CHES 2019

3-Share Threshold Implementation of AES S-box without Fresh Randomness

Takeshi Sugawara

The University of Electro-Communications, Japan
University of Michigan, US

This work is funded by JSPS KAKENHI Grant Number 17H06681 and JP18H05289

Overview

Methodology

Threshold implementation
(Nicova et al., ICICS2006)

Changing of the guards
(Daemen, CHES2017)

Generalized
Changing of the guards
(This work)

Implementation

Difficulty in realizing
3-share + Uniform TI for
AES and Keccak for 10+ years

3-Share + Uniform Keccak S-box
(Daemen, CHES2017)

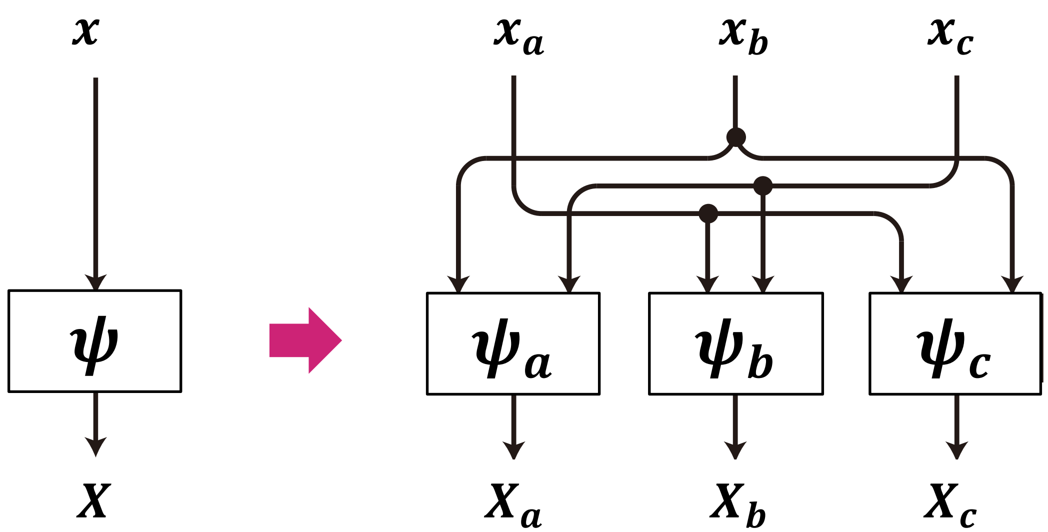
4-Share + Uniform AES S-box
(Wegener & Moradi, COSADE2018)

3-Share + Uniform AES S-box
(This work)

TI: Threshold Implementation

- Implement crypto while keeping shared representation of intermediate variables

Input share (x_a, x_b, x_c) :
$x_a \oplus x_b \oplus x_c = x$



Output share (X_a, X_b, X_c) :
$X_a \oplus X_b \oplus X_c = X$

Sharing $\{\psi_a, \psi_b, \psi_c\}$ maps a share to another share

Correctness:
$\{\psi_a, \psi_b, \psi_c\}$ gives the correct result

Non-completeness:
Each map uses only a proper subset

Uniformity

- **Uniformity about shares**
  - For each (raw) value, all the possible shares should appear equally
  - Necessary for security against statistical attack

- **Uniformity about sharing**
  - The sharing preserves the uniformity about shares:

    Input share is uniform
    $\Longrightarrow$ output share is uniform

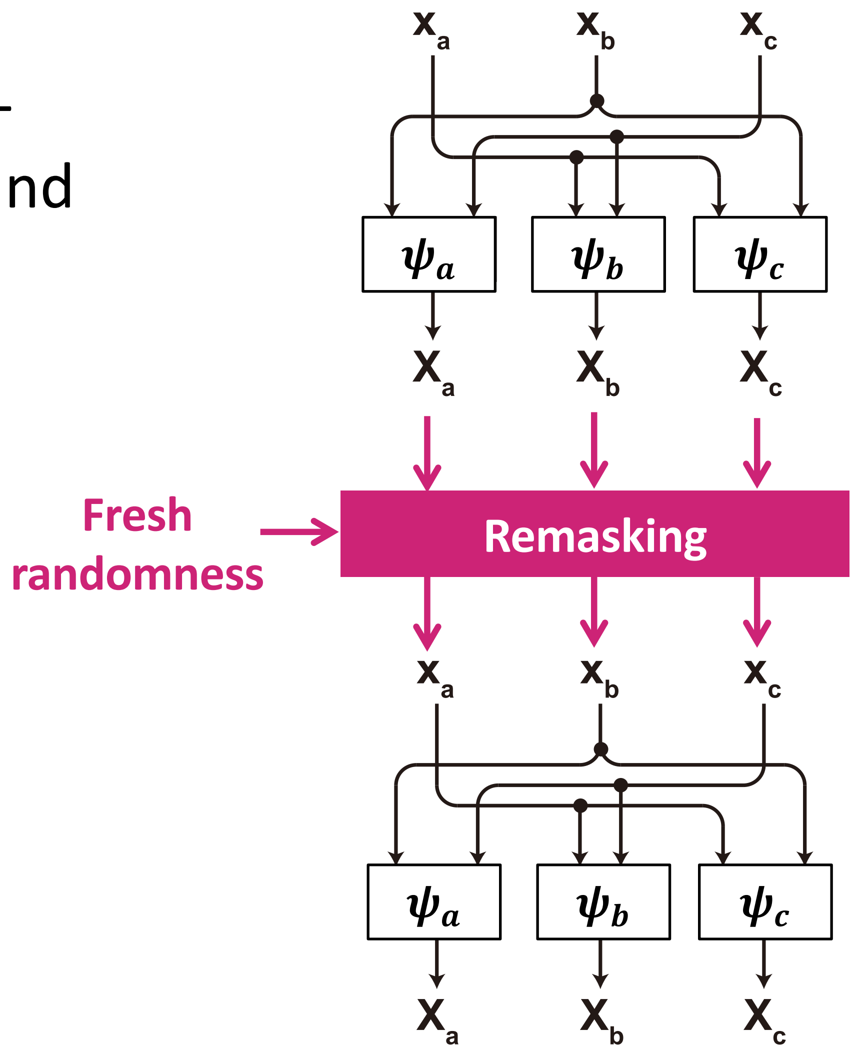
Example:
3-share of 1-bit variable

Raw value	Share	Prob.
0	(0,0,0)	1/16
0	(0,1,1)	1/16
0	(1,0,1)	1/16
0	(1,1,0)	1/16
1	(0,0,1)	3/16
1	(0,1,0)	3/16
1	(1,0,0)	3/16
1	(1,1,1)	3/16

