Packed Multiplication: How to Amortize the Cost of Side-channel Masking?

Weijia Wang, Chun Guo, François-Xavier Standaert,
Yu Yu, Gaëtan Cassiers

Shandong University, China
Université catholique de Louvain, Belgium
Shanghai Jiao Tong University, China

Dec. 7th, 2020

Table of Contents

Table of Contents

Side-Channel Attacks

- A cryptographic primitive has to be implemented and will run in a given environment.
  - Side-channel leakages: power, time, faults ...
  - *Side-channel attack*:

- *Masking* is one of the most investigated countermeasures against Side-channel attacks
  - Each secret-dependent sensitive variable is randomly encoded into several shares.

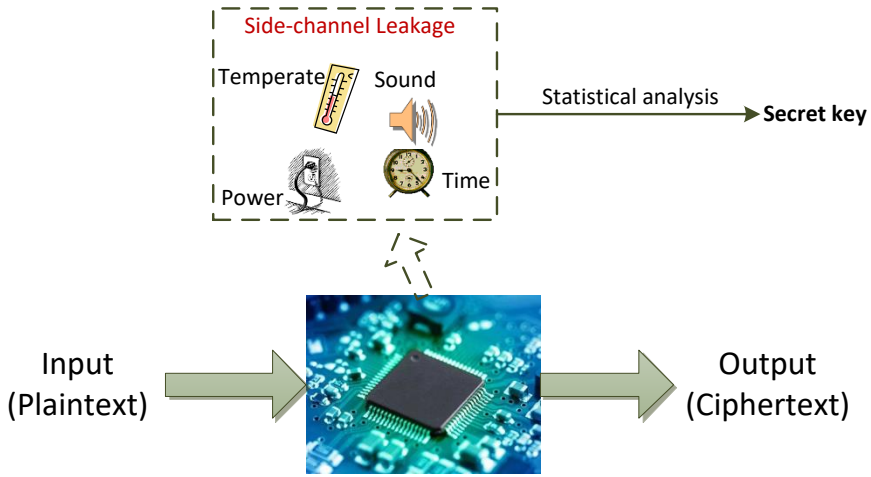
Side-Channel Attacks

- A cryptographic primitive has to be implemented and will run in a given environment.
  - Side-channel leakages: power, time, faults ...
  - *Side-channel attack*:

- *Masking* is one of the most investigated countermeasures against Side-channel attacks
  - Each secret-dependent sensitive variable is randomly encoded into several shares.

Side-Channel Attacks

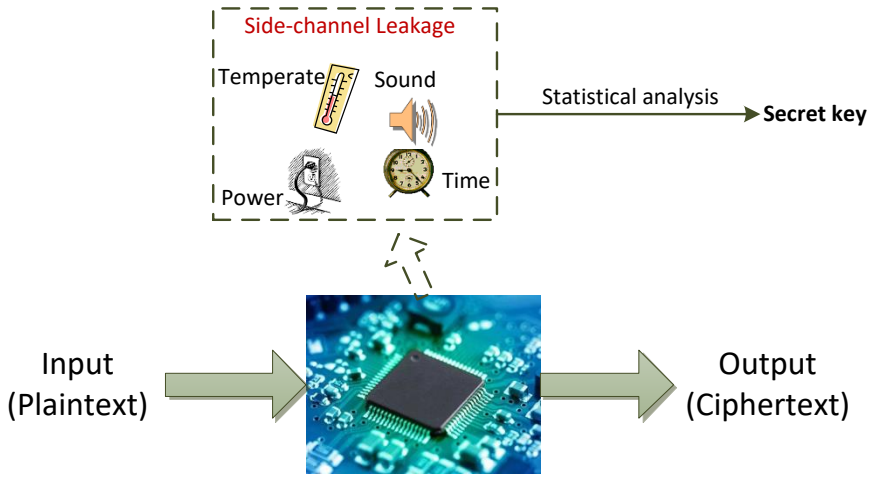
- A cryptographic primitive has to be implemented and will run in a given environment.
  - Side-channel leakages: power, time, faults ...
  - *Side-channel attack*:



- *Masking* is one of the most investigated countermeasures against Side-channel attacks
  - Each secret-dependent sensitive variable is randomly encoded into several shares.

Side-Channel Attacks

- A cryptographic primitive has to be implemented and will run in a given environment.
  - Side-channel leakages: power, time, faults ...
  - *Side-channel attack*:



- *Masking* is one of the most investigated countermeasures against Side-channel attacks
  - Each secret-dependent sensitive variable is randomly encoded into several shares.

Masking

- Randomize the secret (Enc)
  - Secret variable $x \xrightarrow{\text{rand}}$ shares $\hat{x}[1], \ldots, \hat{x}[d]$
    - Boolean masking: $x = \hat{x}[0] \oplus \ldots \oplus \hat{x}[d]$
  - Any d shares *are independent of* x
- Private computations (various gadgets, especially multiplication gadgets).
  - Any d intermediates *are independent of* the input secrets: d-privacy, d-probing security

Masking

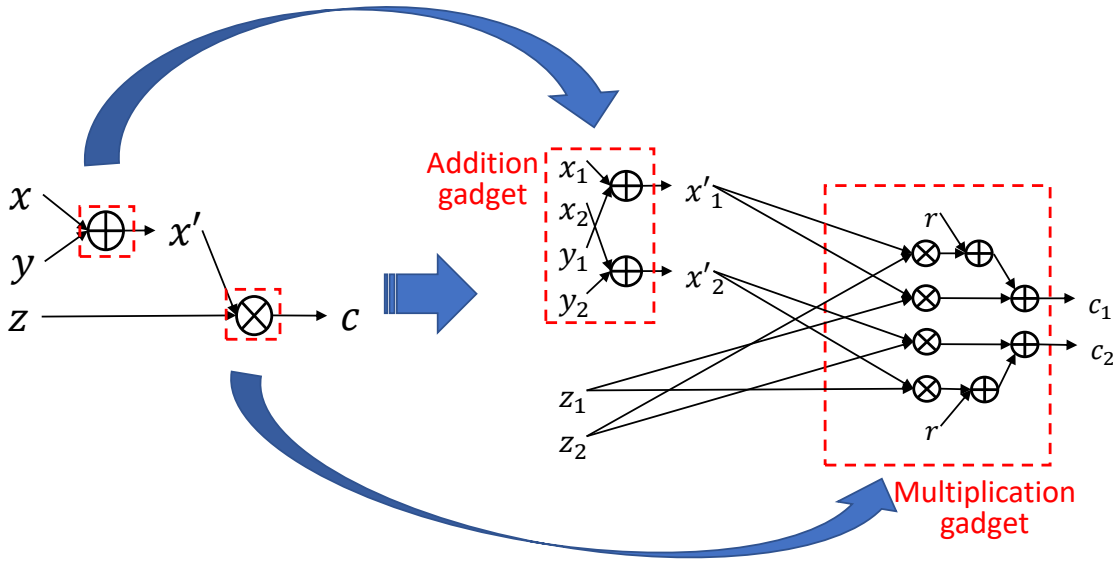
- Randomize the secret (Enc)
  - Secret variable $x \xrightarrow{\text{rand}}$ shares $\hat{x}[1], \ldots, \hat{x}[d]$
    - Boolean masking: $x = \hat{x}[0] \oplus \ldots \oplus \hat{x}[d]$
  - Any d shares *are independent of* x
- Private computations (various gadgets, especially multiplication gadgets).
  - Any d intermediates *are independent of* the input secrets: d-privacy, d-probing security

