Succinct Matrix Commitments and Their Applications

David Wu

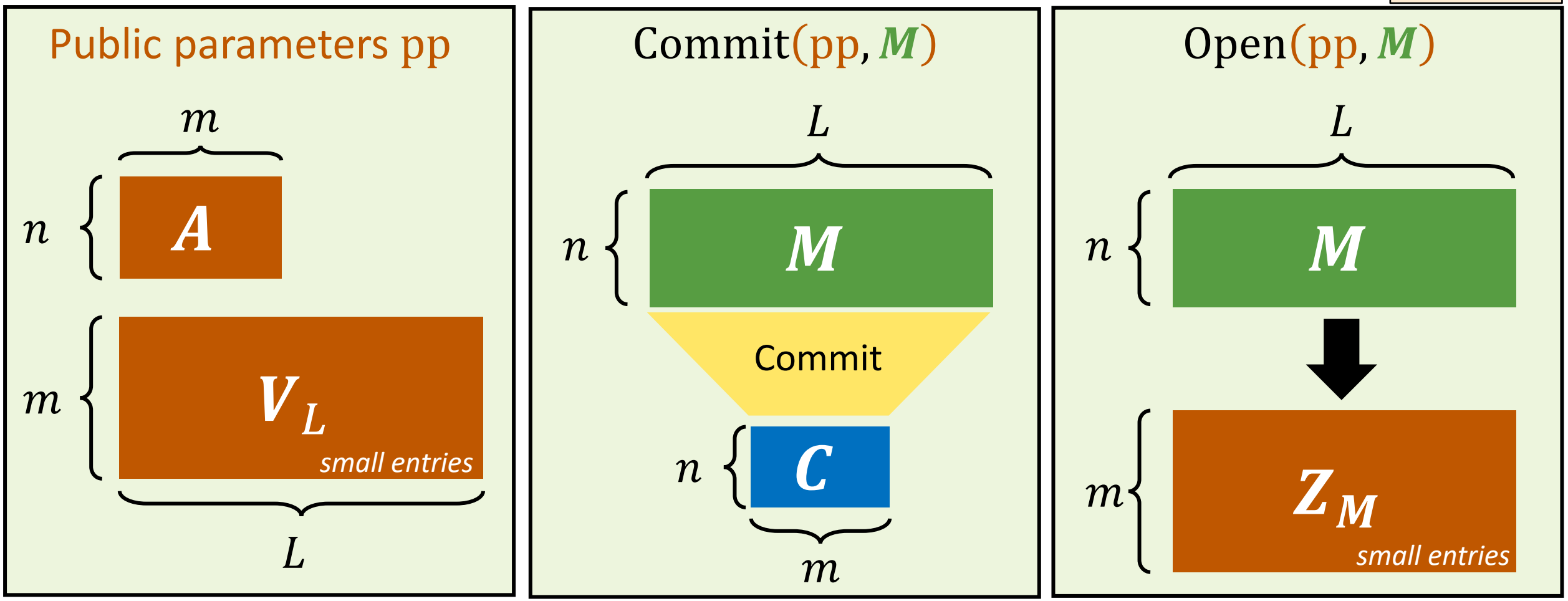
based on joint works with James Bartusek, Nir Bitansky, Jeff Champion, Yevgeniy Dodis, Rachit Garg, Yao-Ching Hsieh, Hoeteck Wee, Daniel Wichs, and Shota Yamada

Matrix Commitments

[Wee24, Wee25]

Succinct commitment to a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times L}$

$m > n \log q$
$L \gg n$



compress the matrix $\boldsymbol{M}$ into a digest $\boldsymbol{C}$

Matrix Commitments

[Wee24, Wee25]

Succinct commitment to a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times L}$

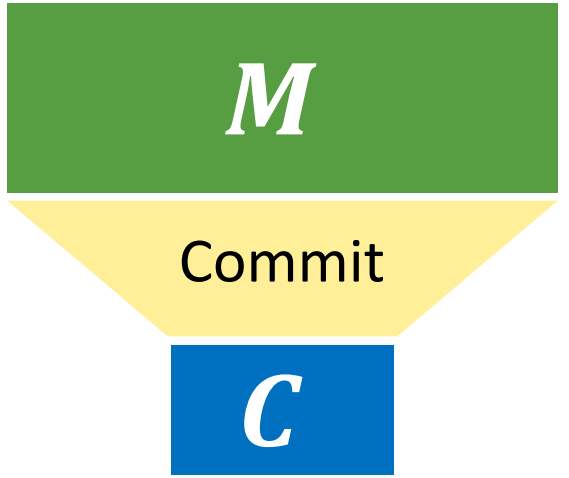
$m > n \log q$

$\text{Commit}(\text{pp}, \boldsymbol{M}) \rightarrow \boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$ $\quad$ $\text{Open}(\text{pp}, \boldsymbol{M}) \rightarrow \boldsymbol{Z_M} \in \mathbb{Z}^{m \times L}$ *low-norm*

---

public parameters $\text{pp} = (\boldsymbol{A}, \boldsymbol{V}_L)$

$\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ $\quad$ $\boldsymbol{V}_L \in \mathbb{Z}^{m \times L}$ *low-norm*



Key opening relation: $\boldsymbol{C} \cdot \boldsymbol{V}_L = \boldsymbol{M} - \boldsymbol{A} \cdot \boldsymbol{Z_M}$

$\boldsymbol{C}$ is a compressed representation of $\boldsymbol{M}$ (width of $\boldsymbol{C}$ independent of $\boldsymbol{M}$)

Can expand $\boldsymbol{C}$ to translation of $\boldsymbol{M}$ by multiplying by a fixed low-norm matrix $\boldsymbol{V_L}$

$\boldsymbol{Z_M}$ must be "taller" since its elements are small (will be taller than $\boldsymbol{M}$ by $> \log q$ factor)

Matrix Commitments

[Wee24, Wee25]

Succinct commitment to a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times L}$

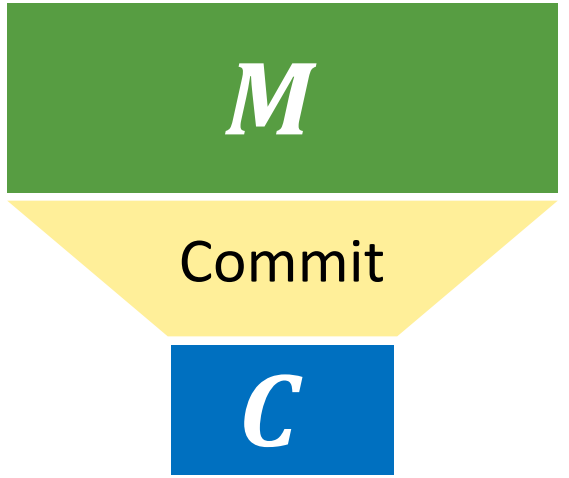
$m > n \log q$

$\text{Commit}(\text{pp}, \boldsymbol{M}) \to \boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$ $\quad$ $\text{Open}(\text{pp}, \boldsymbol{M}) \to \boldsymbol{Z_M} \in \mathbb{Z}^{m \times L}$ *low-norm*

---

public parameters $\text{pp} = (\boldsymbol{A}, \boldsymbol{V}_L)$

$\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ $\quad$ $\boldsymbol{V}_L \in \mathbb{Z}^{m \times L}$ *low-norm*



Key opening relation: $\boldsymbol{C} \cdot \boldsymbol{V}_L = \boldsymbol{M} - \boldsymbol{A} \cdot \boldsymbol{Z_M}$

Security assumption: SIS or LWE problem is hard with respect to $\boldsymbol{A}$ given pp

SIS problem: given $\boldsymbol{A}$, find low-norm $\boldsymbol{0} \neq \boldsymbol{x} \in \mathbb{Z}^m$ where $\boldsymbol{Ax} = 0$

LWE problem: distinguish $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}$ from random when $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ and $\boldsymbol{e} \in \mathbb{Z}^m$ is low-norm

Matrix Commitments

[Wee24, Wee25]

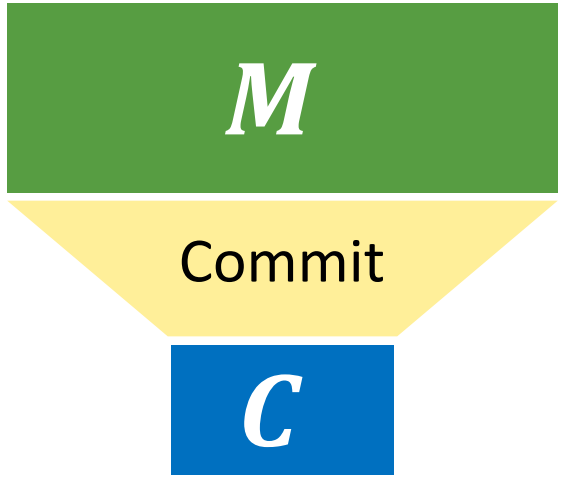
Succinct commitment to a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times L}$

$m > n \log q$

$\text{Commit}(\text{pp}, \boldsymbol{M}) \to \boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$

$\text{Open}(\text{pp}, \boldsymbol{M}) \to \boldsymbol{Z_M} \in \mathbb{Z}^{m \times L}$ *low-norm*

Security assumptions hold under the succinct/decomposed SIS/LWE assumptions



Key opening relation: $\boldsymbol{C} \cdot \boldsymbol{V}_L = \boldsymbol{M} - \boldsymbol{A} \cdot \boldsymbol{Z_M}$

Security assumption: SIS or LWE problem is hard with respect to $\boldsymbol{A}$ given pp

SIS problem: given $\boldsymbol{A}$, find low-norm $\boldsymbol{0} \neq \boldsymbol{x} \in \mathbb{Z}^m$ where $\boldsymbol{Ax} = 0$

LWE problem: distinguish $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}$ from random when $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ and $\boldsymbol{e} \in \mathbb{Z}^m$ is low-norm

So What is it Good For?

Succinct commitment to a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times L}$

$\mathrm{Commit}(\mathrm{pp}, \boldsymbol{M}) \to \boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$

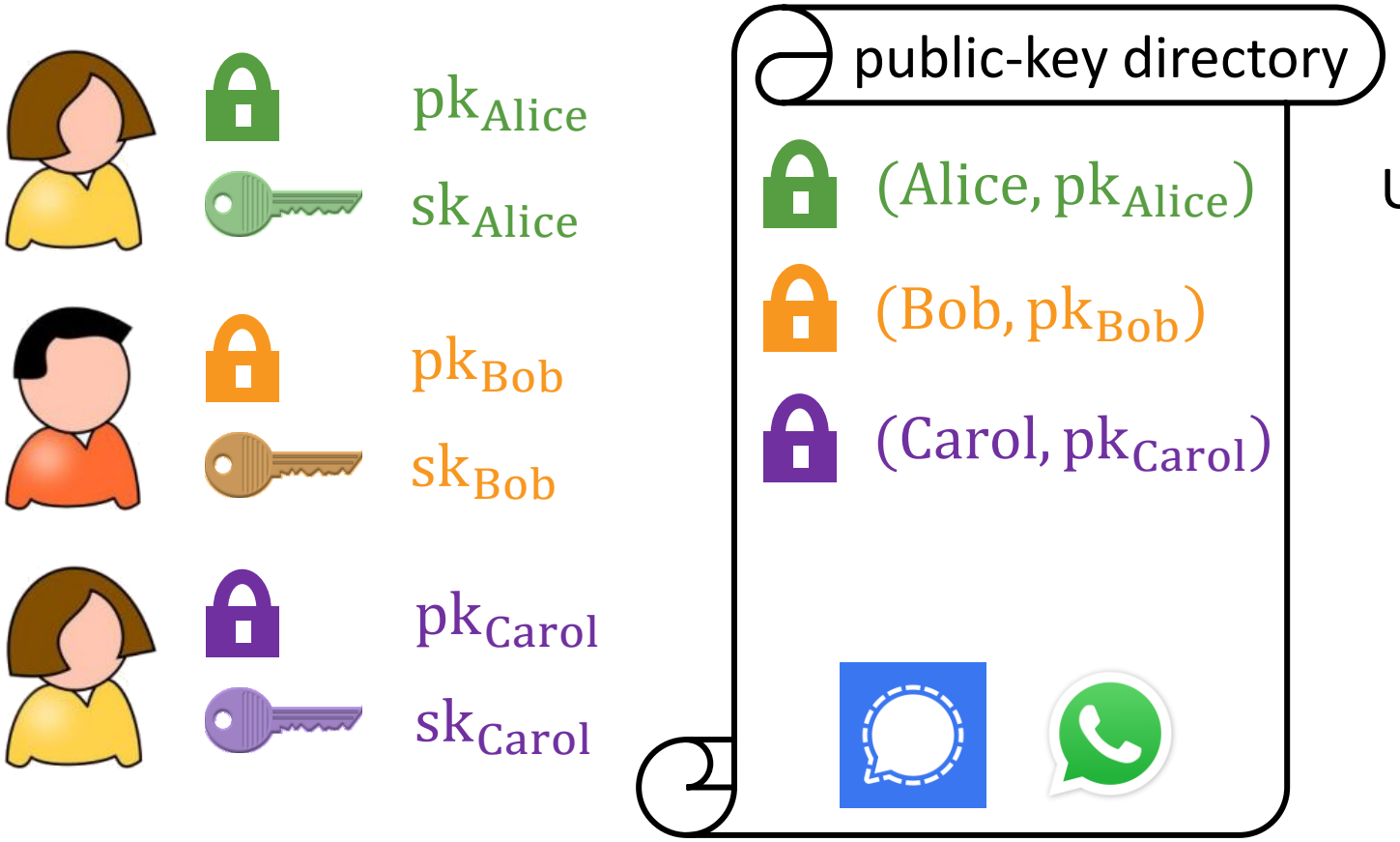
$\mathrm{Open}(\mathrm{pp}, \boldsymbol{M}) \to \boldsymbol{Z_M} \in \mathbb{Z}^{m \times L}$

