\ell] \in \mathbb{Z}_q^{n \times \ell m}$

$\boldsymbol{B}_1 - x_1 \boldsymbol{G}$	$\boldsymbol{B}_2 - x_2 \boldsymbol{G}$	$\cdots$	$\boldsymbol{B}_\ell - x_\ell \boldsymbol{G}$

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

Given any function $f: \{0,1\}^\ell \to \{0,1\}$, there exists a **short** matrix $\boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}}$ where

$$\underbrace{\left(\boldsymbol{B} - \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right)}_{\text{encoding of } \boldsymbol{x} \text{ with respect to } \boldsymbol{B}} \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} = \underbrace{\boldsymbol{B}_f - f(x) \cdot \boldsymbol{G}}_{\text{encoding of } f(\boldsymbol{x}) \text{ with respect to } \boldsymbol{B}_f}$$

Given $\boldsymbol{B}$ and f, can efficiently compute the matrix $\boldsymbol{B}_f$

Homomorphic Commitments

[GVW15]

Goal: commit to $\boldsymbol{x} \in \{0,1\}^\ell$ and open to $y = f(\boldsymbol{x}) \in \{0,1\}$

Evaluation binding: cannot open a commitment to both 0 and 1 with respect to the same function f

---

public parameters: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$

commitment: $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} + \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}$ where $\boldsymbol{R} \leftarrow \{0,1\}^{m \times \ell m}$

opening to function f: $\boldsymbol{R}_f = \boldsymbol{R} \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} \in \mathbb{Z}_q^{n \times m}$

verification: check $\boldsymbol{R}_f$ is short and $\boldsymbol{A}\boldsymbol{R}_f = \boldsymbol{B}_f - y \cdot \boldsymbol{G} \in \mathbb{Z}_q^{n \times m}$

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

Homomorphic Commitments

[GVW15]

Correctness:

$$\boldsymbol{A}\boldsymbol{R}_f = \boldsymbol{A}\boldsymbol{R} \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} = \left(\boldsymbol{B} - \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right) \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} = \boldsymbol{B}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G}$$

Security: Openings to 0 and 1 reveals trapdoor for $\boldsymbol{A}$

---

public parameters: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$

commitment: $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} + \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}$ where $\boldsymbol{R} \leftarrow \{0,1\}^{m \times \ell m}$

opening to function f: $\boldsymbol{R}_f = \boldsymbol{R} \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} \in \mathbb{Z}_q^{n \times m}$

verification: check $\boldsymbol{R}_f$ is short and $\boldsymbol{A}\boldsymbol{R}_f = \boldsymbol{B}_f - y \cdot \boldsymbol{G} \in \mathbb{Z}_q^{n \times m}$

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

Compressing using Succinct LWE

[WW23, Wee24]

Succinct LWE trapdoor can be used to compress $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{R} + \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}$

$$[\boldsymbol{I} \otimes \boldsymbol{A} \mid \boldsymbol{W}] \cdot \boldsymbol{T} = \begin{bmatrix} \boldsymbol{A} & & & \boldsymbol{W}_1 \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & \boldsymbol{W}_\ell \end{bmatrix} \begin{bmatrix} \boldsymbol{T}_1 \\ \vdots \\ \boldsymbol{T}_\ell \\ \underline{\boldsymbol{T}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} & & \\ & \ddots & \\ & & \boldsymbol{G} \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G} = (\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{I})(\boldsymbol{I} \otimes \boldsymbol{G}) &= (\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{I})[\boldsymbol{I} \otimes \boldsymbol{A} \mid \boldsymbol{W}] \cdot \boldsymbol{T} \\ &= (\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{I})[\boldsymbol{I} \otimes \boldsymbol{A} \mid \boldsymbol{W}] \cdot \begin{bmatrix} \overline{\boldsymbol{T}} \\ \underline{\boldsymbol{T}} \end{bmatrix} \\ &= \boldsymbol{A}(\boldsymbol{I} \otimes \boldsymbol{x}^{\mathrm{T}})\overline{\boldsymbol{T}} + (\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{I})\boldsymbol{W}\underline{\boldsymbol{T}} \end{aligned}$$

Compressing using Succinct LWE

[WW23, Wee24]

Same technique applies to [BGGHNSVV14] ABE scheme: gives ABE with succinct ciphertexts (and broadcast encryption)

$$\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G} = \boldsymbol{A}(\boldsymbol{I} \otimes \boldsymbol{x}^{\mathrm{T}})\overline{\boldsymbol{T}} + (\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{I})\boldsymbol{W}\underline{\boldsymbol{T}}$$

$$= \boldsymbol{A} \underbrace{\sum_{i \in [\ell]} x_i \boldsymbol{T}_{\mathrm{i}}}_{\boldsymbol{R} \in \mathbb{Z}_q^{m \times \ell m}} + \underbrace{\sum_{i \in [\ell]} x_i \boldsymbol{W}_i}_{\boldsymbol{C} \in \mathbb{Z}_q^{n \times m}} \cdot \underline{\boldsymbol{T}}$$

$\boldsymbol{C}$ is a ***succinct*** commitment to $\boldsymbol{x}$

$$\boxed{\textbf{Observe: } \boldsymbol{C}\underline{\boldsymbol{T}} = -\boldsymbol{A}\boldsymbol{R} + \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}}$$

