DPA-Resistance without routing constraints?

A cautionary note about MDPL security

Benedikt Gierlichs

ESAT / SCD - COSIC
Katholieke Universiteit Leuven
Belgium

CHES 2007, Vienna

Outline

1. Introduction
2. Attack strategy
3. Experimental results
4. Conclusion

- Power analysis threatens black-box secure cryptography implemented in embedded devices (due to CMOS)
- Countermeasures at software level:
  - randomize the link between leakage and processed data (masking)
  - hide the leakage in the time domain (random order execution, random process interrupts)
- Countermeasures at circuit level:
  - hide the leakage in the time domain (sliding clock, asynchronous designs)
  - reduce the SNR (noise generators, current scramblers)
- Countermeasures at gate level:
  - randomize the link between leakage and processed data (masking)
  - reduce the leakage (SABL, WDDL, CML, etc.)

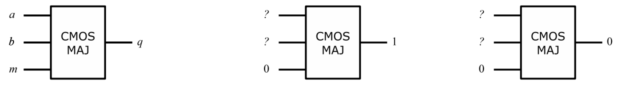
- Power analysis threatens black-box secure cryptography implemented in embedded devices (due to CMOS)
- Countermeasures at software level:
  - randomize the link between leakage and processed data (masking)
  - hide the leakage in the time domain (random order execution, random process interrupts)
- Countermeasures at circuit level:
  - hide the leakage in the time domain (sliding clock, asynchronous designs)
  - reduce the SNR (noise generators, current scramblers)
- Countermeasures at gate level:
  - randomize the link between leakage and processed data (masking)
  - reduce the leakage (SABL, WDDL, CML, etc.)

- Power analysis threatens black-box secure cryptography implemented in embedded devices (due to CMOS)
- Countermeasures at software level:
  - randomize the link between leakage and processed data (masking)
  - hide the leakage in the time domain (random order execution, random process interrupts)
- Countermeasures at circuit level:
  - hide the leakage in the time domain (sliding clock, asynchronous designs)
  - reduce the SNR (noise generators, current scramblers)
- Countermeasures at gate level:
  - randomize the link between leakage and processed data (masking)
  - reduce the leakage (SABL, WDDL, CML, etc.)

- Power analysis threatens black-box secure cryptography implemented in embedded devices (due to CMOS)
- Countermeasures at software level:
  - randomize the link between leakage and processed data (masking)
  - hide the leakage in the time domain (random order execution, random process interrupts)
- Countermeasures at circuit level:
  - hide the leakage in the time domain (sliding clock, asynchronous designs)
  - reduce the SNR (noise generators, current scramblers)
- Countermeasures at gate level:
  - randomize the link between leakage and processed data (masking)
  - reduce the leakage (SABL, WDDL, CML, etc.)

- MDPL is a masked logic style and builds up on standard CMOS cells (however, eventually it adopts masking and DRP)

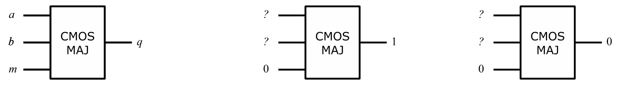
standard



$q = \text{maj}(a, b, m)$ $q = 1$ $q = 0$

- MDPL is a masked logic style and builds up on standard CMOS cells (however, eventually it adopts masking and DRP)