$\mathrm{pp} = \left(\boldsymbol{A} \in \mathbb{Z}^{n \times m}, \boldsymbol{V}_L \in \mathbb{Z}^{m \times L}\right)$

$\boldsymbol{C} \cdot \boldsymbol{V}_L = \boldsymbol{M} - \boldsymbol{A} \cdot \boldsymbol{Z_M}$

Functional commitments for all circuits + SNARGs for P/poly	[WW23a, WW23b, Wee24, Wee25]
Optimal broadcast encryption	[Wee25, AMR26, GY26a]
Distributed broadcast encryption	[CW24, CHW25, WW25, AMR26, CW26, GY26a, GY26b]
Secure group messaging	[BBDGW26]
Silent threshold encryption	[GY26b, GY26c, CWY26]
Nearly-optimal KP-ABE and CP-ABE for circuits	[Wee24, Wee25]
ABE for bounded-space Turing machines (and other uniform models)	[AMR26, WWW26]
Registered ABE for circuits	[CHW25, AMR25, WW25]
Succinct computational secret sharing for DNFs	[CW26, GY26c]
Complexity separations: $\mathrm{P} \neq \mathrm{NC}$ and $\mathrm{P} \neq \mathrm{L}$	[AMR26b]

Group Messaging and Broadcast Encryption

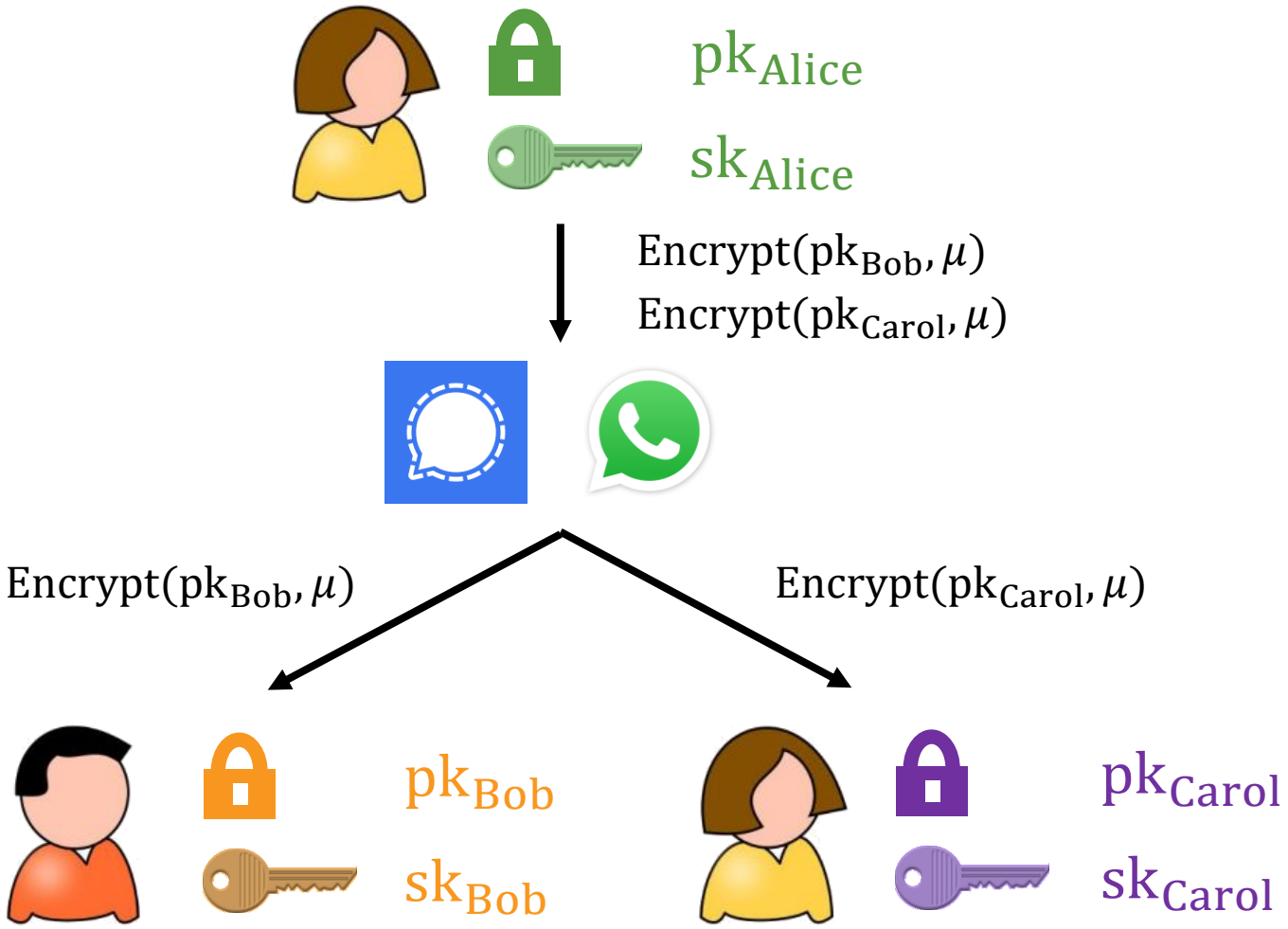


Users generate public/private keys independently

Goal: Group of users want to initiate an encrypted group chat

Group Messaging and Broadcast Encryption

Naïve solution: to send a message, user encrypts the message to each group member



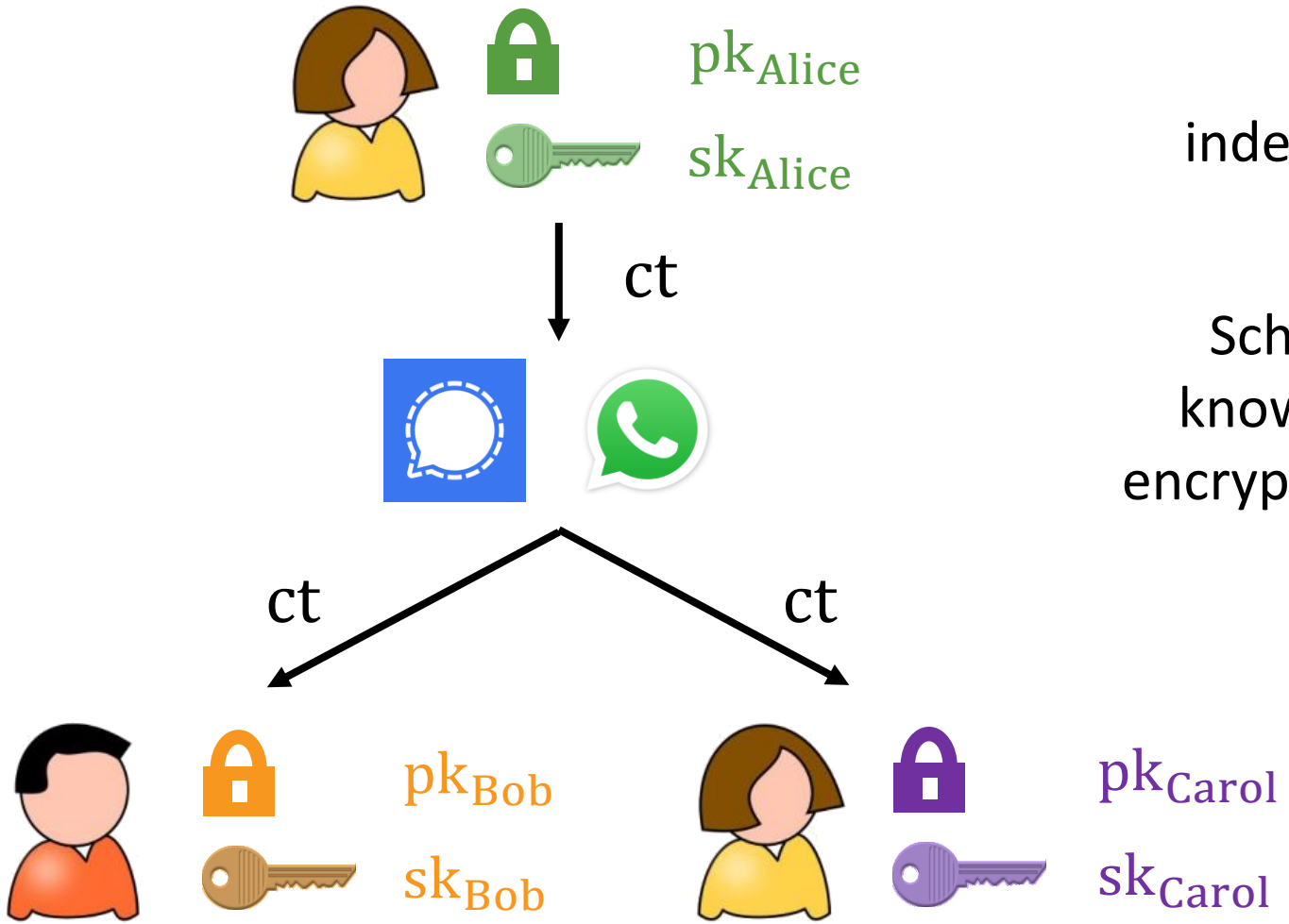
Alice's communication is **linear** in the size of the group

Question: can we compress the ciphertext?

Can we encrypt a message to N public keys $pk_1, \ldots, pk_N$ with ciphertext size $o(N)$?

All the public keys are sampled **independently**

Group Messaging and Broadcast Encryption

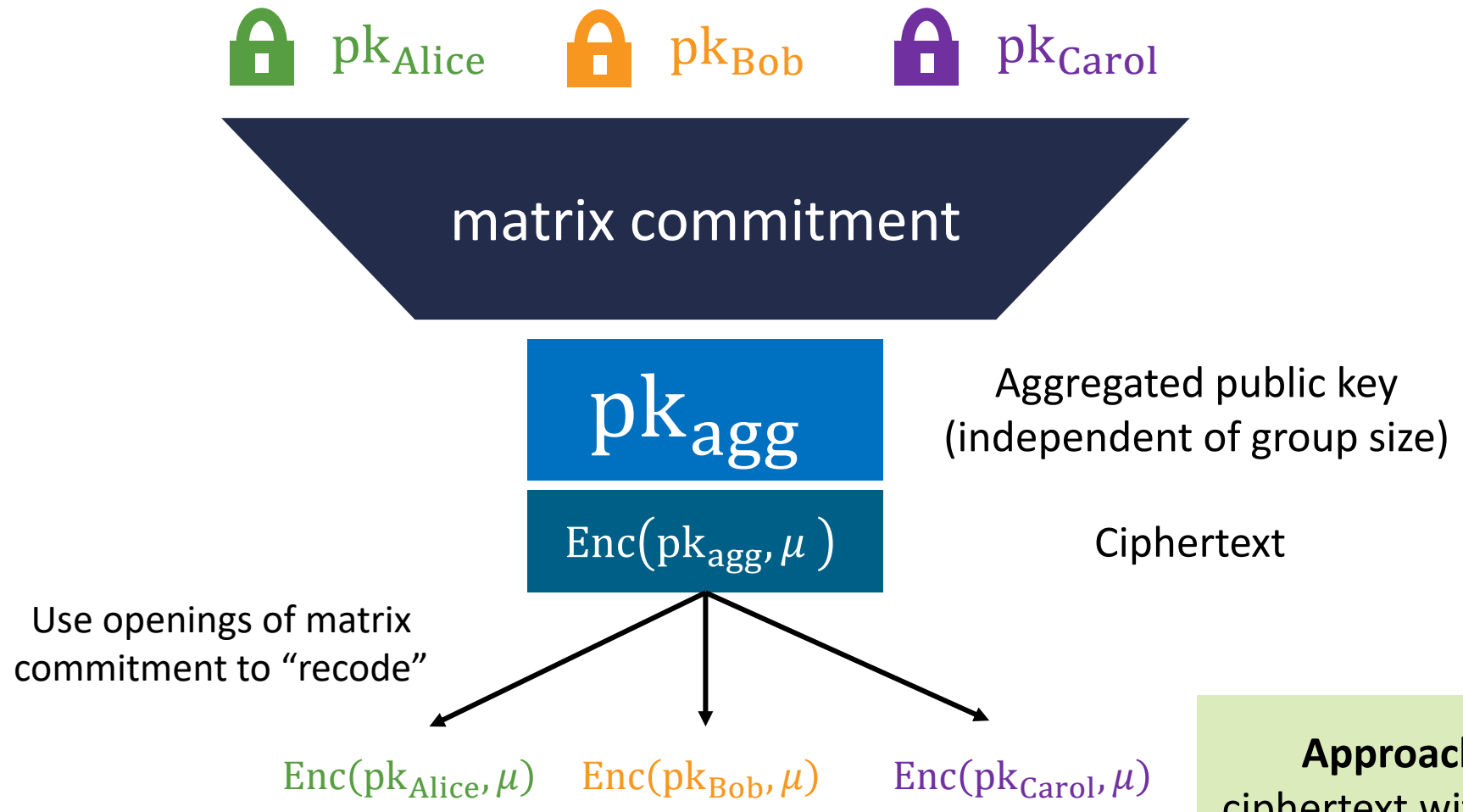


Goal: size of the ciphertext ct is independent of the number of receivers

Scheme with optimal parameters only known from strong tools such as witness encryption or indistinguishability obfuscation

