THE MMAP STRIKES BACK:

Obfuscation and New Multilinear Maps Immune to CLT13 Zeroizing Attacks

Fermi Ma and Mark Zhandry

RETURN OF GGH15:

Provable Security Against Zeroizing Attacks

James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry

Multilinear Maps

[BS03,GGH13,CLT13,GGH15]

Levels: $1, \dots, \kappa$, Plaintext Ring R

Secret

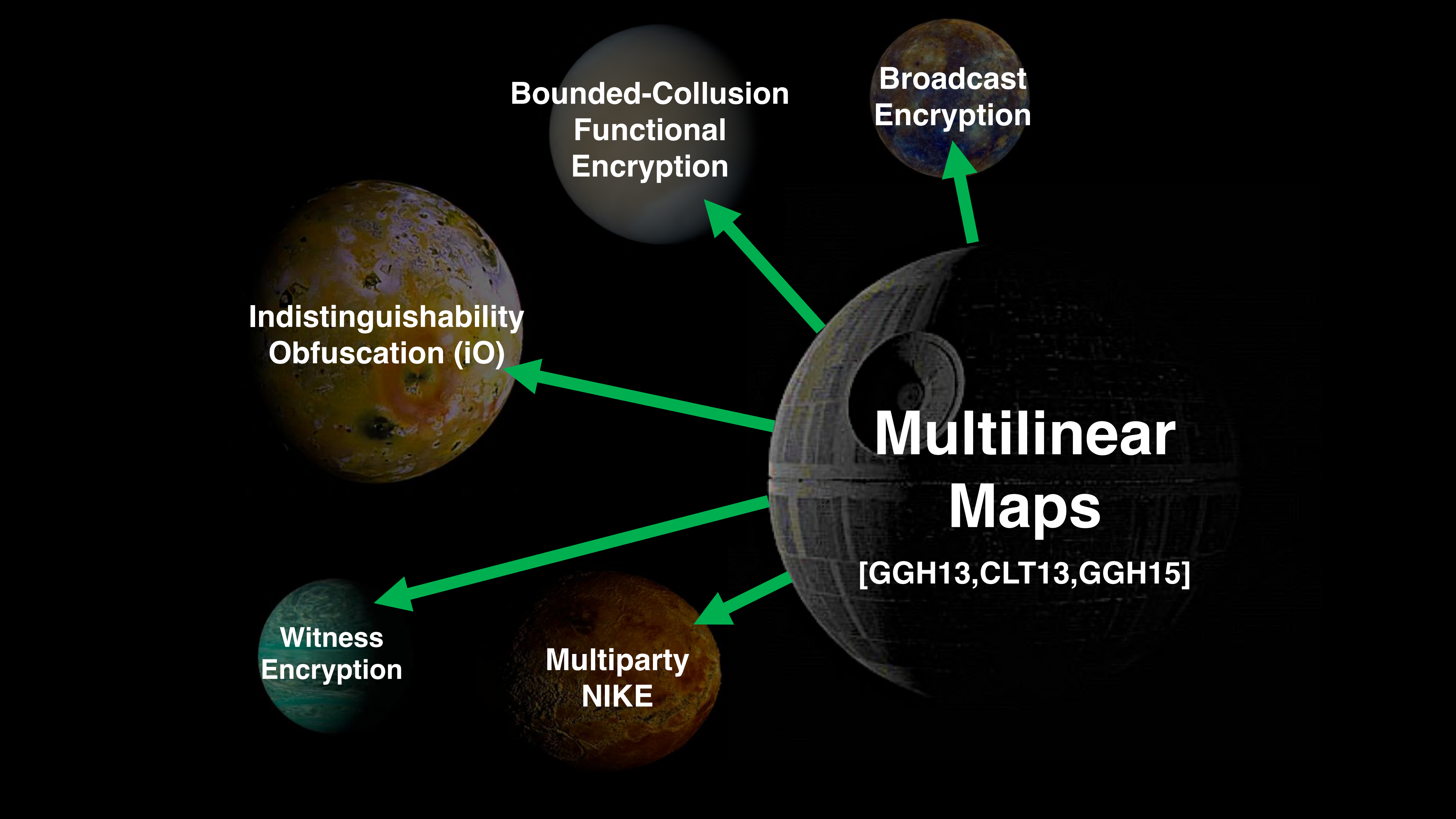
$a \in R, i \in \{1, \dots, \kappa\}$ —Encode→ $[a]_i$

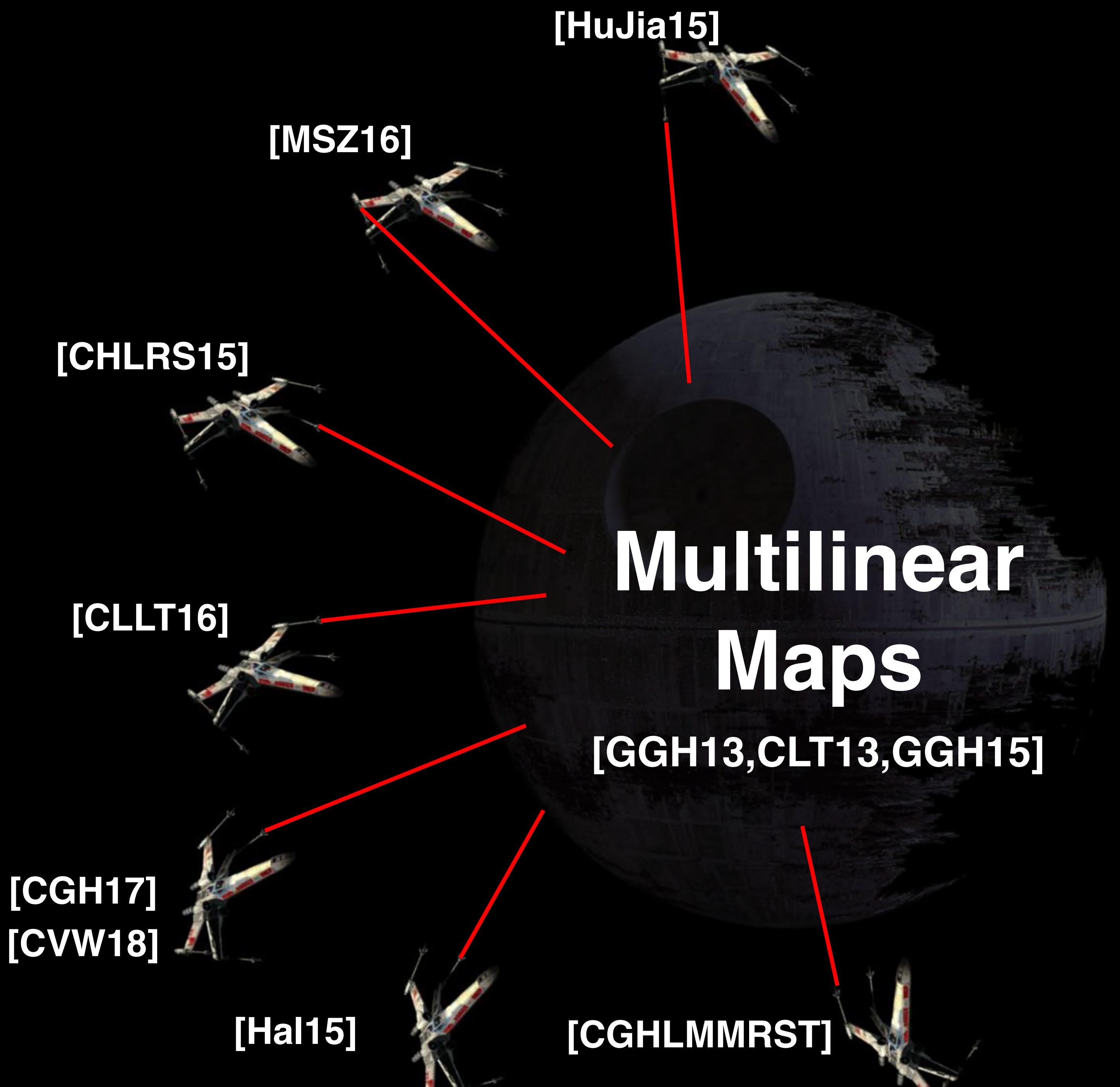
Public

$[a]_i + [b]_i \longrightarrow [a+b]_i$

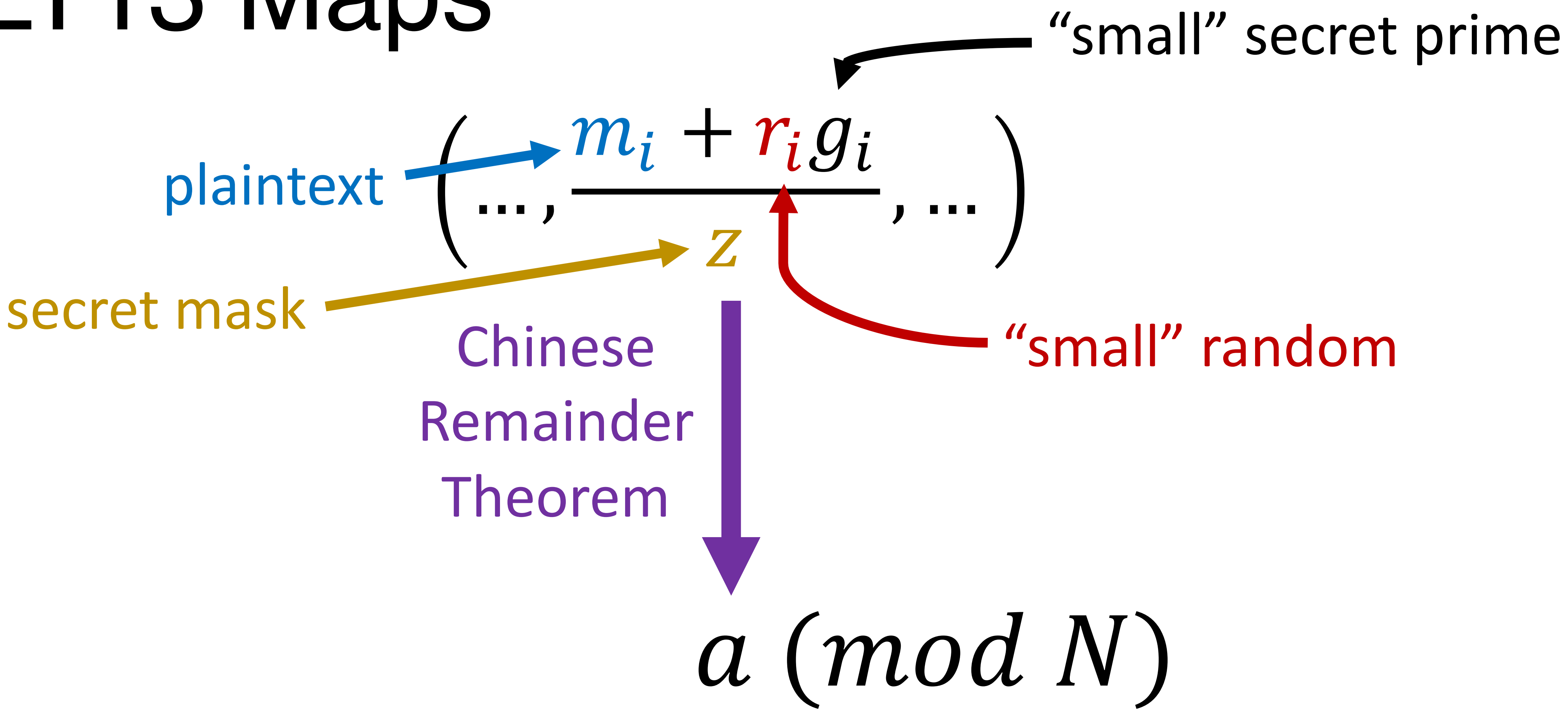
$[a]_i \times [b]_j \longrightarrow [ab]_{i+j}$

