KATHOLIEKE UNIVERSITEIT
LEUVEN

Threshold Implementations

Vincent Rijmen

Eurocrypt 2015

Product cipher [Shannon, 1949]

Cryptanalysis

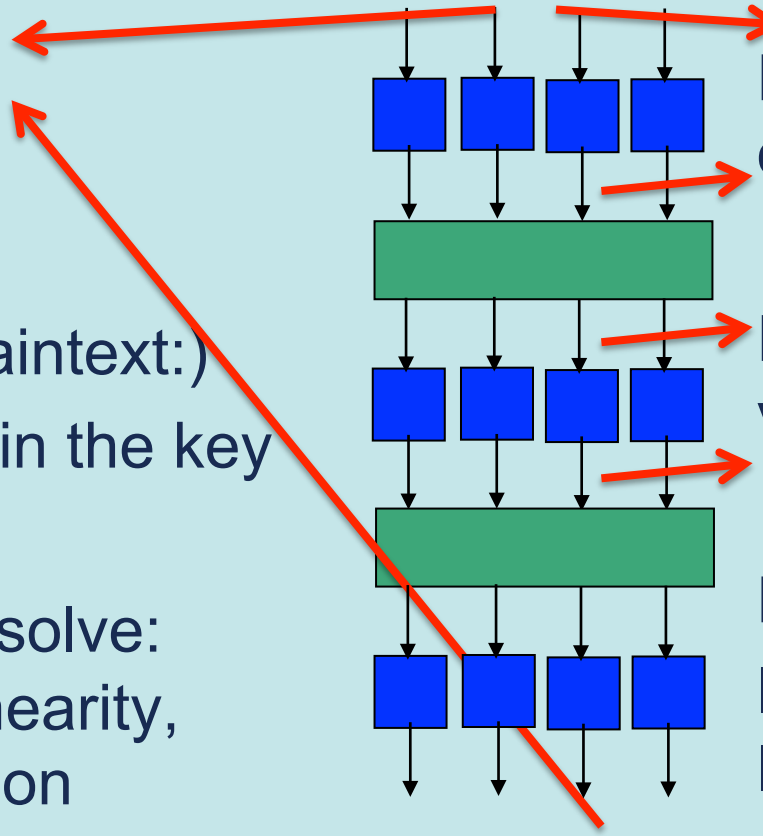
Side-channel attacks

$c = E(k,p)$

(Known plaintext:)

Equations in the key

Difficult to solve:
high nonlinearity,
high diffusion



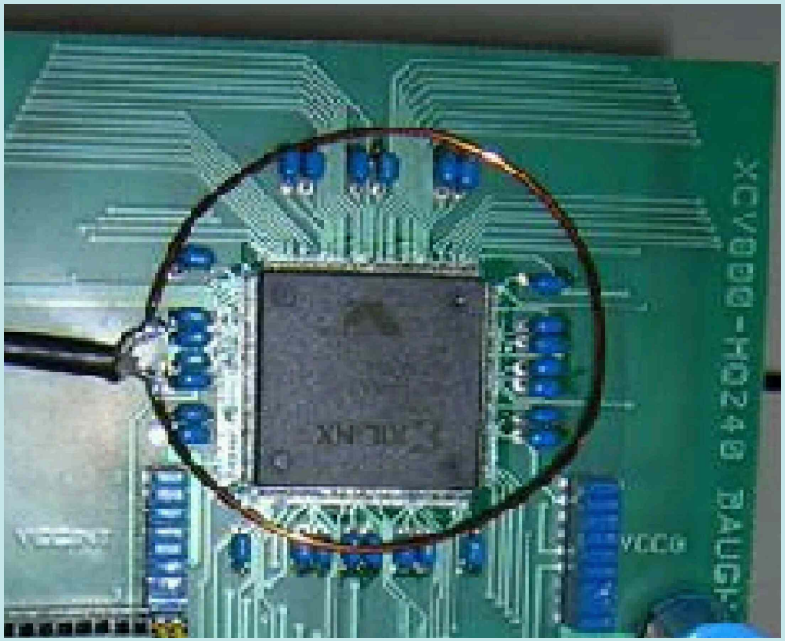
Device gives information on its internal state