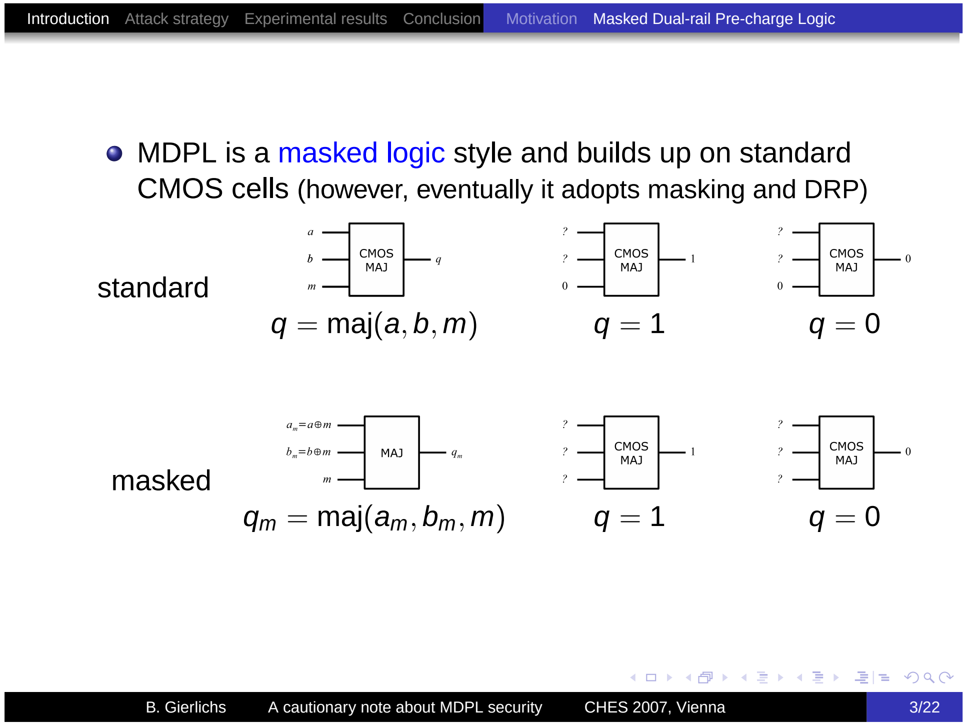
standard



$q = \text{maj}(a, b, m)$ $q = 1$ $q = 0$

$\Downarrow$ $\Downarrow$

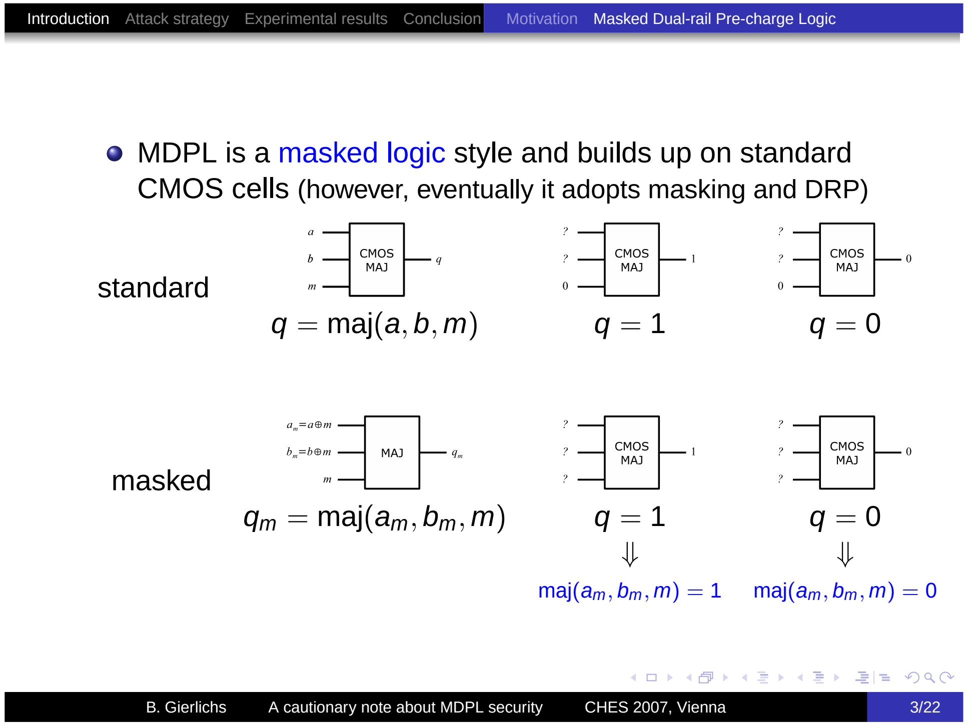
$a = b = 1$ $a = b = 0$ or $a \neq b$



- MDPL is a masked logic style and builds up on standard CMOS cells (however, eventually it adopts masking and DRP)