$[a]_\kappa$ —Zero-Test→ Yes/No

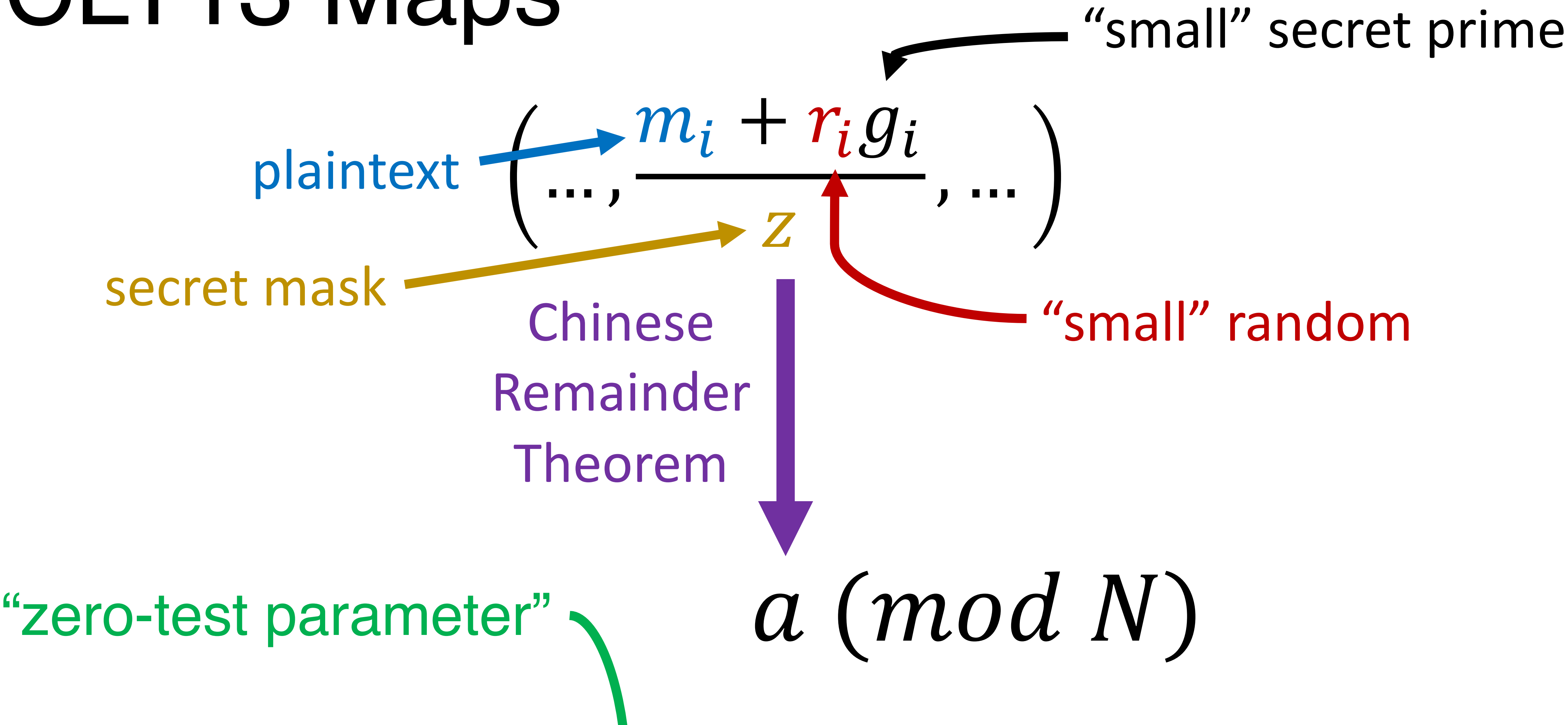

Bounded-Collusion Functional Encryption
Broadcast Encryption
Indistinguishability Obfuscation (iO)
Multilinear Maps
[GGH13,CLT13,GGH15]
Witness Encryption
Multiparty NIKE


[HuJia15]
[MSZ16]
[CHLRS15]
Multilinear
Maps
[GGH13,CLT13,GGH15]
[CLLT16]
[CGH17]
[CVW18]
[Hal15]
[CGHLMMRST]

CLT13 Maps


"small" secret prime
plaintext
$\left(\ldots, \frac{m_i + r_i g_i}{z}, \ldots \right)$
secret mask
"small" random
Chinese Remainder Theorem
$a \pmod N$

CLT13 Maps



Zero Test: $p_{ZT} \cdot a \pmod N \ll N?$

Zeroizing Attack on CLT13 [CHLRS15]

Setting

$$b^{(1)} = \left(\ldots, \frac{B_i^{(1)}}{z}, \ldots\right), b^{(2)} = \left(\ldots, \frac{B_i^{(2)}}{z}, \ldots\right)$$

$$a^{(1)}, \ldots, a^{(n)}, c^{(1)}, \ldots, c^{(n)}$$

Where each $a^{(i)} \cdot b^{(j)} \cdot c^{(k)}$ is encoding of zero

Zeroizing Attack on CLT13 [CHLRS15]

Setting

$$b^{(1)} = \left(\dots, \frac{B_i^{(1)}}{z}, \dots \right), b^{(2)} = \left(\dots, \frac{B_i^{(2)}}{z}, \dots \right)$$

$$a^{(1)}, \dots, a^{(n)}, c^{(1)}, \dots, c^{(n)}$$

Where each $a^{(i)} \cdot b^{(j)} \cdot c^{(k)}$ is encoding of zero

Attack Steps

1. Form matrices $\boldsymbol{W}, \boldsymbol{Y}$ by zero-testing each $a^{(i)} \cdot b^{(j)} \cdot c^{(k)}$.
2. Compute eigenvalues of $\boldsymbol{W}^{-1}\boldsymbol{Y}$:

$$\dots, \frac{B_i^{(2)}}{B_i^{(1)}}, \dots$$

3. GCD on eigenvalues reveal secret parameters.

Observation: CHLRS15 computes char-poly(M) where entries of M are zero-test results. Roots are numerators $a_i + r_i g_i$.

Attack Steps

1. Form matrices $\boldsymbol{W}, \boldsymbol{Y}$ by zero-testing each $a^{(i)} \cdot b^{(j)} \cdot c^{(k)}$.
2. Compute eigenvalues of $\boldsymbol{W}^{-1}\boldsymbol{Y}$:

$$\ldots, \frac{B_i^{(2)}}{B_i^{(1)}}, \ldots$$

3. GCD on eigenvalues reveal secret parameters.

Observation: CHLRS15 computes char-poly(M) where entries of M are zero-test results. Roots are numerators $a_i + r_i g_i$.

Solving polynomial for CLT13 numerators is ***only known attack strategy***. [See also: CGHLMMRST15, CLLT16]

Attack Steps

1. Form matrices $\boldsymbol{W}, \boldsymbol{Y}$ by zero-testing each $a^{(i)} \cdot b^{(j)} \cdot c^{(k)}$.
2. Compute eigenvalues of $\boldsymbol{W}^{-1}\boldsymbol{Y}$:

$$\ldots, \frac{B_i^{(2)}}{B_i^{(1)}}, \ldots$$

3. GCD on eigenvalues reveal secret parameters.

Observation: CHLRS15 computes char-poly(M) where entries of M are zero-test results. Roots are numerators $a_i + r_i g_i$.

Solving polynomial for CLT13 numerators is ***only known attack strategy***. [See also: CGHLMMRST15, CLLT16]

zero-test results

CLT13 numerators

- $Q(\{t_j\}_j, \{s_i\}_i) = 0$
- $Q(\{t_j\}_j, \{\mathbf{S}_i\}_i) \neq 0$

formal variable

Step 1: Weak Model

(inspired by MSZ16 and GMMSSZ16)

Extend Generic Model to ***allow adversary*** to perform a zeroizing attack.

Step 1: Weak Model

(inspired by MSZ16 and GMMSSZ16)

Extend Generic Model to ***allow adversary*** to perform a zeroizing attack.

$\{h^{(i)}\}_i$

$p(\{h^{(i)}\}_i)$

"zero" / "non-zero"

Generic Model

Plaintexts $m^{(1)}, \ldots, m^{(k)}$.

Handles $h^{(1)}, \ldots, h^{(k)}$.

Zero Test Queries Return "zero" if

- $p(\{m^{(i)}\}_i) = 0$
- degree κ.

Step 1: Weak Model

(inspired by MSZ16 and GMMSSZ16)

Extend Generic Model to ***allow adversary*** to perform a zeroizing attack.

$\{h^{(i)}\}_i$

$p(\{h^{(i)}\}_i)$

"zero" / "non-zero"

$Q(\{\boldsymbol{T}_j\}_j, \{\boldsymbol{S}_i\}_i)$

"successful" / "unsuccessful"

Generic Model + Zeroizing Attacks

Plaintexts $m^{(1)}, \ldots, m^{(k)}$.

Handles $h^{(1)}, \ldots, h^{(k)}$.

Zero Test Queries Return "zero" if

- $p(\{m^{(i)}\}_i) = 0$
- degree κ.

New: Return *post-zero-test* handle "$\boldsymbol{T}$" if zero.

Post Zero Test Return "WIN" if

- $Q(\{t_j\}_j, \{s_i\}_i) = 0$
- $Q(\{t_j\}_j, \{\boldsymbol{S}_i\}_i) \not\equiv 0$

Step 1: Weak Model

Extend Generic Model to ***allow adversary*** to perform a zeroizing attack.

Step 2: Annihilation Theorem

If you can perform a zeroizing attack, you can annihilate “zero-test polynomials”.

Step 1: Weak Model

Extend Generic Model to ***allow adversary*** to perform a zeroizing attack.

Step 2: Annihilation Theorem

If you can perform a zeroizing attack, you can annihilate “zero-test polynomials”.

If x, y are CLT13 encodings, and $x^2 + xy$ is a top-level zero, **the zero-test polynomial** is the formal polynomial $\boldsymbol{x}^2 + \boldsymbol{x}\boldsymbol{y}$.

Theorem: If  can mount a zeroizing attack,  can “cancel out” linearly independent zero-test polynomials.

Step 1: Weak Model

Extend Generic Model to ***allow adversary*** to perform a zeroizing attack.

Step 2: Annihilation Theorem

If you can perform a zeroizing attack, you can annihilate "zero-test polynomials".

Step 3: Zeroizing-Immune Schemes

Obtain constructions where annihilating zero-test polynomials is hard.

Step 1: Weak Model

Extend Generic Model to ***allow adversary*** to perform a zeroizing attack.

Step 2: Annihilation Theorem

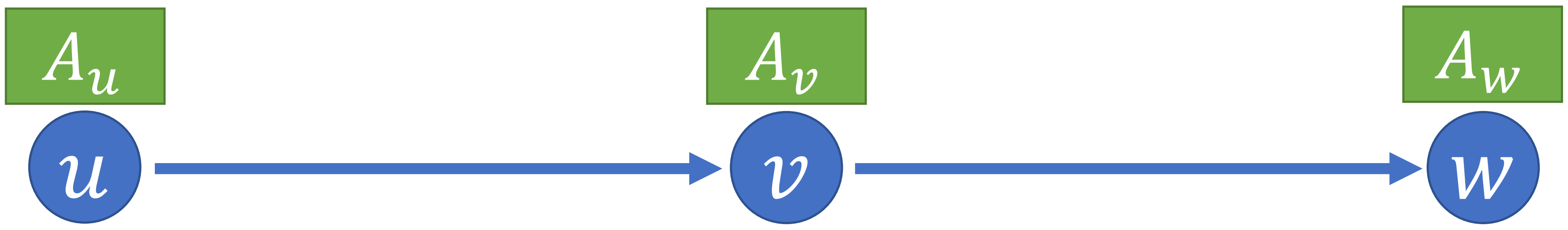
If you can perform a zeroizing attack, you can annihilate "zero-test polynomials".

Step 3: Zeroizing-Immune Schemes

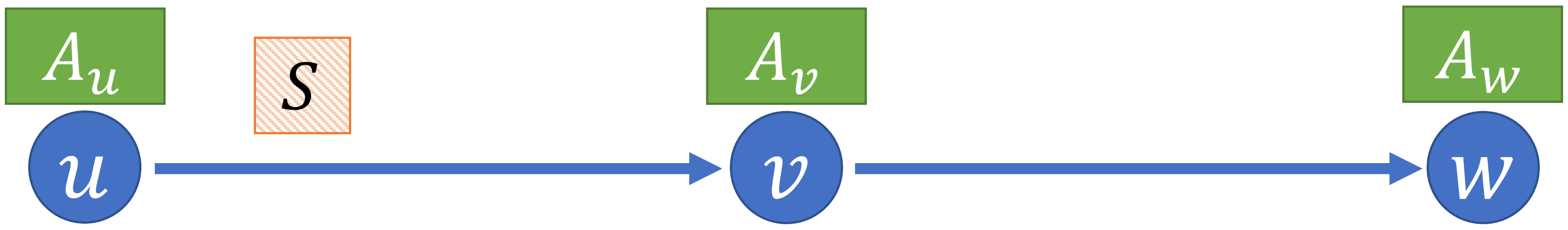
Obtain constructions where annihilating zero-test polynomials is hard.