Uniformity is difficult to satisfy

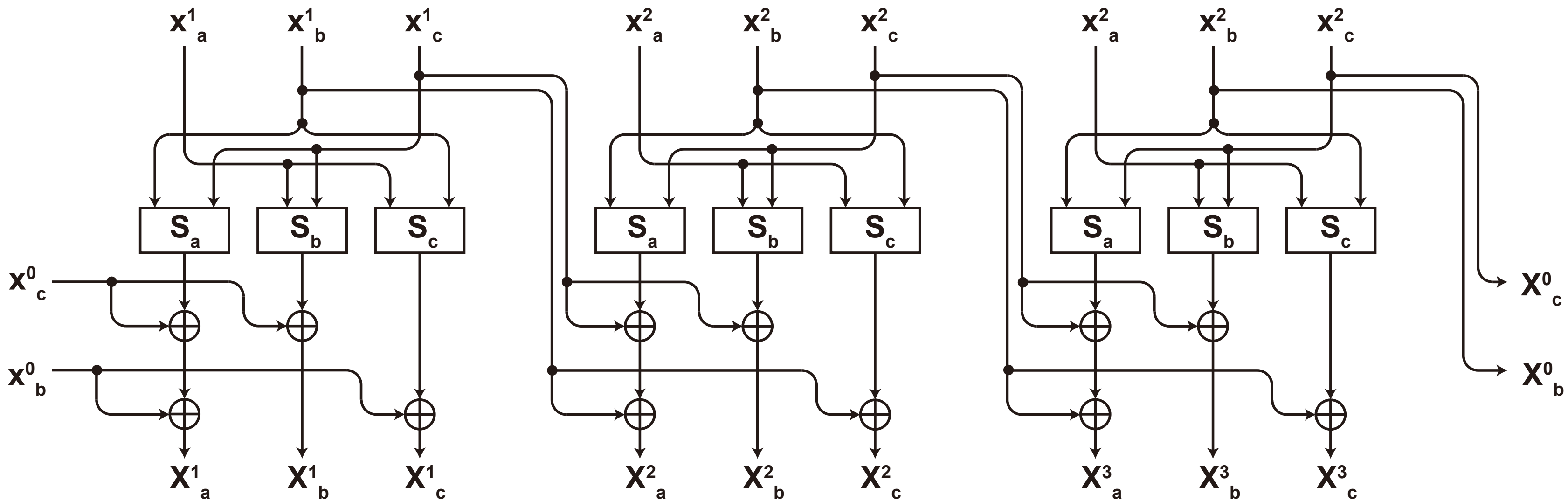
- There had been no 3-share + uniform sharing for Keccak and AES S-boxes until 2017

- If no uniformity, we should add fresh randomness to make the output share uniform again
  - 1—10 Kbits/AES
  - 10—50 bits/cycle



CotG: Changing of the Guards (Daemen, CHES2017)

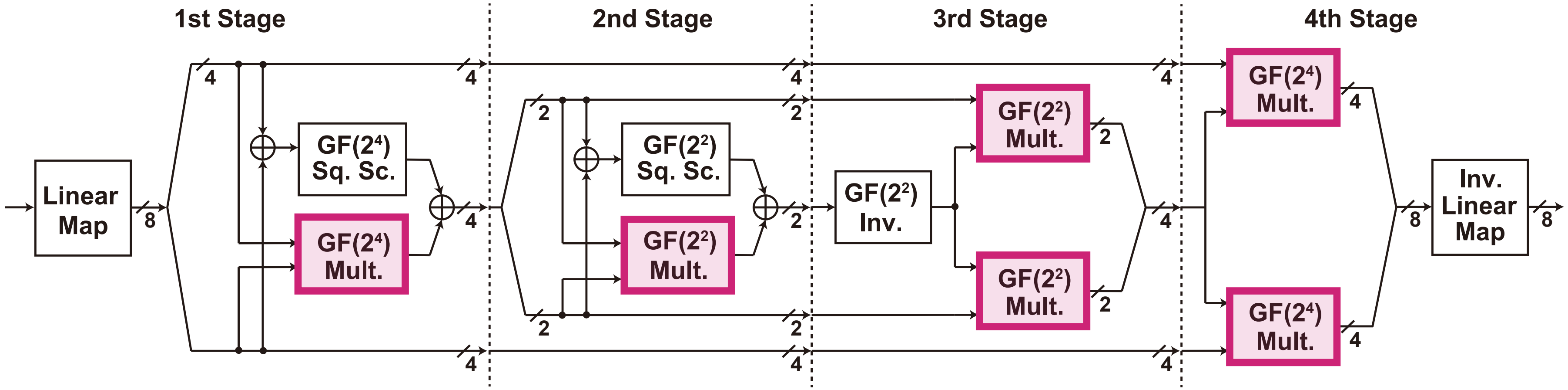
- Using a neighboring input share for (pseudo) remasking
- **Applicable to bijective mapping**
  - Succeeded in making 3-share + uniform Keccak S-box



