Power analysis of implementations protected with secret sharing
Application to KECCAK

Joan Daemen\textsuperscript{1}, Michaël Peeters\textsuperscript{2}, Gilles Van Assche\textsuperscript{1}

Joint work with
Guido Bertoni\textsuperscript{1}, Nicolas Debande\textsuperscript{3} and Thanh-Ha Le\textsuperscript{3}

\textsuperscript{1}STMicroelectronics \textsuperscript{2}NXP Semiconductors \textsuperscript{3}Morpho

ESC 2013, Mondorf-Les-Bains, January 14-18
Outline

1. Introduction
2. Exploiting power consumption
3. Attacking unprotected KECCAK
4. Generalization to quadratic functions
5. Parasitic correlations
6. Attacking protected KECCAK
Outline

1 Introduction
2 Exploiting power consumption
3 Attacking unprotected KECCAK
4 Generalization to quadratic functions
5 Parasitic correlations
6 Attacking protected KECCAK
Keyed mode: part of input is secret key
Security relies on secrecy of inner state
Attack it with side channel attacks
KECCAK structure: sponge and duplex

- Keyed mode: part of input is secret key
- Security relies on secrecy of inner state
- Attack it with side channel attacks
**KECCAK-\(f\): the permutations in KECCAK**

- Operates on 3D state:
  - (5 × 5)-bit slices
  - \(2^\ell\)-bit lanes
  - param. \(0 \leq \ell < 7\)

- Round function \(R\) with 5 steps:
  - \(\theta\): mixing layer
  - \(\rho\): bit transposition
  - \(\pi\): bit transposition
  - \(\chi\): non-linear layer
  - \(\iota\): round constants

- \# rounds: \(12 + 2\ell\) for \(b = 2^\ell 25\)
  - 12 rounds in KECCAK-\(f[25]\)
  - 24 rounds in KECCAK-\(f[1600]\)
Straightforward hardware architecture
Side-channel attacks

- Exploit information leakage in implementations
- Timing attacks
  - e.g., cache-miss attacks
- Power analysis
  - Simple (SPA): few measurement suffice
  - Differential (DPA): multiple measurements and statistical methods
- Electromagnetic analysis
  - Similar to power analysis
  - Simple (SEMA) or differential (DEMA)
Current in a CMOS NAND gate

- **Static current**
  - either top (PMOS) transistors or bottom (NMOS) transistors have high resistance
  - very small
  - leakage increases as dimensions decrease
  - weak data dependency

- **Dynamic current: when switching**
  - temporary short-circuit
  - (dis)charging of load capacitances
  - short spike
  - strong data dependency
Current in a CMOS NAND gate

- **Static current**
  - either top (PMOS) transistors or bottom (NMOS) transistors have high resistance
  - very small
  - leakage increases as dimensions decrease
  - weak data dependency

- **Dynamic current: when switching**
  - temporary short-circuit
  - (dis)charging of load capacitances
  - short spike
  - strong data dependency
Current in a CMOS NAND gate

- **Static current**
  - either top (PMOS) transistors or bottom (NMOS) transistors have high resistance
  - very small
  - leakage increases as dimensions decrease
  - weak data dependency

- **Dynamic current:** when switching
  - temporary short-circuit
  - (dis)charging of load capacitances
  - short spike
  - strong data dependency
More complex combinatorial circuits

- Examples:
  - XOR
  - Two-to-one multiplexer

- More complex switching behaviour
- Multiple gates switched in series
  - transients due to differences in delays: glitches
  - complex dependency of input bits
  - depends on specific layout
Power analysis of implementations protected with secret sharing Application to Keccak

Introduction

Storage elements: registers

- Flip-flop:
  - clocked: can only change value on rising edge of clock
  - current spike when stored bit changes value
Summary

- Combinatorial circuits:
  - current consumption may exhibit multiple peaks
  - depending on the values of multiple inputs
  - due to propagation: glitches

- Registers:
  - current peak when register changes value
  - $0 \Rightarrow 1$ may have different consumption as $1 \Rightarrow 0$