Group Messaging and Broadcast Encryption

[WW25, BBDGW26]



Approach: "recode" an LWE ciphertext with respect to $\mathrm{pk}_{\mathrm{agg}}$ to a ciphertext with respect to $\mathrm{pk}_{\mathrm{user}}$

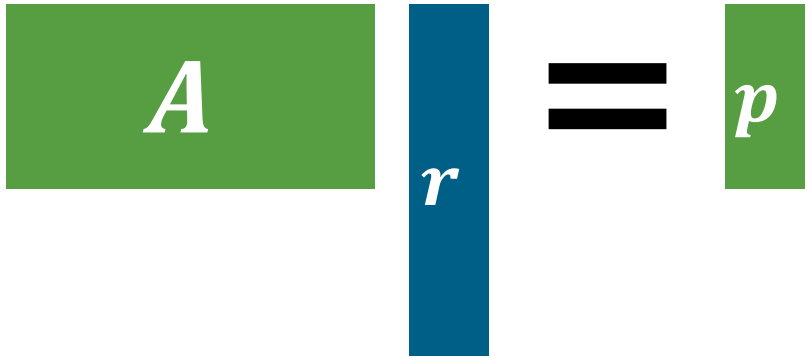
Group Messaging and Broadcast Encryption

[WW25, BBDGW26]

Starting point: dual Regev encryption

pk: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, $\boldsymbol{p} \in \mathbb{Z}_q^n$

sk: low-norm $\boldsymbol{r}$ where $\boldsymbol{Ar} = \boldsymbol{p}$



Ciphertext (encrypting $\mu \in \{0,1\}$):

$\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ $\qquad$ $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}}$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_2 + \mu \cdot \lfloor q/2 \rfloor$

Multiply by $\boldsymbol{r}$ ↓

Decryption: $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{r}$ → $\mu \cdot \lfloor q/2 \rfloor + (e_2 - \boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{r})$

$\|\boldsymbol{r}\|, \|\boldsymbol{e}_1\|, |e_2| \ll q \;\Rightarrow\; |e_2 - \boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{r}| \ll q$

Learning with errors (LWE)

$[\boldsymbol{A} \mid \boldsymbol{p}]$, $\boldsymbol{s}^{\mathrm{T}}[\boldsymbol{A} \mid \boldsymbol{p}] + [\boldsymbol{e}_1^{\mathrm{T}} \mid e_2]$

is pseudorandom

Group Messaging and Broadcast Encryption

[WW25, BBDGW26]

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





$$\mathrm{pk}_1 = \boldsymbol{t}_1 = \boldsymbol{A}\boldsymbol{r}_1 + \boldsymbol{p} \in \mathbb{Z}_q^n$$

$$\mathrm{sk}_1 = \boldsymbol{r}_1 \leftarrow \{0,1\}^m$$





$$\mathrm{pk}_2 = \boldsymbol{t}_2 = \boldsymbol{A}\boldsymbol{r}_2 + \boldsymbol{p} \in \mathbb{Z}_q^n$$

$$\mathrm{sk}_2 = \boldsymbol{r}_2 \leftarrow \{0,1\}^m$$

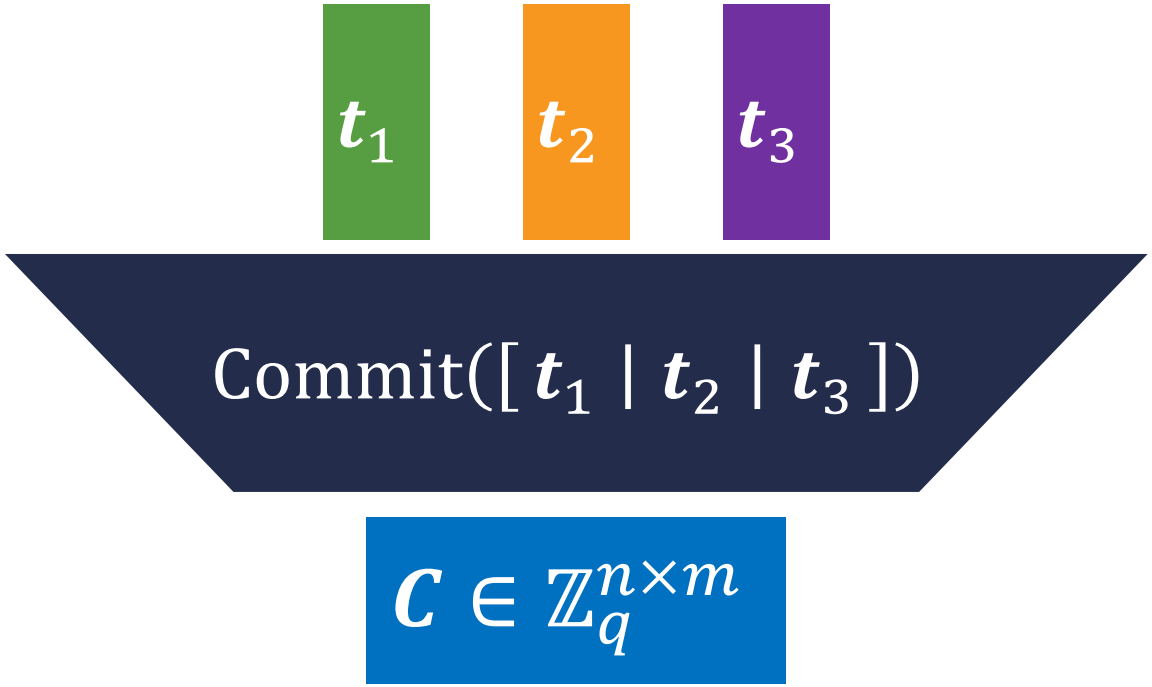




$$\mathrm{pk}_3 = \boldsymbol{t}_3 = \boldsymbol{A}\boldsymbol{r}_3 + \boldsymbol{p} \in \mathbb{Z}_q^n$$

$$\mathrm{sk}_3 = \boldsymbol{r}_3 \leftarrow \{0,1\}^m$$

Aggregated group key:

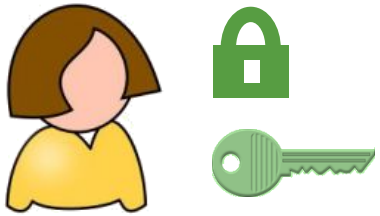


Invariant: $\boldsymbol{C}\boldsymbol{v}_i = \boldsymbol{t}_i - \boldsymbol{A}\boldsymbol{z}_i$ where $\boldsymbol{v}_i, \boldsymbol{z}_i$ are low-norm

Group Messaging and Broadcast Encryption

[WW25, BBDGW26]

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



$$\mathrm{pk}_1 = \boldsymbol{t}_1 = \boldsymbol{A}\boldsymbol{r}_1 + \boldsymbol{p} \in \mathbb{Z}_q^n$$

$$\mathrm{sk}_1 = \boldsymbol{r}_1 \leftarrow \{0,1\}^m$$

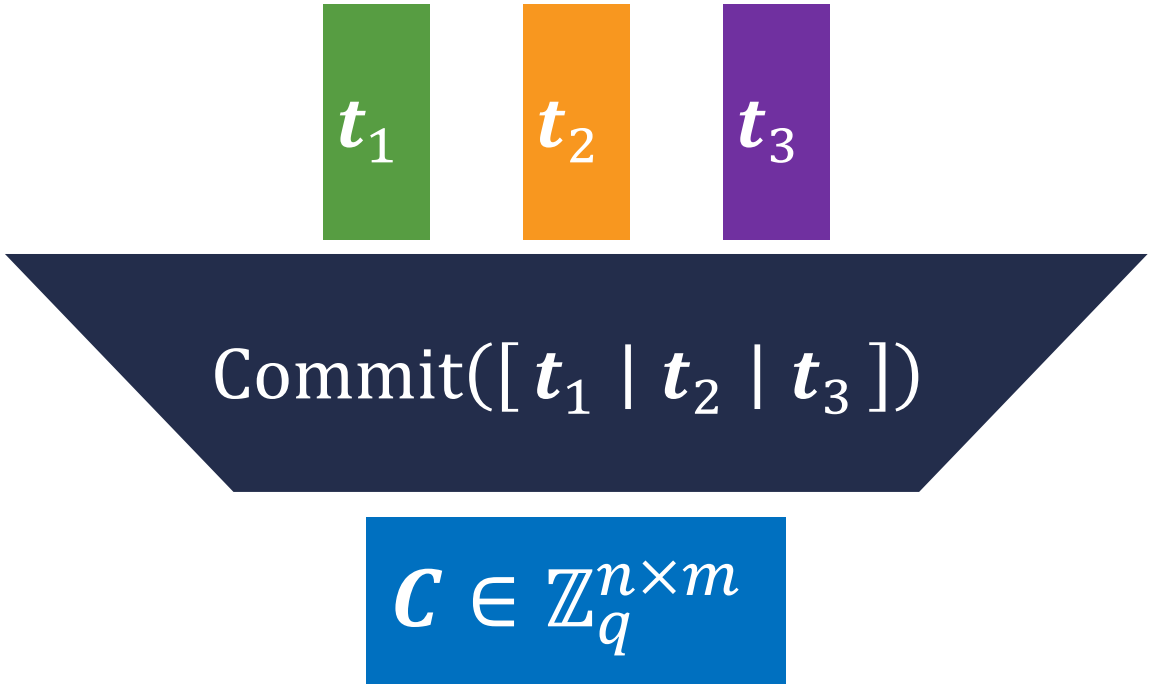
Ciphertext:

(suppressing error terms)

$\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ $\qquad$ $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rfloor$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C}$

Use $\boldsymbol{v}_i, \boldsymbol{z}_i$ to recode to encryption under $\boldsymbol{t}_i$:

Aggregated group key:

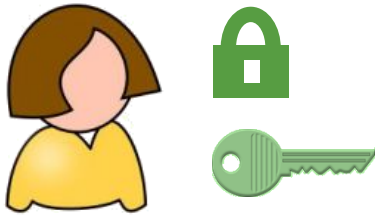


Invariant: $\boldsymbol{C}\boldsymbol{v}_i = \boldsymbol{t}_i - \boldsymbol{A}\boldsymbol{z}_i$ where $\boldsymbol{v}_i, \boldsymbol{z}_i$ are low-norm

Group Messaging and Broadcast Encryption

[WW25, BBDGW26]

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



$$\mathrm{pk}_1 = \boldsymbol{t}_1 = \boldsymbol{A}\boldsymbol{r}_1 + \boldsymbol{p} \in \mathbb{Z}_q^n$$

$$\mathrm{sk}_1 = \boldsymbol{r}_1 \leftarrow \{0,1\}^m$$

Ciphertext: *(suppressing error terms)*

$\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ $\qquad$ $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rfloor$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C}$