Why we can't use CotG for 3-share AES S-box

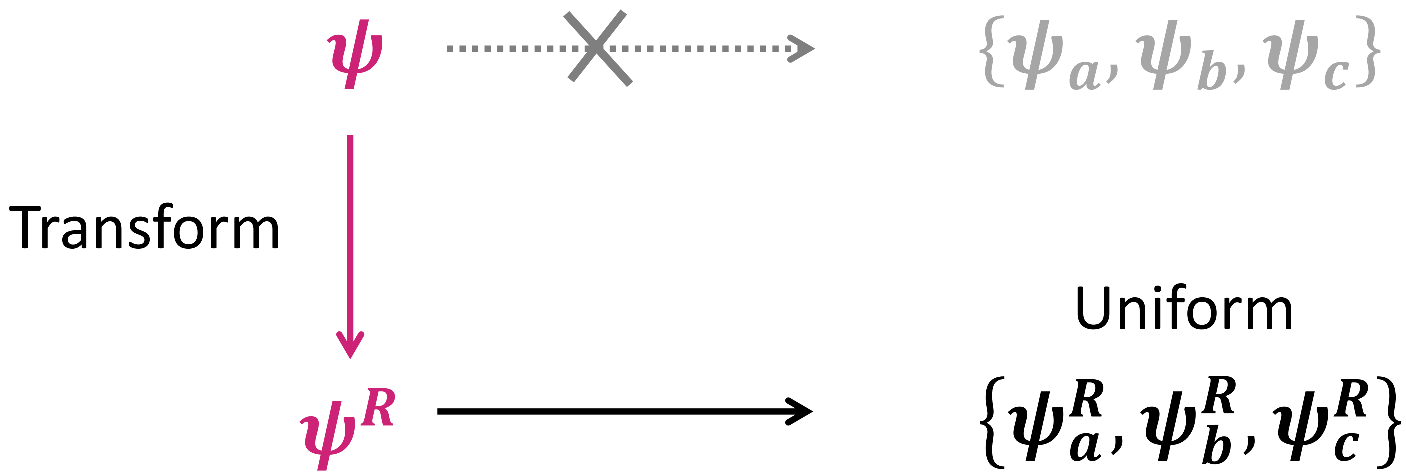
- We need to decompose S-box to reduce the number of shares, and we get **multiplications that are not bijective**

Canright's S-box implementation



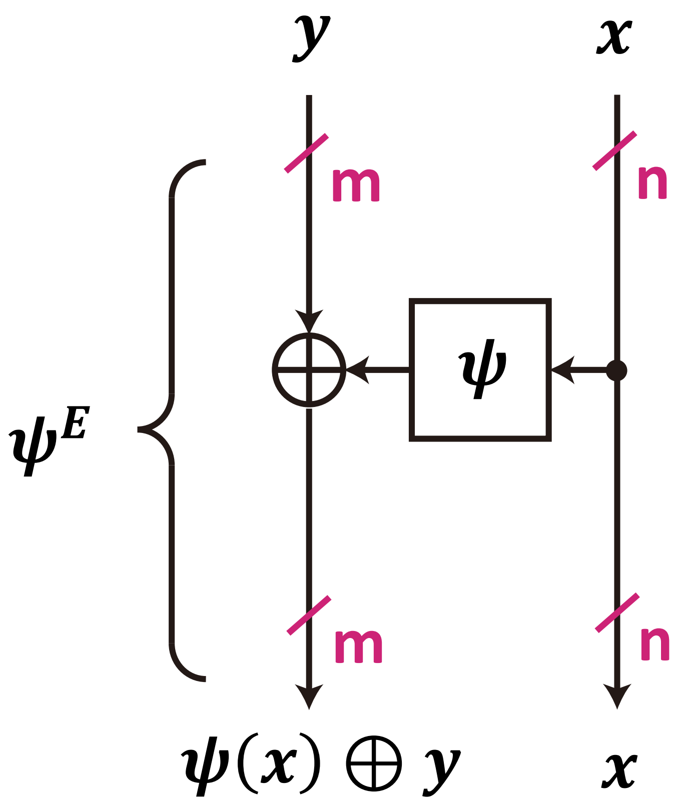
Basic idea toward generalization

- Transform the target mapping $\boldsymbol{\psi}$ into an **equivalent** mapping $\boldsymbol{\psi^R}$ that has a uniform sharing