- For BMSZ16 Obfuscation and BLRSZZ16 ORE it is provably hard to annihilate zero-test polynomials (from standard assumptions [GMMSSZ16])
- New multilinear map hard to annihilate (under new non-standard assumption).

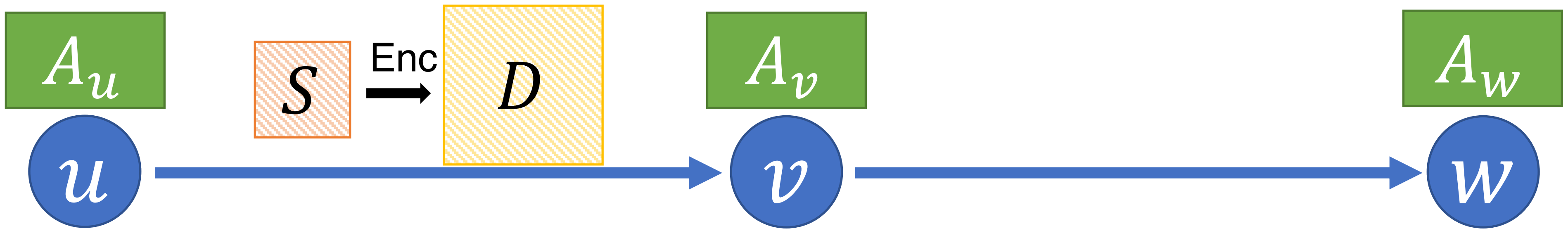
GGH15 Construction



GGH15
Construction

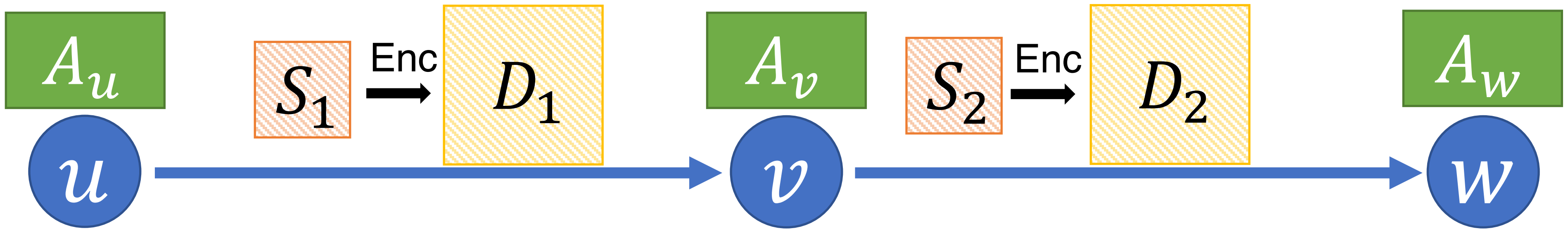


GGH15 Construction

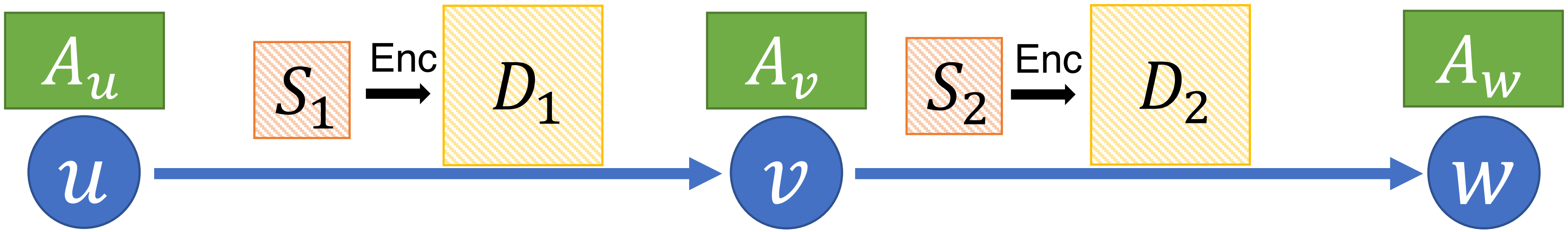


