Side Channel Attacks against Block Ciphers Implementations and Countermeasures

Emmanuel Prouff
e.prouff@gmail.com

ANSSI

MCrypt – August 2014

Plan

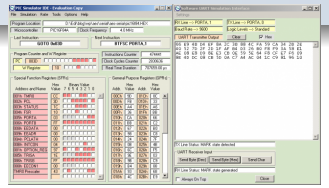
Advanced Side Channel Attacks (DPA like attacks)

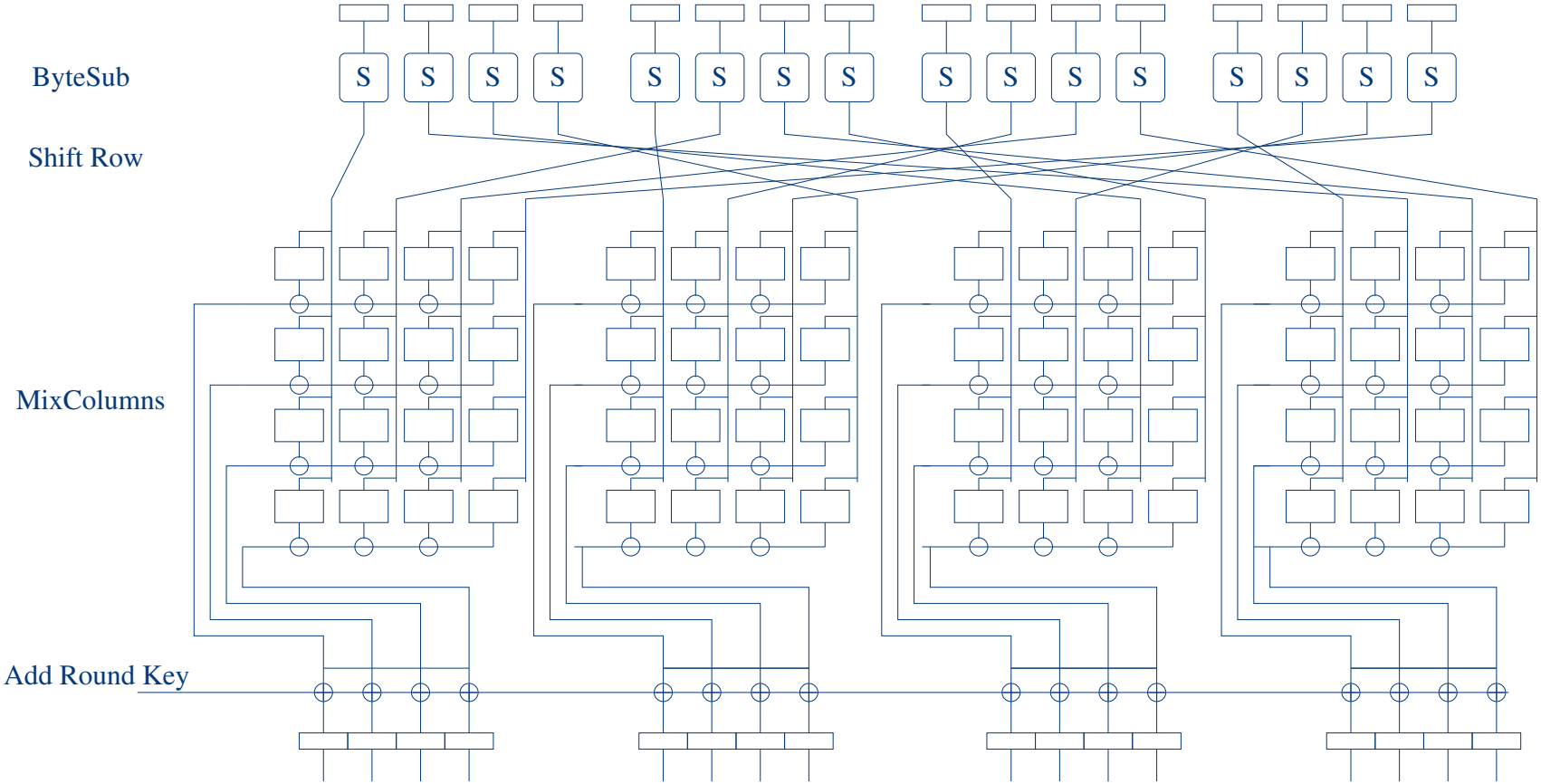
Introduction

- Advanced Side Channel Attacks can extract information from observations in contexts where SPA fails.
- They involve statistical tools (simple – difference of means tests – or sophisticated – mutual information processing –).
- They need several (between 10 and more than 10^6) traces such that:
  - the secret is constant,
  - the inputs are different and [optional] known.
  - [optional] some knowledge about the device architecture, the implementation or the noise characteristics.
- They follow a divide-and-conquer approach: the secret is rebuild piece by piece, where each piece is deduced from the behavior of an intermediate result. The size of the piece usually depends on the architecture size (*e.g.* 8, 16 or 32 bits).

Advanced Side Channel Attacks (DPA like attacks)

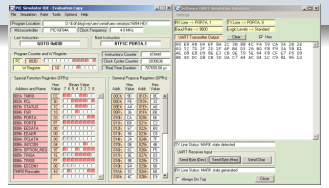
AES Round - 8-bit Software Implementation

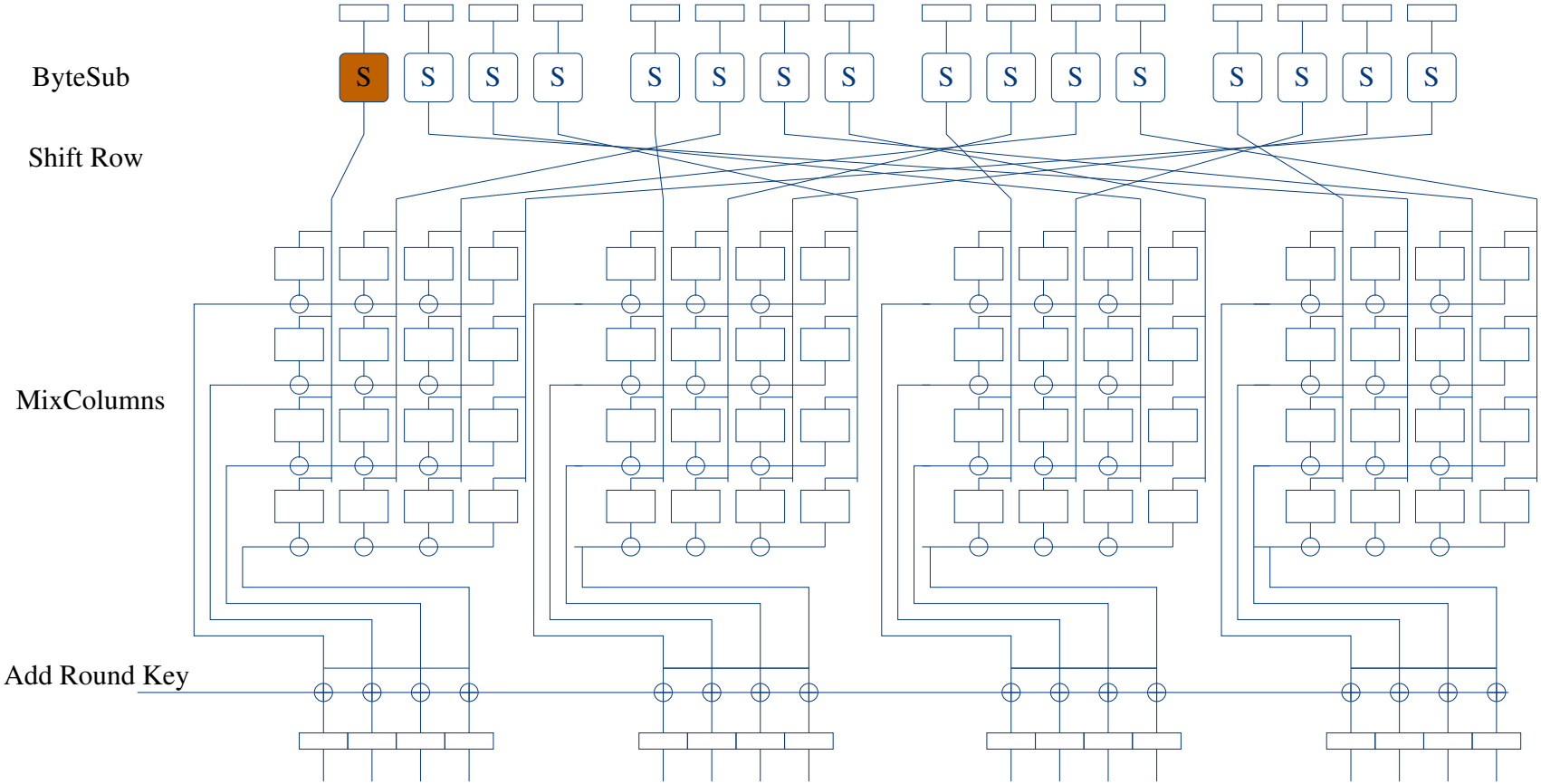




Advanced Side Channel Attacks (DPA like attacks)

