Threshold Implementations: Comprehend and Apply

Svetla Nikova, KU Leuven, Belgium

June 8, 2013

KATHOLIEKE UNIVERSITEIT LEUVEN

KATHOLIEKE UNIVERSITEIT LEUVEN

Side-channel attacks

- Normal attacks: $c = E(k, p)$
  - Known plaintext: equations in the key
  - High nonlinearity, difficult to solve
- Device executing the cryptographic algorithm leaks information on internal state
- Instantaneous leakage depends on intermediate variables, which results in equations
  - That have lower nonlinearity
  - That may contain noise

Example 1: Square-and-multiply

Compute m^e

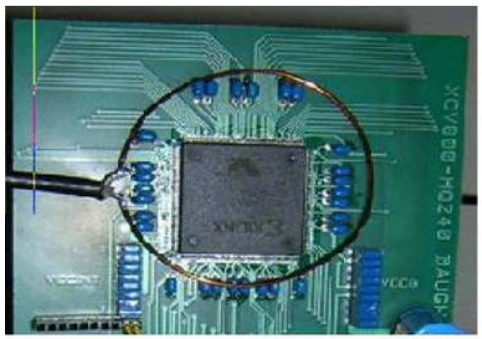
$x = 1$

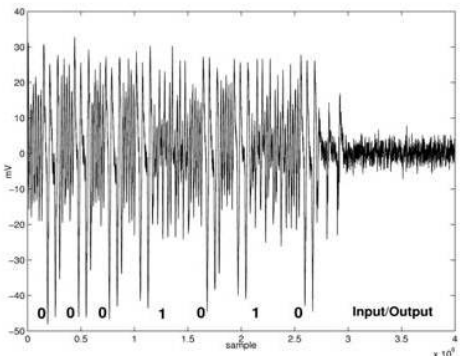
`for` $i = t$ `downto` 2

$x = x_2$

`If` $e_i = 1$ `then` $x = x * m$

- Leakage signal is very clear $\rightarrow$ Simple Power Analysis (SPA)
- Leakage is avoidable





Example 2: AES

- SPA attacks exploit **key-dependent patterns** within a trace
- SPA attacks use only **one or very few traces**.

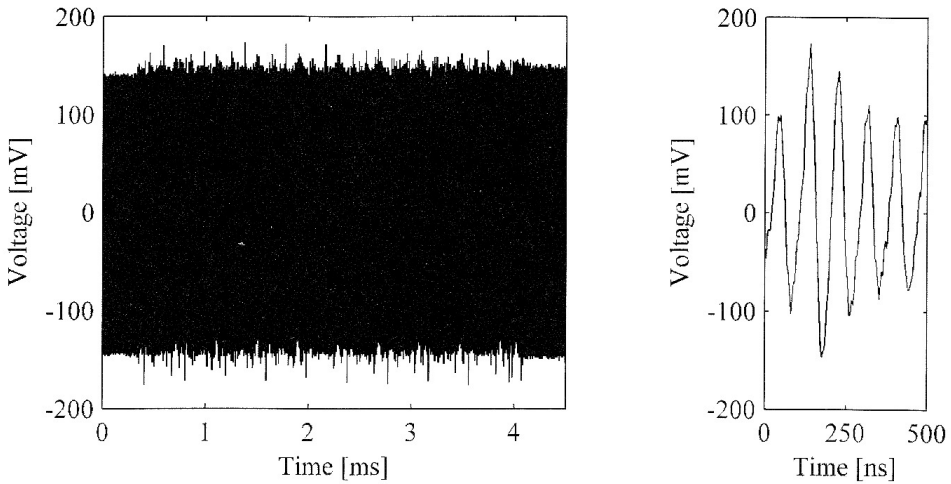


Figure: Power consumption of the microcontroller during AES encryption execution. Zoomed view of the power trace.

(ref. "Power Analysis Attacks", S. Mangard, E. Oswald, T. Popp)

KATHOLIEKE UNIVERSITEIT LEUVEN

Differential Power Analysis (DPA)

- DPA attacks exploit the **data dependency** of the power consumption of crypto devices.
- DPA attacks use a **large number of traces** and analyze the power consumption at a fixed moment of time as a function of the proccessed data.

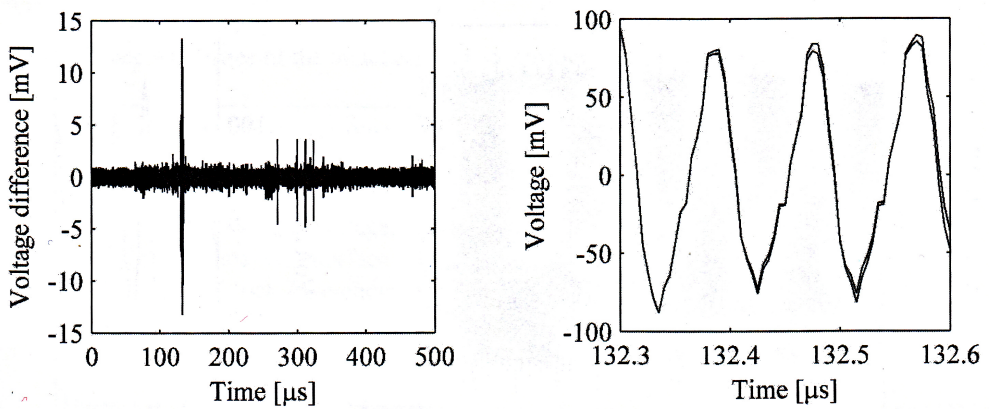


Figure: Difference of the mean power trace for MSB=1 and the mean power trace for MSB=0. Zoomed view of the mean power trace.

(ref. "Power Analysis Attacks", S. Mangard, E. Oswald, T. Popp)

KATHOLIEKE UNIVERSITEIT LEUVEN

Countering power attacks

- Ensure constant power consumption
  - Constant instruction sequence
  - Use special hardware logic styles
- Avoid statistical correlation between secret key and data processed
  - *Masking*
  - Counters attacks that use repeated measurements and statistics to remove the noise

Countermeasures at different levels

- Hardware logic style
  $\rightarrow$ Relieves cryptographers
  BUT places burden on hardware designers
- Algorithms and implementations
  $\rightarrow$ Probably lowest feasible level
- Ciphers and Protocols
  $\rightarrow$ New standards, takes time

Countermeasures

We NEED secure implementations against DPA

Countermeasures

We NEED secure implementations against DPA

- Hardware countermeasures
  - Balancing power consumption [Tiri et al., CHES'03]
  - ...
- Masking
  - Masking intermediate values [Chari et al., CRYPTO'99; Goubin et al., CHES'99]
  - Threshold Implementations [Nikova et al., ICISC'08]
  - Shamir's Secret Sharing [Goubin et al., CHES'11; Prouff et al., CHES'11]
  - ...
- Leakage-Resilient Crypto

Countermeasures

We NEED secure implementations against DPA

- Hardware countermeasures
  - Balancing power consumption [Tiri et al., CHES'03]
  - ...
- Masking
  - Masking intermediate values [Chari et al., CRYPTO'99; Goubin et al., CHES'99]
  - Threshold Implementations [Nikova et al., ICISC'08]
  - Shamir's Secret Sharing [Goubin et al., CHES'11; Prouff et al., CHES'11]
  - ...
- Leakage-Resilient Crypto

Problem: Unfeasible circuit size, glitches

KATHOLIEKE UNIVERSITEIT LEUVEN

Hiding

In each clock cycle, consume either

- (close to) random amounts of power
- (close to) equal amounts of power

Hiding only decreases the signal-to-noise ratio (SNR)
Hiding dimensions

- Time
- Amplitude

Masking

Randomized redundant representation:
$v \rightarrow (v_1, \ldots, v_n)$ such that $v = v_1 * \ldots * v_n$
n-th order masking: all $n-1$ intermediate variables are independent of v

The adversary needs to identify n leakage samples and combine their information

Boolean masking: $v_1 = v \oplus m,\ v_2 = m$
Multiplicative masking (zero-value problem): $v_1 = v * m,\ v_2 = m$
Affine Masking: $v_1 = v * m \oplus m_2,\ v_2 = m_1,\ v_3 = m_2$

Masking in Software

Masking Table Look-Ups

Two tables have to be computed T and T_m, where $T_m(v \oplus m) = T(v) \oplus m$

Consequences: the computational effort and amount of memory increases.

Problems with masking

- Unintentional unmasking,
- Glitches

$$HD(v_m, w_m) = HW(v_m \oplus w_m) = HW(v \oplus w)$$

Glitches

Temporary states of the output

Glitches

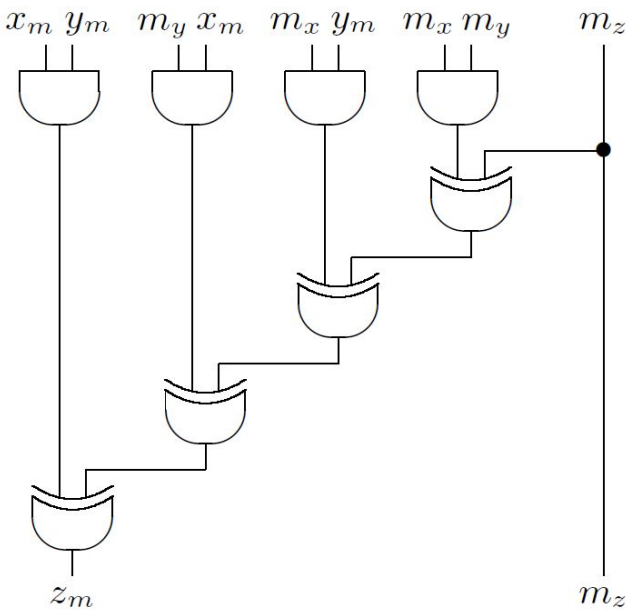
Temporary states of the output

$$z \quad = \quad x \; AND \; y \;\text{, where } x_m = x \oplus m_x, y_m = y \oplus m_y$$