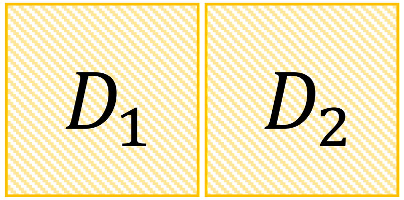
$$A_u D = S A_v + E$$

GGH15 Construction

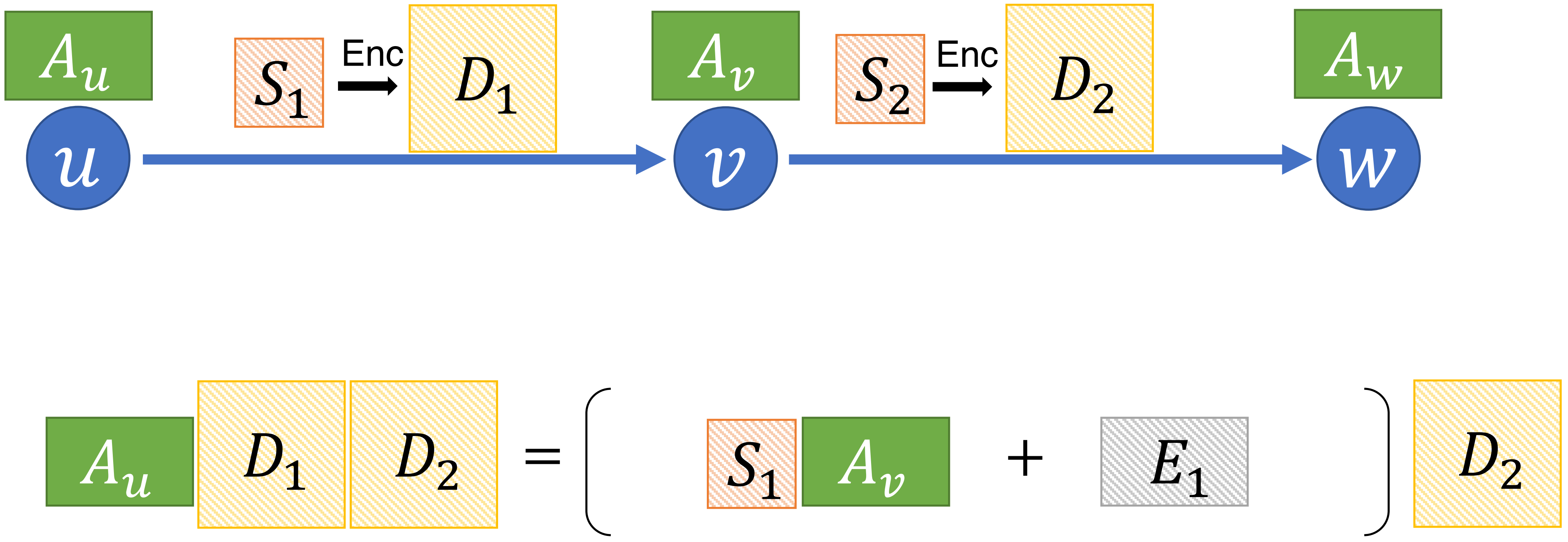


GGH15 Construction

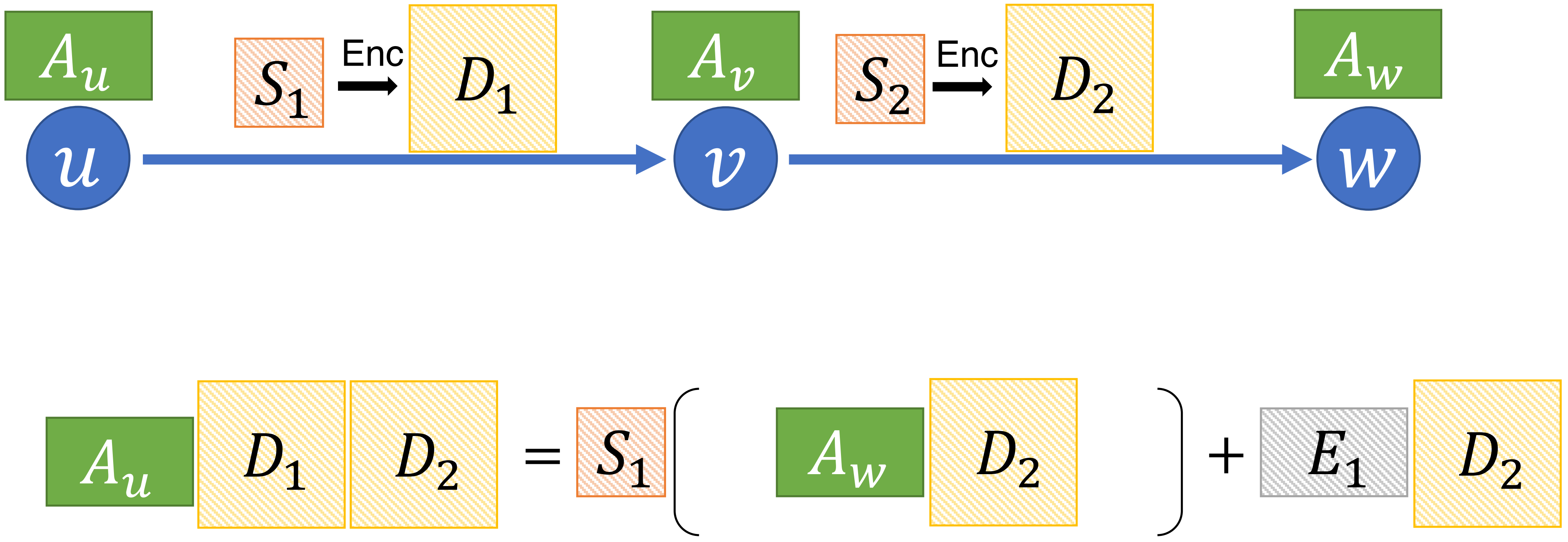




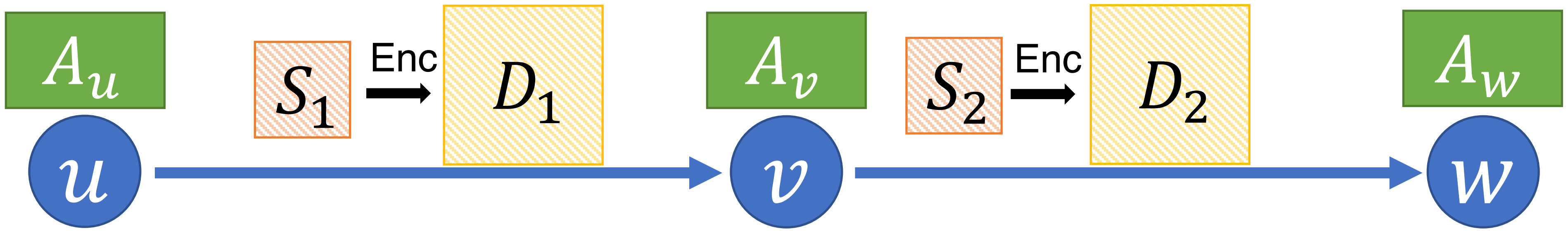
GGH15 Construction



GGH15 Construction

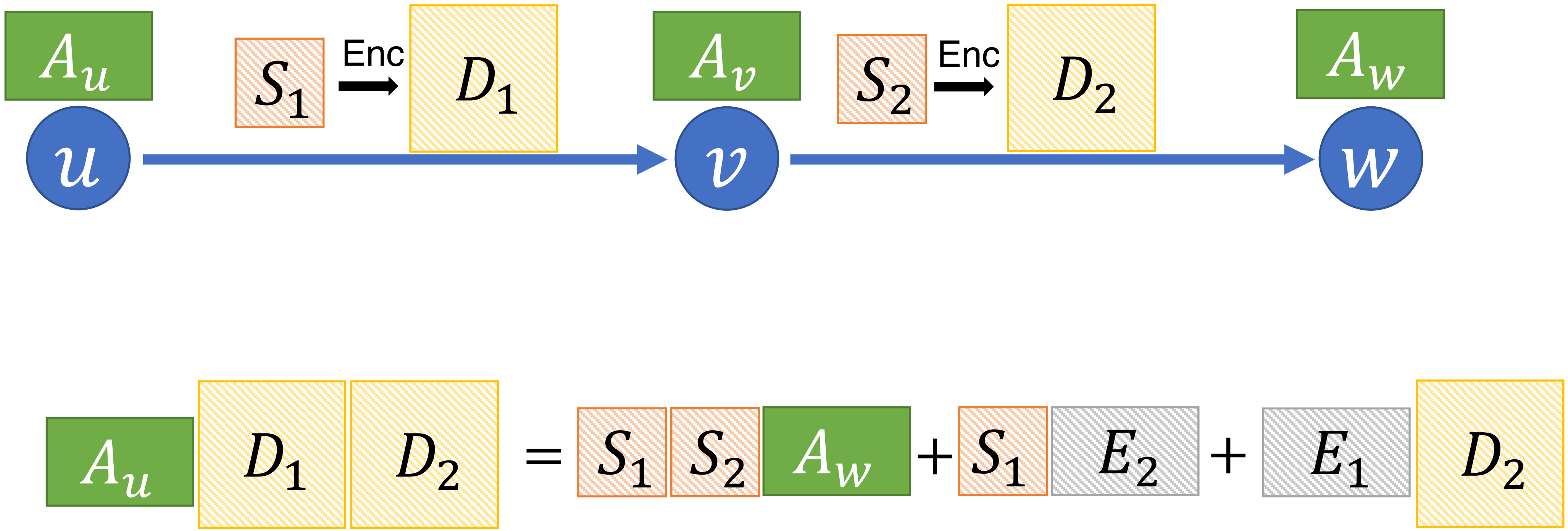


GGH15 Construction

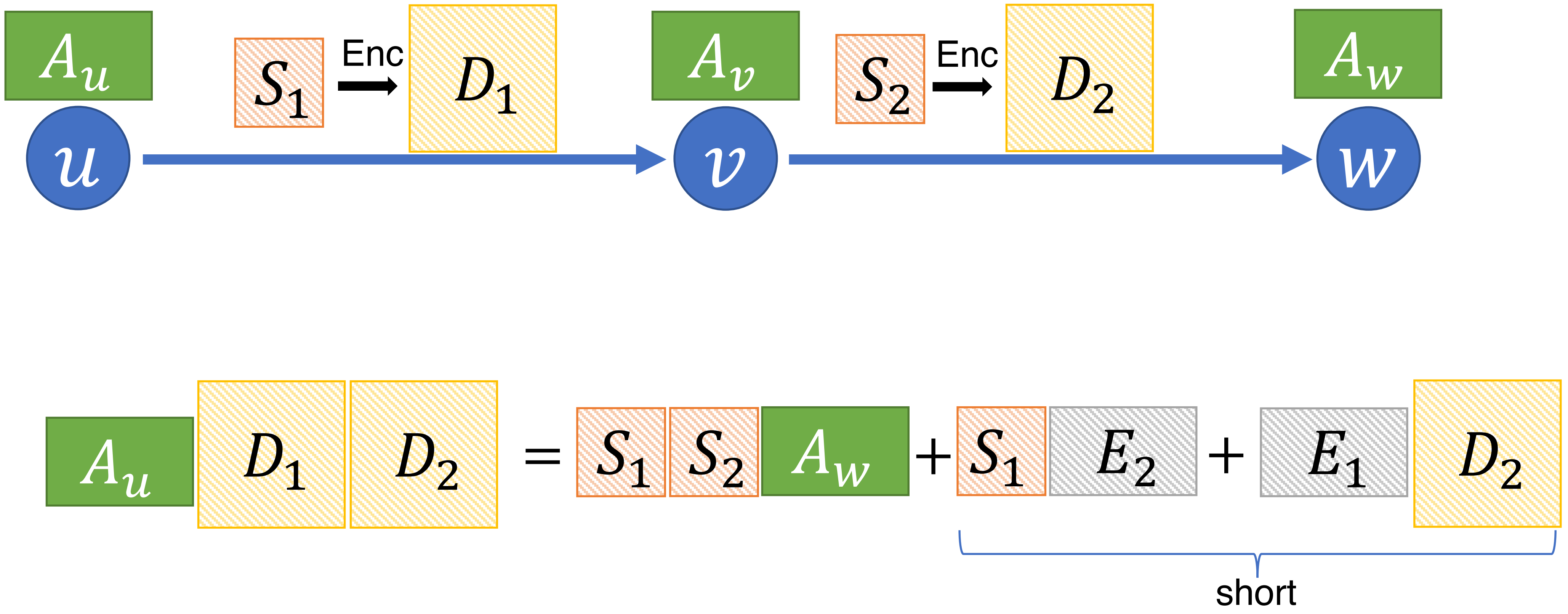