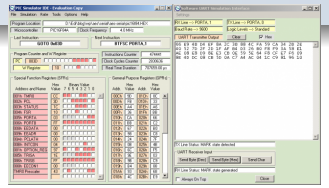
AES Round - 8-bit Software Implementation

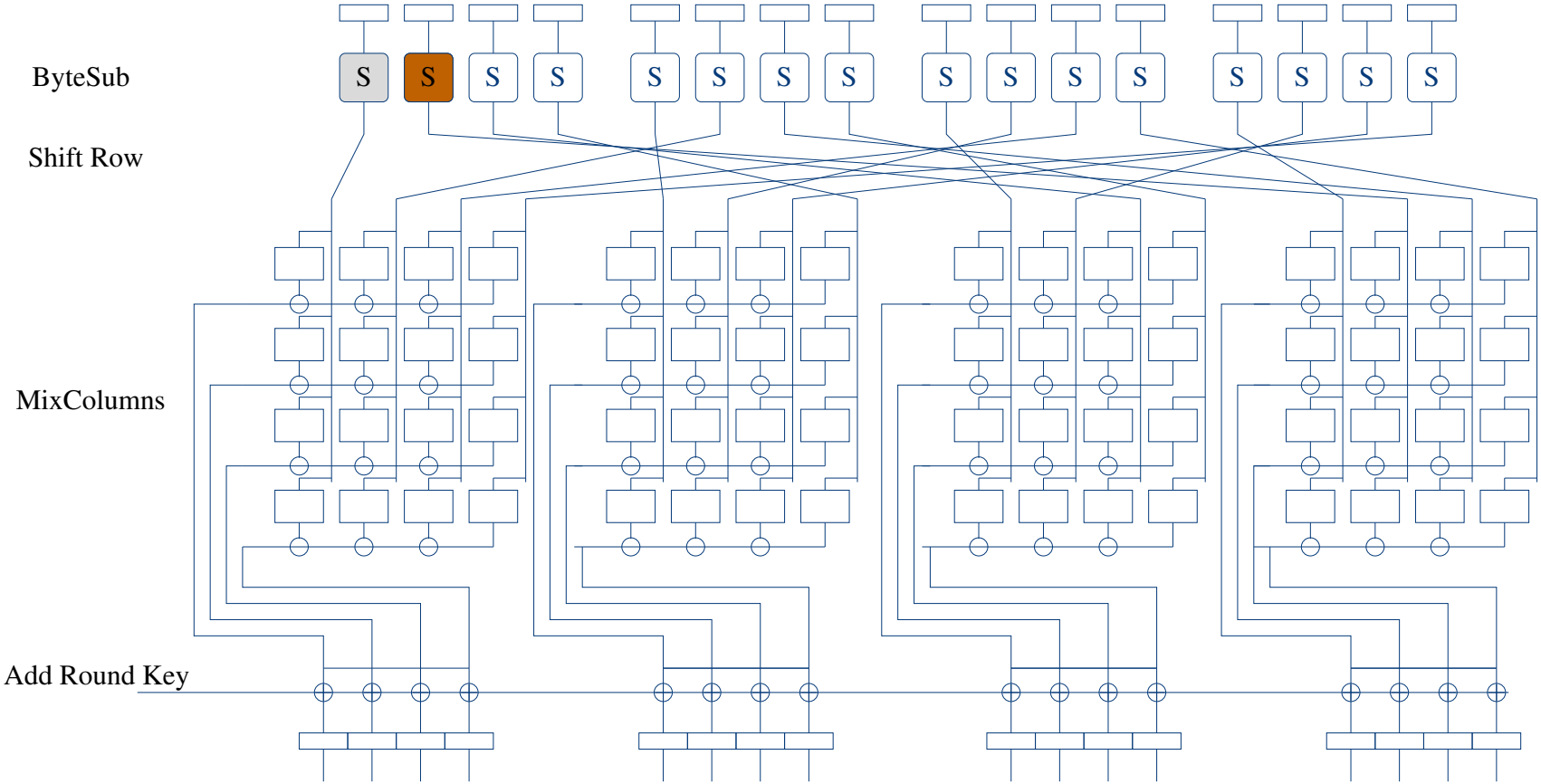




Advanced Side Channel Attacks (DPA like attacks)

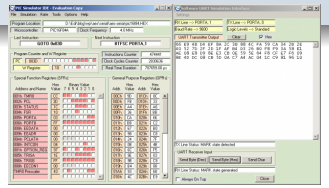
AES Round - 8-bit Software Implementation