Glitches

Temporary states of the output

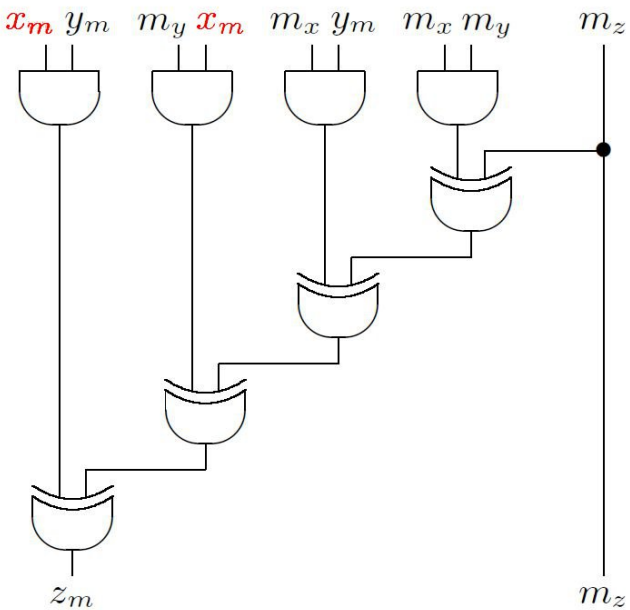
$$z = x \; AND \; y \text{ , where } x_m = x \oplus m_x, y_m = y \oplus m_y$$
$$z_m = x_m y_m \oplus (m_y x_m \oplus (m_x y_m \oplus (m_x m_y \oplus m_z)))$$



Glitches

Temporary states of the output

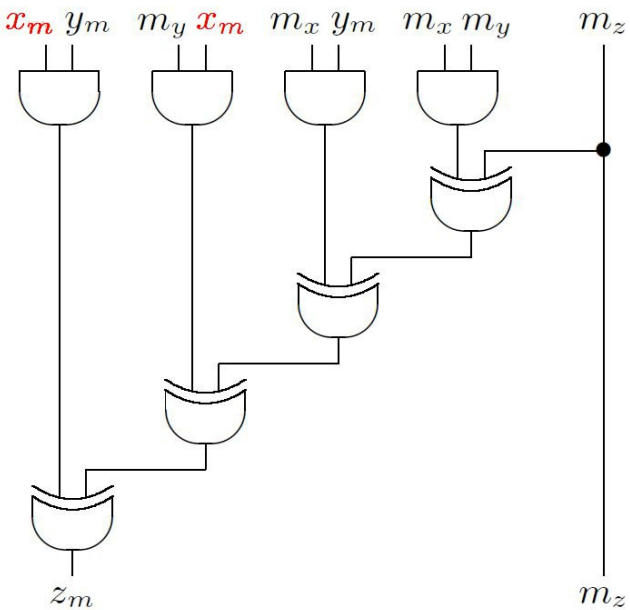
$$z = x \; AND \; y \text{ , where } x_m = x \oplus m_x, y_m = y \oplus m_y$$
$$z_m = x_m y_m \oplus (m_y x_m \oplus (m_x y_m \oplus (m_x m_y \oplus m_z)))$$



Glitches

Temporary states of the output

$$z = x \; AND \; y \text{ , where } x_m = x \oplus m_x, y_m = y \oplus m_y$$
$$z_m = x_m y_m \oplus (m_y x_m \oplus (m_x y_m \oplus (m_x m_y \oplus m_z)))$$



y	m_y	y_m	AND	XOR
0	0	0	0	0
0	1	1	2	2
1	0	1	1	1
1	1	0	1	2

KATHOLIEKE UNIVERSITEIT LEUVEN

Why TI?

Threshold Implementations

- Any hardware technology
- Realistic size
- Provably secure against 1^{st} order DPA

Why TI?

Threshold Implementations

- Any hardware technology
- Realistic size
- Provably secure against 1^{st} order DPA

So far,

- Noekeon [Nikova et al., ICISC'08]
- Multiplication in $GF(4)$ [Nikova et al., ICISC'08]
- Keccak [Bertoni et al., SHA-3 candidates'10]
- Present [Poschmann et al., J.Cryptology'11]
- AES [Moradi et al., Eurocrypt'11]
- All 3×3 and 4×4 S-boxes [Bilgin et al., CHES'12]
- etc.

KATHOLIEKE UNIVERSITEIT LEUVEN

Threshold Implementations

The rest of this lecture,

- Comprehend the TI
  - Re-iterate the 3 properties of TI
  - Their role in the security proofs

Threshold Implementations

The rest of this lecture,

- Comprehend the TI
  - Re-iterate the 3 properties of TI
  - Their role in the security proofs
- Applying TI
  - Analysis of all 3×3 and 4×4 S-boxes
  - More complex example - AES

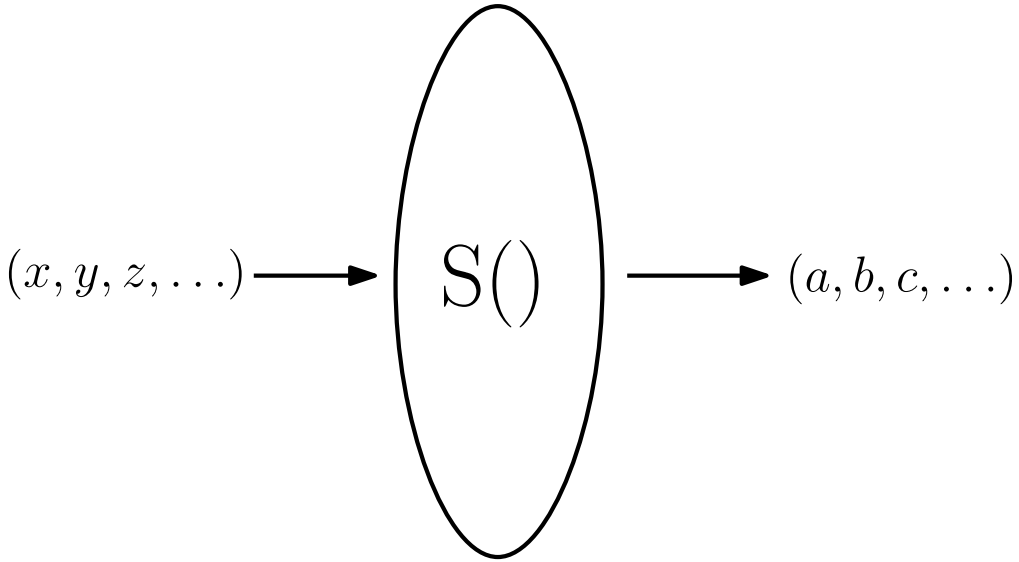
Threshold Implementations

The rest of this lecture,

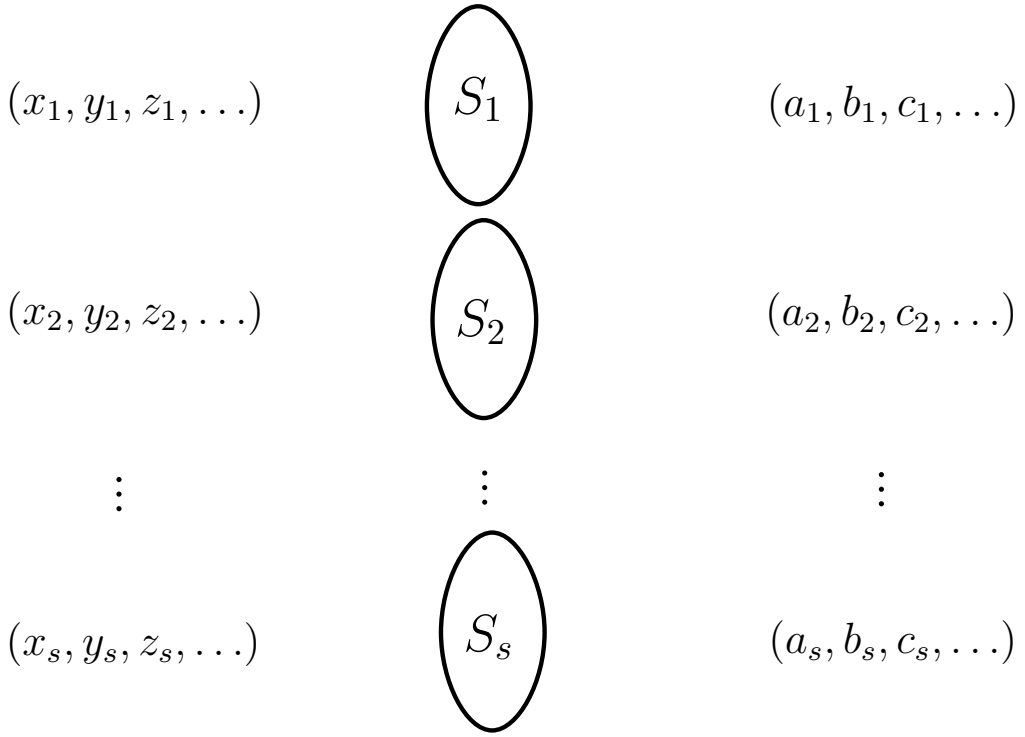
- Comprehend the TI
  - Re-iterate the 3 properties of TI
  - Their role in the security proofs
- Applying TI
  - Analysis of all 3×3 and 4×4 S-boxes
  - More complex example - AES
- Cost of a TI

KATHOLIEKE UNIVERSITEIT LEUVEN

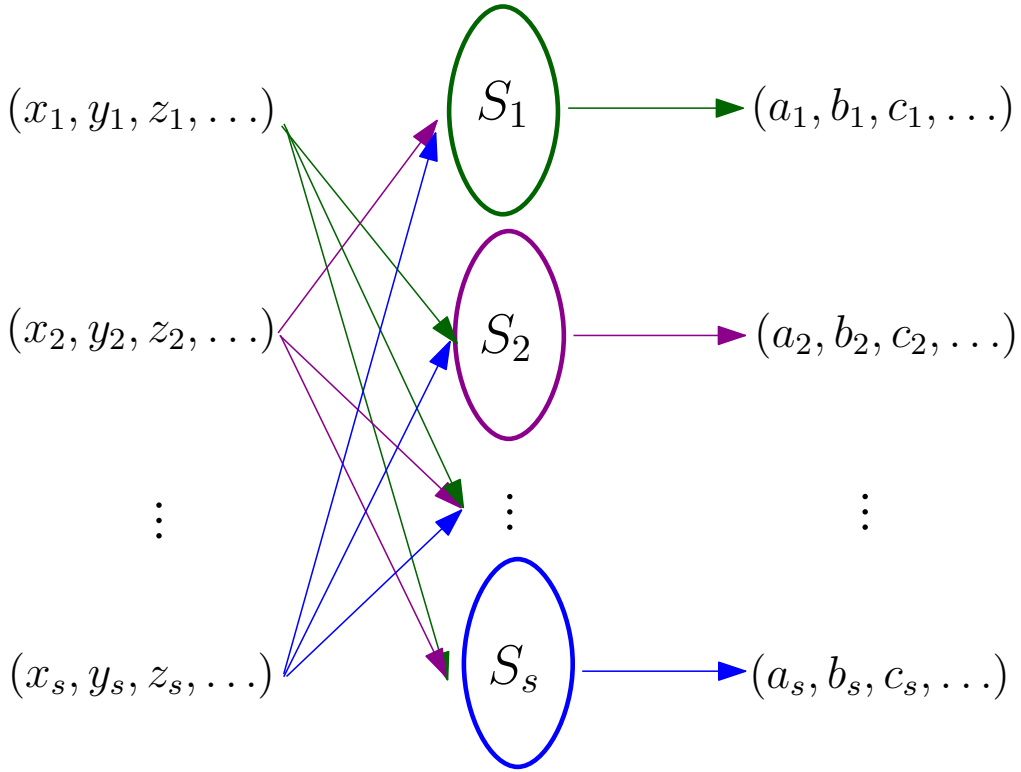
What is TI?