$$A_u D_1 D_2 = S_1 \left(S_2 A_w + E_2 \right) + E_1 D_2$$

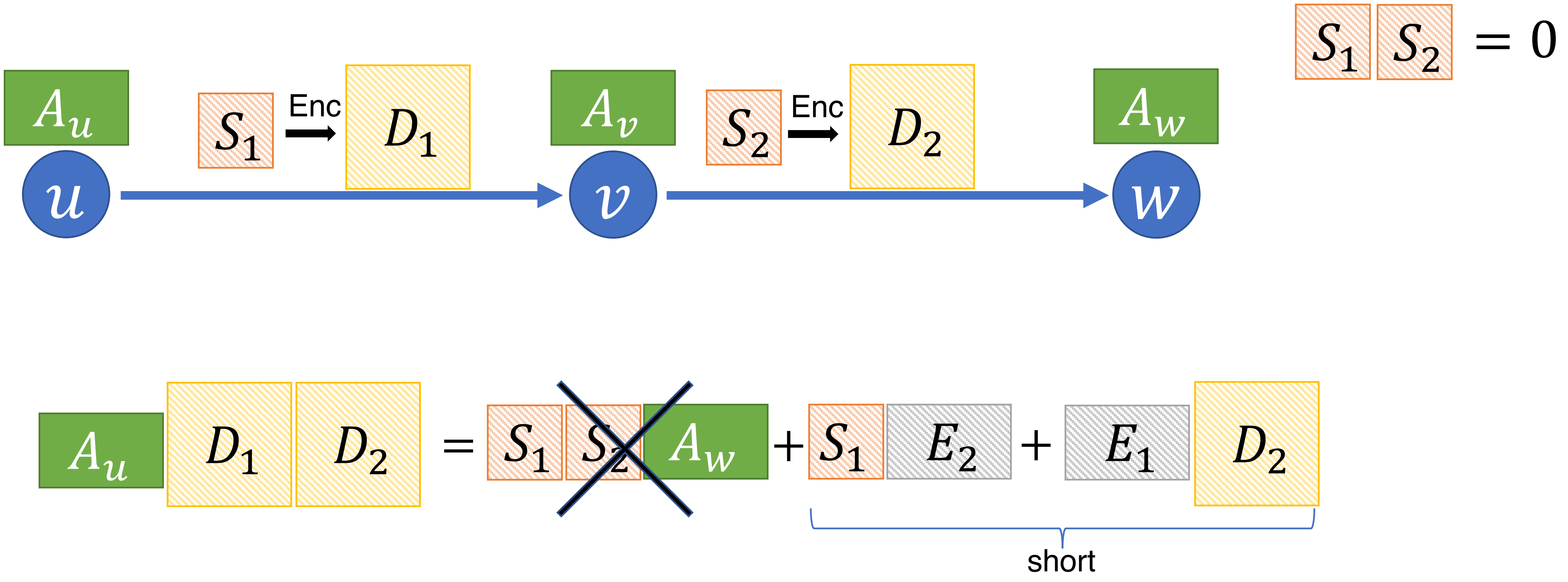
GGH15 Construction



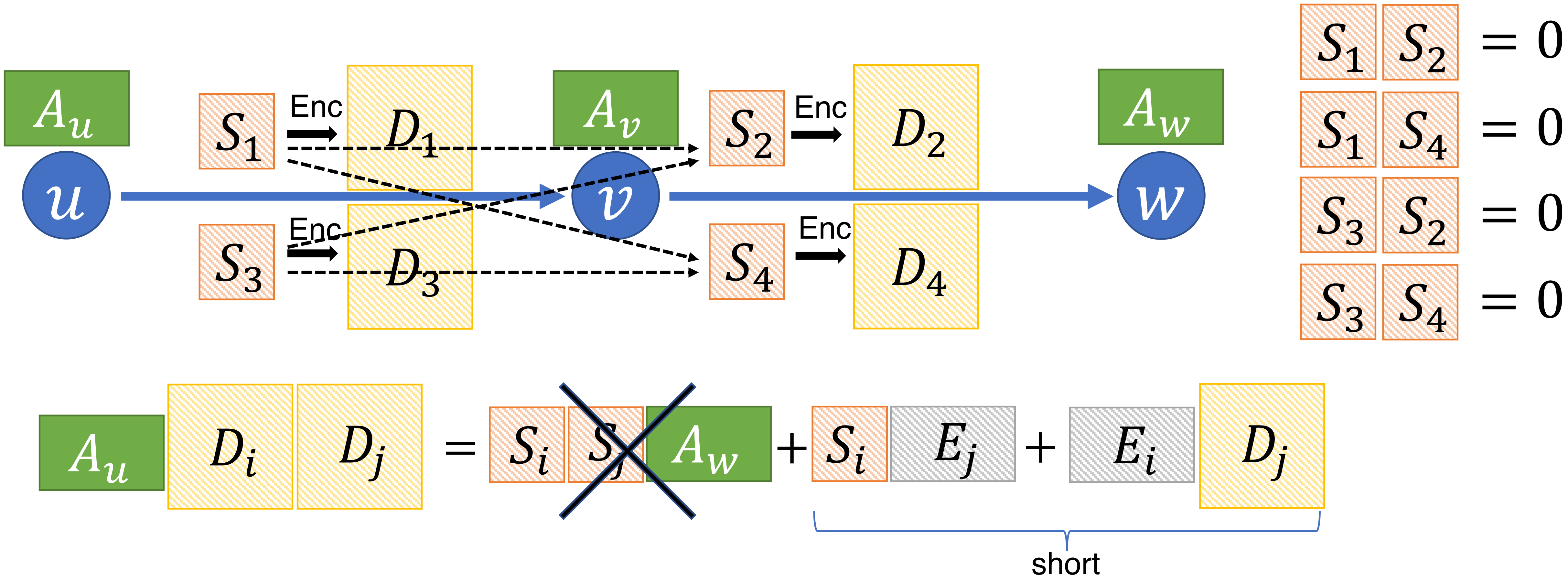
GGH15 Construction



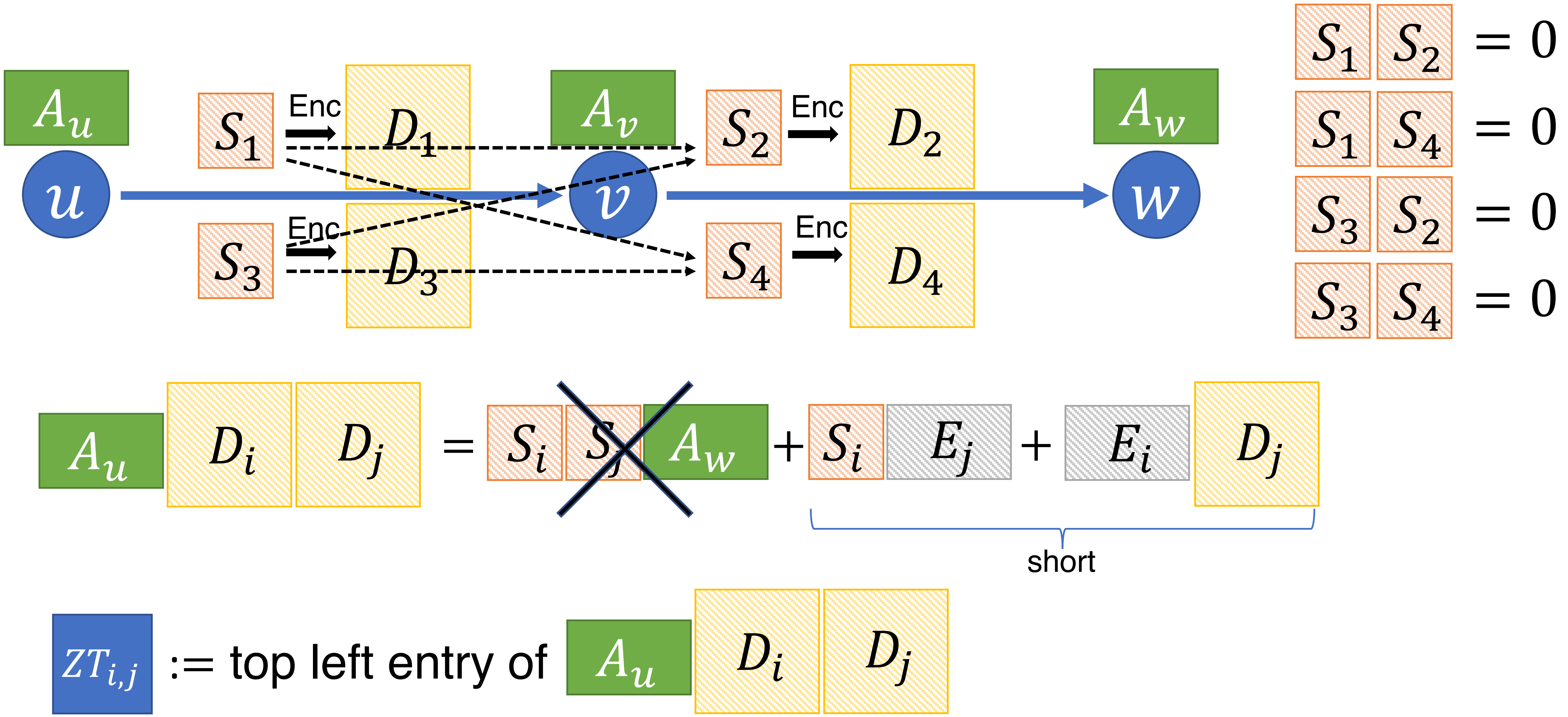
GGH15 Construction



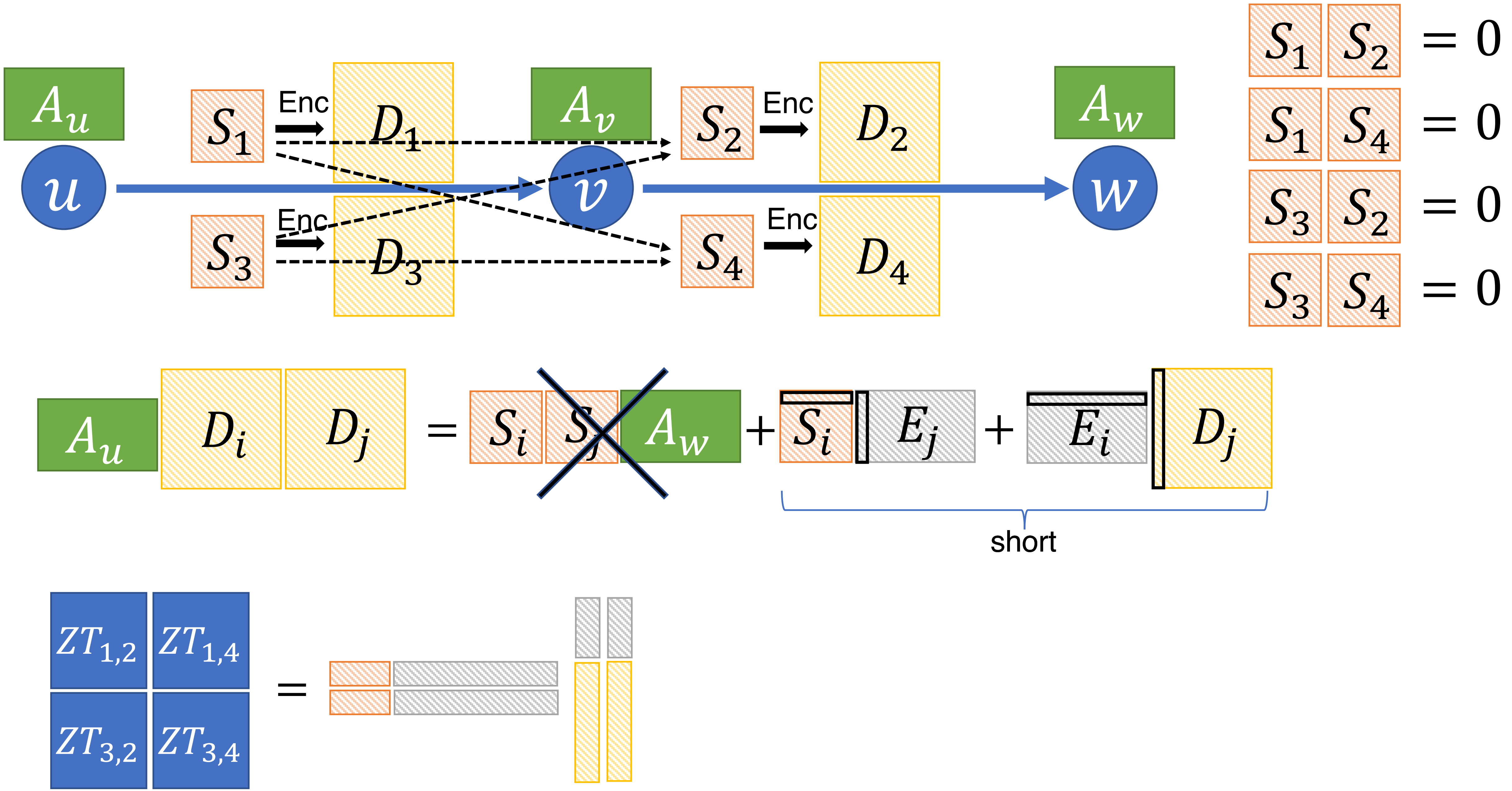
GGH15 Construction



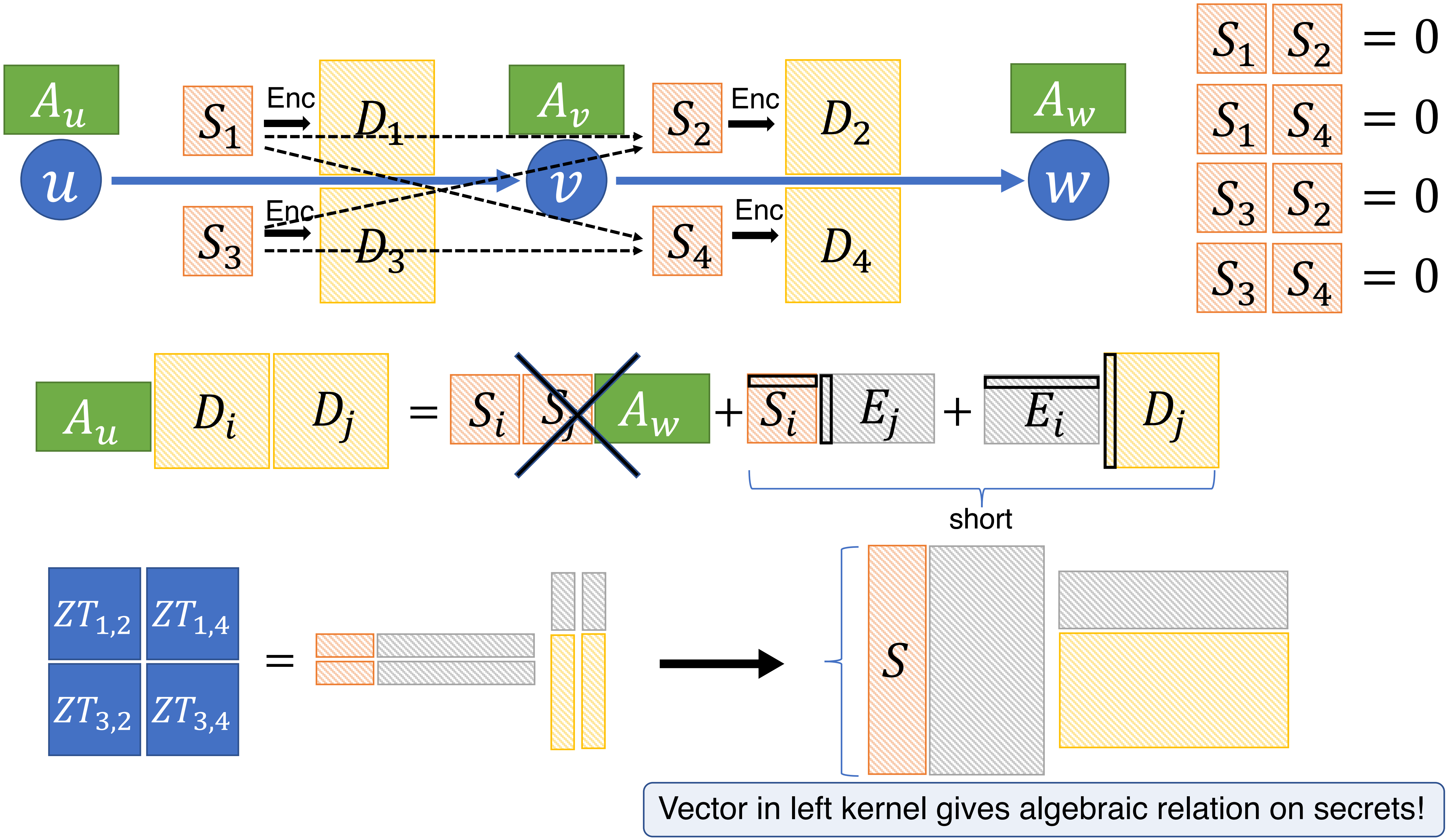
Toy Attack
[CLLT16,CGH17,CVW18]