- Total power consumption: sum of all gates and registers

- Electromagnetic analysis:
  - probing with tiny antenna
  - current close to antenna: high contribution
  - current far from antenna: low contribution
Outline

1. Introduction
2. Exploiting power consumption
3. Attacking unprotected KECCAK
4. Generalization to quadratic functions
5. Parasitic correlations
6. Attacking protected KECCAK
A model of the power consumption

Consumption at any time instance can be modeled as

\[ P = \sum_i T_i[d_i] \]

- \( d_i \): Boolean variables that express activity
  - bit 1 in a given register or gate output at some stage
  - flipping of a specific register or gate output at some stage
- \( T_i[0] \) and \( T_i[1] \): stochastic variables
  - \( T_i[0] \) and \( T_i[1] \) may have different distributions
  - side channel attacks exploit this difference
  - In principle \( T_i[0] \) and \( T_i[1] \) are functions of time
Our simplified model of the power consumption

Only considering differences in mean of $T_i[0]$ and $T_i[1]$ and their variance:

$$P = \alpha + \sum_i v_i (-1)^{d_i}$$

- $\alpha$: normally distributed term independent from activity
  - we will assume it has mean 0
  - can be achieved with normalization

- $d_i$: Boolean variables that express activity

- $v_i$: real weighing factors
  - power consumption: all $v_i$ have similar values
  - electromagnetic: $v_i$ depends on distance to antenna
  - $v_i$ are functions of time
The simplest flavour of DPA

- Goal: recover secret $K$
- Collect traces: power/EM measurement of cipher execution
- Focus on activity $d_i$ that depends on $K$ and known input or output $M$
- Hypothesis $K^*$ on $K$ predicts $d_i$ per trace:
  - by a function $s(K^*, M)$: the selection function
  - Partition traces in two sets $M_0$ and $M_1$ based on $s(K^*, M)$
- Correct hypothesis $d_i = s(K, M)$
  - $d_i = 1$ for all traces in $M_1$
  - $d_i = 0$ for all traces in $M_0$
- Wrong hypothesis: $d_i$ at best uncorrelated with $s(K^*, M)$
The simplest flavour of DPA (cont’d)

- Criterion for hypothesis $K^*$: Difference of Mean (DoM)
  - compute average traces over $M_0$ and $M_1$ respectively
  - take their difference
  - choose $K^*$ that has the highest peak

- Success probability depends on
  - signal: effect of $d_i$ on power consumption
  - noise: all consumption independent of $d_i$
  - number of hypotheses
  - decorrelation of selection function if incorrect hypothesis
Outline

1. Introduction
2. Exploiting power consumption
3. Attacking unprotected Keccak
4. Generalization to quadratic functions
5. Parasitic correlations
6. Attacking protected Keccak
The KECCAK-\(f\) round function

\[ R = \iota \circ \chi \circ \pi \circ \rho \circ \theta \]

- Linear part \( \lambda \) followed by non-linear part \( \chi \)
- \( \lambda = \pi \circ \rho \circ \theta \): mixing followed by bit transposition
- \( \chi \): simple mapping operating on rows:
DPA applied to the simple KECCAK core

- Leakage exploited: switching consumption of register bit 0
- Value switches from $a_0$ to $b_0 + (b_1 + 1)b_2$
- Activity equation: $d = a_0 + b_0 + (b_1 + 1)b_2$
DPA applied to the simple KECCAK core

- Take the case $M = 0$
- We call $K$ the input of $\chi$-block if $M = 0$
- $K$ will be our target
DPA applied to the simple KECCAK core

- We call the effect of $M$ at input of $\chi$: $\mu$
- $\mu = \lambda(M||0^c)$
- Linearity of $\lambda$: $B = K + \lambda(M||0^c)$
DPA applied to the simple KECCAK core

- \( d = a_0 + k_0 + (k_1 + 1)k_2 \) + \( \mu_0 + (\mu_1 + 1)\mu_2 + k_1\mu_2 + k_2\mu_1 \)
- Fact: value of \( q = a_0 + k_0 + (k_1 + 1)k_2 \) is same for all traces
- Let \( M_0 \): traces with \( d = q \) and \( M_1 \): \( d = q + 1 \)
DPA applied to the simple KECCAK core