Masking

- Randomize the secret (Enc)
  - Secret variable $x \xrightarrow{\text{rand}}$ shares $\hat{x}[1], \ldots, \hat{x}[d]$
    - Boolean masking: $x = \hat{x}[0] \oplus \ldots \oplus \hat{x}[d]$
  - Any d shares *are independent of* x
- Private computations (various gadgets, especially multiplication gadgets).
  - Any d intermediates *are independent of* the input secrets: d-privacy, d-probing security

Masking

- Randomize the secret (Enc)
  - Secret variable $x \xrightarrow{\text{rand}}$ shares $\hat{x}[1], \ldots, \hat{x}[d]$
    - Boolean masking: $x = \hat{x}[0] \oplus \ldots \oplus \hat{x}[d]$
  - Any d shares *are independent of* x
- Private computations (various gadgets, especially multiplication gadgets).
  - Any d intermediates *are independent of* the input secrets: d-privacy, d-probing security



Types of Gadgets

- Linear gadgets: trivial implementation.
  - $\mathsf{L}(x) = \mathsf{L}(\hat{x}[0]) \oplus \ldots \oplus \mathsf{L}(\hat{x}[d])$ with L a linear operation.
- Multiplication gadgets example: the ISW multiplication
  - by Yuval **I**shai, Amit **S**ahai and David **W**agner at *CRYPTO '03*.
  - ISW multiplication requires $\frac{d(d+1)}{2}$ random variables and d^2 bilinear multiplications.
    - Complexity: $\mathcal{O}(d^2)$

Types of Gadgets

- Linear gadgets: trivial implementation.
  - $\mathsf{L}(x) = \mathsf{L}(\hat{\mathsf{x}}[0]) \oplus \ldots \oplus \mathsf{L}(\hat{\mathsf{x}}[d])$ with L a linear operation.
- Multiplication gadgets example: the ISW multiplication
  - by Yuval **I**shai, Amit **S**ahai and David **W**agner at *CRYPTO '03*.
  - ISW multiplication requires $\frac{d(d+1)}{2}$ random variables and d^2 bilinear multiplications.
    - Complexity: $\mathcal{O}(d^2)$

Types of Gadgets

- Linear gadgets: trivial implementation.
  - $L(x) = L(\hat{x}[0]) \oplus \ldots \oplus L(\hat{x}[d])$ with L a linear operation.
- Multiplication gadgets example: the ISW multiplication
  - by Yuval **I**shai, Amit **S**ahai and David **W**agner at *CRYPTO '03*.

$\hat{x}[1]\hat{y}[1]$	$\hat{x}[1]\hat{y}[2]$	$\hat{x}[1]\hat{y}[3]$
$\hat{x}[2]\hat{y}[1]$	$\hat{x}[2]\hat{y}[2]$	$\hat{x}[2]\hat{y}[3]$
$\hat{x}[3]\hat{y}[1]$	$\hat{x}[3]\hat{y}[2]$	$\hat{x}[3]\hat{y}[3]$

  - ISW multiplication requires $\frac{d(d+1)}{2}$ random variables and d^2 bilinear multiplications.
    - Complexity: $\mathcal{O}(d^2)$

Types of Gadgets

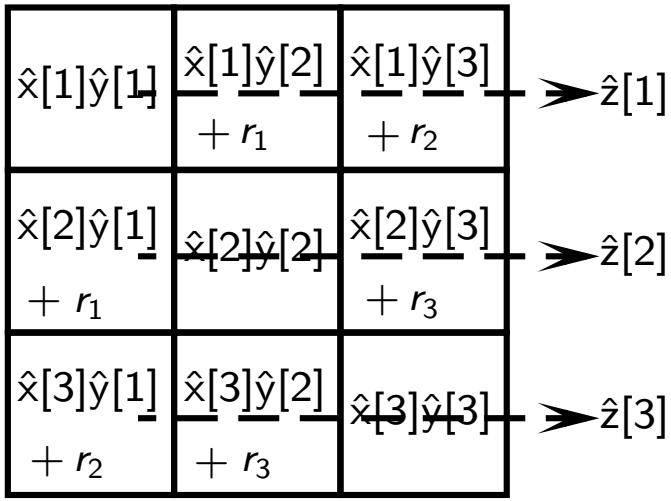
- Linear gadgets: trivial implementation.
  - $\mathsf{L}(x) = \mathsf{L}(\hat{x}[0]) \oplus \ldots \oplus \mathsf{L}(\hat{x}[d])$ with L a linear operation.
- Multiplication gadgets example: the ISW multiplication
  - by Yuval **I**shai, Amit **S**ahai and David **W**agner at *CRYPTO '03*.

$\hat{x}[1]\hat{y}[1]$	$\hat{x}[1]\hat{y}[2] + r_1$	$\hat{x}[1]\hat{y}[3] + r_2$
$\hat{x}[2]\hat{y}[1] + r_1$	$\hat{x}[2]\hat{y}[2]$	$\hat{x}[2]\hat{y}[3] + r_3$
$\hat{x}[3]\hat{y}[1] + r_2$	$\hat{x}[3]\hat{y}[2] + r_3$	$\hat{x}[3]\hat{y}[3]$

  - ISW multiplication requires $\frac{d(d+1)}{2}$ random variables and d^2 bilinear multiplications.
    - Complexity: $\mathcal{O}(d^2)$

Types of Gadgets

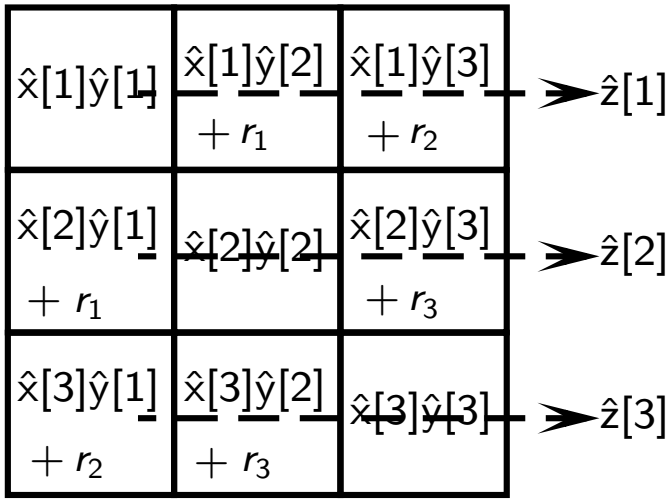
- Linear gadgets: trivial implementation.
  - $\mathsf{L}(x) = \mathsf{L}(\hat{x}[0]) \oplus \ldots \oplus \mathsf{L}(\hat{x}[d])$ with L a linear operation.
- Multiplication gadgets example: the ISW multiplication
  - by Yuval **I**shai, Amit **S**ahai and David **W**agner at *CRYPTO '03*.