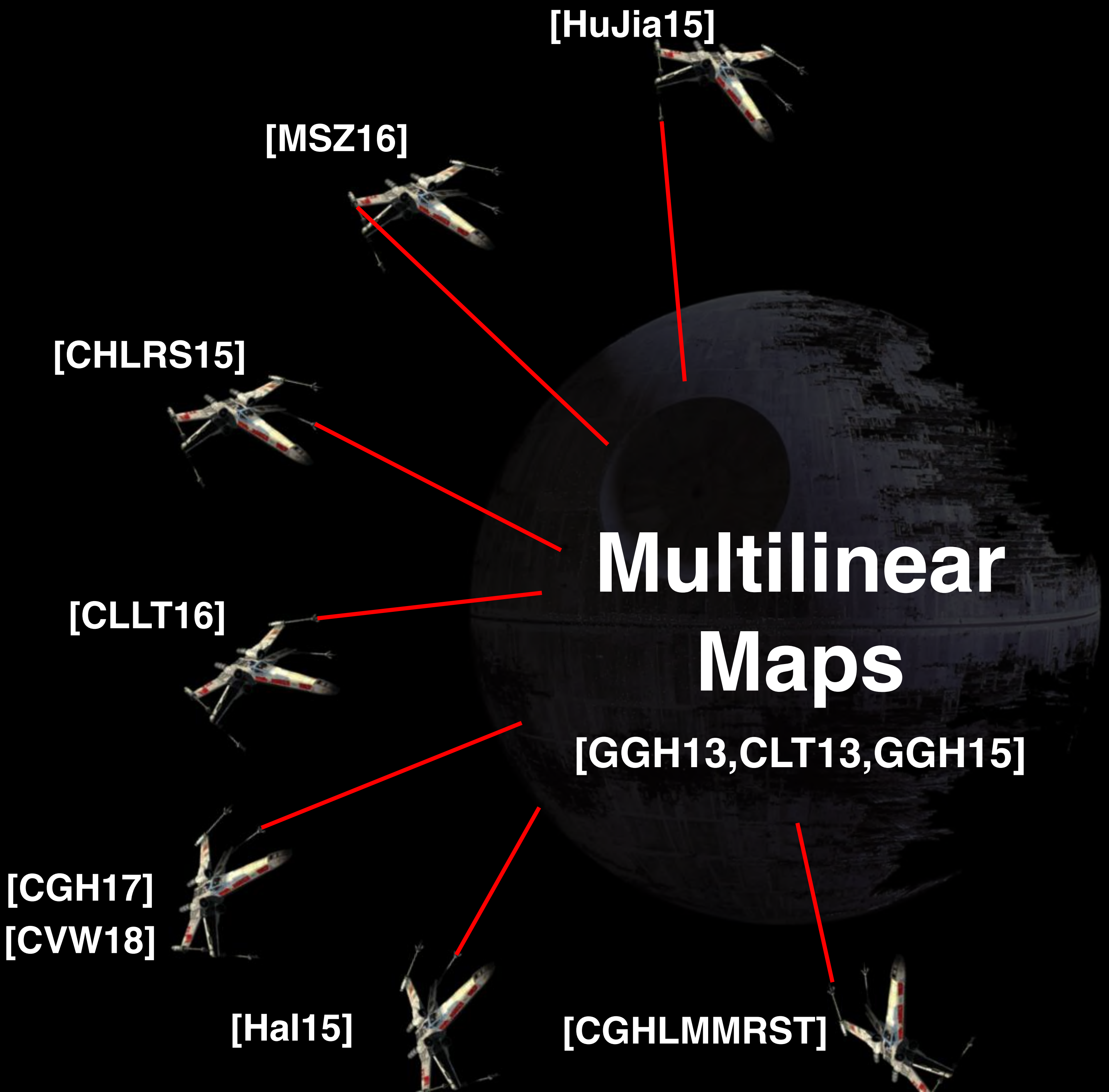
A_u
A_v
A_w
u
v
w
S_1
Enc
D_1
S_3
Enc
D_3
S_2
Enc
D_2
S_4
Enc
D_4
S_1 S_2 = 0
S_1 S_4 = 0
S_3 S_2 = 0
S_3 S_4 = 0
A_u D_i D_j = S_i S_j A_w + S_i E_j + E_i D_j
short
$ZT_{i,j}$:= top left entry of A_u D_i D_j

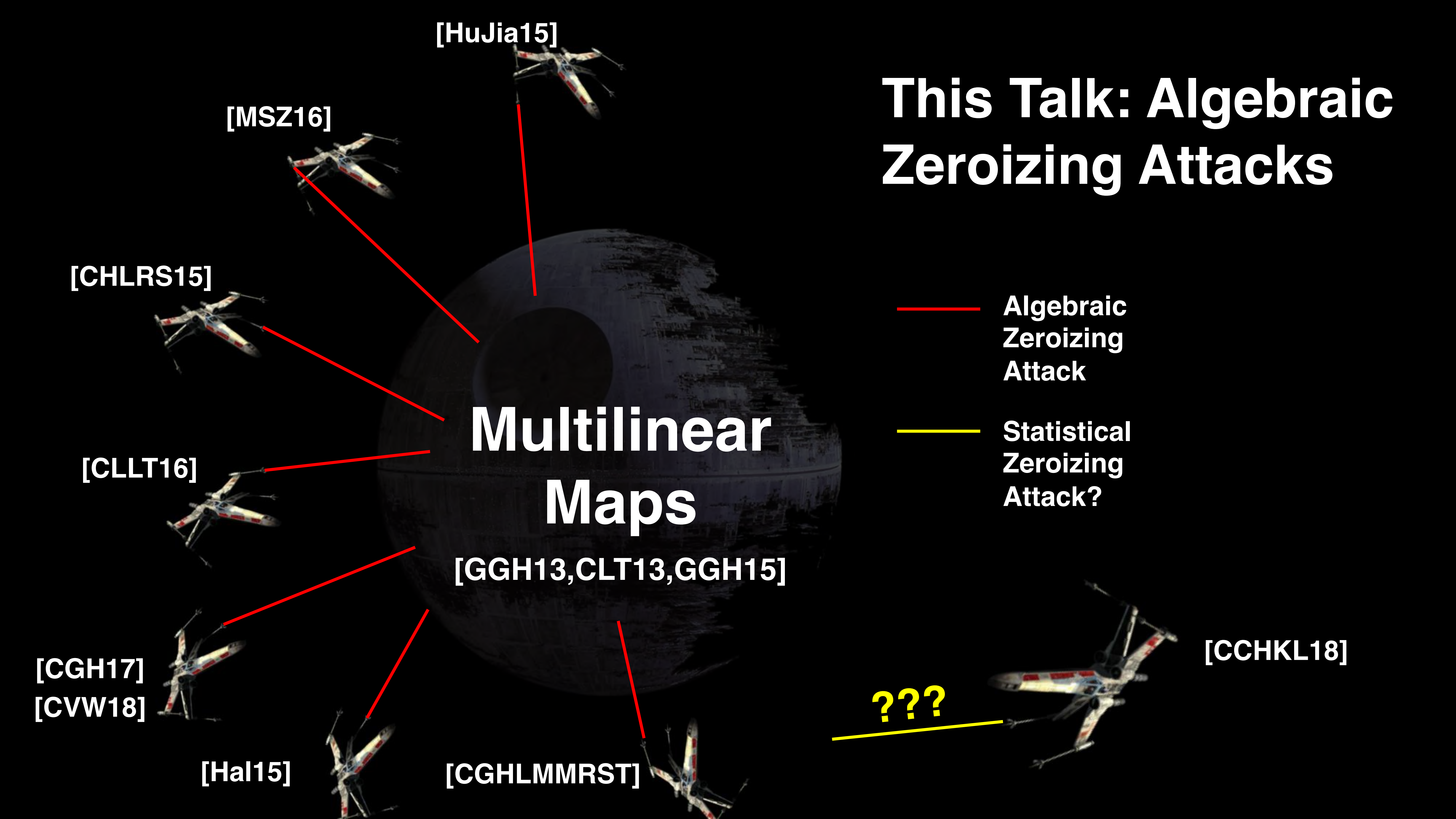


A_u
u
S_1
Enc
D_1
S_3
Enc
D_3
A_v
v
S_2
Enc
D_2
S_4
Enc
D_4
A_w
w
$S_1 S_2 = 0$
$S_1 S_4 = 0$
$S_3 S_2 = 0$
$S_3 S_4 = 0$
$A_u D_i D_j = S_i S_j A_w + S_i E_j + E_i D_j$
short
$ZT_{1,2}$ $ZT_{1,4}$
$ZT_{3,2}$ $ZT_{3,4}$
=


A_u
u
S_1
Enc
D_1
A_v
v
S_2
Enc
D_2
A_w
w
S_3
Enc
D_3
S_4
Enc
D_4
$S_1 S_2 = 0$
$S_1 S_4 = 0$
$S_3 S_2 = 0$
$S_3 S_4 = 0$
A_u D_i D_j = S_i S_j A_w + S_i E_j + E_i D_j
short
$ZT_{1,2}$ $ZT_{1,4}$
$ZT_{3,2}$ $ZT_{3,4}$
=
S
Vector in left kernel gives algebraic relation on secrets!

This Talk: Algebraic Zeroizing Attacks





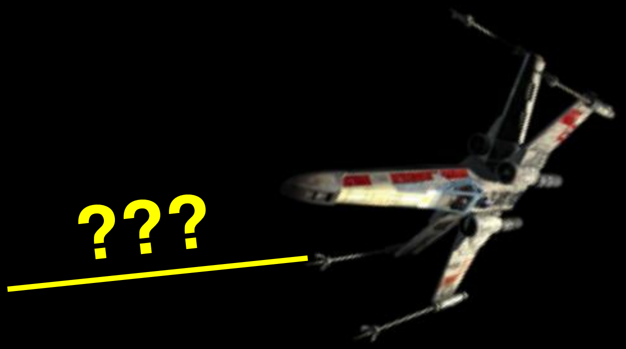
This Talk: Algebraic Zeroizing Attacks
[HuJia15]
[MSZ16]
[CHLRS15]
[CLLT16]
[CGH17]
[CVW18]
[Hal15]
[CGHLMMRST]
Multilinear Maps
[GGH13,CLT13,GGH15]
Algebraic Zeroizing Attack
Statistical Zeroizing Attack?
???
[CCHKL18]

Statistical Zeroizing Attack: Cryptanalysis of Candidates of BP Obfuscation over GGH15 Multilinear Map

Jung Hee Cheon, Wonhee Cho, Minki Hhan, Jiseung Kim, and Changmin Lee

ePrint: **2018/1081**

First polynomial-time, non-algebraic zeroizing attack on GGH15-based obfuscation!



[CCHKL18]

CCHKL18 updated ePrint 23 hours ago:

Note: We temporarily add the disclaimer not to mislead the readers and audiences of TCC.

Disclaimer

The authors of BGMZ obfuscation [4] (TCC'18) report that there are flaws of cryptanalysis of BGMZ obfuscation in Section 5. In particular, the current optimal parameter choice of BGMZ obfuscation is robust against our attack, while the attack lies outside the provable security of BGMZ obfuscation.
The flaws in the analysis in Section 5 are as follows:

- ν is chosen to $\mathsf{poly}(\lambda)$ in this paper whereas the original paper [4] chooses $\nu = 2^\lambda$ (or at least super-polynomial of λ).
- The analysis of our attack claims that $\left(1 + \frac{2}{g}\right)^h$ is polynomial of λ, but it is not true since $g = 5$ is constant.

We remark that our attack gives a constraint on the parameters; BGMZ obfuscation with $\sigma = \exp(\lambda)^a$ can be broken in the same manner with slightly modified proof. We will update the paper as soon as possible.

[a] Interestingly, this choice gives a countermeasure of CVW obfuscation.

GGH15 Algebraic Zeroizing Model

Extend Generic Model to ***allow adversary*** to perform an algebraic zeroizing attack.

$\{h_i\}$ ←

$p_j(\{h_i\}_i)$ →

← ("zero", "$\boldsymbol{T_j}$") / "non-zero"

$Q(\{\boldsymbol{T_j}\}, \{\boldsymbol{S_i}\})$ →

← "successful" / "unsuccessful"

Generic Model + GGH15 Attacks

Graph $\mathbb{G}$, **Plaintexts** $\{S_i, u_i \to v_i\}$

$D_i \leftarrow$ **Enc**$(S_i, u_i \to v_i)$

Handles $\{h_i \to (D_i, S_i, u_i \to v_i)\}$

Zero Test Queries: if

- p_j edge-respecting
- $p_j(\{D_i\}_i) = (T_j, \text{"is zero"})$

Return *post-zero-test* handle "$\boldsymbol{T_j}$"