What is TI?

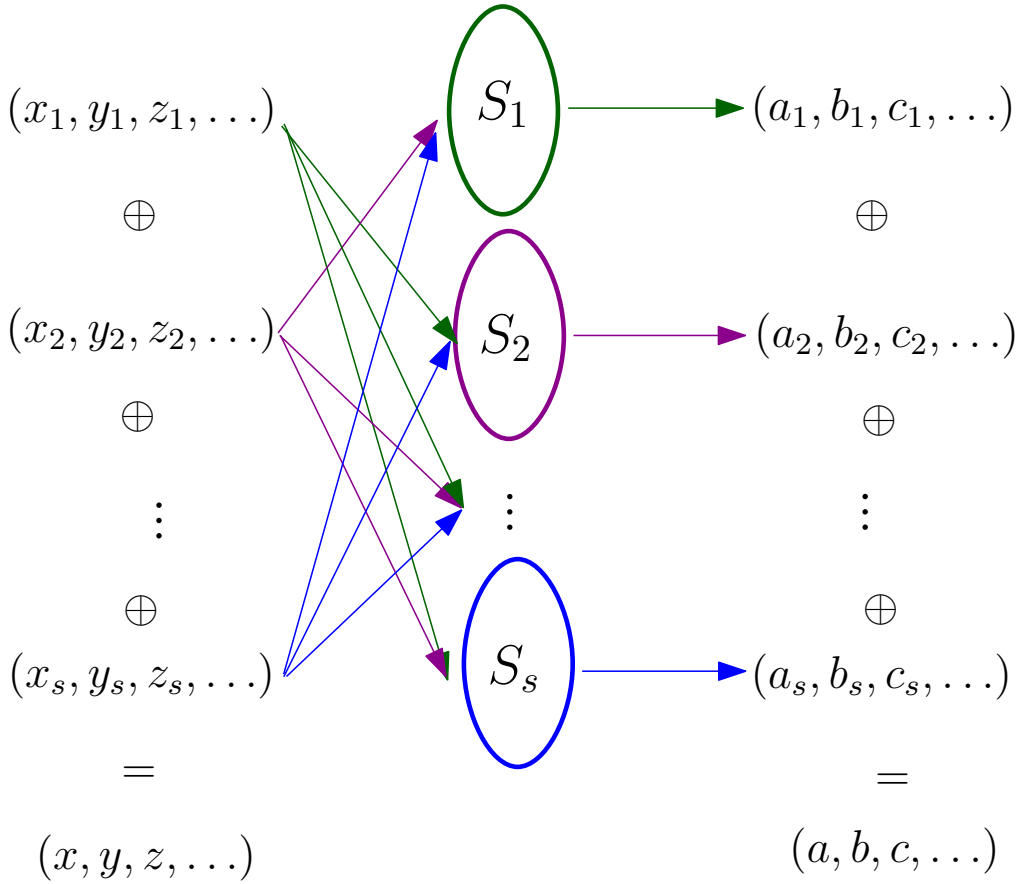

$(x_1, y_1, z_1, \ldots)$
S_1
$(a_1, b_1, c_1, \ldots)$
$(x_2, y_2, z_2, \ldots)$
S_2
$(a_2, b_2, c_2, \ldots)$
$(x_s, y_s, z_s, \ldots)$
S_s
$(a_s, b_s, c_s, \ldots)$

What is TI?



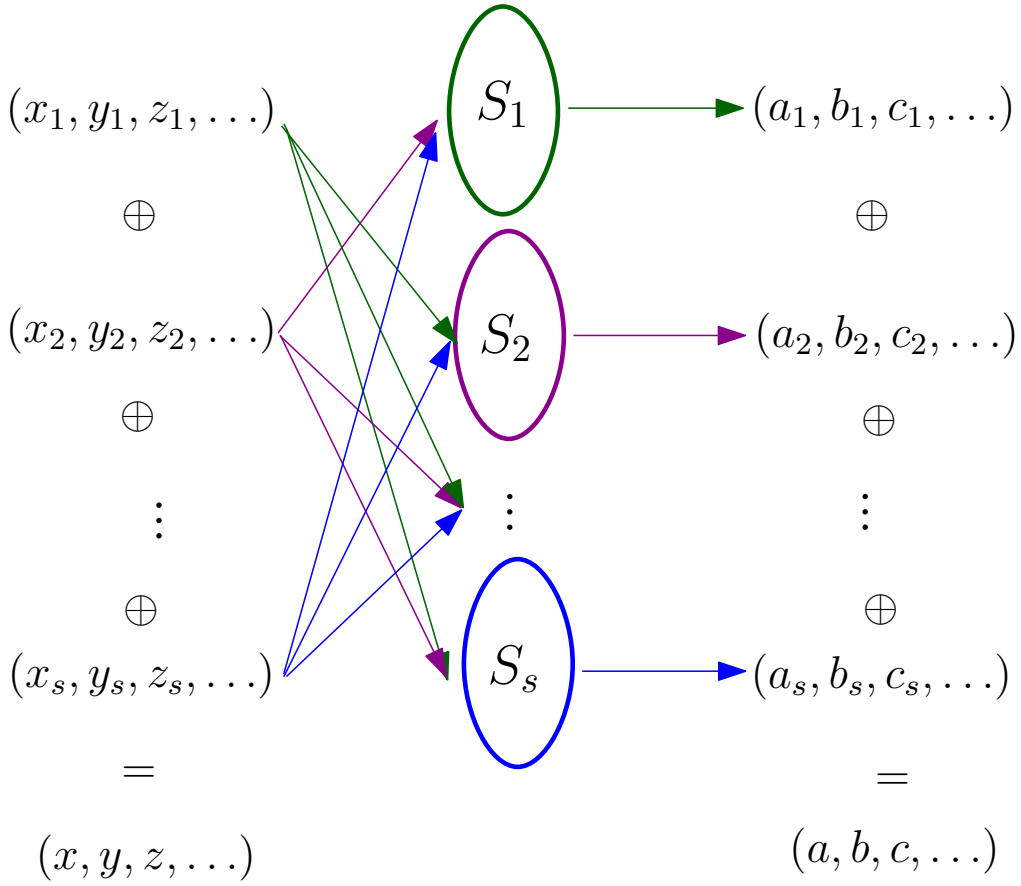
- Non-complete

What is TI?



- Correct
- Non-complete

What is TI?



- Correct
- Non-complete
- Uniform

Uniformity

- S-boxes: If $S(x) = a$ is a bijection, then $S(x_1, x_2, x_3) = (a_1, a_2, a_3)$ is also a bijection.

Uniformity

- S-boxes: If $S(x) = a$ is a bijection, then $S(x_1, x_2, x_3) = (a_1, a_2, a_3)$ is also a bijection.
- Multiplication:

x	y	a=x AND y
0	0	0
0	1	0
1	0	0
1	1	1

a	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
0	4	0	0	4	0	4	4	0
0	4	0	0	4	0	4	4	0
0	4	0	0	4	0	4	4	0
1	0	4	4	0	4	0	0	4
0	12	0	0	12	0	12	12	0
1	0	4	4	0	4	0	0	4

KATHOLIEKE UNIVERSITEIT LEUVEN

Exercises

Consider $f(a, b) = a \times b$ in $GF(2)$, i.e. AND gate.

- Find a correct and non-complete sharing for $f(a, b)$ with 2 shares.

Exercises

Consider $f(a, b) = a \times b$ in $GF(2)$, i.e. AND gate.

- Find a correct and non-complete sharing for $f(a, b)$ with 2 shares.
- It does not exist.

Exercises

Consider $f(a, b) = a \times b$ in $GF(2)$, i.e. AND gate.

- Find a correct and non-complete sharing for $f(a, b)$ with 2 shares.
- It does not exist.
- Find a sharing for $f(a, b)$ with 3 shares, which is correct.
- Find correct and non-complete sharing for $f(a, b)$ with 3 shares.

Exercises

Consider $f(a, b) = a \times b$ in $GF(2)$, i.e. AND gate.

- Find a correct and non-complete sharing for $f(a, b)$ with 2 shares.
- It does not exist.
- Find a sharing for $f(a, b)$ with 3 shares, which is correct.
- Find correct and non-complete sharing for $f(a, b)$ with 3 shares.

$$
\begin{aligned}
F_1(a_2, a_3, b_2, b_3) &= a_2 b_2 + a_2 b_3 + a_3 b_2 \\
F_2(a_1, a_3, b_1, b_3) &= a_3 b_3 + a_1 b_3 + a_3 b_1 \\
F_3(a_1, a_2, b_1, b_2) &= a_1 b_1 + a_1 b_2 + a_2 b_1
\end{aligned}
$$

KATHOLIEKE UNIVERSITEIT LEUVEN

Exercises

Consider $f(a, b) = a \times b$ in $GF(2)$, i.e. AND gate.

- How many correct and non-complete sharings for $f(a, b)$ with 3 shares exist?

$$
\begin{aligned}
F_1(a_2, a_3, b_2, b_3) &= a_2 b_2 + a_3 b_3 + a_2 b_3 + a_3 b_2 \\
F_2(a_1, a_3, b_1, b_3) &= a_1 b_3 + a_3 b_1 \\
F_3(a_1, a_2, b_1, b_2) &= a_1 b_1 + a_1 b_2 + a_2 b_1
\end{aligned}
$$

KATHOLIEKE UNIVERSITEIT LEUVEN

Exercises

Consider $f(a, b) = a \times b$ in $GF(2)$, i.e. AND gate.

- Is the sharing you found an uniform sharing?
- Find a correct and non-complete sharing for $f(a, b)$ with 4 shares?
- (Homework) find a correct, non-complete and uniform sharing for $f(a, b)$ with 4 shares?

Exercises

Consider $f(a, b) = a \times b$ in $GF(2)$, i.e. AND gate.