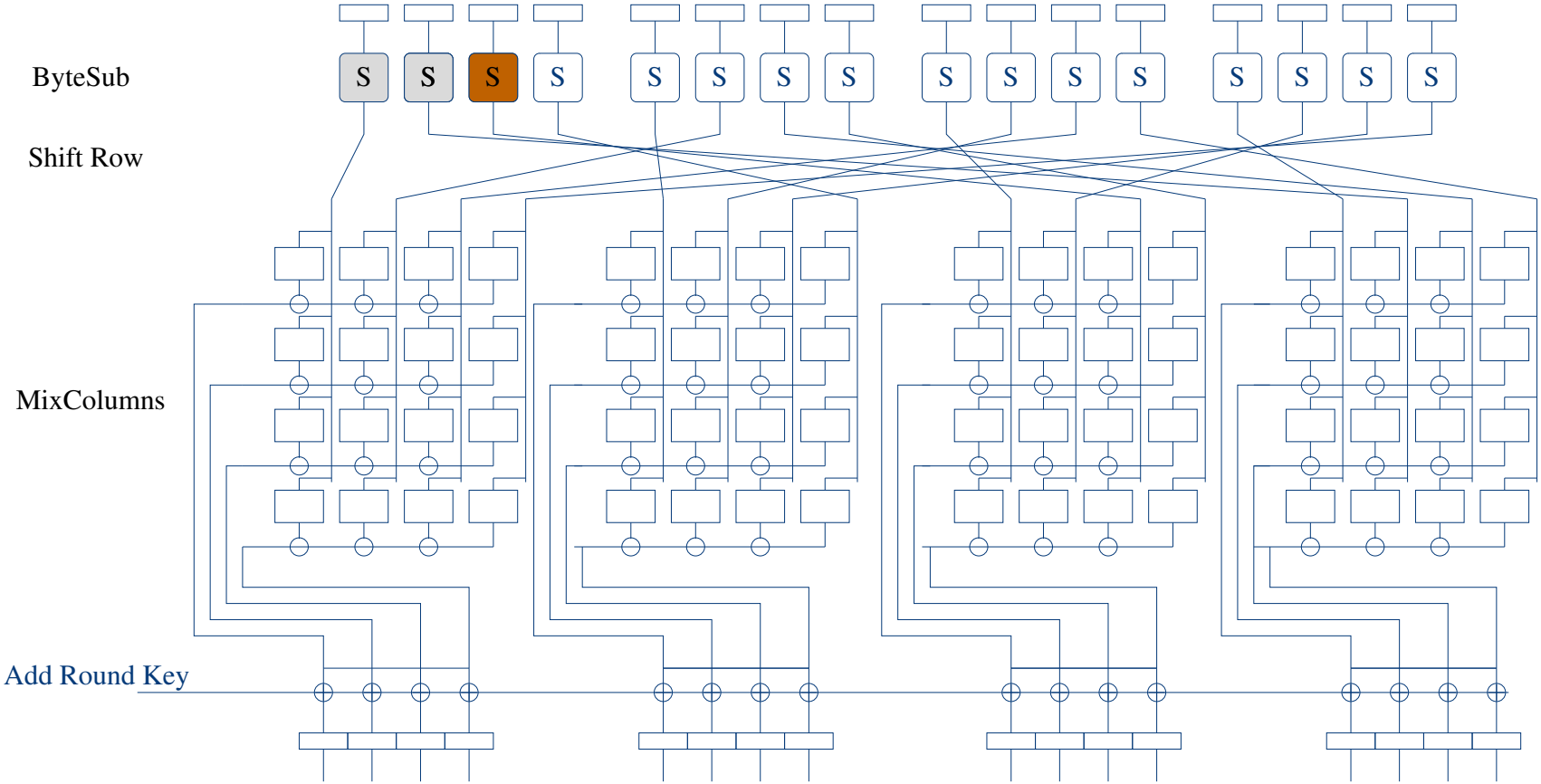




Advanced Side Channel Attacks (DPA like attacks)

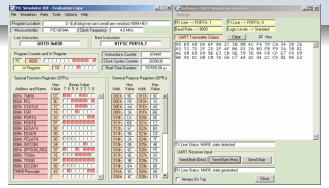
AES Round - 8-bit Software Implementation

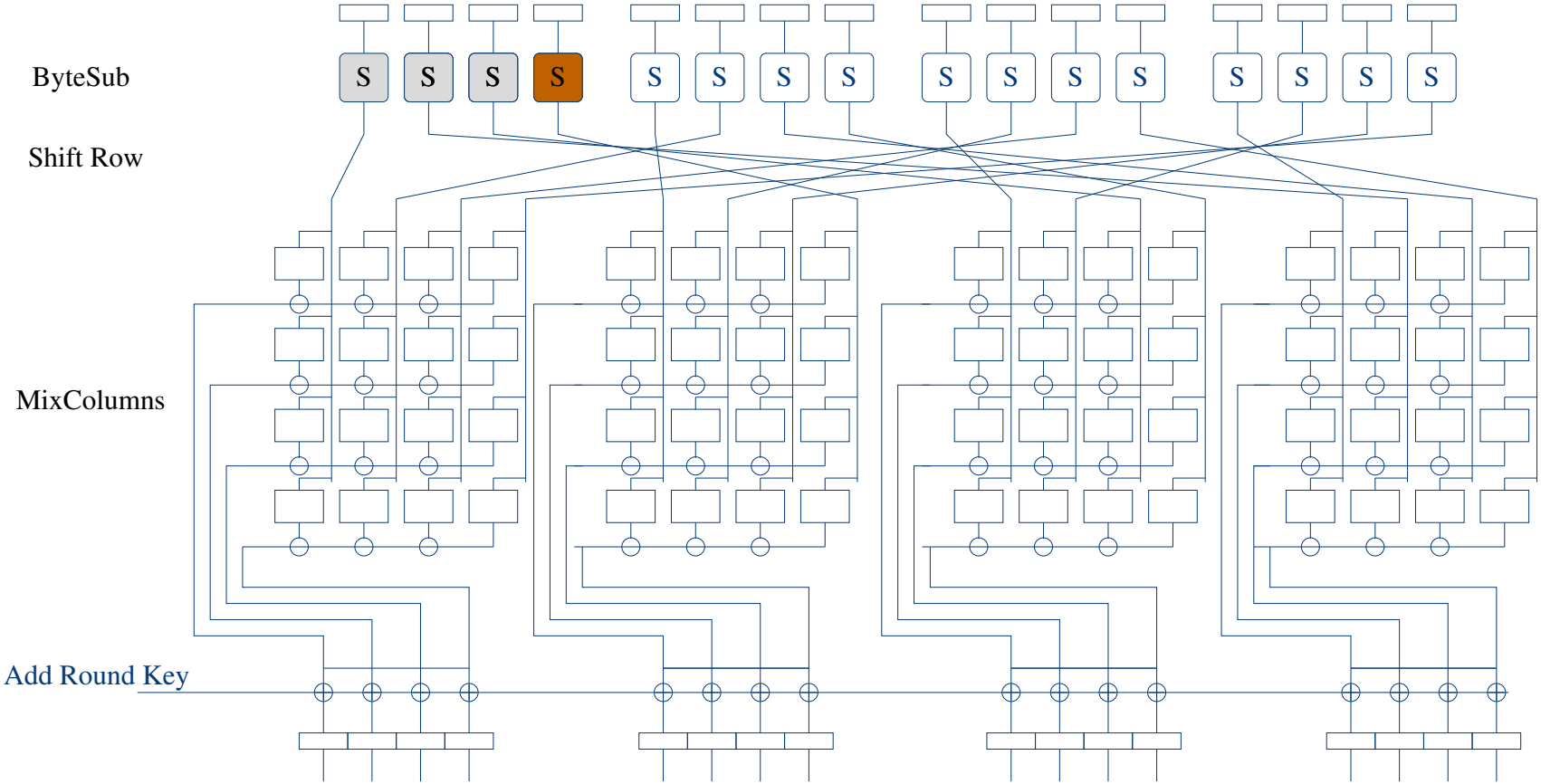




Advanced Side Channel Attacks (DPA like attacks)

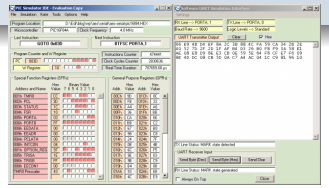
AES Round - 8-bit Software Implementation