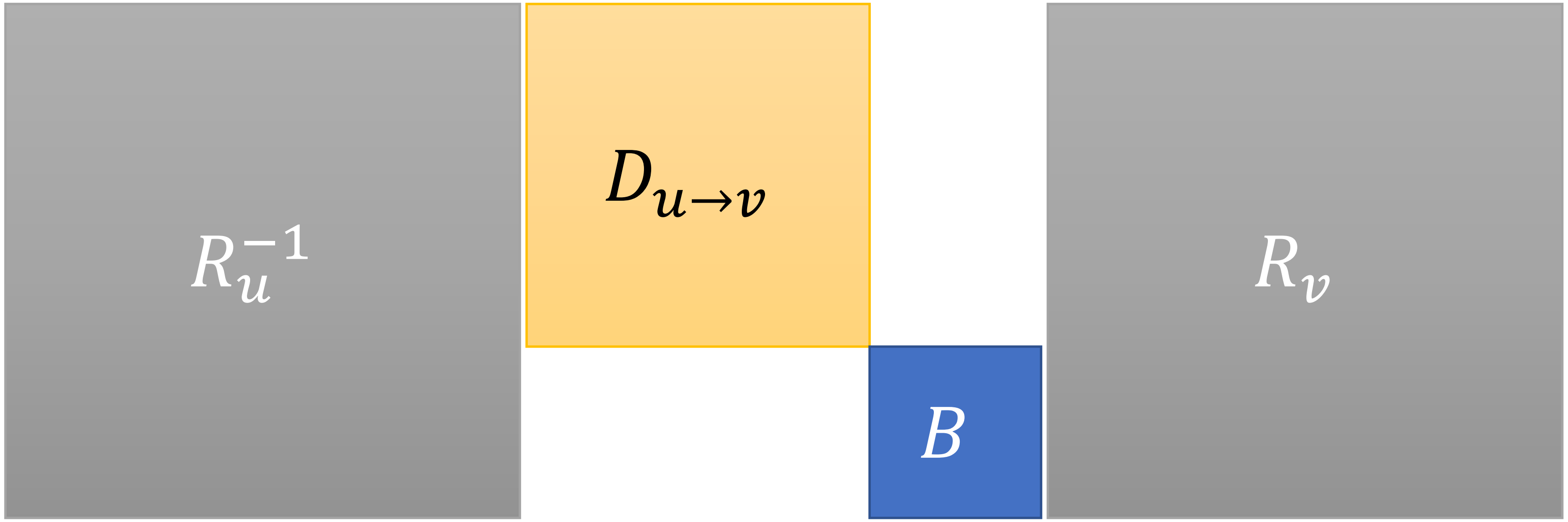
Post Zero Test Return "WIN" if

- $Q(\{T_j\}, \{S_i\}) = 0$
- $Q(\{\boldsymbol{T_j}\}, \{S_i\}) \not\equiv 0$, $Q(\{T_j\}, \{\boldsymbol{S_i}\}) \not\equiv 0$

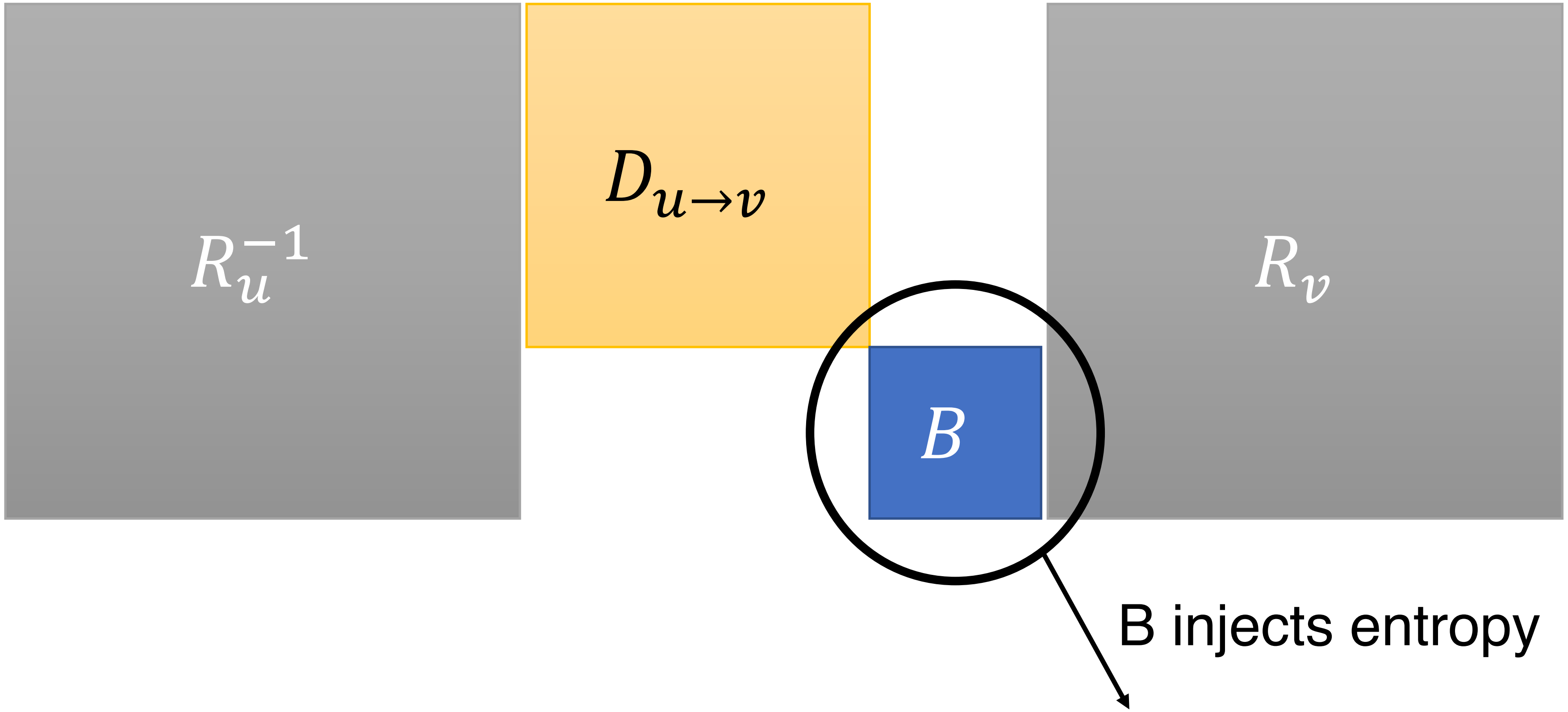
Our GGH15 Variant

$$D_{u \to v}$$

Our GGH15 Variant

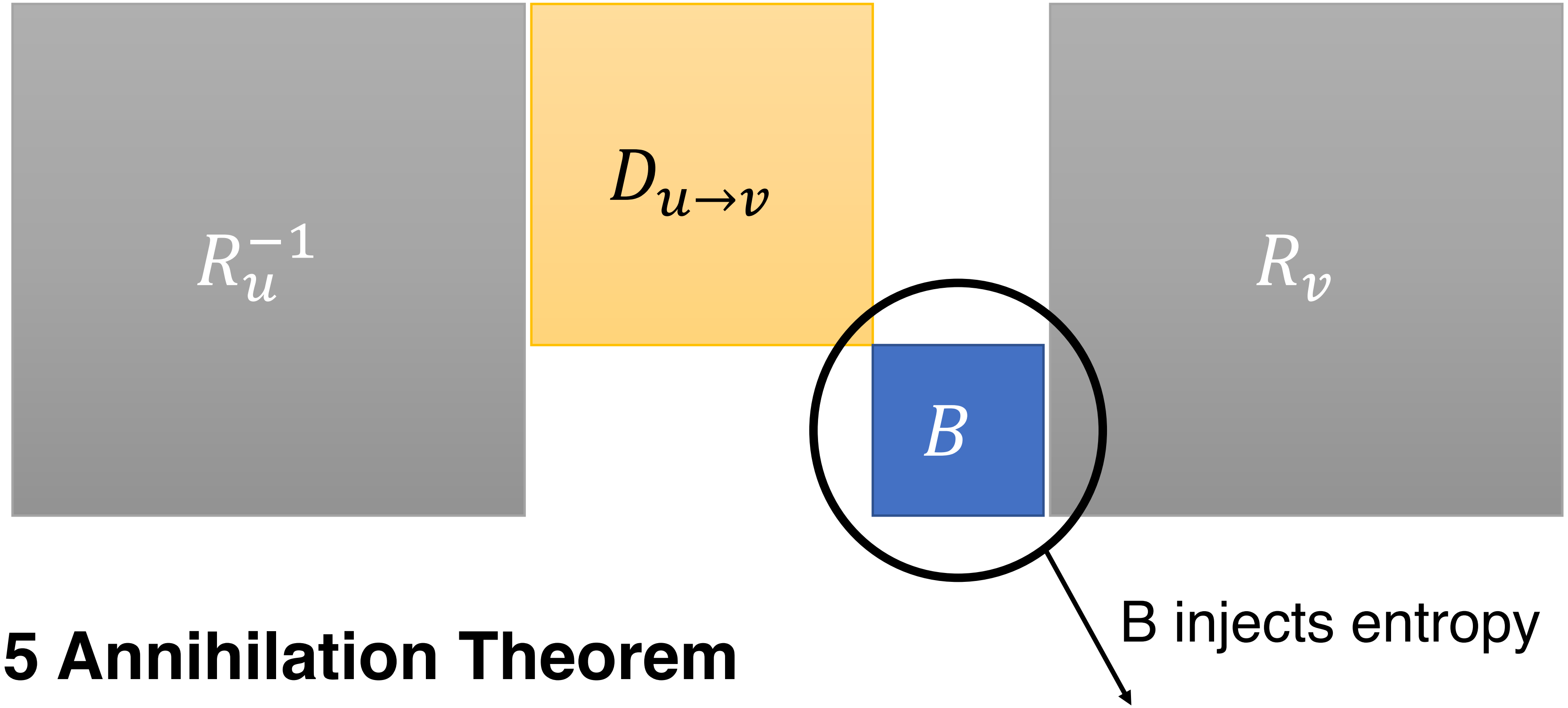


Our GGH15 Variant



algebraic relation involving $\{T_j\}$ → annihilation of zero-test polynomials $\{p_j\}$

Our GGH15 Variant



GGH15 Annihilation Theorem

hardness of annihilating zero-test polynomials → security in our model!

Branching Program (BP) Obfuscation

$$f \rightarrow \begin{matrix} S_{1,1} & S_{2,1} & S_{3,1} & S_{4,1} & S_{5,1} & S_{6,1} \\ S_{1,2} & S_{2,2} & S_{3,2} & S_{4,2} & S_{5,2} & S_{6,2} \end{matrix}$$

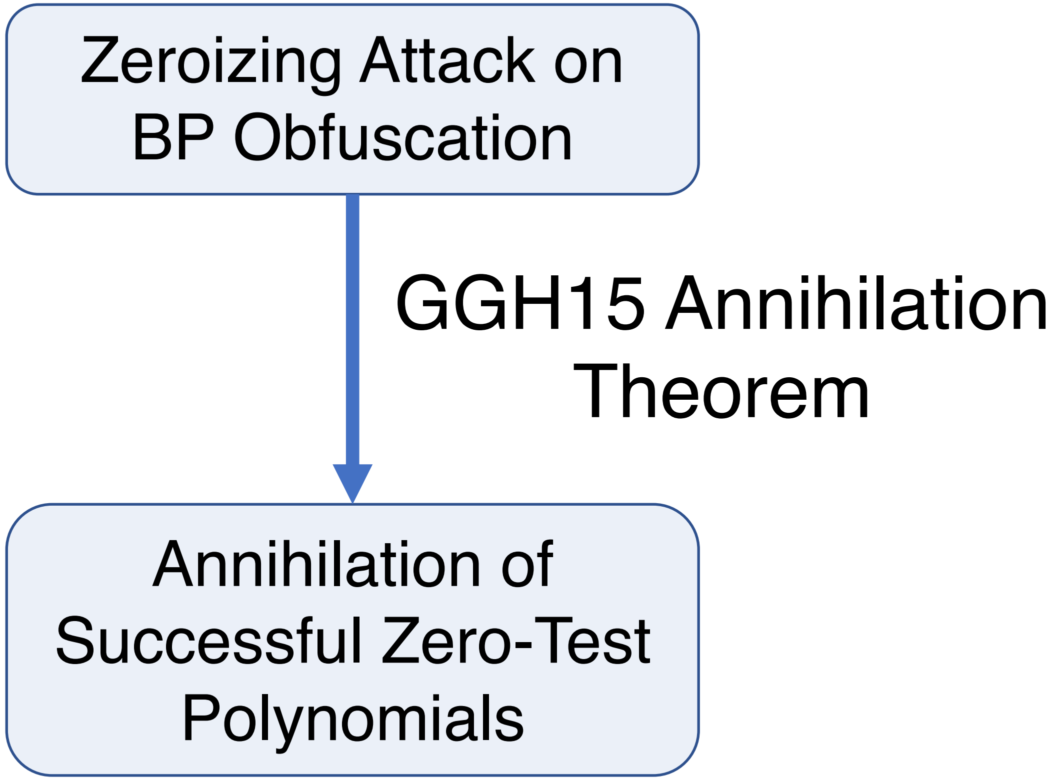
$$f(x) = 1 \leftrightarrow \prod_i S_{i,x_{inp(i)}} = 0$$