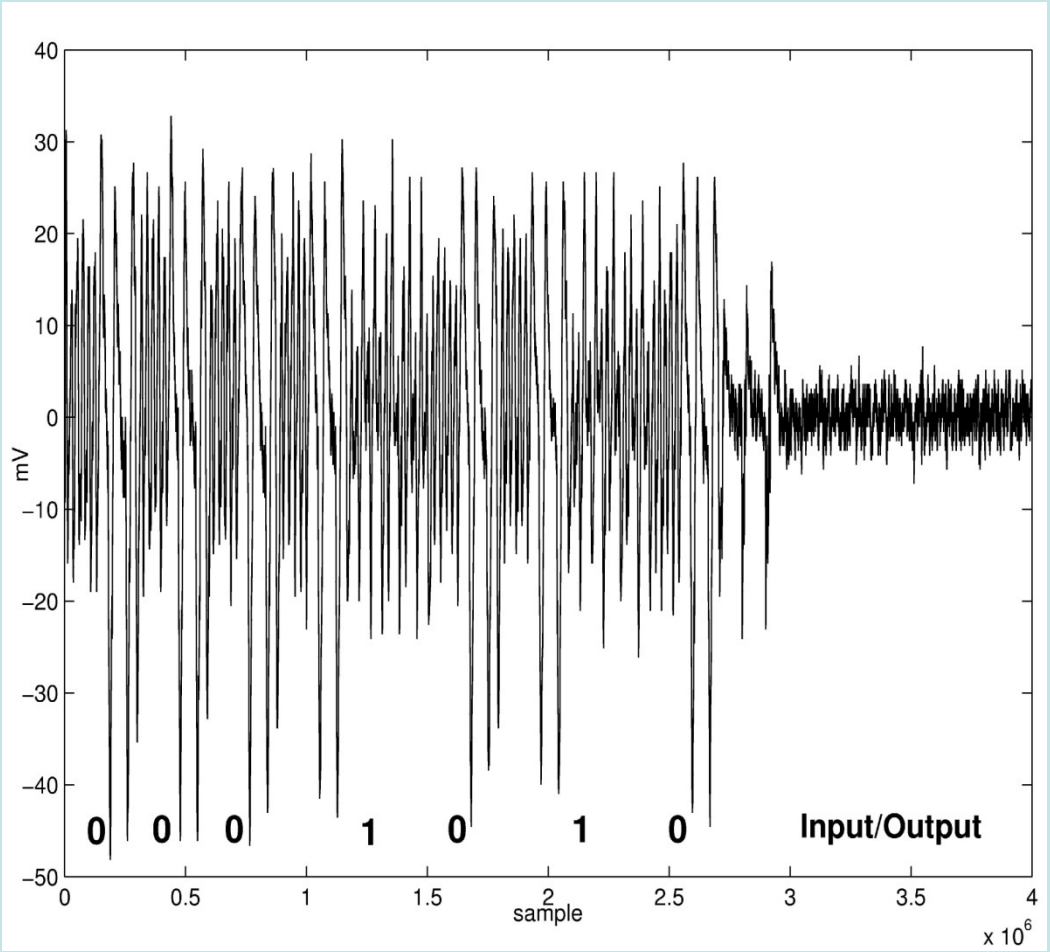
Equations in intermediate variables

More easy to solve:

low nonlinearity,
low diffusion

Power attacks (EM radiation)





Power consumption depends on:

- Instructions executed
- Data processed

Hardware sidechannels

- Switching a logic cell from “0” to “1” consumes energy
- The amount of energy depends on:
  - transistor design
  - process variations,
  - length of connection line,
  - crosstalk between lines,
  - ....
- Leaks Hamming weight of stored variables
  - Sometimes more information leaks

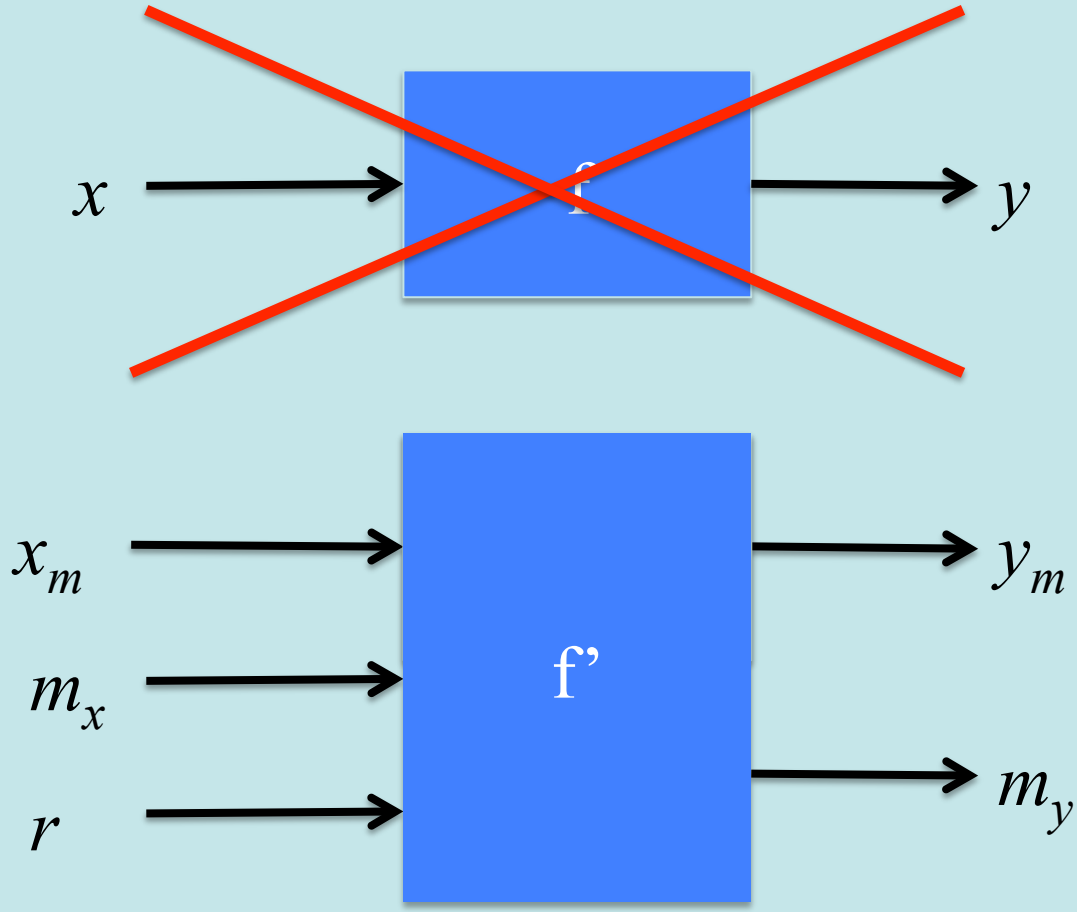
Countering power attacks

1. Balancing power consumption
   - Constant instruction sequence
   - Special hardware logic styles [Tiri+ 2003]

2. Masking [Chari+ 1999]
   - Removes correlation between secret key and data processed

3. Leakage-resilient cryptography
   - Ephemeral keys [Kocher 2005]

Masking



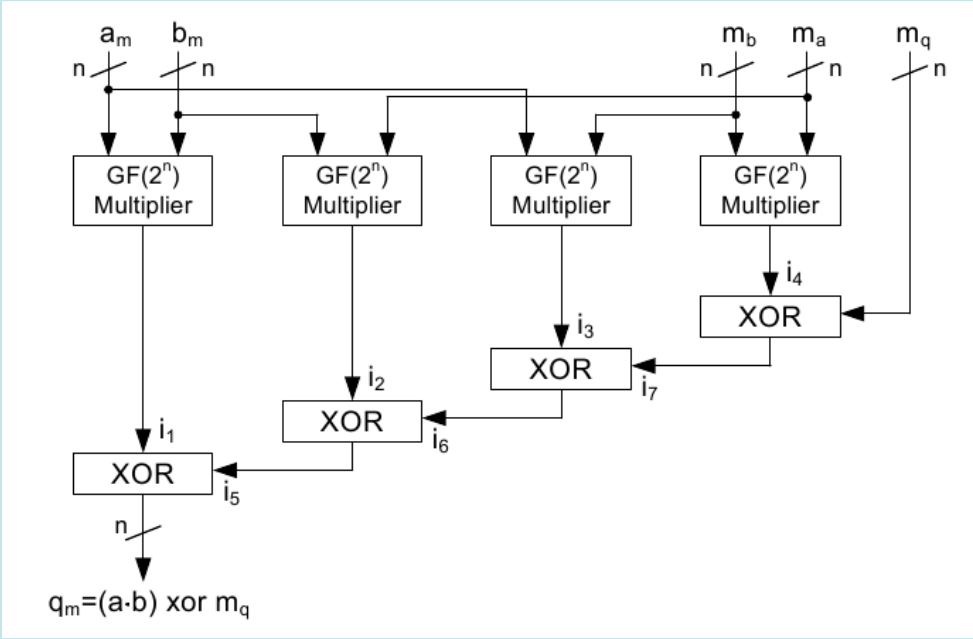
$$m_x + x_m = x$$
$$m_y + y_m = y$$

Private circuits [Ishai+ ‘03]