  - ISW multiplication requires $\frac{d(d+1)}{2}$ random variables and d^2 bilinear multiplications.
    - Complexity: $\mathcal{O}(d^2)$

Types of Gadgets

- Linear gadgets: trivial implementation.
  - $\mathsf{L}(x) = \mathsf{L}(\hat{x}[0]) \oplus \ldots \oplus \mathsf{L}(\hat{x}[d])$ with L a linear operation.
- Multiplication gadgets example: the ISW multiplication
  - by Yuval **I**shai, Amit **S**ahai and David **W**agner at *CRYPTO '03*.



  - ISW multiplication requires $\frac{d(d+1)}{2}$ random variables and d^2 bilinear multiplications.
    - Complexity: $\mathcal{O}(d^2)$

Reducing implementation overheads

- A challenge regarding the practice usage is the implementation complexity including:
  - computational complexity, and
  - randomness complexity.
- A natural idea: to concentrate on designing better gadget.
  - 'Local' optimization.
  - Isolating approach: considers every gadget isolatedly.
- Our solution: 'Global' optimization.
  - The amortization of multiple parallel multiplications.
    - The more parallel multiplications, the lower averaged cost.
  - Packed approach:
    - computes ℓ masked multiplications in parallel for $\ell \geq 1$.
    - The computational complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2 + d\ell + \ell)$
    - The randomness complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2)$

Reducing implementation overheads

- A challenge regarding the practice usage is the implementation complexity including:
  - computational complexity, and
  - randomness complexity.
- A natural idea: to concentrate on designing better gadget.
  - 'Local' optimization.
  - Isolating approach: considers every gadget isolatedly.
- Our solution: 'Global' optimization.
  - The amortization of multiple parallel multiplications.
    - The more parallel multiplications, the lower averaged cost.
  - Packed approach:
    - computes ℓ masked multiplications in parallel for $\ell \geq 1$.
    - The computational complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2 + d\ell + \ell)$
    - The randomness complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2)$

Reducing implementation overheads

- A challenge regarding the practice usage is the implementation complexity including:
  - computational complexity, and
  - randomness complexity.
- A natural idea: to concentrate on designing better gadget.
  - 'Local' optimization.
  - Isolating approach: considers every gadget isolatedly.
- Our solution: 'Global' optimization.
  - The amortization of multiple parallel multiplications.
    - The more parallel multiplications, the lower averaged cost.
  - Packed approach:
    - computes ℓ masked multiplications in parallel for $\ell \geq 1$.
    - The computational complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2 + d\ell + \ell)$
    - The randomness complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2)$

Reducing implementation overheads

- A challenge regarding the practice usage is the implementation complexity including:
  - computational complexity, and
  - randomness complexity.
- A natural idea: to concentrate on designing better gadget.
  - 'Local' optimization.
  - Isolating approach: considers every gadget isolatedly.
- Our solution: 'Global' optimization.
  - The amortization of multiple parallel multiplications.
    - The more parallel multiplications, the lower averaged cost.
  - Packed approach:
    - computes ℓ masked multiplications in parallel for $\ell \geq 1$.
    - The computational complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2 + d\ell + \ell)$
    - The randomness complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2)$

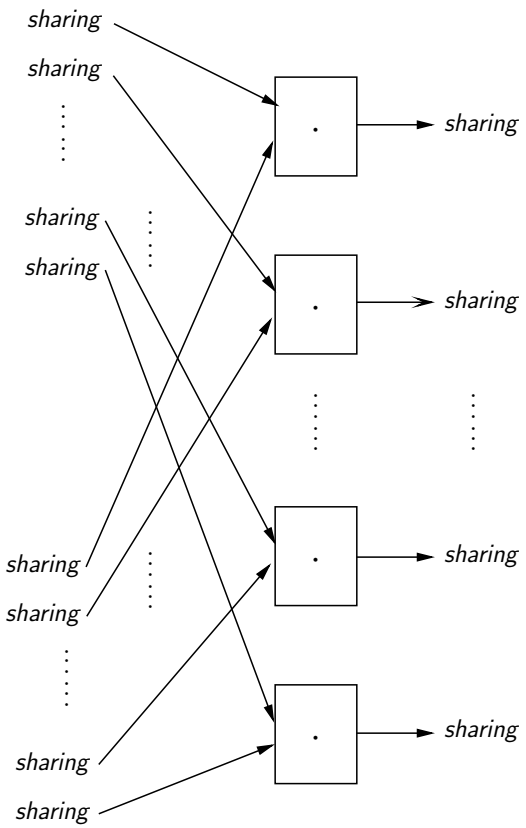
Reducing implementation overheads

- A challenge regarding the practice usage is the implementation complexity including:
  - computational complexity, and
  - randomness complexity.
- A natural idea: to concentrate on designing better gadget.
  - 'Local' optimization.
  - Isolating approach: considers every gadget isolatedly.
- Our solution: 'Global' optimization.
  - The amortization of multiple parallel multiplications.
    - The more parallel multiplications, the lower averaged cost.
  - Packed approach:
    - computes ℓ masked multiplications in parallel for $\ell \geq 1$.
    - The computational complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2 + d\ell + \ell)$
    - The randomness complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2)$