Distributed Key Generation

Including a trapdoor in the public parameters also useful for distributed setup

Instead of giving out trapdoor for $\boldsymbol{A}$ (insecure), give out a trapdoor for a matrix related to $\boldsymbol{A}$ (which suffices for correctness)

Enables applications to constructing *trustless* cryptographic primitives (e.g., distributed broadcast encryption and registered ABE)

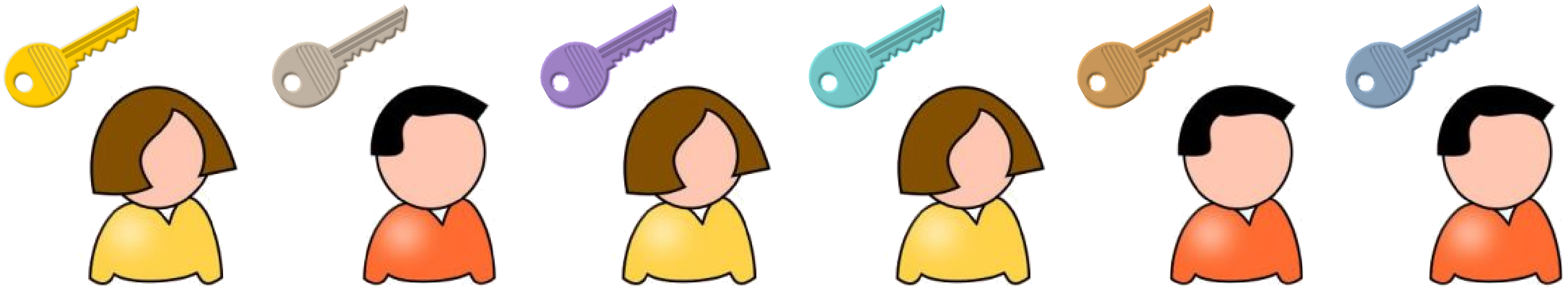
Broadcast Encryption

[FN93]