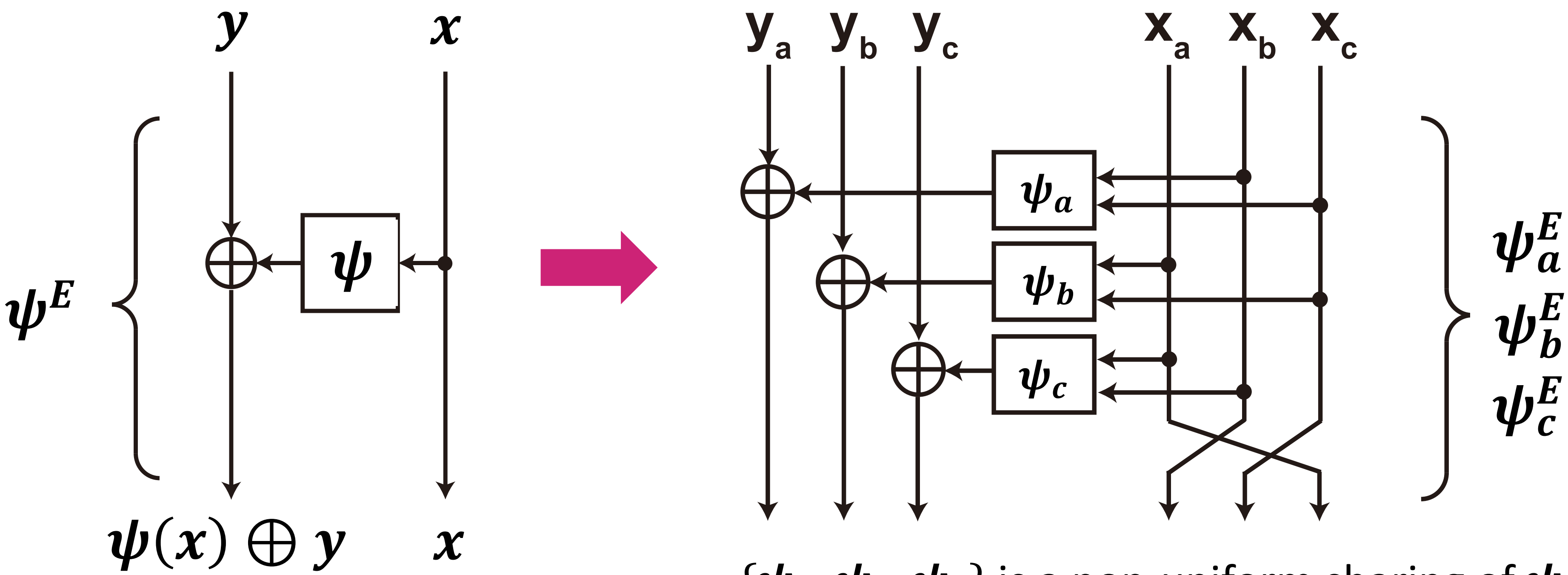
Expansion

- Transforming the target $\boldsymbol{\psi}$ into a bijective mapping $\boldsymbol{\psi}^{\boldsymbol{E}}$ using **the (unbalanced) Feistel network**



Expansion cont.

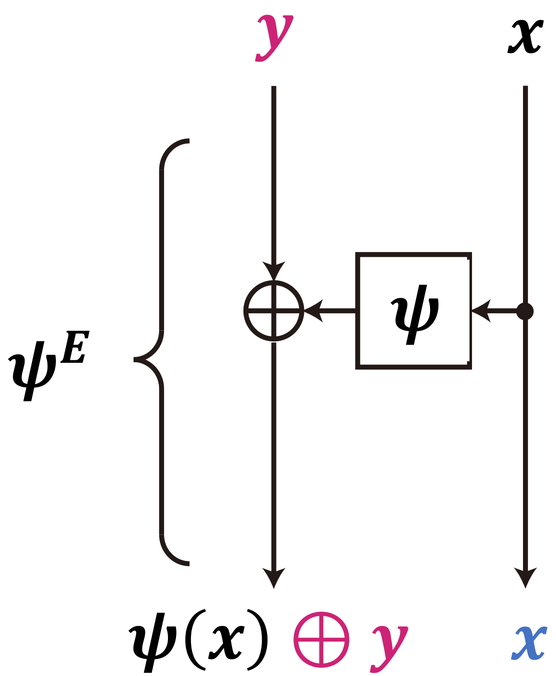
- ψ^E **always has a uniform sharing** $\{\psi_a^E, \psi_a^E, \psi_a^E\}$
  - ∵ The sharing is bijective because the Feistel structure is preserved
  - ∵ A sharing is bijective $\Longrightarrow$ the sharing is uniform