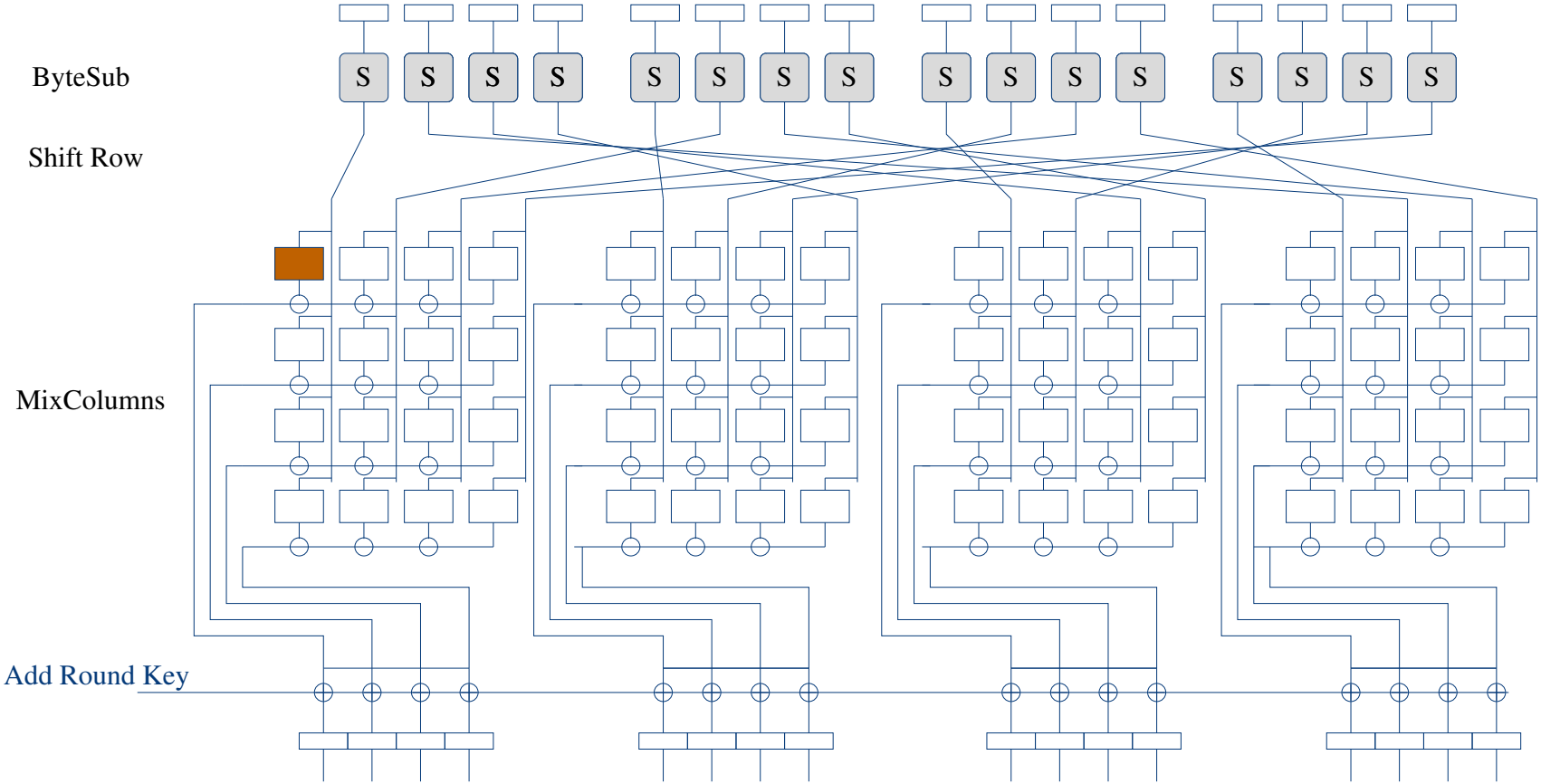




Advanced Side Channel Attacks (DPA like attacks)

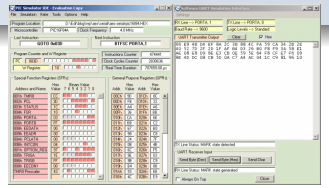
AES Round - 8-bit Software Implementation

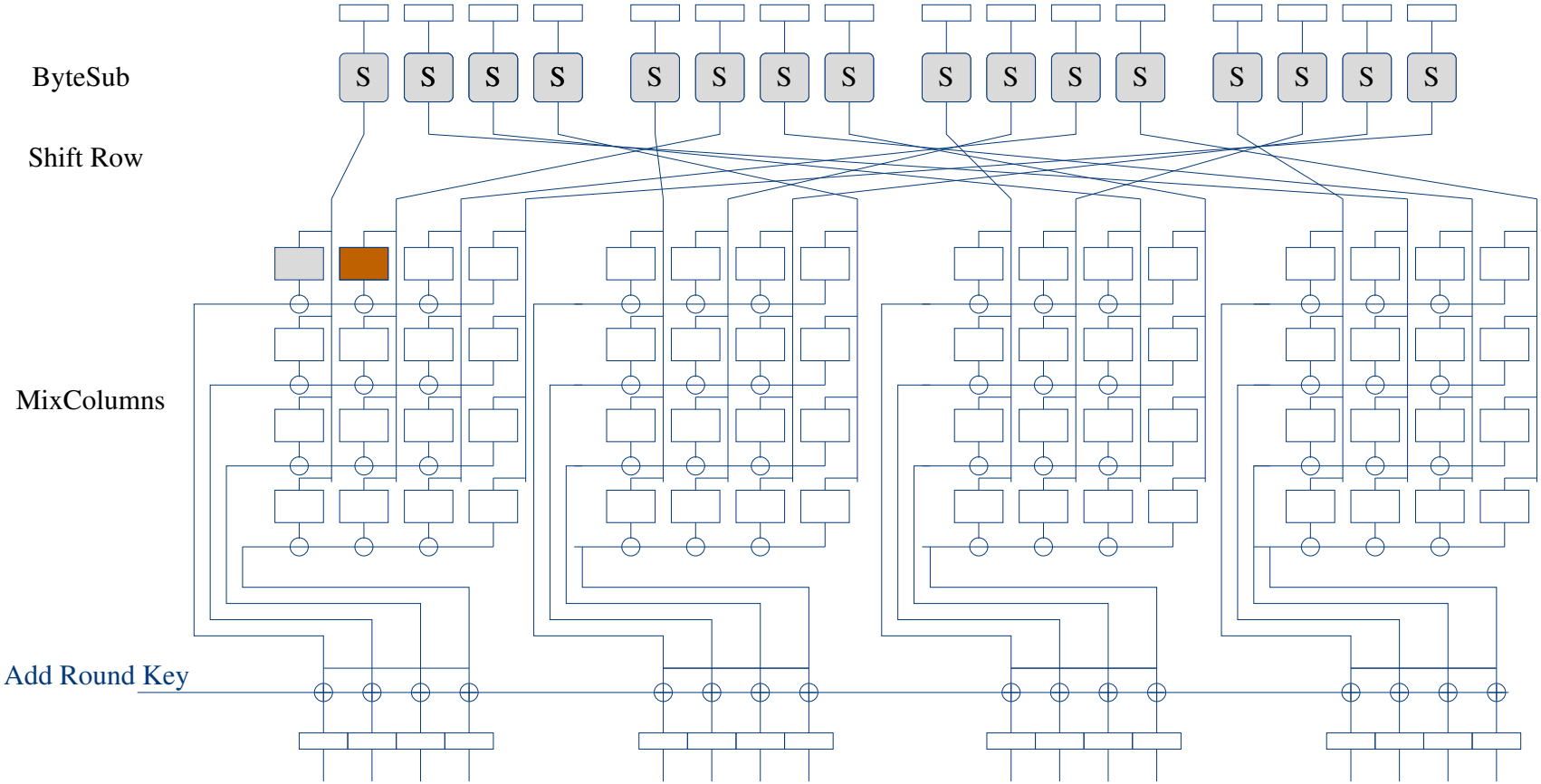




Advanced Side Channel Attacks (DPA like attacks)

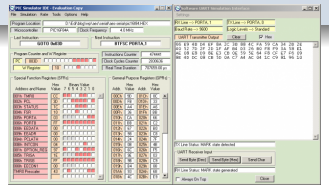
AES Round - 8-bit Software Implementation