- f = XOR: easy
- f = AND: construction given in the paper
- f = anything else: combinations of these two
  - Circuit size $O(nt^2)$

- Proof of security

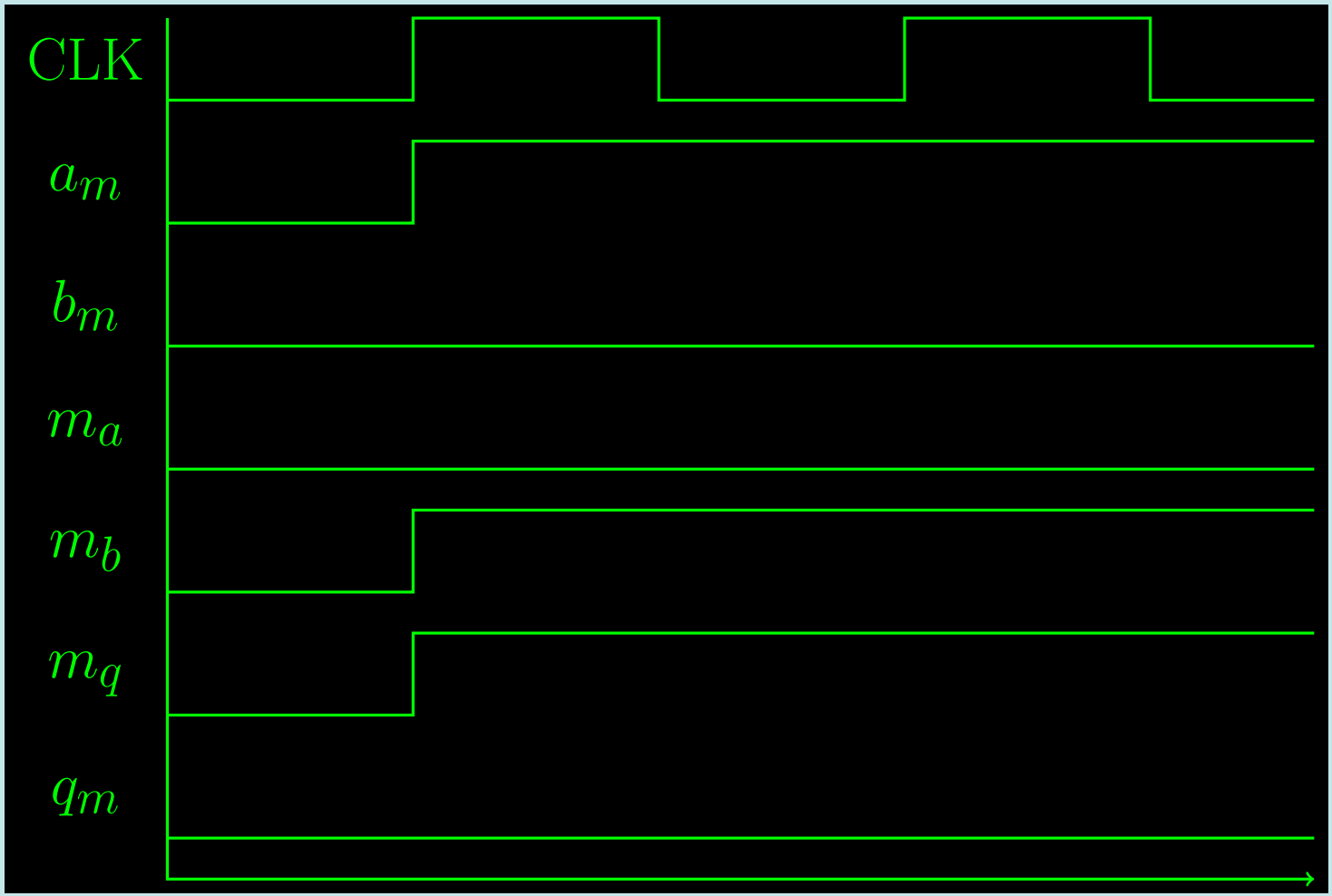
Masked multiplier [Trichina+ '04]