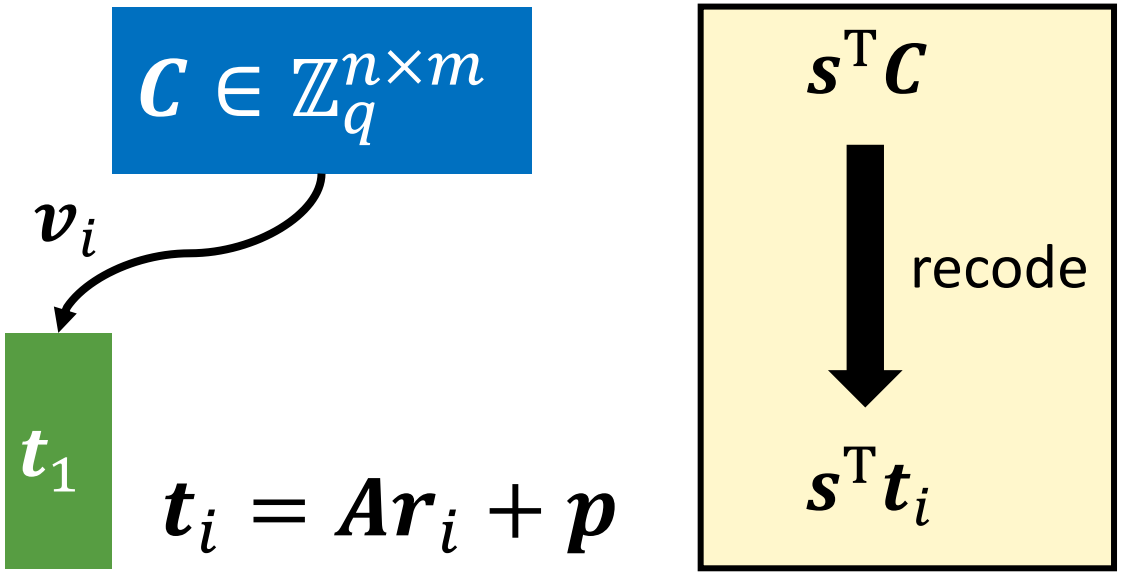
Use $\boldsymbol{v}_i, \boldsymbol{z}_i$ to recode to encryption under $\boldsymbol{t}_i$:

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C} \cdot \boldsymbol{v}_i + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \cdot \boldsymbol{z}_i \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{t}_i - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{z}_i + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{z}_i$$

$$= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{t}_i = \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{r}_i + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p}$$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C} \cdot \boldsymbol{v}_i + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \cdot \boldsymbol{z}_i - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \cdot \boldsymbol{r}_i \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p}$$

Aggregated group key:

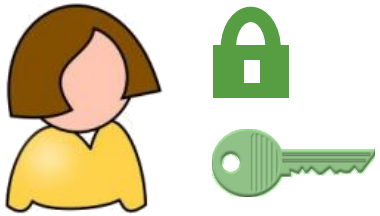


Invariant: $\boldsymbol{C}\boldsymbol{v}_i = \boldsymbol{t}_i - \boldsymbol{A}\boldsymbol{z}_i$ where $\boldsymbol{v}_i, \boldsymbol{z}_i$ are low-norm

Group Messaging and Broadcast Encryption

[WW25, BBDGW26]

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



$$\mathrm{pk}_1 = \boldsymbol{t}_1 = \boldsymbol{A}\boldsymbol{r}_1 + \boldsymbol{p} \in \mathbb{Z}_q^n$$

$$\mathrm{sk}_1 = \boldsymbol{r}_1 \leftarrow \{0,1\}^m$$

Ciphertext:

(suppressing error terms)

$\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ $\quad$ $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rfloor$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C}$

Overall ciphertext size:

$$|\mathrm{ct}| = \mathrm{poly}(\lambda)$$

(independent of group size)

Aggregated group key:

$\boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$

$\boldsymbol{v}_i$

$\boldsymbol{t}_1$

$$\boldsymbol{t}_i = \boldsymbol{A}\boldsymbol{r}_i + \boldsymbol{p}$$

$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C}$

recode

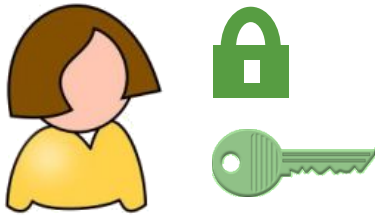
$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{t}_i$

Invariant: $\boldsymbol{C}\boldsymbol{v}_i = \boldsymbol{t}_i - \boldsymbol{A}\boldsymbol{z}_i$ where $\boldsymbol{v}_i, \boldsymbol{z}_i$ are low-norm

Group Messaging and Broadcast Encryption

[WW25, BBDGW26]

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



$$\mathrm{pk}_1 = \boldsymbol{t}_1 = \boldsymbol{A}\boldsymbol{r}_1 + \boldsymbol{p} \in \mathbb{Z}_q^n$$

$$\mathrm{sk}_1 = \boldsymbol{r}_1 \leftarrow \{0,1\}^m$$

Ciphertext:

(suppressing error terms)

$\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$ $\quad$ $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rfloor$, $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C}$

Can augment scheme to support incremental updates and key refreshes ⇒ continuous group key-agreement and secure group messaging

Aggregated group key:

$\boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$

$\boldsymbol{v}_i$

$\boldsymbol{t}_1$

$$\boldsymbol{t}_i = \boldsymbol{A}\boldsymbol{r}_i + \boldsymbol{p}$$

$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{C}$

recode

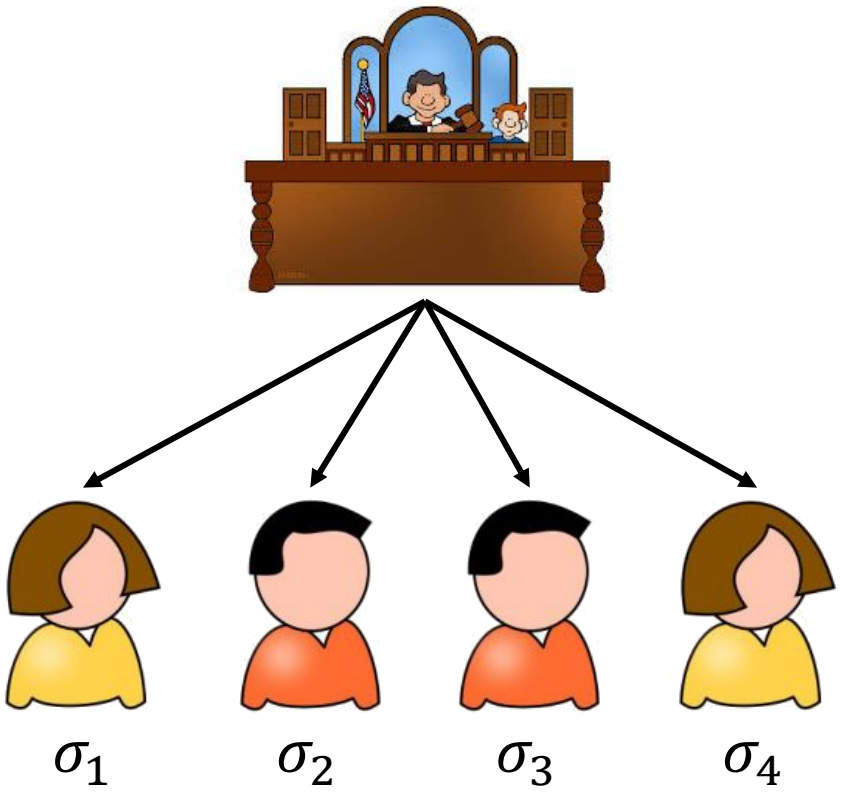
$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{t}_i$

Invariant: $\boldsymbol{C}\boldsymbol{v}_i = \boldsymbol{t}_i - \boldsymbol{A}\boldsymbol{z}_i$ where $\boldsymbol{v}_i, \boldsymbol{z}_i$ are low-norm

Succinct Computational Secret Sharing

[ABIKLV25]

secret s
monotone policy P



shares of s

Correctness: any set of users T where $P(T) = 1$ can reconstruct s

Security: shares for users T where $P(T) = 0$ should hide s

Succinctness: share size σ_i should be polylogarithmic in size of P

For general monotone policies on n-bits (e.g., truth table of size 2^n), best information-theoretic construction has share size $\approx 2^{0.585n}$

Best information-theoretic lower bound: $\Omega(n^2/\log n)$

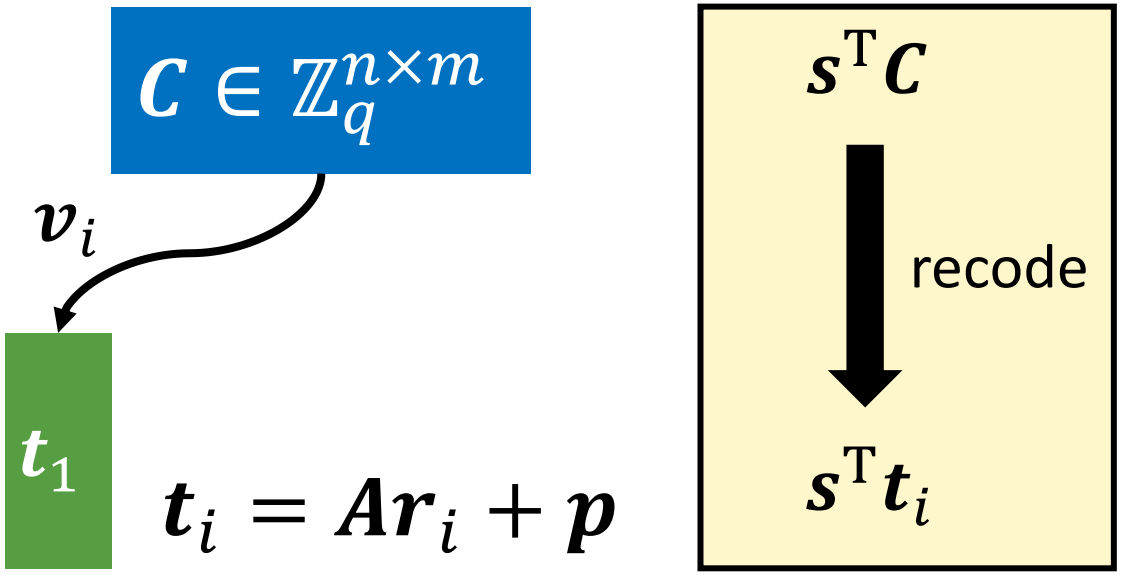
With computational assumptions: $\text{poly}(n) = \text{poly}(\log |P|)$

Succinct Computational Secret Sharing

[CW26]

Recall approach to group messaging:

$$\boldsymbol{s} \leftarrow \mathbb{Z}_q^n \qquad \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}\,,\ \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + \mu \cdot \lfloor q/2 \rfloor\,,\ \boldsymbol{s}^{\mathrm{T}}\boldsymbol{C}$$



r_i is user i's secret key

Invariant: if party i knows $\boldsymbol{r}_i$, can recover $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p}$ and decrypt

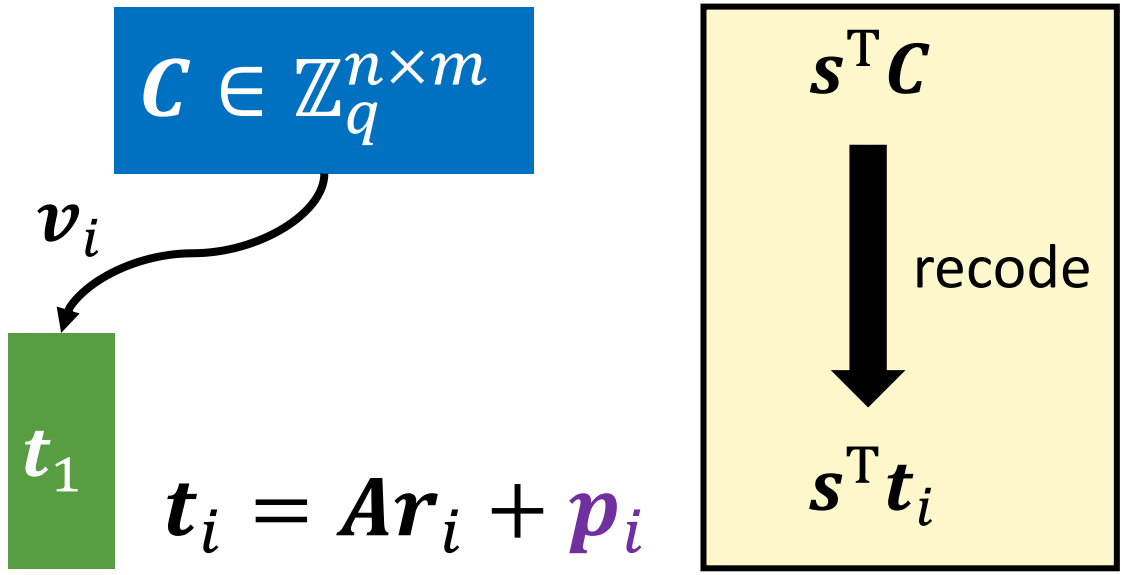
Succinct Computational Secret Sharing for DNFs

[CW26]