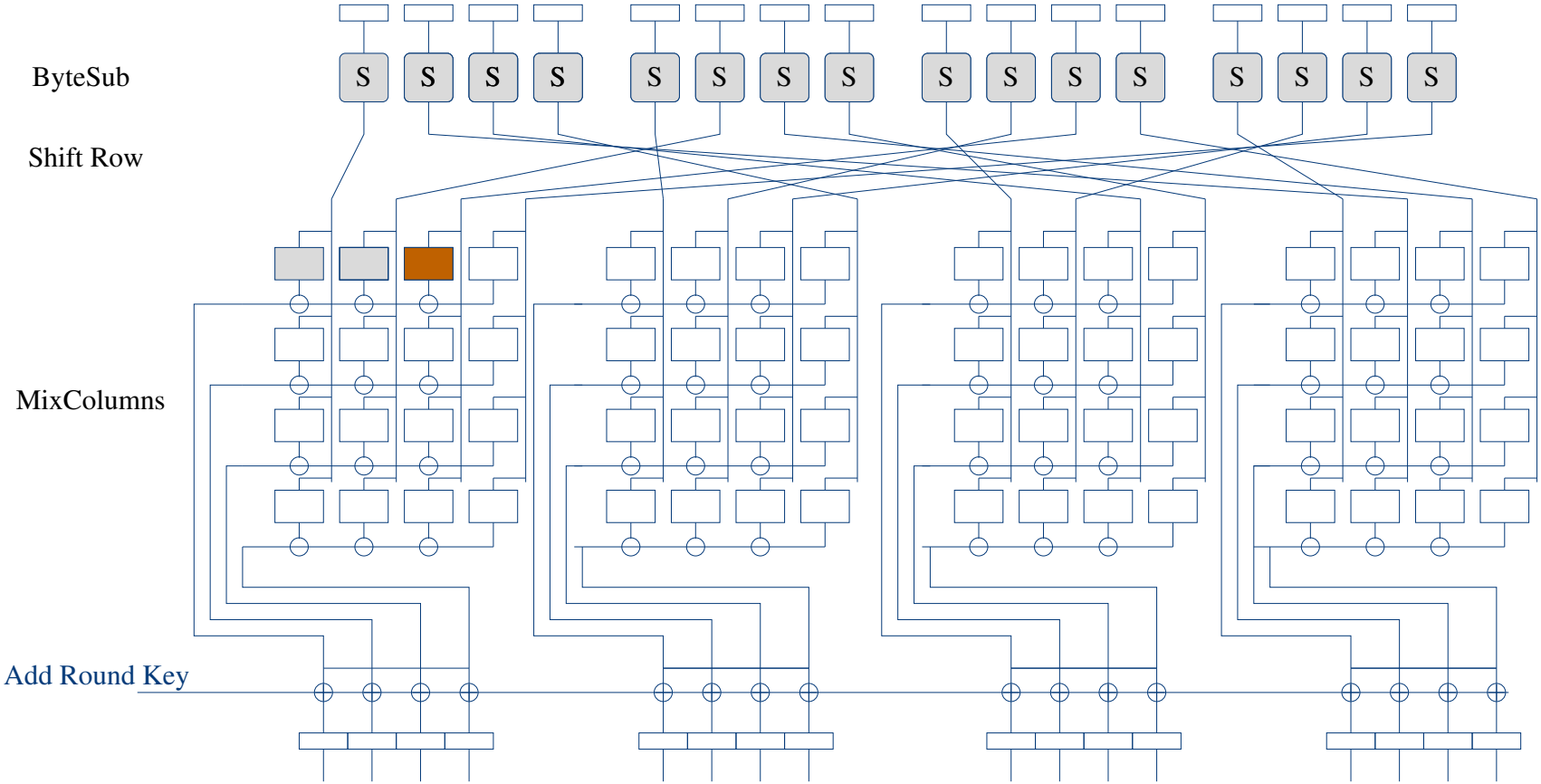




Advanced Side Channel Attacks (DPA like attacks)

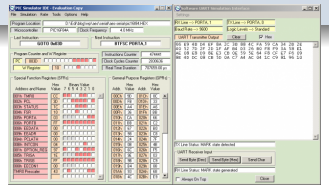
AES Round - 8-bit Software Implementation

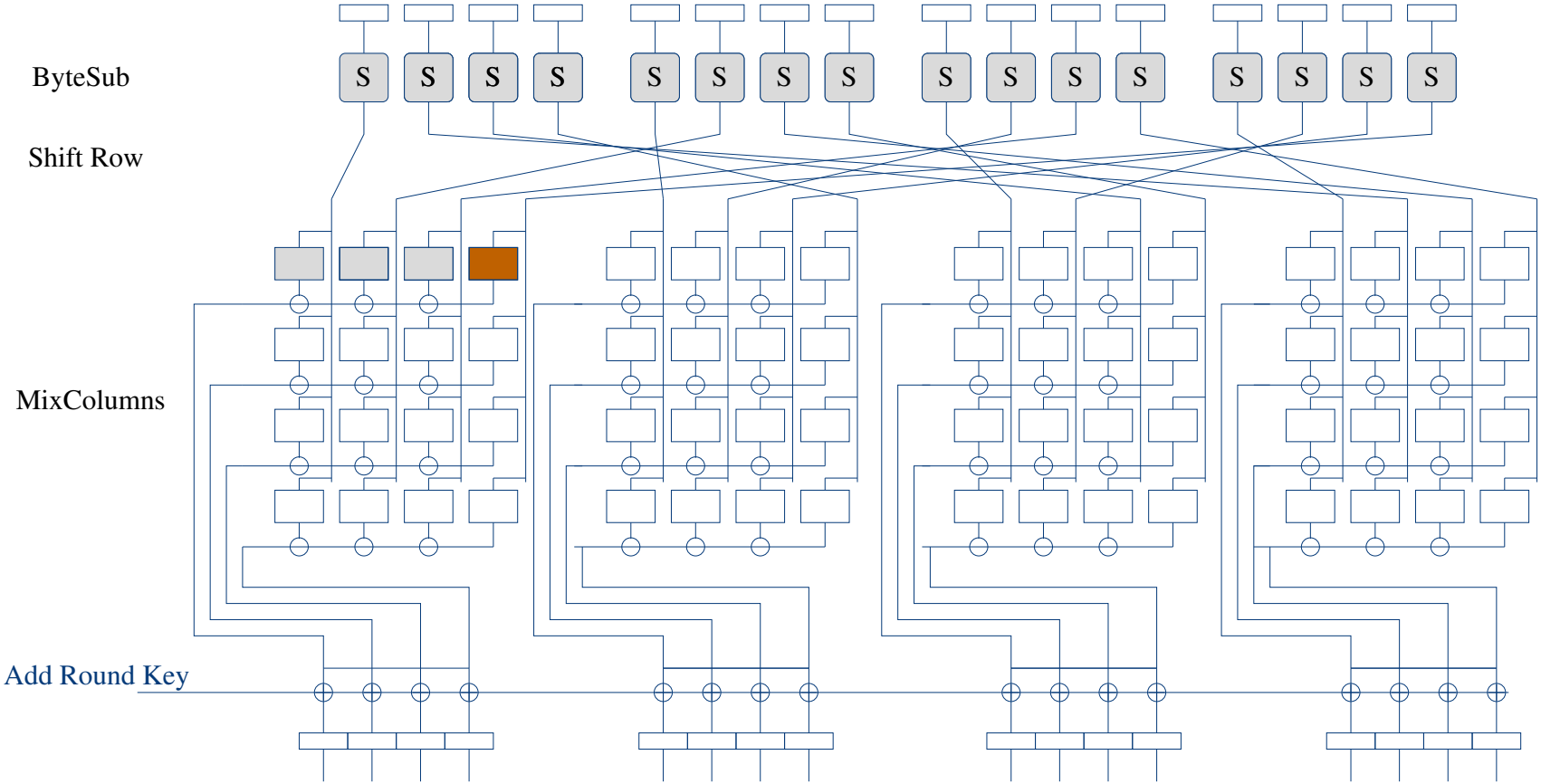




Advanced Side Channel Attacks (DPA like attacks)

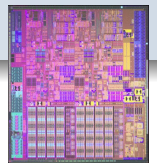
AES Round - 8-bit Software Implementation