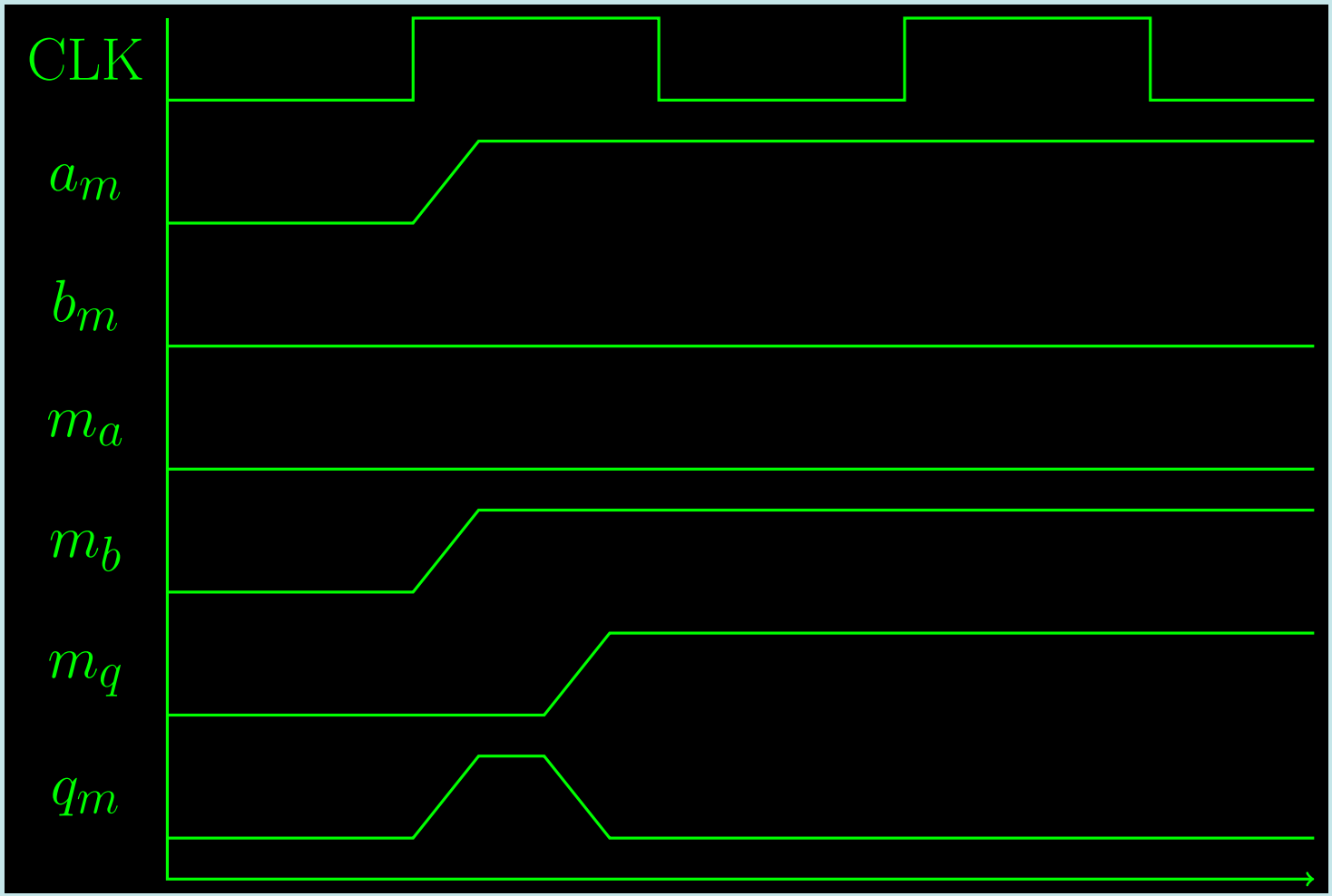
$$q_m = a_m b_m + (m_b a_m + (m_a b_m + (m_a m_b + m_q)))$$

- 1 multiplier becomes 4 multipliers + 4 XORs
- None of the signals is correlated to a, b or q
- Security proof
- Assumptions: computation model
  - Discrete-time
  - Imperative programming style

Logic analyser (digital view)



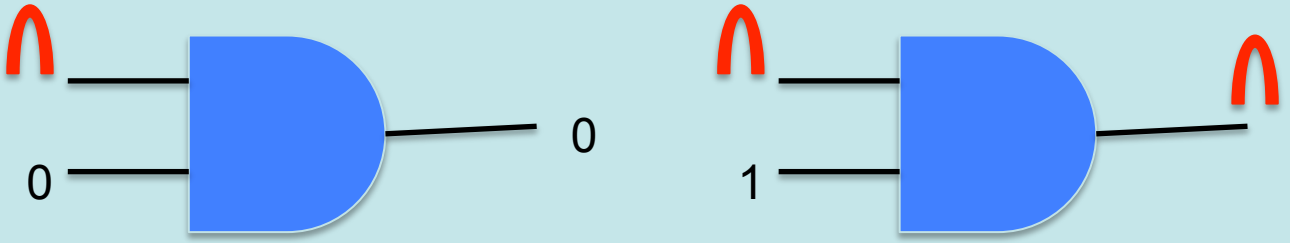
Signal analyzer (analog view)



Transient effects in hardware

- Changing values takes time: transition period
- Delays depend on details of the circuit lay-out
  - Race conditions
- Transient effects account for almost all the power consumption of CMOS circuits

Crisis



- Propagation of transient effects depends on the input of the combinational circuit
- Dependency is non-linear
- Modeling requires knowledge of low-level circuit details

- Security breakdown

Threshold implementations

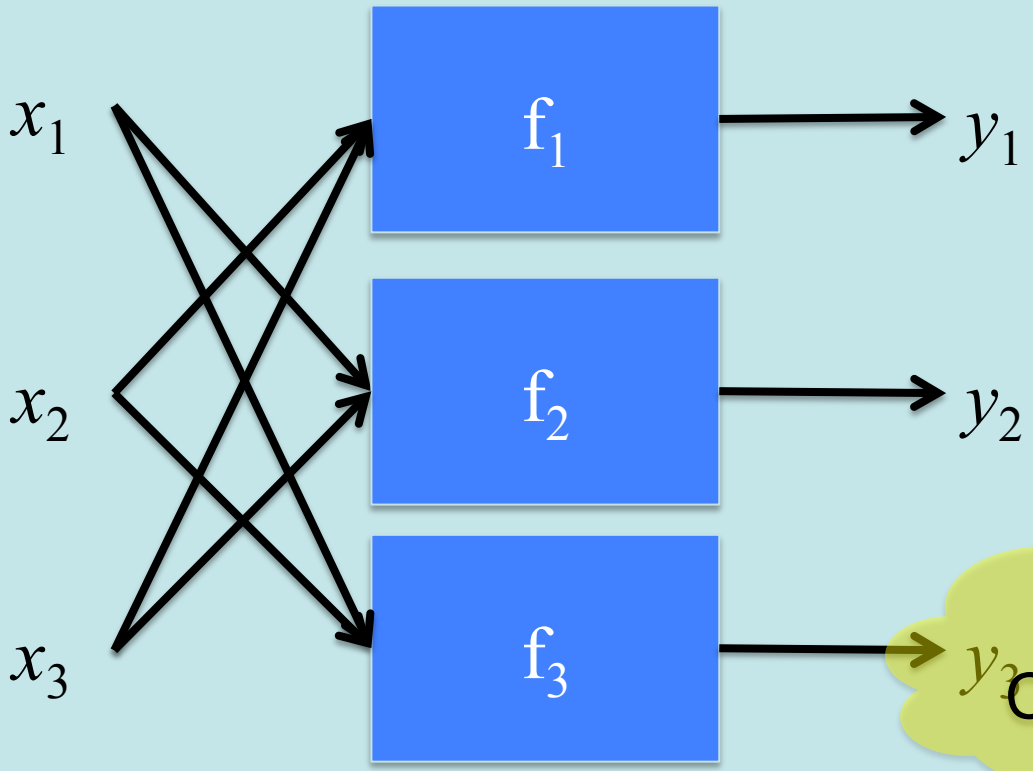
- Don’t rely on the behaviour of hardware implementations of combinational logic
- Assume that combinational logic leaks information on all its inputs