$\{\psi_a, \psi_b, \psi_c\}$ is a non-uniform sharing of ψ

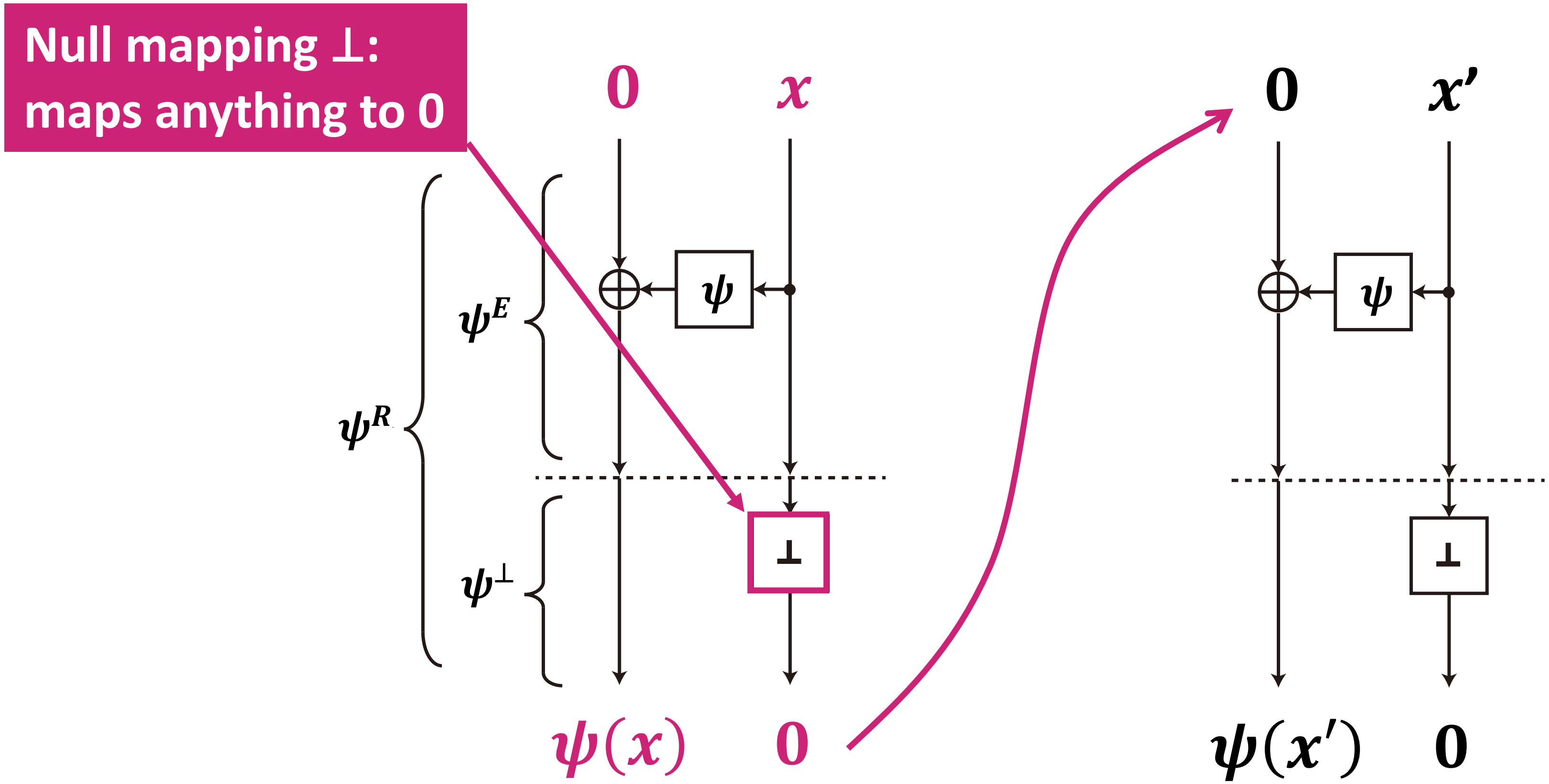
Expansion is not enough

- Feeding $\boldsymbol{\psi}^{\boldsymbol{E}}(\boldsymbol{x})$ to CotG does not make a lot of sense since it outputs $\boldsymbol{\psi}(\boldsymbol{x}) \oplus \boldsymbol{y}$ instead of $\boldsymbol{\psi}(\boldsymbol{x})$
- **$\boldsymbol{y}$ should be 0 and we need to get it from somewhere**



Restriction

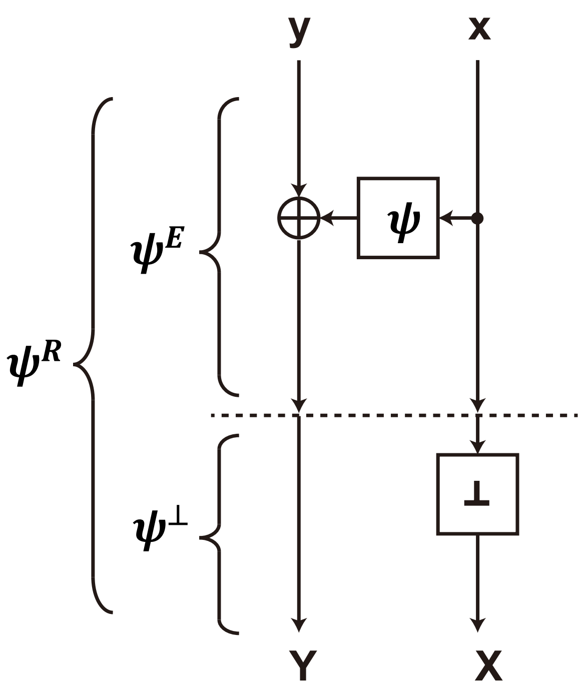
- **Converting the unnecessary output to zero**
- Feeding it to a neighboring mapping as a zero input

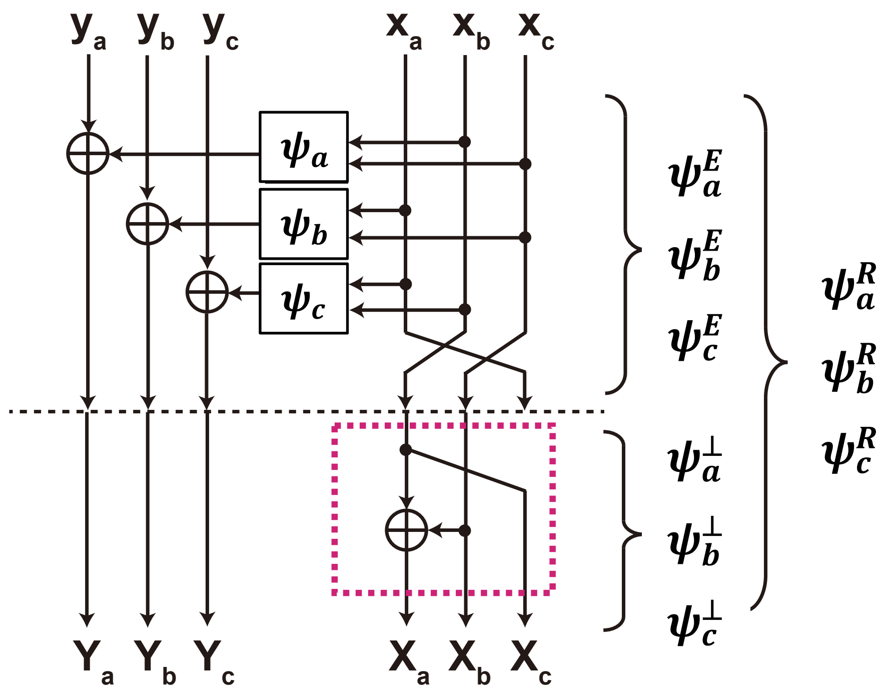


Restriction cont.