message m

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

Ciphertext specifies a set of users

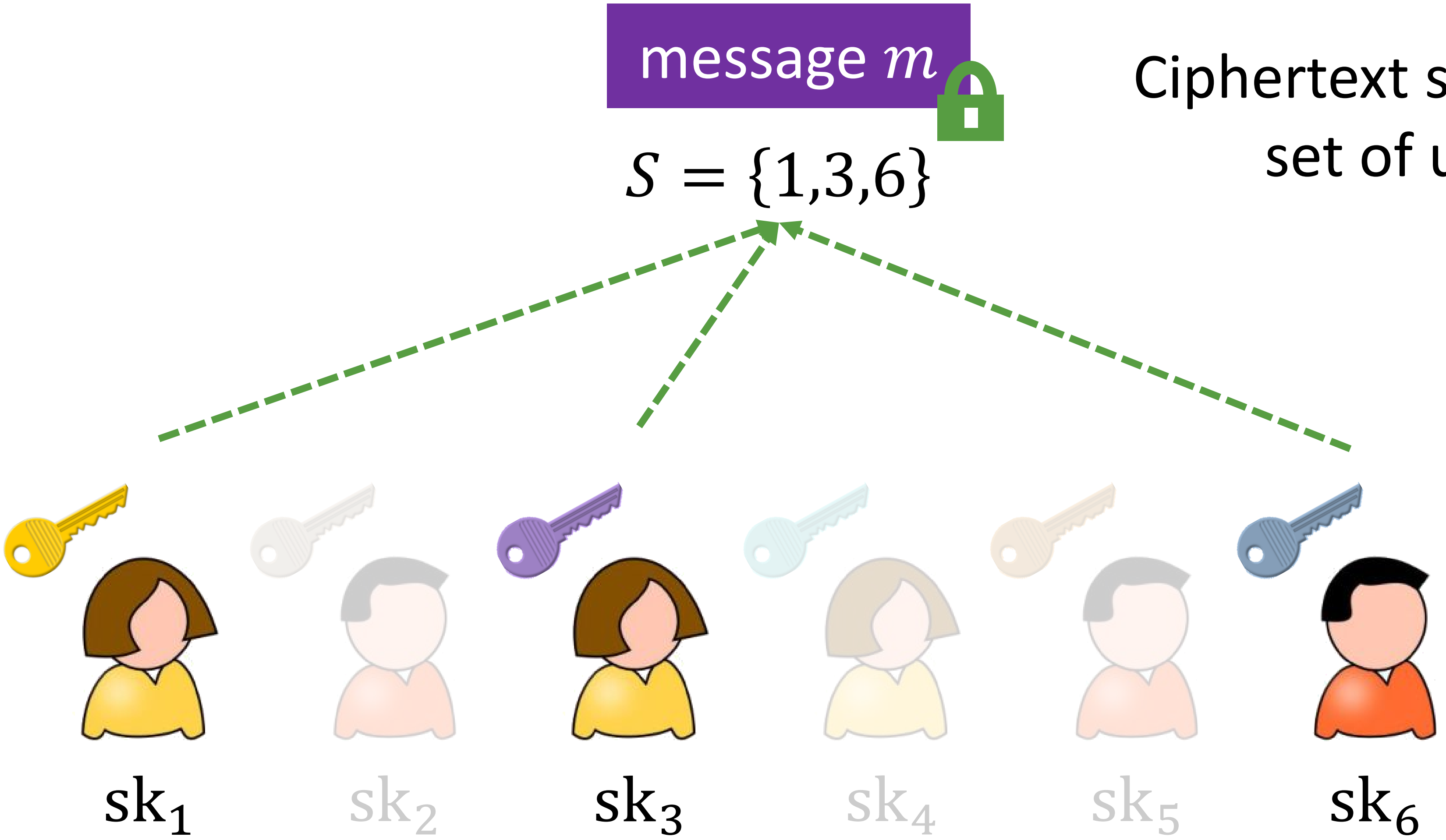


sk_1 sk_2 sk_3 sk_4 sk_5 sk_6

Broadcast Encryption

[FN93]

Functionality: Users in the set can decrypt



Ciphertext specifies a set of users

Broadcast Encryption

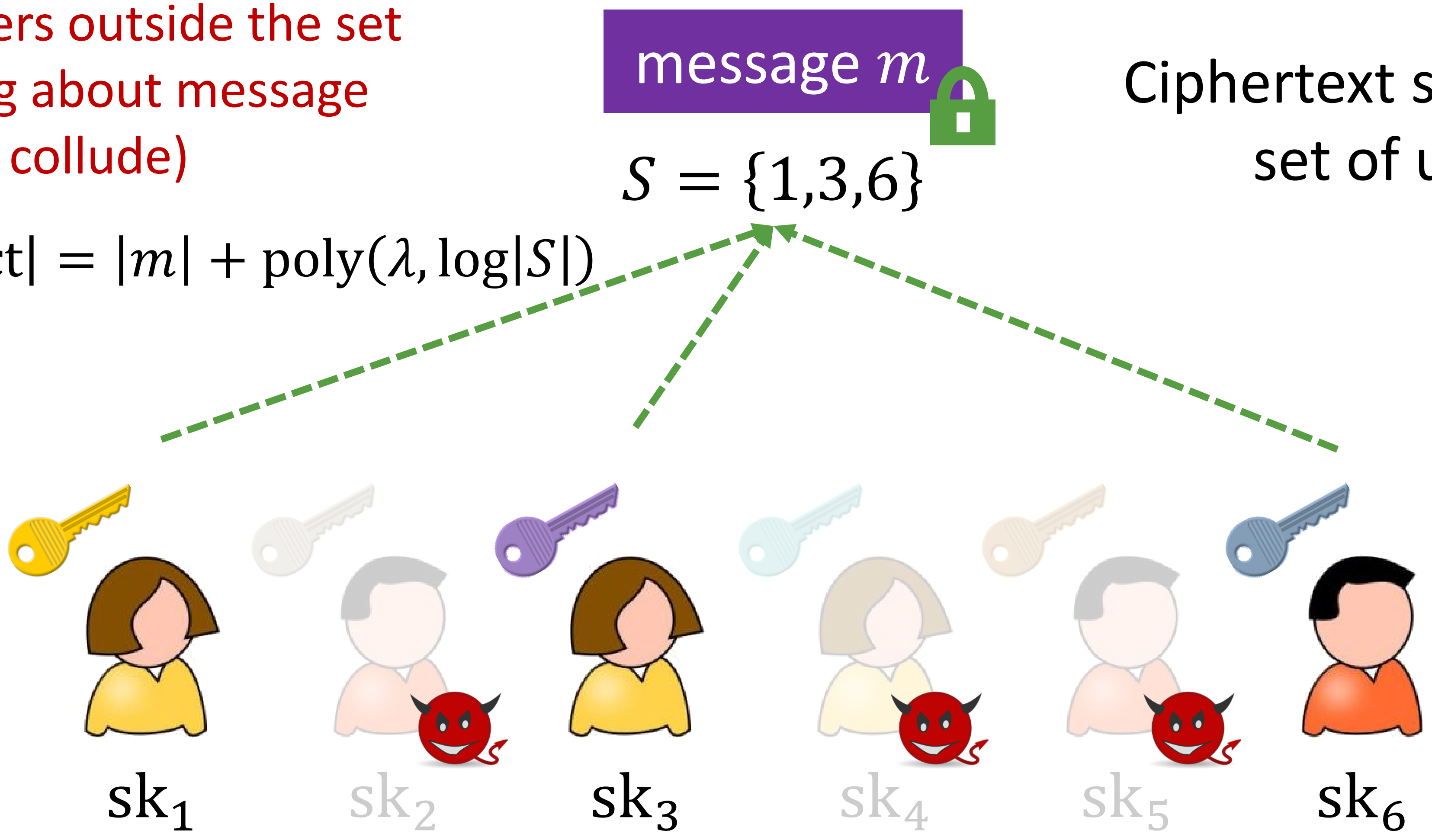
[FN93]

Functionality: Users in the set can decrypt

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

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

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)