- Secret sharing

$$x_1 + x_2 + x_3 = x$$
$$y_1 + y_2 + y_3 = y$$

Non-completeness

$y = \mathrm{f}(x)$



Multi-Party Computation!

$$x_1 + x_2 + x_3 = x$$
$$y_1 + y_2 + y_3 = y$$

Security: main theorem

Average power consumption of the circuit is independent of x

- f_i depends only on (x_{i+1}, x_{i-1}), is independent of x_i
- Power consumption of f_i is independent of x_i
- If (x_{i+1}, x_{i-1}) independent of x, then the power consumption of f_i is independent of x

- Average power consumption of the circuit = sum of average power consumptions of f_i
- Hence, independent of x

Assumptions

- x_i uniformly random
  - Knowledge of n-1 shares gives no information on x
- f_i implementation depends only on $(x_1,\ldots, x_{i+1}, x_{i-1},\ldots, x_n)$:
  - No cross-talk from x_i
- Suitable f_i have to exist:

$$\mathrm{f}_1\left(x_2,x_3,\ldots,x_n\right)+\mathrm{f}_2\left(x_1,x_3,\ldots,x_n\right)+\cdots+\mathrm{f}_n\left(x_1,x_2,\ldots,x_{n-1}\right)=\mathrm{f}\left(x_1,x_2,\ldots,x_n\right)$$

  - Trivial for linear f
  - Research problem for most of the interesting f

Example: multiplier

- 3 shares

$$z = \mathrm{f}(x, y) = x \cdot y$$

$$z_1 = \mathrm{f}_1(x_2, x_3, y_2, y_3) = x_2 y_2 + x_2 y_3 + x_3 y_2$$

$$z_2 = \mathrm{f}_2(x_1, x_3, y_1, y_3) = x_3 y_3 + x_1 y_3 + x_3 y_1$$

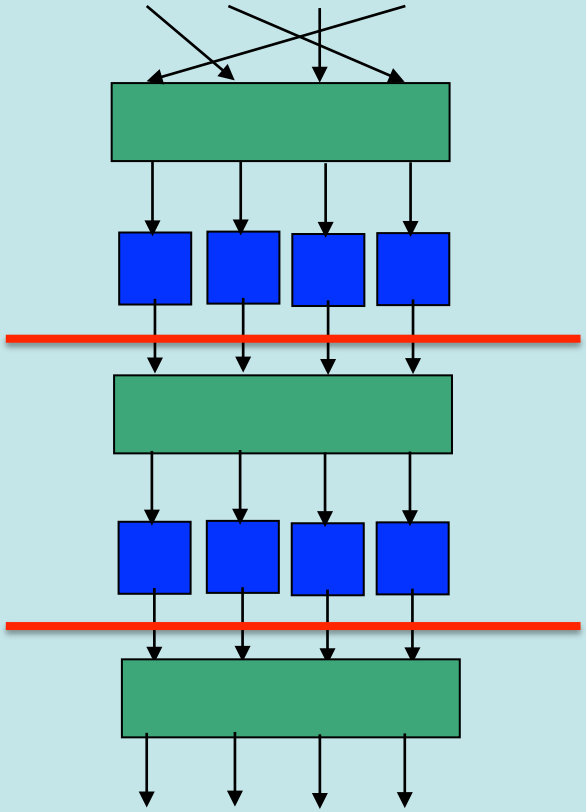
$$z_3 = \mathrm{f}_3(x_1, x_2, y_1, y_2) = x_1 y_1 + x_1 y_2 + x_1 y_3$$

- Secure, even with transient effects
- No extra random input required

Related approach [Prouff+ 2011]

- Shamir's secret sharing
- BGW secure multiparty computation protocol
  - Construction for f_i
- Circuit size O(t^3), extra randomness O(t^2)

Arbitrary functions (ciphers)



- Hardware size increases with the number of shares
- Functions with algebraic degree n require $n+1$ shares
- Strong ciphers have high algebraic degree

Registers

- Combinational logic between registers has lower algebraic degree
- Registers limit propagation of transient effects
- Protect each stage individually

Assumptions!

- The inputs of each stage need to be uniformly distributed
- The input of the 2nd step = output of 1st step
- Outputs of 1st step need to be uniformly distributed
  - remasking, or
  - extra property for f_i

Extra property for (f_1,f_2,…)

$$\forall y \in \mathrm{F}_{\mathrm{out}}, \forall \left(y_1, y_2, \ldots, y_{s_{\mathrm{out}}}\right), \forall x \in \mathrm{F}_{\mathrm{in}} \text{ with } f(x) = y:$$

$$\#\left\{\left(x_1, x_2, \ldots, x_{s_{\mathrm{in}}}\right) \mid \left(x_1, x_2, \ldots, x_{s_{\mathrm{in}}}\right) \rightarrow \left(y_1, y_2, \ldots, y_{s_{\mathrm{out}}}\right)\right\} = \frac{\left(\#\mathrm{F}_{\mathrm{in}}\right)^{s_{in}-1}}{\left(\#\mathrm{F}_{\mathrm{out}}\right)^{s_{\mathrm{out}}-1}}$$

- With s_{in}, s_{out} be the number of shares in input, output
- Always $x_1 + x_2 + \cdots + x_{s_{\mathrm{in}}} = x$ and $y_1 + y_2 + \cdots + y_{s_{\mathrm{out}}} = y$

In words

Every y_i-tuple for the same y gets an equal amount of “hits”

If $s_{\text{in}} = s_{\text{out}}$:

$\mathrm{f} : x \rightarrow y$ is an invertible function

$\Updownarrow$

$\mathrm{F} : (x_1, x_2, \ldots, x_s) \rightarrow (y_1, y_2, \ldots, y_s)$ is an invertible function

Multiplier v1.0

$$z_1 = \mathrm{f}_1\left(x_2, x_3, y_2, y_3\right) = x_2y_2 + x_2y_3 + x_3y_2$$

$$z_2 = \mathrm{f}_2\left(x_1, x_3, y_1, y_3\right) = x_3y_3 + x_1y_3 + x_3y_1$$

$$z_3 = \mathrm{f}_3\left(x_1, x_2, y_1, y_2\right) = x_1y_1 + x_1y_2 + x_1y_3$$

x	y	$z_1z_2z_3$							
		000	011	110	101	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

Multiplier v2.0

$$z_1 = (x_3 + x_4)(y_2 + y_3) + x_2 + x_4$$

$$z_2 = (x_1 + x_3)(y_1 + y_4) + y_1 + x_4$$

$$z_3 = (x_2 + x_4)(y_1 + y_4) + x_2$$

$$z_4 = (x_1 + x_2)(y_2 + y_3) + y_1$$

x	y	$z_1z_2z_3z_4$															
		0000	0011	0110	0110	1100	0101	1010	1111	0001	0010	0100	1000	0111	1011	1101	1110
0	0	8	8	8	8	8	8	8	8	0	0	0	0	0	0	0	0
0	1	8	8	8	8	8	8	8	8	0	0	0	0	0	0	0	0
1	0	8	8	8	8	8	8	8	8	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	8	8	8	8	8	8	8	8

In practice: Noekeon

- Block cipher submitted to NESSIE (2000)
- Lightweight
- Ultra compact and fast in hardware

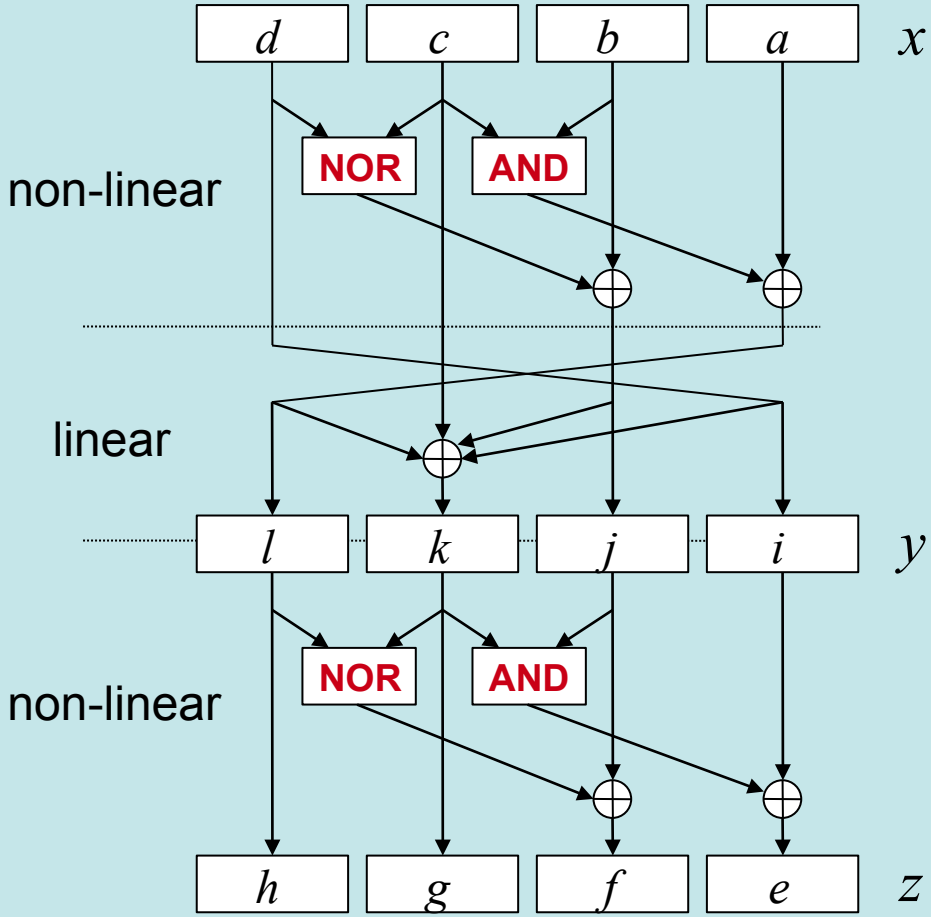
- 4-bit S-box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	7	A	2	C	4	8	F	0	5	9	1	E	3	D	B	6

$$S(x) = NL(L(NL(x))$$

The S-box

Two non-linear parts with *deg=2*:

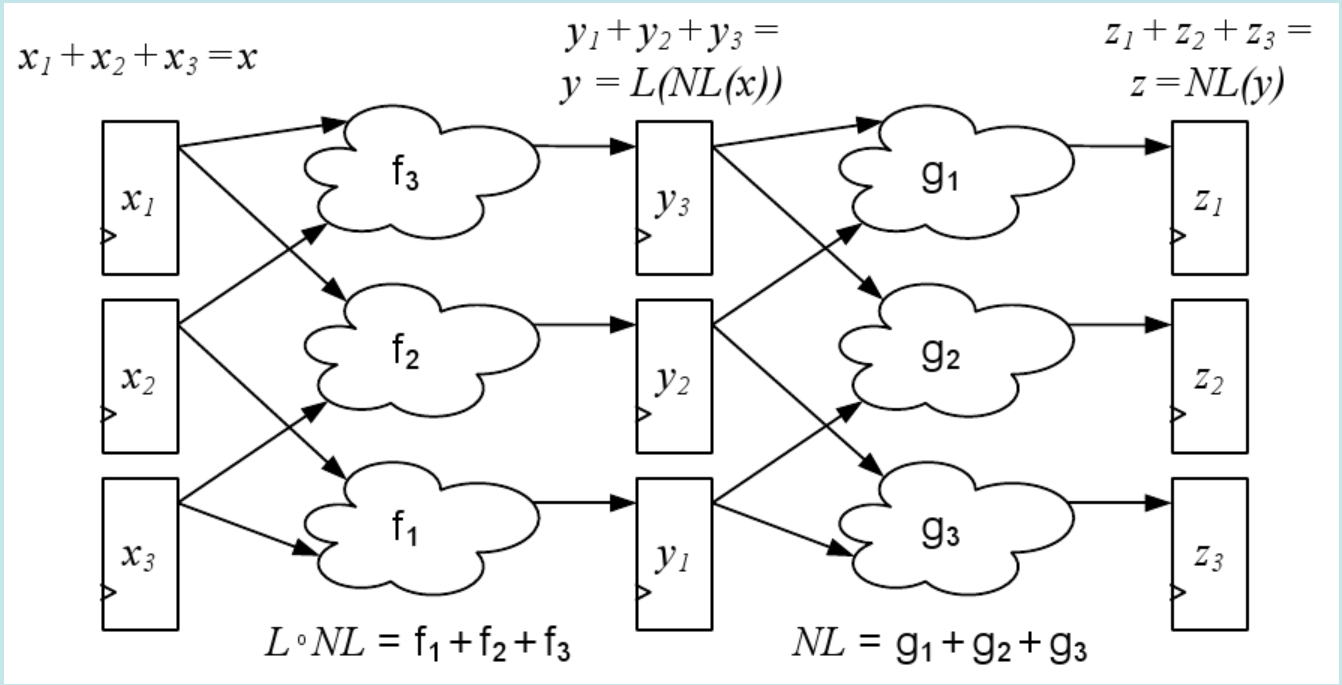


$$i = d$$
$$j = 1 + b + c + d + cd$$
$$k = 1 + a + b + bc + cd$$
$$l = a + bc$$

$$e = i + jk$$
$$f = 1 + j + k + l + kl$$
$$g = k$$
$$h = l$$

Protecting the S-box

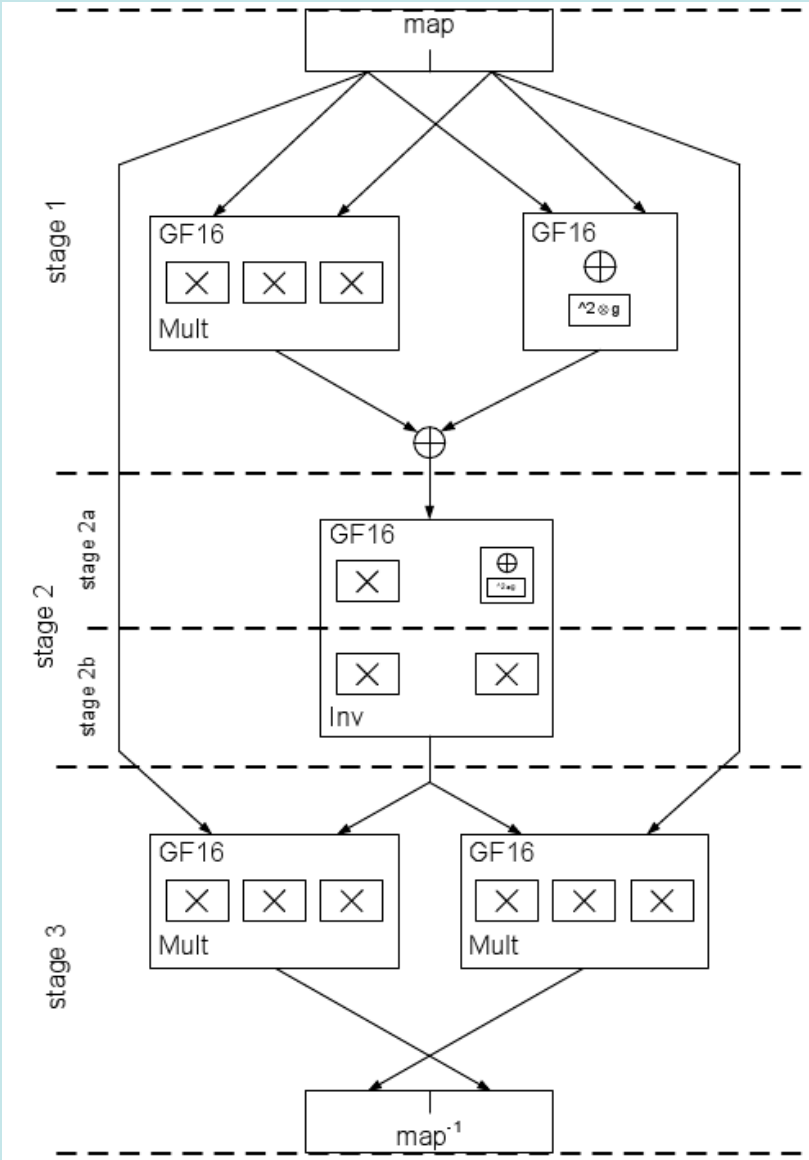
- 3 shares
- 1 intermediate register layer
- Each function:
  - independent of at least one share
  - uniform input and output distribution



AES

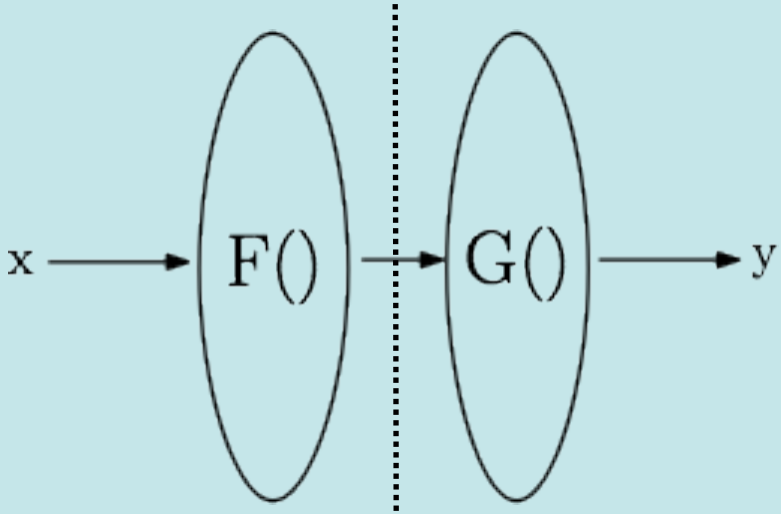
- Nonlinear part = inversion over GF(256)
- Tower field approach