- Is the sharing you found an uniform sharing?
- Find a correct and non-complete sharing for $f(a, b)$ with 4 shares?
- (Homework) find a correct, non-complete and uniform sharing for $f(a, b)$ with 4 shares?

Theorem

To TI share a function with algebraic degree d, at least $d + 1$ shares are necessary.

Uniform Masking and Non-completeness

Let $x \in \mathcal{F}^m$ denote the input of the (unshared) function f.
Let X be correct and uniform masking of x i.e. $X \in \mathrm{Sh}(x)$, and F be a sharing of f.

Uniform Masking and Non-completeness

Let $x \in \mathcal{F}^m$ denote the input of the (unshared) function f.
Let X be correct and uniform masking of x i.e. $X \in \mathrm{Sh}(x)$,
and F be a sharing of f.

Definition (Uniform masking)

A masking X is *uniform* if and only if there exists a constant p such that for all x we have:
if $X \in \mathrm{Sh}(x)$ then $\Pr(X|x) = p$, else $\Pr(X|x) = 0$.

Uniform Masking and Non-completeness

Let $x \in \mathcal{F}^m$ denote the input of the (unshared) function f.
Let X be correct and uniform masking of x i.e. $X \in \mathrm{Sh}(x)$,
and F be a sharing of f.

Definition (Uniform masking)

A masking X is *uniform* if and only if there exists a constant p such that for all x we have:
if $X \in \mathrm{Sh}(x)$ then $\Pr(X|x) = p$, else $\Pr(X|x) = 0$.

Definition (Correctness)

The sharing F (of f) is *correct* if and only if
$\forall X \in \mathrm{Sh}(x), \forall Y \in \mathrm{Sh}(y) : F(X) = Y \Leftrightarrow f(x) = y$.

Uniform Masking and Non-completeness

Let $x \in \mathcal{F}^m$ denote the input of the (unshared) function f.
Let X be correct and uniform masking of x i.e. $X \in \mathrm{Sh}(x)$,
and F be a sharing of f.

Definition (Uniform masking)

A masking X is *uniform* if and only if there exists a constant p such that for all x we have:
if $X \in \mathrm{Sh}(x)$ then $\Pr(X|x) = p$, else $\Pr(X|x) = 0$.

Definition (Correctness)

The sharing F (of f) is *correct* if and only if
$\forall X \in \mathrm{Sh}(x), \forall Y \in \mathrm{Sh}(y) : F(X) = Y \Leftrightarrow f(x) = y$.

Definition (Non-completeness)

A sharing F (of f) is *non-complete* if every component function of F is independent of at least one share of X.

KATHOLIEKE UNIVERSITEIT LEUVEN

Security Proofs (1)

Let X_i denote the i-th share in X.
Let $X_{\bar{i}}$ denote the vector obtained by removing X_i from X.

Lemma

If the masking of x is uniform, then the stochastic functions $X_{\bar{i}}$ and x are independent (for any choice of i).

Security Proofs (1)

Let X_i denote the i-th share in X.
Let $X_{\bar{i}}$ denote the vector obtained by removing X_i from X.

Lemma

If the masking of x is uniform, then the stochastic functions $X_{\bar{i}}$ and x are independent (for any choice of i).

Theorem (1)

If the masking of x is uniform and the circuit F is non-complete, then any single component function of F does not leak information on x.

Security Proofs (2)

Even though the single component functions of F can be made independent of x, we cannot achieve independence for the whole circuit. However, due to the linearity of the expectation operator, we can still prove independence of the average value of any physical characteristic P of an implementation of the circuit.

Theorem (2)

If the masking of x is uniform and the circuit F is incomplete, then the expected value (average) of P over all masks is constant.

Uniformity (1)

Let $c = f(a, b) = a \times b$. Define F as follows:

$$\begin{aligned} c_1 &= F_1(a_2, a_3, b_2, b_3) = a_2 b_2 + a_2 b_3 + a_3 b_2 \\ c_2 &= F_2(a_1, a_3, b_1, b_3) = a_3 b_3 + a_1 b_3 + a_3 b_1 \\ c_3 &= F_3(a_1, a_2, b_1, b_2) = a_1 b_1 + a_1 b_2 + a_2 b_1 \,. \end{aligned}$$

If the masking of the input $x = (a, b)$ is uniform, then the masking of c is distributed as follows.

Table: Number of times that a masking $c_1 c_2 c_3$ occurs for a given input.

(a,b)	000	011	101	110	001	010	100	111
(0,0)	7	3	3	3	0	0	0	0
(0,1)	7	3	3	3	0	0	0	0
(1,0)	7	3	3	3	0	0	0	0
(1,1)	0	0	0	0	5	5	5	1

However in order to satisfy the uniformity of masking definition for c, we would need that the 16 non-zero values were equal to $2^{2(3-1)-1(3-1)} = 4$.

Uniformity (2)

Theorem 1 guarantees no leakage of information in *this* circuit!
Theorem 1 *does not* apply if c is used as input of a second circuit!
Example: let $e = d \times c$

$$e_1 = F_1(c_2, c_3, d_2, d_3) = c_2 d_2 + c_2 d_3 + c_3 d_2 \,.$$

Table: Number of times that a masking $e_1 e_2 e_3$ occurs for a given input (a, b, d).

(a,b,d)	000	011	101	110	001	010	100	111
(0,0,0)	37	9	9	9	0	0	0	0
(0,0,1)	37	9	9	9	0	0	0	0
(0,1,0)	37	9	9	9	0	0	0	0
(0,1,1)	37	9	9	9	0	0	0	0
(1,0,0)	37	9	9	9	0	0	0	0
(1,0,1)	37	9	9	9	0	0	0	0
(1,1,0)	31	11	11	11	0	0	0	0
(1,1,1)	0	0	0	0	21	21	21	1

The average Hamming weight for $(a, b, d) = (1, 1, 0)$ equals $33/32$, whereas it equals $27/32$ in the first six rows.

KATHOLIEKE UNIVERSITEIT LEUVEN

Uniformity - Remedy

Firstly, we can apply *re-masking*, i.e. by adding new masks to the shares c_1, c_2, c_3, we make the distribution uniform.
Secondly, we can impose an extra condition on F, such that the distribution of the output is always uniform.

Definition

The circuit F is *uniform* if and only if

$$\forall x \in \mathcal{F}^m, \forall y \in \mathcal{F}^n \text{ with } f(x) = y, \forall Y \in \mathrm{Sh}(y) :$$
$$|\{X \in \mathrm{Sh}(x) | F(X) = Y\}| = \frac{2^{m(s_x-1)}}{2^{n(s_y-1)}} .$$

Theorem (3)

If X, the masking of x is uniform and the circuit F is uniform, then the masking $Y = F(X)$ of $y = f(x)$ is uniform.

Consequences

Theorem 1 and Theorem 2 can be proven using only the correctness and incompleteness properties.
The uniformity property is needed only if several circuits are cascaded (*pipelined*), and even then it can be avoided with re-masking.
However, implementations of the AES S-box using the tower field approach result in several blocks acting in parallel on partially shared inputs. In such a situation, "local uniformity" of distributions does not necessarily lead to "global uniformity".
For example, let f, g be two functions acting on the same input x. Then, even if F, G are uniform circuits, producing uniform $Y_1 = F(X)$ and $Y_2 = G(X)$, this does not imply that (Y_1, Y_2) is uniform.

Affine Equivalence Classes

S_1 and S_2 are affine equivalent if there exists affine mappings A and B s.t. $S_1 = B \circ S_2 \circ A$.

	3×3 Sboxes	4×4 Sboxes
Affine	1	1
Quadratic	3	6
Cubic	-	295

Affine Equivalence Classes

S_1 and S_2 are affine equivalent if there exists affine mappings A and B s.t. $S_1 = B \circ S_2 \circ A$.

	3×3 Sboxes	4×4 Sboxes
Affine	1	1
Quadratic	3	6
Cubic	-	295

- For all $n \geq 3$, $n \times n$ affine bijections are in alternating group A_{2^n}

Affine Equivalence Classes

S_1 and S_2 are affine equivalent if there exists affine mappings A and B s.t. $S_1 = B \circ S_2 \circ A$.

	3×3 Sboxes	4×4 Sboxes
Affine	1	1
Quadratic	3	6
Cubic	-	295

- For all $n \geq 3$, $n \times n$ affine bijections are in alternating group A_{2^n}
- All 4×4 quadratic Sboxes are in A_{16}

Examples

Class1 ANF form of $F(w, v, u)$[01234576]

$$\begin{aligned} F1 &= 0 + u + w * v \\ F2 &= 0 + v \\ F3 &= 0 + w \end{aligned}$$

Class2 ANF form of $F(w, v, u)$[01234675]

$$\begin{aligned} F1 &= 0 + u + w * u + w * v \\ F2 &= 0 + v + w * u \\ F3 &= 0 + w \end{aligned}$$

Class3 ANF form of F(w,v,u)[01243675]

$$\begin{aligned} F1 &= 0 + u + v * u + w \\ F2 &= 0 + v + v * u + w + w * v \\ F3 &= 0 + v * u + w * u + w * v \end{aligned}$$

KATHOLIEKE UNIVERSITEIT LEUVEN

Computing with S-boxes

$$S_1[01243675] = A[01326754] \circ S_2[05326147] \circ B[05273614]$$

$$S_2[05326147] = A^{-1}[01327645] \circ S_1[01243675] \circ B^{-1}[06247153]$$

$$\begin{array}{llll} 01234567 & & & \\ 05273614 & 1 \to 5 & 4 \to 3 & 5 \to 6 \\ 05326147 & 5 \to 1 & 3 \to 2 & 6 \to 4 \\ 01326754 & 1 \to 1 & 2 \to 3 & 4 \to 6 \end{array}$$

KATHOLIEKE UNIVERSITEIT LEUVEN

KATHOLIEKE UNIVERSITEIT LEUVEN

Direct Sharing

$$
\begin{aligned}
S(x, y, z) &= x + yz \\
S_1 &= x_2 + y_2 z_2 + y_2 z_3 + y_3 z_2 \\
S_2 &= x_3 + y_3 z_3 + y_3 z_1 + y_1 z_3 \\
S_3 &= x_1 + y_1 z_1 + y_1 z_2 + y_2 z_1
\end{aligned}
$$

KATHOLIEKE UNIVERSITEIT LEUVEN

Direct Sharing

$$
\begin{aligned}
S(x,y,z) &= x + yz \\
S_1 &= x_2 + y_2z_2 + y_2z_3 + y_3z_2 \\
S_2 &= x_3 + y_3z_3 + y_3z_1 + y_1z_3 \\
S_3 &= x_1 + y_1z_1 + y_1z_2 + y_2z_1
\end{aligned}
$$

	3×3 Sboxes	$4 \times 4 Sboxes$
Affine	1/1	1/1
Quadratic	1/3	3/6
Cubic	-	0/295

Direct Sharing

	3×3 Sboxes	$4 \times 4 Sboxes$
Affine	A_0^3	A_0^4
Quadratic	Q_1^3, Q_2^3, Q_3^3	Q_4^4, Q_{12}^4, Q_{293}^4, Q_{294}^4, Q_{299}^4, Q_{300}^4

Direct Sharing

	3×3 Sboxes	$4 \times 4 Sboxes$
Affine	A_0^3	A_0^4
Quadratic	Q_1^3, Q_2^3, Q_3^3	Q_4^4, Q_{12}^4, Q_{293}^4, Q_{294}^4, Q_{299}^4, Q_{300}^4

Q: What is the relation?