$q = \text{maj}(a, b, m)$ $q = 1$ $q = 0$

$q_m = \text{maj}(a_m, b_m, m)$ $q = 1$ $q = 0$



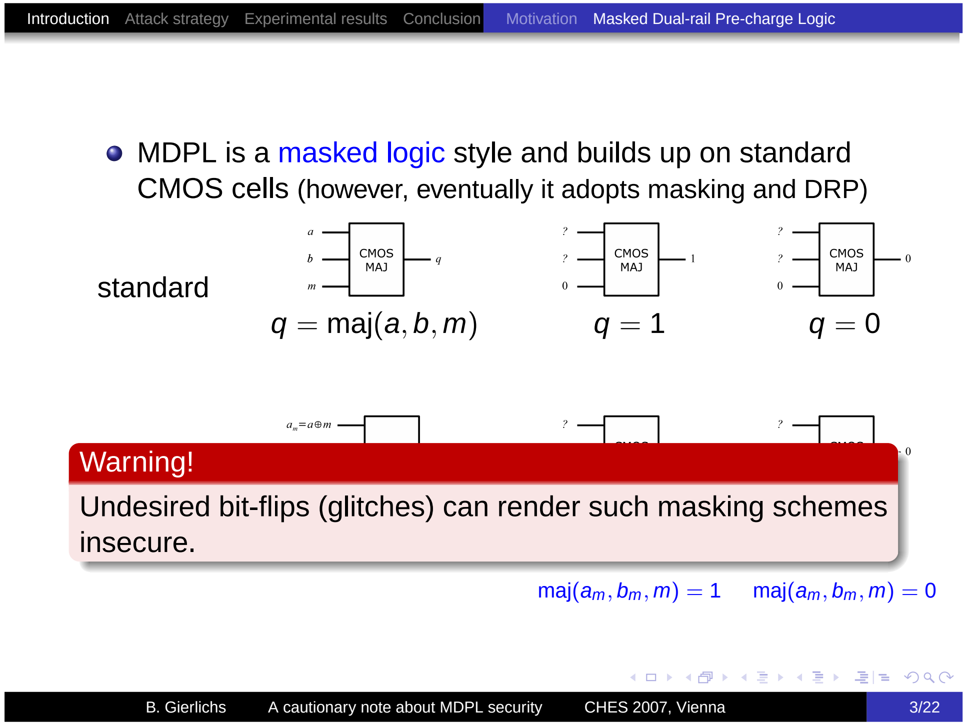
- MDPL is a masked logic style and builds up on standard CMOS cells (however, eventually it adopts masking and DRP)

$q = \text{maj}(a, b, m)$ $q = 1$ $q = 0$

$q_m = \text{maj}(a_m, b_m, m)$ $q = 1$ $q = 0$

$\Downarrow$ $\Downarrow$

$\text{maj}(a_m, b_m, m) = 1$ $\text{maj}(a_m, b_m, m) = 0$



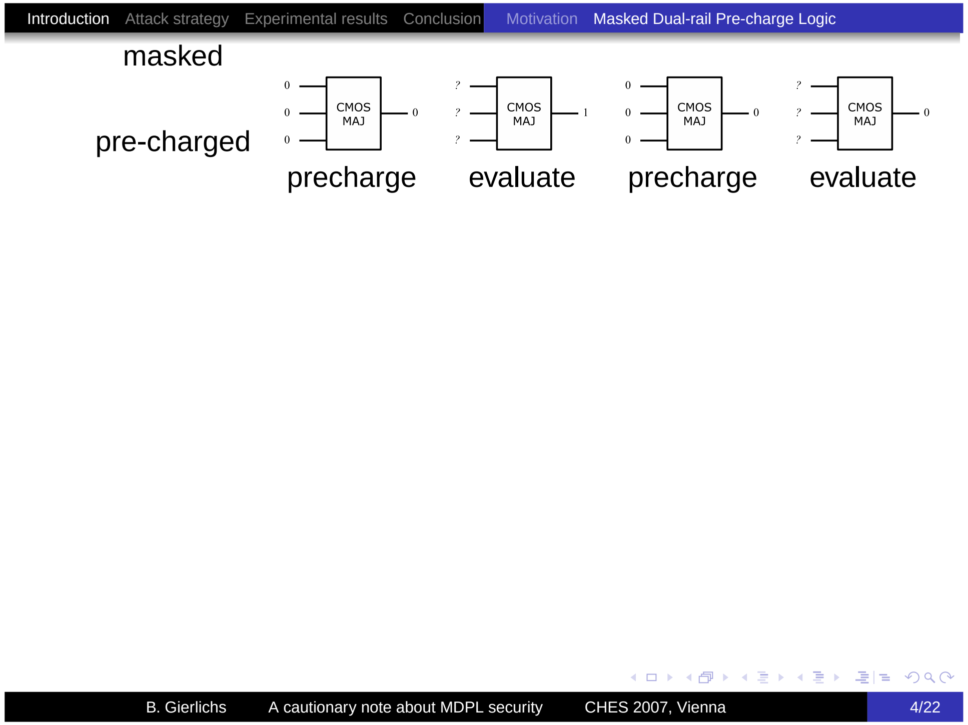
- MDPL is a masked logic style and builds up on standard CMOS cells (however, eventually it adopts masking and DRP)