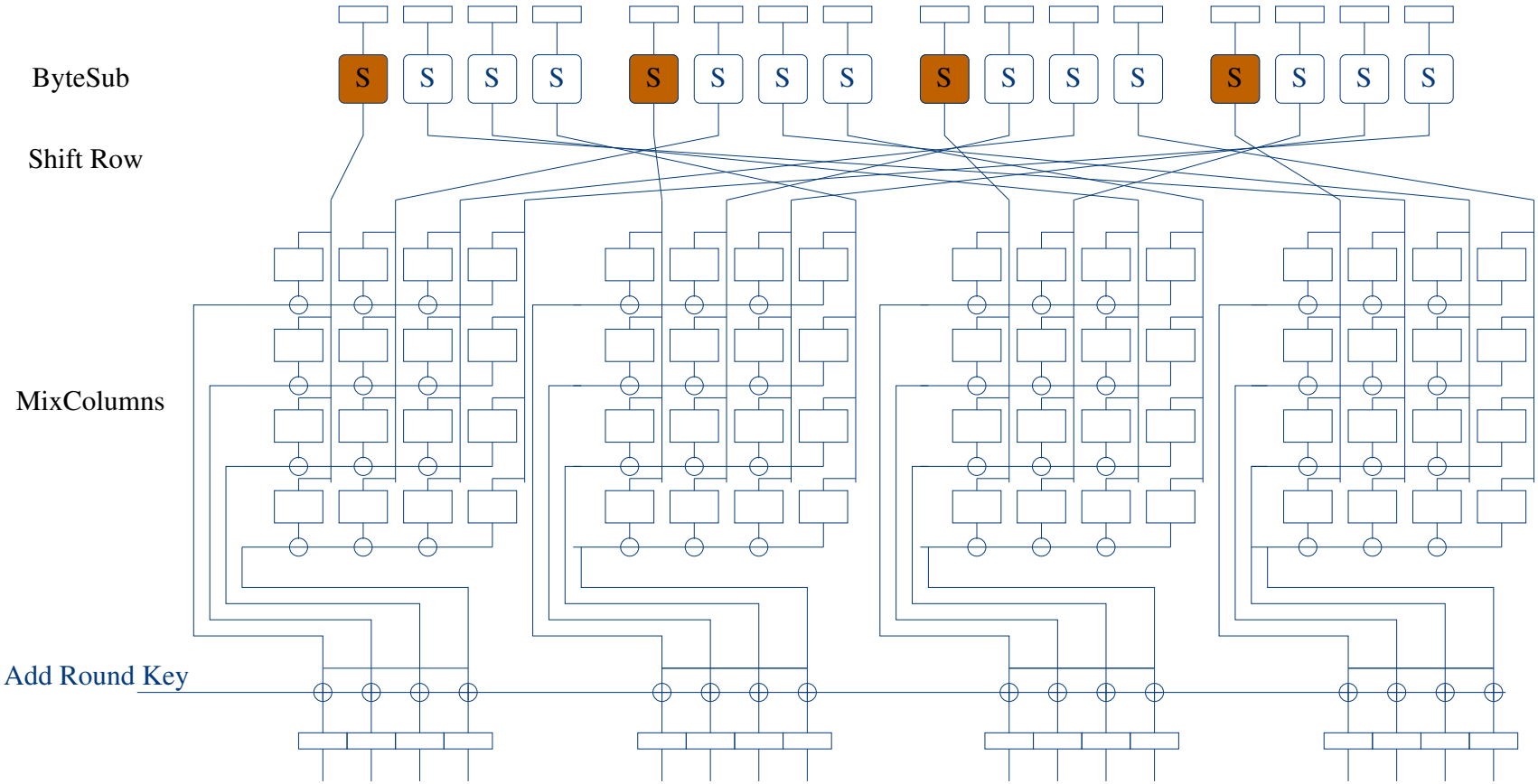




Advanced Side Channel Attacks (DPA like attacks)

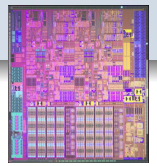
AES Round - Parallel Hardware Implementation

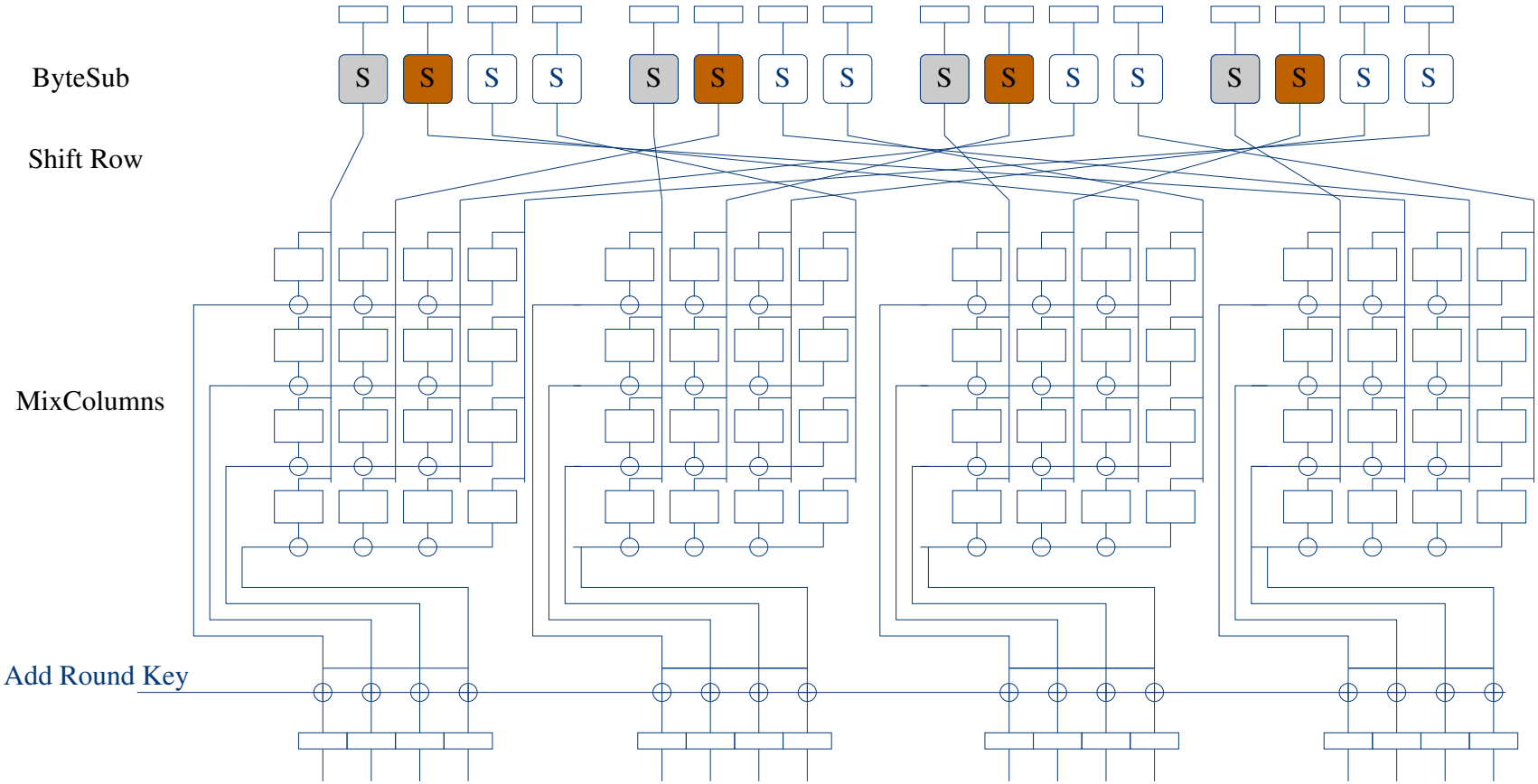




Advanced Side Channel Attacks (DPA like attacks)

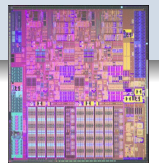
AES Round - Parallel Hardware Implementation