- The null mapping ⊥ has a uniform sharing
  - $\{x_a, x_b, x_c\} \mapsto \{x_b \oplus x_c, x_b, x_c,\}$

Converting unnecessary share to another one representing 0



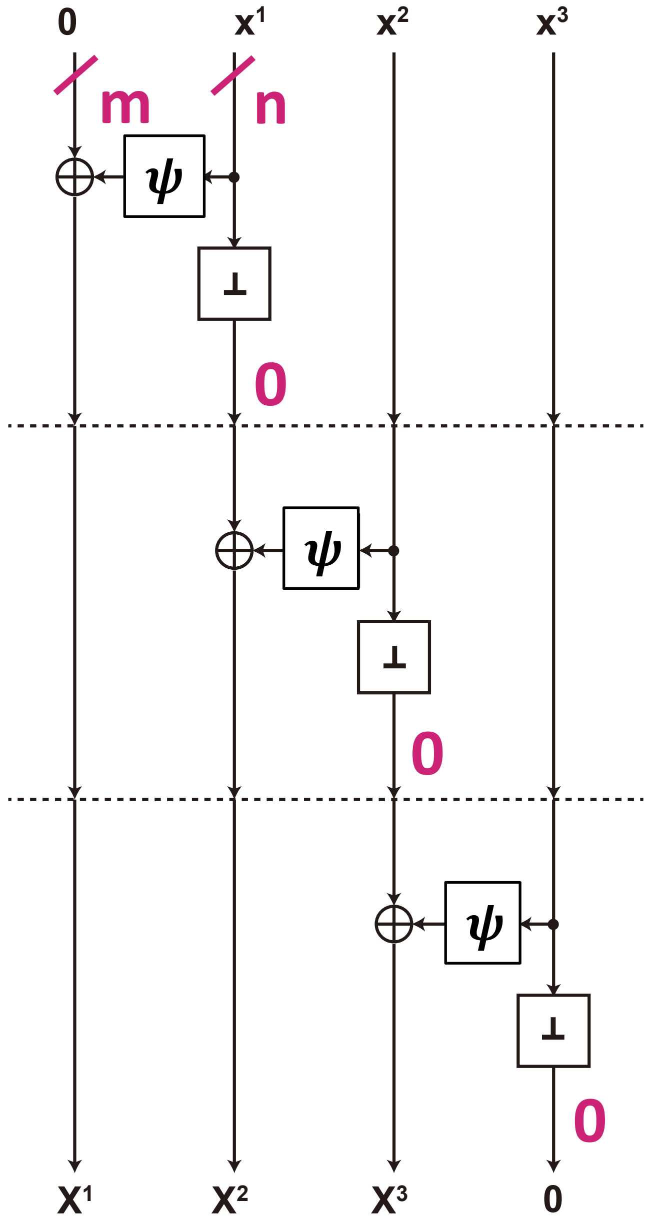


Chaining

- For a target map having the same input and output sizes ($\boldsymbol{m} = \boldsymbol{n}$), we can easily chain zero outputs and inputs

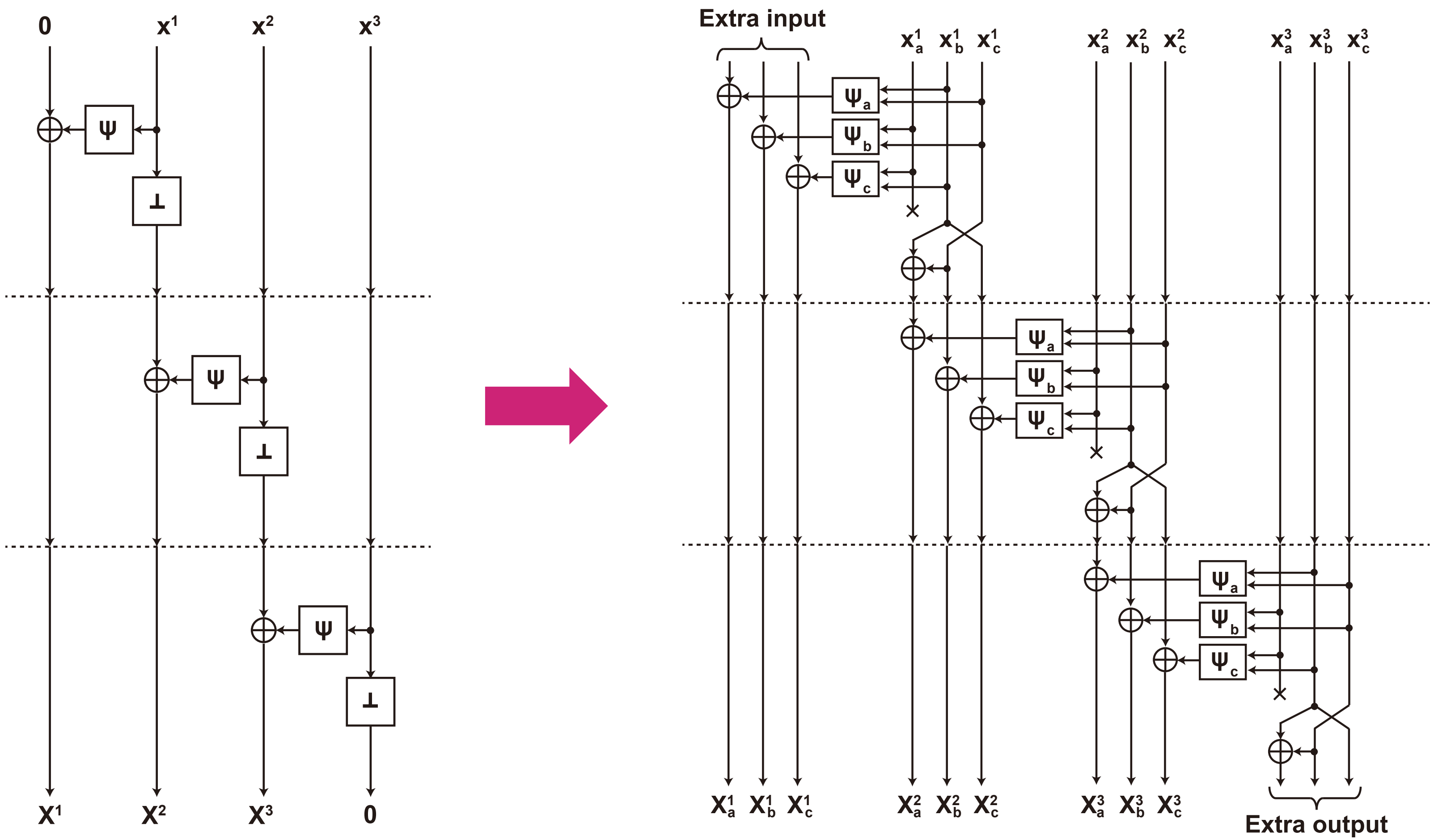
- The right figure shows 3-parallel mapping given by

$$(\mathbf{0}, \boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3) \mapsto (\boldsymbol{\psi}(\boldsymbol{x}^1), \boldsymbol{\psi}(\boldsymbol{x}^2), \boldsymbol{\psi}(\boldsymbol{x}^3), \mathbf{0})$$