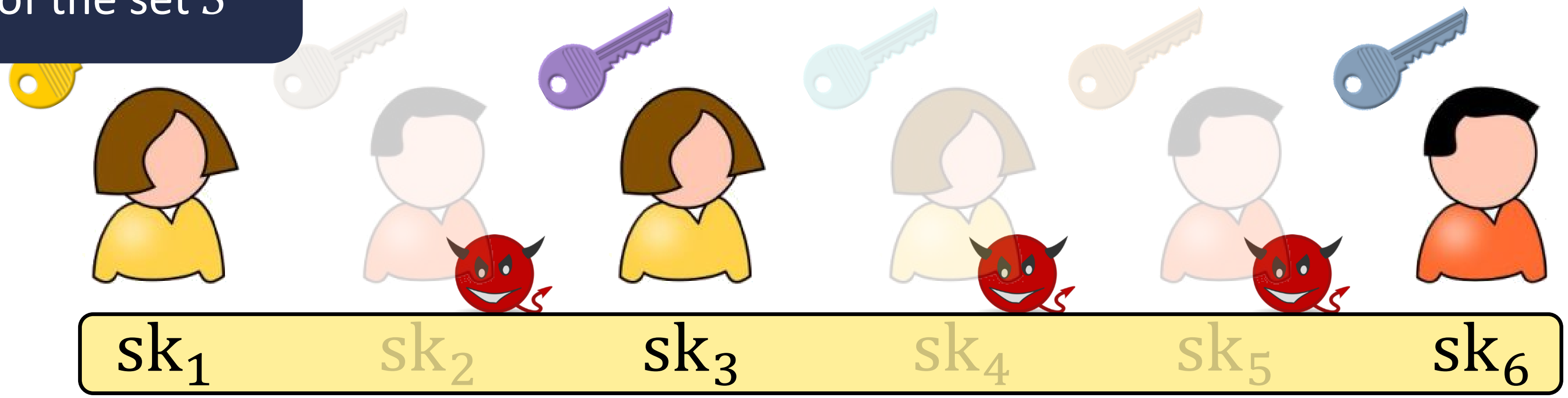
message m

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

Ciphertext specifies a set of users

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

Note: decryption requires knowledge of the set S



Where do the secret keys come from?

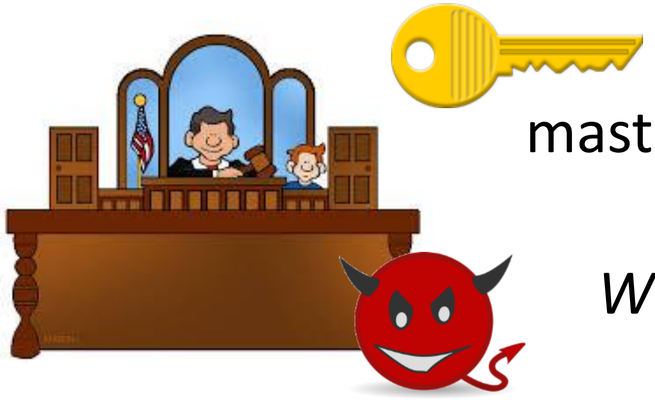
Broadcast Encryption

[FN93]

