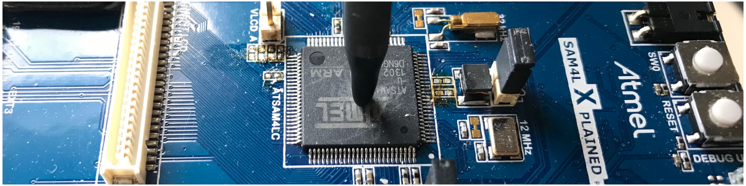


Bitslice Masking and Improved Shuffling: How and When to Mix Them in Software?

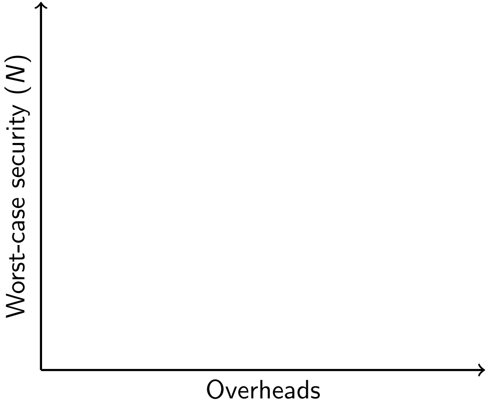
Melissa Azouaoui, Olivier Bronchain, Vincent Grosso
Kostas Papagiannopoulos, François-Xavier Standaert

CHES 2022, Leuven, Belgium

University of Amsterdam NXP UCLouvain

Contents

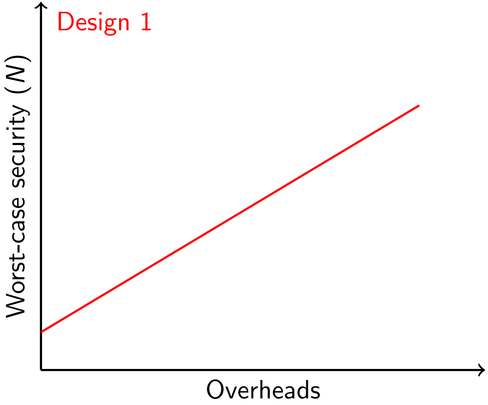
Design space for side-channel countermeasures



Countermeasures compared on:

- Run time overheads.
- Worst-case security (N).

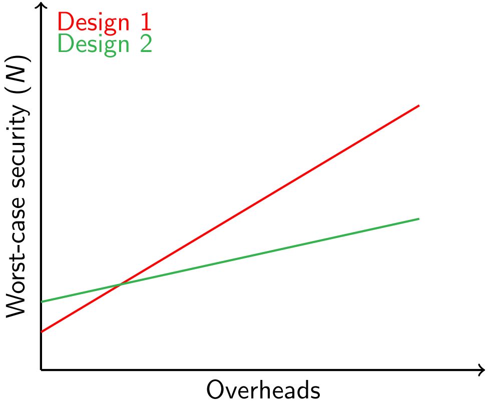
Design space for side-channel countermeasures



Countermeasures compared on:

- Run time overheads.
- Worst-case security (N).

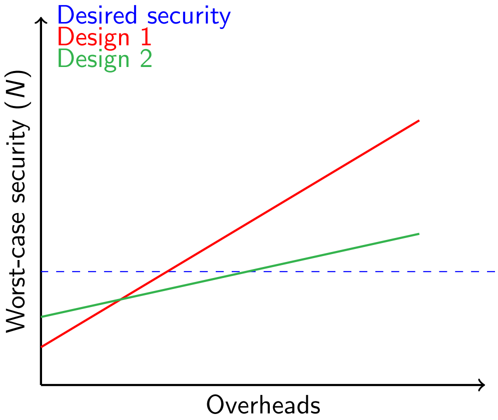
Design space for side-channel countermeasures



Countermeasures compared on:

- Run time overheads.
- Worst-case security (N).

Design space for side-channel countermeasures

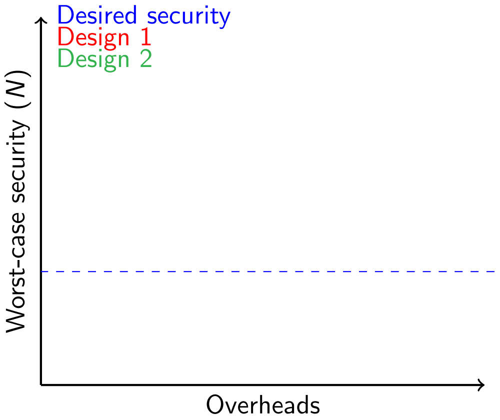


Countermeasures compared on:

- Run time overheads.
- Worst-case security (N).

$\rightarrow$ Best design depends on desired security.

Design space for side-channel countermeasures



Countermeasures compared on:

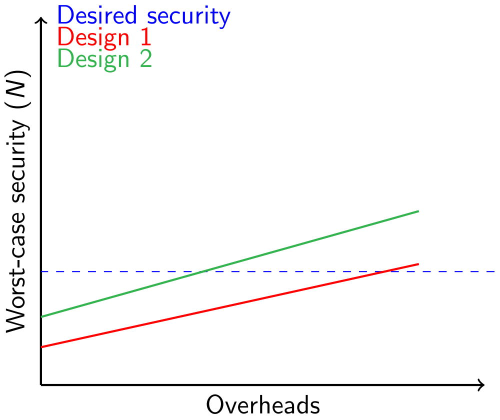
- Run time overheads.
- Worst-case security (N).

$\rightarrow$ Best design depends on desired security.

Best design is device dependent:

- Noise level.
- Platform architecture.

Design space for side-channel countermeasures



Countermeasures compared on:

- Run time overheads.
- Worst-case security (N).

$\rightarrow$ Best design depends on desired security.

Best design is device dependent:

- Noise level.
- Platform architecture.

Existing side-channel countermeasures $\big(\mathrm{MI}(X; L) < 1\big)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i; L)} \approx \frac{c}{\mathrm{MI}(X^i; L)^d}$$

- Data Layout:

0-th share	$\boxed{x^0}$
1-th share	$\boxed{x^1}$
2-th share	$\boxed{x^2}$

Existing side-channel countermeasures $\left(\mathrm{MI}(X;L) < 1\right)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i;L)} \approx \frac{c}{\mathrm{MI}(X^i;L)^d}$$

- Data Layout:

0-th share	x^0
1-th share	x^1
2-th share	x^2

Shuffling:

- Randomized processing order.
- Execution based on a perm. of size η
- Noise addition:

$$N \approx \frac{\eta \cdot c}{\mathrm{MI}(X;L)}$$

- Data Layout:

Perm:					
data:					

indep. data

Existing side-channel countermeasures $\left(\mathrm{MI}(X; L) < 1\right)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i; L)} \approx \frac{c}{\mathrm{MI}(X^i; L)^d}$$

- Data Layout:

0-th share	x^0
1-th share	x^1
2-th share	x^2

Shuffling:

- Randomized processing order.
- Execution based on a perm. of size η
- Noise addition:

$$N \approx \frac{\eta \cdot c}{\mathrm{MI}(X; L)}$$

- Data Layout:

Perm:	2	4	1	3	0
data:			x_2		

$\longleftrightarrow$ indep. data

Existing side-channel countermeasures $\left(\mathrm{MI}(X;L) < 1\right)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i;L)} \approx \frac{c}{\mathrm{MI}(X^i;L)^d}$$

- Data Layout:

0-th share	x^0
1-th share	x^1
2-th share	x^2

Shuffling:

- Randomized processing order.
- Execution based on a perm. of size η
- Noise addition:

$$N \approx \frac{\eta \cdot c}{\mathrm{MI}(X;L)}$$

- Data Layout:

Perm:	2	**4**	1	3	0
data:			x_2		x_4

$\longleftrightarrow$ indep. data

Existing side-channel countermeasures $\left(\mathrm{MI}(X;L) < 1\right)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i;L)} \approx \frac{c}{\mathrm{MI}(X^i;L)^d}$$

- Data Layout:

0-th share	x^0
1-th share	x^1
2-th share	x^2

Shuffling:

- Randomized processing order.
- Execution based on a perm. of size η
- Noise addition:

$$N \approx \frac{\eta \cdot c}{\mathrm{MI}(X;L)}$$

- Data Layout:

Perm:	2	4	**1**	3	0
data:		x_1	x_2		x_4

indep. data

Existing side-channel countermeasures $\left(\mathrm{MI}(X;L) < 1\right)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i;L)} \approx \frac{c}{\mathrm{MI}(X^i;L)^d}$$

- Data Layout:

0-th share	x^0
1-th share	x^1
2-th share	x^2

Shuffling:

- Randomized processing order.
- Execution based on a perm. of size η
- Noise addition:

$$N \approx \frac{\eta \cdot c}{\mathrm{MI}(X;L)}$$

- Data Layout:

Perm:	2	4	1	3	0
data:		x_1	x_2	x_3	x_4

$\longleftrightarrow$ indep. data

Existing side-channel countermeasures $\left(\mathrm{MI}(X;L) < 1\right)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i;L)} \approx \frac{c}{\mathrm{MI}(X^i;L)^d}$$

- Data Layout:

0-th share	x^0
1-th share	x^1
2-th share	x^2

Shuffling:

- Randomized processing order.
- Execution based on a perm. of size η
- Noise addition:

$$N \approx \frac{\eta \cdot c}{\mathrm{MI}(X;L)}$$

- Data Layout:

Perm:	2	4	1	3	**0**
data:	x_0	x_1	x_2	x_3	x_4

$\longleftrightarrow$ indep. data

Existing side-channel countermeasures $\left(\mathrm{MI}(X;L) < 1\right)$

Masking:

- Randomized the data processed.
- Sharing of $x := (x^0, x^1, \ldots, x^{d-1})$
- Noise amplification:

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i;L)} \approx \frac{c}{\mathrm{MI}(X^i;L)^d}$$

- Data Layout:

0-th share	x^0
1-th share	x^1
2-th share	x^2

Shuffling:

- Randomized processing order.
- Execution based on a perm. of size η
- Noise addition:

$$N \approx \frac{\eta \cdot c}{\mathrm{MI}(X;L)}$$

- Data Layout:

Perm:					
data:	x_0	x_1	x_2	x_3	x_4

$\longleftrightarrow$ indep. data

→ **How to amplify shuffling thanks to masking ? (η^d)**

Design space for side-channel countermeasures

1. Security:
   - Explore design space for shuffling + masking.
   - Evaluate the security:
     - Paper & pencil.
     - Confirmed with simulations.

Rivain et al. [RPD09]:

- Linear layers: $\binom{d\cdot\eta}{d}$
- Non-linear layers: η

Design space for side-channel countermeasures

1. Security:
   - Explore design space for shuffling + masking.
   - Evaluate the security:
     - Paper & pencil.
     - Confirmed with simulations.

2. Performances:
   - Explore perf. bitslice and shuffle.
   - Benchmarks on Cortex-M4.

Rivain et al. [RPD09]:

- Linear layers: $\binom{d\cdot\eta}{d}$
- Non-linear layers: η

Design space for side-channel countermeasures

1. Security:
   - Explore design space for shuffling + masking.
   - Evaluate the security:
     - Paper & pencil.
     - Confirmed with simulations.

2. Performances:
   - Explore perf. bitslice and shuffle.
   - Benchmarks on Cortex-M4.

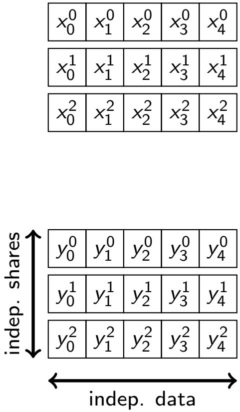
3. Performances vs. security:
   - Pertinence of masking and shuffling combination.

Rivain et al. [RPD09]:

- Linear layers: $\binom{d\cdot\eta}{d}$
- Non-linear layers: η

Contents

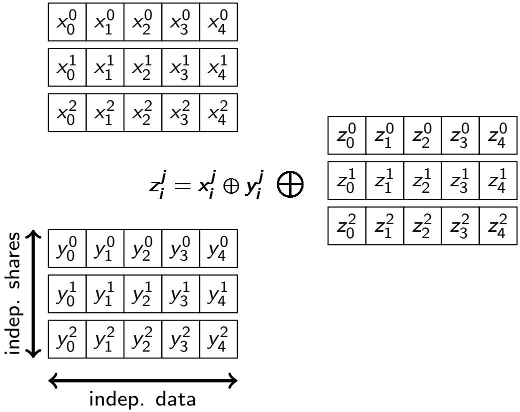
Protecting masked linear layers



Setting:

- $\eta = 5$ independent data x_j and y_j
- $d = 3$ shares x^i and y^i.

Protecting masked linear layers



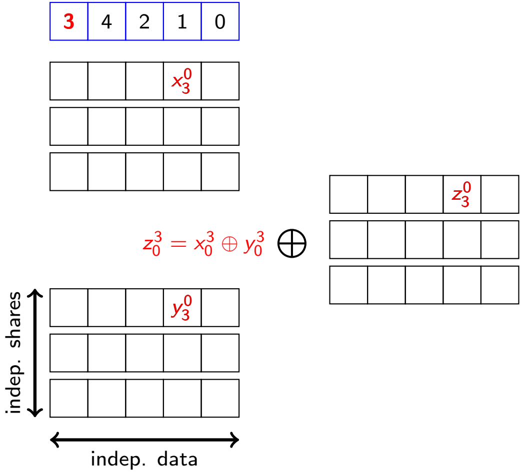
Setting:

- $\eta = 5$ independent data x_j and y_j
- $d = 3$ shares x^i and y^i.

Goal:

- Compute all: $z_j^i = x_j^i \oplus y_j^i$

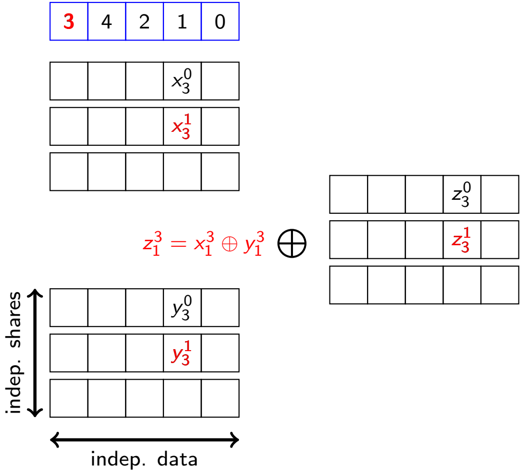
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

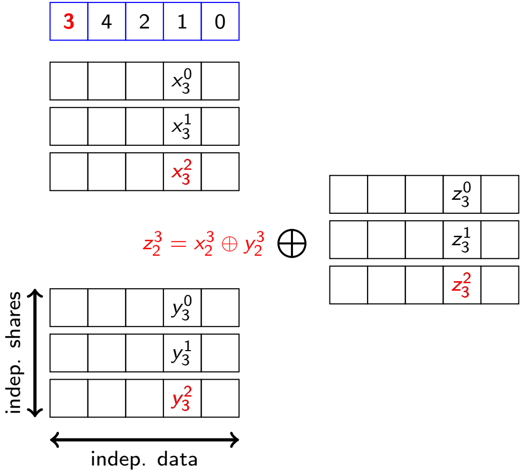
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

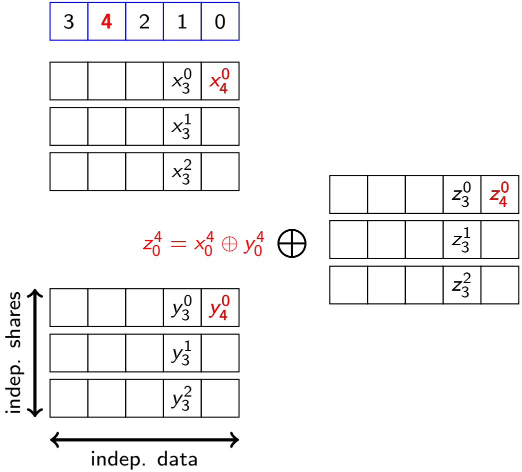
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

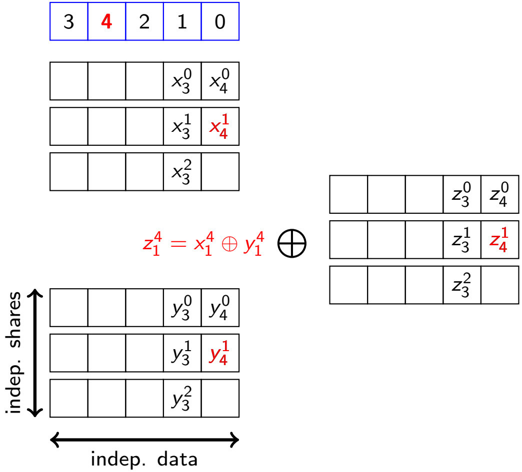
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

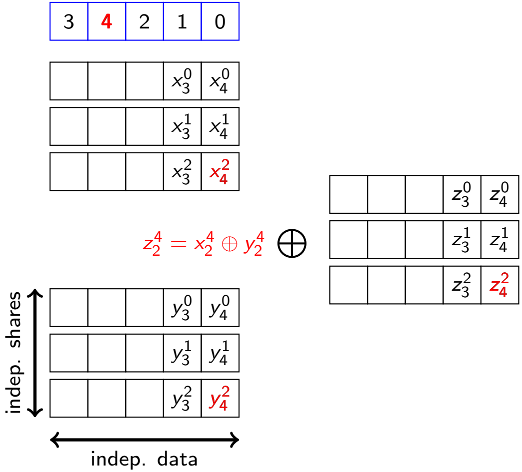
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