- Selection: \( s(M, K^*) = \mu_0 + (\mu_1 + 1)\mu_2 + k_1^*\mu_2 + k_2^*\mu_1 \)
- Values of \( \mu_1 \) and \( \mu_2 \) computed from \( M \)
- Hypothesis has two bits only: \( k_1^* \) and \( k_2^* \)
DPA applied to the simple KECCAK core (cont’d)

- Correct hypothesis $K$
  - traces in $M_0$: $d = q$
  - traces in $M_1$: $d = q + 1$

- Incorrect hypothesis $K^* = K + \Delta$
  - trace in $M_0$: $d = q + \mu_1\delta_2 + \mu_2\delta_1$
  - trace in $M_1$: $d = q + \mu_1\delta_2 + \mu_2\delta_1 + 1$

- Remember: $\mu = \lambda(M\rvert|0^c)$
  - random inputs $M$ lead to random $\mu_1$ and $\mu_2$
  - Incorrect hypothesis: $d$ uncorrelated with $\{M_0, M_1\}$
The distribution of the power consumption

\[ P = \alpha + \sum_{i} v_i (-1)^{d_i(K,M)} \]

- Consider the distribution of \( P(M) \) when \( s_i(M,K^*) = s \)
- Assume \( d_j \) with \( j \neq i \) is uncorrelated to \( s_i(M,K^*) \)
- \( P \) is sum of many stochastic variables: normal distribution
- Correct hypothesis
  - variance: \( \sigma^2[\alpha] + \sum_{j \neq i} v_j^2 \)
  - mean: \( v_i (-1)^{q \oplus s} \)
- Incorrect hypothesis
  - variance: \( \sigma^2[\alpha] + \sum_j v_j^2 \)
  - mean: 0
- Normal distributions with different mean, similar variance
Kullback-Leibler divergence

Kullback-Leibler divergence of a distribution $f$ with respect to $g$:

$$D(f\|g) = \int f(t)(\log(f(t)) - \log(g(t)))dt.$$ 

The number of samples of $f$, required to distinguish it from $g$ is inversely proportional to $D(f\|g)$

$$D(\text{correct}\|\text{incorrect}) \approx \frac{v_i^2}{2\sigma^2} = \frac{v_i^2}{2\left(\sigma^2[\alpha] + \sum_j v_j^2\right)}$$
The distribution of DoM

- Variance: \[ \frac{1}{\#M} \left( \sigma^2[\alpha] + \sum_{j \neq i} v_j^2 \right) \]
- Mean for incorrect hypothesis: 0, for correct: \( v_i(-1)^q \)
Predicting the success probability

\( G_h(\sigma^2) \) expresses the success probability assuming \( v_i = 1 \) for \( h \) hypotheses:

\[
G_h(\sigma^2) = \int_{0}^{\infty} \left( \text{erf} \left( \frac{t}{\sqrt{2}\sigma} \right) \right)^{h-1} \left( \mathcal{N}_{(-1;\sigma^2)}(t) + \mathcal{N}_{(1;\sigma^2)}(t) \right) dt.
\]

In our case, \( h = 4 \) and \( \Pr(\text{success}) = G_4 \left( \frac{\sigma^2[\alpha] + \sum_j v_j^2}{\#Mv_i^2} \right) = G_4 \left( \frac{1}{2D(\text{correct}||\text{incorrect})\#M} \right). \)
Result of experiments

- Simulation for all widths of KECCAK-f
- $\sigma^2[\alpha] = 0$ and $v_i = 1$
- Results perfectly match predictions

![Graph showing probability of success and number of traces for various b values.](image-url)
Result of experiments

DoM for various $b$
DoM (theory) for $b=1600$
Outline

1. Introduction
2. Exploiting power consumption
3. Attacking unprotected KECCAK
4. Generalization to quadratic functions
5. Parasitic correlations
6. Attacking protected KECCAK
Generalization to quadratic combinatorial logic