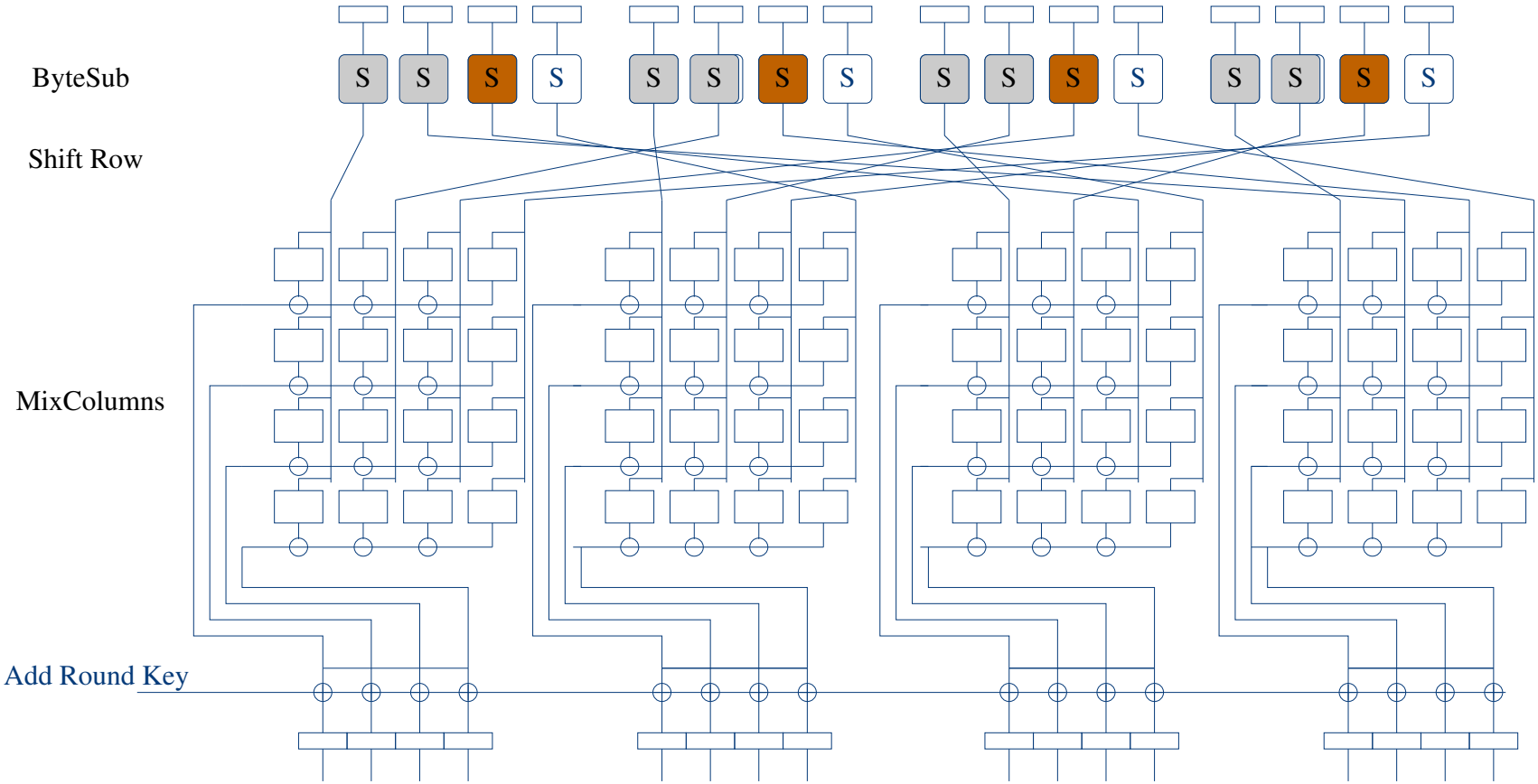




Advanced Side Channel Attacks (DPA like attacks)

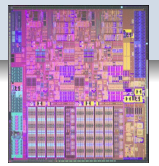
AES Round - Parallel Hardware Implementation

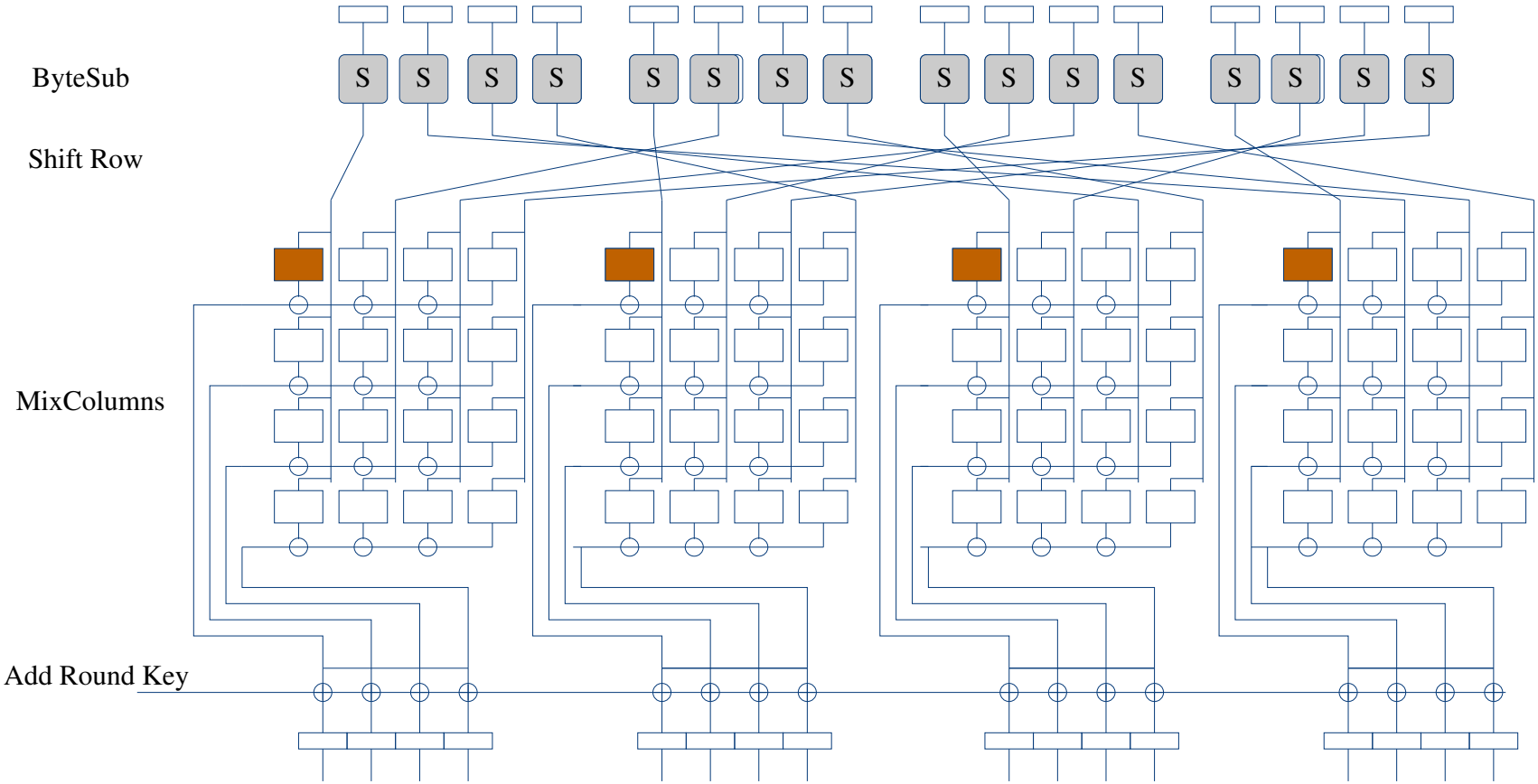




Advanced Side Channel Attacks (DPA like attacks)

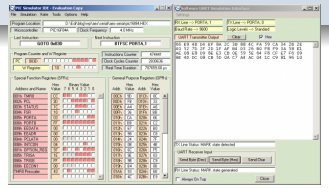
AES Round - Parallel Hardware Implementation