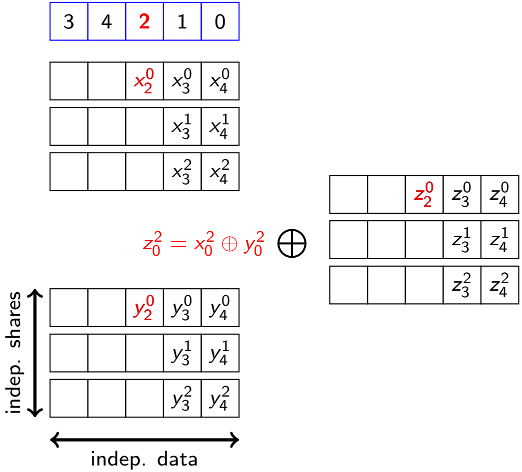
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

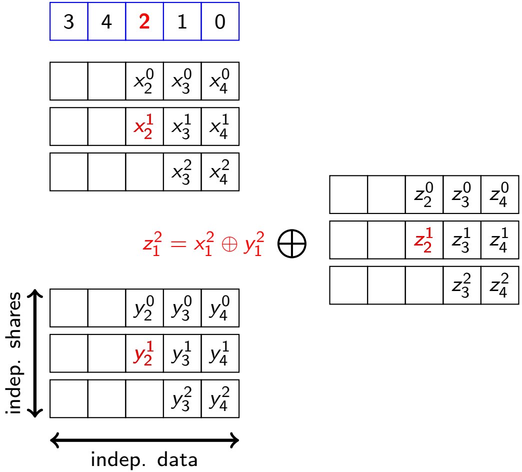
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

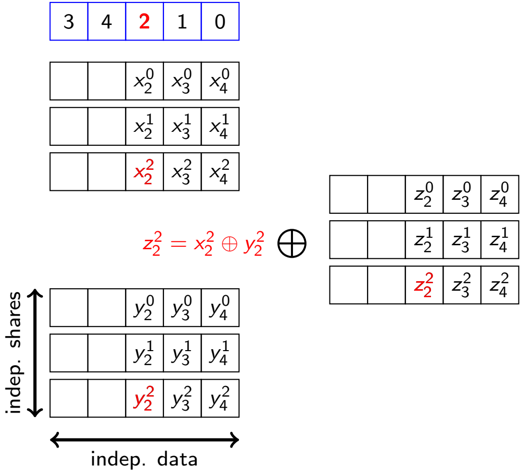
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

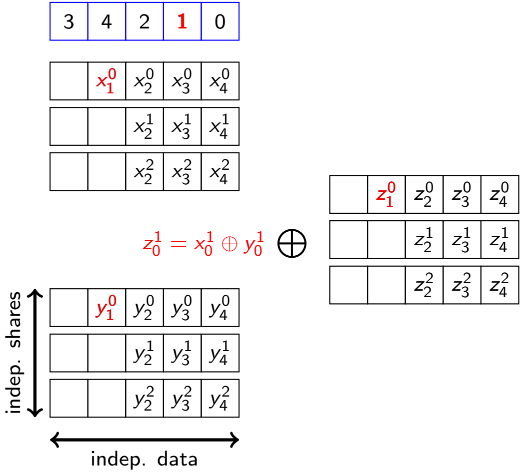
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

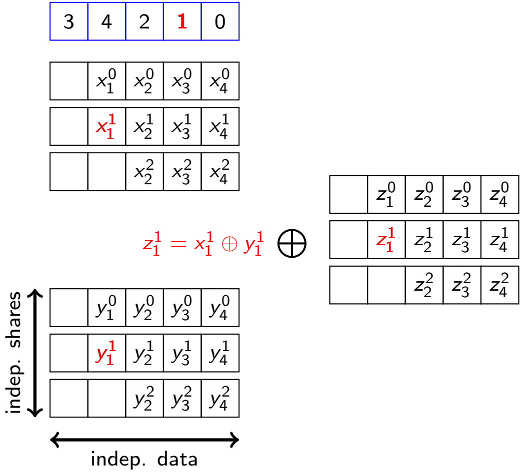
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

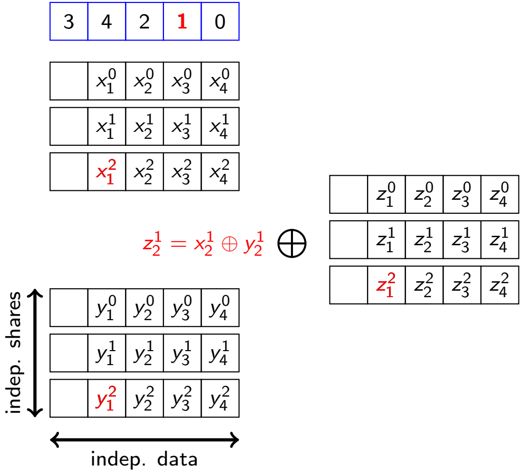
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

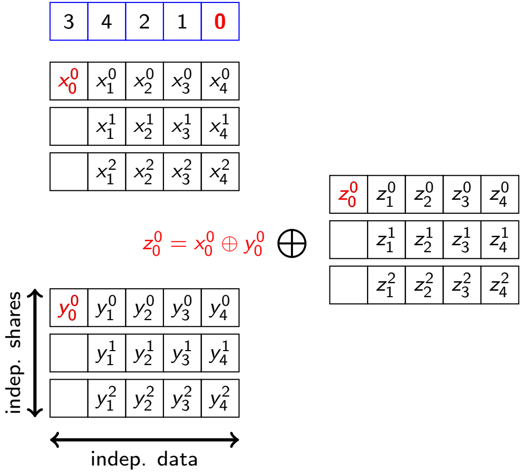
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

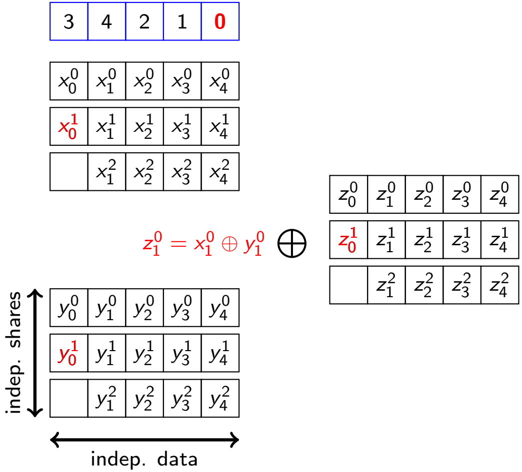
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

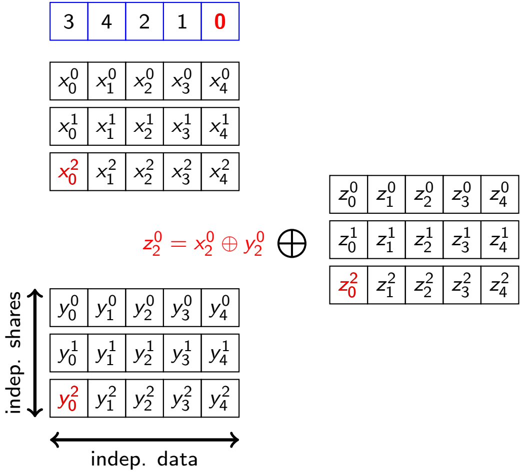
Shuffling-tuples on linear layers: description



Description:

- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

Shuffling-tuples on linear layers: description



Description:

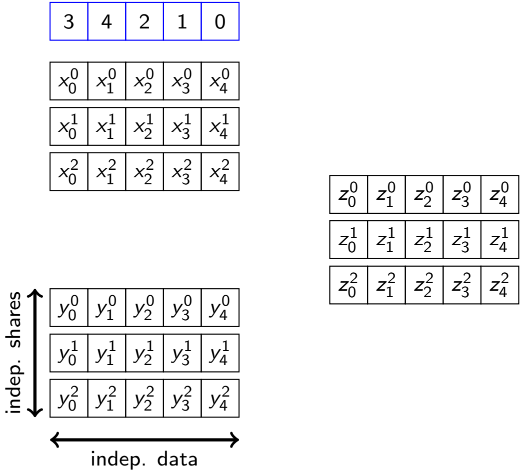
- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

Decrease $\mathrm{MI}(X; L)$ by a factor η.

$$N \approx \frac{c \cdot \eta}{\prod_i \mathrm{MI}(X^i; L)}$$

$\rightarrow$ **Masking does not amplify shuffling.**

Shuffling-tuples on linear layers: description



Description:

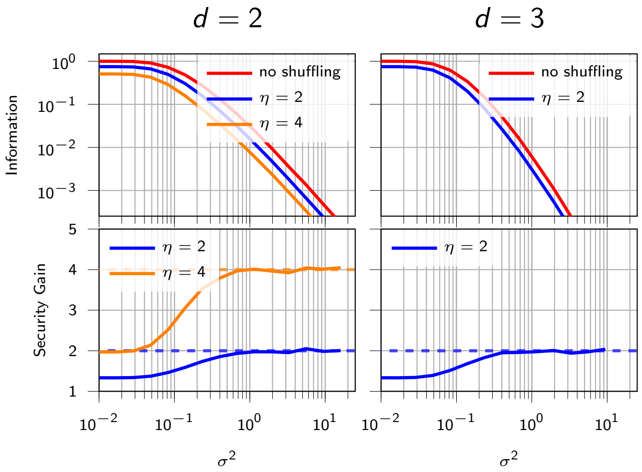
- Shuffle between variables
- Permutations:
  - Number: 1.
  - Size: η.