$q = \text{maj}(a, b, m)$ $q = 1$ $q = 0$

Warning!

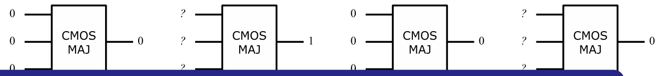
Undesired bit-flips (glitches) can render such masking schemes insecure.

$\text{maj}(a_m, b_m, m) = 1$ $\text{maj}(a_m, b_m, m) = 0$



precharge evaluate precharge evaluate

masked



pre-charged

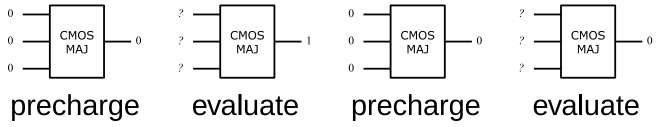
OK, no glitches...

But what about the large mask signal tree?

- the mask signal tree is also precharged to 0
- it should be feasible to distinguish 0 → 0 and 0 → 1
- and hence to recover the mask's value for every clock cycle

masked

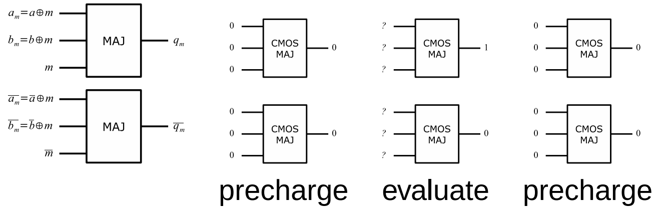
pre-charged



masked

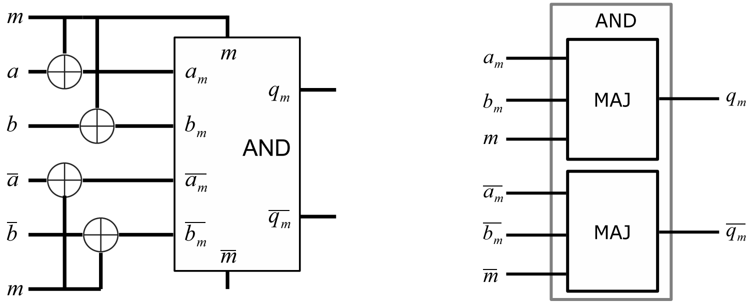
precharged

and dual-rail

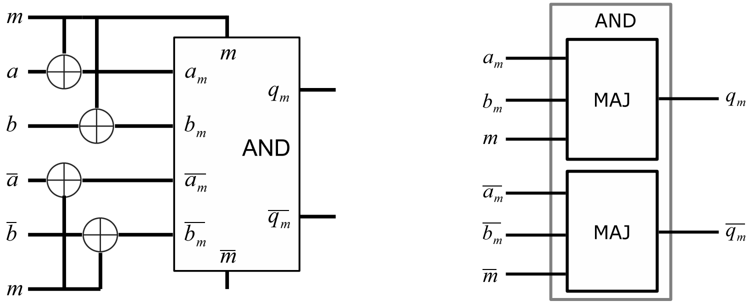


$$q_m = \text{maj}(a_m, b_m, m)$$

$$\overline{q}_m = \text{maj}(\overline{a}_m, \overline{b}_m, \overline{m})$$



- we develop a probabilistic model of switching activity
- **A**, **B** are uniform on $\mathcal{S} := \{0, 1\}$
- **M**'s distribution on $\mathcal{S}$ depends on bias α in the PRNG
- **T** is output transition on wire q_m, $\mathcal{T} := \{0 \rightarrow 0, 0 \rightarrow 1\}$
- $E(\mathbf{T} = t)$ is observable output transition energy
- $E(\mathbf{T} = 0 \rightarrow 1) = \delta$, $E(\overline{\mathbf{T}} = 0 \rightarrow 1) = \gamma$
- δ and γ are gate specific ⇒ attack a single AND gate



