

LOCKABLE OBFUSCATION

VBB OBFUSCATION FOR
COMPUTE-AND-COMPARE PROGRAMS

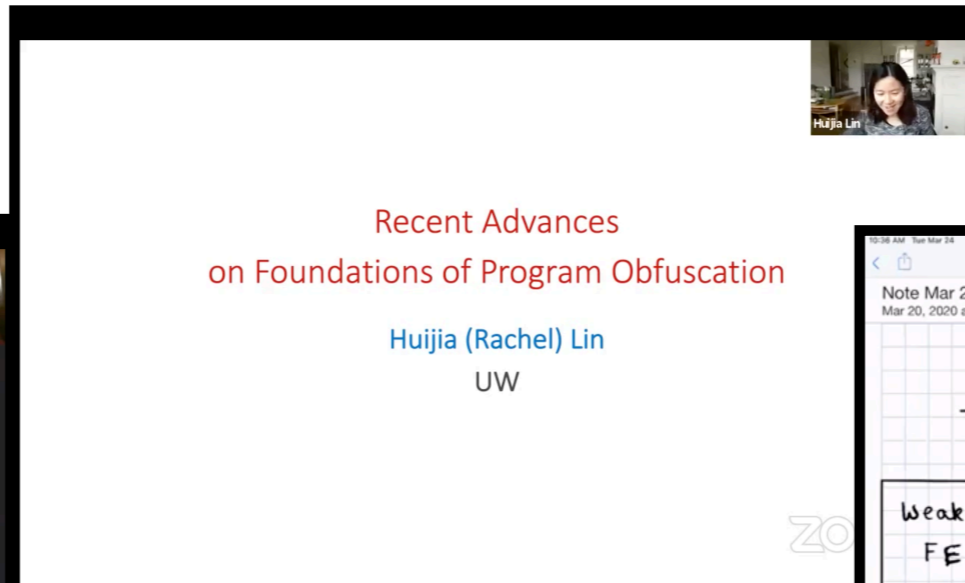
VENKATA KOPPULA

(Talk based on concurrent works: [Goyal, K, Waters], [Wichs, Zirdelis])

CODE OBFUSCATION: COMPILING CODE TO HIDE THE IMPLEMENTATION

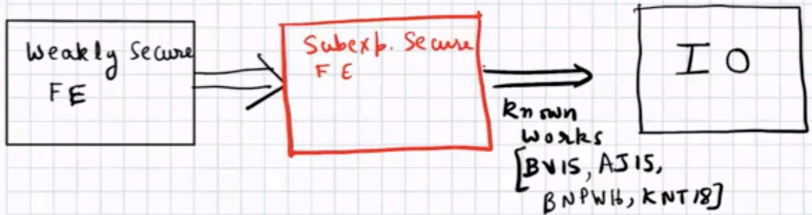
Recent Advances
on Foundations of Program Obfuscation

Huijia (Rachel) Lin
UW



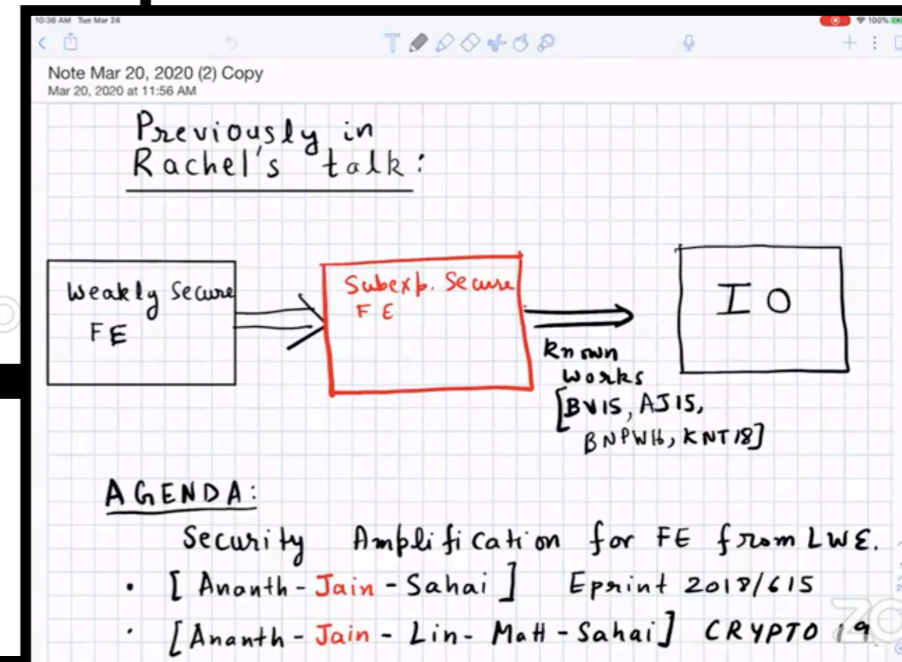
Note Mar 20, 2020 (2) Copy
Mar 20, 2020 at 11:56 AM

Previously in Rachel's talk:



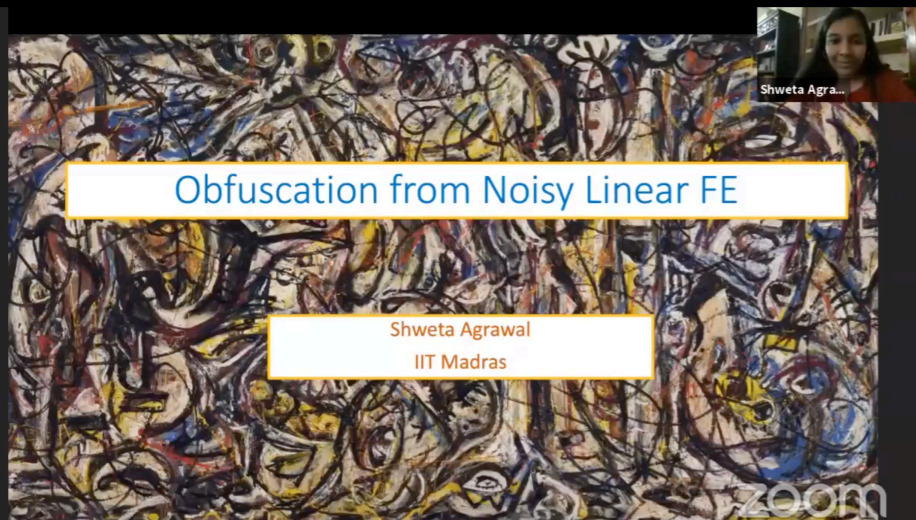
AGENDA:

- Security Amplification for FE from LWE.
- [Ananth-Jain-Sahai] Eprint 2018/615
- [Ananth-Jain-Lin-Math-Sahai] CRYPTO 19




Obfuscation from Noisy Linear FE

Shweta Agrawal
IIT Madras



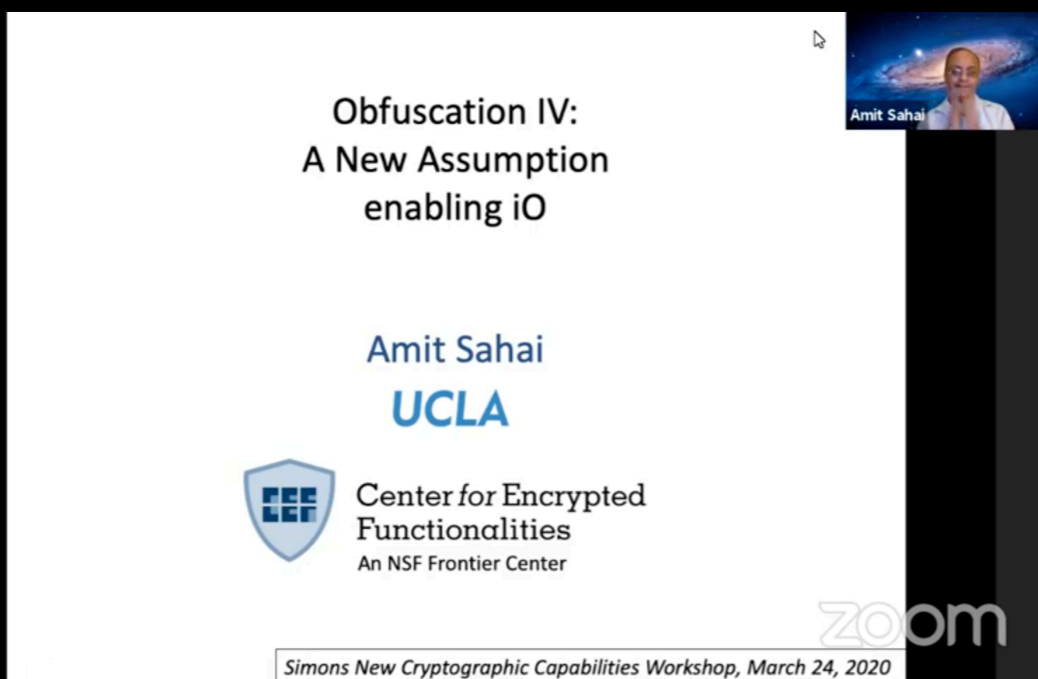
Obfuscation IV:
A New Assumption
enabling iO

Amit Sahai
UCLA



Center for Encrypted Functionalities
An NSF Frontier Center

Simons New Cryptographic Capabilities Workshop, March 24, 2020



Cryptanalysis of Candidate Program Obfuscators

Yilei Chen
[Visa Research]

2020 Simons Lattice Program

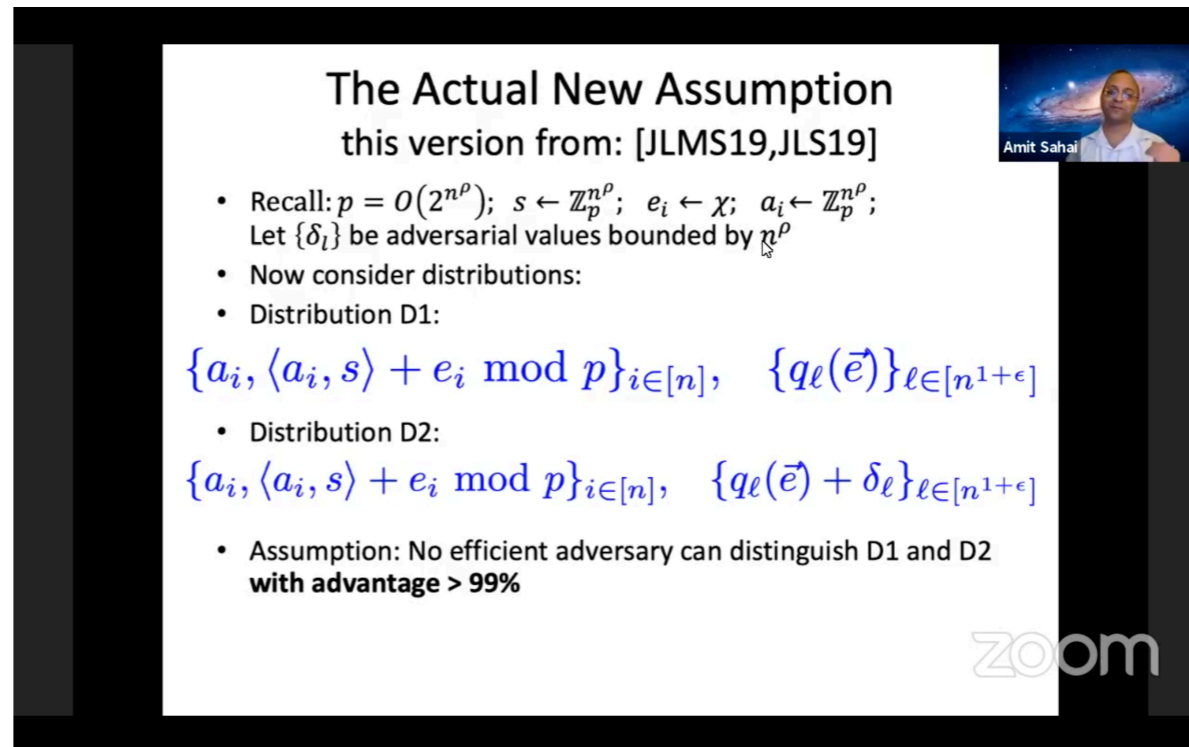
CODE OBFUSCATION: COMPILING CODE TO HIDE THE IMPLEMENTATION

GGH13 + GGHRSW



Timeline not drawn to scale

CODE OBFUSCATION: COMPILING CODE TO HIDE THE IMPLEMENTATION



The Actual New Assumption
this version from: [JLMS19, JLS19]

- Recall: $p = O(2^{n^\rho})$; $s \leftarrow \mathbb{Z}_p^{n^\rho}$; $e_i \leftarrow \chi$; $a_i \leftarrow \mathbb{Z}_p^{n^\rho}$;
Let $\{\delta_l\}$ be adversarial values bounded by n^ρ
- Now consider distributions:
- Distribution D1:
 $\{a_i, \langle a_i, s \rangle + e_i \bmod p\}_{i \in [n]}, \{q_\ell(\vec{e})\}_{\ell \in [n^{1+\epsilon}]}$
- Distribution D2:
 $\{a_i, \langle a_i, s \rangle + e_i \bmod p\}_{i \in [n]}, \{q_\ell(\vec{e}) + \delta_\ell\}_{\ell \in [n^{1+\epsilon}]}$
- Assumption: No efficient adversary can distinguish D1 and D2
with advantage $> 99\%$

zoom

GGH13 + GGHRSW

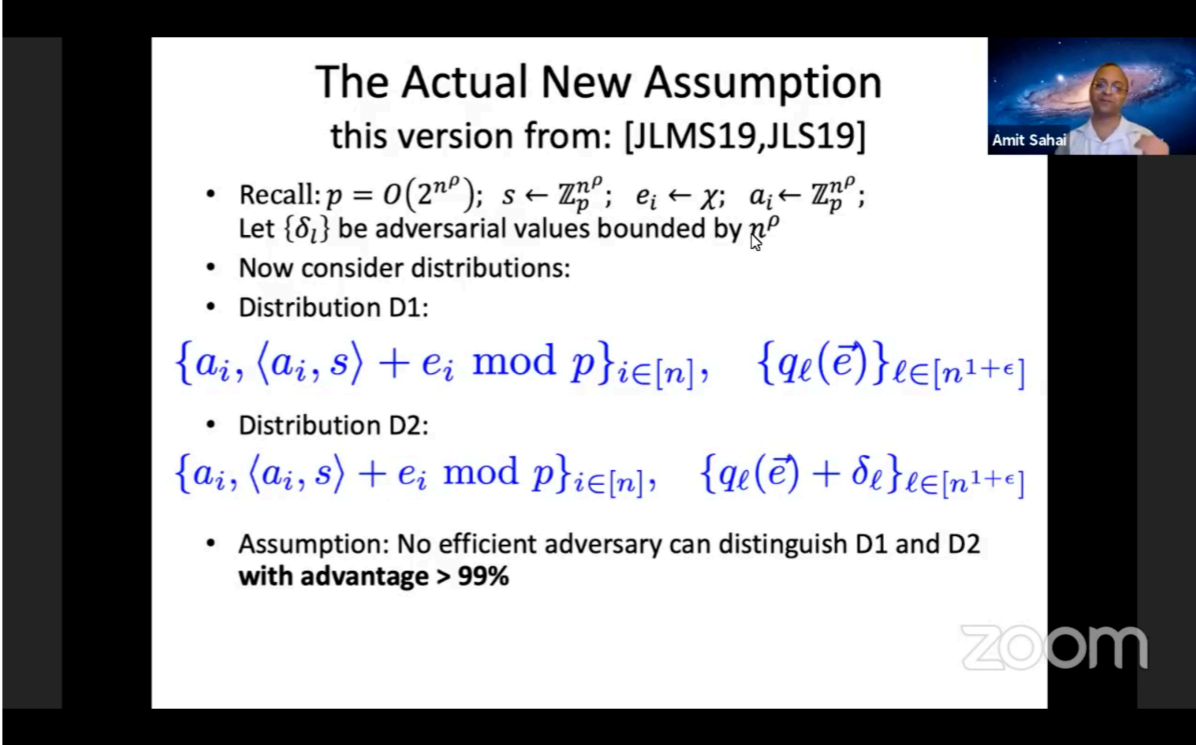


CODE OBFUSCATION: COMPILING CODE TO HIDE THE IMPLEMENTATION

GGH13 + GGHRSW

The Actual New Assumption
this version from: [JLMS19,JLS19]

- Recall: $p = O(2^{n^\rho})$; $s \leftarrow \mathbb{Z}_p^{n^\rho}$; $e_i \leftarrow \chi$; $a_i \leftarrow \mathbb{Z}_p^{n^\rho}$;
Let $\{\delta_l\}$ be adversarial values bounded by n^ρ
- Now consider distributions:
- Distribution D1:
 $\{a_i, \langle a_i, s \rangle + e_i \bmod p\}_{i \in [n]}, \{q_\ell(\vec{e})\}_{\ell \in [n^{1+\epsilon}]}$
- Distribution D2:
 $\{a_i, \langle a_i, s \rangle + e_i \bmod p\}_{i \in [n]}, \{q_\ell(\vec{e}) + \delta_\ell\}_{\ell \in [n^{1+\epsilon}]}$
- Assumption: No efficient adversary can distinguish D1 and D2
with advantage $> 99\%$



IO from
standard assumptions

Timeline not drawn to scale

CODE OBFUSCATION: COMPILING CODE TO HIDE THE IMPLEMENTATION

What functions can be obfuscated using standard assumptions (e.g. LWE) ?

CODE OBFUSCATION: COMPILING CODE TO HIDE THE IMPLEMENTATION

What functions can be obfuscated using standard assumptions (e.g. LWE) ?

Prior Works: VBB Obfuscation for simple function classes.

- Point Functions [C97, ...]
- Hyperplanes [CRV09]
- Conjunctions [BVWW16, ...]

CODE OBFUSCATION: COMPILING CODE TO HIDE THE IMPLEMENTATION

What functions can be obfuscated using standard assumptions (e.g. LWE) ?

Prior Works: VBB Obfuscation for simple function classes.

- Point Functions [C97, ...]
- Hyperplanes [CRV09]
- Conjunctions [BVWW16, ...]
- **Today: Compute-and-compare programs**

THE FUNCTION CLASS: COMPUTE-AND-COMPARE PROGRAMS

THE FUNCTION CLASS: COMPUTE-AND-COMPARE PROGRAMS

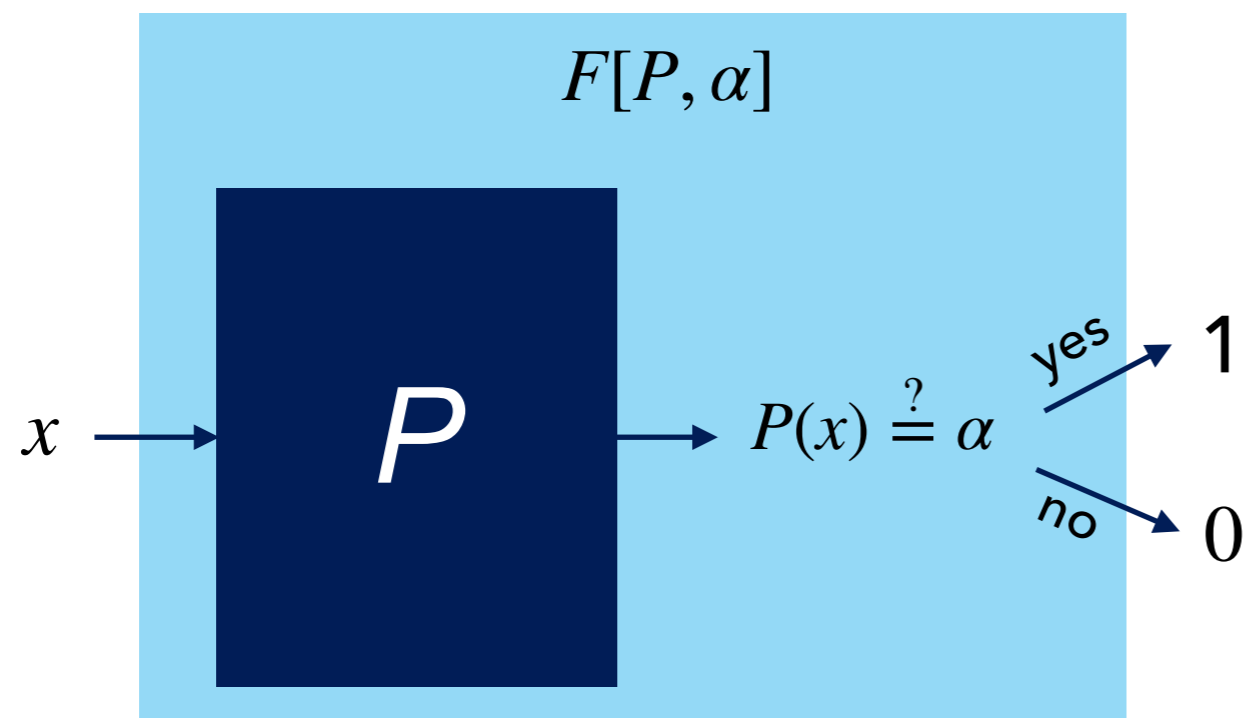
Every function parameterized by
program P and string α



$F[P, \alpha]$

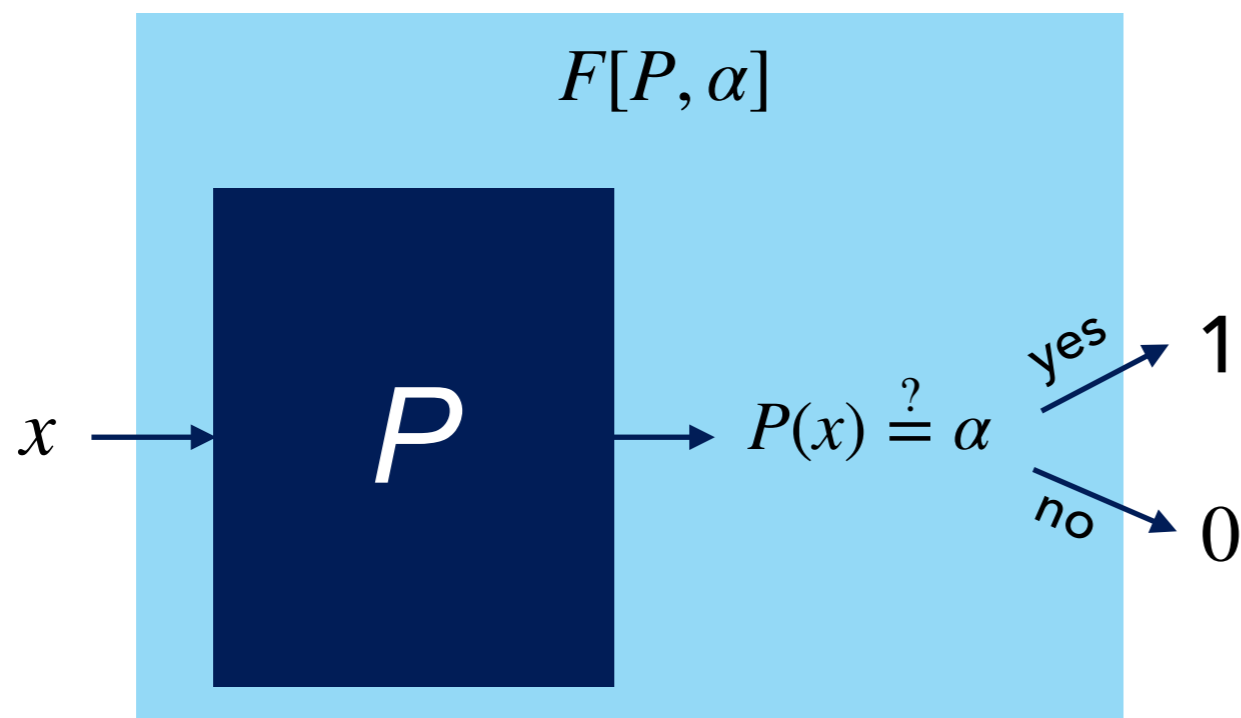
THE FUNCTION CLASS: COMPUTE-AND-COMPARE PROGRAMS

Every function parameterized by
program P and string α



THE FUNCTION CLASS: COMPUTE-AND-COMPARE PROGRAMS

Every function parameterized by program P and string α



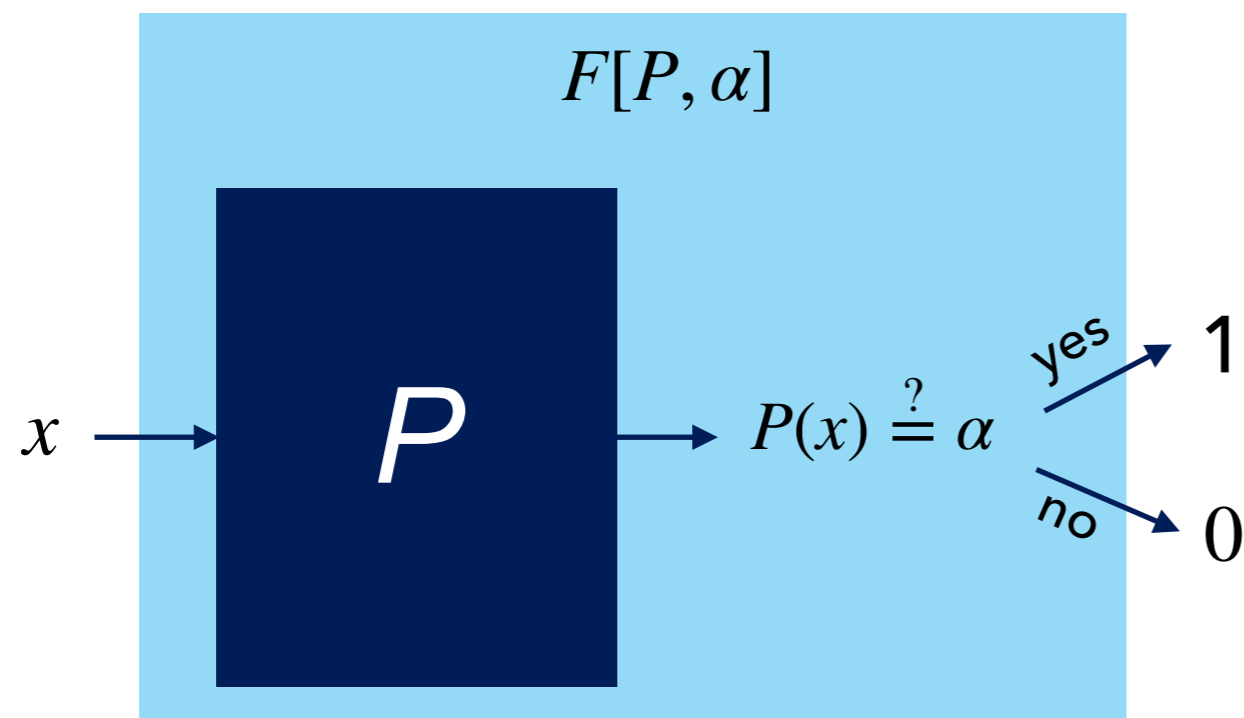
POINT FUNCTIONS

$$\text{PF}[\alpha](x) = \begin{cases} 1 & \text{if } x = \alpha \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PF}[\alpha] \equiv F[\text{Id}, \alpha]$$

THE FUNCTION CLASS: COMPUTE-AND-COMPARE PROGRAMS

Every function parameterized by program P and string α



POINT FUNCTIONS

$$\text{PF}[\alpha](x) = \begin{cases} 1 & \text{if } x = \alpha \\ 0 & \text{otherwise} \end{cases}$$