- Large search space
- Ongoing research



Present

- 4-bit Sbox without structure
- Constructing f_i: search space too large
- Decompose into sequence of functions with known sharing circuit [Poschmann+ 2011]



S = G o F

All invertible 4-bit S-boxes

- 302 classes of affine-equivalent S-boxes

remark	unshared	3 shares				4 shares			5 shares
		1	2	3	4	1	2	3	1
affine	1	1				1			1
quadratic	6	5	1			6			6
cubic in A_{16}	30		28	2			30		30
cubic in A_{16}	114			113	1			114	114
cubic in $S_{16} \setminus A_{16}$	151					4	22	125	151

[Bilgin+, 2012]

Secure implementation of Keccak

Keccak nonlinearity:

- 5-bit S-box
- 320 instances per round

Secure implementation:

- 4 shares
- 3 shares + remasking
  - Start: 2 fresh random bits per state bit (3200 bits)
  - Reduced to 4 fresh random bits per S-box (1280 bits)
  - Re-use random bits for next S-box (4 bits)

[Bilgin+ 2013]

PRIMATEs

- Authenticated-Encryption ciphers
  - Submitted to CAESAR competition
- Designed for TI
  - 5-bit S-box
  - Good resistance against linear and differential cryptanalysis
  - Small hardware (threshold) implementation

[Andreeva+ 2014]

Higher-order attacks

Types:

1. Higher-order statistics but single measurement per data

2. Higher order statistics and multiple measurements per data, measured simultaneously

3. Higher order statistics and multiple measurements per data, possibly with delay between the measurements

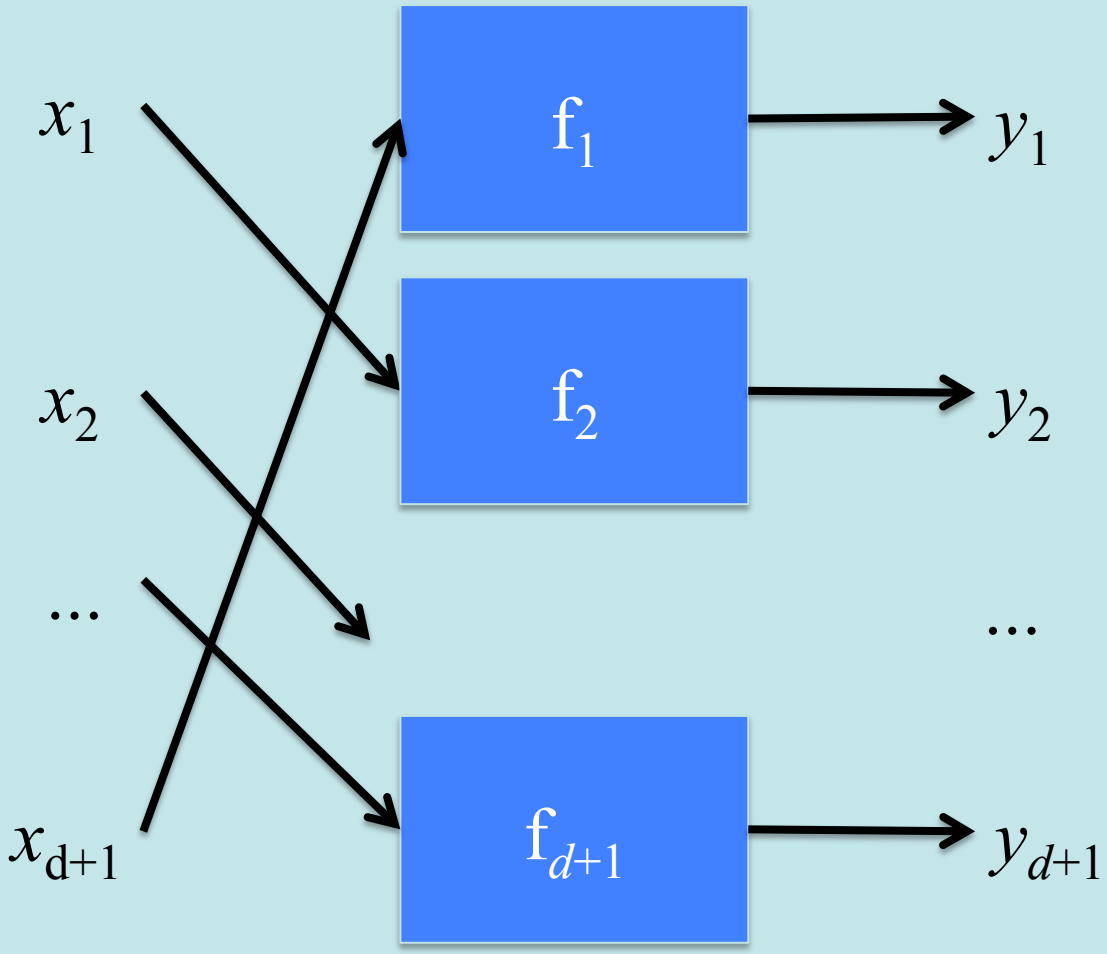
d^{th}-order non-completeness

- All combinations of up to d functions f_i must not depend on at least one x_j
- Protects against d^{th}-order attacks of Type 1 and Type 2

[Bilgin+ 2014]

Linear functions

Use d+1 shares



Functions of degree r

- There always exists a circuit with
  - $s_{\text{in}} = rd + 1$ input shares
  - $s_{\text{out}} = \text{Comb}(s_{\text{in}}, r)$ output shares (and functions f_i)
- Other $(s_{\text{in}}, s_{\text{out}})$-combinations exist
- Extra reduction step to decrease number of shares from s_{out} to s_{in}

Further work

- Security against fault attacks
  - Induce hardware failure while measuring signals
  - Techniques from robust multiparty computation (?)
- Security against attacks using non-linear combination of signals measured at different times
  - Alternatives to remasking
- Incorporate assumptions about the power of the adversary

Thanks

- Begül Bilgin
- Joan Daemen
- Thomas De Cnudde
- Benedikt Gierlichs
- Venzislav Nikov
- Svetla Nikova
- Christian Rechberger
- Martin Schläffer
- Georg Stütz
- Natalia Tokareva
- Gilles Van Assche
- Valeriya Vitkup