Central **trusted** authority generates keys

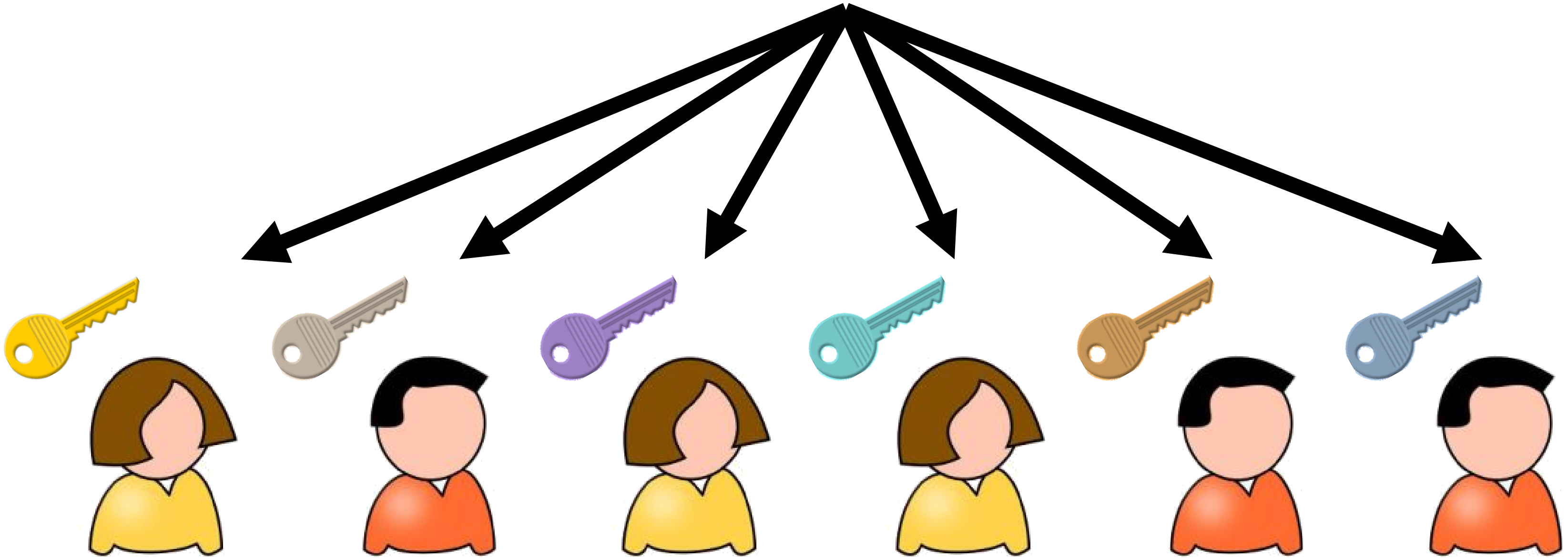
Built-in **key escrow**

Central point of failure



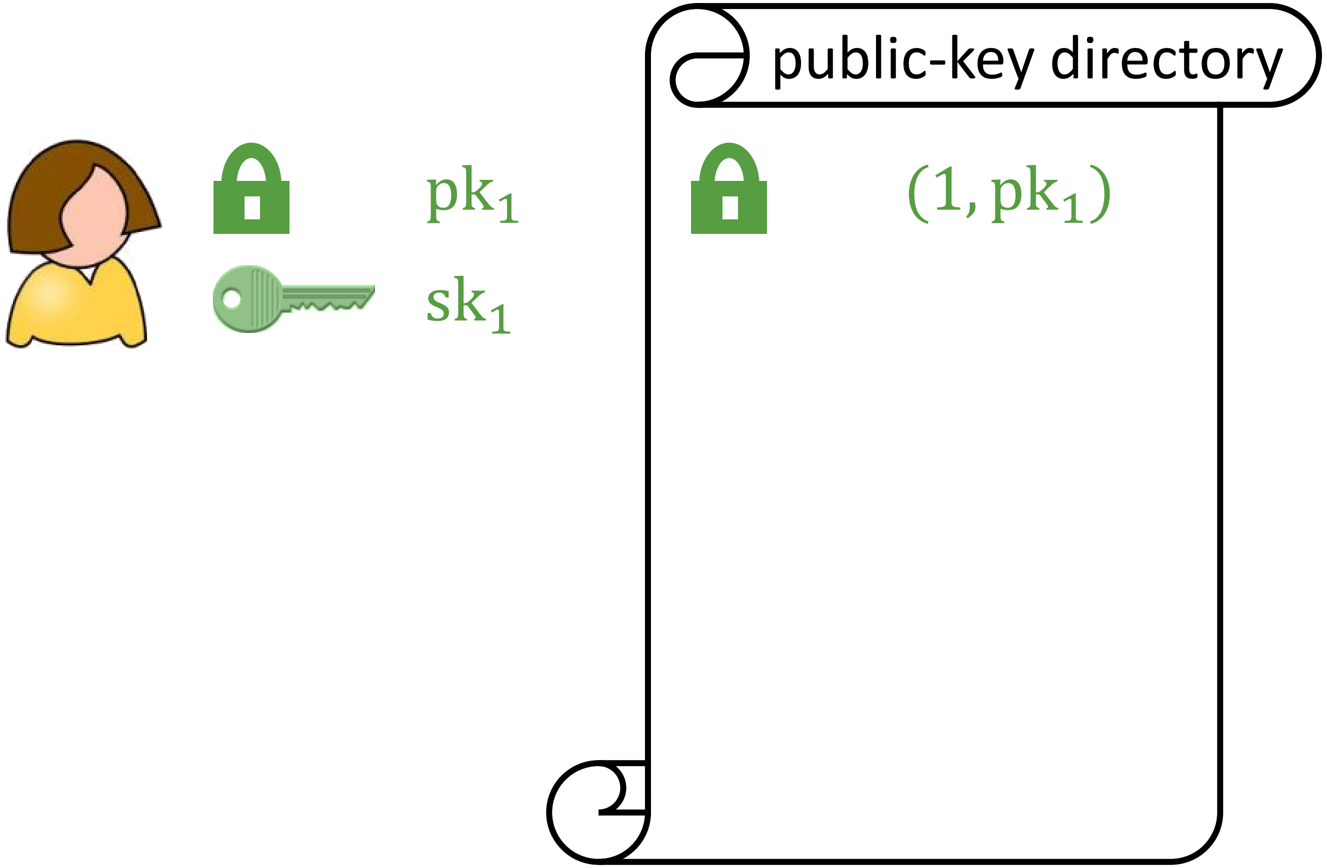
master secret key

What if the key issuer is compromised?



Distributed Broadcast Encryption

[BZ14]