We can express $d_i$ a function of second degree

$$d_i = (K + AM)^T Z_i (K + AM) + c_i$$

with

- $Z_i$: square matrix $|K|
- A$: matrix with $|K|$ rows and $|M|$ columns
Quadratic combinatorial logic: selection function

- Generic expression of second degree:
  \[ d_i = (K + AM)^T Z_i (K + AM) + c_i \]

- Selection function for \( d_i \) based on hypothesis \( K^* \)
  \[ s_i(M, K^*) = M^T (A^T Z_i A) M + M^T (A^T (Z_i + Z_i^T)) K^* \]

- Offset for \( d_i \):
  \[ q_i(K) = K^T Z_i K + c_i \]

- For correct hypothesis: \( d_i = q_i(K) + s_i(M, K) \)
Quadratic combinatorial logic (cont’d)

- Let $\Gamma_i = A^T (Z_i + Z_i^T)$
- Let $\Omega_i = A^T Z_i A$
- Selection function becomes

$$s_i(M, K^*) = M^T \Omega_i M + M^T \Gamma_i K^*$$

- Incorrect hypothesis $K^*$ may partition traces correctly:

$$\Gamma_i K^* = \Gamma_i K \Rightarrow s_i(M, K^*) = s_i(M, K)$$
Quadratic combinatorial logic (cont’d)

- Decompose $K$ into $\kappa + k$ with
  - $\kappa \in \ker \Gamma_i$ implying $\Gamma_i \kappa = 0$
  - $k \in \mathcal{V}$ with $\mathcal{V} \cap \ker \Gamma_i = \{0\}$
  - So $k \neq 0$ implies $\Gamma_i k \neq 0$

- Hypothesis on $k$ instead of $K$
  - Dimension of $\mathcal{V}$ is $\text{rank} \Gamma_i$
  - $2^r$ hypotheses with $r = \text{rank} \Gamma_i$
  - Selection function now becomes:
    \[ s_i(M, k^*) = M^T \Omega_i M + M^T \Gamma_i k^* \]

- Incorrect hypothesis $k^* = k + \delta$ with $\delta \neq 0$
  - $s_i(M, k^*) = d_i(M, K) + q_i(K) + M^T \Gamma_i \delta$
  - If $M$ is random, no correlation between $d_i$ and $s_i(M, k^*)$
Outline

1. Introduction
2. Exploiting power consumption
3. Attacking unprotected KECCAK
4. Generalization to quadratic functions
5. Parasitic correlations
6. Attacking protected KECCAK
Parasitic correlations

Question: does other activity $d_j$ with $j \neq i$ act as noise across $\{M_0, M_1\}$?

- Not if $d_j$ is correlated with $s_i(M, k^*)$: parasitic correlations
- Component $j$ of consumption:

\[ d_j(M, K) = q_j(K) + M^T \Omega_j M + M^T \Gamma_j K \]

- Selection function aiming at $d_i$ with $k^* = k + \delta$

\[ s_i(M, k^*) = M^T \Omega_i M + M^T \Gamma_i (K + \delta) \]

- Correlation between $d_j(M, K)$ and $s_i(M, k^*)$ is imbalance of:

\[ f_{ij} = q_j(K) + M^T (\Omega_j + \Omega_i) M + M^T (\Gamma_j + \Gamma_i) K + M^T \Gamma_i \delta \]
Parasitic correlations (cont’d)

\[ f_{ij} = q_j(K) + M^T (\Omega_j + \Omega_i) M + M^T (\Gamma_j + \Gamma_i) K + M^T \Gamma_i \delta \]