- we develop a probabilistic model of switching activity
- **A**, **B** are uniform on $\mathcal{S} := \{0, 1\}$
- **M**'s distribution on $\mathcal{S}$ depends on bias α in the PRNG
- **T** is output transition on wire q_m, $\mathcal{T} := \{0 \rightarrow 0, 0 \rightarrow 1\}$
- $E(\mathbf{T} = t)$ is observable output transition energy
- $E(\mathbf{T} = 0 \rightarrow 1) = \delta$, $E(\overline{\mathbf{T}} = 0 \rightarrow 1) = \gamma$
- δ and γ are gate specific ⇒ attack a single AND gate

$\mathbf{A}_m$	$\mathbf{B}_m$	**M**	bias	**A**	**B**	**T**	$\overline{\mathbf{T}}$	E
0	0	0	α	0	0	$0 \to 0$	$0 \to 1$	γ
0	0	1	$1-\alpha$	1	1	$0 \to 0$	$0 \to 1$	γ
0	1	0	α	0	1	$0 \to 0$	$0 \to 1$	γ
0	1	1	$1-\alpha$	1	0	$0 \to 1$	$0 \to 0$	δ
1	0	0	α	1	0	$0 \to 0$	$0 \to 1$	γ
1	0	1	$1-\alpha$	0	1	$0 \to 1$	$0 \to 0$	δ
1	1	0	α	1	1	$0 \to 1$	$0 \to 0$	δ
1	1	1	$1-\alpha$	0	0	$0 \to 1$	$0 \to 0$	δ

$$\mathbb{P}_{\mathbf{T}} = \{0 \to 1 : 0.75 - 0.5\alpha, \quad 0 \to 0 : 0.25 + 0.5\alpha\}$$

$\mathbf{A}_m$	$\mathbf{B}_m$	**M**	bias	**A**	**B**	**T**	$\overline{\mathbf{T}}$	E
0	0	0	α	0	0	$0 \to 0$	$0 \to 1$	γ
0	0	1	$1-\alpha$	1	1	$0 \to 0$	$0 \to 1$	γ
0	1	0	α	0	1	$0 \to 0$	$0 \to 1$	γ
0	1	1	$1-\alpha$	1	0	$0 \to 1$	$0 \to 0$	δ
1	0	0	α	1	0	$0 \to 0$	$0 \to 1$	γ
1	0	1	$1-\alpha$	0	1	$0 \to 1$	$0 \to 0$	δ
1	1	0	α	1	1	$0 \to 1$	$0 \to 0$	δ
1	1	1	$1-\alpha$	0	0	$0 \to 1$	$0 \to 0$	δ

Suppose bits a and b are key-dependent intermediate results; based on a key guess, filter $a \neq b$

$\Theta = E(\mathbf{T}|a = b = 0) - E(\mathbf{T}|a = b = 1)$

correct guess: $\Theta = 2\alpha\gamma - 2\alpha\delta + \delta - \gamma$

$\mathbf{A}_m$	$\mathbf{B}_m$	**M**	bias	**A**	**B**	**T**	$\overline{\mathbf{T}}$	E
0	0	0	α	0	0	$0 \to 0$	$0 \to 1$	γ
0	0	1	$1-\alpha$	1	1	$0 \to 0$	$0 \to 1$	γ
0	1	0	α	0	1	$0 \to 0$	$0 \to 1$	γ
0	1	1	$1-\alpha$	1	0	$0 \to 1$	$0 \to 0$	δ
1	0	0	α	1	0	$0 \to 0$	$0 \to 1$	γ
1	0	1	$1-\alpha$	0	1	$0 \to 1$	$0 \to 0$	δ
1	1	0	α	1	1	$0 \to 1$	$0 \to 0$	δ
1	1	1	$1-\alpha$	0	0	$0 \to 1$	$0 \to 0$	δ

Suppose bits a and b are key-dependent intermediate results; based on a key guess, filter $a \neq b$