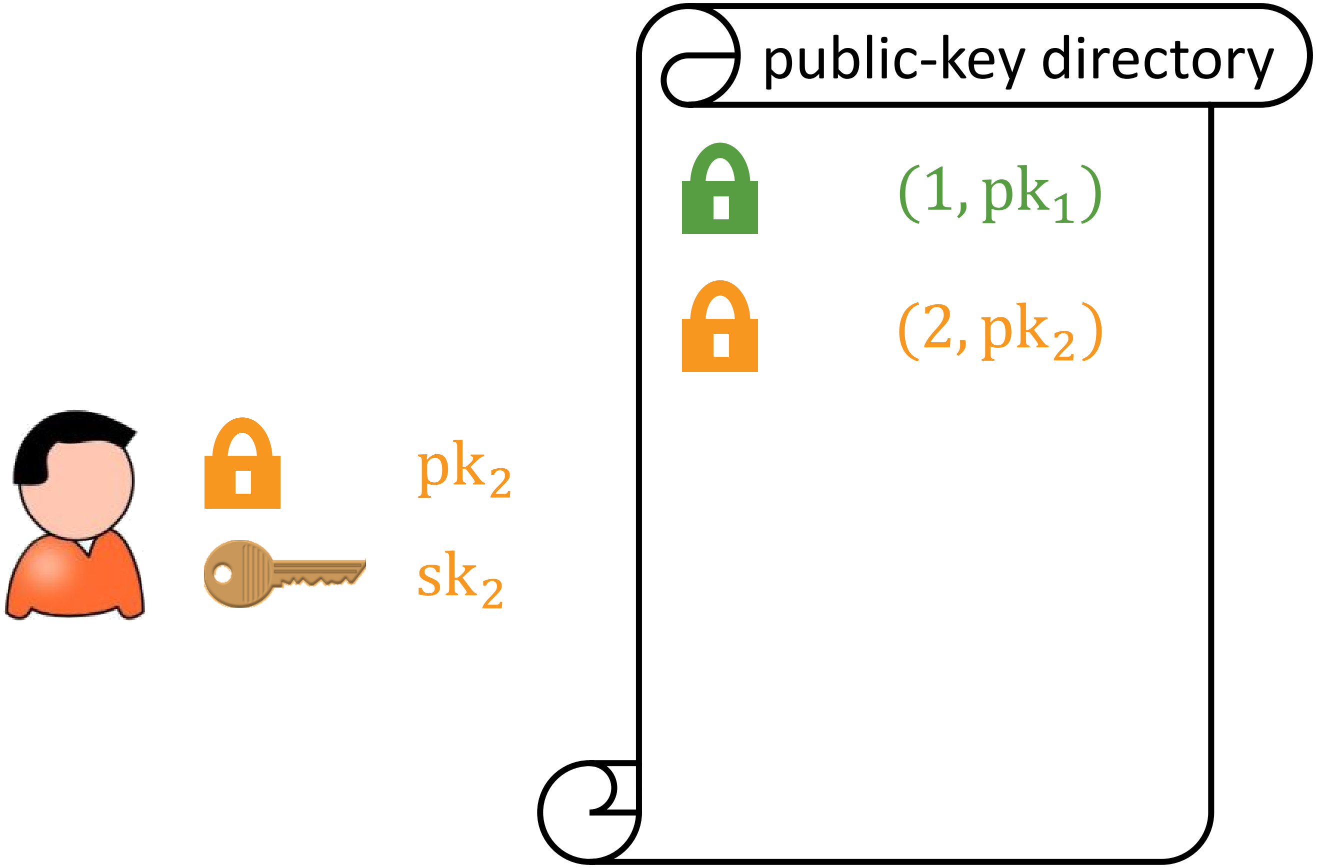


Users generate public/private keys independently (as in public-key encryption)

Broadcast encryption without a central authority

Distributed Broadcast Encryption

[BZ14]

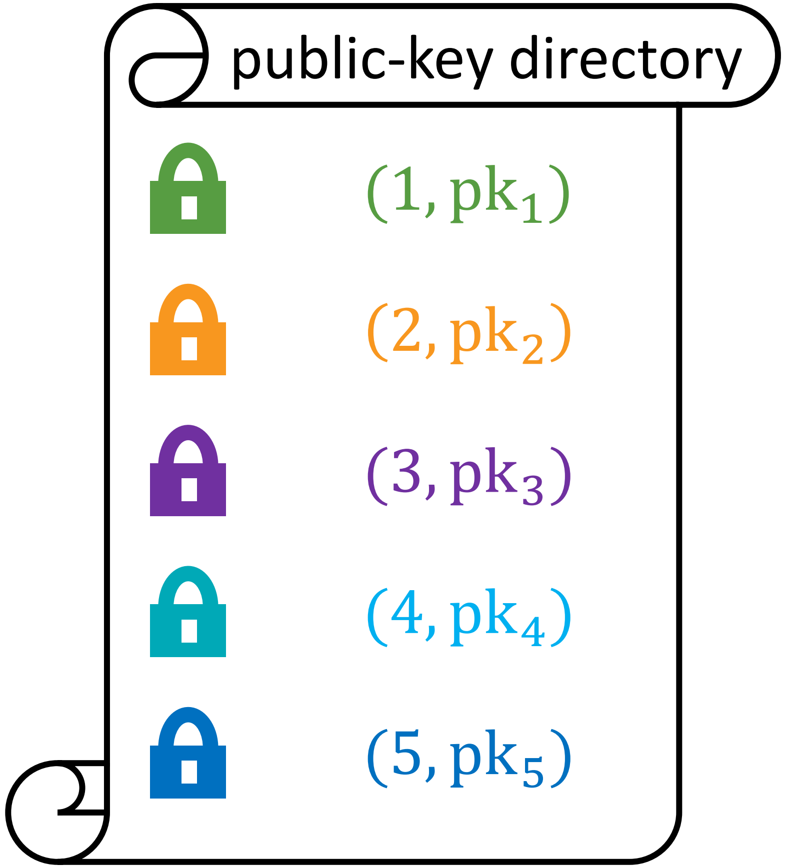


Users generate public/private keys independently (as in public-key encryption)