- \( f_{ij} \) is balanced iff it has at least one isolated degree-1 term
- Degree-2 terms come from \( M^T (\Omega_j + \Omega_i) M \)
- Terms \( M^T (\Gamma_j + \Gamma_i) K \) and \( M^T \Gamma_i \delta \) supply linear terms
  - may be absorbed in degree-2 terms
  - \( d_j \) may be parasitic for \( s_i \) for subset of values of \( K \)
  - \( d_j \) may be parasitic for \( s_i \) for subset of hypotheses
- Correlation may be constructive or destructive
  - This depends on \( K, \delta \) and \( q_i(K) + q_j(K) \)
  - Success probability becomes key-dependent
Parasitic correlations in KECCAK

\[ f_{ij} \text{ with } i = (x, y, z) \text{ and } j = (x', y', z') \text{ is} \]

\[
f_{ij} = \mu_i + (\mu_{i+1} + 1)\mu_{i+2} + k_{i+1}^*\mu_{i+2} + k_{i+2}^*\mu_{i+1} + \\
\mu_j + (\mu_{j+1} + 1)\mu_{j+2} + k_{j+1}\mu_{j+2} + k_{j+2}\mu_{j+1} + q_i
\]

with \( i + 1 \) shorthand for \( (x + 1, y, z) \) and \( j + 1 \) for \( (x' + 1, y', z') \).

- Parasitic correlations if \( \mu_i(M) = \mu_j(M) \)
- Otherwise either \( \mu_j \) or \( \mu_i \) is an isolated linear term
- This can occur in KECCAK if the rate is smaller than \( 4/5b \) with \( b \) the width of KECCAK-\( f \)
Intermezzo: how $\theta$ works

- Compute parity $c_{x,z}$ of each column
- Add to each cell the parities of two nearby columns
Parasitic correlations in KECCAK

- Bits at output of $\theta$ in outer part are equal per column
- $\lambda = \pi \circ \rho \circ \theta$ with $\pi$ and $\rho$ just moving bits around
- So parasitic correlations in KECCAK occur if positions $j$ and $i$
  - come from same column
  - both come from inner part
Parasitic correlations in KECCAK (cont’d)

- If parasitic correlation, there are two degree-2 terms:
  - \((\mu_{i+1} + x_0)(\mu_{i+2} + x_1)\) and \((\mu_{j+1} + x_2)(\mu_{j+2} + x_3)\)
  - with \(x_i\) depending on \(K\) and the hypothesis

- If independent, correlation amplitude is 1/4

- If \(\mu_{i+1} = \mu_{j+1}\) or \(\mu_{i+2} = \mu_{j+2}\), recombination:
  - to single degree-2 term, parasitic correlation 1/2, or
  - introduction of linear term, removing parasitic correlation
  - Parasitic correlation 1/2 for two \(k^*\) and 0 for other two
  - Possible for certain positions depending on \(\rho\)

- If \(\mu_{i+1} = \mu_{j+1}\) and \(\mu_{i+2} = \mu_{j+2}\):
  - Parasitic correlation of 1 for one \(k^*\) and 0 for other three
  - Can only happen in toy widths 25, 50
Parasitic correlations in Keccak (cont’d)

- Assume: single parasitic correlation with amplitude 1/4
- Possibilities:
  - Mean for correct hypothesis: $-1.25, -0.75, 0.75$ or $1.25$
  - Mean for 3 incorrect hypothesis: $-0.25$ or $+0.25$
  - $4 \times 2^3$ possible combinations
- Actual mean values depend on $K$: all 32 equally likely
- Best strategy: choose hypothesis with highest peak
- Impact on success probability
  - Mean of correct $\pm 0.75$: $\Pr(\text{success})$ degenerates strongly
  - Mean of correct $\pm 1.25$: $\Pr(\text{success})$ still degenerates
Outline

1. Introduction
2. Exploiting power consumption
3. Attacking unprotected KECCAK
4. Generalization to quadratic functions
5. Parasitic correlations
6. Attacking protected KECCAK
Countermeasures

- Different levels
  - Transistor-level: e.g. WDDL, SecLib, ...
  - Platform-level: redundancy, adding jitter, noise, ...
  - Program-level: dummy instructions, randomized order, ...
  - Algorithmic level: depends on algebraic operations
  - Protocol level: key usage limits, session keys, ...