Isolating vs. Packed Approaches

sharing: the list of shares, e.g., $\hat{x}[0], \ldots, \hat{x}[d]$

ℓ: the number of multiplications

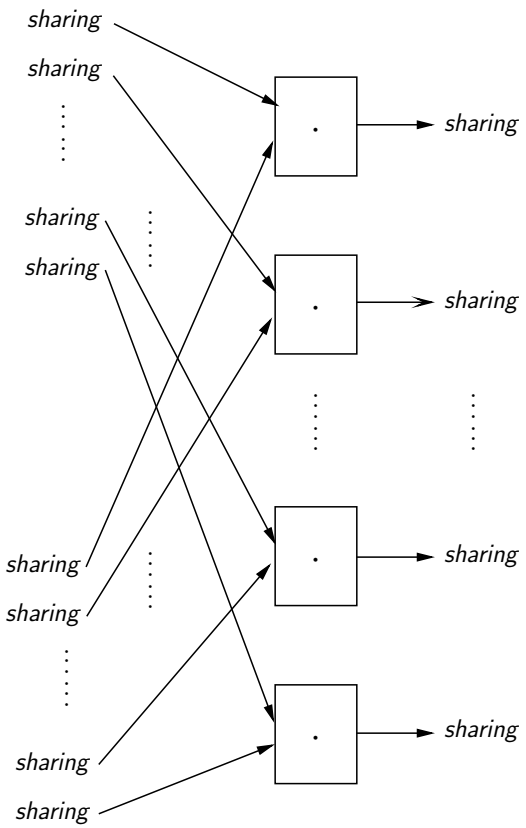


(a) ℓ masked multiplications with isolating approach

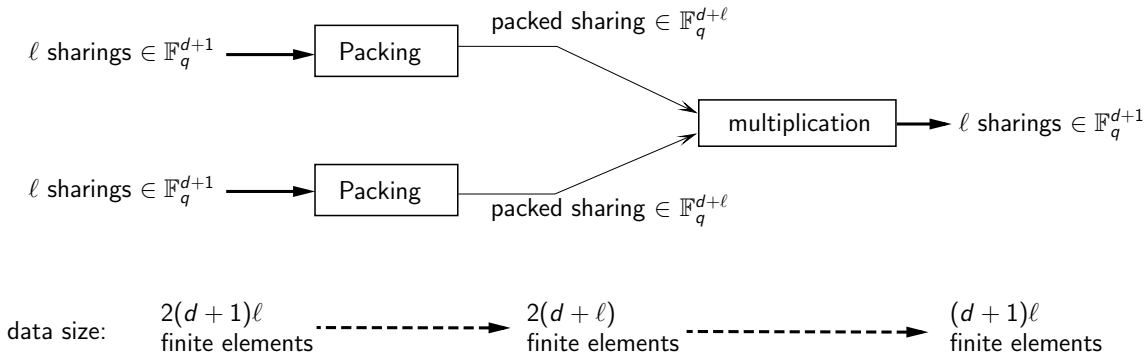
Isolating vs. Packed Approaches

sharing: the list of shares, e.g., $\hat{x}[0], \ldots, \hat{x}[d]$

ℓ: the number of multiplications

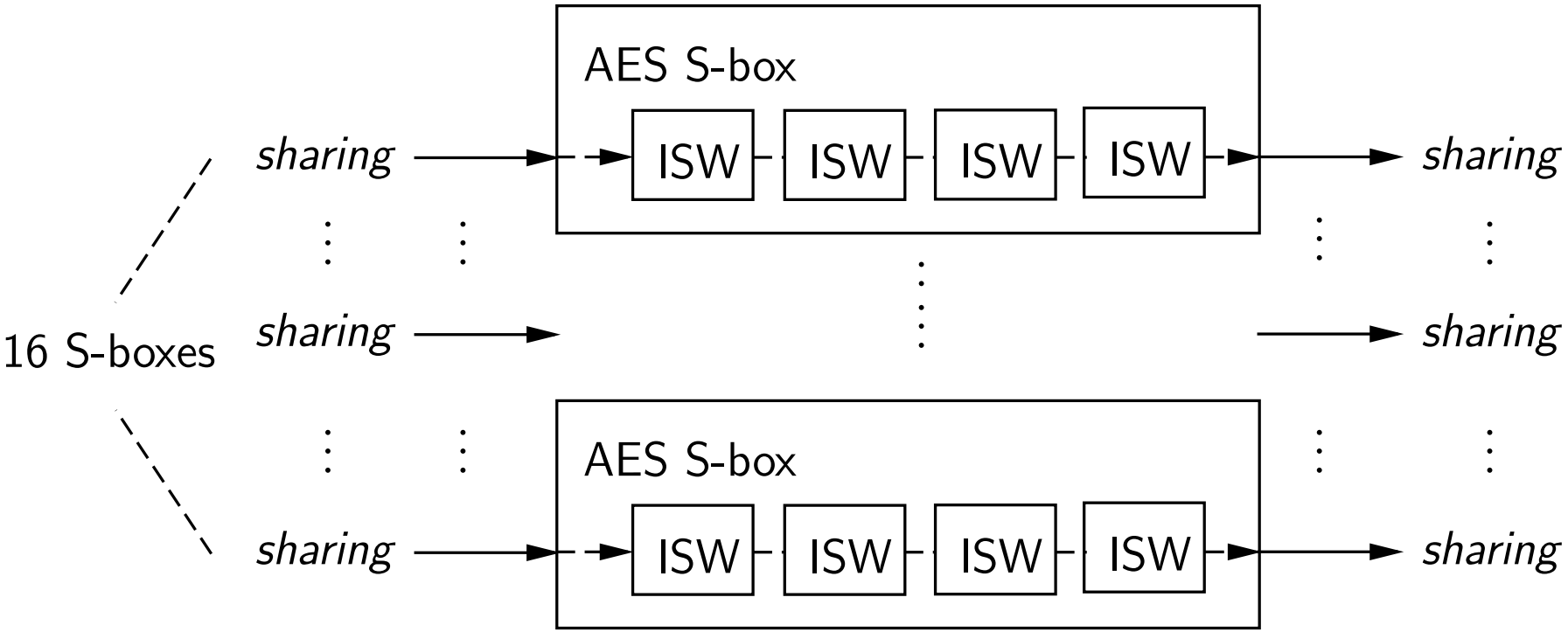


(a) ℓ masked multiplications with isolating approach



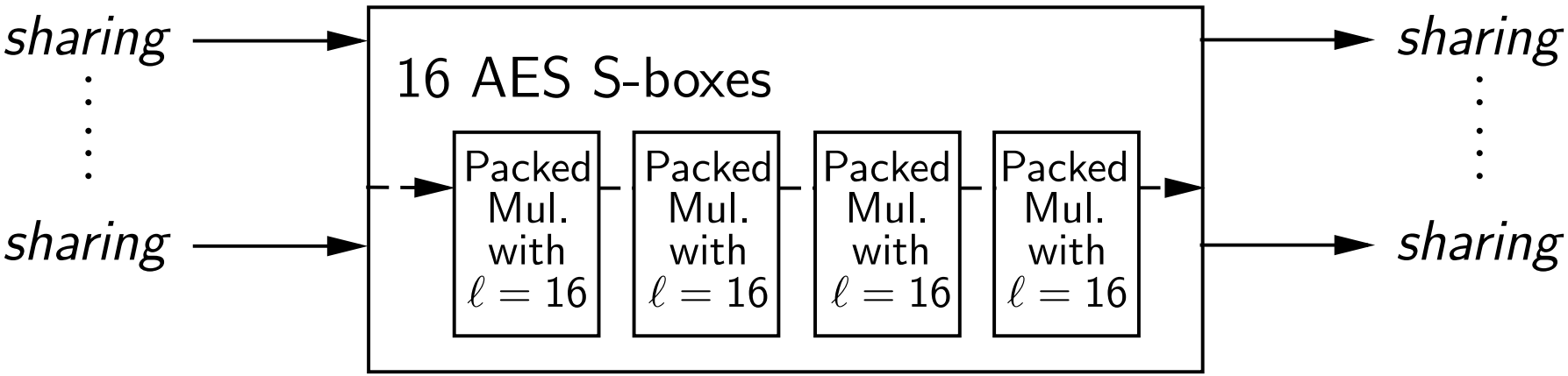
(b) ℓ masked multiplications with our packed approach

Isolating vs. Packed Approaches: an example



Isolating approach: 16×4 ISW multiplications over $\mathbb{F}_{2^8}$
$64d^2$ bilinear multiplications and $64d(d+1)/2$ random bytes

Isolating vs. Packed Approaches: an example



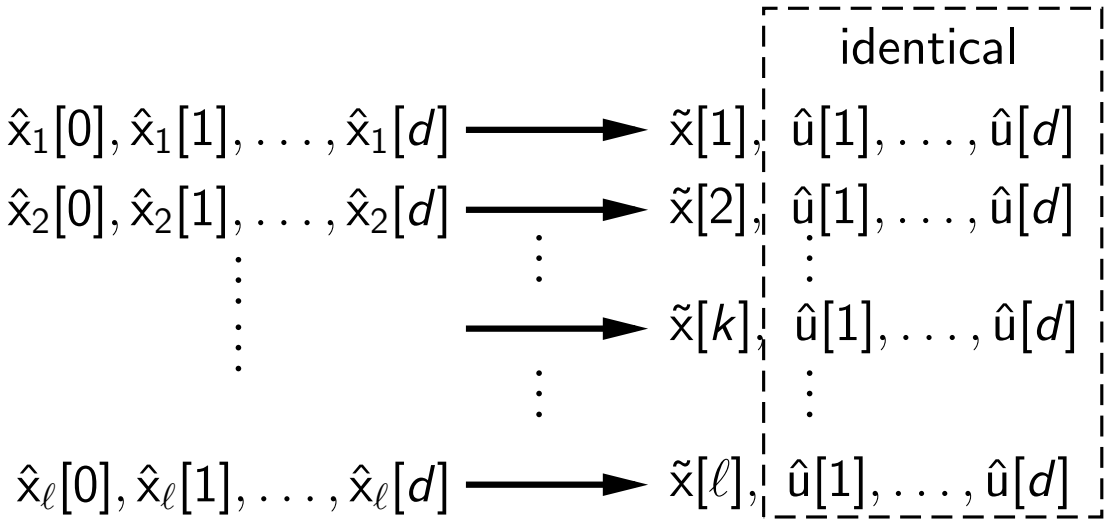
Packed approach: 4 packed multiplications
$d^2 + 32d + 16$ bilinear multiplications and $d(d+1)/2$ random bytes

- Number of bilinear multiplications: $64d^2 \rightarrow d^2 + 32d + 16$.
- Number of random bytes: $64d(d+1)/2 \rightarrow d(d+1)/2$.

Table of Contents

The Packing

- Input: Boolean sharings $\{\hat{x}_i\}_{i=1}^{\ell}$
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{d})$



$$x_k \equiv \tilde{x}[k] \oplus \langle (\hat{u}[1], \ldots, \hat{u}[d]), a_k \rangle \text{ with } a_k \in \mathbb{F}_q^{d}$$

An simple example of Packing with $d = 2$ and $\ell = 3$

- Input: Boolean sharings $\hat{x}_1 \in \mathbb{F}_q^3$, $\hat{x}_2 \in \mathbb{F}_q^3$ and $\hat{x}_3 \in \mathbb{F}_q^3$.