Simple Obfuscation Construction:

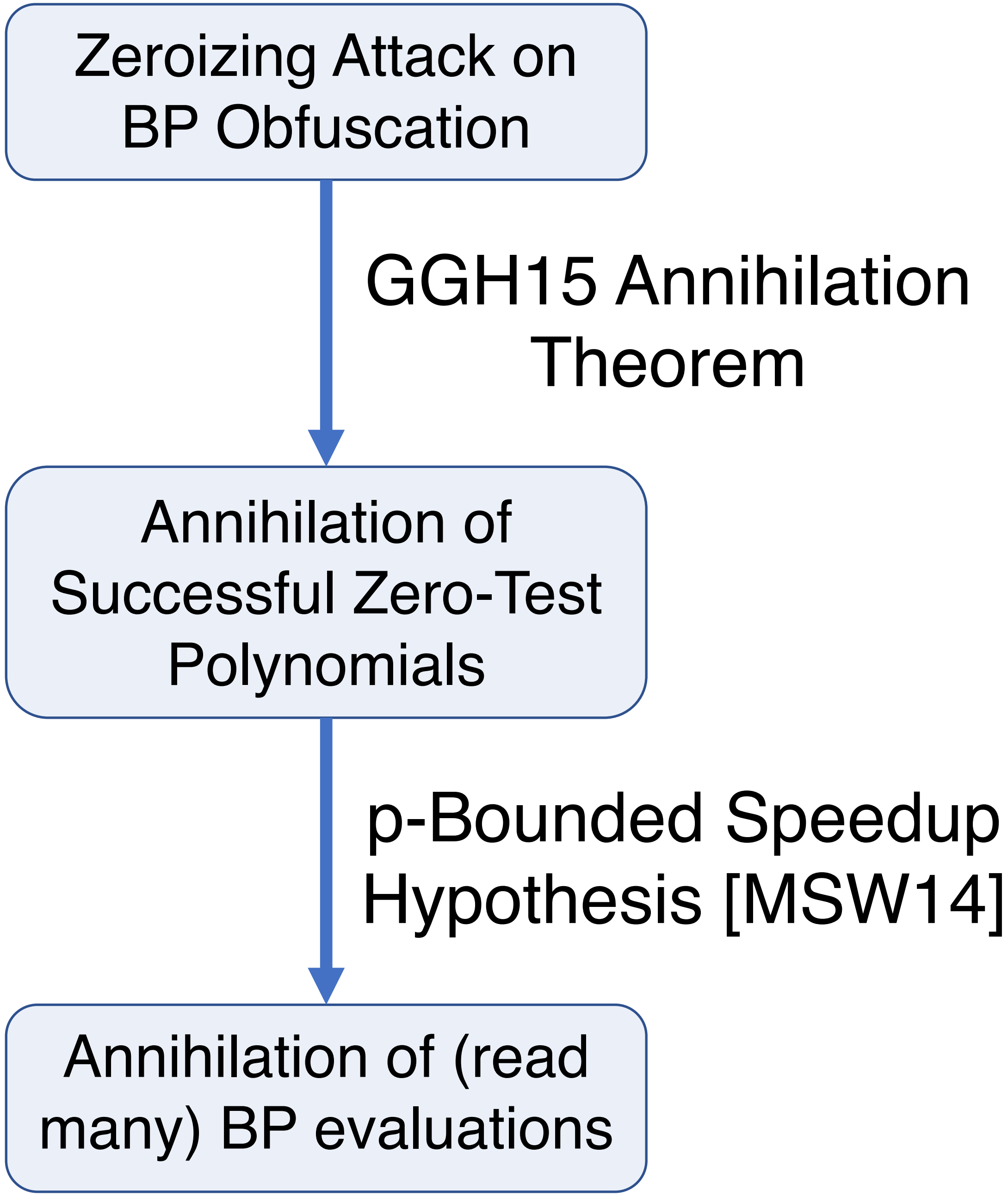
Encode $S_{i,b}$ matrices with our new GGH15 variant

Zeroizing Attack on BP Obfuscation

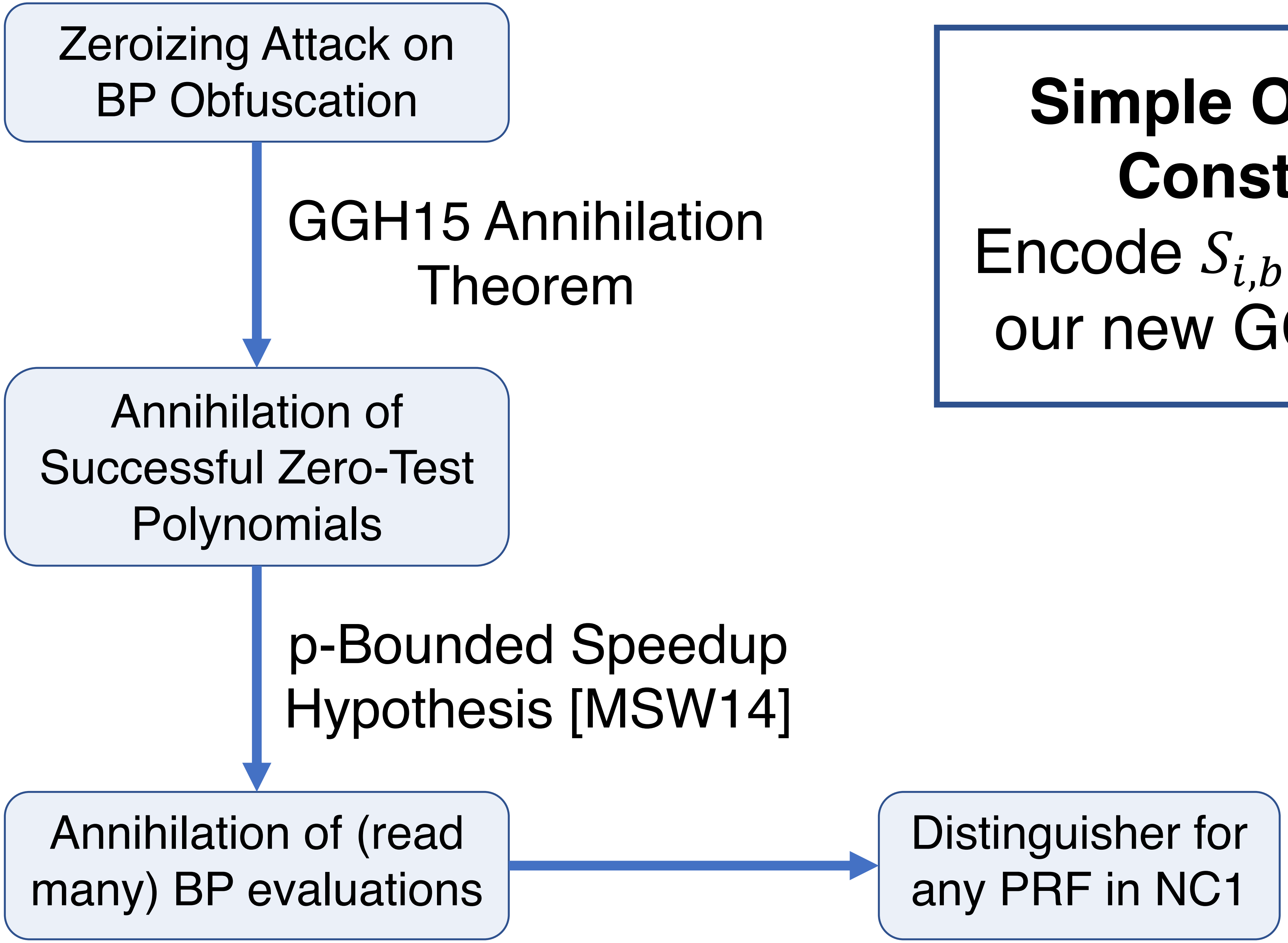
Simple Obfuscation Construction:
Encode $S_{i,b}$ matrices with our new GGH15 variant



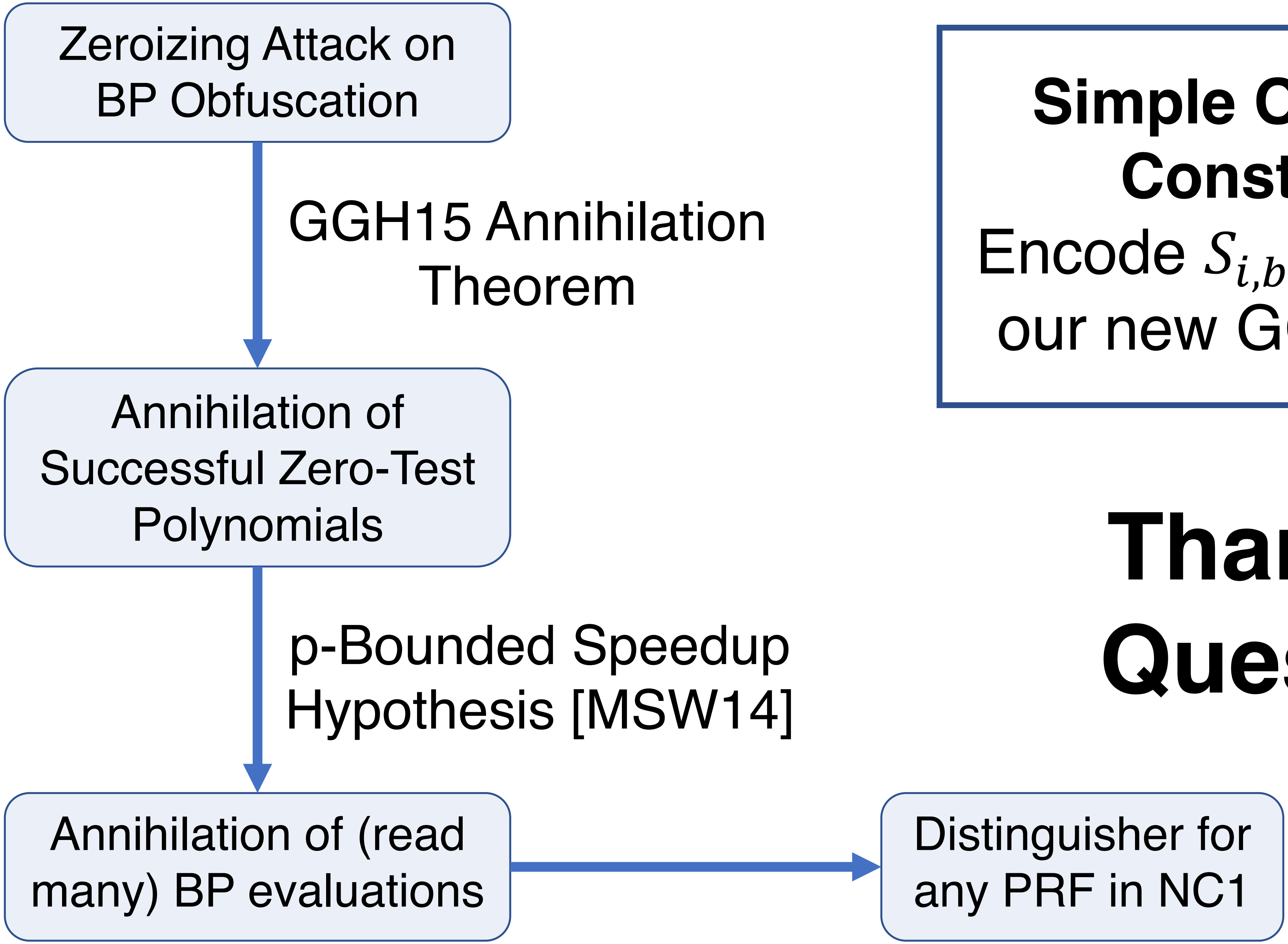
Simple Obfuscation Construction:
Encode $S_{i,b}$ matrices with our new GGH15 variant



Simple Obfuscation Construction:
Encode $S_{i,b}$ matrices with our new GGH15 variant



Simple Obfuscation Construction:
Encode $S_{i,b}$ matrices with our new GGH15 variant



Simple Obfuscation Construction:
Encode $S_{i,b}$ matrices with our new GGH15 variant

Thank you! Questions?