- No such thing as 100 % security
- Robustness: combine countermeasures at different levels
- Cost: area and consumption increase, loss of speed, ...
Secret sharing

- Countermeasure at algorithmic level:
  - Split variables in *random* shares: $x = a \oplus b \oplus \ldots$
  - Keep computed variables *independent* from *native* variables
  - Protection against $n$-th order DPA: at least $n + 1$ shares
Software: two-share masking

\[ \chi : x_i \leftarrow x_i + (x_{i+1} + 1)x_{i+2} \text{ becomes:} \]

\[
\begin{align*}
a_i &\leftarrow a_i + (a_{i+1} + 1)a_{i+2} + a_{i+1}b_{i+2} \\
b_i &\leftarrow b_i + (b_{i+1} + 1)b_{i+2} + b_{i+1}a_{i+2}
\end{align*}
\]

- Independence from native variables, if:
  - we compute left-to-right
  - we avoid leakage in register or bus transitions

\[ \lambda = \pi \circ \rho \circ \theta \text{ becomes:} \]

\[
\begin{align*}
a &\leftarrow \lambda(a) \\
b &\leftarrow \lambda(b)
\end{align*}
\]
**Software: two-share masking (faster)**

- Making it **faster!**
- $\chi$ becomes:

\[
\begin{align*}
a_i &\leftarrow a_i + (a_{i+1} + 1)a_{i+2} + a_{i+1}b_{i+2} + (b_{i+1} + 1)b_{i+2} + b_{i+1}a_{i+2} \\
b_i &\leftarrow b_i
\end{align*}
\]

- Precompute $R = b + \lambda(b)$
- $\lambda = \pi \circ \rho \circ \theta$ becomes:

\[
\begin{align*}
a &\leftarrow \lambda(a) + R \\
b &\leftarrow b
\end{align*}
\]
Hardware: two shares are not enough

- Unknown order in combinatorial logic!

\[ a_i \leftarrow a_i + (a_{i+1} + 1)a_{i+2} + a_{i+1}b_{i+2} \]
Using a threshold secret-sharing scheme

- Idea: **incomplete** computations only
  - Each circuit does not leak anything
    - [Nikova, Rijmen, Schläffer 2008]

- Number of shares: at least $1 +$ algebraic degree
  - 3 shares are necessary for $\chi$

- For higher degrees: degree-2 layers plus latches
Glitches as second-order effect

On a three-share implementation:
- A glitch can leak about two shares, say, $a + b$
- Another part can leak $c$
- $\Rightarrow$ as if two shares only!
Three-share masking for $\chi$

- Implementing $\chi$ in three shares:
  
  $$a_i \leftarrow b_i + (b_{i+1} + 1)b_{i+2} + b_{i+1}c_{i+2} + c_{i+1}b_{i+2}$$
  $$b_i \leftarrow c_i + (c_{i+1} + 1)c_{i+2} + c_{i+1}a_{i+2} + a_{i+1}c_{i+2}$$
  $$c_i \leftarrow a_i + (a_{i+1} + 1)a_{i+2} + a_{i+1}b_{i+2} + b_{i+1}a_{i+2}$$

- We present two architectures that implement this.
One-cycle round architecture
Power analysis of implementations protected with secret sharing. Application to Keccak.

Three-cycle round architecture.
High-order DPA

- Leakage of single bit is unrelated to native values
- \( n \)-th order DPA
  - Exploits statistical moments from \( n \)-th order
- \( n \)-th dimensional DPA
  - Considers \( n \) points in time
Simple example of second-order DPA

- Native \( x \) shared as \( x = a \oplus b \)
- When computing \( a \): \( P(t_1) = \alpha + (-1)^a \)
- When computing \( b \): \( P(t_2) = \alpha + (-1)^b \)
- Product:

\[
P(t_1)P(t_2) = \alpha^2 + \alpha(-1)^a + \alpha(-1)^b + (-1)^{a \oplus b} \\
= \alpha' + (-1)^x
\]
What happens if we compute simultaneously?