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^3, \mathbb{F}_q^2)$

1. Randomly generate $Q \in \mathbb{F}_q^{d \times d}$. As $d = 2$, $Q \in \mathbb{F}_q^{2 \times 2}$.
2. Sum the columns of matrix $Q \rightarrow \hat{u}$.

- $[1,1]Q$
- $([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2])) \oplus \hat{x}_1[0]$
- $\rightarrow \tilde{x}[1]$

$\cdots$

- $[1,3]Q$
- $([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2])) \oplus \hat{x}_3[0]$
- $\rightarrow \tilde{x}[3]$

An simple example of Packing with $d = 2$ and $\ell = 3$

- Input: Boolean sharings $\hat{x}_1 \in \mathbb{F}_q^3$, $\hat{x}_2 \in \mathbb{F}_q^3$ and $\hat{x}_3 \in \mathbb{F}_q^3$.
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^3, \mathbb{F}_q^2)$

1. Randomly generate $Q \in \mathbb{F}_q^{d \times d}$. As $d = 2$, $Q \in \mathbb{F}_q^{2 \times 2}$.
2. Sum the columns of matrix $Q \rightarrow \hat{u}$.

- $[1,1]Q$
- $([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2])) \oplus \hat{x}_1[0]$
- $\rightarrow \tilde{x}[1]$

$\cdots$

- $[1,3]Q$
- $([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2])) \oplus \hat{x}_3[0]$
- $\rightarrow \tilde{x}[3]$

An simple example of Packing with $d = 2$ and $\ell = 3$

- Input: Boolean sharings $\hat{x}_1 \in \mathbb{F}_q^3$, $\hat{x}_2 \in \mathbb{F}_q^3$ and $\hat{x}_3 \in \mathbb{F}_q^3$.
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^3, \mathbb{F}_q^2)$

1. Randomly generate $Q \in \mathbb{F}_q^{d \times d}$. As $d = 2$, $Q \in \mathbb{F}_q^{2 \times 2}$.
2. Sum the columns of matrix $Q \rightarrow \hat{u}$.