Decrease $\mathrm{MI}(X; L)$ by a factor η.

$$N \approx \frac{c \cdot \eta}{\prod_i \mathrm{MI}(X^i; L)}$$

$\rightarrow$ **Masking does not amplify shuffling.**

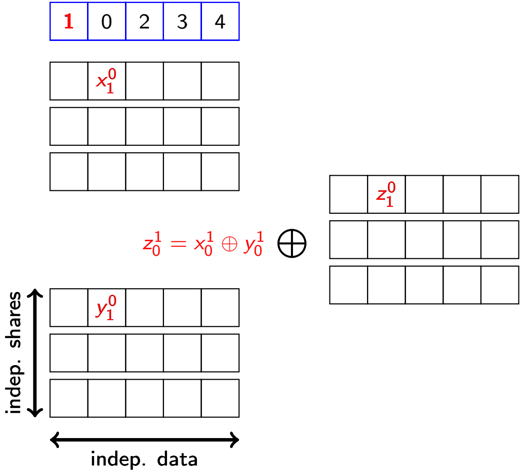
Shuffling-shares on linear layers: simulations



Expected security:

$$N \approx \frac{c \cdot \eta}{\prod_i \mathrm{MI}(X^i; L)}$$

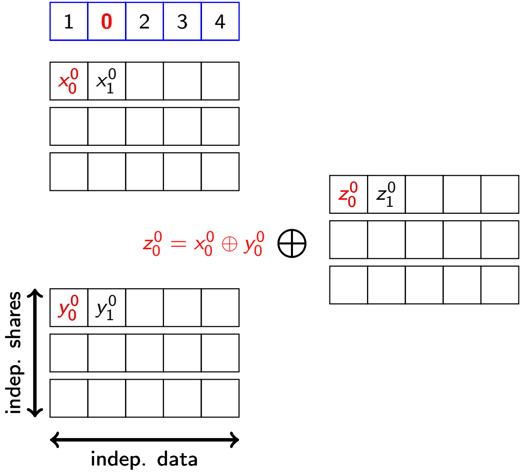
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

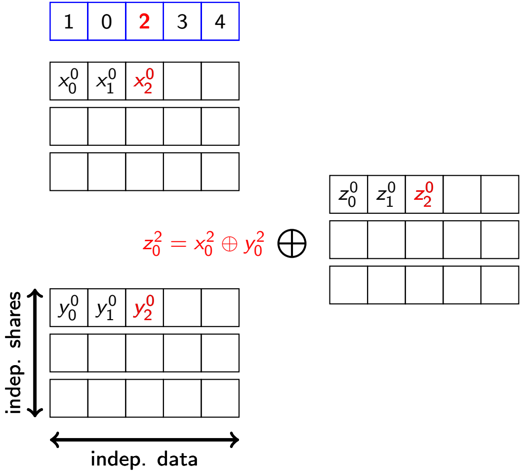
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

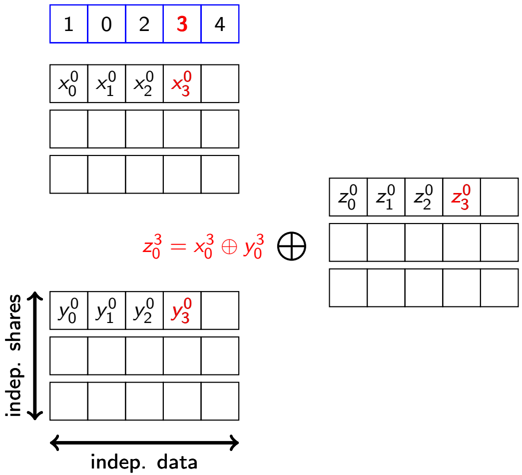
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

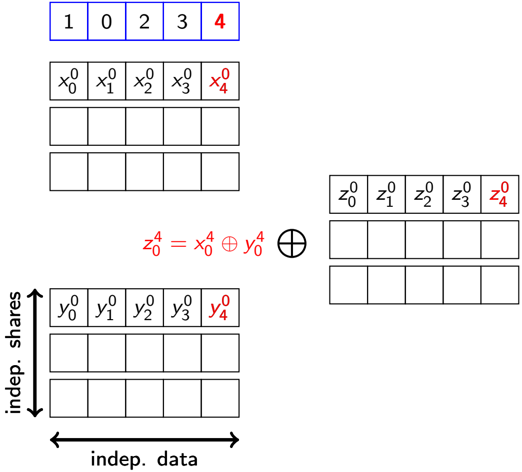
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

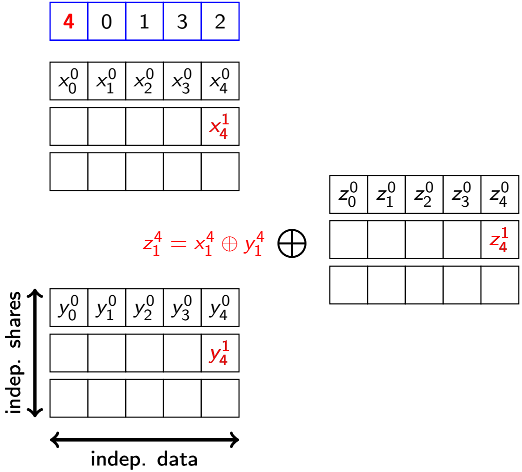
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

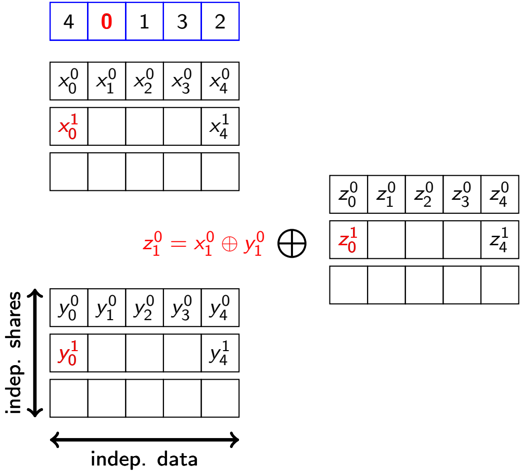
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

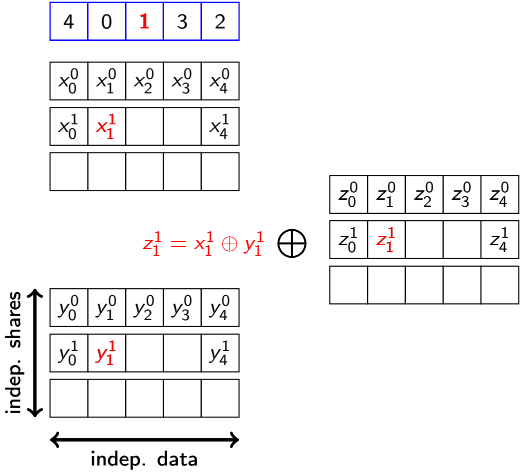
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

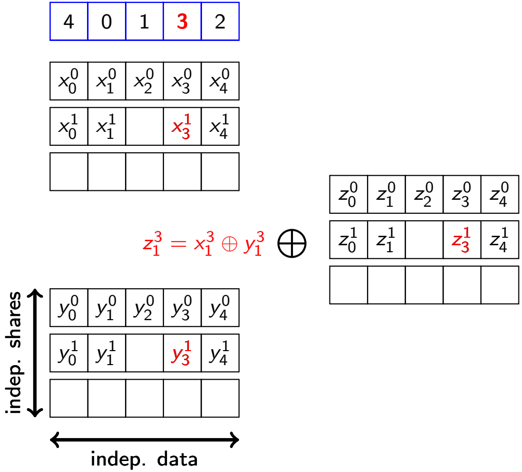
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