Recall approach to group messaging:

$$s \leftarrow \mathbb{Z}_q^n \qquad s^{\mathrm{T}}A \,,\, s^{\mathrm{T}}p + \mu \cdot \lfloor q/2 \rfloor \,,\, s^{\mathrm{T}}C$$

In more detail: secret share p according to a **non-succinct** linear secret sharing scheme, attach a public key to each share, and commit to the public keys



r_i is user i's secret key

Invariant: if party i knows r_i, can recover $s^{\mathrm{T}}p$ and decrypt

Secret sharing: if party i knows r_i, can recover $s^{\mathrm{T}}p_i$ where p_i is a share of p

Approach: commit to t_i that now recodes to p_i for each user (each party can have multiple t_i)

Conclusion: succinct CSS for DNFs with $\mathrm{poly}(\lambda)$-size shares

[some details omitted]

Homomorphic Computation using Lattices

[GSW13, BGGHNSVV14]

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



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

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

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

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

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

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

Attribute-Based Encryption

[BGGHNSVV14]

"dual Regev public key" attribute-encoding matrix

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm}$

Secret key for $\boldsymbol{f}$: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Ciphertext with attribute $\boldsymbol{x}$: $\boldsymbol{s} \leftarrow \mathbb{Z}_q^n$

$$\boxed{\begin{array}{l} \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}} \\ \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B} - \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right) + \boldsymbol{e}_2^{\mathrm{T}} \\ \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_3 + \mu \cdot \lfloor q/2 \rfloor \end{array}}$$

$$\approx \left[\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} \mid \boldsymbol{s}^{\mathrm{T}}\boldsymbol{B}_f\right]\boldsymbol{v}_f$$

multiply by $\boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}}$ $\quad \approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{B}_f$

$$\approx \boldsymbol{s}^{\mathrm{T}}\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f$$

$$\approx \boldsymbol{s}^{\mathrm{T}}\boldsymbol{p}$$

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

Attribute-Based Encryption

[BGGHNSVV14]

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm}$

"dual Regev public key" attribute-encoding matrix

Secret key for $\boldsymbol{f}$: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Ciphertext with attribute $\boldsymbol{x}$:

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}}$$
$$\boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B} - \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right) + \boldsymbol{e}_2^{\mathrm{T}}$$
$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_3 + \mu \cdot \lfloor q/2 \rfloor$$

Not succinct because $\left|\boldsymbol{B} - \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right| = \boldsymbol{L} \cdot nm \log q$

Need to encode attribute to compute on it

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

Succinct Attribute-Based Encryption

[Wee24, Wee25]

"dual Regev public key" attribute-encoding matrix

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm}$

Secret key for $\boldsymbol{f}$: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Ciphertext with attribute $\boldsymbol{x}$:

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{B} - \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right) + \boldsymbol{e}_2^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_3 + \mu \cdot \lfloor q/2 \rfloor$$

[Wee24, Wee25] approach: compress $\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}$

- Let $\boldsymbol{C}_{\boldsymbol{x}} \in \mathbb{Z}_q^{n \times m}$ be a commitment to $\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}$
- Then $\boldsymbol{C}_{\boldsymbol{x}}\boldsymbol{V} = \left(\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right) - \boldsymbol{A}\boldsymbol{Z}_{\boldsymbol{x}}$

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

Succinct Attribute-Based Encryption

[Wee24, Wee25]

"dual Regev public key" attribute-encoding matrix

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm} \longrightarrow \widetilde{\boldsymbol{B}} \in \mathbb{Z}_q^{n \times m}$

public parameters independent of attribute length!

Secret key for f: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Ciphertext with attribute x:

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\left(\widetilde{\boldsymbol{B}} - \boldsymbol{C}_{\boldsymbol{x}}\right) + \boldsymbol{e}_2^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_3 + \mu \cdot \lfloor q/2 \rfloor$$

[Wee24, Wee25] approach: compress $\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}$

- Let $\boldsymbol{C}_{\boldsymbol{x}} \in \mathbb{Z}_q^{n \times m}$ be a commitment to $\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}$
- Then $\boldsymbol{C}_{\boldsymbol{x}}\boldsymbol{V} = \left(\boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right) - \boldsymbol{A}\boldsymbol{Z}_{\boldsymbol{x}}$
- Set $\boldsymbol{B} = \widetilde{\boldsymbol{B}}\boldsymbol{V}$

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

Succinct Attribute-Based Encryption

[Wee24, Wee25]

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm}$ → $\widetilde{\boldsymbol{B}} \in \mathbb{Z}_q^{n \times m}$

"dual Regev public key" attribute-encoding matrix

public parameters independent of attribute length!

Secret key for f: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Ciphertext with attribute x: $\widetilde{\boldsymbol{B}}\boldsymbol{V} = \boldsymbol{B}$ and $\boldsymbol{C}_x\boldsymbol{V} = \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G} - \boldsymbol{A}\boldsymbol{Z}_x$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\left(\widetilde{\boldsymbol{B}} - \boldsymbol{C}_x\right) + \boldsymbol{e}_2^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_3 + \mu \cdot \lfloor q/2 \rfloor$$

$$\begin{aligned}
\left(\widetilde{\boldsymbol{B}} - \boldsymbol{C}_x\right)\boldsymbol{V} \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} &= \left(\boldsymbol{B} - \boldsymbol{x}^{\mathrm{T}} \otimes \boldsymbol{G}\right) \cdot \boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{Z}_x\boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} \\
&= \boldsymbol{B}_f - f(\boldsymbol{x}) \cdot \boldsymbol{G} - \boldsymbol{A}\boldsymbol{Z}_x\boldsymbol{H}_{\boldsymbol{B},f,\boldsymbol{x}} \\
&= \boldsymbol{B}_f - \boldsymbol{A} \cdot \textbf{low-norm} \quad \text{when } f(\boldsymbol{x}) = 0
\end{aligned}$$

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

Succinct Attribute-Based Encryption

[Wee24, Wee25]

"dual Regev public key" attribute-encoding matrix

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^{n}, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm} \longrightarrow \widetilde{\boldsymbol{B}} \in \mathbb{Z}_q^{n \times m}$

public parameters independent of attribute length!

Secret key for f: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Ciphertext with attribute x:

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\left(\widetilde{\boldsymbol{B}} - \boldsymbol{C}_{\boldsymbol{x}}\right) + \boldsymbol{e}_2^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_3 + \mu \cdot \lfloor q/2 \rfloor$$

Appealing properties:

- Public parameters independent of attribute length
- Ciphertext independent of attribute length

Natural question: can we support **uniform** policy families?

Goal: Secret key for Turing machine that works for **all** input lengths (a priori unbounded)

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

ABE for Uniform Computations

"dual Regev public key" attribute-encoding matrix

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm} \longrightarrow \widetilde{\boldsymbol{B}} \in \mathbb{Z}_q^{n \times m}$

public parameters independent of attribute length!

Secret key for f: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Ciphertext with attribute x:

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}_1^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\left(\widetilde{\boldsymbol{B}} - \boldsymbol{C}_{\boldsymbol{x}}\right) + \boldsymbol{e}_2^{\mathrm{T}}$$

$$\boldsymbol{s}^{\mathrm{T}}\boldsymbol{p} + e_3 + \mu \cdot \lfloor q/2 \rfloor$$

Appealing properties:

- Public parameters independent of attribute length
- Ciphertext independent of attribute length

Natural question: can we support **uniform** policy families?

Goal: Secret key for Turing machine that works for **all** input lengths (a priori unbounded)

Ciphertext size: fine since we can commit to inputs of any length

ABE for Uniform Computations

"dual Regev public key" attribute-encoding matrix

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm} \longrightarrow \widetilde{\boldsymbol{B}} \in \mathbb{Z}_q^{n \times m}$

public parameters independent of attribute length!

Secret key for f: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Key generation will be a problem

Given matrix $\boldsymbol{B} = \widetilde{\boldsymbol{B}}\boldsymbol{V} \in \mathbb{Z}^{n \times \ell m}$, key-generation needs to compute $\boldsymbol{B}_f$

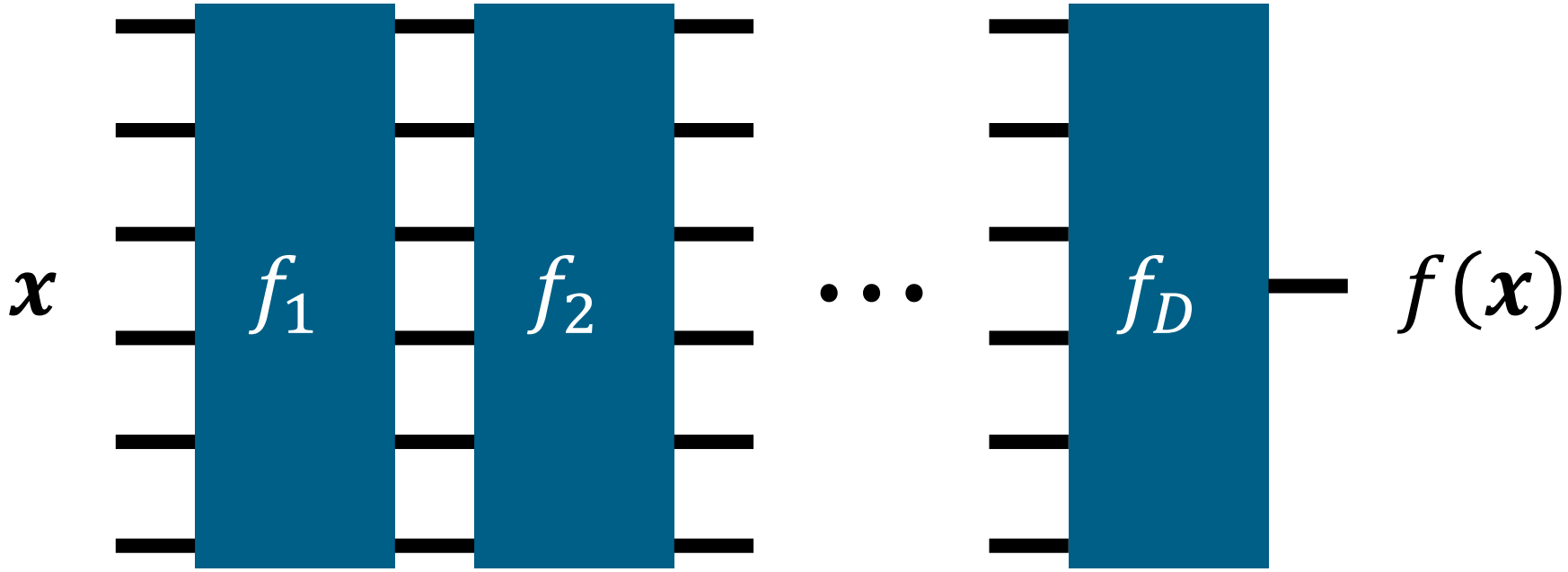
Computing $\boldsymbol{B}_f$ is homomorphic evaluation of f (as a circuit) on $\boldsymbol{B}$

In general, should not expect to compute $\boldsymbol{B}_f$ faster than evaluating f

*What if f is **structured**?*

Computing B_f Quickly

[WWW26]



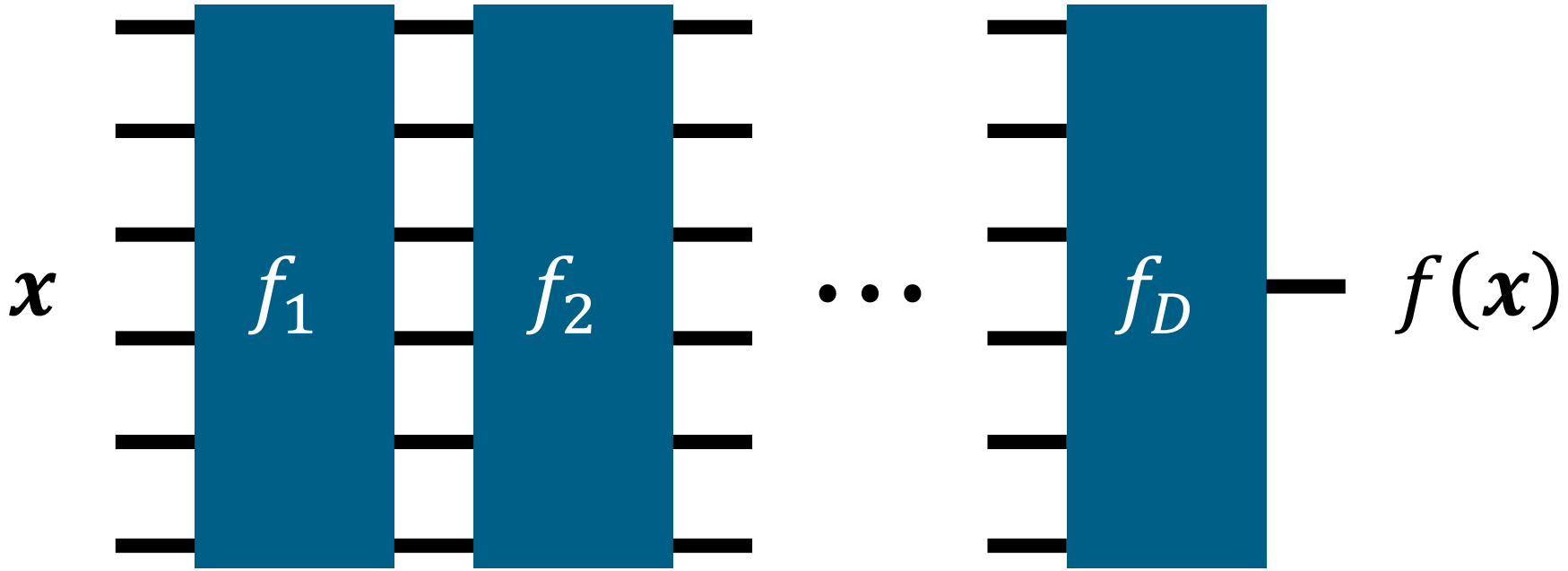
Suppose f is a layered Boolean circuit

$f_d : \{0,1\}^L \to \{0,1\}^L$ suppose f_d computes a quadratic function

Let $f_{d,k}(x_1, \ldots, x_\ell)$ be the k^{th} bit of $f_d(x_1, \ldots, x_\ell)$