Direct Sharing

	3×3 Sboxes	$4 \times 4 Sboxes$
Affine	A_0^3	A_0^4
Quadratic	Q_1^3, Q_2^3, Q_3^3	Q_4^4, Q_{12}^4, Q_{293}^4, Q_{294}^4, Q_{299}^4, Q_{300}^4

Q: What is the relation?

A:

$$\begin{aligned} Q_1^3 &\rightarrow Q_4^4 \\ Q_2^3 &\rightarrow Q_{12}^4 \\ Q_3^3 &\rightarrow Q_{300}^4 \end{aligned}$$

Direct Sharing

	3×3 Sboxes	$4 \times 4 Sboxes$
Affine	A_0^3	A_0^4
Quadratic	Q_1^3, Q_2^3, Q_3^3	Q_4^4, Q_{12}^4, Q_{293}^4, Q_{294}^4, Q_{299}^4, Q_{300}^4

Q: What is the relation?

A:

$$\begin{aligned} Q_1^3 &\rightarrow Q_4^4 \\ Q_2^3 &\rightarrow Q_{12}^4 \\ Q_3^3 &\rightarrow Q_{300}^4 \end{aligned}$$

$$S(w,v,u) = (y1,y2,y3) \rightarrow S(x,w,v,u) = (y1,y2,y3,x)$$

KATHOLIEKE UNIVERSITEIT LEUVEN

Correction Terms

$$
\begin{aligned}
S(x, y, z) &= x + yz \\
S_1 &= \cancel{x_2} + y_2 z_2 + y_2 z_3 + y_3 z_2 + \cancel{x_2} + x_3 \\
S_2 &= \cancel{x_3} + y_3 z_3 + y_3 z_1 + y_1 z_3 + \cancel{x_3} + x_1 \\
S_3 &= \cancel{x_1} + y_1 z_1 + y_1 z_2 + y_2 z_1 + \cancel{x_1} + x_2
\end{aligned}
$$

Correction Terms

$$
\begin{aligned}
S(x,y,z) &= x + yz \\
S_1 &= \cancel{x_2} + y_2z_2 + y_2z_3 + y_3z_2 + \cancel{x_2} + x_3 \\
S_2 &= \cancel{x_3} + y_3z_3 + y_3z_1 + y_1z_3 + \cancel{x_3} + x_1 \\
S_3 &= \cancel{x_1} + y_1z_1 + y_1z_2 + y_2z_1 + \cancel{x_1} + x_2
\end{aligned}
$$

	3×3 S-boxes	4×4 S-boxes
Affine	A_0	A_0
Quadratic	Q_1, Q_2, Q_3	Q_4, Q_{12}, Q_{293}, Q_{294}, Q_{299}, Q_{300}

KATHOLIEKE UNIVERSITEIT LEUVEN

Correction Terms

$$
\begin{aligned}
S(x,y,z) &= x + yz \\
S_1 &= \cancel{x_2} + y_2z_2 + y_2z_3 + y_3z_2 + \cancel{x_2} + x_3 \\
S_2 &= \cancel{x_3} + y_3z_3 + y_3z_1 + y_1z_3 + \cancel{x_3} + x_1 \\
S_3 &= \cancel{x_1} + y_1z_1 + y_1z_2 + y_2z_1 + \cancel{x_1} + x_2
\end{aligned}
$$

	3×3 S-boxes	4×4 S-boxes
Affine	A_0	A_0
Quadratic	Q_1, Q_2, Q_3	Q_4, Q_{12}, Q_{293}, Q_{294}, Q_{299}, Q_{300}

Work for n shares with m variables is $2^{3(m+\binom{m}{2})n}$

3x3 S-box with 3 shares $2^{18\times 3} = 2^{54}$

KATHOLIEKE UNIVERSITEIT LEUVEN

Properties of the sharing (1)

Theorem

If there exists a proper sharing for an Sbox S, every Sbox that belongs to the same class with S can be shared.

Example: Consider mini-Keccak $mK \in Q_3^3$

$$mK_i = x^i + x^{i+2} + x^{i+2} * x^{i+1}$$

The function is rotation symmetric and the index i is taken mod 3. An affine equivalent S-box S is obtained from mK by changing the variables $(x^0, x^1, x^2) \rightarrow (x^0 + x^2, x^1, x^2)$

$$\begin{aligned} S_0 &= x^0 + \cancel{x^2} + x^1 * x^2 + \cancel{x^2} \\ S_1 &= x^1 + x^0 + \cancel{x^2} + x^2 * x^0 + \cancel{x^2} \\ S_2 &= x^2 + x^1 + x^0 * x^1 + x^1 * x^2 \end{aligned}$$

KATHOLIEKE UNIVERSITEIT LEUVEN

Properties of the sharing (2)

The latter can be written also as $S = mK \circ \mathcal{A}$, where $\mathcal{A}$ is a linear transformation.

$$\mathcal{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} x^0 \\ x^1 \\ x^2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

In general $\mathcal{A}$ consists of a matrix A and affine vector b (here 0).

Q: Can we find an uniform *direct* sharing for mini Keccak mK with 5 shares?
A: We cannot, but we can find uniform *direct* sharing for the affine equivalent S-box S.

KATHOLIEKE UNIVERSITEIT LEUVEN

Properties of the sharing (3)

Let the linear term u and the quadratic term uv be shared as follows:

$u \to (u_2, u_3, u_4, u_5, u_1)$

$uv \to ((v_2 + v_3 + v_4 + v_5)(u_2 + u_3 + u_4 + u_5), v_1(u_3 + u_4 + u_5) + u_1(v_3 + v_4 + v_5) + u_1 v_1, v_1 u_2 + u_1 v_2, 0, 0)$

Let's denote by $\tilde{S}$ the shared S-box S.
We take the first shares of S_0, S_1 and S_2, then the second shares, and so on finishing with the 5-th shares of S.

Properties of the sharing (4)

Note that $mK = S \circ \mathcal{A}$ since $\mathcal{A}^{-1} = \mathcal{A}$.
Now we construct the affine (here the linear) transformation for the sharing $\tilde{\mathcal{A}}$ by applying the $\mathcal{A}^{-1}$ affine transform to each tuple of shares (x_i^0, x_i^1, x_i^2) for $i = 1, \ldots, 5$.

$$\tilde{\mathcal{A}} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} x_i^0 \\ x_i^1 \\ x_i^2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$\widetilde{mK} = \tilde{S} \circ \tilde{\mathcal{A}}$ is an uniform sharing for mK.

Properties of the sharing (5)

The final result is:

$$mK_{i,1} \triangleq x_2^i + x_2^{i+2} + ((x_2^{i+2} + x_3^{i+2} + x_4^{i+2} + x_5^{i+2})(x_2^{i+1} + x_3^{i+1} + x_4^{i+1} + x_5^{i+1}))$$
$$mK_{i,2} \triangleq x_3^i + x_3^{i+2} + (x_1^{i+1}(x_3^{i+2} + x_4^{i+2} + x_5^{i+2}) + x_1^{i+2}(x_3^{i+1} + x_4^{i+1} + x_5^{i+1}) + x_1^{i+1}x_1^{i+2})$$
$$mK_{i,3} \triangleq x_4^i + x_4^{i+2} + (x_1^{i+1}x_2^{i+2} + x_1^{i+2}x_2^{i+1})$$
$$mK_{i,4} \triangleq x_5^i + x_5^{i+2}$$
$$mK_{i,5} \triangleq x_1^i + x_1^{i+2}$$

for $i = 0, 2$

$$mK_{1,1} \triangleq x_2^1 + (x_2^0 + x_3^0 + x_4^0 + x_5^0) + ((x_2^0 + x_3^0 + x_4^0 + x_5^0)(x_2^2 + x_3^2 + x_4^2 + x_5^2))$$
$$mK_{1,2} \triangleq x_3^1 + x_1^0 + (x_1^2(x_3^0 + x_4^0 + x_5^0) + x_1^0(x_3^2 + x_4^2 + x_5^2) + x_1^2x_1^0)$$
$$mK_{1,3} \triangleq x_4^1 + (x_1^2x_2^0 + x_1^0x_2^2)$$
$$mK_{1,4} \triangleq x_5^1$$
$$mK_{1,5} \triangleq x_1^1$$

Note that the direct sharing of mK has to change for equation 1 in order to achieve uniformity.

KATHOLIEKE UNIVERSITEIT LEUVEN

Properties for sharing (6)

On my web-page a SW-framework for sharing/decomposing small S-boxes is available
`http://homes.esat.kuleuven.be/~snikova/ti_tools.html`

The sharing process:
1. For 3, 4 or 5 shares use the "direct sharing" and search for an affine equivalent S-box which can be uniformly shared.
2. Find the affine transformation between these two S-boxes.
3. Return the direct sharing back to the targeted S-box.