- Native $x$ shared as $x = a \oplus b$
- When simultaneous: $P(t) = \alpha + (-1)^a + (-1)^b$
- No linear correlation $C(P(t), x) = 0$
- But square:
  \[
  P(t)^2 = \alpha^2 + 2 + \alpha(-1)^a + \alpha(-1)^b + (-1)^{a \oplus b}
  = \alpha' + (-1)^x
  \]
- Or mutual information analysis (MIA):
  - $I(P(t); X) > 0$
Coding a bit in 1, 2 or 3 shares

(a) (b) (c)

(d) (e) (f)
Power consumption for three shares in parallel

\[ P = \alpha + \sum_{i} T[d_i(K, M)] \]

\[ T[d] = (-1)^A + (-1)^B + (-1)^{A \oplus B \oplus d} \]

- \( \Pr[T_0 = -1] = 3/4 \) and \( \Pr[T_0 = +3] = 1/4 \)
- \( \Pr[T_1 = +1] = 3/4 \) and \( \Pr[T_1 = -3] = 1/4 \)
- Mean 0, variance 3
Power consumption for three shares in parallel

\[
P = \alpha + \sum_{i=1}^{b} T[d_i(K, M)]
\]

- Consider the distribution of \(P(M)\) when \(s_i(M, K^*) = s\)
- Again, \(d_j \neq i\) uncorrelated to \(s_i(M, K^*)\) and normal dist.
- **Correct hypothesis:**
  \[
  \frac{1}{4}N(-3(-1)^q \oplus s, 3(b - 1)) + \frac{3}{4}N((-1)^q \oplus s, 3(b - 1))
  \]
- **Incorrect hypothesis:** \(N(0, 3b)\)
Kullback-Leibler divergence

\[ D(\text{correct} \parallel \text{incorrect}) \approx \frac{1}{9b^3} \]
Difference of asymmetry (DoA)

\[ \Delta_{\text{DoA}}(K^*) = |\mathbb{E}[P(M, K)^3|s_i(M, K^*) = 0] - \mathbb{E}[P(M, K)^3|s_i(M, K^*) = 1]| \]

- **Correct hypothesis**: \( \mathbb{E}[\Delta_{\text{DoA}}] = 12 \)
- **Incorrect hypothesis**: \( \mathbb{E}[\Delta_{\text{DoA}}] = 0 \)
- **Variance**: \( \sigma^2(\Delta_{\text{DoA}}) \approx \frac{1}{\#M} 24(3b)^3 \)
The distribution of DoA

- Variance: \( \sigma^2(\Delta_{\text{DoA}}) \approx \frac{1}{\#M} 24(3b)^3 \)
- Mean for incorrect hypothesis: 0, for correct: 12

- Same reasoning as for DoM, except \( \#M \sim b^3 \)
Result of experiments

- Simulation for KECCAK-\(f[25]\) to KECCAK-\(f[100]\)
- \(\sigma^2[\alpha] = 0\) and \(v_i = 1\)
- Positions \(i\) that do not exhibit parasitic correlations
- Results perfectly match predictions
Result of experiments

![Graph showing probability of success vs. number of traces for different values of b (25, 50, 100) for DoA and Cumulant MIA.]
KL divergence and probability of success

<table>
<thead>
<tr>
<th></th>
<th>Unprotected</th>
<th>Three-share</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D(\text{correct} \parallel \text{incorrect})$</td>
<td>$1/2b$</td>
<td>$1/9b^3$</td>
</tr>
<tr>
<td>$P_{\text{success}}$</td>
<td>$G_h(b/#M)$</td>
<td>$G_h(9b^3/2#M)$</td>
</tr>
</tbody>
</table>

$$P_{\text{success}} = G_h \left( \frac{1}{2D(\text{correct} \parallel \text{incorrect})\#M} \right)$$
Concluding remarks

- Minimalist approach
  - Focus on simple set of operations
  - Analyze leakage of one bit at a time

- Goals
  - Gain understanding on how it can leak
  - Gain confidence in countermeasures

- Degree-2 operations most suitable for masking?
  - See also [DPVR Noekeon] and [DPV FSE 2000]
Questions?

Thanks for your attention!

More information on
http://keccak.noekeon.org/
http://sponge.noekeon.org/