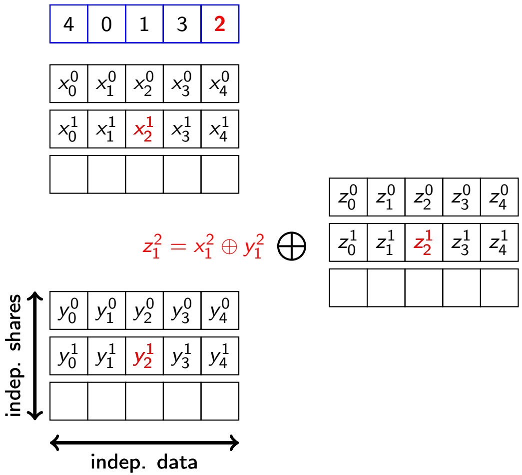
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

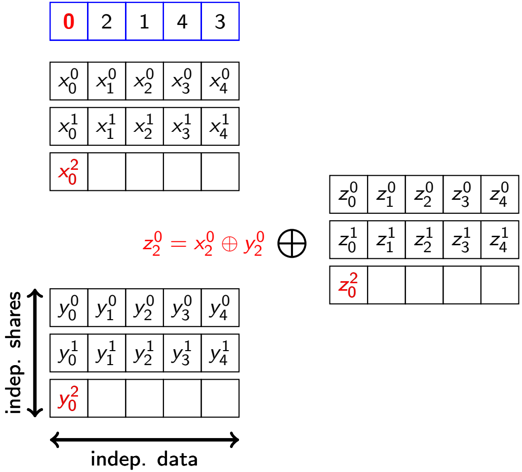
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

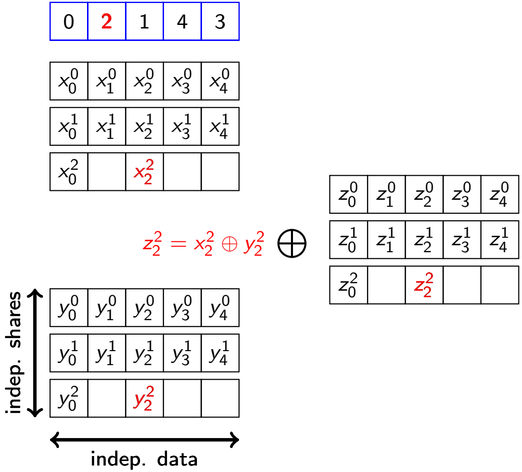
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

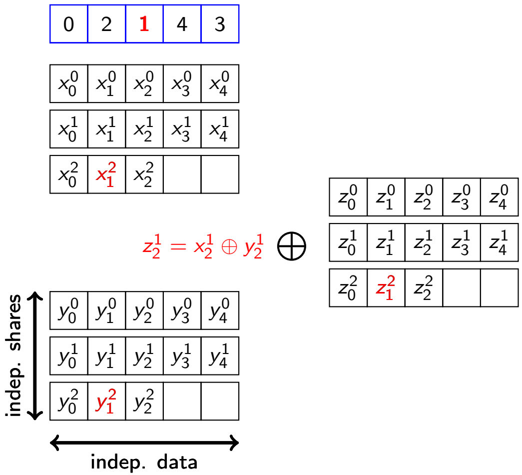
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

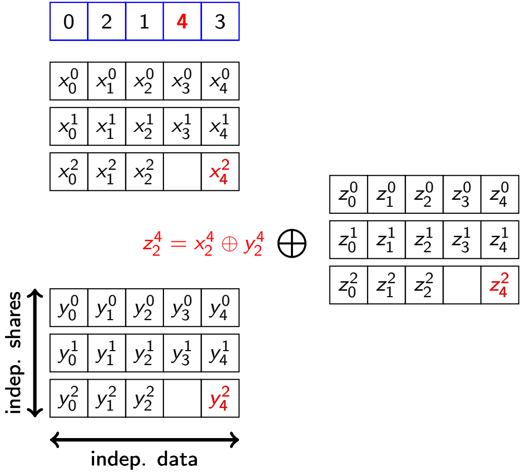
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

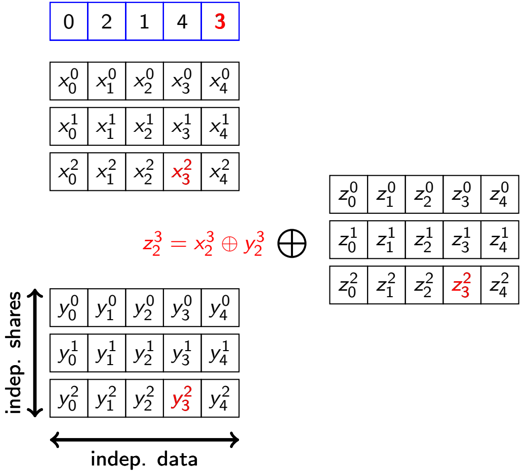
Shuffling-shares on linear layers: description



Description:

- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

Shuffling-shares on linear layers: description



Description:

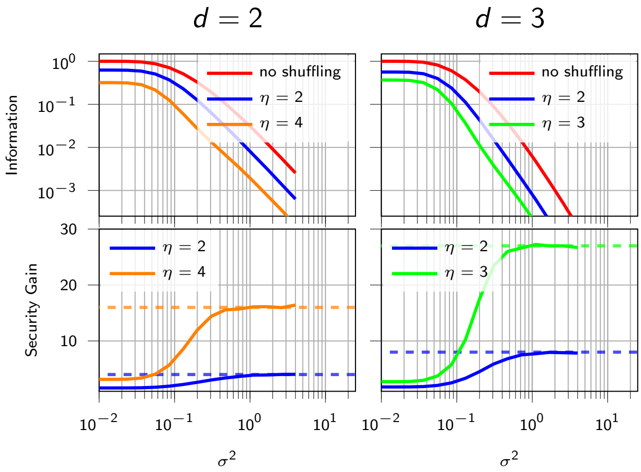
- Shuffle the i-th share of each x_j.
- Permutations:
  - Number: d.
  - Size: η.

Decrease $\mathrm{MI}(X^i; L)$ by a factor η.

$$N \approx \frac{c}{\prod_i \mathrm{MI}(X^i; L)/\eta} \approx \frac{c \cdot \eta^d}{\mathrm{MI}(X^i; L)^d}$$

$\rightarrow$ **Masking amplifies shuffling.**

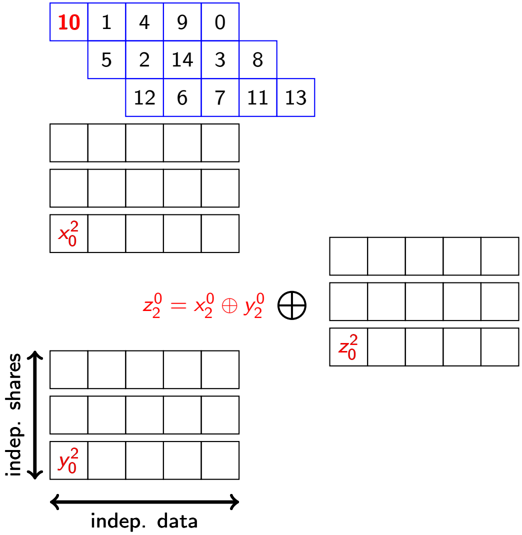
Shuffling-shares on linear layers: simulations



Expected security:

$$N \approx \frac{c \cdot \eta^d}{\mathrm{MI}(X^i; L)^d}$$

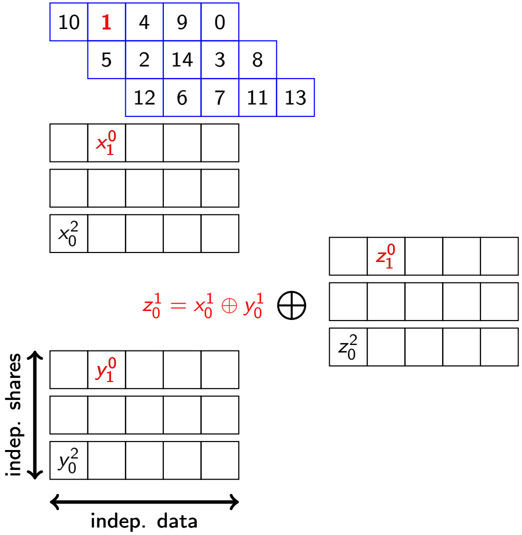
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

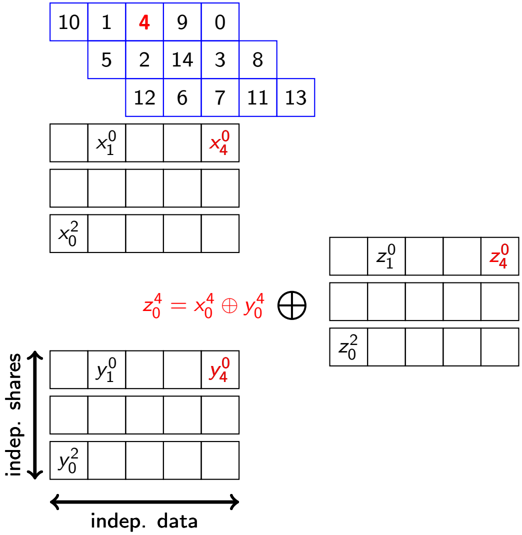
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