$\Theta = E(\mathbf{T}|a = b = 0) - E(\mathbf{T}|a = b = 1)$

correct guess: $\Theta = 2\alpha\gamma - 2\alpha\delta + \delta - \gamma$

$\mathbf{A}_m$	$\mathbf{B}_m$	**M**	bias	**A**	**B**	**T**	$\overline{\mathbf{T}}$	E
0	0	0	α	0	0	$0 \to 0$	$0 \to 1$	γ
0	0	1	$1-\alpha$	1	1	$0 \to 0$	$0 \to 1$	γ
0	1	0	α	0	1	$0 \to 0$	$0 \to 1$	γ
0	1	1	$1-\alpha$	1	0	$0 \to 1$	$0 \to 0$	δ
1	0	0	α	1	0	$0 \to 0$	$0 \to 1$	γ
1	0	1	$1-\alpha$	0	1	$0 \to 1$	$0 \to 0$	δ
1	1	0	α	1	1	$0 \to 1$	$0 \to 0$	δ
1	1	1	$1-\alpha$	0	0	$0 \to 1$	$0 \to 0$	δ

Suppose bits a and b are key-dependent intermediate results; based on a key guess, filter $a \neq b$

$\Theta = E(\mathbf{T}|a = 0, b = 0) - E(\mathbf{T}|a = 1, b = 1)$

correct guess: $\Theta = 2\alpha\gamma - 2\alpha\delta + \delta - \gamma$

1 bit wrong: $\Theta = 0$

$\mathbf{A}_m$	$\mathbf{B}_m$	**M**	bias	**A**	**B**	**T**	$\overline{\mathbf{T}}$	E
0	0	0	α	0	0	$0 \to 0$	$0 \to 1$	γ
0	0	1	$1-\alpha$	1	1	$0 \to 0$	$0 \to 1$	γ
0	1	0	α	0	1	$0 \to 0$	$0 \to 1$	γ
0	1	1	$1-\alpha$	1	0	$0 \to 1$	$0 \to 0$	δ
1	0	0	α	1	0	$0 \to 0$	$0 \to 1$	γ
1	0	1	$1-\alpha$	0	1	$0 \to 1$	$0 \to 0$	δ
1	1	0	α	1	1	$0 \to 1$	$0 \to 0$	δ
1	1	1	$1-\alpha$	0	0	$0 \to 1$	$0 \to 0$	δ

Suppose bits *a* and *b* are key-dependent intermediate results; based on a key guess, filter $a \neq b$

$\Theta = E(\mathbf{T}|a=0, b=0) - E(\mathbf{T}|a=1, b=1)$

correct guess: $\Theta = 2\alpha\gamma - 2\alpha\delta + \delta - \gamma$

1 bit wrong: $\Theta = 0$

both bits wrong: $\Theta = 2\alpha\delta - 2\alpha\gamma + \gamma - \delta$

- DPA peak correct guess: $\Theta = 2\alpha\gamma - 2\alpha\delta + \delta - \gamma$
- DPA peak both bits wrong: $\Theta = 2\alpha\delta - 2\alpha\gamma + \gamma - \delta$
- DPA peak 1 bit wrong: $\Theta = 0$

- for any bias $\alpha \neq 0.5$ and $\delta \neq \gamma$ (no differential routing)
  - three different values for Θ possible
  - a guess that is wrong in 1 bit is distinguishable without further knowledge about α, δ, γ
  - this property can be exploited to sieve key candidates!

- DPA peak correct guess: $\Theta = 2\alpha\gamma - 2\alpha\delta + \delta - \gamma$
- DPA peak both bits wrong: $\Theta = 2\alpha\delta - 2\alpha\gamma + \gamma - \delta$
- DPA peak 1 bit wrong: $\Theta = 0$