- $[1,1]Q$
- $([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2])) \oplus \hat{x}_1[0]$
- $\rightarrow \tilde{x}[1]$

$\cdots$

- $[1,3]Q$
- $([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2])) \oplus \hat{x}_3[0]$
- $\rightarrow \tilde{x}[3]$

An simple example of Packing with $d = 2$ and $\ell = 3$

- Input: Boolean sharings $\hat{x}_1 \in \mathbb{F}_q^3$, $\hat{x}_2 \in \mathbb{F}_q^3$ and $\hat{x}_3 \in \mathbb{F}_q^3$.
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^3, \mathbb{F}_q^2)$

1. Randomly generate $Q \in \mathbb{F}_q^{d \times d}$. As $d = 2$, $Q \in \mathbb{F}_q^{2 \times 2}$.
2. Sum the columns of matrix $Q \rightarrow \hat{u}$.

- $[1,1]Q$
- $([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2])) \oplus \hat{x}_1[0]$
- $\rightarrow \tilde{x}[1]$

$\cdots$

- $[1,3]Q$
- $([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2])) \oplus \hat{x}_3[0]$
- $\rightarrow \tilde{x}[3]$

An simple example of Packing with $d = 2$ and $\ell = 3$

- Input: Boolean sharings $\hat{x}_1 \in \mathbb{F}_q^3$, $\hat{x}_2 \in \mathbb{F}_q^3$ and $\hat{x}_3 \in \mathbb{F}_q^3$.
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^3, \mathbb{F}_q^2)$

1. Randomly generate $Q \in \mathbb{F}_q^{d \times d}$. As $d = 2$, $Q \in \mathbb{F}_q^{2 \times 2}$.
2. Sum the columns of matrix $Q \rightarrow \hat{u}$.

- $[1,1]Q$
- $([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2])) \oplus \hat{x}_1[0]$
- $\rightarrow \tilde{x}[1]$

$\cdots$

- $[1,3]Q$
- $([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2])) \oplus \hat{x}_3[0]$
- $\rightarrow \tilde{x}[3]$

An simple example of Packing with $d = 2$ and $\ell = 3$

- Input: Boolean sharings $\hat{x}_1 \in \mathbb{F}_q^3$, $\hat{x}_2 \in \mathbb{F}_q^3$ and $\hat{x}_3 \in \mathbb{F}_q^3$.
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^3, \mathbb{F}_q^2)$

1. Randomly generate $Q \in \mathbb{F}_q^{d \times d}$. As $d = 2$, $Q \in \mathbb{F}_q^{2 \times 2}$.
2. Sum the columns of matrix $Q \rightarrow \hat{u}$.

- $[1,1]Q$
- $([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2])) \oplus \hat{x}_1[0]$
- $\rightarrow \tilde{x}[1]$

$\cdots$

- $[1,3]Q$
- $([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2])) \oplus \hat{x}_3[0]$
- $\rightarrow \tilde{x}[3]$

An simple example of Packing with $d = 2$ and $\ell = 3$

- Input: Boolean sharings $\hat{x}_1 \in \mathbb{F}_q^3$, $\hat{x}_2 \in \mathbb{F}_q^3$ and $\hat{x}_3 \in \mathbb{F}_q^3$.
- Output: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^3, \mathbb{F}_q^2)$

1. Randomly generate $Q \in \mathbb{F}_q^{d \times d}$. As $d = 2$, $Q \in \mathbb{F}_q^{2 \times 2}$.
2. Sum the columns of matrix $Q \rightarrow \hat{u}$.

- $[1,1]Q$
- $([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2]))$
- $\sum ([1,1]Q \oplus (\hat{x}_1[1], \hat{x}_1[2])) \oplus \hat{x}_1[0]$
- $\rightarrow \tilde{x}[1]$

$\cdots$

- $[1,3]Q$
- $([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2]))$
- $\sum ([1,3]Q \oplus (\hat{x}_3[1], \hat{x}_3[2])) \oplus \hat{x}_3[0]$
- $\rightarrow \tilde{x}[3]$

$$\text{It ensures } \begin{cases} x_1 \equiv \tilde{x}[1] \oplus \langle (\hat{u}[1], \hat{u}[2]), [1,1] \rangle \\ x_2 \equiv \tilde{x}[2] \oplus \langle (\hat{u}[1], \hat{u}[2]), [1,2] \rangle \\ x_3 \equiv \tilde{x}[3] \oplus \langle (\hat{u}[1], \hat{u}[2]), [1,3] \rangle \end{cases}$$

The Multiplication over Packed Sharings

- Input: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{n-1})$ and $(\tilde{y}, \hat{v}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{d})$
- Output: Boolean sharings $\{\hat{z}_i\}_{i=1}^{\ell}$

$x_k \equiv \tilde{x}[k] \oplus \hat{u}[1]a_k[1] \oplus \hat{u}[2]a_k[2]$

$y_k \equiv \tilde{y}[k] \oplus \hat{v}[1]a_k[1] \oplus \hat{v}[2]a_k[2]$

$\tilde{x}[k]\tilde{y}[k]$	$\tilde{x}[k]\hat{v}[2]$	$\tilde{x}[k]\hat{v}[1]$
$\tilde{y}[k]\hat{u}[1]$	$\hat{u}[1]\hat{v}[1]$	$\hat{u}[1]\hat{v}[2]$
$\tilde{y}[k]\hat{u}[2]$	$\hat{u}[2]\hat{v}[1]$	$\hat{u}[2]\hat{v}[2]$

- It works with all the values of k with random variables unchanged.
  - Amortization!

The Multiplication over Packed Sharings

- Input: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{n-1})$ and $(\tilde{y}, \hat{v}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{d})$
- Output: Boolean sharings $\{\hat{z}_i\}_{i=1}^{\ell}$

$x_k \equiv \tilde{x}[k] \oplus \hat{u}[1]a_k[1] \oplus \hat{u}[2]a_k[2]$

$y_k \equiv \tilde{y}[k] \oplus \hat{v}[1]a_k[1] \oplus \hat{v}[2]a_k[2]$

$\tilde{x}[k]\tilde{y}[k]$	$\tilde{x}[k]\hat{v}[2]$	$\tilde{x}[k]\hat{v}[1]$
$\tilde{y}[k]\hat{u}[1]$	$\hat{u}[1]\hat{v}[1]$	$\hat{u}[1]\hat{v}[2]$
$\tilde{y}[k]\hat{u}[2]$	$\hat{u}[2]\hat{v}[1]$	$\hat{u}[2]\hat{v}[2]$