Computing B_f Quickly

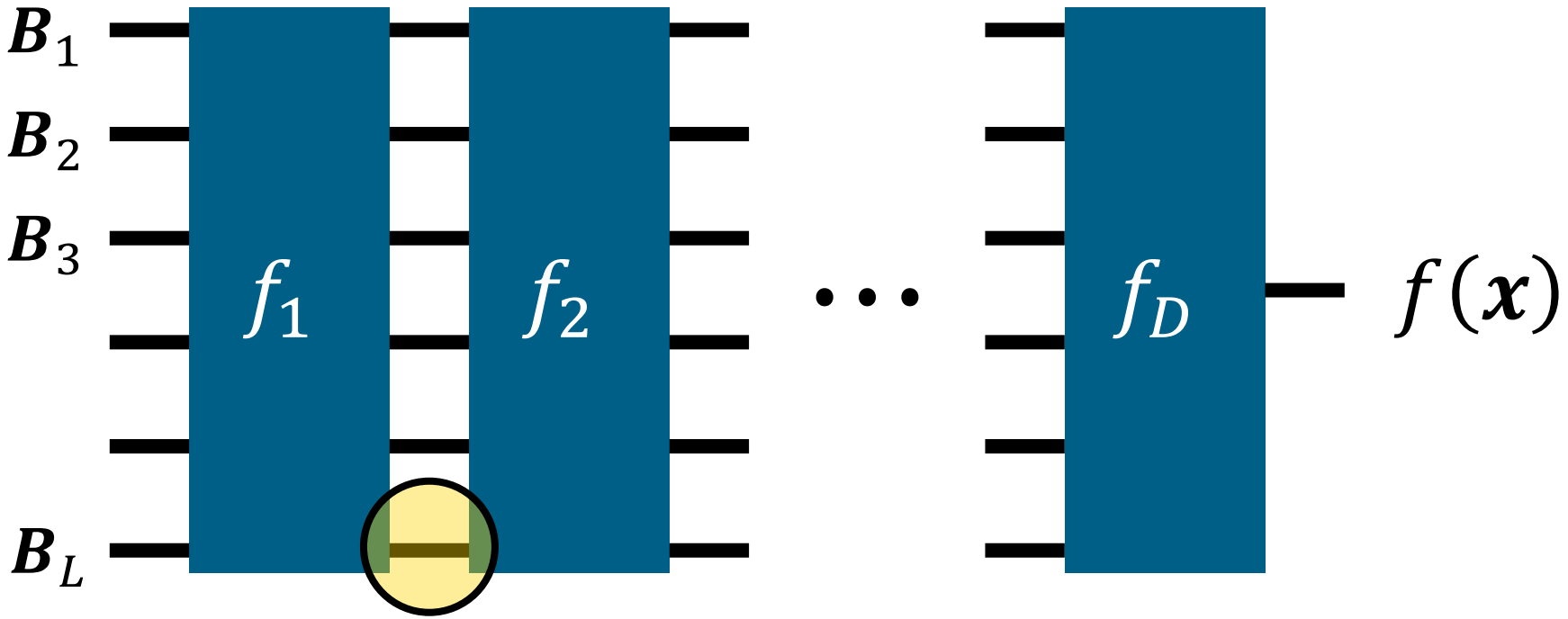
[WWW26]



What does $\boldsymbol{B}_f$ look like?

Computing B_f Quickly

[WWW26]



Associate B_i with each wire

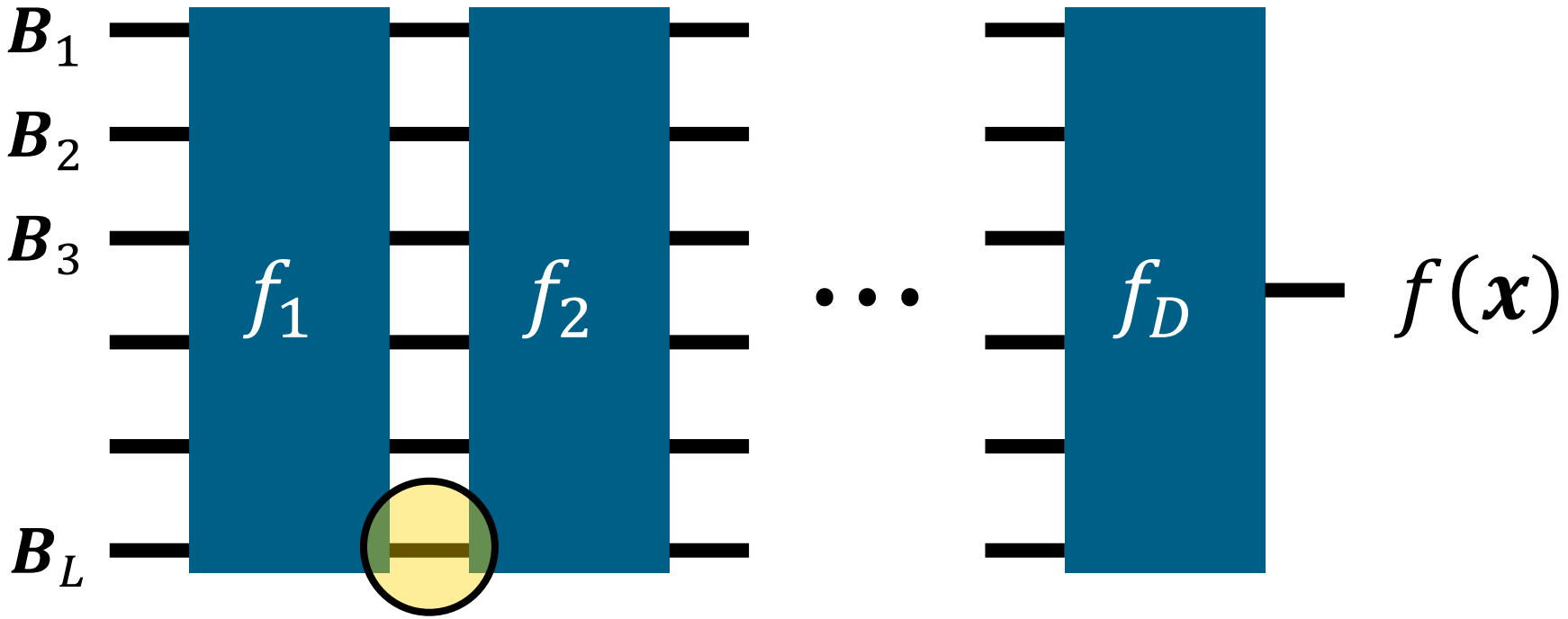
if gate computes $x_i + x_j$: associate $B_i + B_j$

if gate computes $x_i x_j$: associate $B_i G^{-1}(B_j)$

What does B_f look like?

Computing B_f Quickly

[WWW26]



Associate $\boldsymbol{B}_i$ with each wire

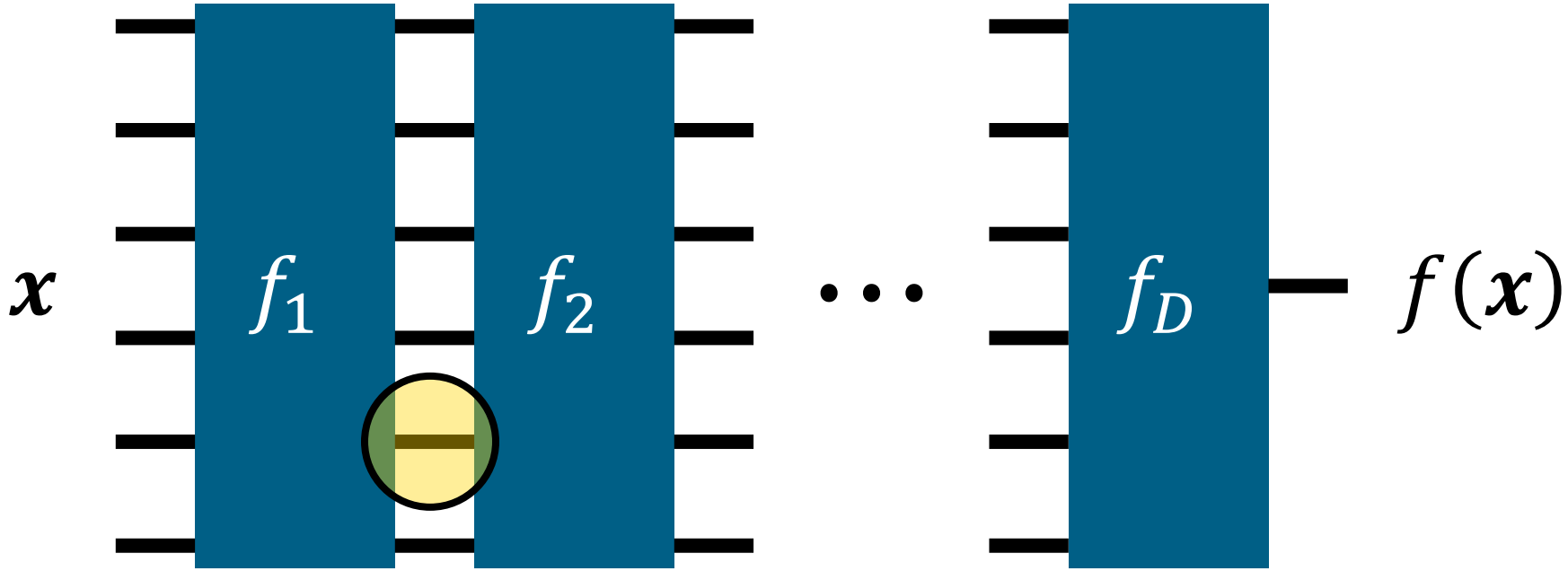
if gate computes $x_i + x_j$: associate $\boldsymbol{B}_i + \boldsymbol{B}_j$

if gate computes $x_i x_j$: associate $\boldsymbol{B}_i \boldsymbol{G}^{-1}(\boldsymbol{B}_j)$

Need to compute matrix for each wire… requires time $|f|$

Computing B_f Quickly

[WWW26]



Recall: in succinct ABE scheme from [Wee24, Wee25], we start with commitment to input x