Decomposition

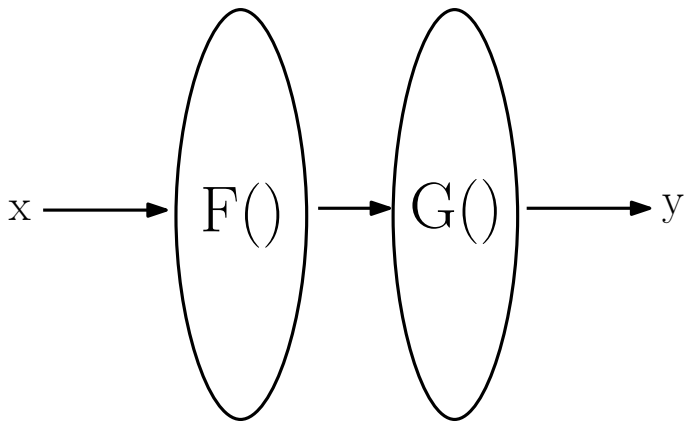
Idea [Poschmann et al., J.Cryptology'11]

Generate S-boxes by combination of others

Decomposition

Idea [Poschmann et al., J.Cryptology'11]

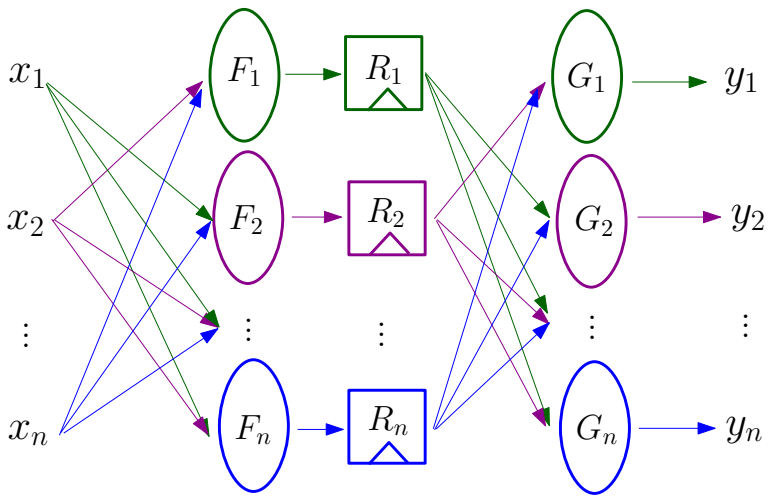
Generate S-boxes by combination of others



Decomposition

Idea [Poschmann et al., J.Cryptology'11]

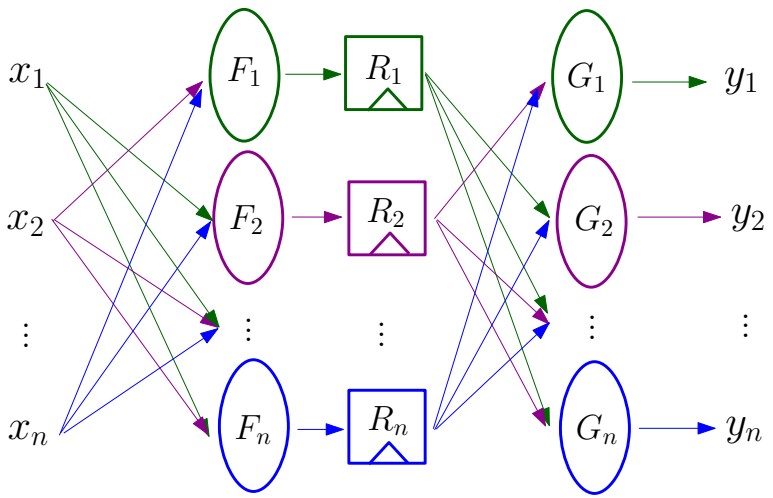
Generate S-boxes by combination of others



Decomposition

Idea [Poschmann et al., J.Cryptology'11]

Generate S-boxes by combination of others

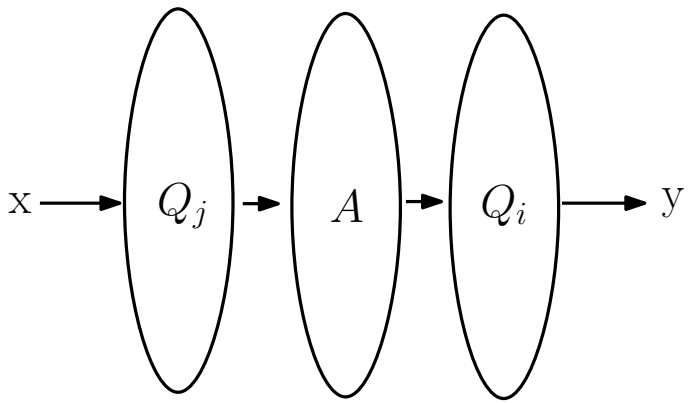


Present S-box (4×4):

Q_{12}	$\times$	Q_{12}
Q_{293}	$\times$	Q_{300}
Q_{294}	$\times$	Q_{299}
Q_{299}	$\times$	Q_{294}
Q_{299}	$\times$	Q_{299}
Q_{300}	$\times$	Q_{293}
Q_{300}	$\times$	Q_{300}

KATHOLIEKE UNIVERSITEIT LEUVEN

Decomposition



Lemma

All cubic permutations S, that have decomposition length 2, are affine equivalent to

$$S_{ixj} = Q_i \circ A \circ Q_j$$

where $i, j \in \{4, 12, 293, 294, 299, 300\}$

Decomposition

Theorem

A 4×4 bijection can be decomposed using quadratic bijections if and only it belongs to A_{16}.

Decomposition

Theorem

A 4×4 bijection can be decomposed using quadratic bijections if and only it belongs to A_{16}.

Lemma

Let $\tilde{S}$ be a permutation in $S_{16} \setminus A_{16}$, then any permutation from $S_{16} \setminus A_{16}$ can be represented as a product of $\tilde{S}$ and a permutation from A_{16}

Overview of Classes

Overview of # of classes w.r.t # of shares and layers of decomposition

	unshared			3 shares				4 shares			5 shares
# of layers	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**1**
quadratic	6			5	1			6			6
cubics in A_{16}		30			28	2			30		30
cubics in A_{16}			114			113	1			114	114
cubics in $S_{16} \setminus A_{16}$		-			-			4	22	125	151

Overview of Classes

Overview of # of classes w.r.t # of shares and layers of decomposition

	unshared			3 shares				4 shares			5 shares
# of layers	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**1**
quadratic	6			5	1			6			6
cubics in A_{16}		30			28	2			30		30
cubics in A_{16}			114			113	1			114	114
cubics in $S_{16} \setminus A_{16}$		-			-			4	22	125	151

Results

We can share

- All quadratic S-boxes with 3 shares

Results

We can share

- All quadratic S-boxes with 3 shares
- Almost half of the cubic S-boxes with 3 shares with at most 4 decomposition layers

Results

We can share

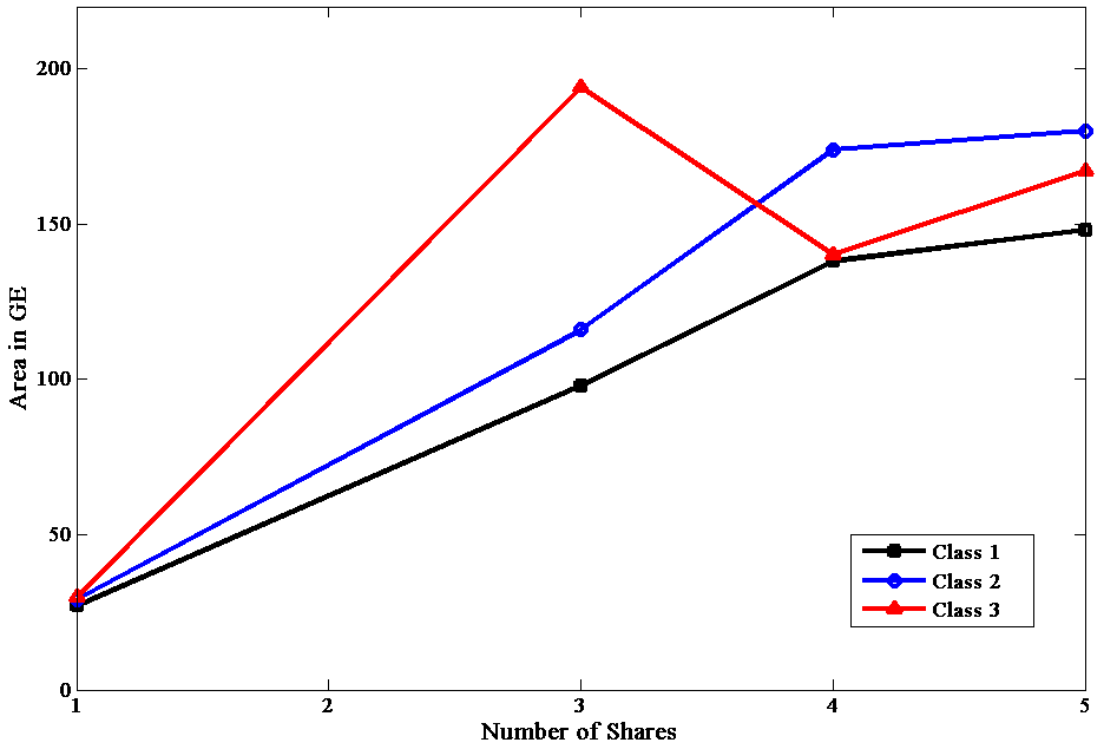
- All quadratic S-boxes with 3 shares
- Almost half of the cubic S-boxes with 3 shares with at most 4 decomposition layers
- All S-boxes with 4 shares with at most 3 decomposition layers

Results

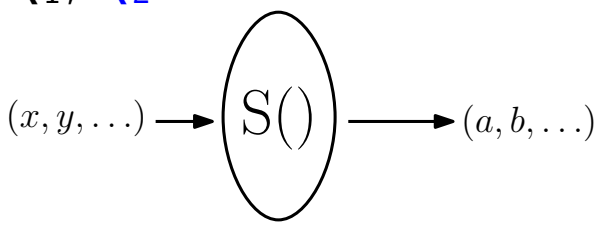
We can share

- All quadratic S-boxes with 3 shares
- Almost half of the cubic S-boxes with 3 shares with at most 4 decomposition layers
- All S-boxes with 4 shares with at most 3 decomposition layers
- All S-boxes with 5 shares without decomposition

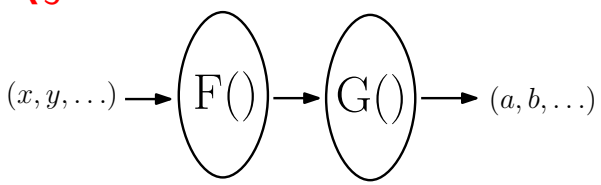
Quadratic 3 × 3 S-boxes



Q_1, Q_2:

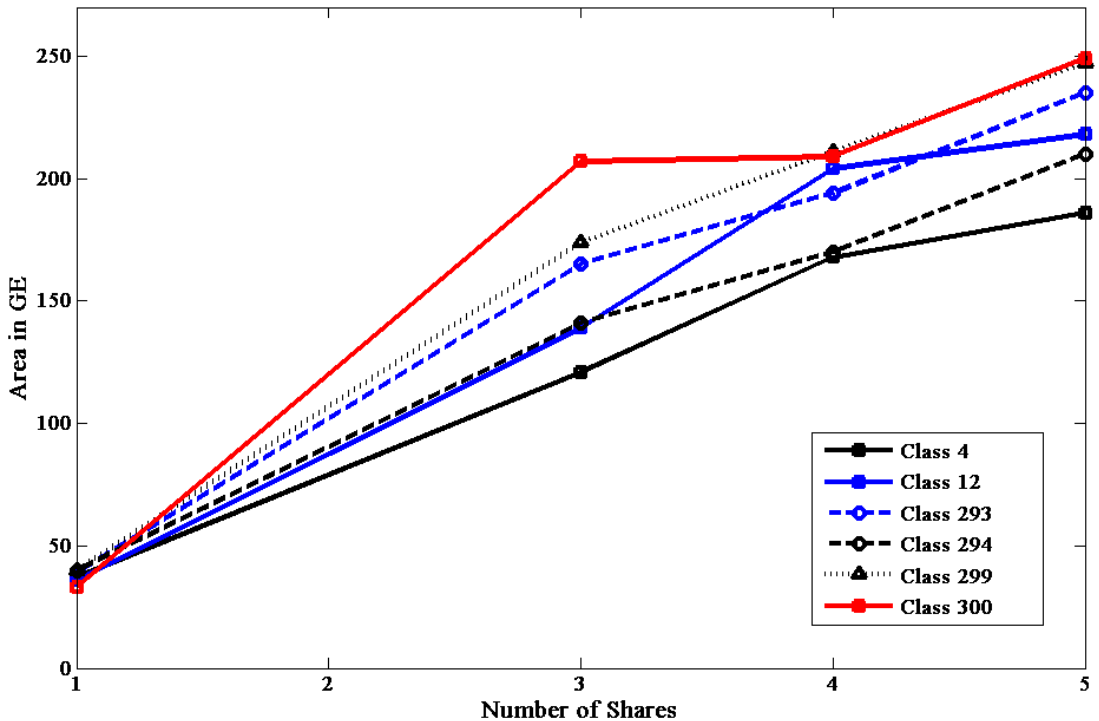


Q_3:

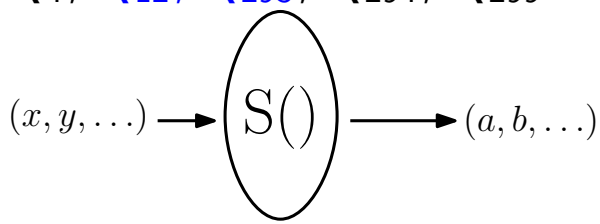


TSMC 0.18μm standard cell library

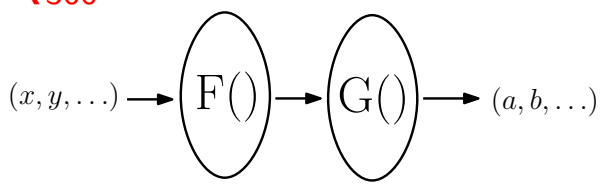
Quadratic 4 × 4 S-boxes



Q_4, Q_{12}, Q_{293}, Q_{294}, Q_{299}:

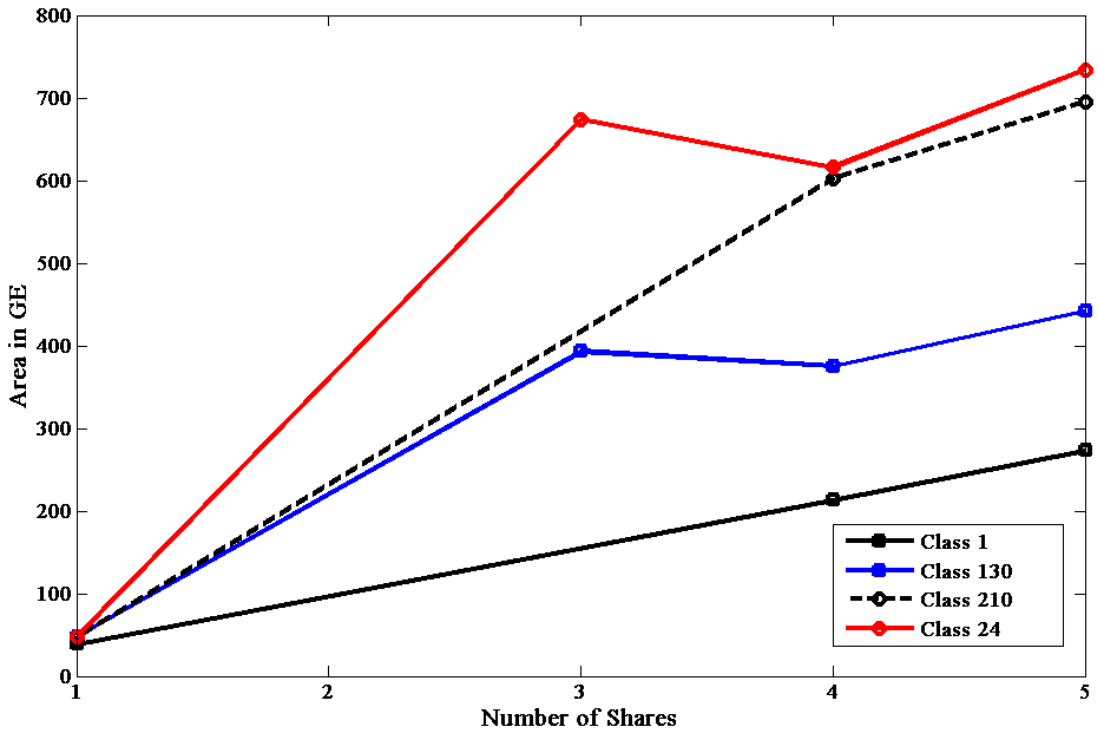


Q_{300}:



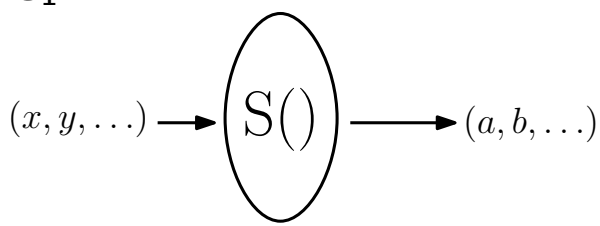
TSMC 0.18μm standard cell library

Cubic 4 × 4 S-boxes

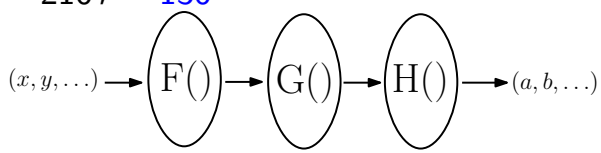


TSMC 0.18μm standard cell library

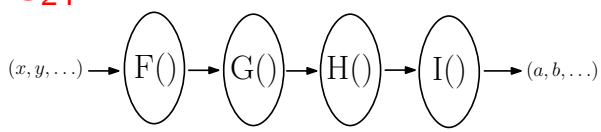
C_1:



C_{210}, C_{130}:



C_{24}:



Quadratic Sboxes in S_8

3×3 S-boxes Class # in S_8		Sharing Length (L)	Original S-box	Unshared Decomposed L reg	Shared 3 shares L reg	Shared 4 shares 1 reg	Shared 5 shares 1 reg
$\mathcal{Q}_1^3$	Min	1	27.66	-	98.66	138.00	148.00
	Max		29.66		121.66	150.00	185.66
$\mathcal{Q}_2^3$	Min	1	29.00	-	116.66	174.00	180.00
	Max		29.66		155.00	226.66	220.33
$\mathcal{Q}_3^3$	Min	2	30.00	50.00	194.33	140.00	167.00
	Max		32.00	51.00	201.00	194.33	228.66

TSMC 0.18μm standard cell library

KATHOLIEKE UNIVERSITEIT LEUVEN

Quadratic Sboxes in S_{16}

4×4 S-boxes Quadratic Class # in S_{16}		Sharing Length (L)	Original S-box	Unshared Decomposed L reg	Shared 3 shares L reg	Shared 4 shares 1 reg	Shared 5 shares 1 reg
$\mathcal{Q}_4^4$	Min	1	37.33	-	121.33	168.33	186.33
	Max		44.00		223.33	258.00	309.00
$\mathcal{Q}_{12}^4$	Min	1	36.66	-	139.33	204.00	218.00
	Max		48.00		253.33	290.33	340.66
$\mathcal{Q}_{293}^4$	Min	1	39.33	-	165.33	194.33	235.00
	Max		48.66		297.33	313.00	358.33
$\mathcal{Q}_{294}^4$	Min	1	40.00	-	141.33	170.33	210.33
	Max		49.66		261.00	240.00	255.00
$\mathcal{Q}_{299}^4$	Min	1	40.33	-	174.33	211.00	247.00
	Max		48.00		298.00	295.33	294.66
$\mathcal{Q}_{300}^4$	Min	2	33.66	58.00	207.33	209.66	249.33
	Max		52.66	70.00	346.00	295.00	342.33

TSMC 0.18μm standard cell library

KATHOLIEKE UNIVERSITEIT LEUVEN

Cubic Sboxes in S_{16}

4×4 S-boxes Cubic Class # in S_{16}	Sharing Length (L, L')	Original S-box	Unshared Decomposed L' reg	Shared 3 shares L reg	Shared 4 shares L' reg	Shared 5 shares 1 reg
$\mathcal{C}_1^4 \in S_{16} \setminus A_{16}$	1,1	39.66		–	213.66	273.66
$\mathcal{C}_3^4 \in S_{16} \setminus A_{16}$	1,1	40.33		–	230.33	286.33
$\mathcal{C}_{13}^4 \in S_{16} \setminus A_{16}$	1,1	40.33		–	260.00	319.00
$\mathcal{C}_{301}^4 \in S_{16} \setminus A_{16}$	1,1	39.33		–	289.33	350.33
$\mathcal{C}_{150}^4 \in A_{16}$	2,2	46.33	71.66	305.33	430.66	414.33
$\mathcal{C}_{130}^4 \in A_{16}$	3,2	48.00	97.33	393.00	375.66	442.66
$\mathcal{C}_{24}^4 \in A_{16}$	4,3	48.33	151.33	674.00	616.66	734.66
$\mathcal{C}_{257}^4 \in S_{16} \setminus A_{16}$	2,2	47.66	73.66	-	486.00	594.00
$\mathcal{C}_{210}^4 \in S_{16} \setminus A_{16}$	3,3	47.66	119.33	-	602.00	695.33