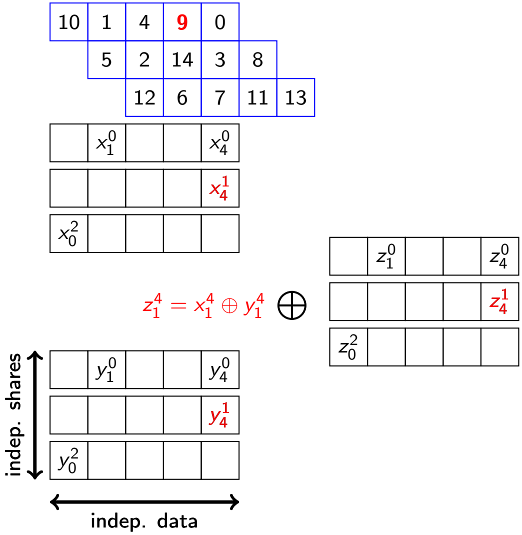
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

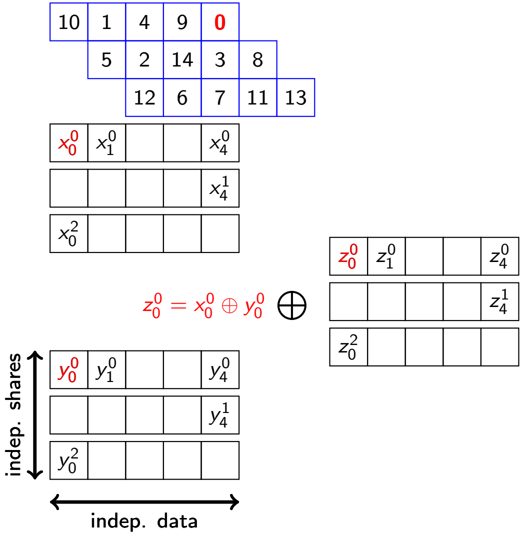
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

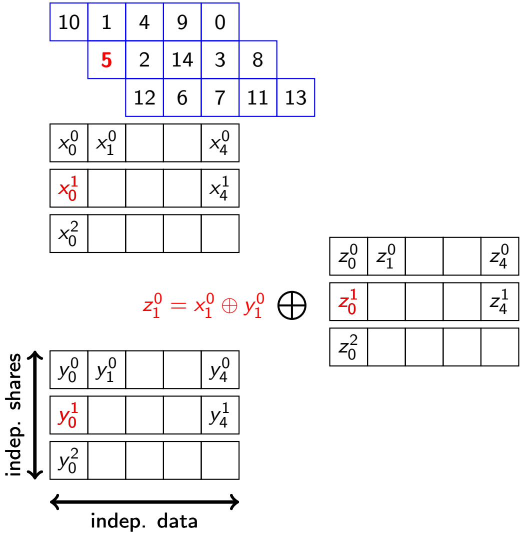
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

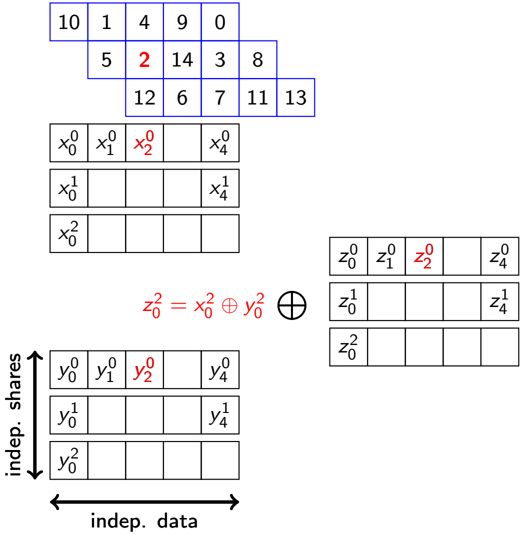
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

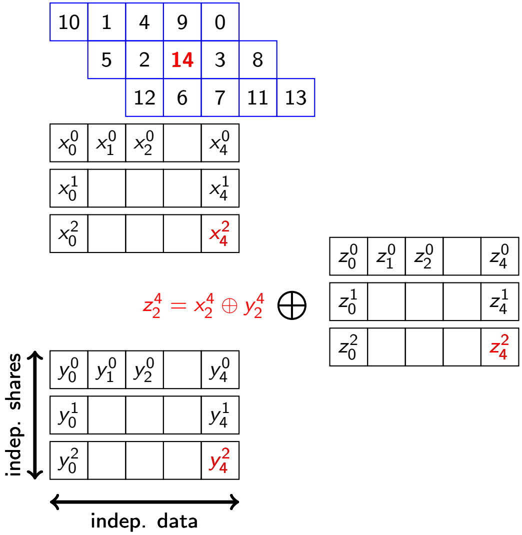
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

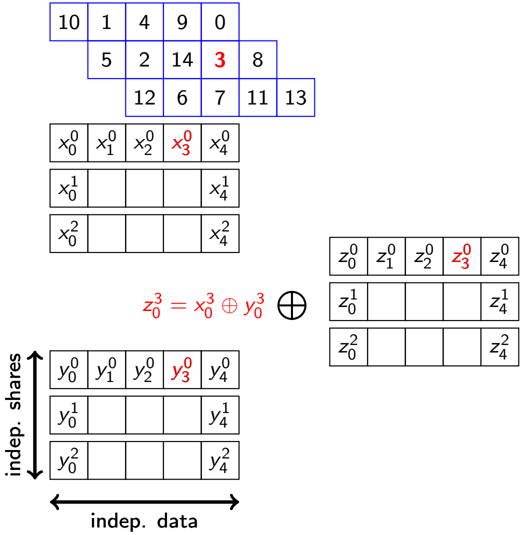
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

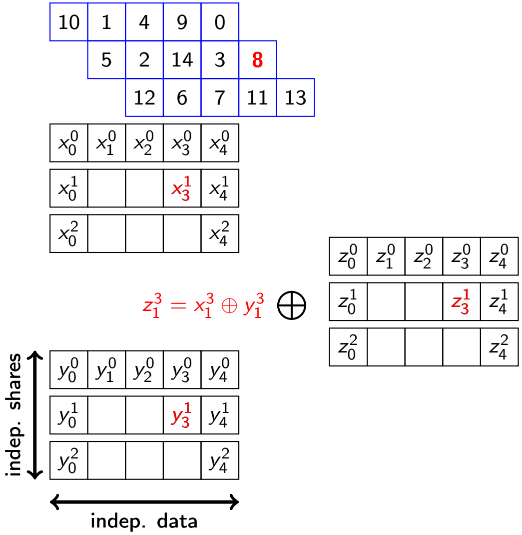
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

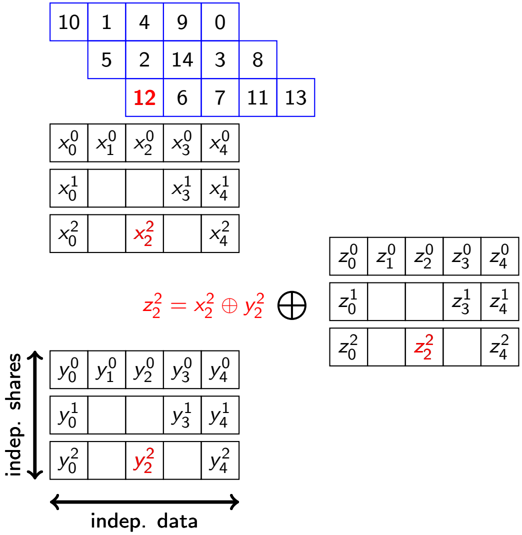
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

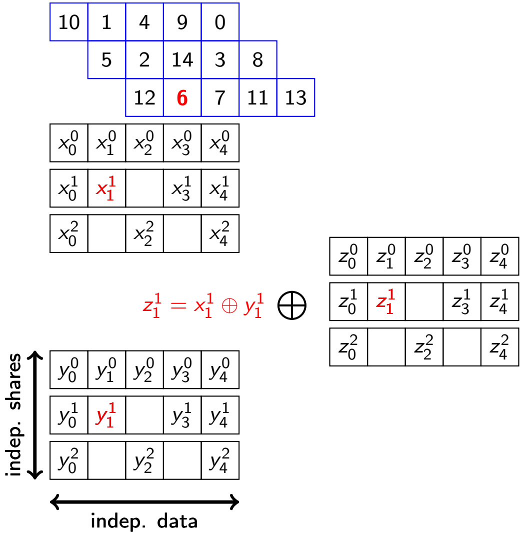
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