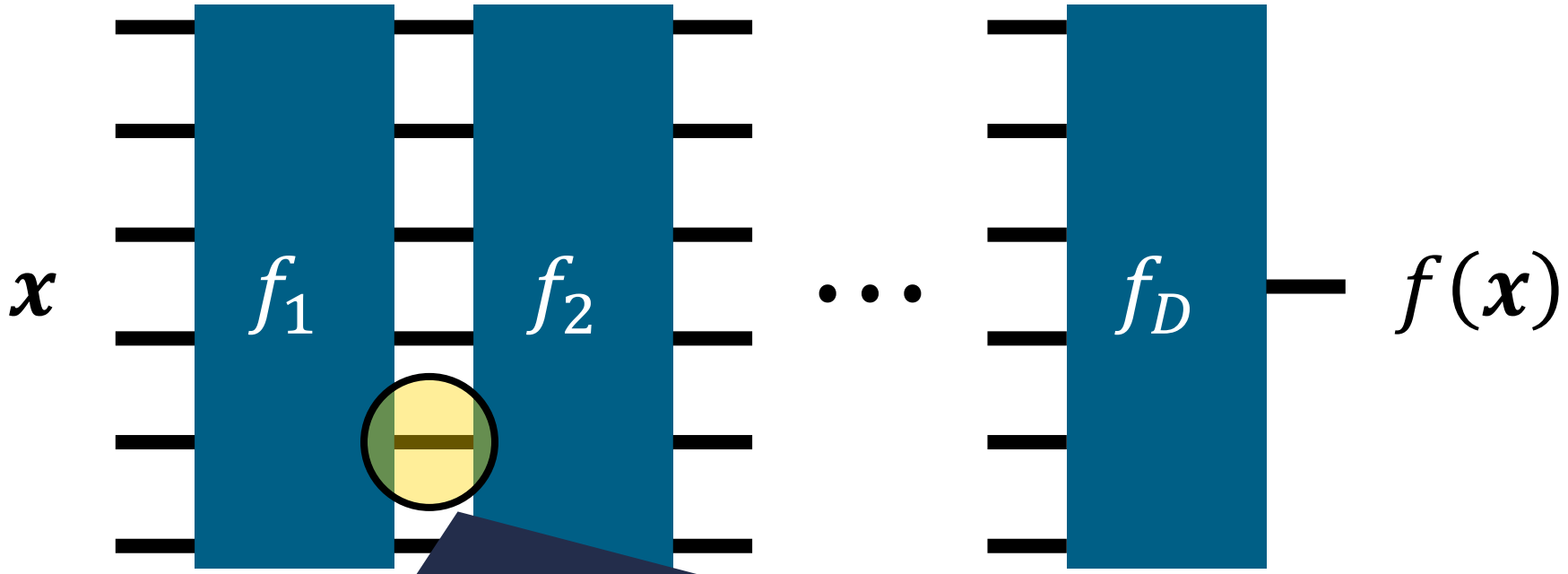
Idea: instead of computing wire-by-wire, compute *layer-by-layer* using **succinct** commitments

[AMR25, Wee25]

$$C_x = \sum_{k\in[L]} x_k G W_k \longrightarrow C_{f_1(x)} = \sum_{k\in[L]} f_{1,k}(x) G W_k$$

Computing B_f Quickly

[WWW26]



Recall: in succinct ABE

Idea: instead of comp

$$C_x = \sum_{k\in[L]} x_k G$$

nput x

mmitments

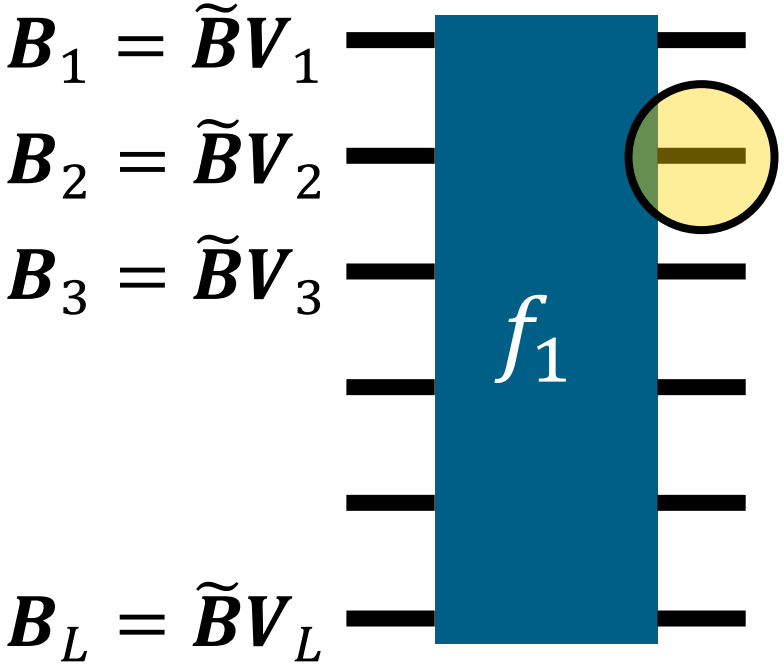
[AMR25, Wee25]

$$f_{1,k}(x_1, \ldots, x_L) = \sum_{i,j\in[L]} g_1(i,j,k) \cdot x_i x_j$$

g_1: "wiring function" for f_1

Computing B_f Quickly

[WWW26]



Associate $\boldsymbol{B}_i$
with each wire

$$f_{1,k}(x_1, \ldots, x_L) = \sum_{i,j \in [L]} g_1(i,j,k) \cdot x_i x_j$$

$$\boldsymbol{B}_{1,k} = \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \boldsymbol{B}_i \boldsymbol{G}^{-1}(\boldsymbol{B}_j)$$

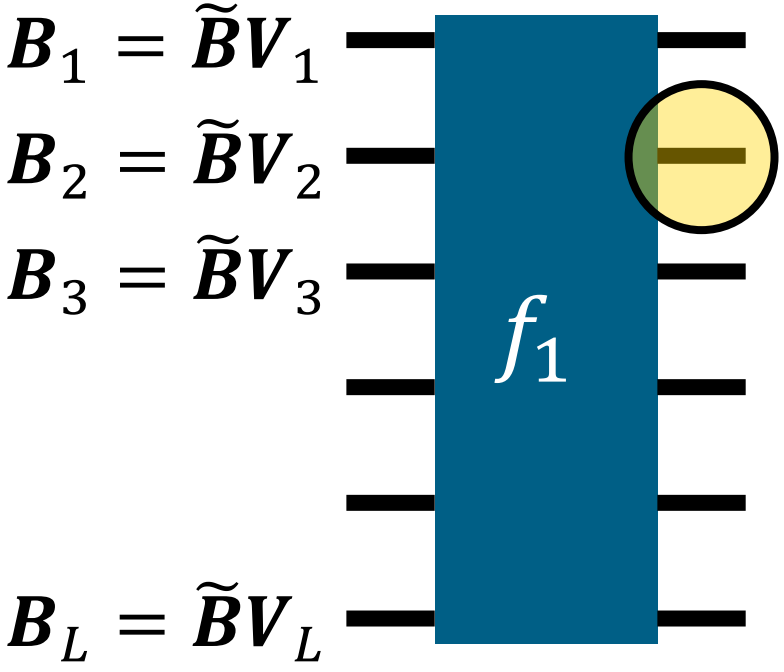
$$= \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \widetilde{\boldsymbol{B}}\boldsymbol{V}_i \boldsymbol{G}^{-1}(\widetilde{\boldsymbol{B}}\boldsymbol{V}_j)$$

if gate computes $x_i + x_j$: associate $\boldsymbol{B}_i + \boldsymbol{B}_j$

if gate computes $x_i x_j$: associate $\boldsymbol{B}_i \boldsymbol{G}^{-1}(\boldsymbol{B}_j)$

Computing B_f Quickly

[WWW26]



Associate $\boldsymbol{B}_i$ with each wire

$$f_{1,k}(x_1, \ldots, x_L) = \sum_{i,j \in [L]} g_1(i,j,k) \cdot x_i x_j$$

$$\boldsymbol{B}_{1,k} = \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \boldsymbol{B}_i \boldsymbol{G}^{-1}(\boldsymbol{B}_j)$$

$$= \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \widetilde{\boldsymbol{B}}\boldsymbol{V}_i \boldsymbol{G}^{-1}(\widetilde{\boldsymbol{B}})\boldsymbol{V}_j$$

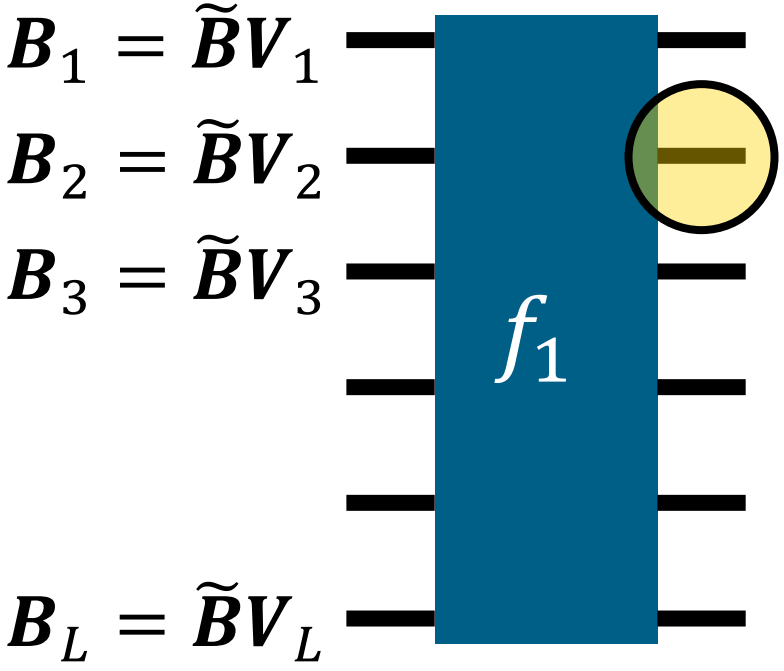
$\boldsymbol{V}_j$ is low-norm

if gate computes $x_i + x_j$: associate $\boldsymbol{B}_i + \boldsymbol{B}_j$

if gate computes $x_i x_j$: associate $\boldsymbol{B}_i \boldsymbol{G}^{-1}(\boldsymbol{B}_j)$

Computing B_f Quickly

[WWW26]



$$f_{1,k}(x_1, \ldots, x_L) = \sum_{i,j \in [L]} g_1(i,j,k) \cdot x_i x_j$$

$$\boldsymbol{B}_{1,k} = \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \boldsymbol{B}_i \boldsymbol{G}^{-1}(\boldsymbol{B}_j)$$

$$= \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \widetilde{\boldsymbol{B}}\boldsymbol{V}_i \boldsymbol{G}^{-1}(\widetilde{\boldsymbol{B}})\boldsymbol{V}_j$$

$\boldsymbol{V}_j$ is low-norm

$$= \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \widetilde{\boldsymbol{B}} \underbrace{\left(\mathrm{bits}(\widetilde{\boldsymbol{B}})^{\mathrm{T}} \otimes \boldsymbol{I}\right)\left(\boldsymbol{I} \otimes \mathrm{vec}(\boldsymbol{V}_i)\right)}_{\boldsymbol{V}_i \boldsymbol{G}^{-1}(\widetilde{\boldsymbol{B}})} \boldsymbol{V}_j$$

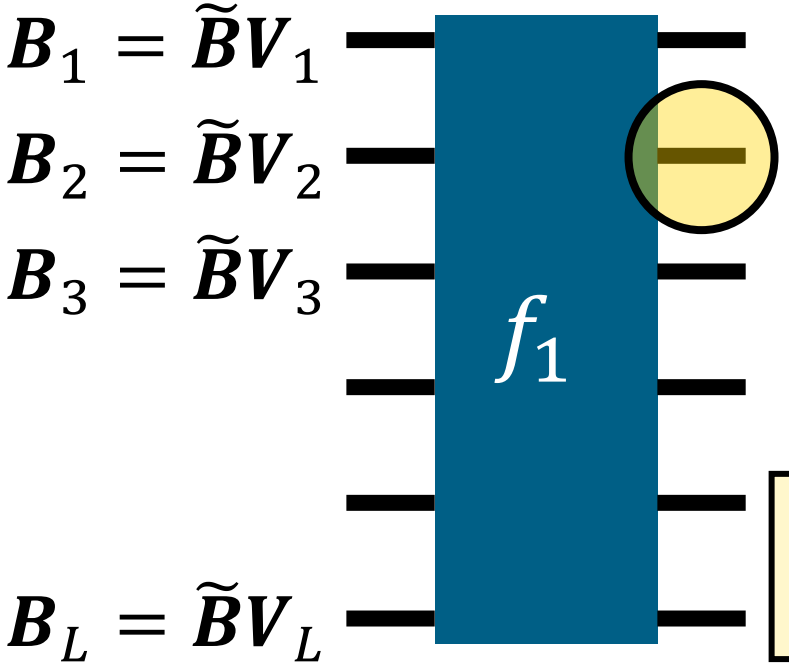
"compactification"

evaluation invariant

$$(\widetilde{\boldsymbol{B}} - \boldsymbol{C}_{\boldsymbol{x}})\boldsymbol{V}\boldsymbol{H}_k = \boldsymbol{B}_{1,k} - f_{1,k}(\boldsymbol{x})\boldsymbol{G} - \boldsymbol{A}\boldsymbol{Z}_{\boldsymbol{k}}$$

Computing B_f Quickly

[WWW26]



$$f_{1,k}(x_1, \ldots, x_L) = \sum_{i,j \in [L]} g_1(i,j,k) \cdot x_i x_j$$

$$B_{1,k} = \sum_{i,j \in [L\}} g_1(i,j,k) \cdot \tilde{B} \left(\text{bits}(\tilde{B})^{\mathrm{T}} \otimes I \right) \left(I \otimes \text{vec}(V_i) \right) V_j$$

$$\tilde{B}_1 = \sum_{k \in [L]} B_{1,k} W_k = \left(\text{bits}(\tilde{B})^{\mathrm{T}} \otimes \tilde{B} \right) \sum_{i,j,k \in [L]} g_1(i,j,k) \cdot \left(I \otimes \text{vec}(V_i) \right) V_j W_k$$