TSMC 0.18μm standart cell library

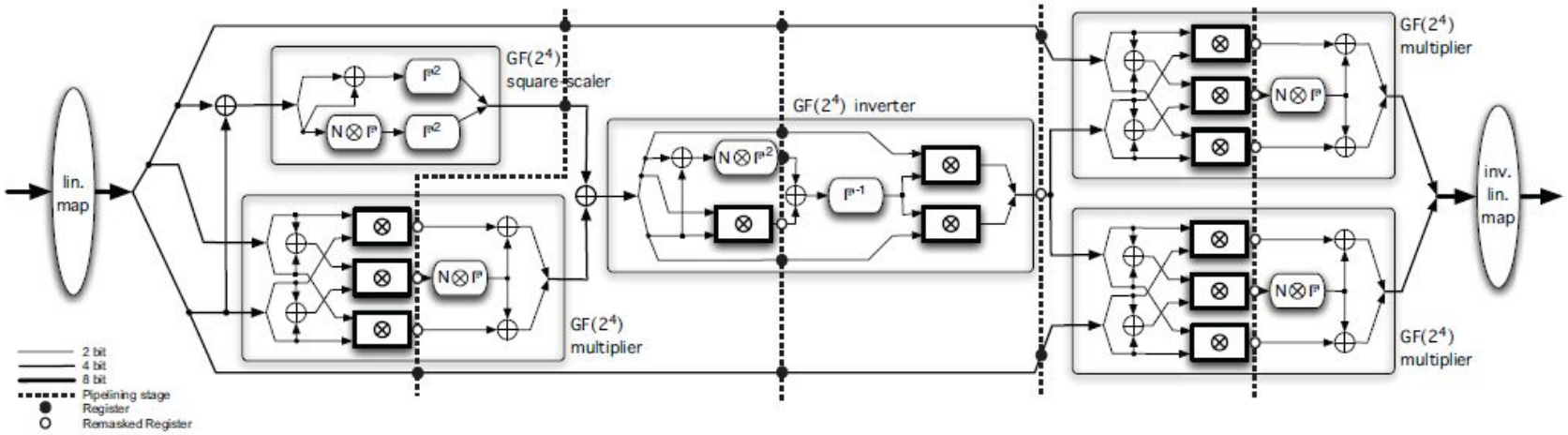
KATHOLIEKE UNIVERSITEIT LEUVEN

Cost Comparison

3 shares				4 shares			5 shares	remark
1	2	3	4	1	2	3	1	
3.6–5.2	6.3–6.5	–	–	5.0–7.6	–	–	5.4–7.4	quadratics in S_8
3.3–6.2	6.2–6.6	–	–	4.3–6.4	–	–	5.1–7.4	quadratics in S_{16}
–	6.0–6.6	7.7–8.2	13.9	–	7.3–9.3	12.8	8.2–15.2	cubics in A_{16}
–	–	–	–	5.4–10.2	8.4–10.2	12.6	10.2–14.6	cubics in $S_{16} \setminus A_{16}$

AES - Pushing the limits

[Moradi et al., Eurocrypt 2011]



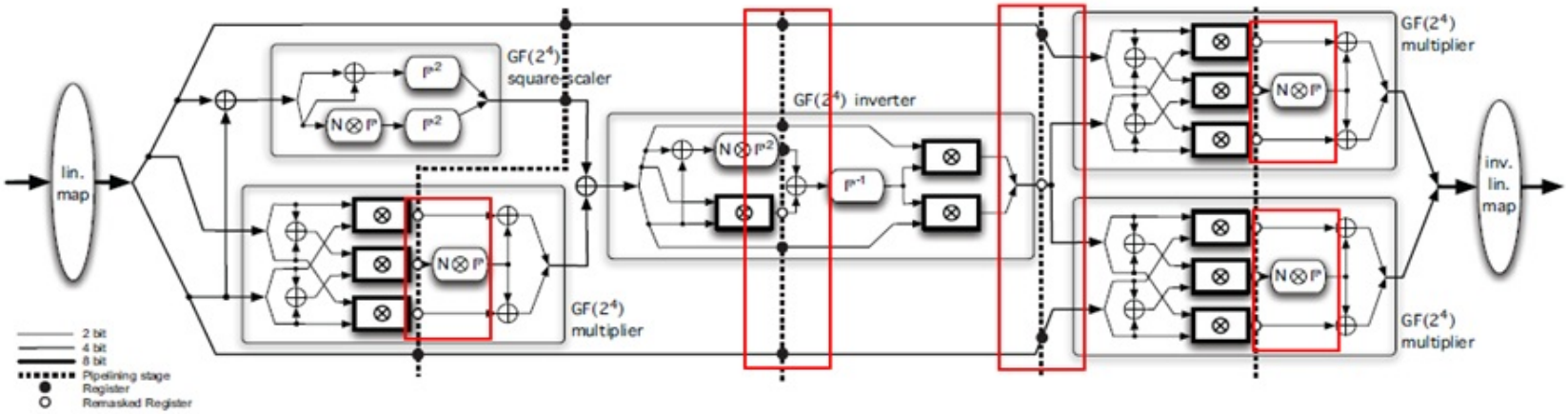
Composite field representation of the S-box [Canright, CHES 2005]. The thick lined rectangles are multipliers in $GF(4)$, which are the only non-linear parts.

The S-box is split in 5 pipelined stages (4 registers increase the area cost).

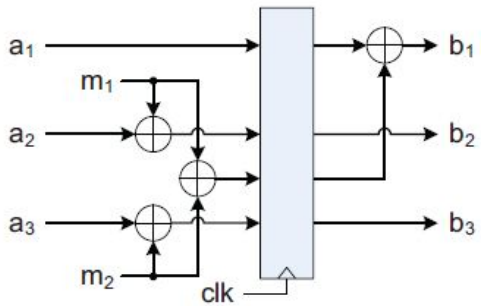
Although uniform sharing is used the parallel implementation destroys the "global uniformity" and the authors have to use re-sharing.

KATHOLIEKE UNIVERSITEIT LEUVEN

AES - Pushing the limits



To achieve "global uniformity" the authors have to use re-sharing (48 bits per S-box call).



AES - More Efficient TI

As a starting point we use the composite field representation of the S-box [Canright, CHES 2005].
Our approach:

- Uniform sharing on bigger blocks e.g. working in $GF(2^4)$ or even in $GF(2^8)$.
- Using 3 shares is not always giving best result.
- Uniformity can be relaxed and non-uniform sharings can be used too.

We have two versions: one version with uniformity satisfied and second version with relaxed uniformity.

KATHOLIEKE UNIVERSITEIT LEUVEN

AES TI - Comparison

Recall [Poschmann et al., JoC 2010] results:
Present S-box - 32 GE - TI shared 355 GE (1109%).
Present cipher - 1111 GE (in 547 cycles)
TI shared 3582 GE i.e. 322% (in 578 cycles i.e. 106%).
[Moradi et al., Eurocrypt 2011]
AES S-box - 233 GE; AES cipher - 2601 GE (in 226 cycles).

AES TI - Comparison

Recall [Poschmann et al., JoC 2010] results:
Present S-box - 32 GE - TI shared 355 GE (1109%).
Present cipher - 1111 GE (in 547 cycles)
TI shared 3582 GE i.e. 322% (in 578 cycles i.e. 106%).
[Moradi et al., Eurocrypt 2011]
AES S-box - 233 GE; AES cipher - 2601 GE (in 226 cycles).

	S-box	%	Total	%	cycles	%
Moradi et al.	4.2	1821	11.1	427	266	118
Version 1	4.2	1803	9.0	345	266	118
Version 2	3.0	1284	8.0	311	246	109

The TI shared S-box become smaller if the shares are chosen properly and the uniformity is used only when required. Naturally all these reflects in a smaller (total) implementation, with % closer to those of Present.

KATHOLIEKE UNIVERSITEIT LEUVEN

AES TI - Comparison

Recall [Poschmann et al., JoC 2010] results:
Present S-box - 32 GE - TI shared 355 GE (1109%).
Present cipher - 1111 GE (in 547 cycles)
TI shared 3582 GE i.e. 322% (in 578 cycles i.e. 106%).
[Moradi et al., Eurocrypt 2011]
AES S-box - 233 GE; AES cipher - 2601 GE (in 226 cycles).

	S-box	%	Total	%	cycles	%
Moradi et al.	4.2	1821	11.1	427	266	118
Version 1	4.2	1803	9.0	345	266	118
Version 2	3.0	1284	8.0	311	246	109

TI in general introduces a very small overhead in performance. However for complex S-boxes (as AES) we were able to achieve comparable area as simpler (e.g. Present) only at the additional request of random bits.

KATHOLIEKE UNIVERSITEIT LEUVEN

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes
- But number of decomposition layers are necessary

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes
- But number of decomposition layers are necessary
- TI is applied even to "complex" S-boxes as AES with similar overhead

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes
- But number of decomposition layers are necessary
- TI is applied even to "complex" S-boxes as AES with similar overhead
- Less number of shares does NOT imply smaller area

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes
- But number of decomposition layers are necessary
- TI is applied even to "complex" S-boxes as AES with similar overhead
- Less number of shares does NOT imply smaller area
- The number of shares CAN vary

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes
- But number of decomposition layers are necessary
- TI is applied even to "complex" S-boxes as AES with similar overhead
- Less number of shares does NOT imply smaller area
- The number of shares CAN vary
- TI is also performance efficient

KATHOLIEKE UNIVERSITEIT LEUVEN

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes
- But number of decomposition layers are necessary
- TI is applied even to "complex" S-boxes as AES with similar overhead
- Less number of shares does NOT imply smaller area
- The number of shares CAN vary
- TI is also performance efficient
- Uniformity remedy - e.g. resharing

Conclusion

- TI is extended to all "simpler" 3×3, 4×4 and DES S-boxes
- But number of decomposition layers are necessary
- TI is applied even to "complex" S-boxes as AES with similar overhead
- Less number of shares does NOT imply smaller area
- The number of shares CAN vary
- TI is also performance efficient
- Uniformity remedy - e.g. resharing
- But when resharing is used certain number of fresh randomness is required

Conclusion

- TI provides provable protection against 1-st order DPA even in presence of glitches

Conclusion

- TI provides provable protection against 1-st order DPA even in presence of glitches
- It requires few assumptions on the hardware leakage behavior

Conclusion

- TI provides provable protection against 1-st order DPA even in presence of glitches
- It requires few assumptions on the hardware leakage behavior
- in summary TI allows to construct secure realistic-size circuits without intervention and design iterations

Thank you!

KATHOLIEKE UNIVERSITEIT LEUVEN