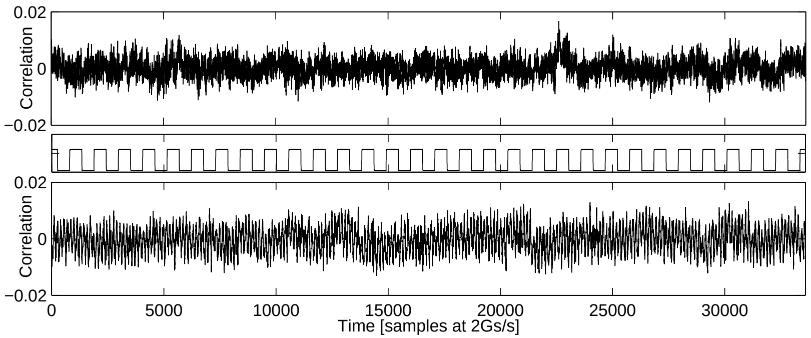
- for any bias $\alpha \neq 0.5$ and $\delta \neq \gamma$ (no differential routing)
  - three different values for Θ possible
  - a guess that is wrong in 1 bit is distinguishable without further knowledge about α, δ, γ
  - this property can be exploited to sieve key candidates!

- Side Channel Analysis Resistant Design flow - SCARD
- 8051 μC + AES-128 co-processor in CMOS and several secured logic styles, incl. MDPL
- Masks for MDPL generated on chip by controllable PRNG
- Measurements represent current drain in dedicated core VDD

- 2 sets of measurements:
  - 100 000 traces, 2GS/s, PRNG bias $\alpha = 1$ ($m = 0$)
  - 200 000 traces, 2GS/s, PRNG bias $\alpha =$ unknown

- Attack: correlation
- Target: simultaneous transition of four 8 bit registers
- prediction: $H_i = \mathrm{HW}(R_i \oplus D_i)$, flip-flops not precharged
- Attacking 8 key bytes in parallel is not practical, however we want to
  - show that MDPL with disabled masking is vulnerable to a “standard” attack
  - show that MDPL with enabled masking resists the same “standard” attack
  - ⇒ verify that the PRNG has been setup and started correctly

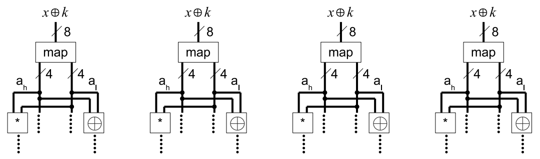
Masking disabled ($\alpha = 1$)



Masking enabled ($\alpha = ?$)

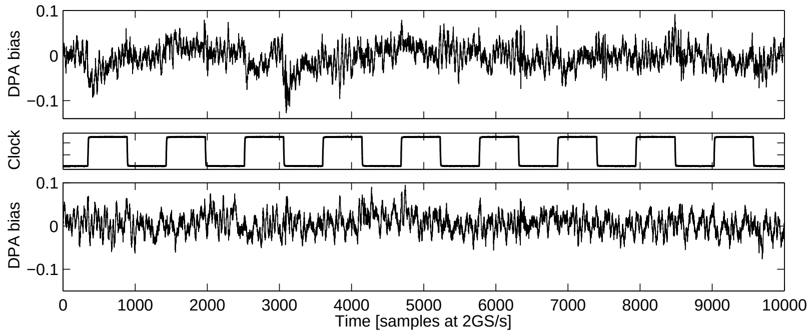
(Correlation attack, correct key, 32 bit intermediate result)

- Is the output transition energy difference ⊖ measurable?
- AES Sbox implemented in combinational logic using composite field representation

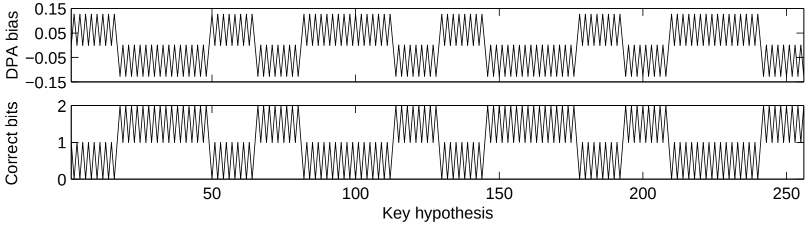


- One 4 × 4-bit multiplier comprises 16 AND gates
- DPA against a single AND gate in a parallelized, pipelined, and MDPL protected VLSI circuit

Correct guess on **A** and **B**



Wrong guess in *either* **A** *or* **B**



- All key guesses which lead to a guess that is incorrect in 1 bit can be rejected without knowledge of α, δ, γ
- We verify that leakage occurs and can be exploited
- Attack next AND gate for further sieving...

So what is the bias α?