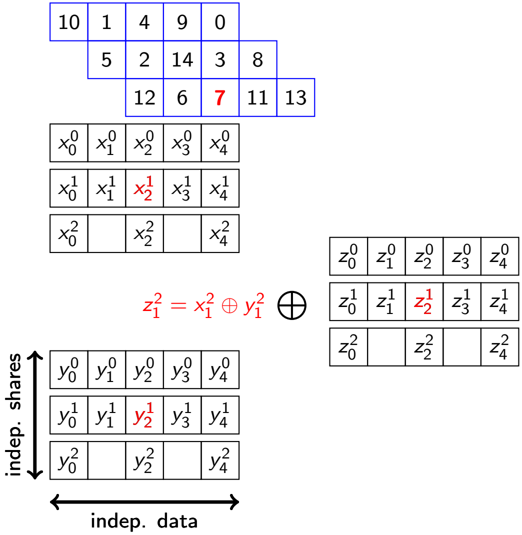
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

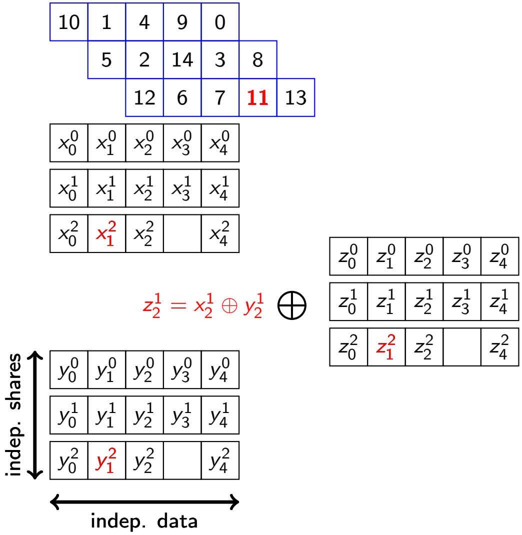
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

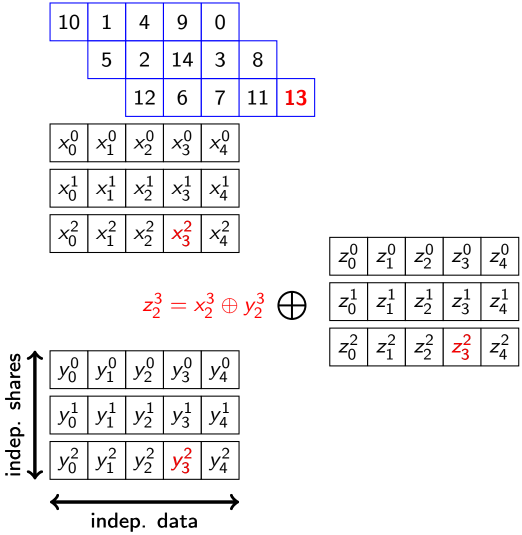
Shuffling-everything on linear layers: description



Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

Shuffling-everything on linear layers: description



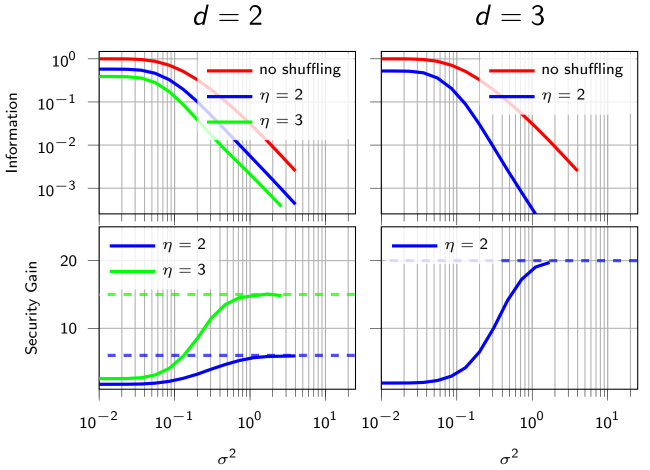
Description:

- Shuffle all the possible operations.
- Permutations:
  - Number: 1.
  - Size: $d \cdot \eta$.

$$N \approx \frac{c \cdot \binom{d \cdot \eta}{d}}{\prod_i \mathrm{MI}(X^i; L)}$$

$\rightarrow$ **Masking amplifies shuffling.**

Shuffling-everything on linear layers: simulations



Expected security:

$$N \approx \frac{c \cdot \binom{d\cdot\eta}{d}}{\prod_i \mathrm{MI}(X^i; L)}$$

Contents

Non-linear layers: summary of the results

For shuffled multiplications:

- Shuffling-shares and shuffling-tuples still apply with similar gain.
- Shuffling-everything could not be analyzed with paper & pencil:
  - Permutation on the output shares is not uniform.

	Linear layer			Non-linear layer		
	Gain	$\|$perm.$\|$	# perm.	Gain	$\|$perm.$\|$	# perm.
shuffling-tuples	η	η	1	η	η	1
shuffling-shares	η^d	η	d	η^d	η	d^2
shuffling-everything	$\binom{d\cdot\eta}{d}$	$d \cdot \eta$	1	?	?	?

Table: Summary of the shuffling + masking combinations.

Non-linear layers: summary of the results

For shuffled multiplications:

- Shuffling-shares and shuffling-tuples still apply with similar gain.
- Shuffling-everything could not be analyzed with paper & pencil:
  - Permutation on the output shares is not uniform.

	Linear layer			Non-linear layer		
	Gain	‖perm.‖	# perm.	Gain	‖perm.‖	# perm.
shuffling-tuples	η	η	1	η	η	1
shuffling-shares	η^d	η	d	η^d	η	d^2
shuffling-everything	$\binom{d\cdot\eta}{d}$	$d \cdot \eta$	1	?	?	?

Table: Summary of the shuffling + masking combinations.

$\rightarrow$ **Next focus on shuffling-shares.**

Contents

Time versus security for shuffled ISW: open questions

Bitslice masking:

- Favors large #ANDs.
- Profits from parallelism.
- Randomness usage:

$$\#\mathrm{AND} \cdot \frac{d \cdot (d-1)}{2}$$

Time versus security for shuffled ISW: open questions

Bitslice masking:

- Favors large #ANDs.
- Profits from parallelism.
- Randomness usage:

$$\#\mathrm{AND} \cdot \frac{d \cdot (d-1)}{2}$$

Shuffling:

- Favors large #ANDs.
- Profits from serialization.
- Randomness usage:

$$d^2 \cdot \eta \cdot \log_2 \eta$$

Time versus security for shuffled ISW: open questions

Bitslice masking:

- Favors large #ANDs.
- Profits from parallelism.
- Randomness usage:

$$\#\text{AND} \cdot \frac{d \cdot (d-1)}{2}$$

Shuffling:

- Favors large #ANDs.
- Profits from serialization.
- Randomness usage:

$$d^2 \cdot \eta \cdot \log_2 \eta$$

Challenges when protecting ISW:

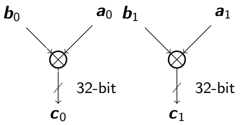
- Should we favor parallelism or serialization.
- Does it depend on the platform ?
- Does it depend on the primitive to protect ?

Time versus security for shuffled ISW: design space (#AND = 64)

Time versus security for shuffled ISW: design space (#AND = 64)

Option 1:

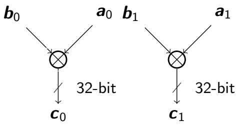
- Only bitsliced ISW.
- 32 bits per reg (full para.).



Time versus security for shuffled ISW: design space (#AND = 64)

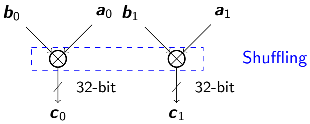
Option 1:

- Only bitsliced ISW.
- 32 bits per reg (full para.).



Option 2:

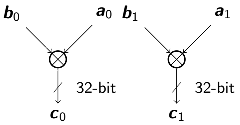
- Shuffled bitsliced ISW.
- 32 bits per reg (full para.).



Time versus security for shuffled ISW: design space (#AND = 64)

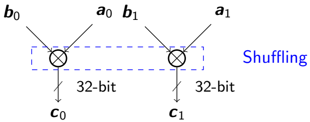
Option 1:

- Only bitsliced ISW.
- 32 bits per reg (full para.).



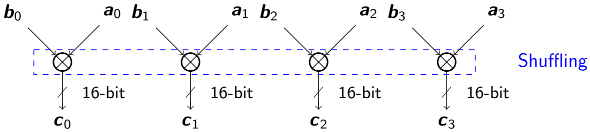
Option 2:

- Shuffled bitsliced ISW.
- 32 bits per reg (full para.).