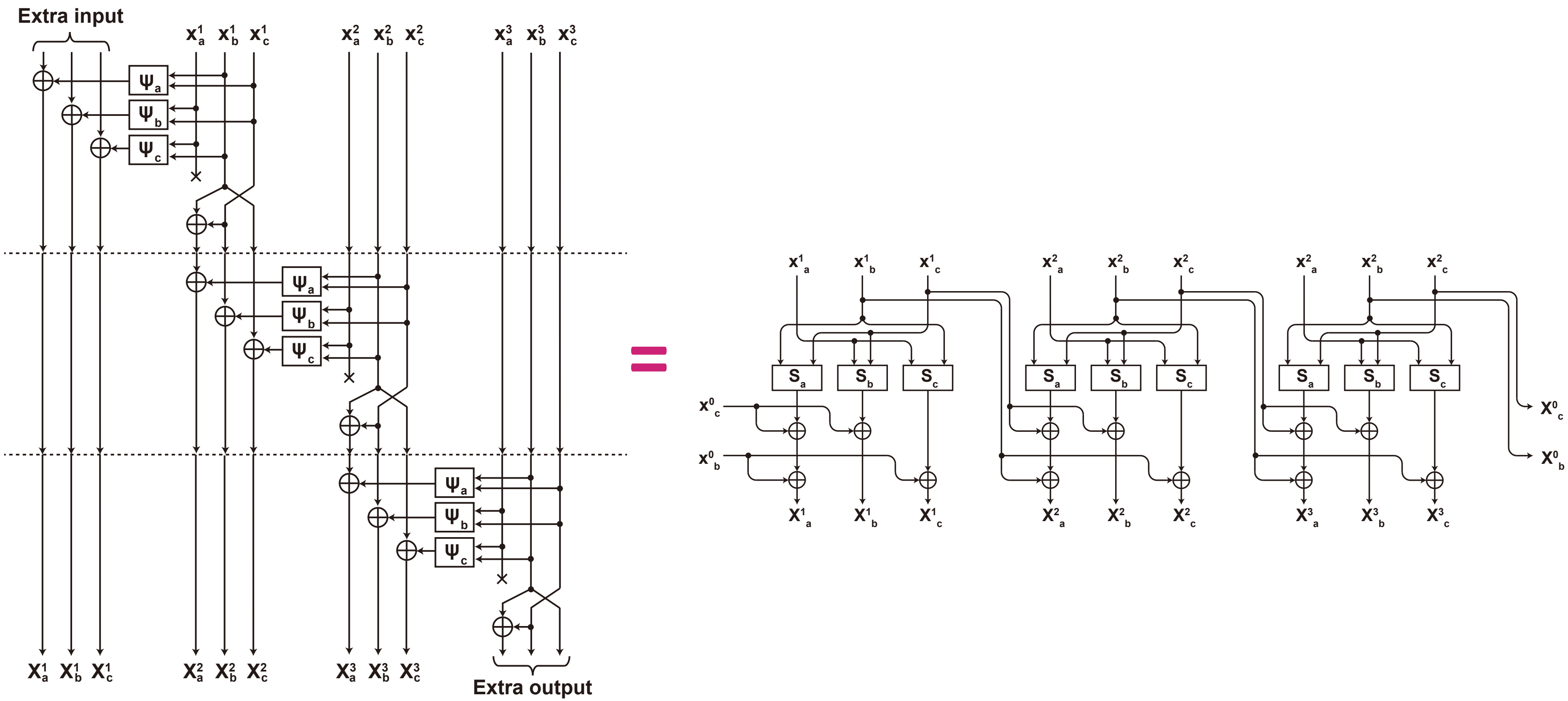
Chaining cont.

- By substituting each $\boldsymbol{\psi}^{\boldsymbol{R}}$ with its sharing, we get a uniform sharing of a layer of parallel $\boldsymbol{\psi}^{\boldsymbol{R}}$s