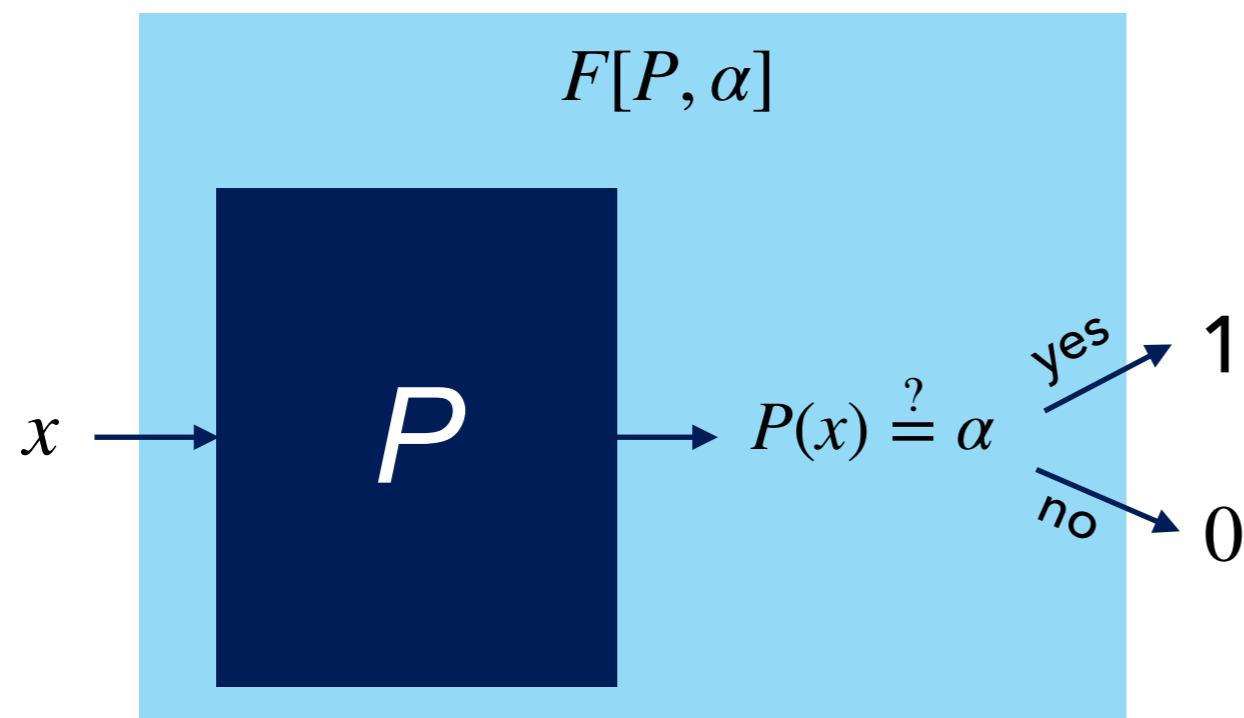
$$\text{PF}[\alpha] \equiv F[\text{Id}, \alpha]$$

CONJUNCTIONS

$$f(x_1, \dots, x_n) = x_1 \wedge x_5 \wedge \bar{x}_7$$

THE FUNCTION CLASS: COMPUTE-AND-COMPARE PROGRAMS

Every function parameterized by program P and string α



POINT FUNCTIONS

$$\text{PF}[\alpha](x) = \begin{cases} 1 & \text{if } x = \alpha \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PF}[\alpha] \equiv F[\text{Id}, \alpha]$$

CONJUNCTIONS

$$f(x_1, \dots, x_n) = x_1 \wedge x_5 \wedge \bar{x}_7$$

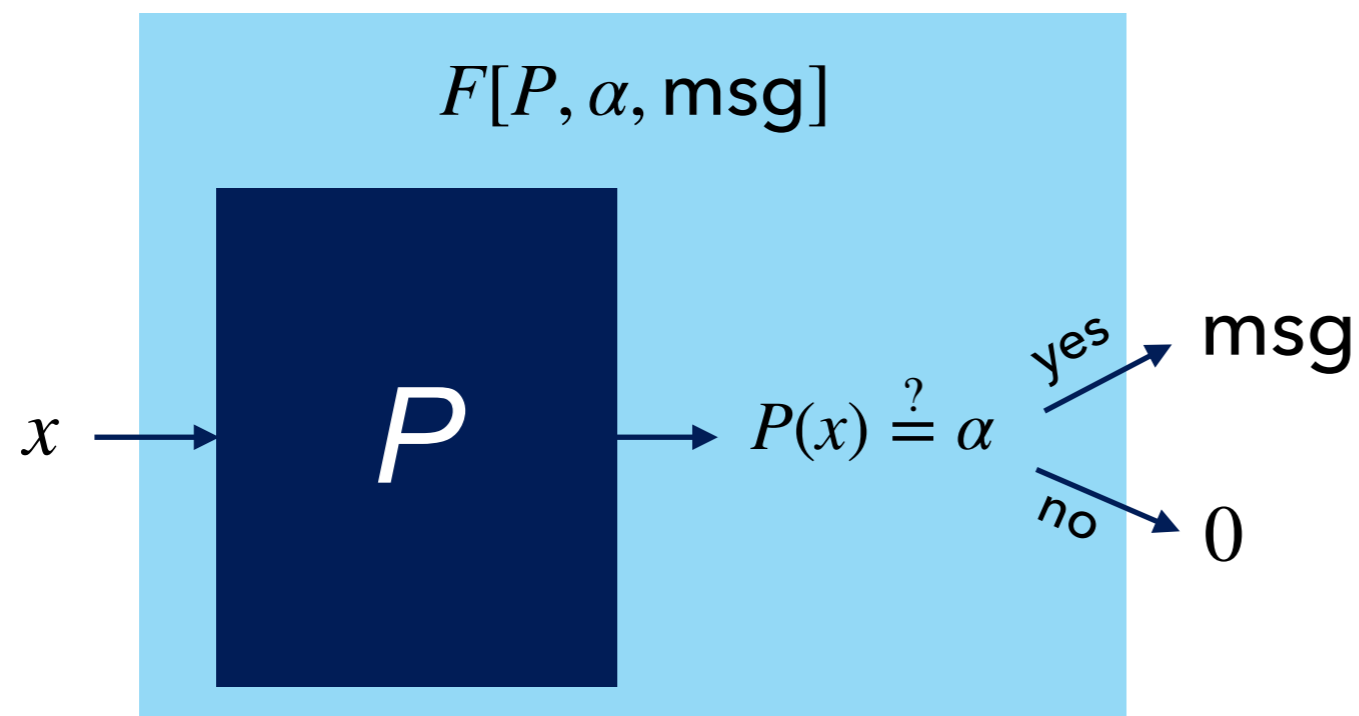
$$P(x_1, \dots, x_n) = (x_1, x_5, x_7)$$

$$\alpha = 110$$

$$f \equiv F[P, \alpha]$$

THE FUNCTION CLASS: COMPUTE-AND-COMPARE PROGRAMS

Every function parameterized by program P and strings α, msg



SECURITY: (AVERAGE) VBB OBFUSCATION

SECURITY: (AVERAGE) VBB OBFUSCATION

SECURITY

SECURITY: (AVERAGE) VBB OBFUSCATION

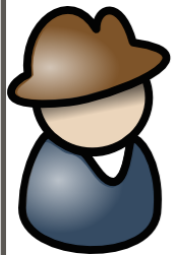
SECURITY

For randomly chosen α
 $\text{Obf}[P, \alpha, \text{msg}]$ hides P, msg

SECURITY: (AVERAGE) VBB OBFUSCATION

SECURITY

For randomly chosen α
 $\text{Obf}[P, \alpha, \text{msg}]$ hides P, msg



P, msg



α : random string

$\text{Obf}[P, \alpha, \text{msg}]$



$\text{Obf}[\#, \#, \#]$

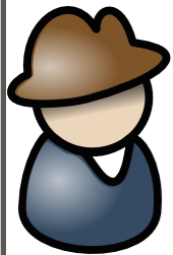
Guess



SECURITY: (AVERAGE) VBB OBFUSCATION

SECURITY

For randomly chosen α ^{lock}
 $\text{Obf}[P, \alpha, \text{msg}]$ hides P, msg



P, msg



α : random string

$\text{Obf}[P, \alpha, \text{msg}]$



$\text{Obf}[\#, \#, \#]$

Guess



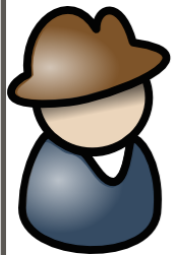
Lets call it
'lockable obfuscation'



LOCKABLE OBFUSCATION (LO)

SECURITY

For randomly chosen α ^{lock}
 $\text{Obf}[P, \alpha, \text{msg}]$ hides P, msg



P, msg



α : random string

$\text{Obf}[P, \alpha, \text{msg}]$



$\text{Obf}[\#, \#, \#]$

Guess

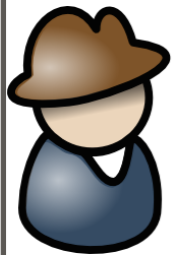


LOCKABLE OBFUSCATION (LO)

SECURITY

For randomly chosen α ^{lock}
 $\text{Obf}[P, \alpha, \text{msg}]$ hides P, msg

lock must be long enough



P, msg



α : random string

$\text{Obf}[P, \alpha, \text{msg}]$



$\text{Obf}[\#, \#, \#]$

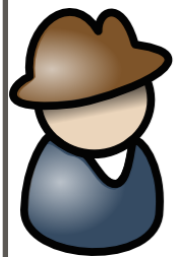
Guess



LOCKABLE OBFUSCATION (LO)

SECURITY

For randomly chosen α ^{lock}
 $\text{Obf}[P, \alpha, \text{msg}]$ hides P, msg



P, msg



α : random string

$\text{Obf}[P, \alpha, \text{msg}]$



$\text{Obf}[\#, \#, \#]$

Guess



lock must be long enough

Single bit lock?



$P = \text{all accepting prog.}$



If $\alpha = 1$, adversary can distinguish

LOCKABLE OBFUSCATION



LEARNING WITH ERRORS

Upgrading security

- Making encryption schemes anonymous
- Witness enc. \rightarrow IO for rejecting programs
- Making secure sketches private

LOCKABLE OBFUSCATION

LEARNING WITH ERRORS



Upgrading security

- Making encryption schemes anonymous
- Witness enc. \rightarrow IO for rejecting programs
- Making secure sketches private

Replacing IO with LO

- Circular security separations
- Random oracle uninstantiability results

LOCKABLE OBFUSCATION

LEARNING WITH ERRORS



LOCKABLE OBFUSCATION: APPLICATIONS

Anonymous encryption schemes
[Bellare, Boldyreva, Desai, Pointcheval 01]

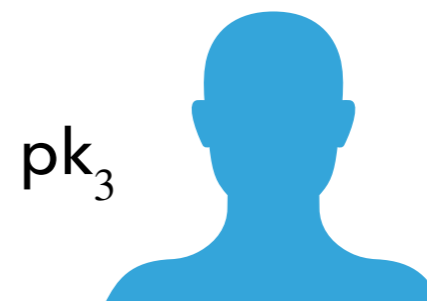
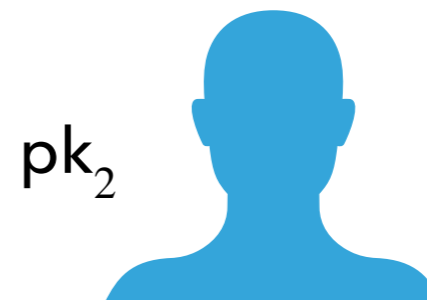
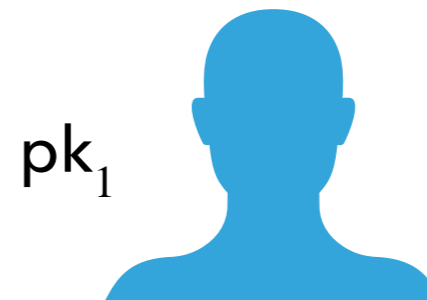
LOCKABLE OBFUSCATION: APPLICATIONS

Anonymous encryption schemes
[Bellare, Boldyreva, Desai, Pointcheval 01]



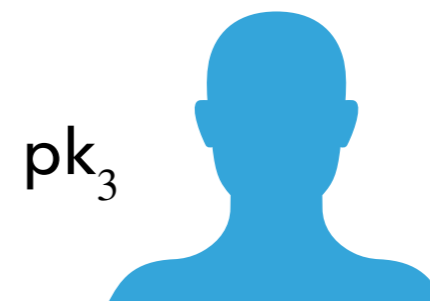
LOCKABLE OBFUSCATION: APPLICATIONS

Anonymous encryption schemes
[Bellare, Boldyreva, Desai, Pointcheval 01]



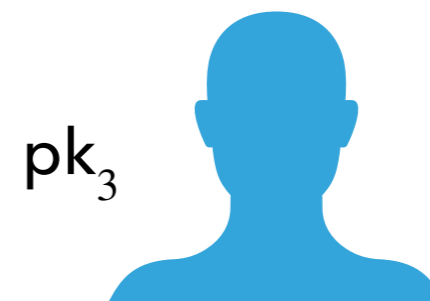
LOCKABLE OBFUSCATION: APPLICATIONS

Anonymous encryption schemes
[Bellare, Boldyreva, Desai, Pointcheval 01]



LOCKABLE OBFUSCATION: APPLICATIONS

Anonymous encryption schemes
[Bellare, Boldyreva, Desai, Pointcheval 01]



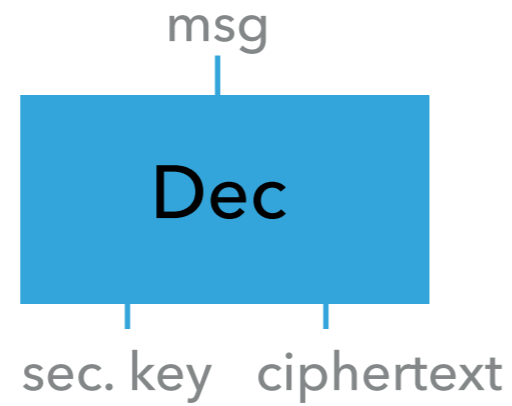
Does adversary learn this message is for pk_3 ?

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$

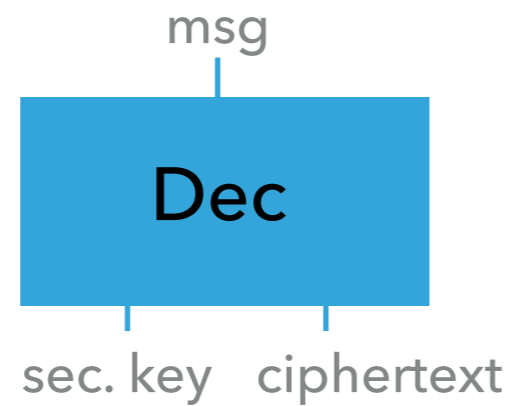
ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$



ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

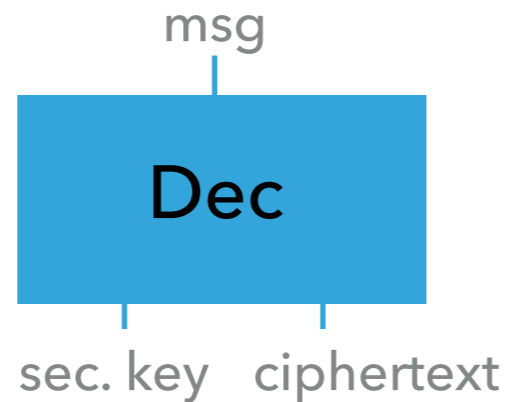
$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$



$\text{Enc-anon}(m, pk)$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$

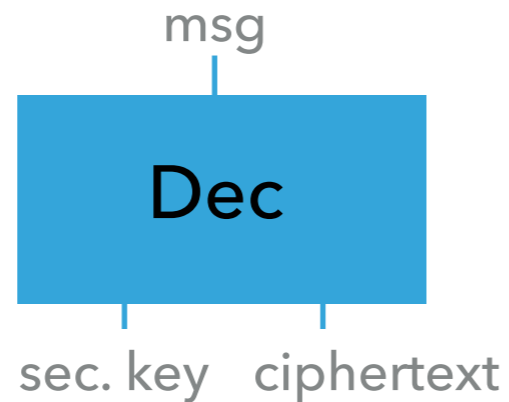


$\text{Enc-anon}(m, pk)$

Choose random string α . $ct = \text{Enc}(\alpha, pk)$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$



$\text{Enc-anon}(m, pk)$

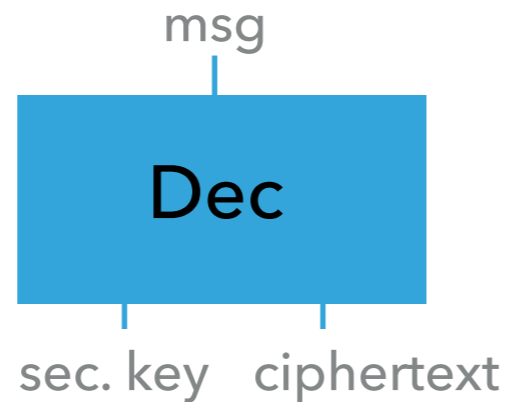
Choose random string α . $ct = \text{Enc}(\alpha, pk)$

P :

A diagram showing a blue rectangular box labeled "Dec(ct, .)". A vertical line points down from the top of the box, and another vertical line points up from the bottom of the box.

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$



$\text{Enc-anon}(m, pk)$

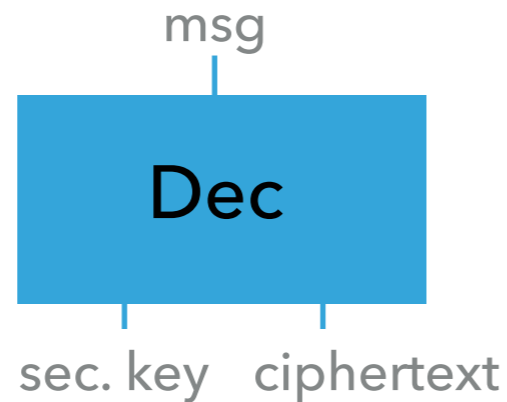
Choose random string α . $ct = \text{Enc}(\alpha, pk)$

P : $\text{Dec}(ct, .)$

Output $\text{Obf}[P, \alpha, m]$ as anon. ciphertext

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$



$\text{Enc-anon}(m, pk)$

Choose random string α . $ct = \text{Enc}(\alpha, pk)$

P : $\text{Dec}(ct, .)$

Output $\text{Obf}[P, \alpha, m]$ as anon. ciphertext

Dec-anon using sk

Run program with input = sk

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$

$\text{Enc-anon}(m, \text{pk})$

Choose random string α . $\text{ct} = \text{Enc}(\alpha, \text{pk})$

P:

$\text{Dec}(\text{ct}, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$

$\text{Enc-anon}(m, pk)$

Choose random string α . $ct = \text{Enc}(\alpha, pk)$

P:

$\text{Dec}(ct, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$

Why decryption works

P:

$\text{Dec}(ct, .)$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$

$\text{Enc-anon}(m, pk)$

Choose random string α . $ct = \text{Enc}(\alpha, pk)$

P: $\text{Dec}(ct, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$

Why decryption works

P: $\text{Dec}(ct, .)$

$P(sk) = \alpha$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

$(\text{Enc}, \text{Dec}) \rightarrow (\text{Enc-anon}, \text{Dec-anon})$

$\text{Enc-anon}(m, \text{pk})$

Choose random string α . $\text{ct} = \text{Enc}(\alpha, \text{pk})$

P: $\text{Dec}(\text{ct}, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$

Why decryption works

P: $\text{Dec}(\text{ct}, .)$

$P(\text{sk}) = \alpha$

$\text{Obf}[P, \alpha, m](\text{sk}) = m$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

Security proof: sketch

$\text{Enc-anon}(m, pk)$

$ct = \text{Enc}(\alpha, pk)$

P :

$\text{Dec}(ct, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

Security proof: sketch



Can guess pk from
 $\text{Obf}(P, \alpha, m)$

$\text{Enc-anon}(m, pk)$

$ct = \text{Enc}(\alpha, pk)$

P :

$\text{Dec}(ct, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

Security proof: sketch



Can guess pk from
 $\text{Obf}(P, \alpha, m)$

$\text{Enc-anon}(m, pk)$

$ct = \text{Enc}(\alpha, pk)$

P :

$\text{Dec}(ct, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$

$\text{Enc-anon}'(m, pk)$

$ct' = \text{Enc}(0, pk)$

P' :

$\text{Dec}(ct', .)$

Anon. ct': $\text{Obf}[P', \alpha, m]$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

Security proof: sketch



Can guess pk from
 $\text{Obf}(P, \alpha, m)$

$\text{Enc-anon}(m, pk)$

$ct = \text{Enc}(\alpha, pk)$

P :

$\text{Dec}(ct, .)$

Anon. ct: $\text{Obf}[P, \alpha, m]$



Security of Enc

$\text{Enc-anon}'(m, pk)$

$ct' = \text{Enc}(0, pk)$

P' :

$\text{Dec}(ct', .)$

Anon. ct': $\text{Obf}[P', \alpha, m]$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

Security proof: sketch



Can guess pk from
 $\text{Obf}(P', \alpha, m)$

$\text{Enc-anon}'(m, pk)$

$ct' = \text{Enc}(0, pk)$

P' :

$\text{Dec}(ct', .)$

Anon. ct' : $\text{Obf}[P', \alpha, m]$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

Security proof: sketch



Can guess pk from
 $\text{Obf}(P', \alpha, m)$

$\text{Enc-anon}'(m, pk)$

$ct' = \text{Enc}(0, pk)$

P' :

$\text{Dec}(ct', .)$

Anon. ct' : $\text{Obf}[P', \alpha, m]$



Security of LO

$\text{Enc-anon}''(m, pk)$

Anon. ct'' : $\text{Obf}[\#, \#, \#]$

ANY ENCRYPTION SCHEME CAN BE MADE ANONYMOUS USING LO.

Security proof: sketch



Can guess pk from
 $\text{Obf}(\#, \#, \#)$



$\text{Enc-anon}^{\text{LO}}(m, pk)$

Anon. ct^{LO}: $\text{Obf}[\#, \#, \#]$

EVEN ADVANCED ENCRYPTION SCHEMES CAN BE MADE ANONYMOUS USING LO.

Broadcast Encryption

[Fiat, Naor 94]

Attribute Based Encryption

[Sahai, Waters 05]

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]