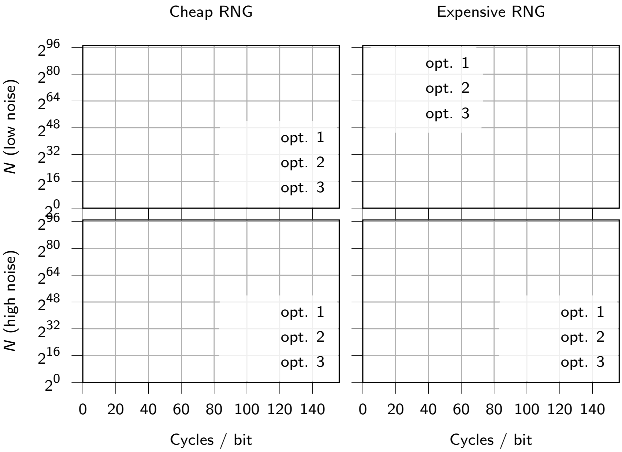


Option 3:

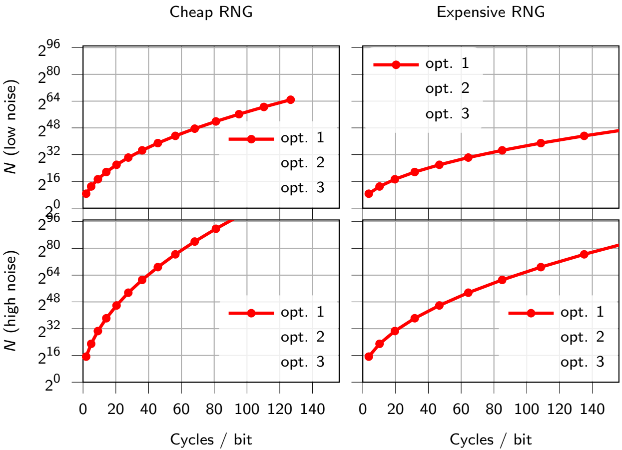
- Shuffled bitsliced ISW.
- 16 bits per reg (inc. ser.).



Time versus security: experimental results

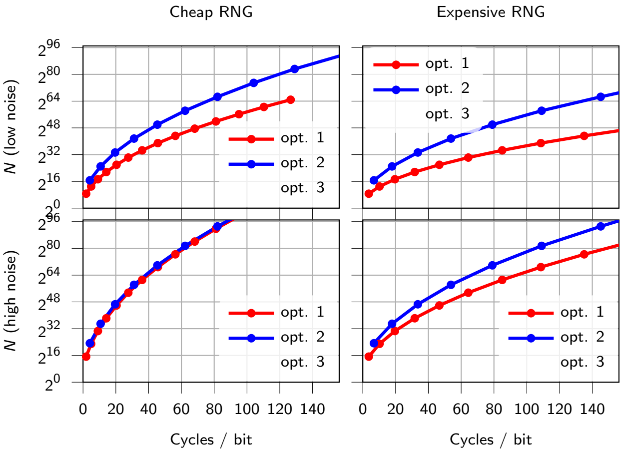


Time versus security: experimental results



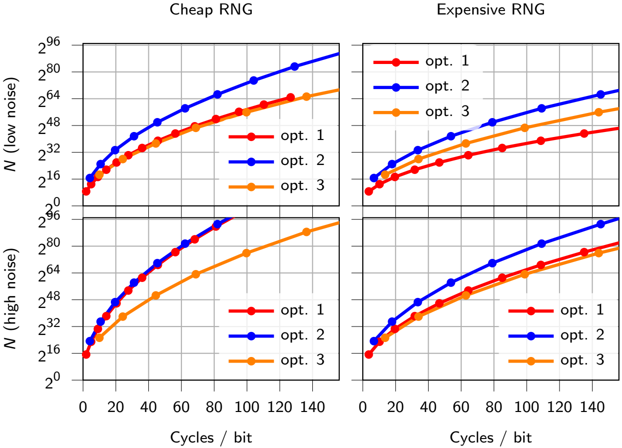
Opt 1: mask. only

Time versus security: experimental results



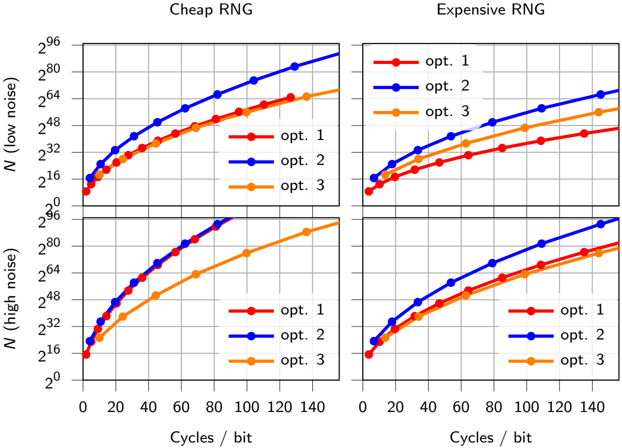
Opt 1: mask. only
Opt 2: mask. & shuffl.

Time versus security: experimental results



Opt 1: mask. only
Opt 2: mask. & shuffl.
Opt 3: lager perm.

Time versus security: experimental results



Opt 1: mask. only
Opt 2: mask. & shuffl.
Opt 3: lager perm.

Take home:
Use fully the registers and then shuffle.

General conclusion for masking and shuffling combination

General conclusion for masking and shuffling combination

General conclusion for masking and shuffling combination

Masking or Masking + shuffling:

- ■: masking is faster.
- □: masking + shuffling is faster.

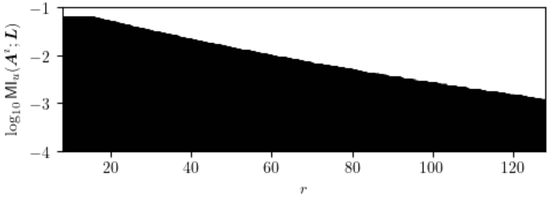


Figure: #AND=128, $N = 2^{64}$

General conclusion for masking and shuffling combination

Masking or Masking + shuffling:

- ■: masking is faster.
- □: masking + shuffling is faster.

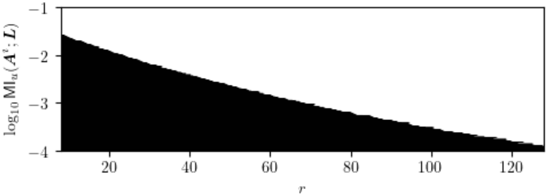


Figure: #AND=256, $N = 2^{64}$

General conclusion for masking and shuffling combination

Masking or Masking + shuffling:

- ■: masking is faster.
- □: masking + shuffling is faster.

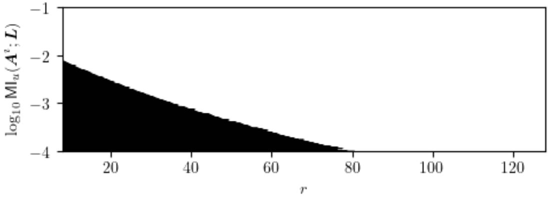


Figure: #AND=512, $N = 2^{64}$

General conclusion for masking and shuffling combination

Masking or Masking + shuffling:

- ■: masking is faster.
- □: masking + shuffling is faster.

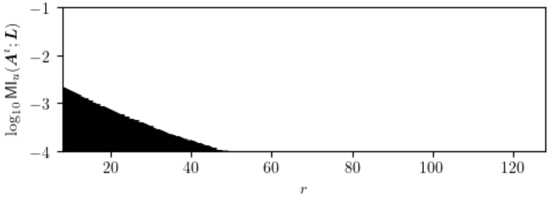


Figure: #AND=1024, $N = 2^{64}$

General conclusion for masking and shuffling combination

When to favor shuffling + masking:

- large of independent $\#AND$.
- expensive randomness r.
- relatively low noise.

Masking or Masking + shuffling:

- ■: masking is faster.
- □: masking + shuffling is faster.

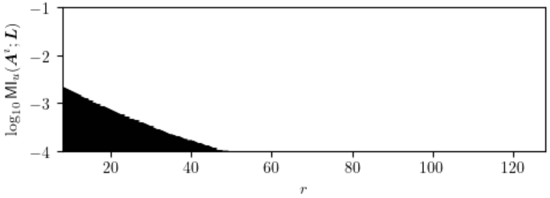


Figure: #AND=1024, $N = 2^{64}$

General conclusion for masking and shuffling combination

When to favor shuffling + masking:

- large of independent $\#AND$.
- expensive randomness r.
- relatively low noise.

Thanks !

Masking or Masking + shuffling:

- ■: masking is faster.
- □: masking + shuffling is faster.

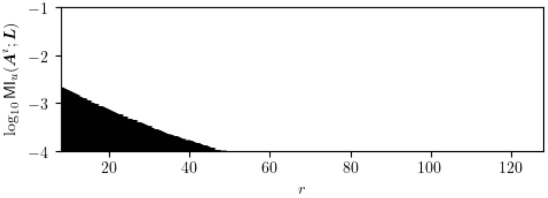


Figure: #AND=1024, $N = 2^{64}$

https://github.com/uclcrypto/bitslice_masking_and_shuffling