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

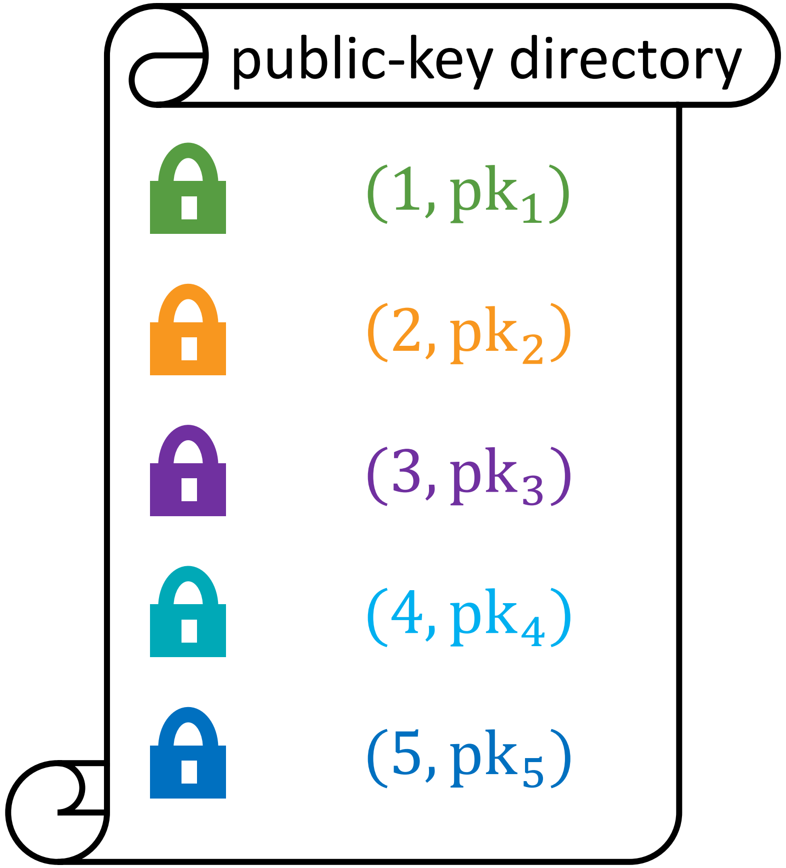
$$\mathrm{Decrypt}(\mathrm{pp}, \{\mathrm{pk}_i\}_{i \in S}, \mathrm{sk}, \mathrm{ct}) \to m$$

Any secret key associated with broadcast set can decrypt

Broadcast encryption without a central authority

Distributed Broadcast Encryption

[BZ14]



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

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

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

Broadcast encryption without a central authority

Starting Point: Centralized Broadcast Encryption

[CW24]

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

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

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

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

Each user associated with **public** matrix $\boldsymbol{U}_i \in \mathbb{Z}_q^{n \times m}$ 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 $S \subseteq [\ell]$:

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

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

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

Noise terms not shown

Starting Point: Centralized Broadcast Encryption

[CW24]

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

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

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

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

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

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

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 $\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{U}_i \boldsymbol{r}_i$

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

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

Starting Point: Centralized Broadcast Encryption

[CW24]

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

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

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

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

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

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

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

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 $\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{U}_i \boldsymbol{r}_i$

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

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

Starting Point: Centralized Broadcast Encryption

[CW24]

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

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

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

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

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

Decryption:

Suffices to recover μ from c_3

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

Starting Point: Centralized Broadcast Encryption

[CW24]

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

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

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

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

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

This is a **centralized** broadcast encryption scheme

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

Distributing the Setup

[CW24]

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

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

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

$\boldsymbol{U}_1$

$\boldsymbol{U}_2$

$\boldsymbol{U}_3$

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

Cross term: $\forall i \neq j : \boldsymbol{y}_{i,j} = \boldsymbol{A}^{-1}(\boldsymbol{U}_i \boldsymbol{r}_j)$

Secret key: $\boldsymbol{y}_{i,i} = \boldsymbol{A}^{-1}(\boldsymbol{p} + \boldsymbol{B}\boldsymbol{r}_i + \boldsymbol{U}_i \boldsymbol{r}_i)$

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

Consider first a simpler problem:

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

Distributing the Setup

[CW24]

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

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

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

$\boldsymbol{p}$

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

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

$\vdots$

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

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

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

Public parameters

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

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

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

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

A More General View

[CW24]

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

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 $j \in [\ell]$:

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

A More General View

[CW24]

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

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

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

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

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

Distributing the Setup

[CW24]

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

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

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

$\boldsymbol{U}_1$

$\boldsymbol{U}_2$

$\boldsymbol{U}_3$

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

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

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

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

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

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

Distributing the Setup

[CW24]

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

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

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

$\boldsymbol{U}_1$

$\boldsymbol{U}_2$

$\boldsymbol{U}_3$

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

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

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

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

Security relies on hardness of LWE with respect to $\boldsymbol{A}$ given trapdoor for $\boldsymbol{V}_\ell$