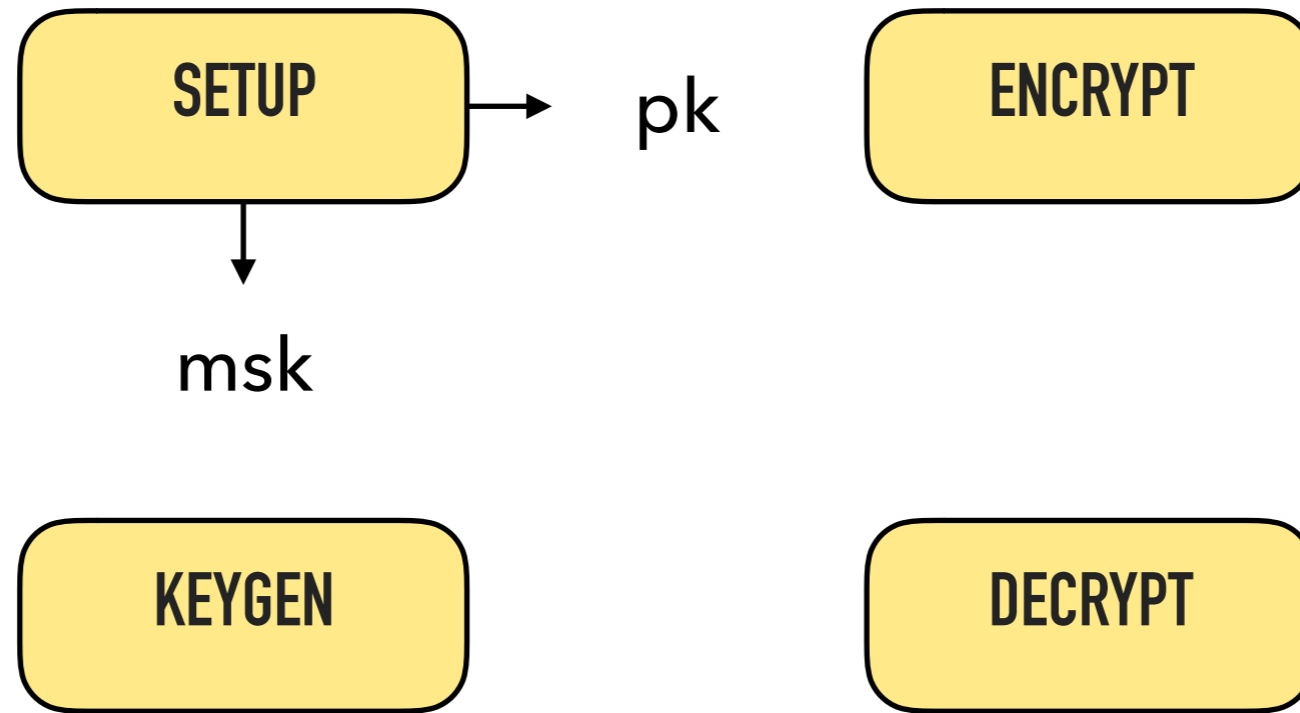
SETUP

ENCRYPT

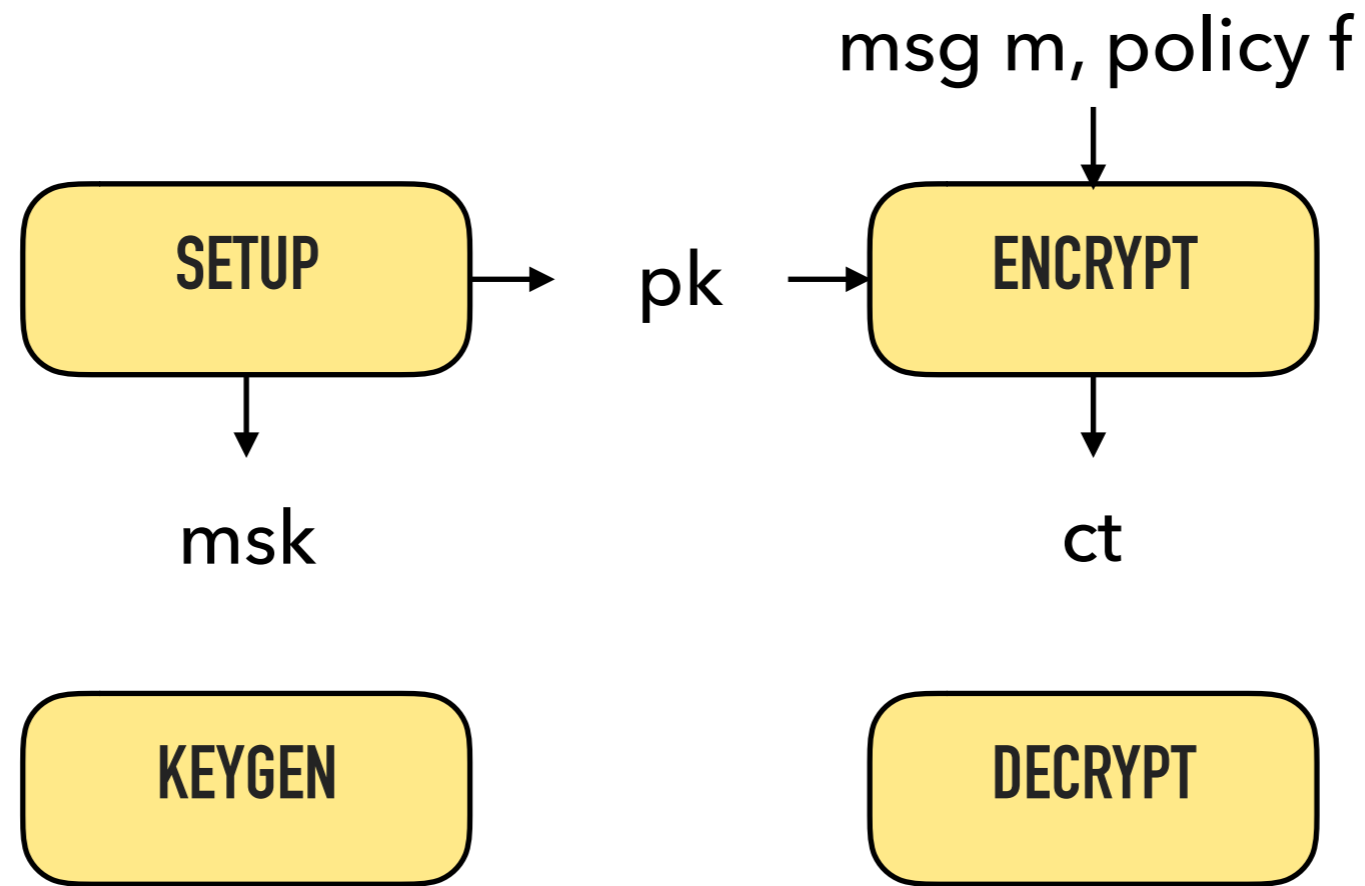
KEYGEN

DECRYPT

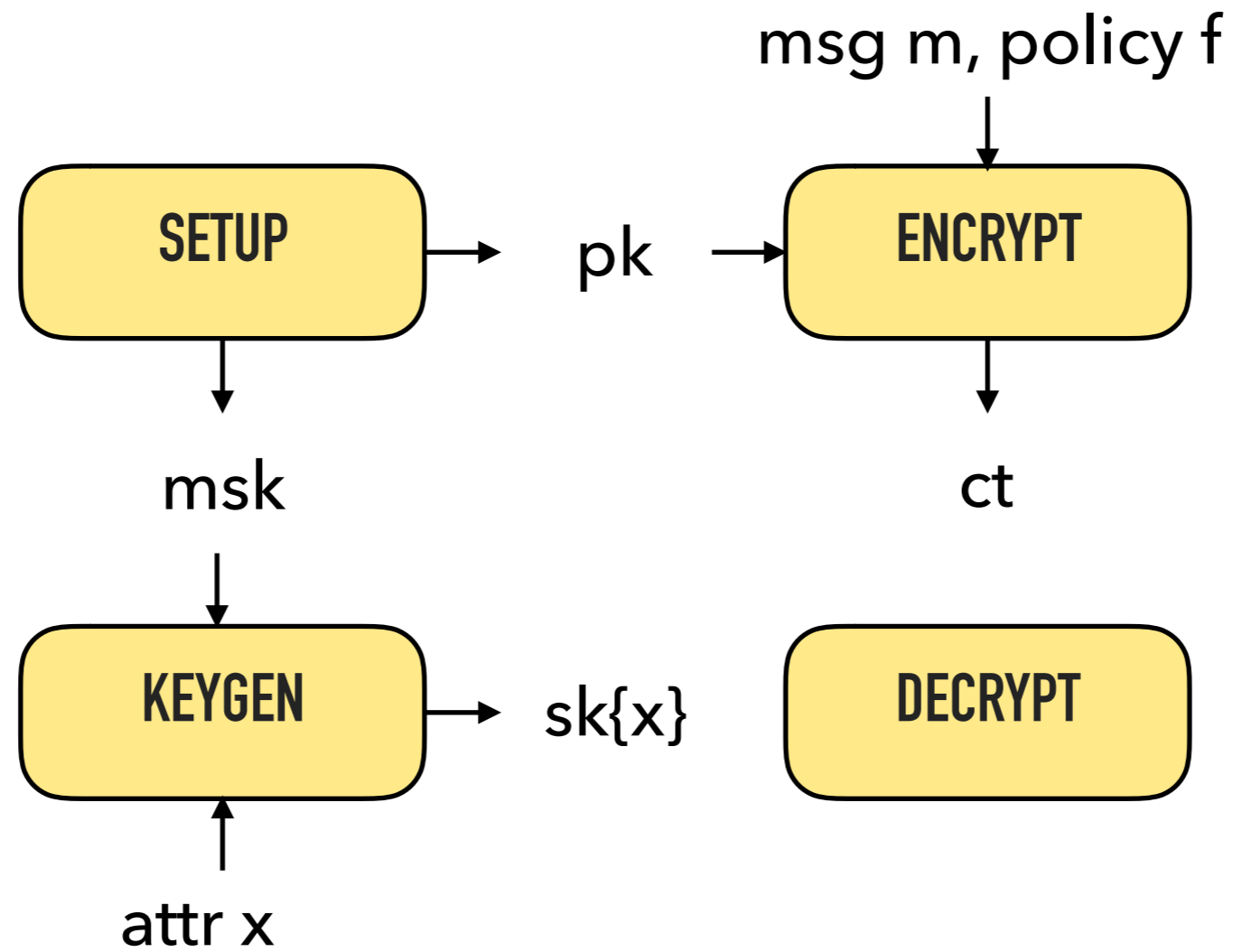
ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



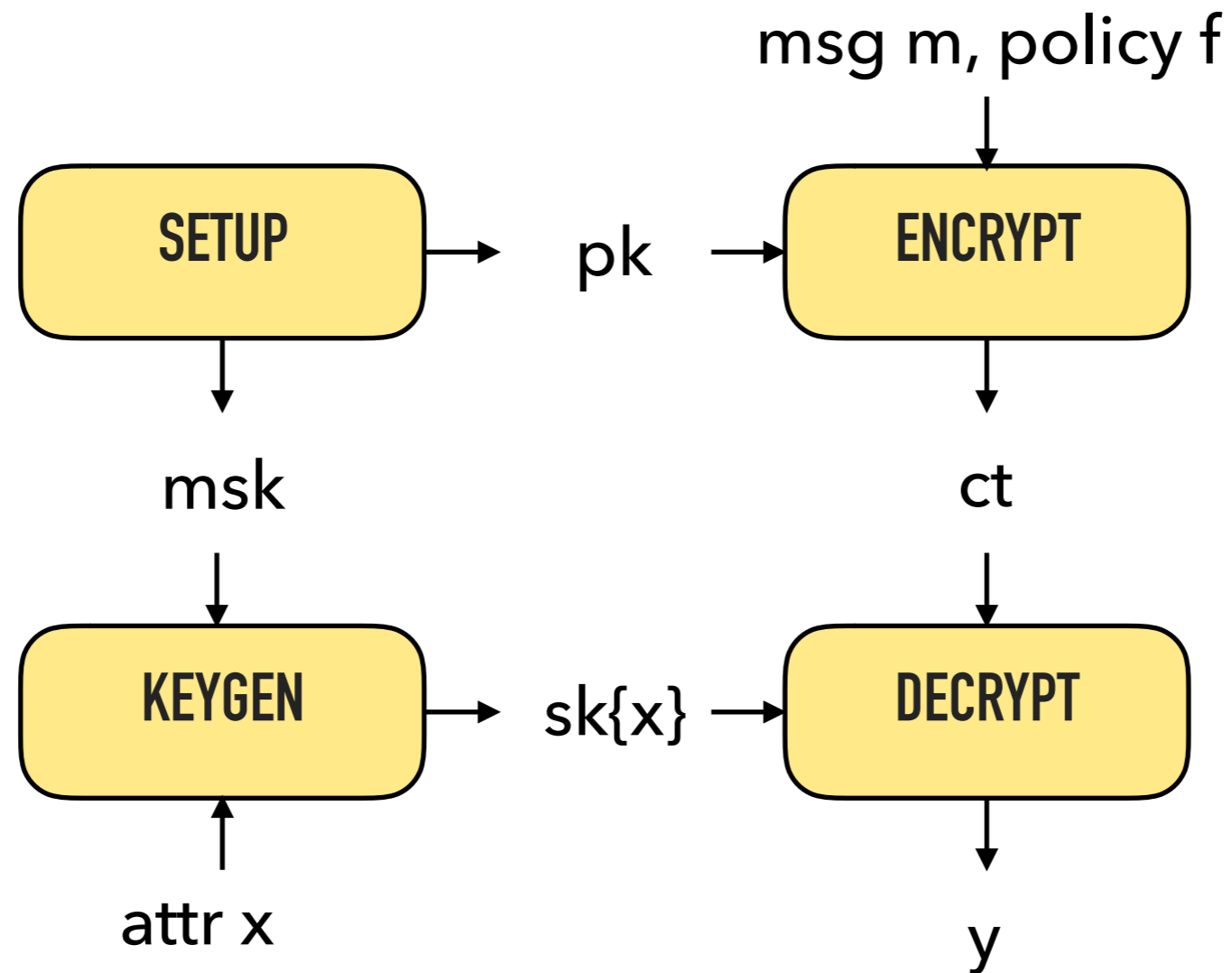
ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



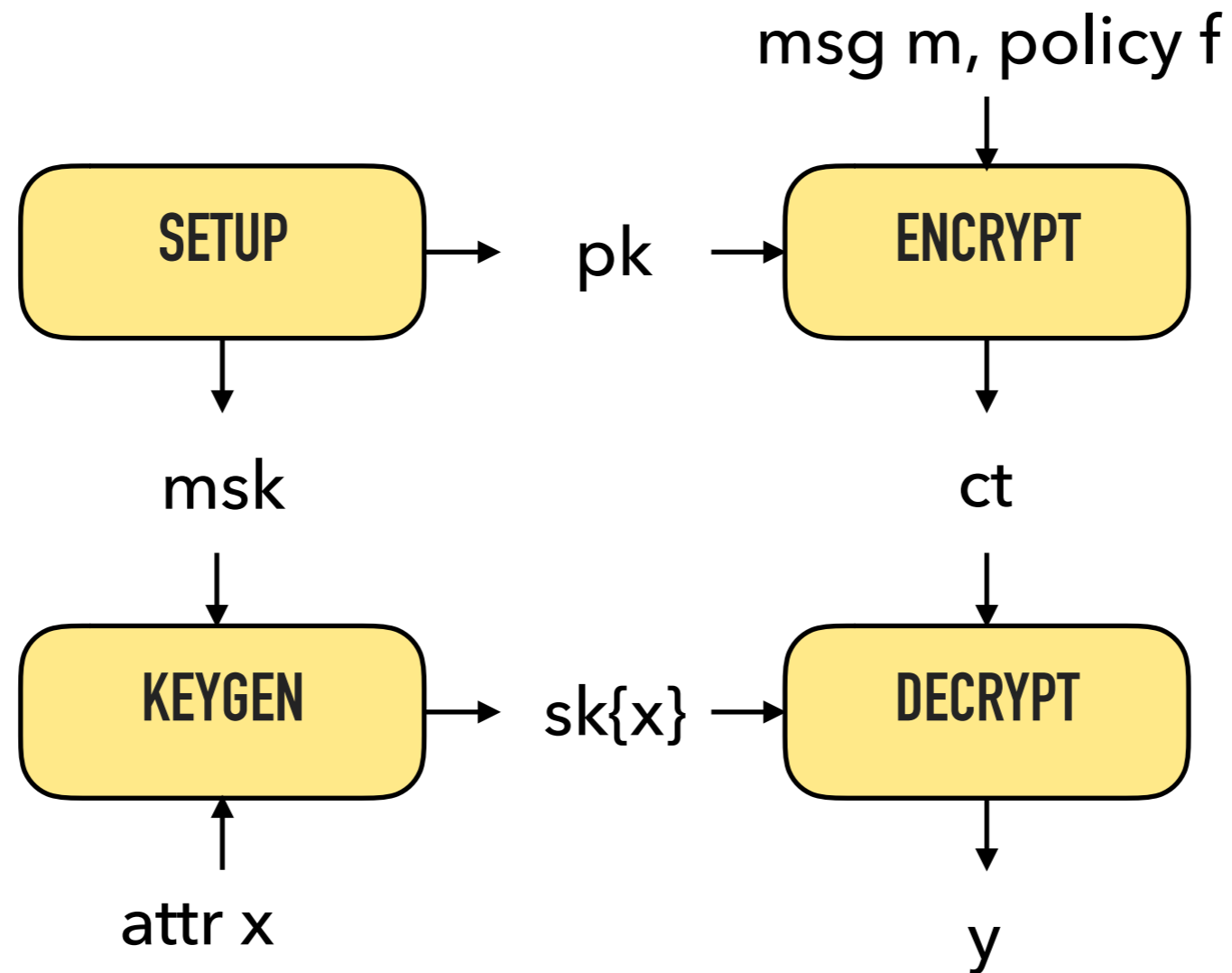
ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



If $f(x) = 1$, then $y = m$

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



$$ct = \begin{array}{|c|c|} \hline m & \text{policy } f \\ \hline \end{array} \quad sk\{x_1\} \quad sk\{x_2\} \quad \dots \quad sk\{x_q\}$$

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



$$ct = \begin{array}{|c|c|} \hline m & \text{policy } f \\ \hline \end{array}$$

$sk\{x_1\}$ $sk\{x_2\}$... $sk\{x_q\}$

ATTRIBUTE BASED
ENCRYPTION

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



$$ct = \begin{matrix} \text{m} & \text{policy } f \end{matrix} \quad sk\{x_1\} \quad sk\{x_2\} \quad \dots \quad sk\{x_q\}$$

ATTRIBUTE BASED ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f may not be hidden

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



ct = m policy f

$sk\{x_1\}$ $sk\{x_2\}$... $sk\{x_q\}$

ATTRIBUTE BASED ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f may not be hidden

PREDICATE ENCRYPTION

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



ct = m policy f

$sk\{x_1\}$ $sk\{x_2\}$... $sk\{x_q\}$

ATTRIBUTE BASED ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f may not be hidden

PREDICATE ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f is hidden
(even if $f(x_k) = 1$ for some k)

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



$$ct = \begin{matrix} m & \text{policy } f \end{matrix} \quad sk\{x_1\} \quad sk\{x_2\} \quad \dots \quad sk\{x_q\}$$

ATTRIBUTE BASED ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f may not be hidden

PREDICATE ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f is hidden
(even if $f(x_k) = 1$ for some k)

IMPLIES OBFUSCATION!

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



$$ct = \boxed{m} \boxed{\text{policy } f} \quad sk\{x_1\} \quad sk\{x_2\} \quad \dots \quad sk\{x_q\}$$

ATTRIBUTE BASED ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f may not be hidden

ONE-SIDED PREDICATE ENCRYPTION

PREDICATE ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f is hidden
(even if $f(x_k) = 1$ for some k)

IMPLIES OBFUSCATION!

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



$$ct = \begin{matrix} m & \text{policy } f \end{matrix} \quad sk\{x_1\} \quad sk\{x_2\} \quad \dots \quad sk\{x_q\}$$

ATTRIBUTE BASED ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f may not be hidden

ONE-SIDED PREDICATE ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

If $f(x_k) = 0$ for all k ,
 f is hidden

PREDICATE ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f is hidden
(even if $f(x_k) = 1$ for some k)

IMPLIES OBFUSCATION!

ATTRIBUTE BASED ENCRYPTION [SAHAI-WATERS 05]



$$ct = \begin{matrix} m & \text{policy } f \end{matrix} \quad sk\{x_1\} \quad sk\{x_2\} \quad \dots \quad sk\{x_q\}$$

ATTRIBUTE BASED ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f may not be hidden

ONE-SIDED PREDICATE ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

If $f(x_k) = 0$ for all k ,
 f is hidden

If $f(x_k) = 1$ for some k ,
 f may not be hidden

PREDICATE ENCRYPTION

If $f(x_k) = 0$ for all k ,
then m is hidden

f is hidden
(even if $f(x_k) = 1$ for some k)

IMPLIES OBFUSCATION!

ATTRIBUTE BASED ENC.

ATTRIBUTE BASED ENC.

Key-Policy

[GPSW 06]

Ciphertext-Policy

[BSW 07]

Multi-Authority

[C 07; CC 09]

Regular Languages

[W 12]

Circuits

[GVW 13]

Circuits : Short keys

[BGGHNSV 13]

NFA

[AMY 19]

ATTRIBUTE BASED ENC.

Key-Policy
[GPSW 06]

Ciphertext-Policy
[BSW 07]

Multi-Authority
[C 07; CC 09]

Regular Languages
[W 12]

Circuits
[GVW 13]

Circuits : Short keys
[BGGHNSV 13]

NFA
[AMY 19]

(ONE-SIDED) PREDICATE ENC.

Circuits : Short keys
[GVW 15]

ATTRIBUTE BASED ENC.

Key-Policy
[GPSW 06]

Ciphertext-Policy
[BSW 07]

Multi-Authority
[C 07; CC 09]

Regular Languages
[W 12]

Circuits
[GVW 13]

Circuits : Short keys
[BGGHNSV 13]

NFA
[AMY 19]

(ONE-SIDED) PREDICATE ENC.

Key-Policy

Ciphertext-Policy

Multi-Authority

Regular Languages

Circuits

Circuits : Short keys
[GVW 15]

NFA

LOCKABLE OBFUSCATION: CONSTRUCTION

LOCKABLE OBFUSCATION: BEHIND THE SCENES

LOCKABLE OBFUSCATION: BEHIND THE SCENES

2-Circular Security Separations

[Bishop, Hohenberger, Waters 15]

LOCKABLE OBFUSCATION: BEHIND THE SCENES

2-Circular Security Separations

[Bishop, Hohenberger, Waters 15]

n-Circular Security Separations

[Alamati, Peikert 16; K, Waters 16]

LOCKABLE OBFUSCATION: BEHIND THE SCENES

2-Circular Security Separations

[Bishop, Hohenberger, Waters 15]

n-Circular Security Separations

[Alamati, Peikert 16; K, Waters 16]

Bit Circular Security Separations

[Goyal, K, Waters 17a]

LOCKABLE OBFUSCATION: BEHIND THE SCENES

2-Circular Security Separations

[Bishop, Hohenberger, Waters 15]

n-Circular Security Separations

[Alamati, Peikert 16; K, Waters 16]

Bit Circular Security Separations

[Goyal, K, Waters 17a]

Lockable Obfuscation

[Goyal, K, Waters 17b; Wichs, Zirdelis 17]

LOCKABLE OBFUSCATION: CONSTRUCTION

STEP 1: Lockable Obf. for NC^1

STEP 2: Bootstrapping

LO for NC^1 + FHE \longrightarrow LO for P/poly

* FHE with NC^1 decryption

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

OBF() : Obfuscator for NC¹

(ENC, DEC, EVAL) : FHE scheme

OBFUSCATION OF CKT. C WITH LOCK α :

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

OBF() : Obfuscator for NC¹

(ENC, DEC, EVAL) : FHE scheme

OBFUSCATION OF CKT. C WITH LOCK α :

ENC (C, SK)

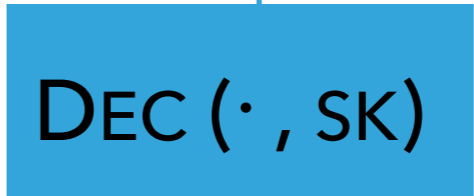
STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

OBF() : Obfuscator for NC¹

(ENC, DEC, EVAL) : FHE scheme

OBFUSCATION OF CKT. C WITH LOCK α :

ENC (C, SK)

OBF ( α)

Obf. of Dec ckt with lock α

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C: $\text{ENC}(C, SK) \text{ OBF} \left(\text{DEC}(\cdot, SK) \alpha \right)$

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C : $\text{ENC}(C, SK) \rightarrow \text{OBF} \left(\text{DEC}(\cdot, SK), \alpha \right)$

Evaluation on input x :

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C : $ENC(C, SK)$ OBF $\left(\begin{array}{c} \boxed{DEC(\cdot, SK)} \\ \alpha \end{array} \right)$

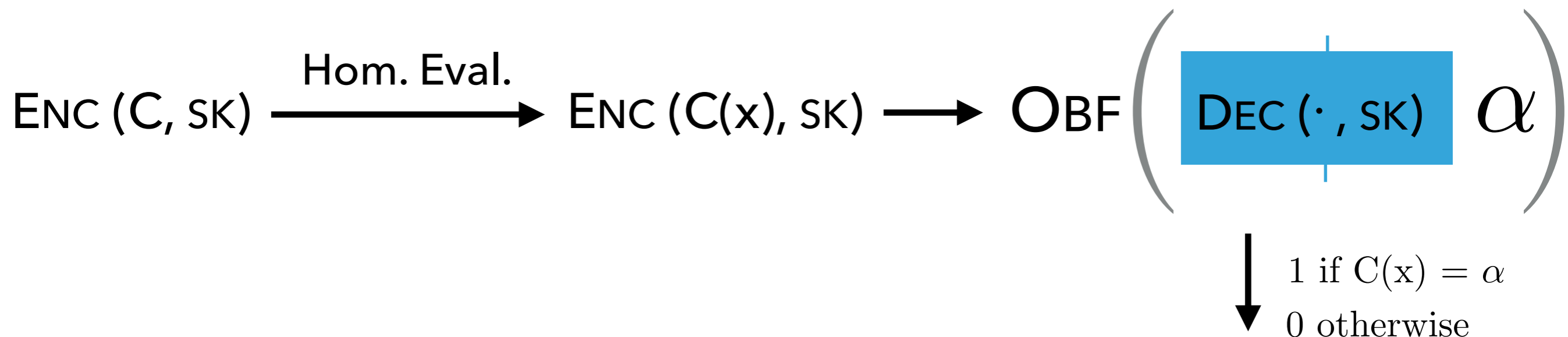
Evaluation on input x :

$ENC(C, SK) \xrightarrow{\text{Hom. Eval.}} ENC(C(x), SK)$

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C : $\text{ENC}(C, SK) \xrightarrow{\text{OBF}} \left(\text{DEC}(\cdot, SK), \alpha \right)$

Evaluation on input x :



STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C: $\text{ENC}(C, SK) \text{ OBF} \left(\text{DEC}(\cdot, SK) \alpha \right)$

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C: $\text{ENC}(C, SK) \text{ OBF} \left(\text{DEC}(\cdot, SK) \alpha \right)$

Security:

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C: $\text{ENC}(C, SK)$ $\text{OBF} \left(\text{DEC}(\cdot, SK) \quad \alpha \right)$

Security:



Security of OBF

$\text{ENC}(C, SK)$ $\text{OBF} \left(\# \quad \# \right)$

STEP 2: BOOTSTRAPPING LOCKABLE OBFUSCATION

Obfuscation of C: $\text{ENC}(C, SK)$ $\text{OBF}\left(\text{DEC}(\cdot, SK), \alpha\right)$

Security:

\approx
Security of OBF

$\text{ENC}(C, SK)$ $\text{OBF}\left(\#, \#\right)$

\approx
Security of FHE

$\text{ENC}(\#, SK)$ $\text{OBF}\left(\#, \#\right)$

LOCKABLE OBFUSCATION: CONSTRUCTION

STEP 1: Lockable Obf. for NC¹

STEP 2: Bootstrapping

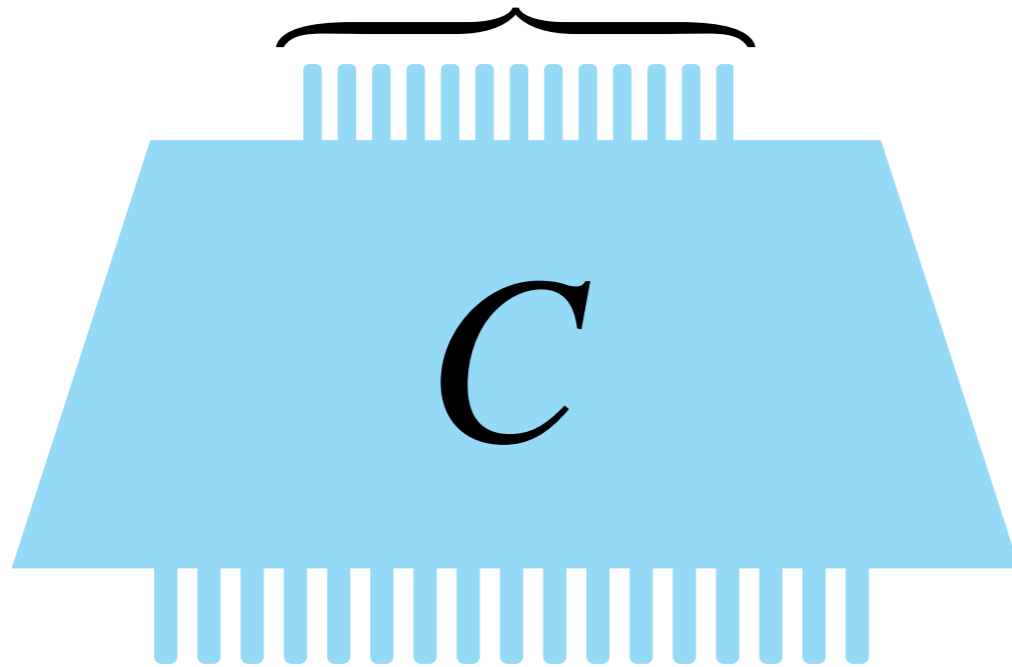
LO for NC¹ + FHE → LO for P/poly

* FHE with NC¹ decryption

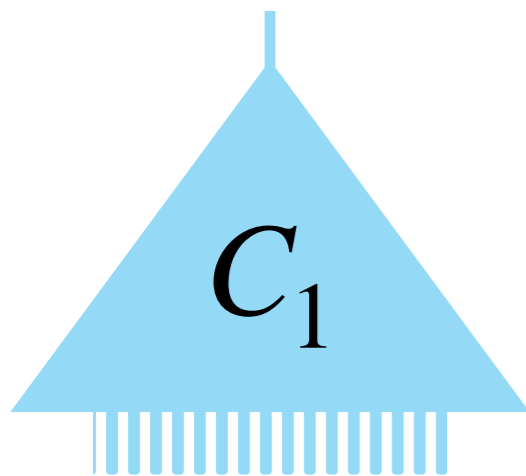


Obfuscate:

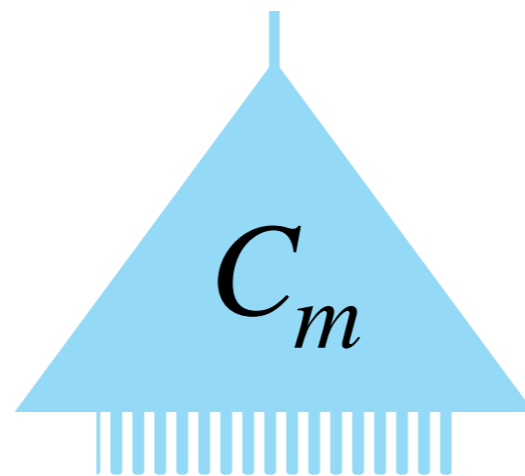
m bits

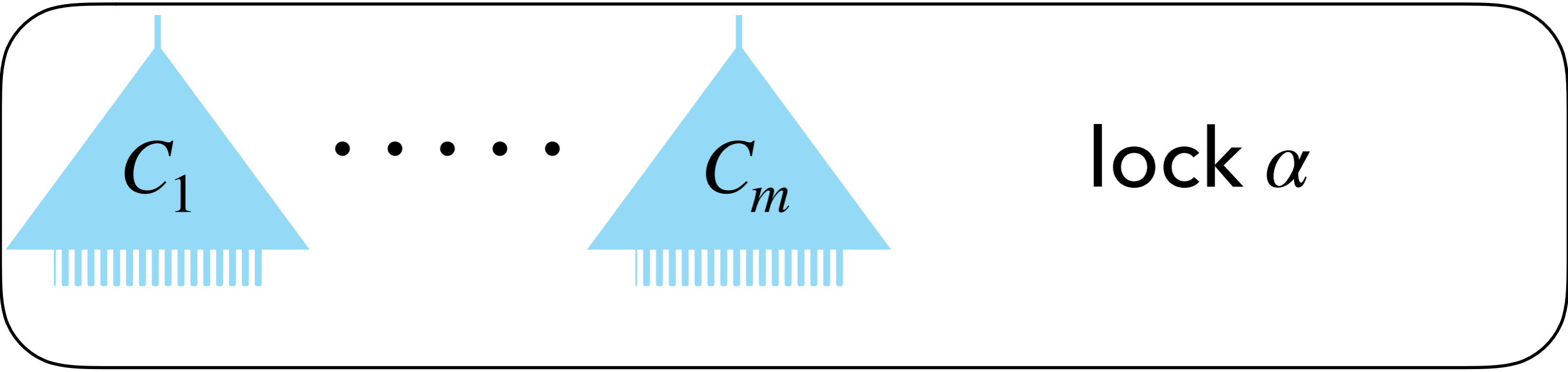


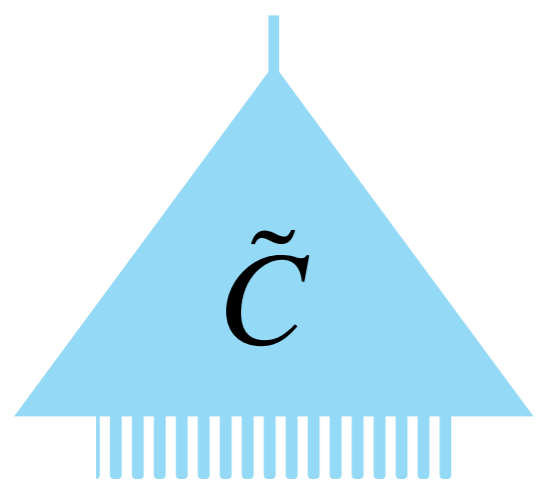
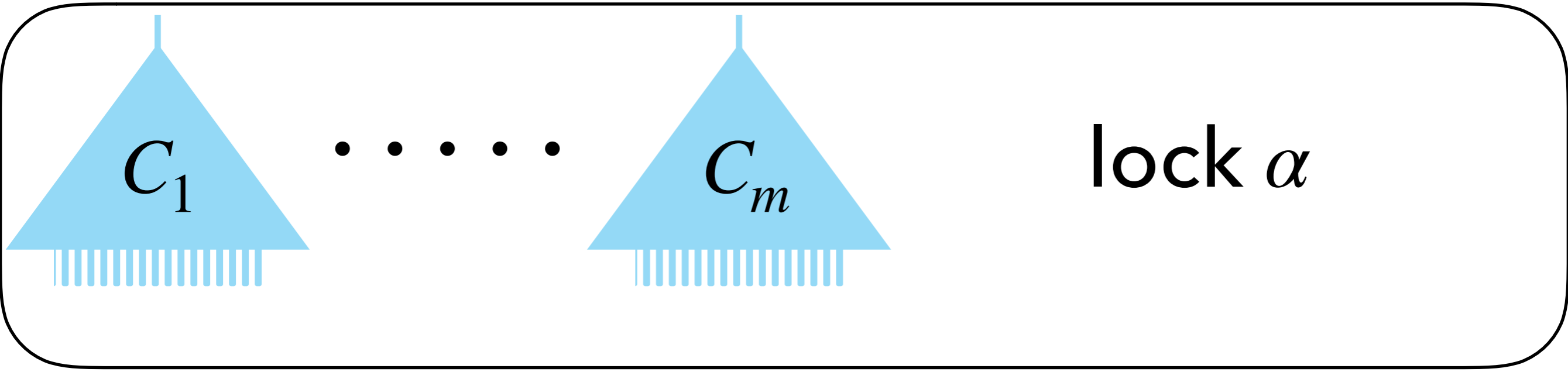
lock $\alpha \in \{0,1\}^m$

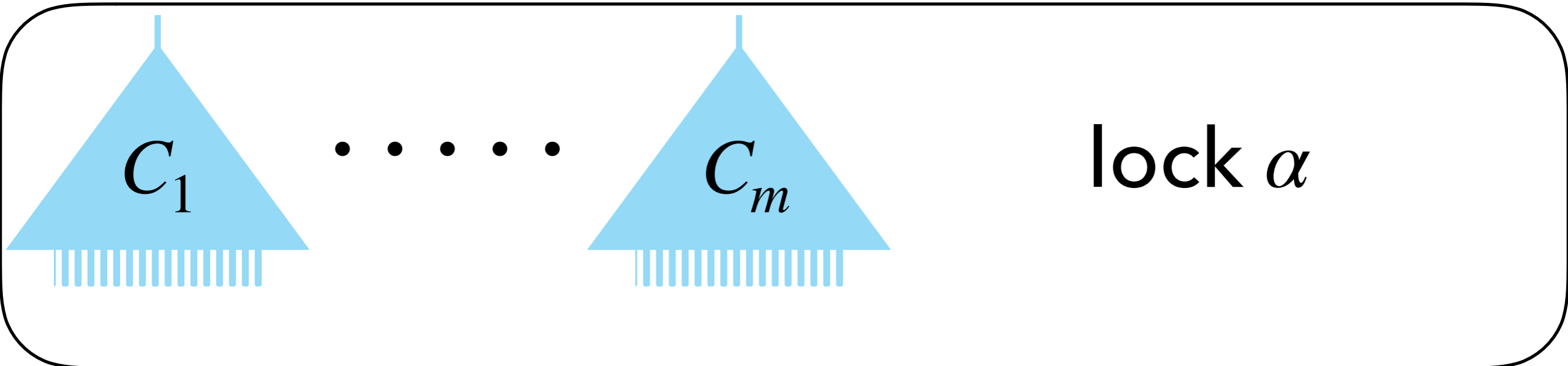


...

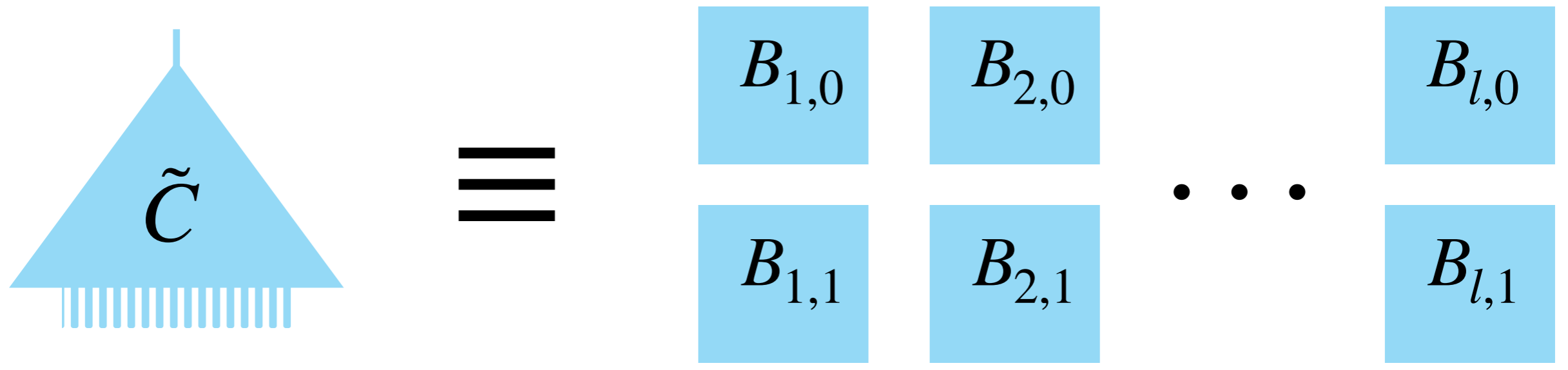


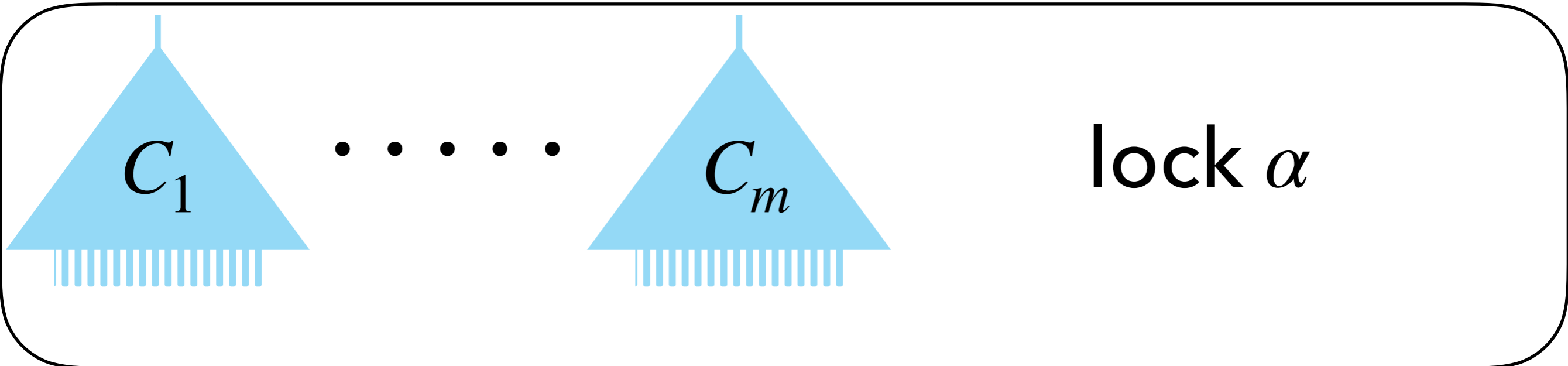




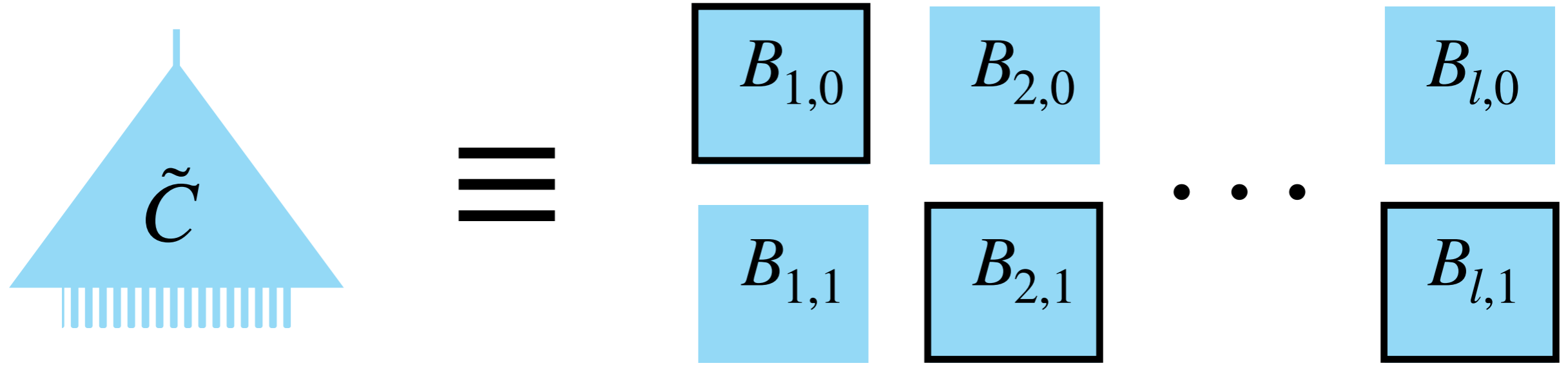


Permutation Matrix Branching Program





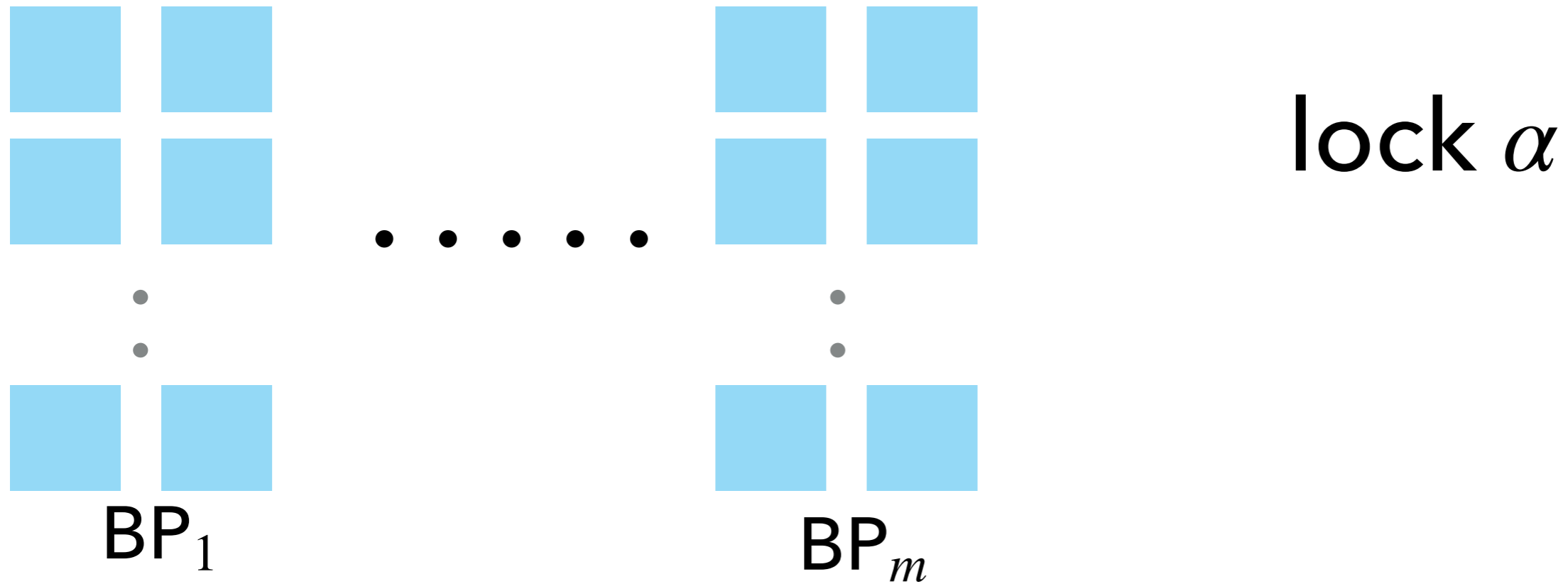
Permutation Matrix Branching Program



$$\tilde{C}(x) = 1 \implies \prod B_{j,x_j} = P_{\text{acc}}$$

$$\tilde{C}(x) = 0 \implies \prod B_{j,x_j} = I$$

Obfuscate:



Evaluation on x : Output 1 iff $BP_i(x) = \alpha_i$ for all i .

Security : Obfuscation hides all BP_i

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

For each i , 'encode' $BP_i : M_{i,0} : M_{i,1}$.

Encoding _{i} hides BP_i

Evaluation of encoding _{i} on x outputs $\approx M_{i,BP_i(x)}$

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

For each i , 'encode' $BP_i : M_{i,0} : M_{i,1}$.

Encoding _{i} hides BP_i

Evaluation of encoding _{i} on x outputs $\approx R_x \cdot M_{i,BP_i(x)}$

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

For each i , 'encode' $BP_i : M_{i,0} : M_{i,1}$.

Encoding _{i} hides BP_i

Evaluation of encoding _{i} on x outputs $\approx R_x \cdot M_{i,BP_i(x)}$

Output all encodings.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

$$\sum_i M_{i, \alpha_i} = 0.$$

Eval of encoding_{*i*} on $x \approx R_x \cdot M_{i, BP_i(x)}$

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

$$\sum_i M_{i,\alpha_i} = 0.$$

Eval of encoding_{*i*} on $x \approx R_x \cdot M_{i,BP_i(x)}$

Evaluation on x :

Encoding_{*i*} $\longrightarrow \approx R_x \cdot M_{i,BP_i(x)}$

Output 1 if the sum is small.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

$$\sum_i M_{i,\alpha_i} = 0.$$

Eval of encoding_{*i*} on $x \approx R_x \cdot M_{i,BP_i(x)}$

Evaluation on x :

if $BP_i(x) = \alpha_i$ for all i

Encoding_{*i*} $\longrightarrow \approx R_x \cdot M_{i,BP_i(x)}$

Output 1 if the sum is small.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

$$\sum_i M_{i,\alpha_i} = 0.$$

Eval of encoding_{*i*} on $x \approx R_x \cdot M_{i,BP_i(x)}$

Evaluation on x :

if $BP_i(x) = \alpha_i$ for all i

Encoding_{*i*} $\longrightarrow \approx R_x \cdot M_{i,BP_i(x)} \qquad \approx R_x \cdot M_{i,\alpha_i}$

Output 1 if the sum is small.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

$$\sum_i M_{i,\alpha_i} = 0.$$

Eval of encoding_{*i*} on $x \approx R_x \cdot M_{i,BP_i(x)}$

Evaluation on x :

if $BP_i(x) = \alpha_i$ for all i

Encoding_{*i*} $\longrightarrow \approx R_x \cdot M_{i,BP_i(x)}$

$$\approx R_x \cdot M_{i,\alpha_i}$$

Output 1 if the sum is small.

$$\approx \sum R_x \cdot M_{i,\alpha_i}$$

$$= R_x \cdot \left(\sum M_{i,\alpha_i} \right)$$

$$= 0$$

'Encode' BP : $M_0 : M_1$.

WANT:

Encoding hides BP

Eval of encoding on $x \approx R_x \cdot M_{\text{BP}(x)}$

$B_{1,0}$

$B_{2,0}$

M_0

$B_{1,1}$

$B_{2,1}$

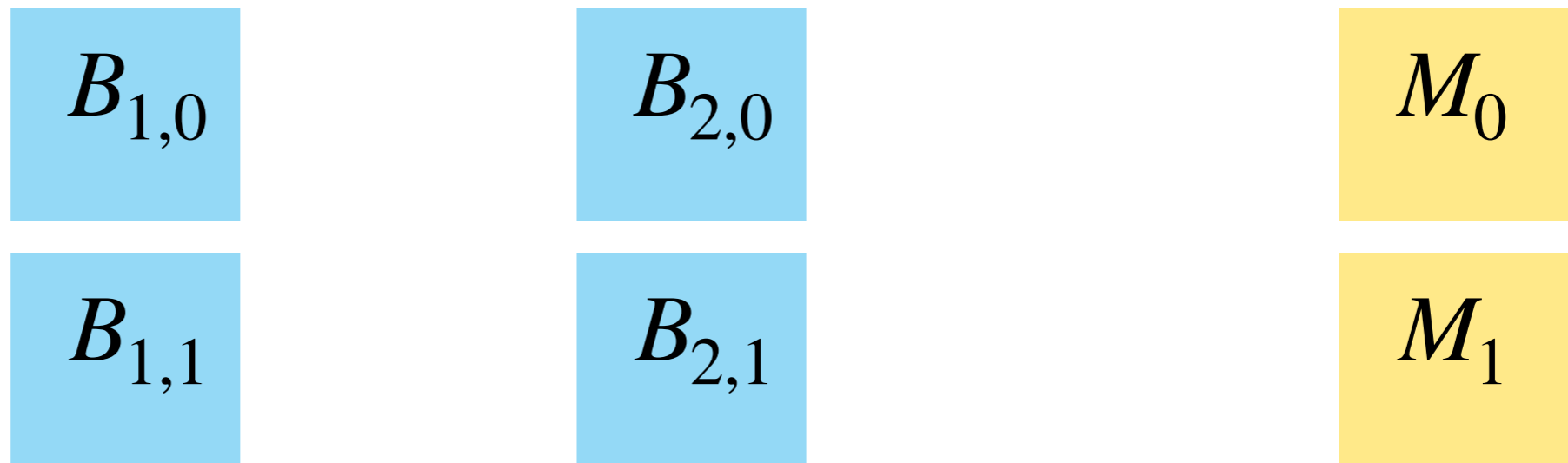
M_1

'Encode' BP : $M_0 : M_1$.

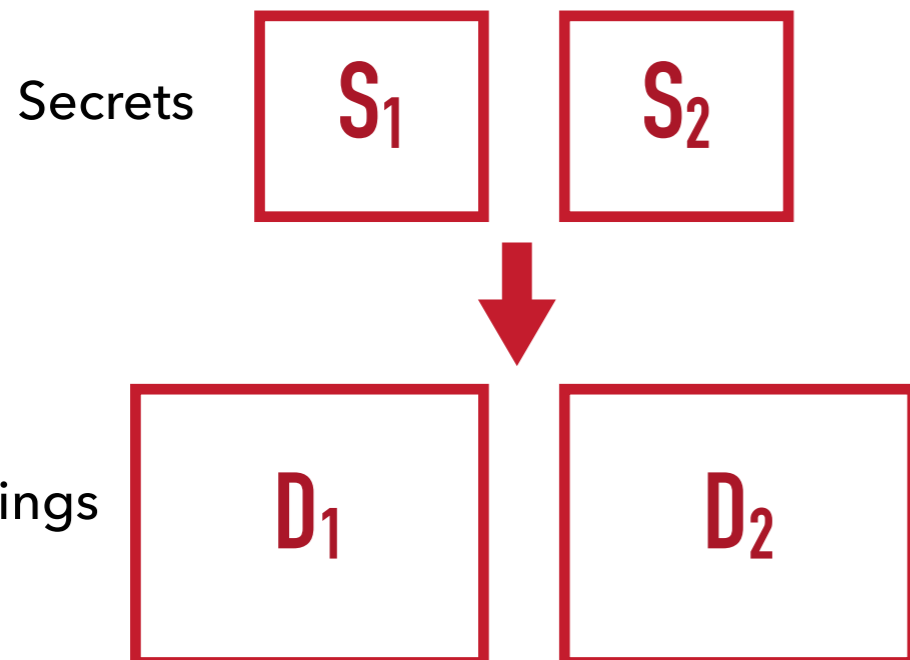
WANT:

Encoding hides BP

Eval of encoding on $x \approx R_x \cdot M_{BP(x)}$

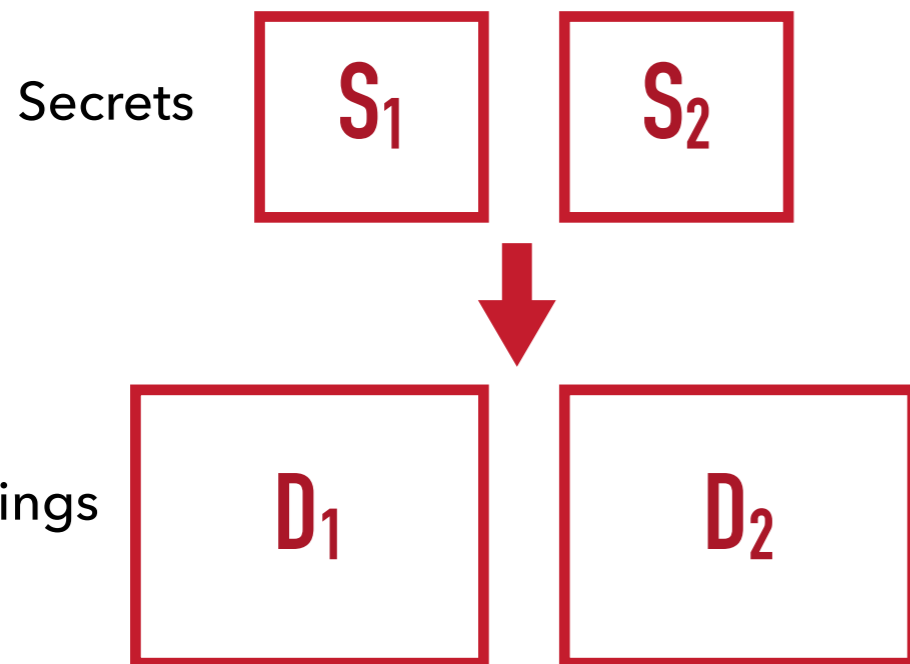


GGH15 ENCODING OF 'MATRIX' SECRETS:



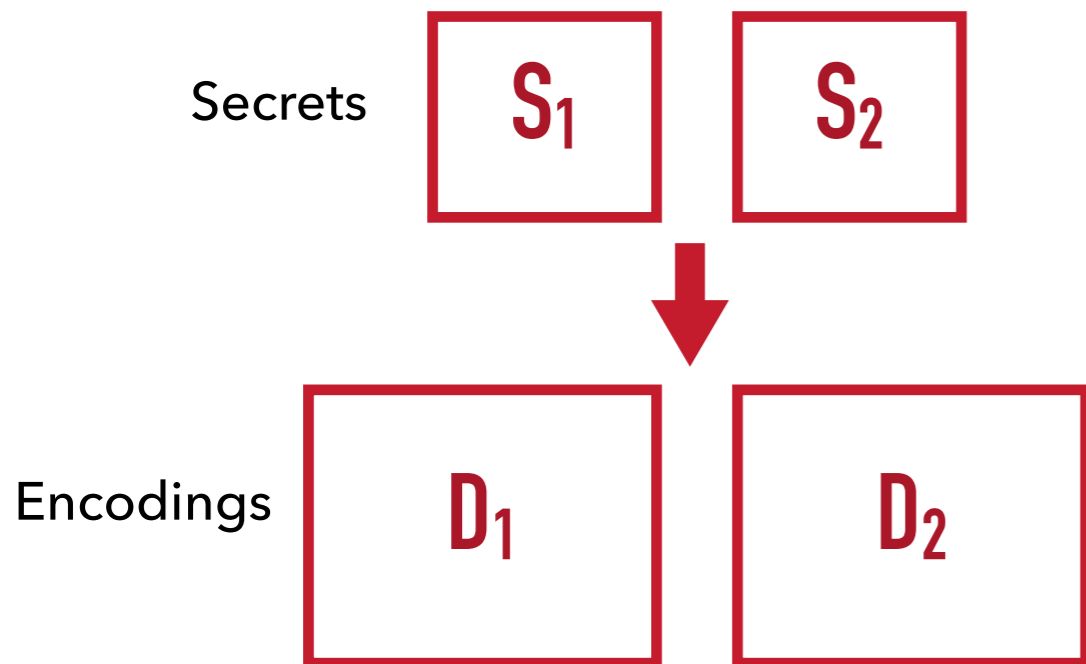
$$\begin{aligned}
 & \text{-----} \\
 & \mathbf{A_0} \times \mathbf{D_1} \times \mathbf{D_2} \quad \text{Functionality} \\
 = & \mathbf{S_1} \mathbf{A_1} \mathbf{D_2} + \mathbf{E_1} \mathbf{D_2} \\
 = & \mathbf{S_1} \mathbf{S_2} \mathbf{A_2} + \mathbf{S_1} \mathbf{E_2} + \mathbf{E_1} \mathbf{D_2} \\
 = & \mathbf{S_1} \mathbf{S_2} \mathbf{A_2} + \text{"small"}
 \end{aligned}$$

GGH15 ENCODING OF 'MATRIX' SECRETS:



$$\begin{aligned}
 & \mathbf{A}_0 \times \mathbf{D}_1 \times \mathbf{D}_2 && \text{Functionality} \\
 = & \mathbf{S}_1 \mathbf{A}_1 \mathbf{D}_2 + \mathbf{E}_1 \mathbf{D}_2 \\
 = & \mathbf{S}_1 \mathbf{S}_2 \mathbf{A}_2 + \mathbf{S}_1 \mathbf{E}_2 + \mathbf{E}_1 \mathbf{D}_2 \\
 = & \mathbf{S}_1 \mathbf{S}_2 \mathbf{A}_2 + \text{"small"}
 \end{aligned}$$

GGH15 ENCODING OF 'MATRIX' SECRETS:

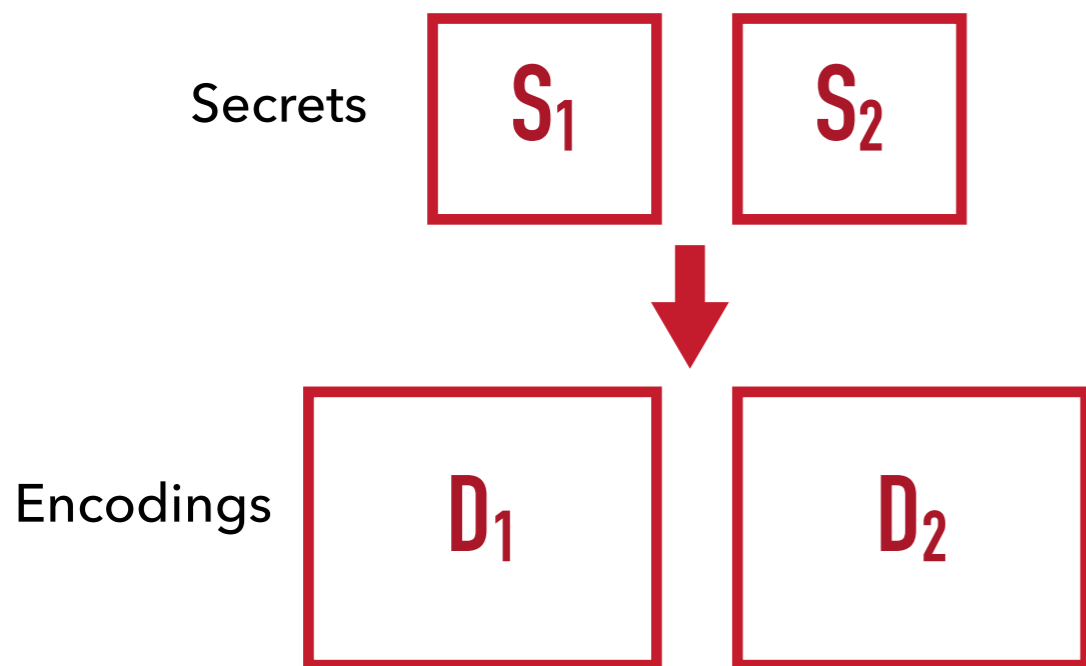


$$\begin{aligned}
 & \mathbf{A}_0 \times \mathbf{D}_1 \times \mathbf{D}_2 && \text{Functionality} \\
 = & \begin{bmatrix} S_1 & \mathbf{A}_1 & \mathbf{D}_2 \end{bmatrix} + \begin{bmatrix} E_1 & \mathbf{D}_2 \end{bmatrix} \\
 = & \begin{bmatrix} S_1 & S_2 & \mathbf{A}_2 \end{bmatrix} + \begin{bmatrix} S_1 & E_2 \end{bmatrix} + \begin{bmatrix} E_1 & \mathbf{D}_2 \end{bmatrix} \\
 = & \begin{bmatrix} S_1 & S_2 & \mathbf{A}_2 \end{bmatrix} + \text{"small"}
 \end{aligned}$$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

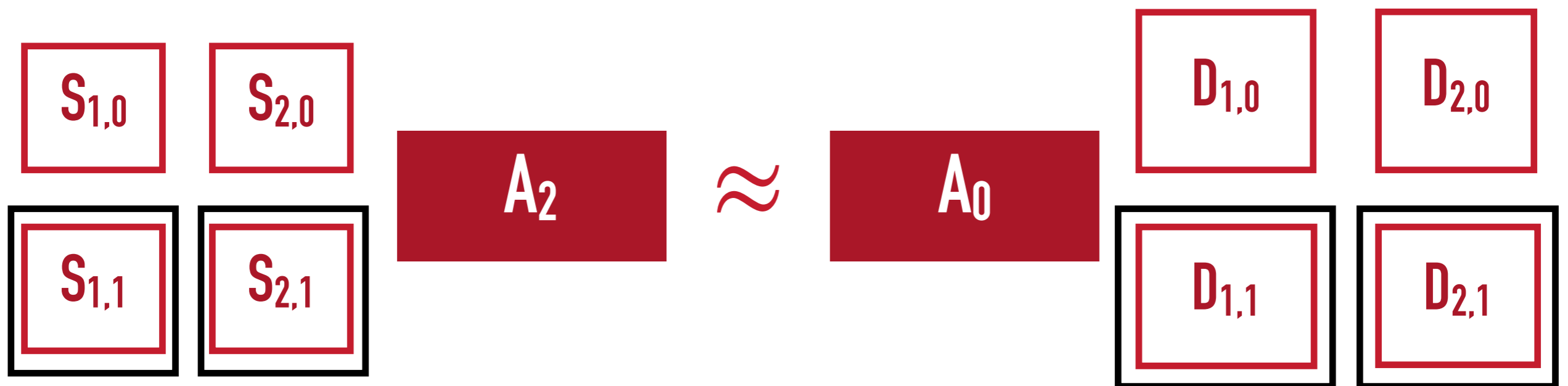


GGH15 ENCODING OF 'MATRIX' SECRETS:

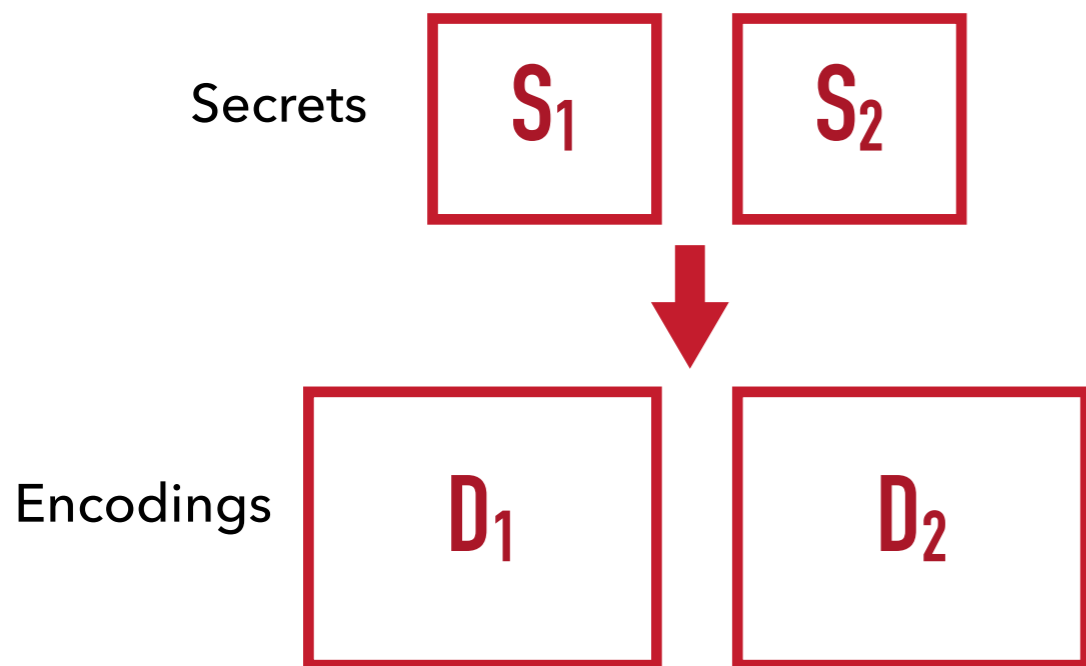


$$\begin{aligned}
 & \text{---} \\
 & \mathbf{A_0} \times \mathbf{D_1} \times \mathbf{D_2} \quad \text{Functionality} \\
 = & \mathbf{S_1} \mathbf{A_1} \mathbf{D_2} + \mathbf{E_1} \mathbf{D_2} \\
 = & \mathbf{S_1} \mathbf{S_2} \mathbf{A_2} + \mathbf{S_1} \mathbf{E_2} + \mathbf{E_1} \mathbf{D_2} \\
 = & \mathbf{S_1} \mathbf{S_2} \mathbf{A_2} + \text{"small"}
 \end{aligned}$$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

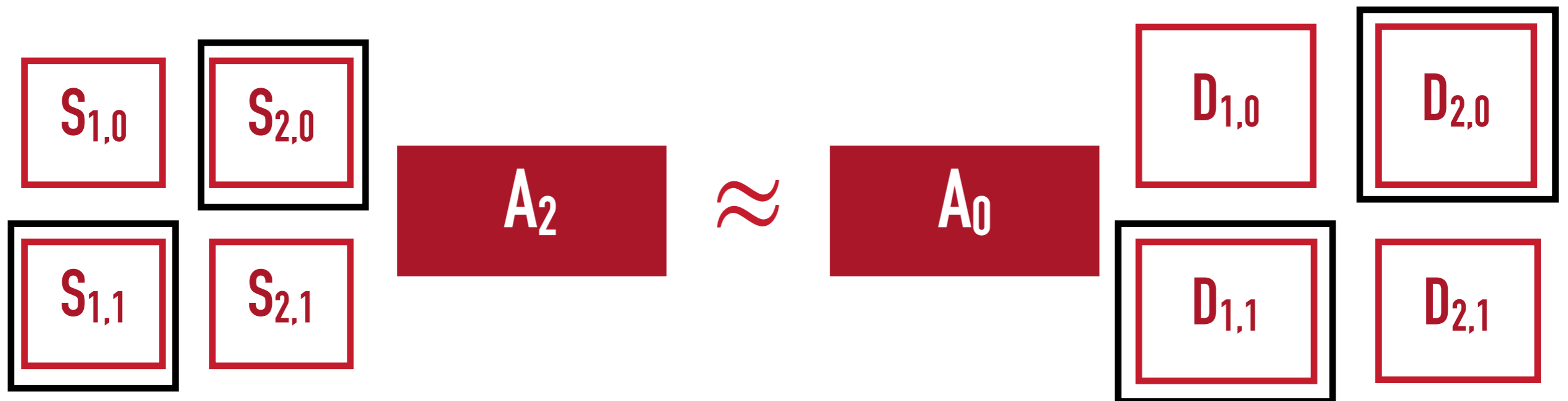


GGH15 ENCODING OF 'MATRIX' SECRETS:

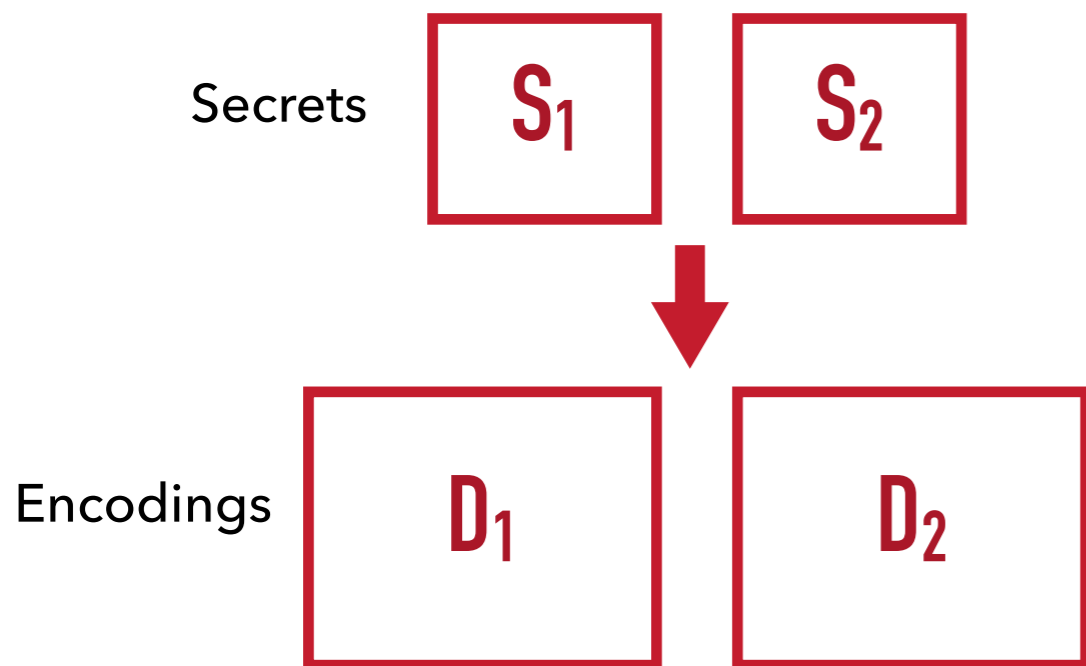


$$\begin{aligned}
 & \mathbf{A}_0 \times \mathbf{D}_1 \times \mathbf{D}_2 && \text{Functionality} \\
 = & \begin{bmatrix} S_1 & \mathbf{A}_1 & \mathbf{D}_2 \end{bmatrix} + \begin{bmatrix} E_1 & \mathbf{D}_2 \end{bmatrix} \\
 = & \begin{bmatrix} S_1 & S_2 & \mathbf{A}_2 \end{bmatrix} + \begin{bmatrix} S_1 & E_2 \end{bmatrix} + \begin{bmatrix} E_1 & \mathbf{D}_2 \end{bmatrix} \\
 = & \begin{bmatrix} S_1 & S_2 & \mathbf{A}_2 \end{bmatrix} + \text{"small"}
 \end{aligned}$$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

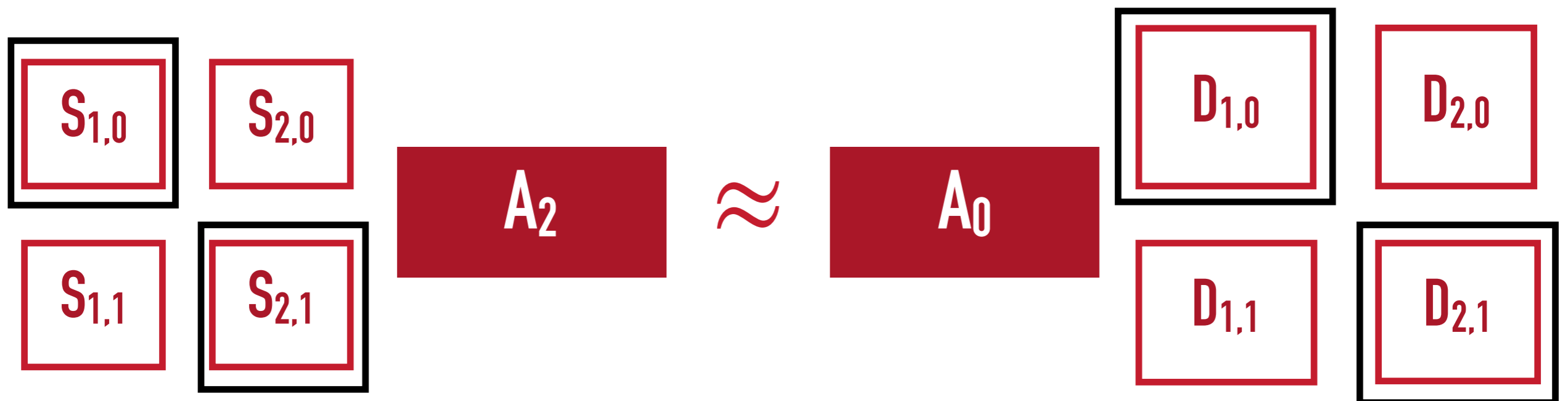


GGH15 ENCODING OF 'MATRIX' SECRETS:

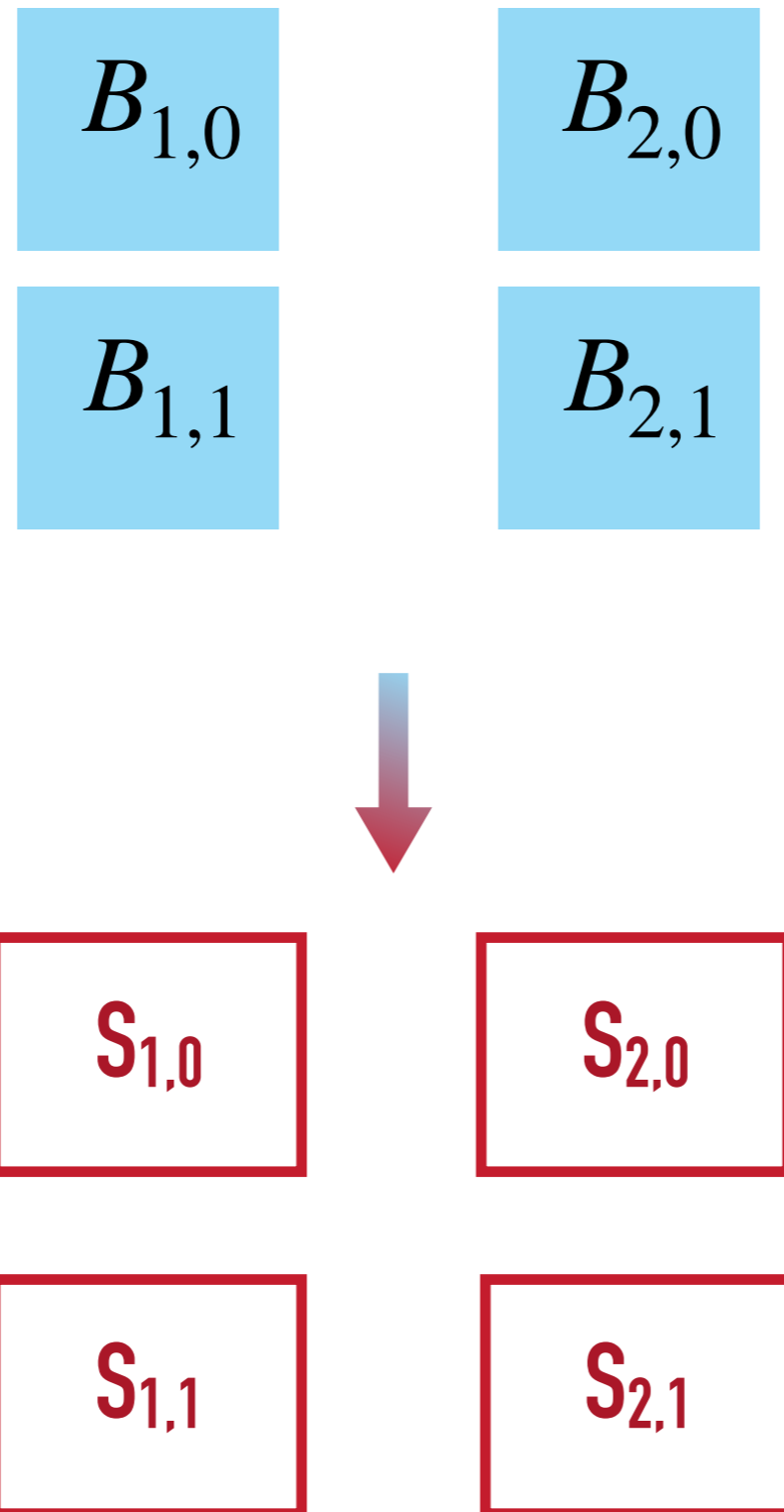


$$\begin{aligned}
 & \mathbf{A}_0 \times \mathbf{D}_1 \times \mathbf{D}_2 \quad \text{Functionality} \\
 = & \begin{bmatrix} S_1 & \mathbf{A}_1 & \mathbf{D}_2 \end{bmatrix} + \begin{bmatrix} E_1 & \mathbf{D}_2 \end{bmatrix} \\
 = & \begin{bmatrix} S_1 & S_2 & \mathbf{A}_2 \end{bmatrix} + \begin{bmatrix} S_1 & E_2 \end{bmatrix} + \begin{bmatrix} E_1 & \mathbf{D}_2 \end{bmatrix} \\
 = & \begin{bmatrix} S_1 & S_2 & \mathbf{A}_2 \end{bmatrix} + \text{"small"}
 \end{aligned}$$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS



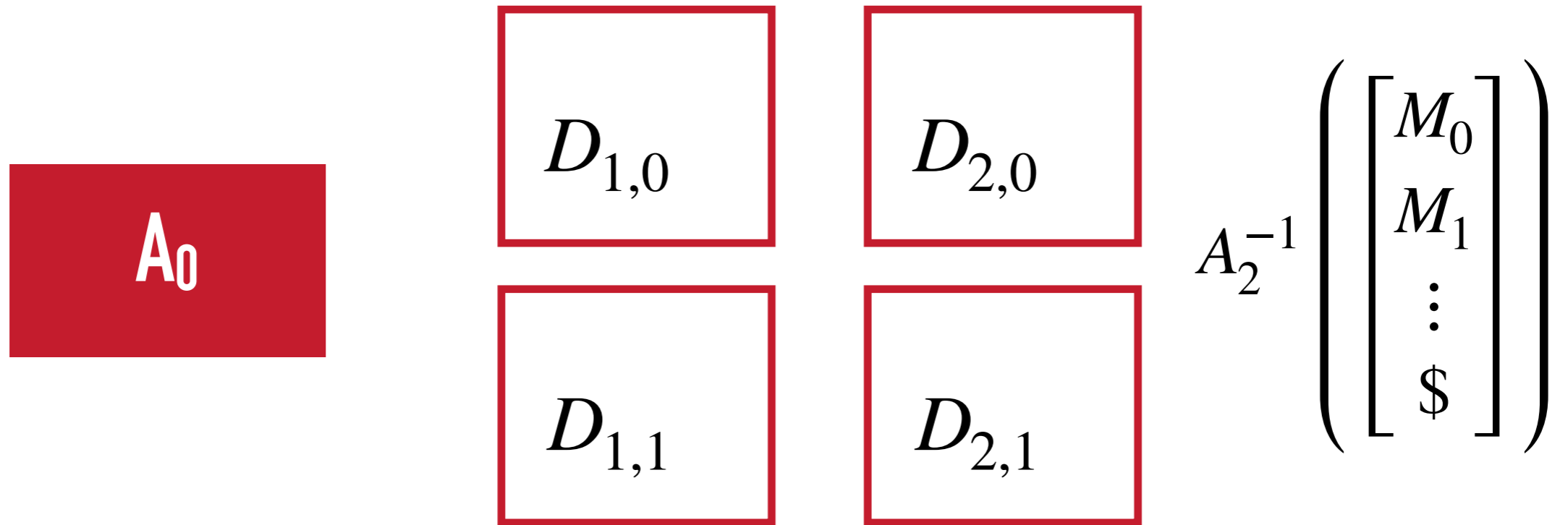
FROM BRANCHING PROGRAMS TO GGH15 SECRETS:



FROM BRANCHING PROGRAMS TO GGH15 SECRETS:

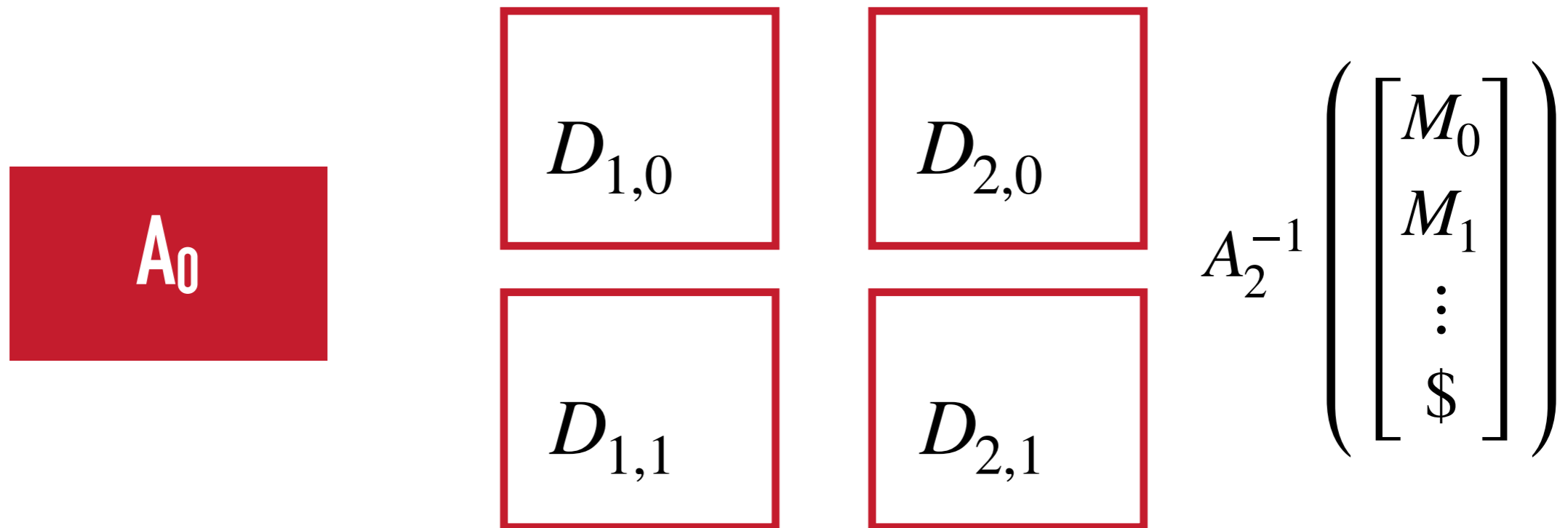
$$\left(\prod B_{i,x_i}\right) \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ \$ \end{bmatrix} = \begin{bmatrix} & & & & \\ & 1 & & & \\ 1 & & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ \$ \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ \$ \end{bmatrix}$$

FROM GGH15 SECRETS TO ENCODINGS:



$$D_{i,b} = A_{i-1}^{-1} \left((B_{i,b} \otimes I) A_i + \text{noise} \right)$$

FROM GGH15 SECRETS TO ENCODINGS:



$$D_{i,b} = A_{i-1}^{-1} \left((B_{i,b} \otimes I) A_i + \text{noise} \right)$$

Eval of encoding on $x = M_{\text{BP}(x)}$

FROM GGH15 SECRETS TO ENCODINGS:

Eval of encoding on $x = M_{\text{BP}(x)}$

FROM GGH15 SECRETS TO ENCODINGS:

Eval of encoding on $x = M_{\text{BP}(x)}$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS



FROM GGH15 SECRETS TO ENCODINGS:

Eval of encoding on $x = M_{\text{BP}(x)}$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

$$\begin{array}{cc} \boxed{S_{1,0}} & \boxed{S_{2,0}} \\ \boxed{S_{1,1}} & \boxed{S_{2,1}} \end{array} \quad \mathbf{A}_2 \quad A_2^{-1} \begin{pmatrix} [M_0] \\ M_1 \\ \vdots \\ \$ \end{pmatrix} \approx \mathbf{A}_0 \quad \begin{array}{cc} \boxed{D_{1,0}} & \boxed{D_{2,0}} \\ \boxed{D_{1,1}} & \boxed{D_{2,1}} \end{array} \quad A_2^{-1} \begin{pmatrix} [M_0] \\ M_1 \\ \vdots \\ \$ \end{pmatrix}$$

FROM GGH15 SECRETS TO ENCODINGS:

Eval of encoding on $x = M_{\text{BP}(x)}$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

$$\begin{array}{c} \boxed{S_{1,0}} \\ \boxed{S_{1,1}} \end{array} \begin{array}{c} \boxed{S_{2,0}} \\ \boxed{S_{2,1}} \end{array} \mathbf{A}_2 A_2^{-1} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix} \approx \mathbf{A}_0 \begin{array}{c} \boxed{D_{1,0}} \\ \boxed{D_{1,1}} \end{array} \begin{array}{c} \boxed{D_{2,0}} \\ \boxed{D_{2,1}} \end{array} A_2^{-1} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix}$$

$$\left(\prod \overbrace{B_{i,x_i}^{S_{i,x_i}}} \otimes I \right) \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix}$$

FROM GGH15 SECRETS TO ENCODINGS:

Eval of encoding on $x = M_{\text{BP}(x)}$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

$$\begin{array}{c} \boxed{S_{1,0}} \\ \boxed{S_{1,1}} \end{array} \begin{array}{c} \boxed{S_{2,0}} \\ \boxed{S_{2,1}} \end{array} \mathbf{A}_2 A_2^{-1} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix} \approx \mathbf{A}_0 \begin{array}{c} \boxed{D_{1,0}} \\ \boxed{D_{1,1}} \end{array} \begin{array}{c} \boxed{D_{2,0}} \\ \boxed{D_{2,1}} \end{array} A_2^{-1} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix}$$

first block of $\left(\prod \overbrace{B_{i,x_i}^{S_{i,x_i}}} \otimes I \right) \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix} = \begin{cases} M_1 & \text{if BP}(x) = 1 \\ M_0 & \text{if BP}(x) = 0 \end{cases}$

FROM GGH15 SECRETS TO ENCODINGS:

Eval of encoding on $x = M_{\text{BP}(x)}$

SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

$$\begin{array}{cc}
 \boxed{S_{1,0}} & \boxed{S_{2,0}} \\
 \boxed{S_{1,1}} & \boxed{S_{2,1}}
 \end{array}
 A_2
 A_2^{-1}
 \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix}
 \approx
 A_0
 \begin{array}{cc}
 \boxed{D_{1,0}} & \boxed{D_{2,0}} \\
 \boxed{D_{1,1}} & \boxed{D_{2,1}}
 \end{array}
 A_2^{-1}
 \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix}$$

first block of $\left(\prod \overbrace{B_{i,x_i}^{S_{i,x_i}}} \otimes I \right) \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix} = \begin{cases} M_1 & \text{if } \text{BP}(x) = 1 \\ M_0 & \text{if } \text{BP}(x) = 0 \end{cases}$

Encoding_{*i*} hides BP_{*i*} ?