$\tilde{B}_1$ is the matrix associated with the matrix-valued function that computes a commitment to $f_1(x)$

will drop A · **low-norm**

$$\left(\tilde{B} - C_x\right) V H_k = B_{1,k} - f_{1,k}(x) G$$

Goal: $\left(\tilde{B} - C_x\right) V H = \tilde{B}_1 - C_{f_1(x)}$

$$C_{f_1(x)} = \sum_{k \in [L]} f_{1,k}(x) G W_k$$

Computing B_f Quickly

[WWW26]

$$\widetilde{\boldsymbol{B}}_1 = \left(\mathrm{bits}\left(\widetilde{\boldsymbol{B}}\right)^{\mathrm{T}} \otimes \widetilde{\boldsymbol{B}}\right) \sum_{i,j,k \in [L]} g_1(i,j,k) \cdot \left(\boldsymbol{I} \otimes \mathrm{vec}(\boldsymbol{V}_i)\right) \boldsymbol{V}_j \boldsymbol{W}_k$$

Naïvely, computing $\widetilde{\boldsymbol{B}}_1$ requires $O\left(L^3\right)$ time, which is $O\left(|f_1|^3\right)$

But turns out we can do better

Observation 1: Suppose we want to compute $\sum_{i \in \{0,1\}^\ell} \boldsymbol{M}_{1,i_1} \boldsymbol{M}_{2,i_2} \cdots \boldsymbol{M}_{\ell,i_\ell}$

There are 2^ℓ products in this sum, but can compute in $O(\ell)$ time:

$$\sum_{i \in \{0,1\}^\ell} \boldsymbol{M}_{1,i_1} \boldsymbol{M}_{2,i_2} \cdots \boldsymbol{M}_{\ell,i_\ell} = \left(\boldsymbol{M}_{1,0} + \boldsymbol{M}_{1,1}\right)\left(\boldsymbol{M}_{2,0} + \boldsymbol{M}_{2,1}\right) \cdots \left(\boldsymbol{M}_{\ell,0} + \boldsymbol{M}_{\ell,1}\right)$$

Computing B_f Quickly

[WWW26]

$$\widetilde{\boldsymbol{B}}_1 = \left(\mathrm{bits}\left(\widetilde{\boldsymbol{B}}\right)^{\mathrm{T}} \otimes \widetilde{\boldsymbol{B}}\right) \sum_{i,j,k \in [L]} g_1(i,j,k) \cdot \left(\boldsymbol{I} \otimes \mathrm{vec}(\boldsymbol{V}_i)\right) \boldsymbol{V}_j \boldsymbol{W}_k$$

Naïvely, computing $\widetilde{\boldsymbol{B}}_1$ requires $O\left(L^3\right)$ time, which is $O\left(|f_1|^3\right)$

Fact: Identify $[L]$ with $\{0,1\}^{\ell}$. Then matrices $\boldsymbol{V}_i, \boldsymbol{W}_k$ are precisely subset products

If $g_1(i,j,k)$ was a constant function, then can compute $\widetilde{\boldsymbol{B}}_1$ in time $\mathrm{poly}(\log L)$

Observation 2: Still true if $g_1(i,j,k)$ can be computed by a read-once branching program

$$g_1(i,j,k) = \boldsymbol{s}^{\mathrm{T}} \boldsymbol{M}_{1,i_1} \cdots \boldsymbol{M}_{\ell,i_\ell} \boldsymbol{M}_{\ell+1,j_1} \cdots \boldsymbol{M}_{2\ell,j_\ell} \boldsymbol{M}_{2\ell+1,k_1} \cdots \boldsymbol{M}_{3\ell,k_\ell} \boldsymbol{t}$$

Then g_1 is also a subset product so $\widetilde{\boldsymbol{B}}_1$ is essentially a “read-twice” branching program

Computing B_f Quickly

[WWW26]

$$\widetilde{\boldsymbol{B}}_1 = \left(\mathrm{bits}\left(\widetilde{\boldsymbol{B}}\right)^{\mathrm{T}} \otimes \widetilde{\boldsymbol{B}}\right) \sum_{i,j,k \in [L]} g_1(i,j,k) \cdot \left(\boldsymbol{I} \otimes \mathrm{vec}(\boldsymbol{V}_i)\right) \boldsymbol{V}_j \boldsymbol{W}_k$$

Naïvely, computing $\widetilde{\boldsymbol{B}}_1$ requires $O\left(L^3\right)$ time, which is $O\left(|f_1|^3\right)$

Fact: Identify $[L]$ with $\{0,1\}^{\ell}$. Then matrices $\boldsymbol{V}_i, \boldsymbol{W}_k$ are precisely subset products

If $g_1(i,j,k)$ was a constant function, then can compute $\widetilde{\boldsymbol{B}}_1$ in time $\mathrm{poly}(\log L)$

Observation 2: Still true if $g_1(i,j,$

$$g_1(i,j,k) = \boldsymbol{s}^{\mathrm{T}} \boldsymbol{M}_{1,i_1} \cdots$$

Can be transformed back into a "read-once" branching program via tensoring

Then g_1 is also a subset product so $\widetilde{\boldsymbol{B}}_1$ is essentially a "read-twice" branching program

ABE for Uniform Computations

[WWW26]

"dual Regev public key" attribute-encoding matrix

Public key: $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}, \quad \boldsymbol{p} \in \mathbb{Z}_q^n, \quad \boldsymbol{B} \in \mathbb{Z}_q^{n \times Lm} \longrightarrow \widetilde{\boldsymbol{B}} \in \mathbb{Z}_q^{n \times m}$

public parameters independent of attribute length!

Secret key for f: low-norm vector $\boldsymbol{v}_f \in \mathbb{Z}^{2m}$ where $\left[\boldsymbol{A} \mid \boldsymbol{B}_f\right]\boldsymbol{v}_f = \boldsymbol{p}$

Take-away: When f is a layered Boolean circuit (depth D, width L) whose wiring functions can be computed by read-once branching programs, $\boldsymbol{B}_f$ can be computed $D \cdot \mathrm{poly}(\log L)$ time

Any log-space uniform f satisfies this property

(adapting Turing machine simulation results from [GKR08, GoI18])

ABE for Uniform Computations

[WWW26]

Uniform evaluations on encoded inputs:

$$\left[\boldsymbol{A} \mid \widetilde{\boldsymbol{B}} - \boldsymbol{C}_{\boldsymbol{x}}\right] \cdot \boldsymbol{H}_{\widetilde{\boldsymbol{B}},f,\boldsymbol{x}} = \widetilde{\boldsymbol{B}}_f - f(x) \cdot \boldsymbol{G}$$

When f is log-space uniform with depth D and size S, we can compute $\widetilde{\boldsymbol{B}}_f$ in time $D \cdot \mathrm{poly}(\log S)$

Give ABE for log-space uniform NC circuits from decomposed LWE

- Supports arbitrary-length attributes (up to 2^{λ})
- Key-generation runs in time $\mathrm{poly}(\lambda)$ and one key works for all input lengths

Log-space uniform NC includes NFA and NL as special subclasses

ABE for Uniform Computations

[WWW26]

Previous lattice-based approaches (that do not go via obfuscation):

- [HLL24, AKY24, CW25]: ABE for TMs (or L/NL) where encryption algorithm depends on running time of Turing machine; security relies on (circular) evasive LWE
- [AMY19a]: secret-key ABE for NFA from LWE
- [AMR26]: ABE for bounded-space TM from decomposed LWE
- **Our work [WWW26]:** log-space uniform NC circuits from decomposed LWE (encryption only depends on the depth bound and key-generation runs in time $\mathrm{poly}(\lambda)$)

Give ABE for log-space uniform NC circuits from decomposed LWE

Supports arbitrary-length attributes (up to 2^{λ})

Key-generation runs in time $\mathrm{poly}(\lambda)$ and one key works for all input lengths

Log-space uniform NC includes NFA and NL as special subclasses

Uniform Computations on Encoded Inputs

[WWW26]

Uniform evaluations on encoded inputs:

$$\left[\boldsymbol{A} \mid \widetilde{\boldsymbol{B}} - \boldsymbol{C}_{\boldsymbol{x}}\right] \cdot \boldsymbol{H}_{\widetilde{\boldsymbol{B}},f,\boldsymbol{x}} = \widetilde{\boldsymbol{B}}_f - f(x) \cdot \boldsymbol{G}$$

When f is log-space uniform with depth D and size S, we can compute $\widetilde{\boldsymbol{B}}_f$ in time $D \cdot \mathrm{poly}(\log S)$

Can also obtain a functional commitment / SNARG for bounded-depth log-space uniform policies

- Yields a SNARG for bounded-depth log-space uniform circuits from plain SIS
- Can view our computation as an algebraic version of classic GKR protocol

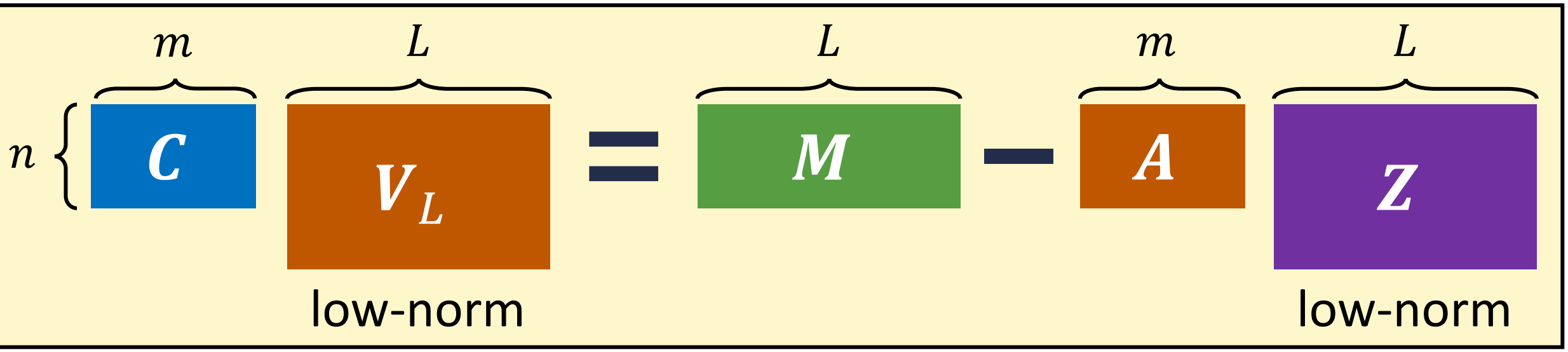
A Useful Abstraction: Matrix Commitments

[Wee24, Wee25]

Succinct commitment to a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times L}$

$\mathrm{Commit}(\mathrm{pp}, \boldsymbol{M}) \to \boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$ $\mathrm{Open}(\mathrm{pp}, \boldsymbol{M}) \to \boldsymbol{Z} \in \mathbb{Z}_q^{m \times L}$

deterministic algorithms



Security property: $(\mathrm{pp}, \boldsymbol{s}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{e}^{\mathrm{T}}) \approx (\mathrm{pp}, \boldsymbol{u}^{\mathrm{T}})$

LWE holds with respect to $\boldsymbol{A}$ *given* pp

Matrix Commitments and Their Applications

Succinct commitment to a matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times L}$

$\text{Commit}(\text{pp}, \boldsymbol{M}) \to \boldsymbol{C} \in \mathbb{Z}_q^{n \times m}$

$\text{Open}(\text{pp}, \boldsymbol{M}) \to \boldsymbol{Z_M} \in \mathbb{Z}^{m \times L}$

$\text{pp} = \left(\boldsymbol{A} \in \mathbb{Z}^{n \times m}, \boldsymbol{V}_L \in \mathbb{Z}^{m \times L}\right)$

$\boldsymbol{C} \cdot \boldsymbol{V}_L = \boldsymbol{M} - \boldsymbol{A} \cdot \boldsymbol{Z_M}$

Functional commitments for all circuits + SNARGs for P/poly	[WW23a, WW23b, Wee24, Wee25]
Optimal broadcast encryption	[Wee25, AMR26, GY26a]
Distributed broadcast encryption	[CW24, CHW25, WW25, AMR26, CW26, GY26a, GY26b]
Secure group messaging	[BBDGW26]
Silent threshold encryption	[GY26b, GY26c, CWY26]
Nearly-optimal KP-ABE and CP-ABE for circuits	[Wee24, Wee25]
ABE for bounded-space Turing machines (and other uniform models)	[AMR26, WWW26]
Registered ABE for circuits	[CHW25, AMR25, WW25]
Succinct computational secret sharing for DNFs	[CW26, GY26c]
Complexity separations: $\text{P} \neq \text{NC}$ and $\text{P} \neq \text{L}$	[AMR26b]