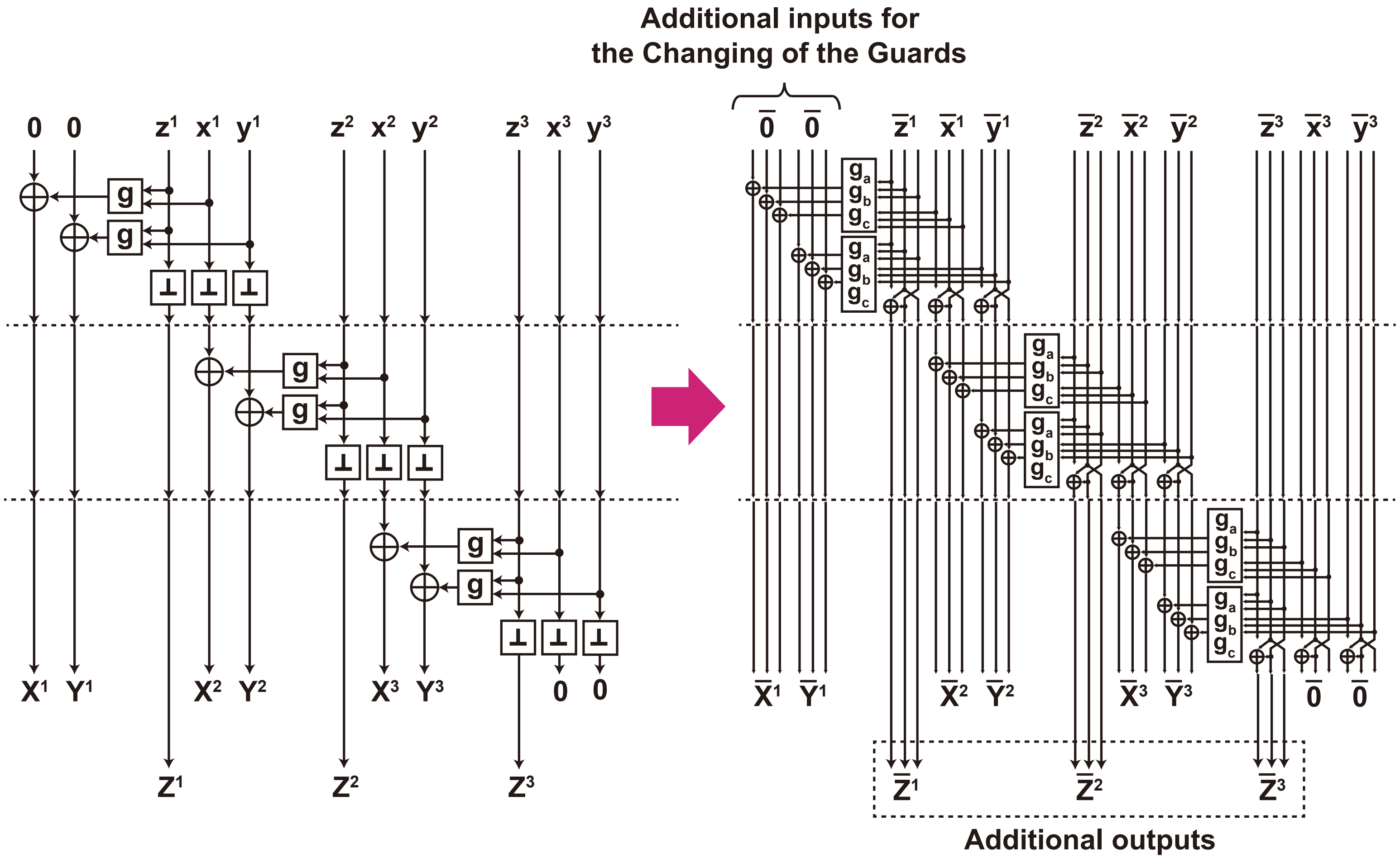
Why it is a generalization of CotG

- **This sharing is the same as Daemen's CotG**
- Now we can also support non-bijective mapping



A map with different input/output sizes

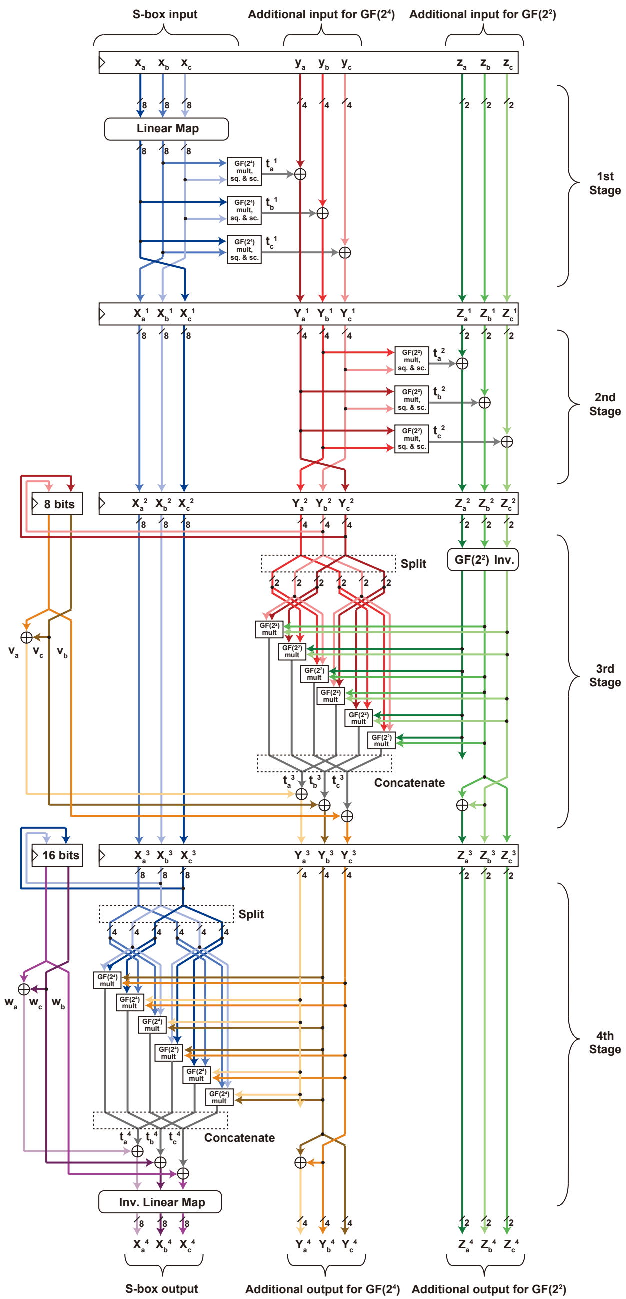
- Input is larger: we get additional zero outputs that we can use later
- Output is larger: we need additional zero inputs



Application to AES S-box

- 4-stage Canright's S-box is expanded to make all the stages uniform
  - + **6-bit** additional input
  - + **6-bit** additional output

- Register overhead ≒ Initial randomness:
  - **6 bits** * 3 shares *16 S-boxes = 288 bits + some more



Conclusion

- A generalization of the Changing of the Guards that supports non-bijective targets

- The first 3-share and uniform threshold implementation of the AES S-box