FROM GGH15 SECRETS TO ENCODINGS:

Eval of encoding on $x = M_{\text{BP}(x)}$

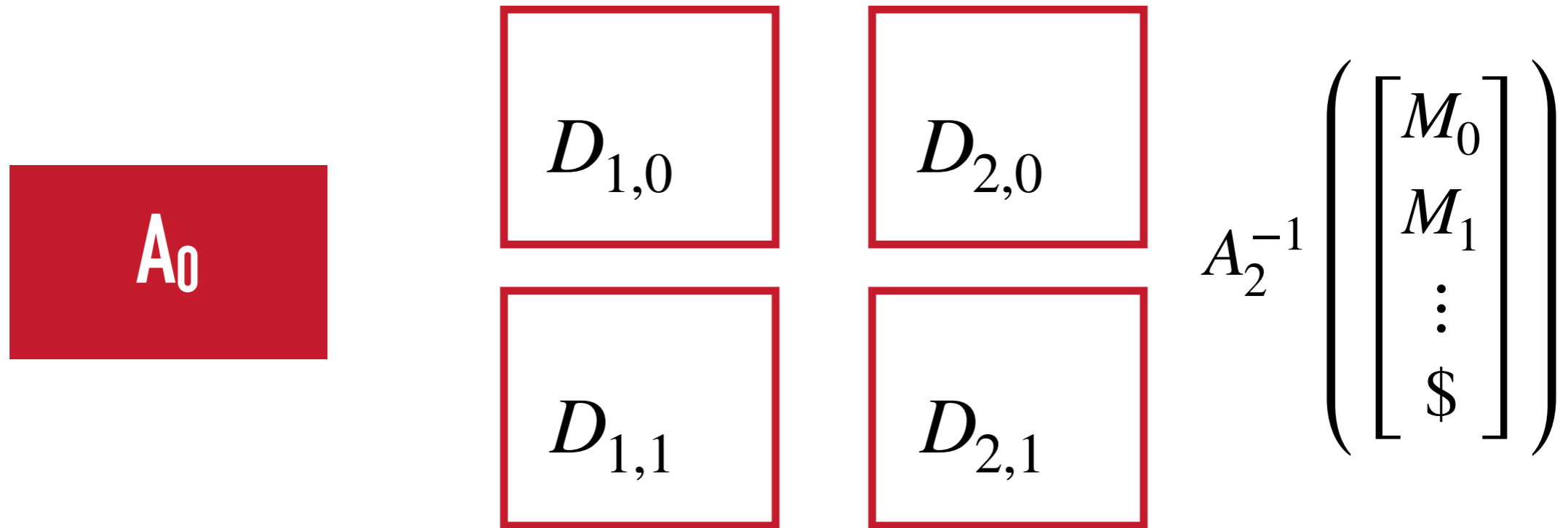
SUBSET PRODUCT OF 'MATRIX' SECRETS \rightarrow SUBSET PRODUCT OF ENCODINGS

$$\begin{array}{cc}
 \boxed{S_{1,0}} & \boxed{S_{2,0}} \\
 \boxed{S_{1,1}} & \boxed{S_{2,1}}
 \end{array}
 \begin{array}{c}
 \mathbf{A}_2 \\
 \mathbf{A}_2^{-1}
 \end{array}
 \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix}
 \approx
 \begin{array}{cc}
 \boxed{D_{1,0}} & \boxed{D_{2,0}} \\
 \boxed{D_{1,1}} & \boxed{D_{2,1}}
 \end{array}
 \begin{array}{c}
 \mathbf{A}_0 \\
 \mathbf{A}_2^{-1}
 \end{array}
 \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix}$$

first block of $\left(\prod \overbrace{B_{i,x_i}^{S_{i,x_i}}} \otimes I \right) \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \$ \end{pmatrix} = \begin{cases} M_1 & \text{if } \text{BP}(x) = 1 \\ M_0 & \text{if } \text{BP}(x) = 0 \end{cases}$

Encoding_{*i*} hides BP_{*i*} ? ~~X~~

SECURE CONSTRUCTION:



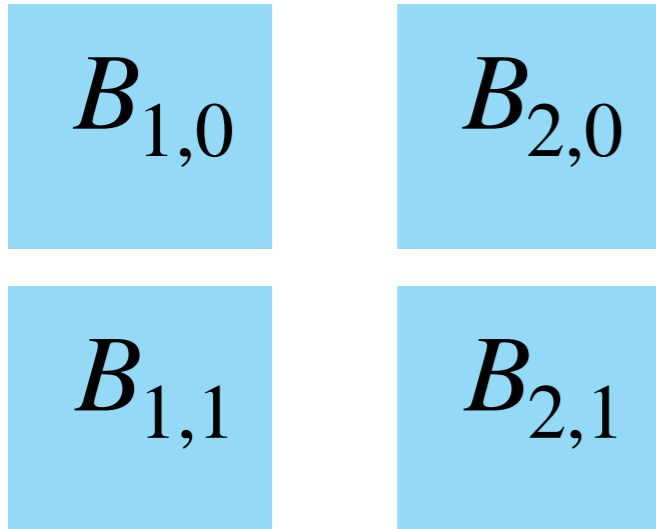
$$D_{i,b} = A_{i-1}^{-1} \left((B_{i,b} \otimes R_{i,b}) A_i + \text{noise} \right)$$

$R_{i,b}$: random low norm

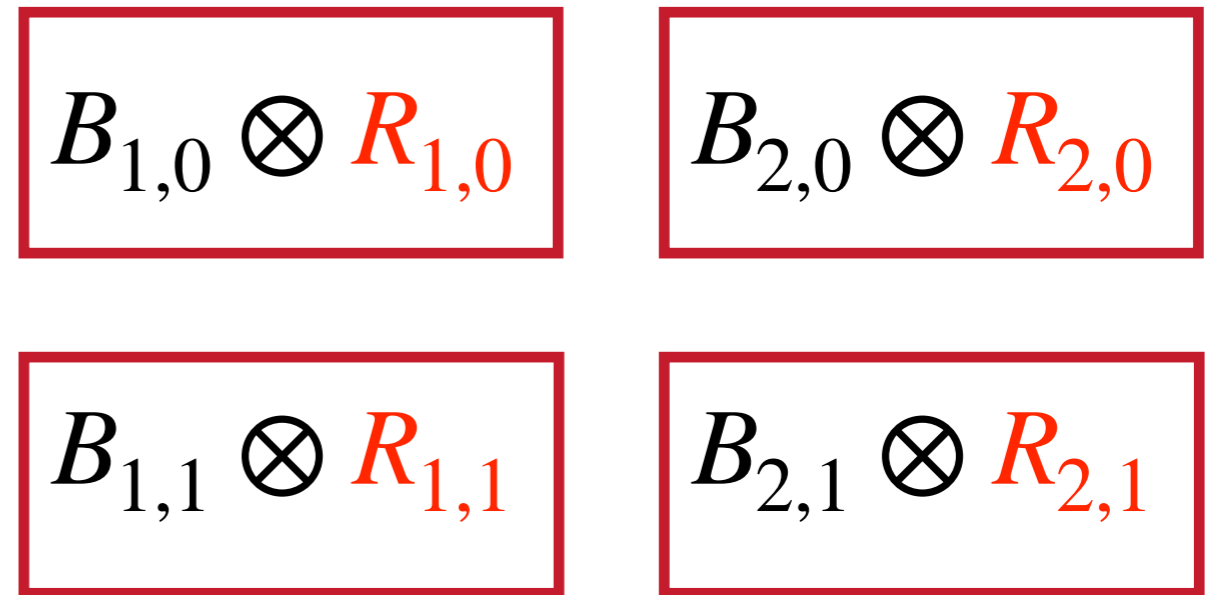
SECURE CONSTRUCTION: SUMMARY

SECURE CONSTRUCTION: SUMMARY

BP Matrices



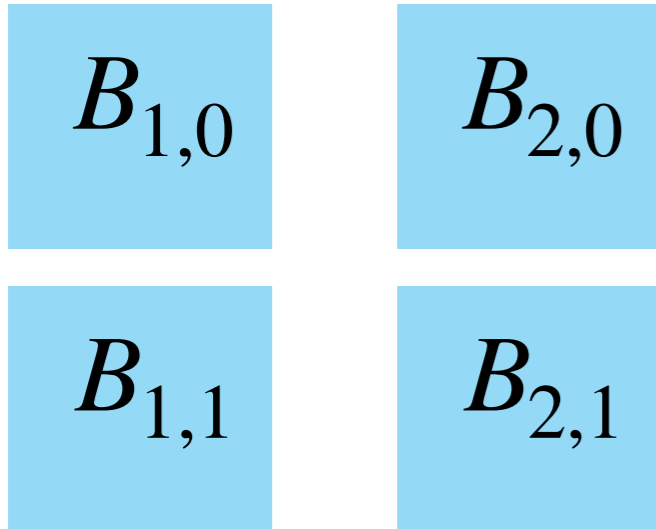
GGH15 Matrix Secrets



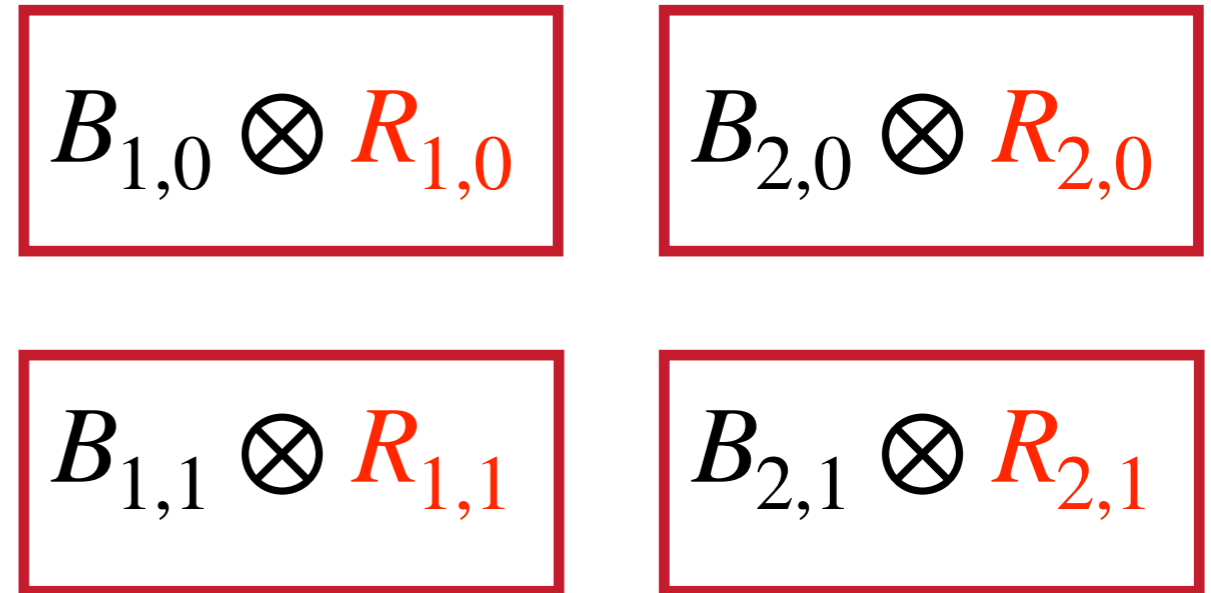
$R_{i,b}$: random low norm

SECURE CONSTRUCTION: SUMMARY

BP Matrices

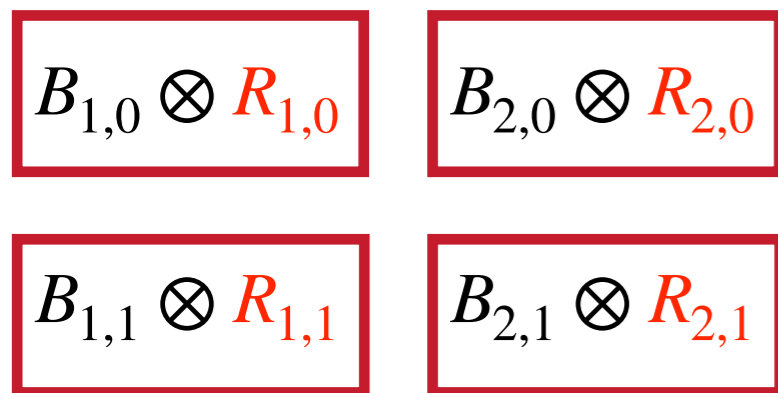


GGH15 Matrix Secrets

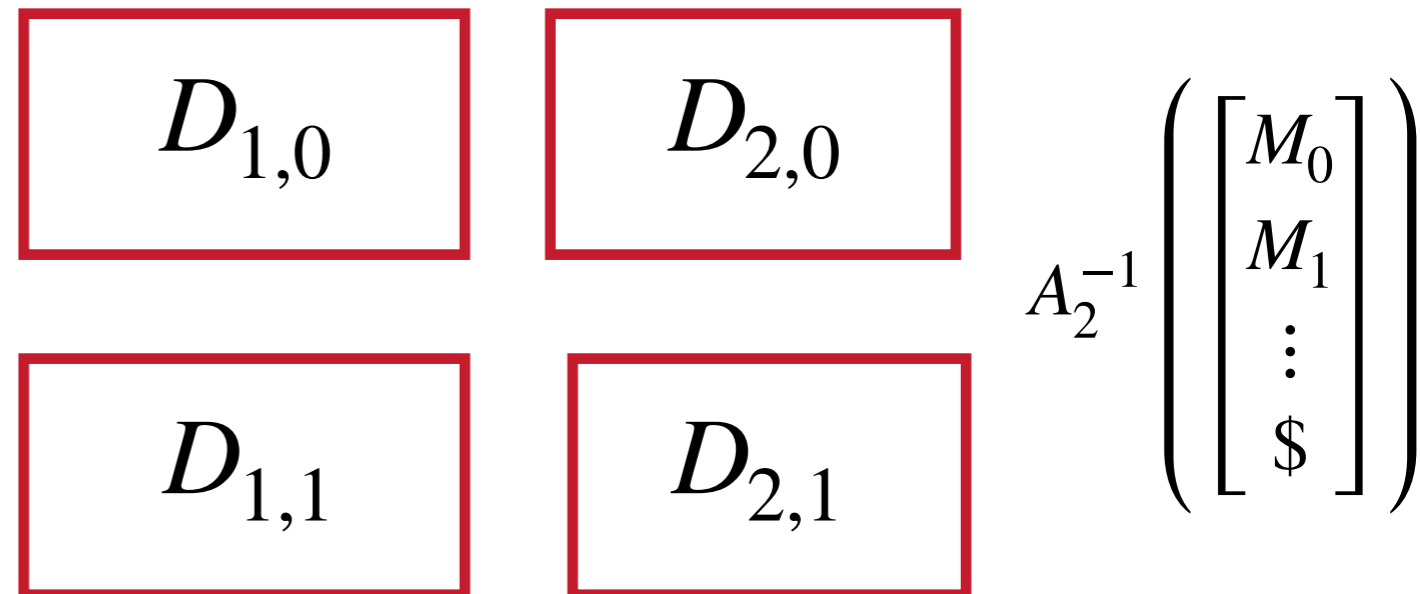


$R_{i,b}$: random low norm

GGH15 Matrix Secrets

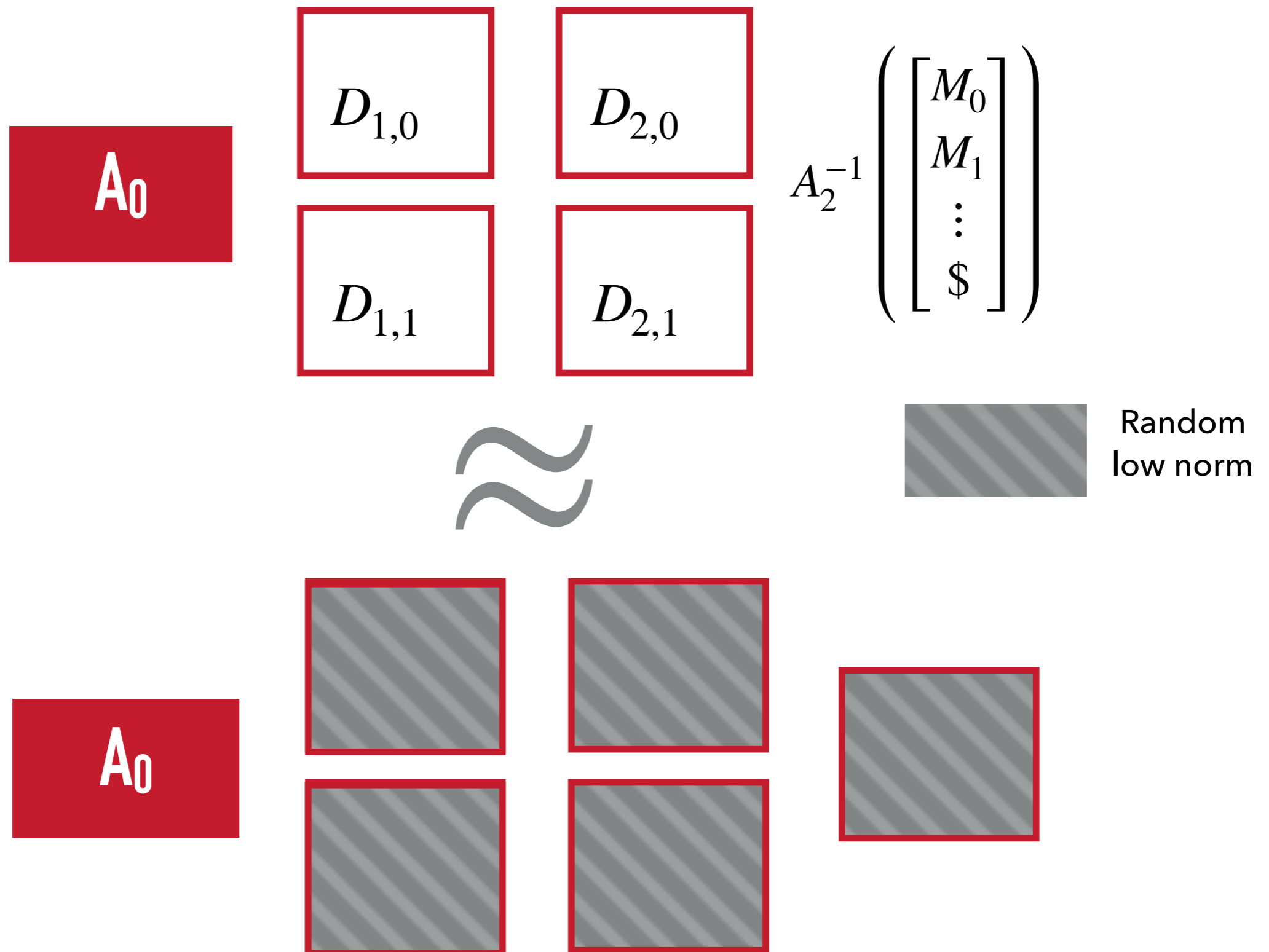


GGH15 Encodings

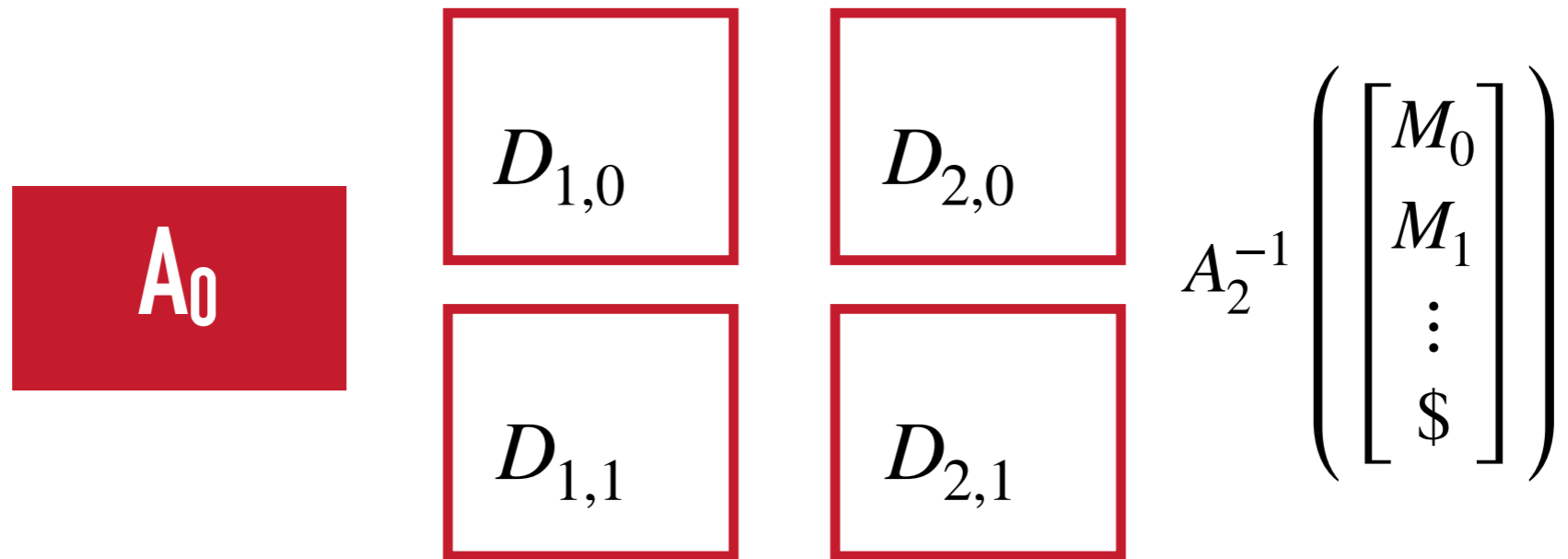


SECURITY LEMMA:

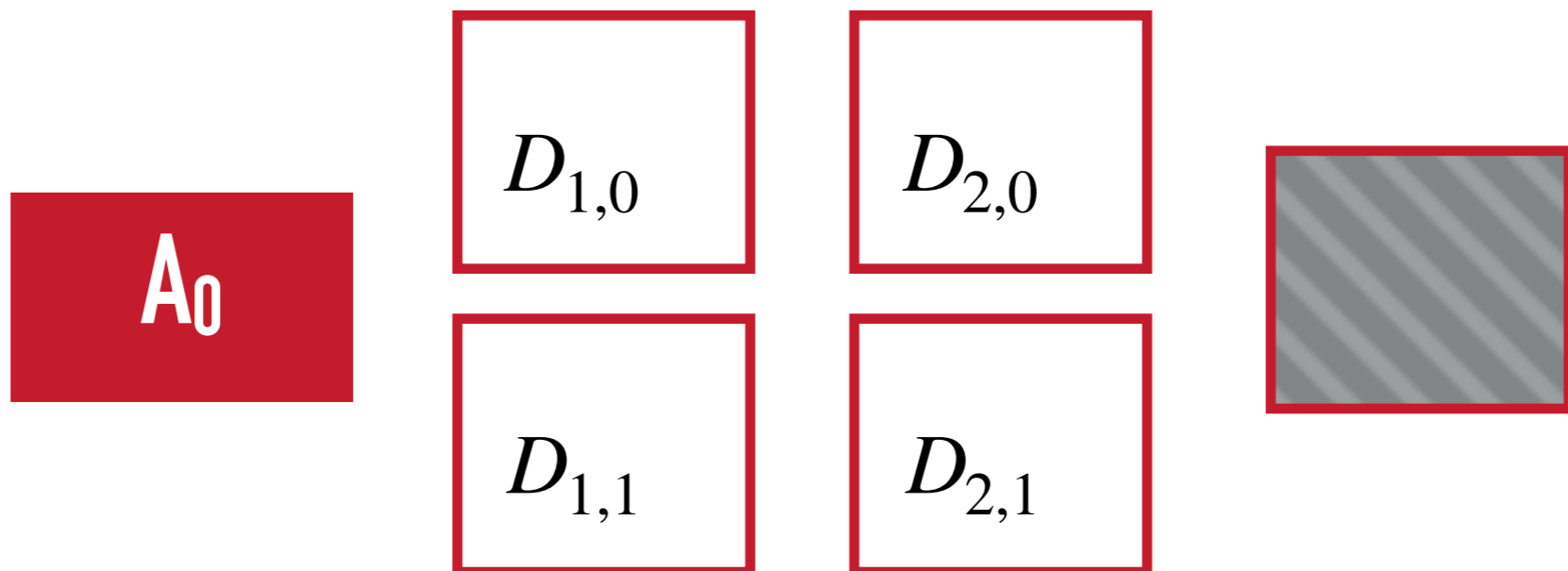
If M_0, M_1 are random, then



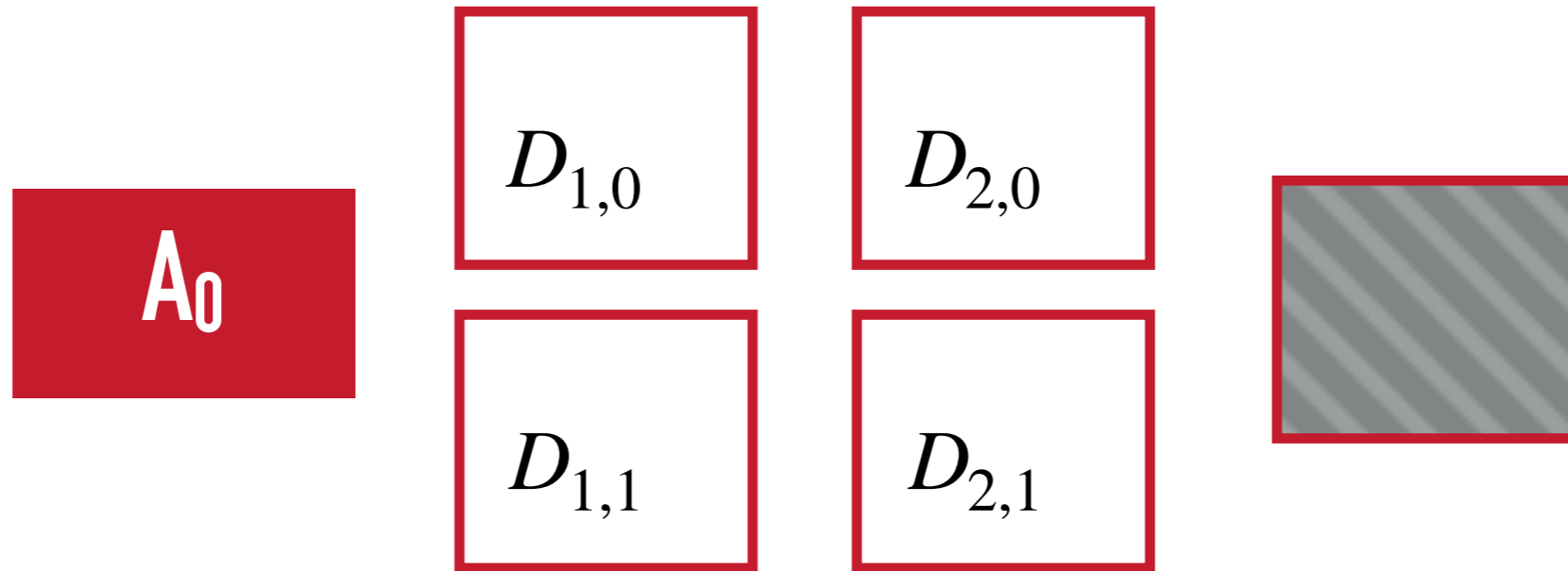
PROOF OF SECURITY LEMMA:



\approx $A^{-1}(\text{uniform}) \approx$ random low norm
[GPV 08]



PROOF OF SECURITY LEMMA:



$$D_{2,b} = A_1^{-1} \left((B_{2,b} \otimes R_{2,b}) A_2 + \text{noise} \right)$$

PROOF OF SECURITY LEMMA:

PROOF OF SECURITY LEMMA:

Permutation LWE

[Canetti, Chen 17]

P : Any Perm. Matrix

$$(A, (P \otimes R) \cdot A + \text{noise})$$

$$\approx$$

$$(A, U)$$

A, U : uniform

R : small entries

PROOF OF SECURITY LEMMA:

Permutation LWE

[Canetti, Chen 17]

P: Any Perm. Matrix

$$(A, (P \otimes R) \cdot A + \text{noise})$$

\approx

$$(A, U)$$

A, U : uniform

R : small entries

$$\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes R \right) \boxed{A}$$

PROOF OF SECURITY LEMMA:

Permutation LWE

[Canetti, Chen 17]

P: Any Perm. Matrix

$$(A, (P \otimes R) \cdot A + \text{noise})$$

$$\approx$$

$$(A, U)$$

A, U : uniform

R : small entries

$$\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes R \right) \begin{array}{|c|c|} \hline A_1 & A_2 \\ \hline A_3 & A_4 \\ \hline \end{array} = \begin{bmatrix} R \cdot A_3 & R \cdot A_4 \\ R \cdot A_1 & R \cdot A_2 \end{bmatrix}$$

PROOF OF SECURITY LEMMA:

Permutation **LWE**

[Canetti, Chen 17]

P: Any Perm. Matrix

$(A, (P \otimes R) \cdot A + \text{noise})$

\approx

(A, U)

A, U : uniform

R : small entries

$$\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes R \right) \begin{array}{|c|c|} \hline A_1 & A_2 \\ \hline A_3 & A_4 \\ \hline \end{array} = \begin{bmatrix} R \cdot A_3 & R \cdot A_4 \\ R \cdot A_1 & R \cdot A_2 \end{bmatrix}$$

PROOF OF SECURITY LEMMA:

Permutation **LWE**

[Canetti, Chen 17]

P: Any Perm. Matrix

$(A, (P \otimes R) \cdot A + \text{noise})$

\approx

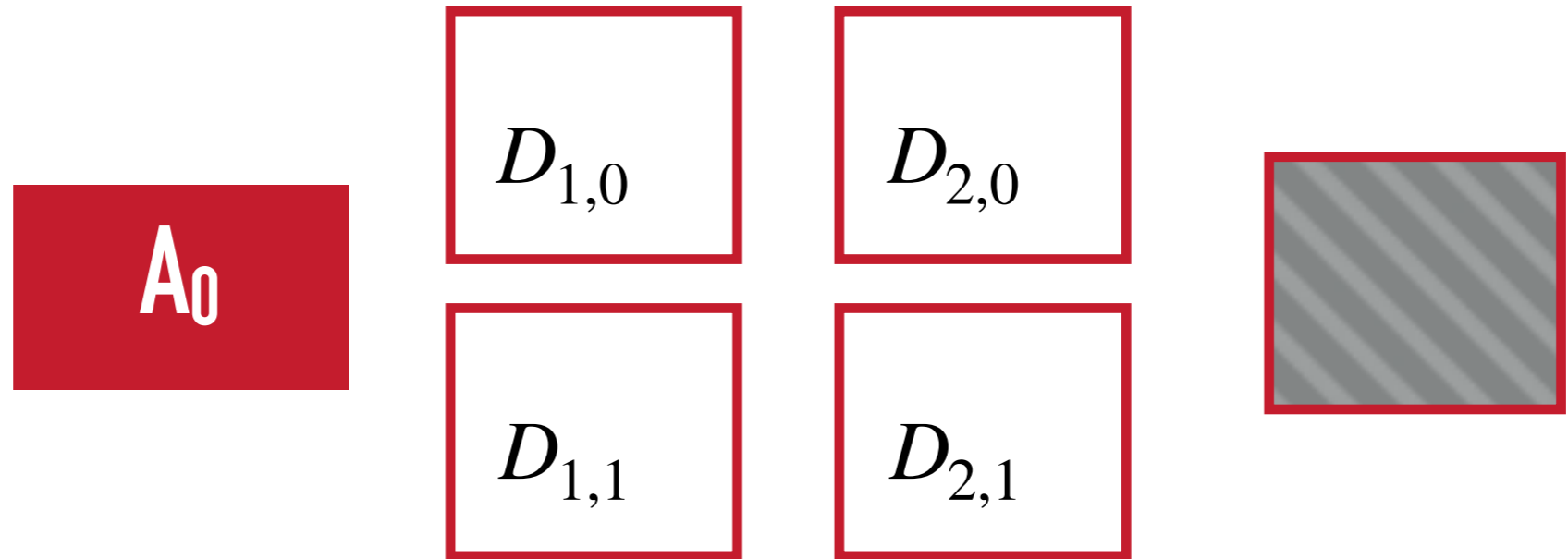
(A, U)

A, U : uniform

R : small entries

$$\left(\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \otimes R \right) \begin{array}{|c|c|} \hline A_1 & A_2 \\ \hline A_3 & A_4 \\ \hline \end{array} = \begin{bmatrix} R \cdot A_1 & R \cdot A_2 \\ R \cdot A_1 & R \cdot A_2 \end{bmatrix}$$

SECURITY LEMMA:

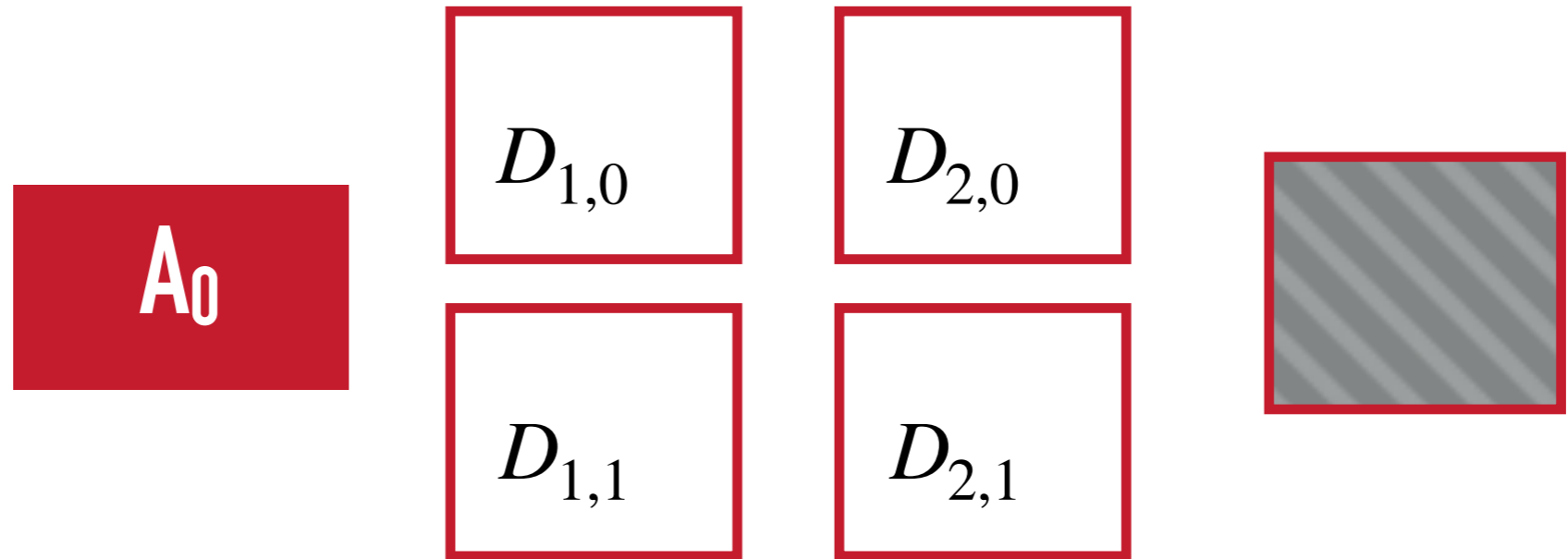


$$D_{2,b} = A_1^{-1} \left((B_{2,b} \otimes R_{2,b}) A_2 + \text{noise} \right)$$

\approx Perm. LWE

$$D_{2,b} = A_1^{-1} \left(\text{random} \right)$$

SECURITY LEMMA:



$$D_{2,b} = A_1^{-1} \left((B_{2,b} \otimes R_{2,b}) A_2 + \text{noise} \right)$$

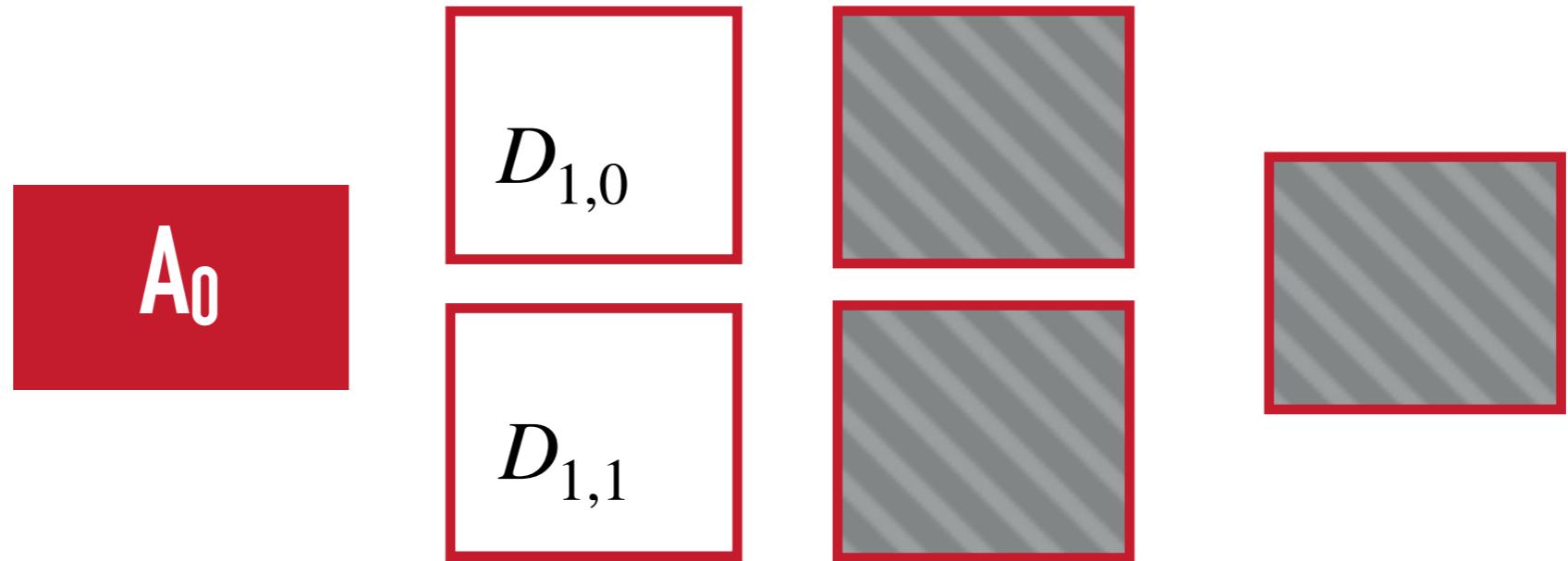
\approx Perm. LWE

$$D_{2,b} = A_1^{-1} \left(\text{random} \right)$$

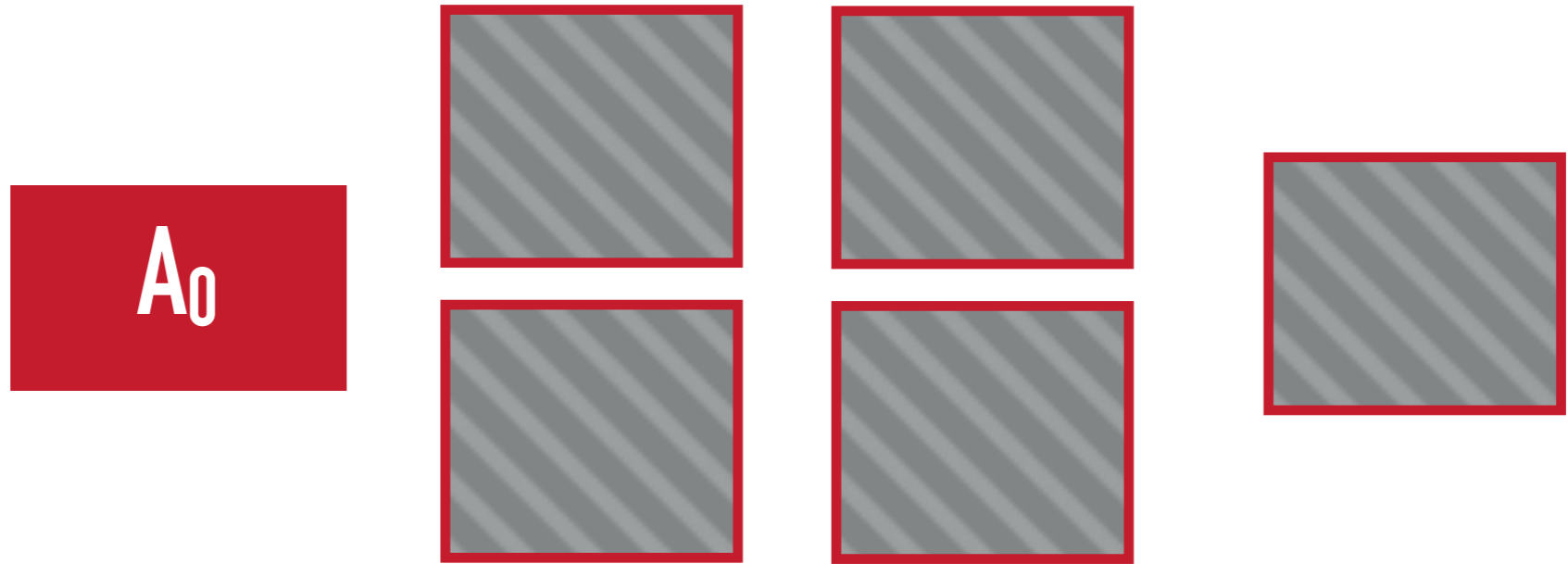
\approx [GPV 08]

low norm random

SECURITY LEMMA:



SECURITY LEMMA:



3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

For each i , 'encode' $BP_i : M_{i,0} : M_{i,1}$.

Encoding _{i} hides BP_i

Evaluation of encoding _{i} on x outputs $\approx R_x \cdot M_{i,BP_i(x)}$

Output all encodings.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

For each i , 'encode' $BP_i : M_{i,0} : M_{i,1}$. needs random $M_{i,0}, M_{i,1}$

Encoding _{i} hides BP_i

Evaluation of encoding _{i} on x outputs $\approx R_x \cdot M_{i,BP_i(x)}$

Output all encodings.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

For each i , 'encode' $BP_i : M_{i,0} : M_{i,1}$. needs random $M_{i,0}, M_{i,1}$

Encoding _{i} hides BP_i

Evaluation of encoding _{i} on x outputs $\approx R_x \cdot M_{i,BP_i(x)}$

Output all encodings.

3-STEP RECIPE FOR OBFUSCATING

$(BP_1, \dots, BP_m, \alpha)$

Choose $2m$ random matrices $M_{i,b}$ s.t. $\sum_i M_{i,\alpha_i} = 0$.

Can use Leftover Hash Lemma if
 $m > \text{poly}(\text{input length}, \text{BP length})$

For each i , 'encode' $BP_i : M_{i,0} : M_{i,1}$. needs random $M_{i,0}, M_{i,1}$

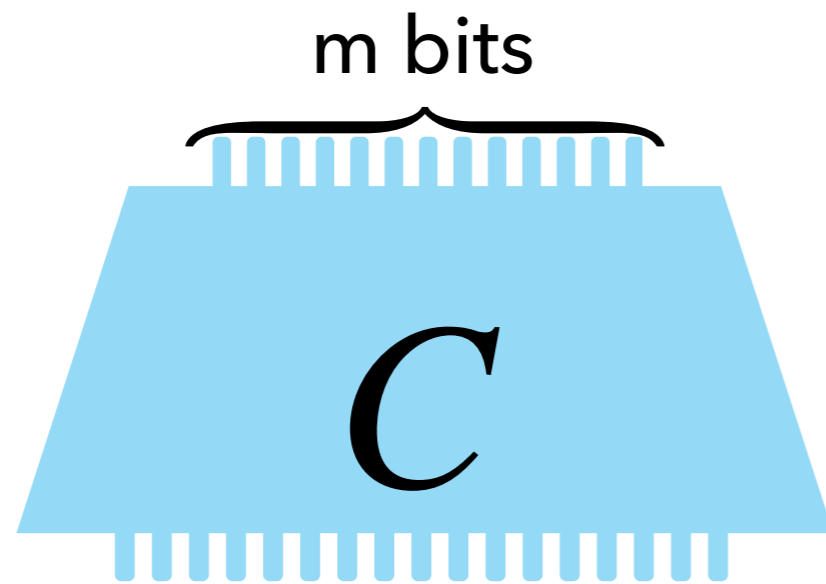
Encoding _{i} hides BP_i

Evaluation of encoding _{i} on x outputs $\approx R_x \cdot M_{i, BP_i(x)}$

Output all encodings.

Output length $>$ poly(input length, BP length) ?

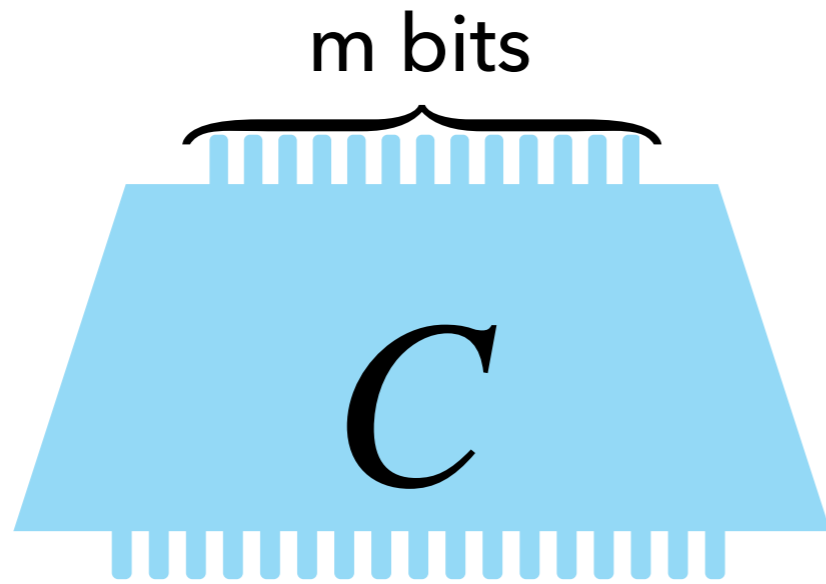
Obfuscate:



lock $\alpha \in \{0,1\}^m$

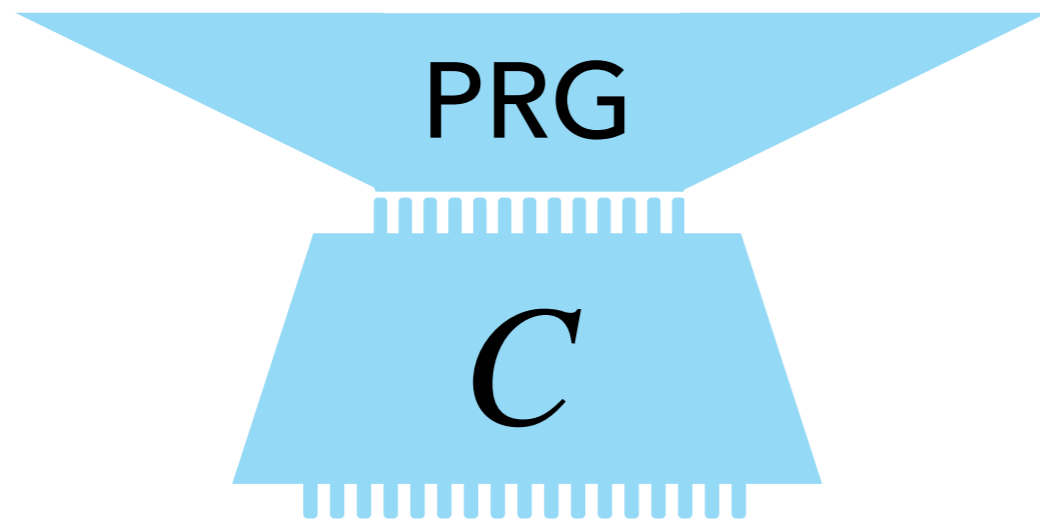
Output length $>$ poly(input length, BP length) ?

Obfuscate:



lock $\alpha \in \{0,1\}^m$

Obfuscate:



lock $\text{PRG}(\alpha) \in \{0,1\}^{m'}$

LOCKABLE OBFUSCATION: CONSTRUCTION



STEP 1: Lockable Obf. for NC¹



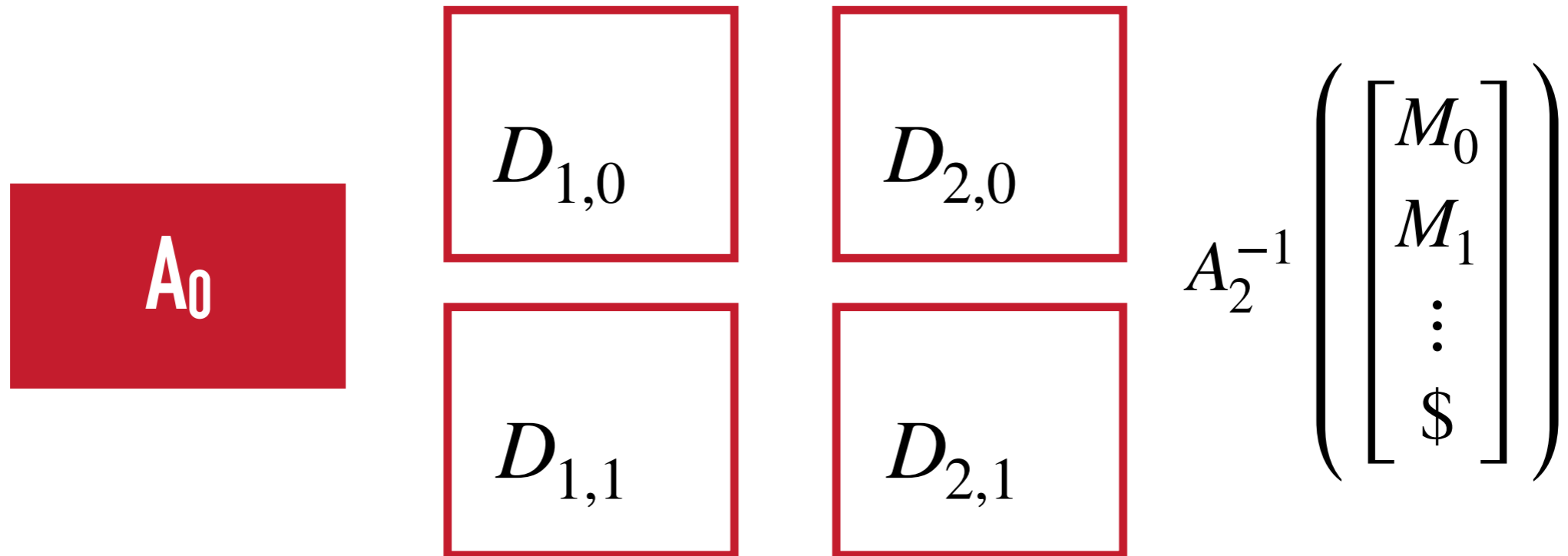
STEP 2: Bootstrapping

LO for NC¹ + FHE \rightarrow LO for P/poly

* FHE with NC¹ decryption

OBFUSCATING NON-PERMUTATION BRANCHING PROGRAMS

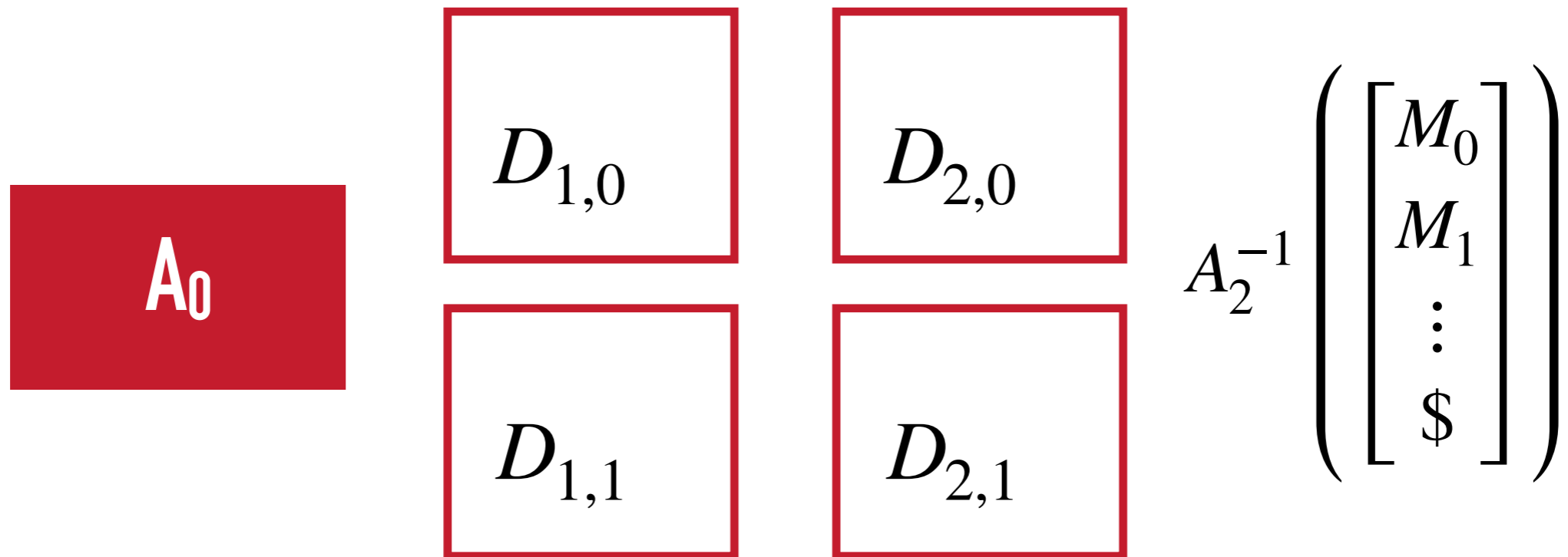
[Chen, Vaikuntanathan, Wee 18]



$$D_{i,b} = A_{i-1}^{-1} \left((B_{i,b} \otimes R_{i,b}) A_i + \text{noise} \right)$$

OBFUSCATING NON-PERMUTATION BRANCHING PROGRAMS

[Chen, Vaikuntanathan, Wee 18]

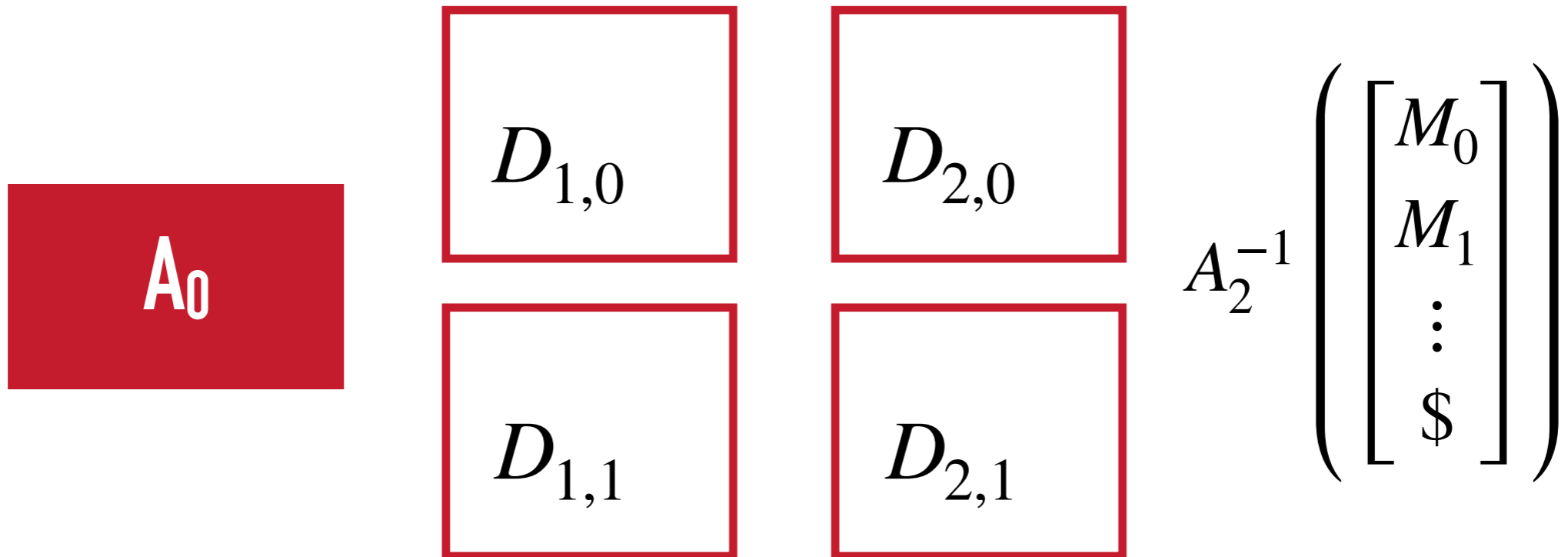


$$D_{i,b} = A_{i-1}^{-1} \left((B_{i,b} \otimes R_{i,b}) A_i + \text{noise} \right)$$

$$D_{i,b} = A_{i-1}^{-1} \left(\begin{bmatrix} B_{i,b} \\ R_{i,b} \end{bmatrix} A_i + \text{noise} \right)$$

OBFUSCATING NON-PERMUTATION BRANCHING PROGRAMS

[Chen, Vaikuntanathan, Wee 18]



$$D_{i,b} = A_{i-1}^{-1} \left((B_{i,b} \otimes R_{i,b}) A_i + \text{noise} \right) \text{ RISHAB'S TALK ?}$$

$$D_{i,b} = A_{i-1}^{-1} \left(\begin{bmatrix} B_{i,b} \\ R_{i,b} \end{bmatrix} A_i + \text{noise} \right) \approx \text{[Blurred Matrix]}$$

LOCKABLE OBFUSCATION

LEARNING WITH ERRORS



Upgrading security

- Making encryption anonymous
- Witness enc. \rightarrow null IO
- Private secure sketches

Replacing IO with LO

- Circular security separations
- Random oracle uninstantiability results

LOCKABLE OBFUSCATION

LEARNING WITH ERRORS

Upgrading security

- Making encryption anonymous
- Witness enc. -> null IO
- Private secure sketches
- Mixed Functional Enc [CVWW18]

Replacing IO with LO

- Circular security separations
- Random oracle uninstantiability results
- CCA vs FE upgradability [BKSW18]

LOCKABLE OBFUSCATION

Zero knowledge

- 3 round weak ZK [BKP19]
- Constant round post quantum ZK [BS19]

LEARNING WITH ERRORS

OPEN PROBLEMS

OPEN PROBLEMS

- ▶ More applications ?