- We don't know...
- We simulated a gate-level netlist of the PRNG
- Statistical analysis of 1 million output bits shows $\alpha = 0.5001$
- This is not a bias, let's call α a deviation

- Open questions:
  - Does the chip conform to the simulation?
  - Is such α enough to enable our attack strategy? $\rightarrow$ distinguish $2\alpha\gamma - 2\alpha\delta + \delta - \gamma$ from 0
  - If not: is it possible that our attack unintentionally exploited circuit anomalies?
- Thorough investigation in the near future...

So what is the bias α?

- We don't know...
- We simulated a gate-level netlist of the PRNG
- Statistical analysis of 1 million output bits shows $\alpha = 0.5001$
- This is not a bias, let's call α a deviation

- Open questions:
  - Does the chip conform to the simulation?
  - Is such α enough to enable our attack strategy? $\rightarrow$ distinguish $2\alpha\gamma - 2\alpha\delta + \delta - \gamma$ from 0
  - If not: is it possible that our attack unintentionally exploited circuit anomalies?
- Thorough investigation in the near future...

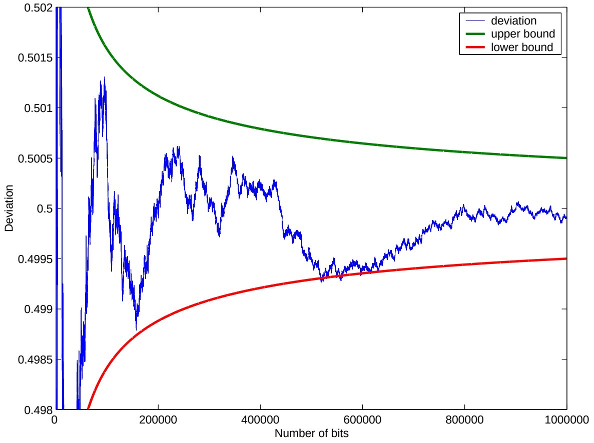
So what is the bias α?

- We don't know...
- We simulated a gate-level netlist of the PRNG
- Statistical analysis of 1 million output bits shows $\alpha = 0.5001$
- This is not a bias, let's call α a deviation

- Open questions:
  - Does the chip conform to the simulation?
  - Is such α enough to enable our attack strategy? $\rightarrow$ distinguish $2\alpha\gamma - 2\alpha\delta + \delta - \gamma$ from 0
  - If not: is it possible that our attack unintentionally exploited circuit anomalies?
- Thorough investigation in the near future...

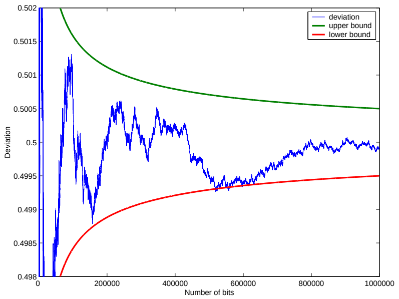
Update

- Statistical analysis of up to 1 million output bits shows



α is $\frac{\sum_n bits}{n}$ –,– is $0.5 \pm 0.5 * \frac{\sqrt{n}}{n}$

Update



- we pick each nth bit → new distribution, same properties
- for 200k samples we have $0.4989 \leq \alpha \leq 0.5011$

→ α gains $\sim$ one order of magnitude

- number of curves is crucial for a successful attack
- attack might not work even for a “good” α, choose a different gate with a better δ - γ contrast

Conclusion

- Probabilistic model for the output transition energy of non-linear MDPL gates
- Depends on the bias α in the source of the randomness
- Output transition energy difference $\ominus$ can be exploited
- Requirements for our attack methodology:
  - slight and realistic PRNG “deviations”
  - unbalanced differential routing
  - knowledge about the circuit layout
- Theoretic approach is verified by experimental results based on a prototype chip

Thank you for your attention.

Questions?

Bibliography I

- T. Popp, S. Mangard: Masked Dual-Rail Pre-charge Logic. In: J. R. Rao and B. Sunar (eds.): Cryptographic Hardware and Embedded Systems – CHES 2005, LNCS 3659, pp. 172–186, Springer, 2005
- Benedikt Gierlichs: DPA-Resistance without routing constraints? In: P. Paillier and I. Verbauwhede (eds.): Cryptographic Hardware and Embedded Systems – CHES 2007, LNCS 4727, pp. 107–120, Springer, 2007