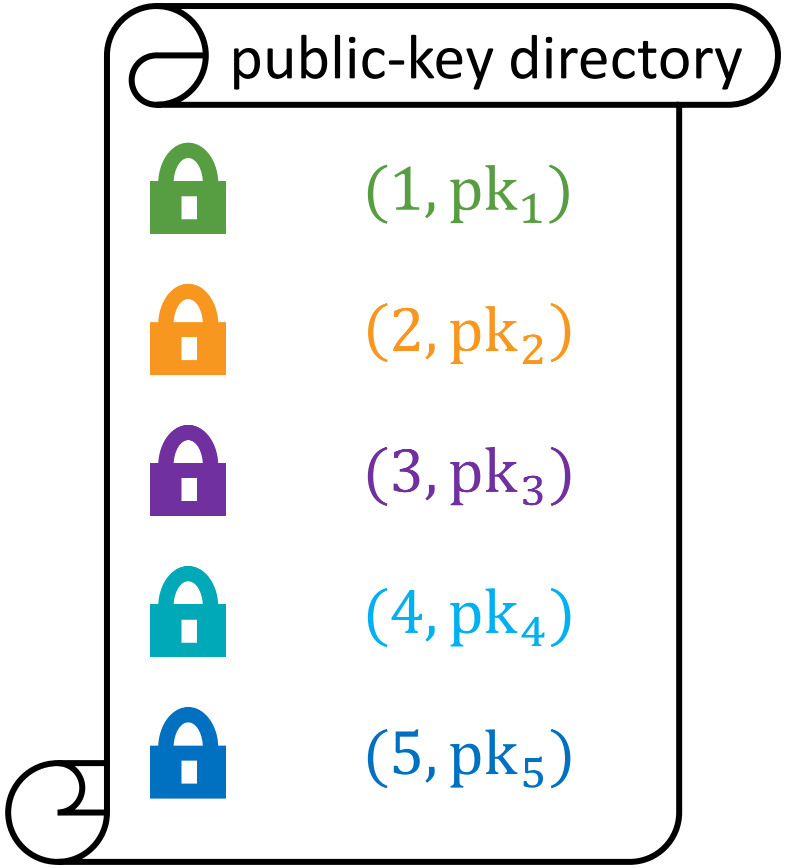
Trapdoor for $\boldsymbol{V}_\ell$ can be obtained by a succinct LWE trapdoor

Succinct LWE trapdoor: preimages of the form $\boldsymbol{A}^{-1}(\boldsymbol{W}_i \boldsymbol{r}_j)$

This trapdoor: preimages of the form $\boldsymbol{A}^{-1}(\boldsymbol{Z}(\boldsymbol{I} \otimes \boldsymbol{r}_i)\boldsymbol{d}_j) = \boldsymbol{A}^{-1}(\boldsymbol{U}_j \boldsymbol{r}_i)$

Distributed Broadcast Encryption

[CW24]



Broadcast encryption without a central authority

Distributed broadcast encryption from ℓ-succinct LWE

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

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

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

Techniques also give registered ABE for general policies in the random oracle model (also from succinct LWE) [CHW25]

Summary

More broadly: having a public trapdoor for a *structured* matrix is very useful

Trapdoor for $\boldsymbol{D}_\ell = \begin{bmatrix} \boldsymbol{A} & & & \boldsymbol{W}_1 \\ & \ddots & & \vdots \\ & & \boldsymbol{A} & \boldsymbol{W}_\ell \end{bmatrix}$ $\Rightarrow$

ABE with succinct ciphertexts [Wee24]
Functional commitments [WW23]
Distributed broadcast encryption [CW24]
(Succinct) registered ABE [CHW24]

Very useful for *compression*

security based on succinct LWE

Trapdoor for $\boldsymbol{D}_\ell = \begin{bmatrix} \boldsymbol{A}_1 & & & \boldsymbol{G} \\ & \ddots & & \vdots \\ & & \boldsymbol{A}_\ell & \boldsymbol{G} \end{bmatrix}$ $\Rightarrow$

Vector commitments [WWW24]
Dual-mode NIZK [WWW24]
Statistical ZAP arguments [BLNWW24]

security based on standard SIS/LWE

New Assumptions in Lattice-Based Cryptography

Evasive LWE:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}\left(1^\lambda\right), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

New Assumptions in Lattice-Based Cryptography

Evasive LWE:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \mathrm{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Powerful framework (has enabled many applications)

Number of counter-examples for private-coin version

New Assumptions in Lattice-Based Cryptography

Evasive LWE:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \mathrm{aux}) \leftarrow \mathrm{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n \times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \text{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \mathrm{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \mathrm{aux}$

Succinct LWE:

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

Falsifiable, instance-independent, still versatile

New Assumptions in Lattice-Based Cryptography

Evasive LWE:

For all efficient samplers Samp and taking $(\boldsymbol{P}, \text{aux}) \leftarrow \text{Samp}(1^\lambda), \boldsymbol{A} \leftarrow \mathbb{Z}_q^{n\times m}, \boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

if	$\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{P}] \approx \text{random}$	given $\boldsymbol{A}, \boldsymbol{P}, \text{aux}$
then	$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \approx$ random	given $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{A}^{-1}(\boldsymbol{P}), \text{aux}$

Succinct LWE:

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

Lots more work to be done!

Understanding hardness (e.g., worst-case/average-case reductions)

Cryptanalysis of the assumption (e.g., how does ℓ or width of $\boldsymbol{W}$ affect security)

New applications (e.g., witness encryption)

Simpler assumptions (e.g., do we need a trapdoor)

Thanks!