OPEN PROBLEMS

- ▶ More applications ?
- ▶ Allowing some auxiliary information on lock/circuit/
message?

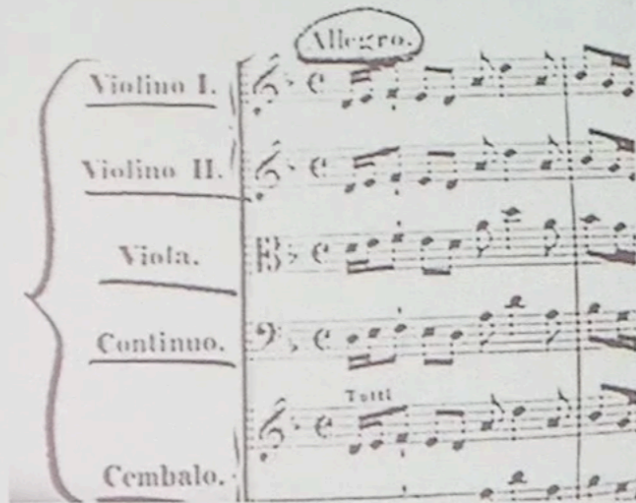
OPEN PROBLEMS

- ▶ More applications ?
- ▶ Allowing some auxiliary information on lock/circuit/message?
- ▶ Obfuscation for other evasive circuits?



[GGH15] Via a different view of the FHE scheme of Gentry, Sahai, Waters 13

Different *motives* / views of GGH15



- The arithmetic operations are just matrix operations in $\mathbb{Z}_q^{n \times m}$:

$$\text{neg}(pp, D) := -D, \text{ add}(pp, D, D') := D + D', \text{ and } \text{mult}(pp, D, D') := D \cdot D'.$$
- To see that negation and addition maintain the right structure, let $D, D' \in \mathbb{Z}_q^{n \times m}$ be two encodings relative to the same path $u \rightsquigarrow v$. Namely $D \cdot A_u = A_v \cdot S + E$ and $D' \cdot A_u = A_v \cdot S' + E'$, with the matrices D, D', E, E', S, S' all small. Then we have

$$-D \cdot A_u = A_v \cdot (-S) + (-E),$$

$$\text{and } (D + D') \cdot A_u = (A_v \cdot S + E) + (A_v \cdot S' + E') = A_v \cdot (S + S') + (E + E'),$$
- and all the matrices $-D, -S, -E, D + D', S + S', E + E'$ are still small. For multiplication, consider encodings D, D' relative to paths $v \rightsquigarrow u$ and $u \rightsquigarrow v$, respectively; then we have

$$(D \cdot D') \cdot A_u = D \cdot (A_v \cdot S' + E')$$

$$= (A_v \cdot S + E) \cdot S' + D \cdot E' = A_v \cdot (S \cdot S') + \underbrace{(E \cdot S' + D \cdot E')}_{E''},$$
- and the matrices $D \cdot D', S \cdot S'$, and E'' are still small.
- Of course, the matrices D, S, E all grow with arithmetic operations, but our parameter-choice ensures that for any encoding relative to any path in the graph $u \rightsquigarrow v$ (of length $\leq d$) we have $D \cdot A_u = A_v \cdot S + E$ where E is still small, specifically $\|E\| < q^{3/4} \leq q/2^{d+1}$.
- ZeroTest(pp, D). Given an encoding D relative to path $u \rightsquigarrow v$ and the matrix A_u , our zero-test procedure outputs 1 if and only if $\|D \cdot A_u\| < q/2^{d+1}$.

[Alamati, Peikert 16],
[Koppula, Waters 16],
[Goyal, Koppula, Waters 17]
“cascaded products” or
“telescoping cancelation”,
motivated by showing circular security counterexamples.

[Canetti, Chen 17]

GGH15 captures two lattice-based PRFs

[Chen, Vaikuntanathan, Wee 18]

A generalization of Kilian randomization

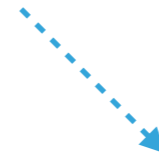
Yilei's talk on 'Lattices, Multilinear Maps and Program Obfuscation'

<https://simons.berkeley.edu/talks/advanced-lattice-based-cryptography-fhe-abe-etc-0>

LOCKABLE OBFUSCATION: BEHIND THE SCENES

LOCKABLE OBFUSCATION: BEHIND THE SCENES

GSW13 Homomorphic Enc

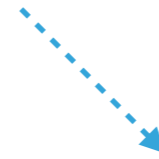


2-circular security separations
[BHW15]

GGH15 Multilinear maps

LOCKABLE OBFUSCATION: BEHIND THE SCENES

GSW13 Homomorphic Enc



2-circular security separations
[BHW15]

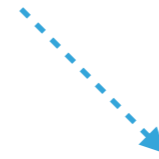
GGH15 Multilinear maps



n-circular security separations
[AP16, KW16]

LOCKABLE OBFUSCATION: BEHIND THE SCENES

GSW13 Homomorphic Enc



2-circular security separations
[BHW15]

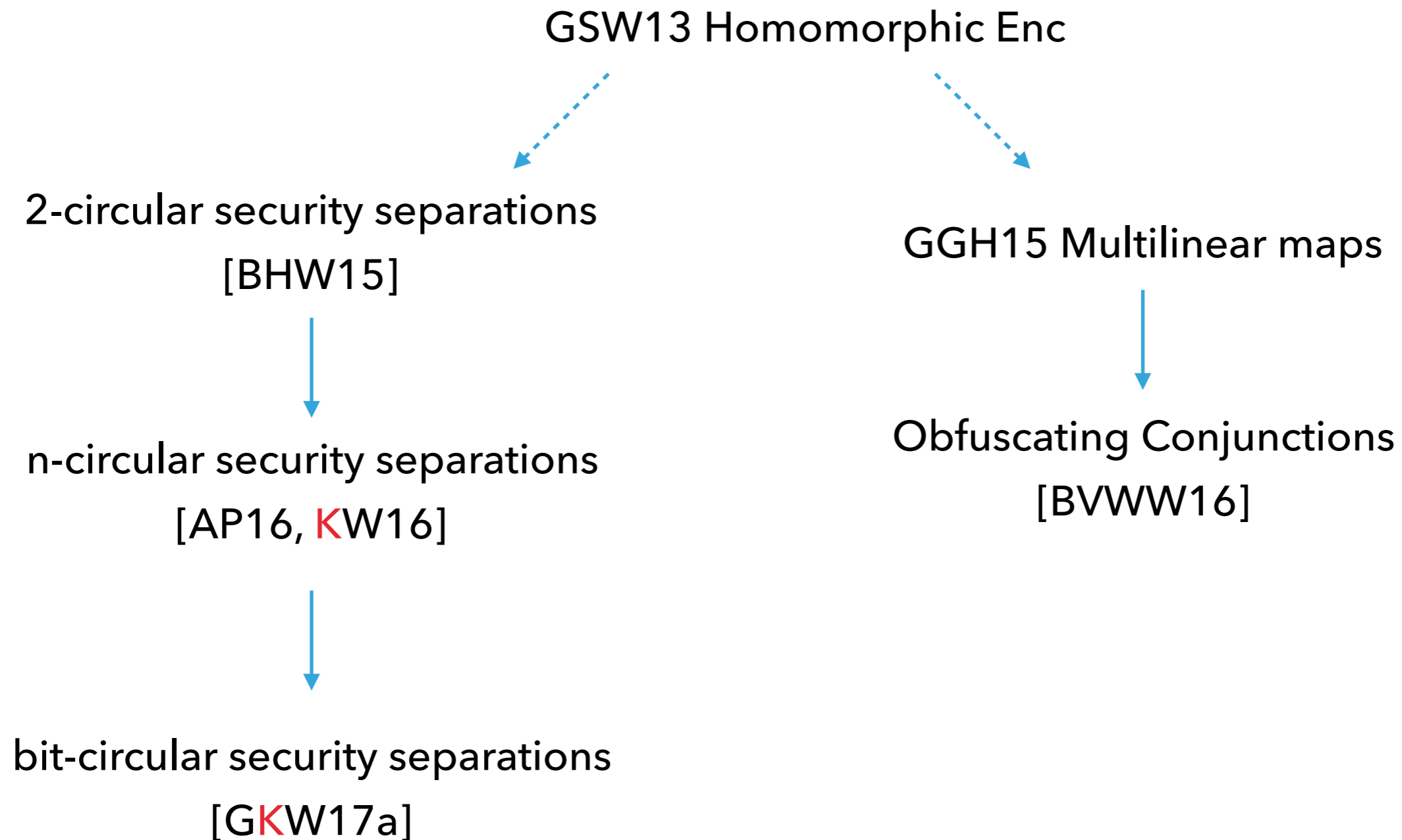
GGH15 Multilinear maps



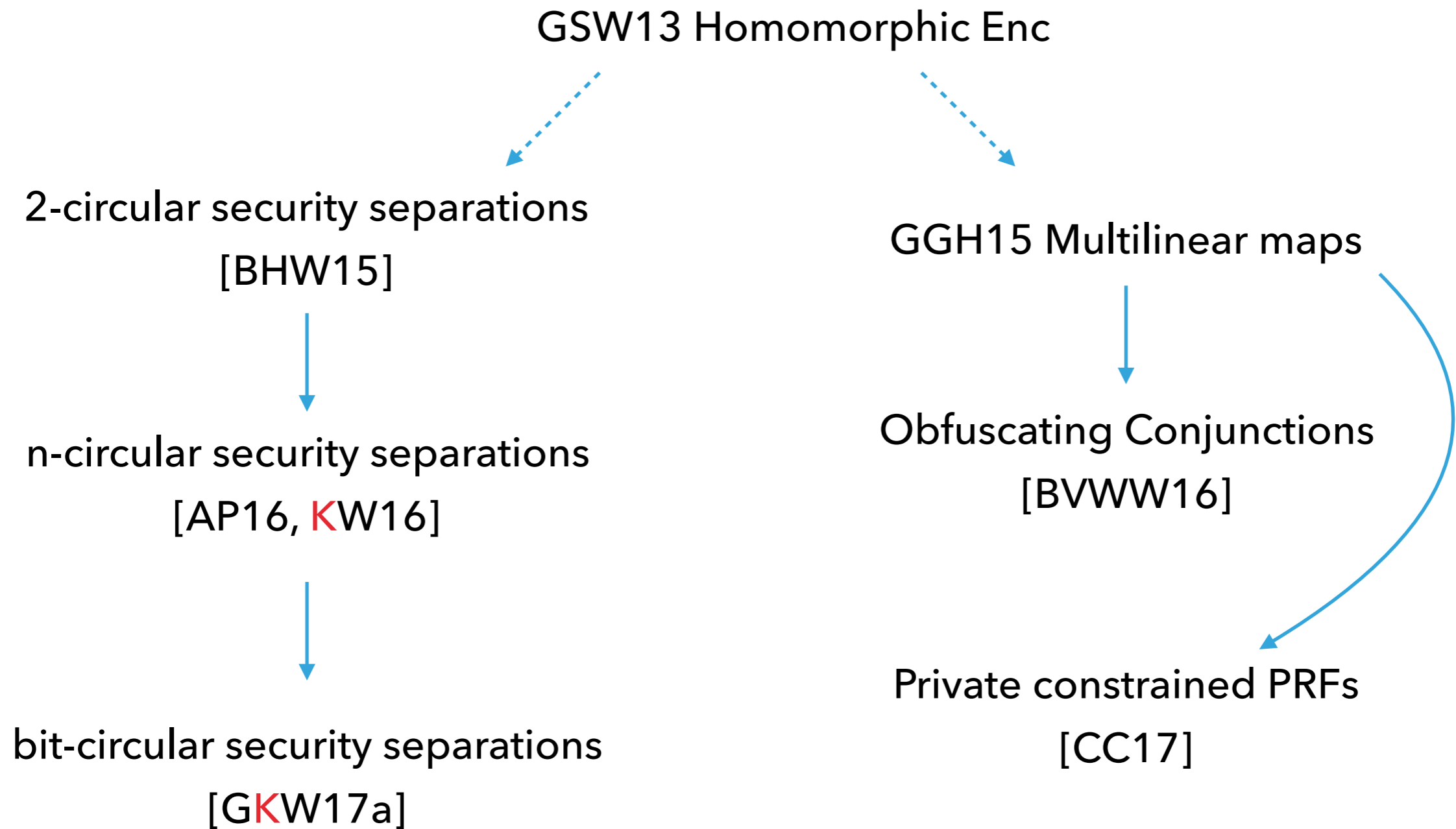
n-circular security separations
[AP16, KW16]

Obfuscating Conjunctions
[BVWW16]

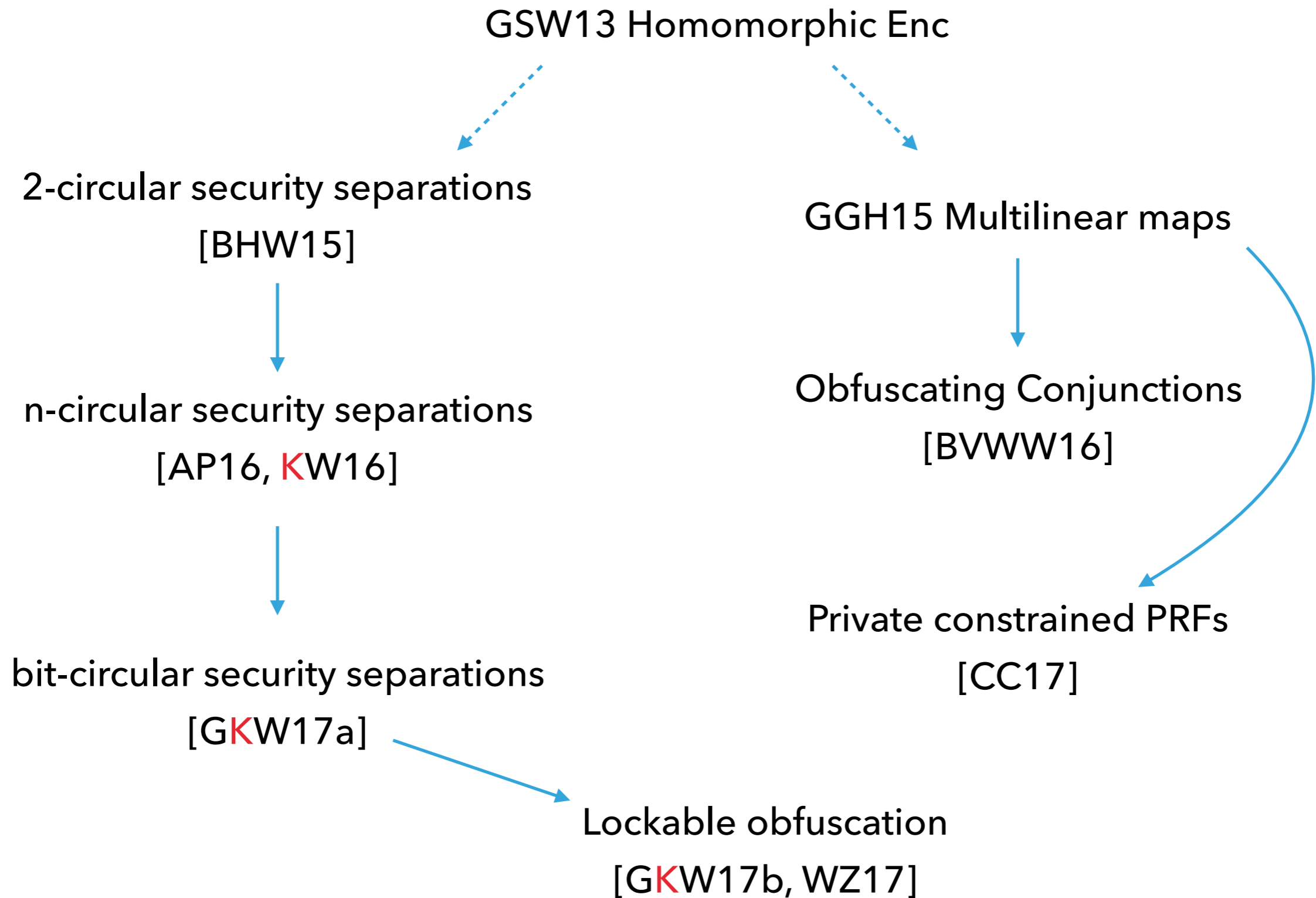
LOCKABLE OBFUSCATION: BEHIND THE SCENES



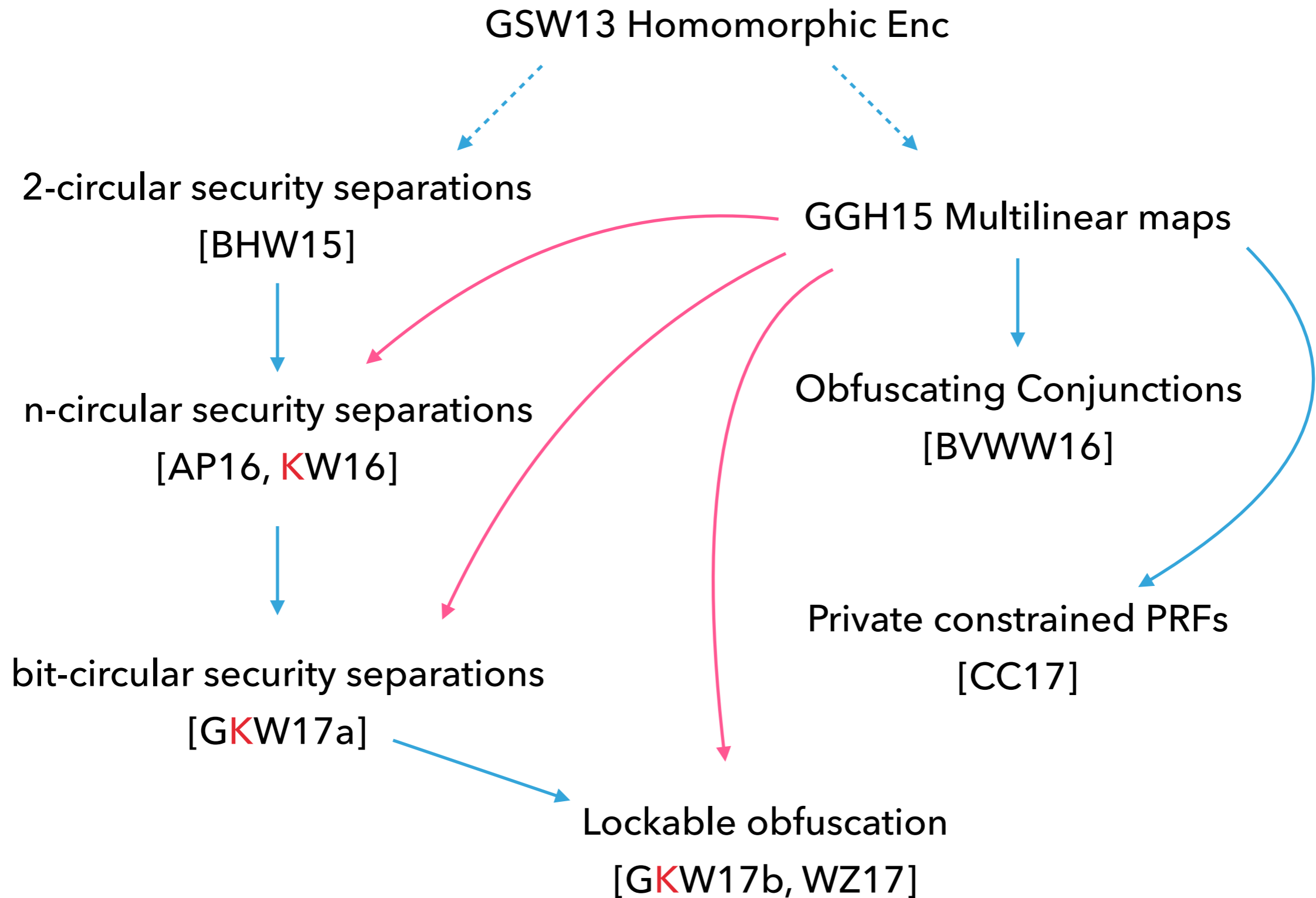
LOCKABLE OBFUSCATION: BEHIND THE SCENES



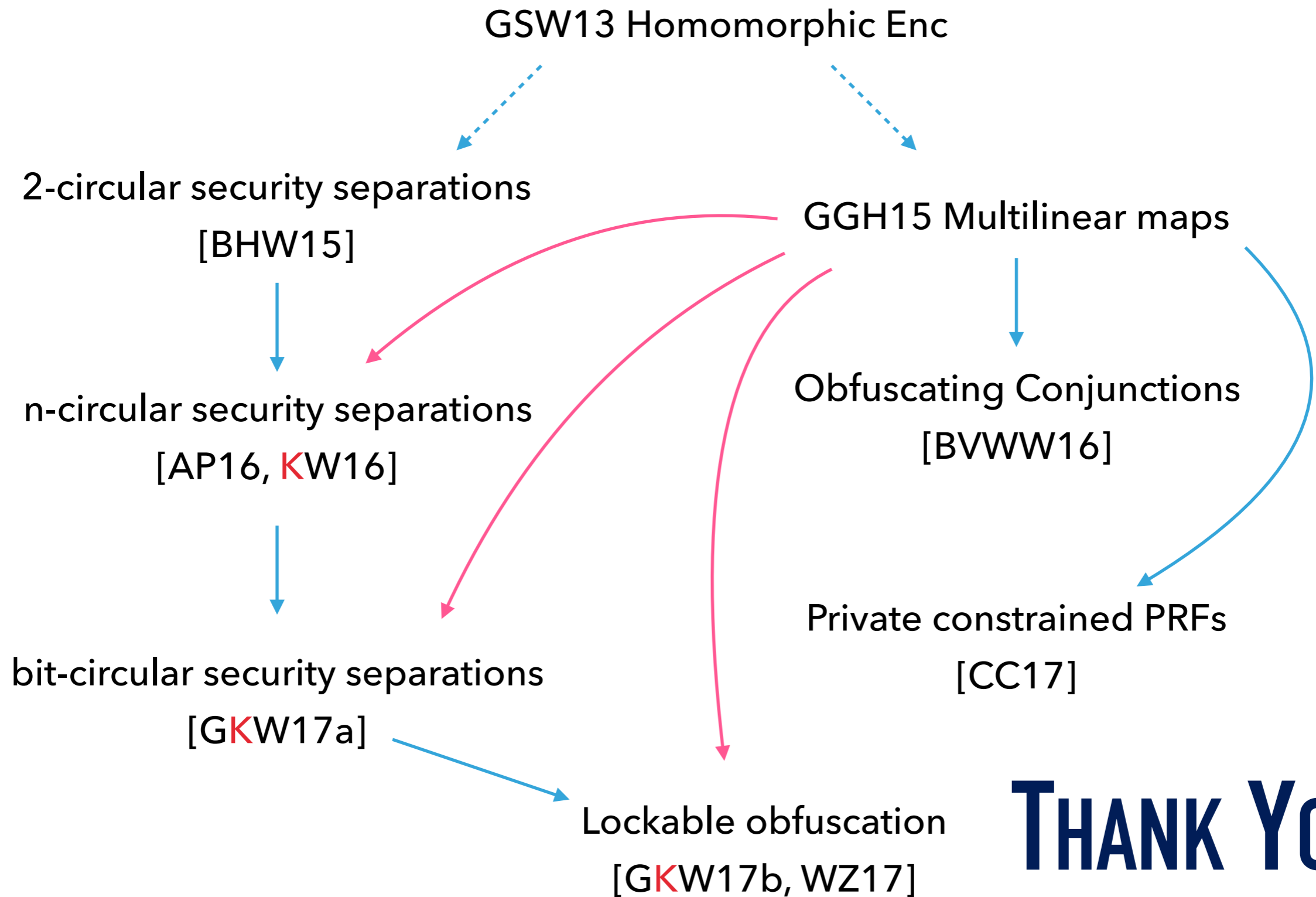
LOCKABLE OBFUSCATION: BEHIND THE SCENES



LOCKABLE OBFUSCATION: BEHIND THE SCENES



LOCKABLE OBFUSCATION: BEHIND THE SCENES



THANK YOU!!