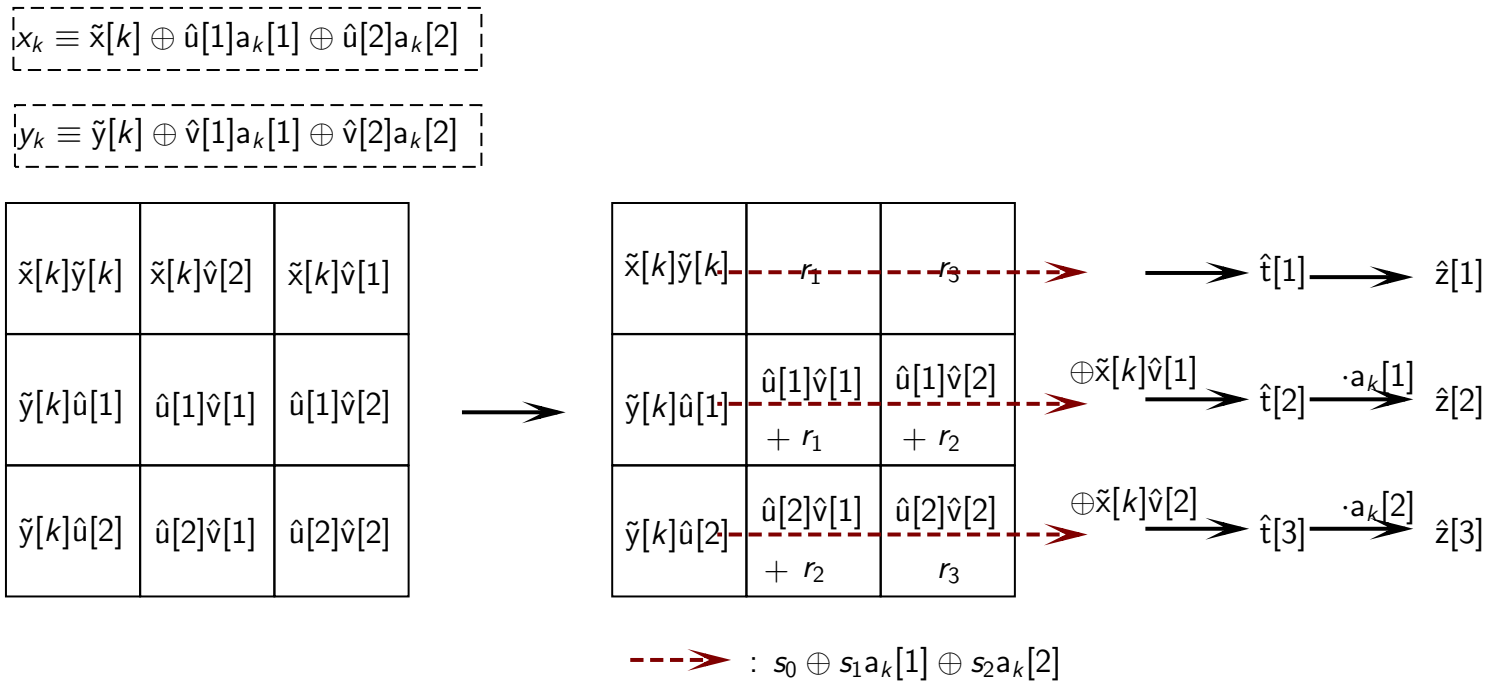
→

$\tilde{x}[k]\tilde{y}[k]$	r_1	r_3
$\tilde{y}[k]\hat{u}[1]$	$\hat{u}[1]\hat{v}[1] + r_1$	$\hat{u}[1]\hat{v}[2] + r_2$
$\tilde{y}[k]\hat{u}[2]$	$\hat{u}[2]\hat{v}[1] + r_2$	$\hat{u}[2]\hat{v}[2] r_3$

- It works with all the values of k with random variables unchanged.
  - Amortization!

The Multiplication over Packed Sharings

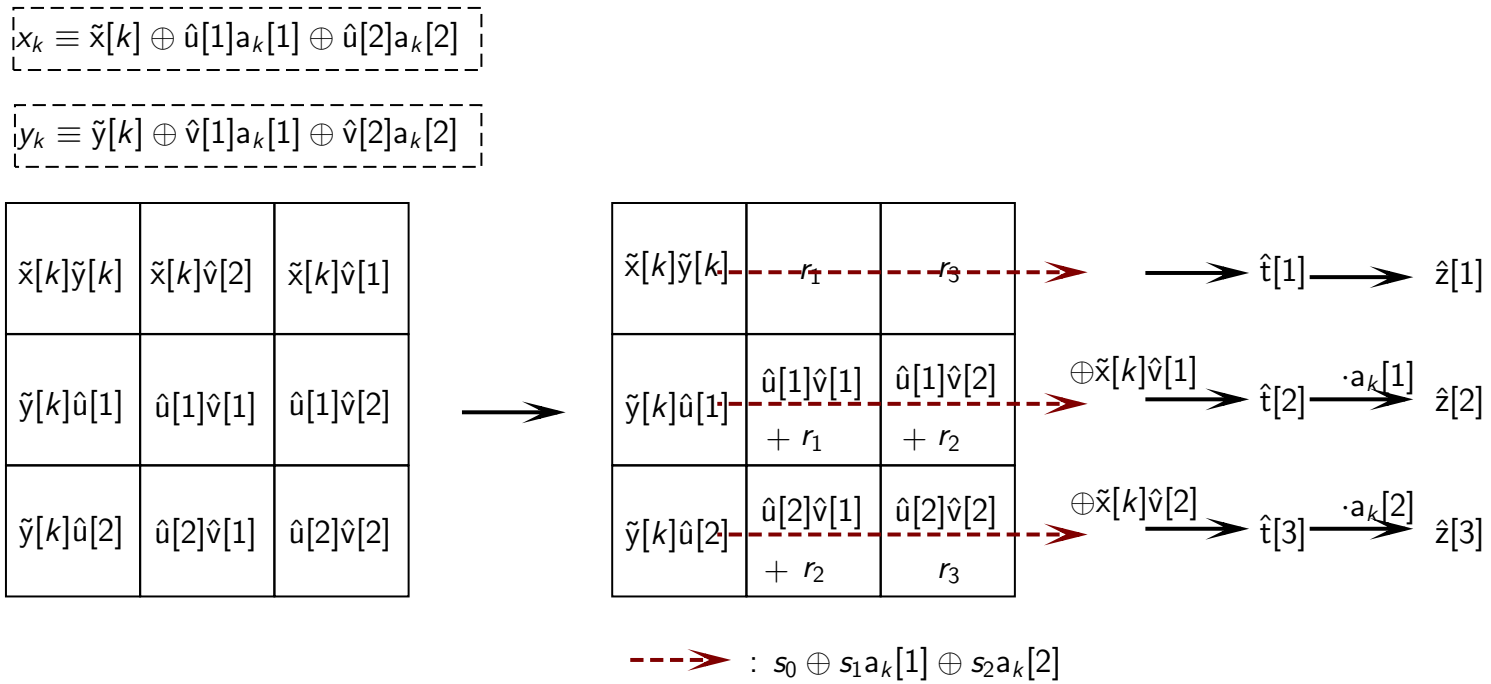
- Input: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{n-1})$ and $(\tilde{y}, \hat{v}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{d})$
- Output: Boolean sharings $\{\hat{z}_i\}_{i=1}^{\ell}$



- It works with all the values of k with random variables unchanged.
  - Amortization!

The Multiplication over Packed Sharings

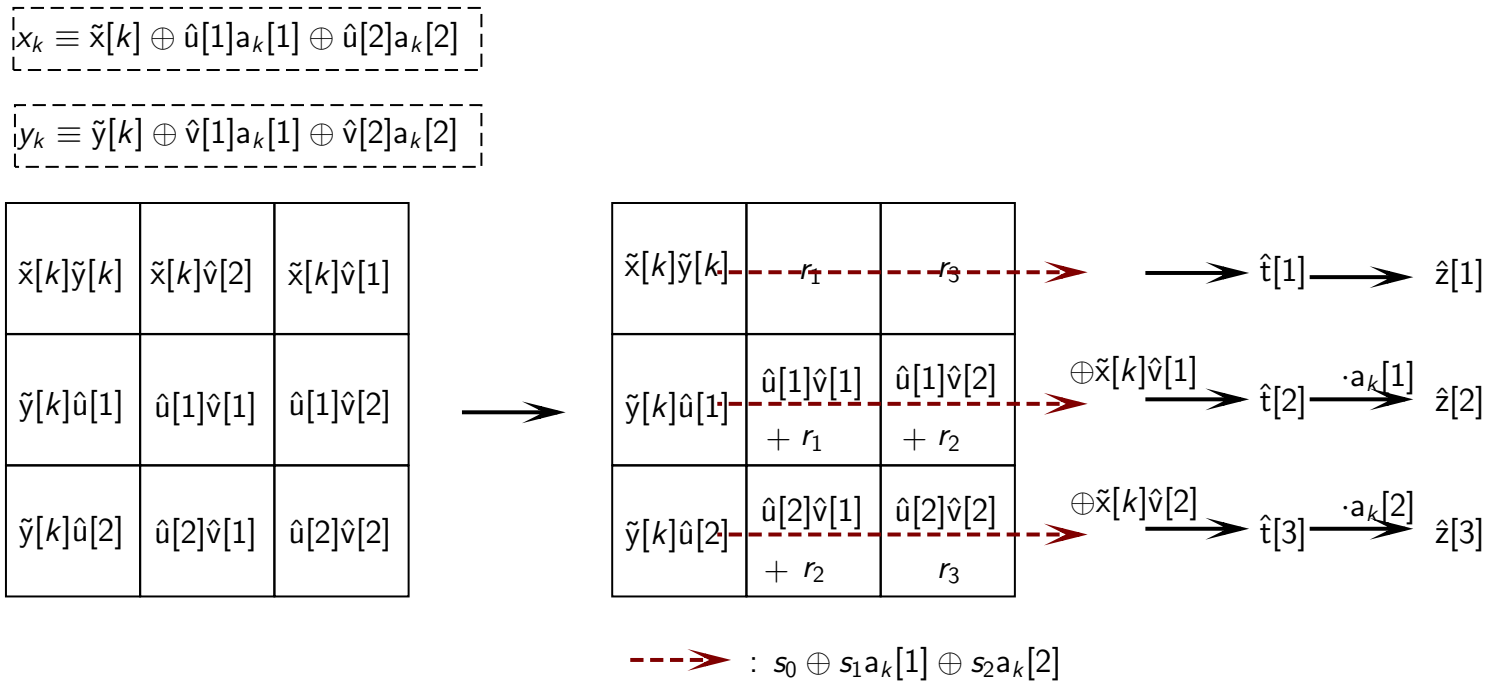
- Input: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{n-1})$ and $(\tilde{y}, \hat{v}) \in (\mathbb{F}_q^{\ell}, \mathbb{F}_q^{d})$
- Output: Boolean sharings $\{\hat{z}_i\}_{i=1}^{\ell}$



- It works with all the values of k with random variables unchanged.
  - Amortization!

The Multiplication over Packed Sharings

- Input: Packed sharings $(\tilde{x}, \hat{u}) \in (\mathbb{F}_q^\ell, \mathbb{F}_q^{n-1})$ and $(\tilde{y}, \hat{v}) \in (\mathbb{F}_q^\ell, \mathbb{F}_q^d)$
- Output: Boolean sharings $\{\hat{z}_i\}_{i=1}^{\ell}$



- It works with all the values of k with random variables unchanged.
  - Amortization!

Table of Contents

Summary of Performances for 16 AES S-boxes

	Cyc. for Comp.	Cyc. for Rand.	Total Cyc.	Code size	RAM size
[1, R.-P. meth.], $d = 4$	19 232	34 944	54 176	4KB	-
[1, Bitsliced meth.], $d = 4$	11 502	17 472	28 974	3.1KB	-
[2, Bit. meth.], $d = 4$	9 222	9 282	18 504	-	-
Our works, $d = 4$	15 998	4 200	20 198	9.8KB	10.9KB
[1, R.-P. meth.], $d = 8$	70 840	163, 072	233 912	4KB	-
[1, Bit. meth.], $d = 8$	34 798	81 536	116 334	3.1KB	-
[2, Bit. meth.], $d = 8$	27 028	43 316	70 344	-	-
Our work, $d = 8$	33 142	13 840	46 982	17KB	11.8KB

Dahmun Goudarzi et al. How Fast Can Higher-Order Masking Be in Software? EUROCRYPT 2017.

Sonia Belaïd et al. Tight Private Circuits: Achieving Probing Security with the Least Refreshing. ASIACRYPT 2018.

Table of Contents

Conclusion and Future works

- We consider ℓ parallel masked operations with probing security order d and reduce the averaged randomness and computational cost via amortization.
  - The computational complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2 + d\ell + \ell)$.
  - The randomness complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2)$.
- Future works.
  - Hardware implementation.
  - Proofs by hand $\rightarrow$ Verified proofs.
    - Formal method and automotive tools.
  - Concrete security evaluation.

Conclusion and Future works

- We consider ℓ parallel masked operations with probing security order d and reduce the averaged randomness and computational cost via amortization.
  - The computational complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2 + d\ell + \ell)$.
  - The randomness complexity decreases: $\mathcal{O}(\ell d^2) \rightarrow \mathcal{O}(d^2)$.
- Future works.
  - Hardware implementation.
  - Proofs by hand $\rightarrow$ Verified proofs.
    - Formal method and automotive tools.
  - Concrete security evaluation.

Thank You!