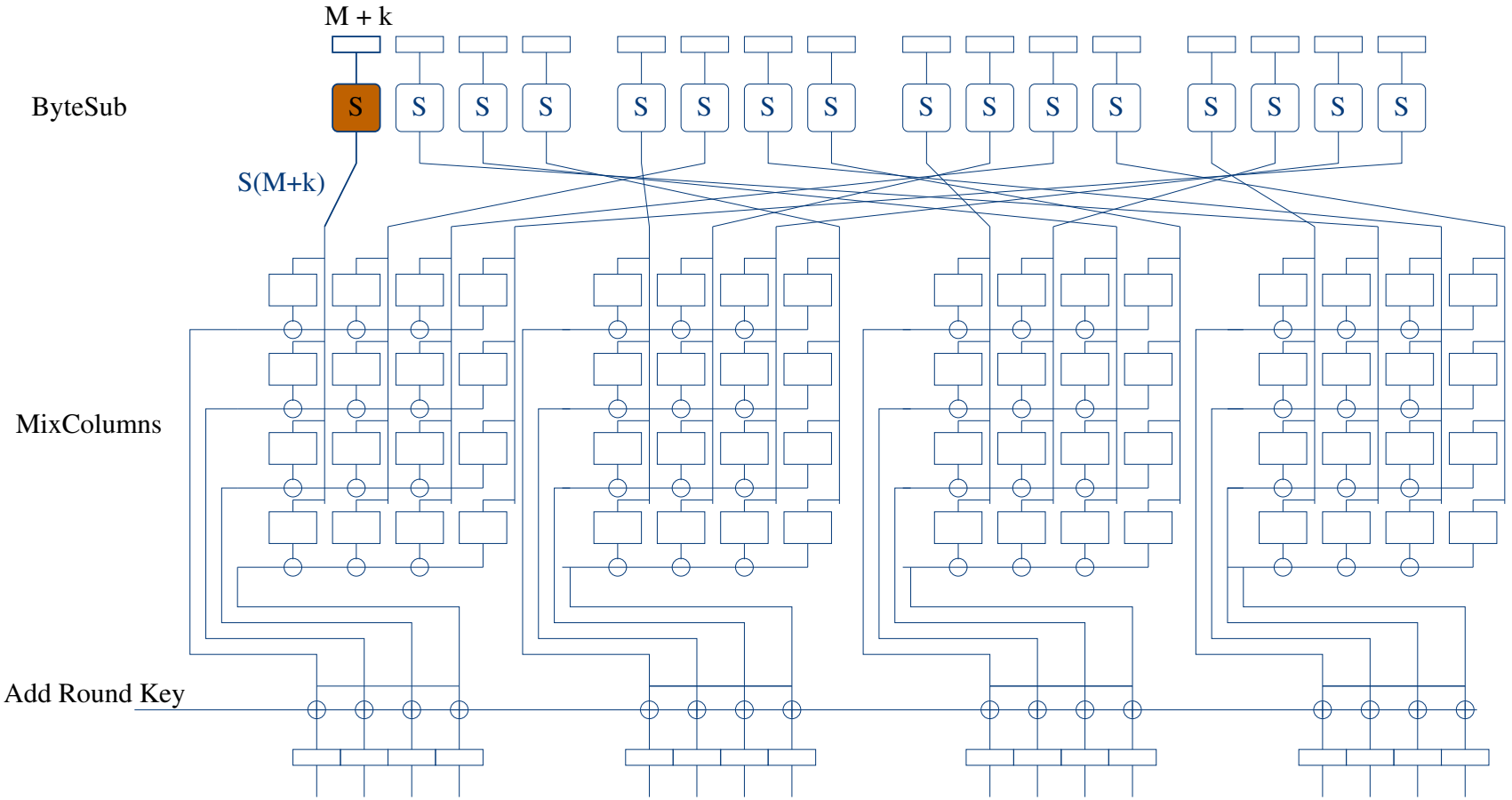




Advanced Side Channel Attacks (DPA like attacks)

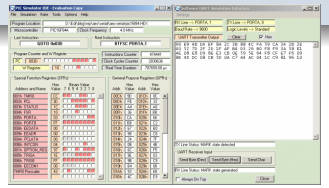
AES Round - Software Implementation – SCA attack

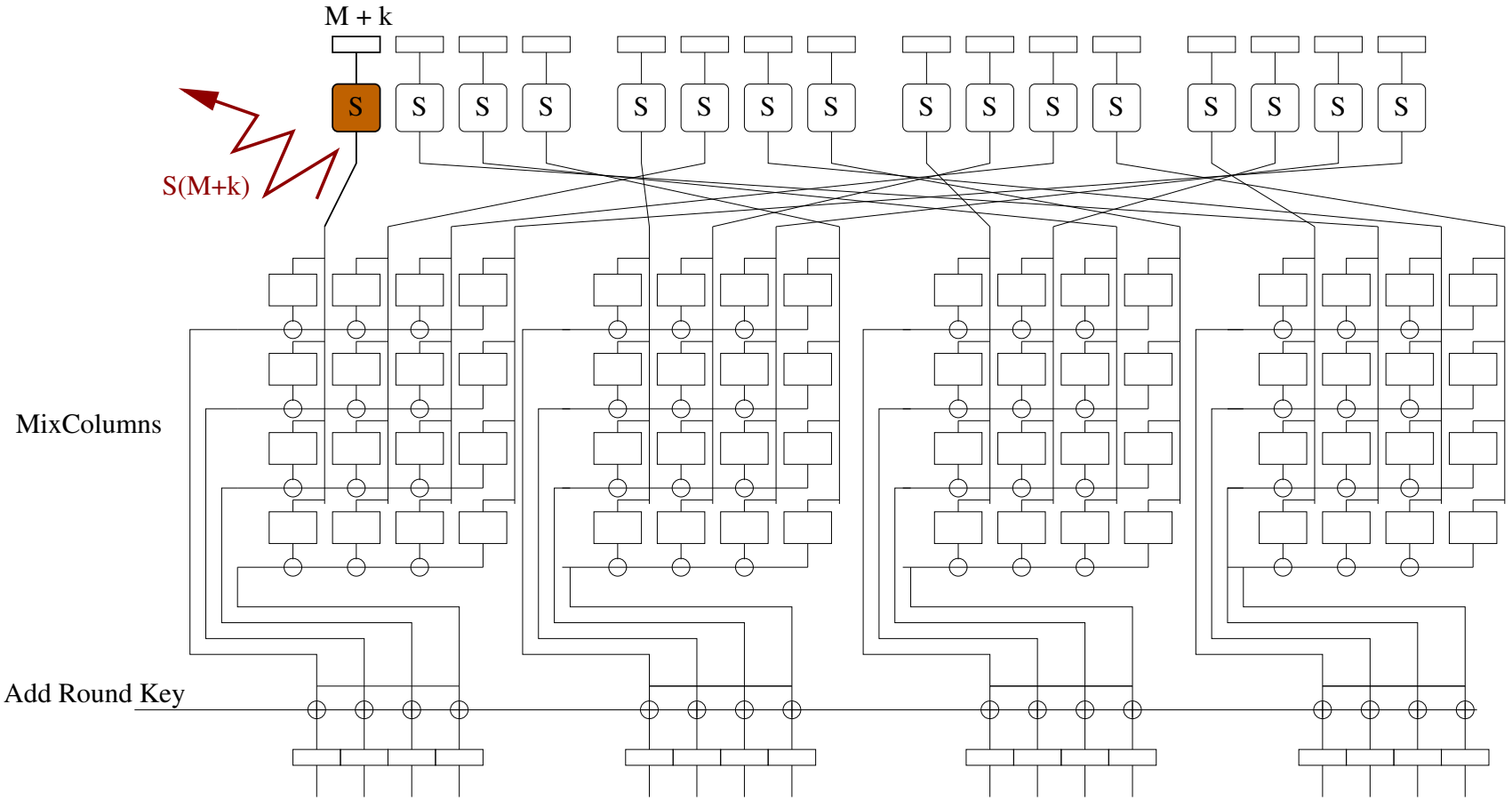




Advanced Side Channel Attacks (DPA like attacks)

AES Round - Software Implementation – SCA attack

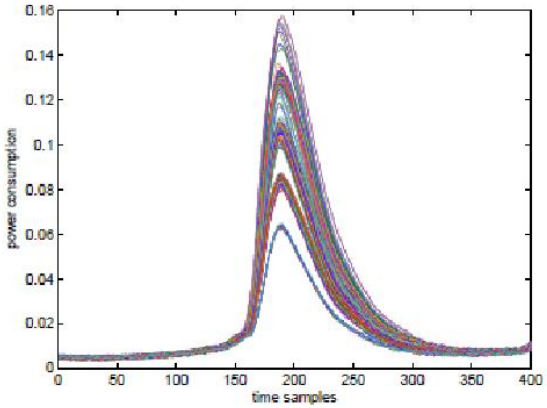


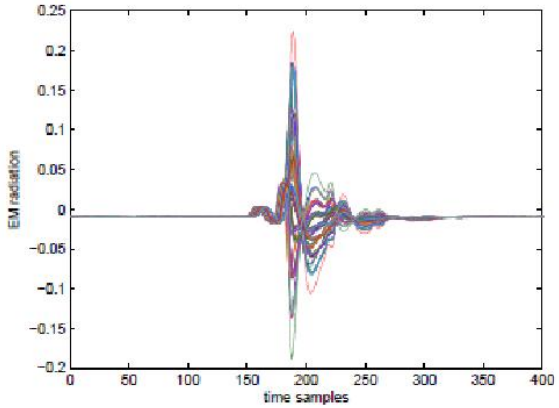


Advanced Side Channel Attacks (DPA like attacks)

Main Observation

Leakage at time t depends on the data manipulated at this time.





1. Power consumption leakage during the manipulation of a 8-bit variable by a card [Kocher, Jaffe and Jun, CRYPTO 1999].
2. Electromagnetic emanation during the same manipulation [Quisquater and Samyde, ESmart 2001].

Note 1: traces repartition does not look random.
Note 2: power consumptions are always positive whereas electromagnetic emanations are signed.

Advanced Side Channel Attacks (DPA like attacks)

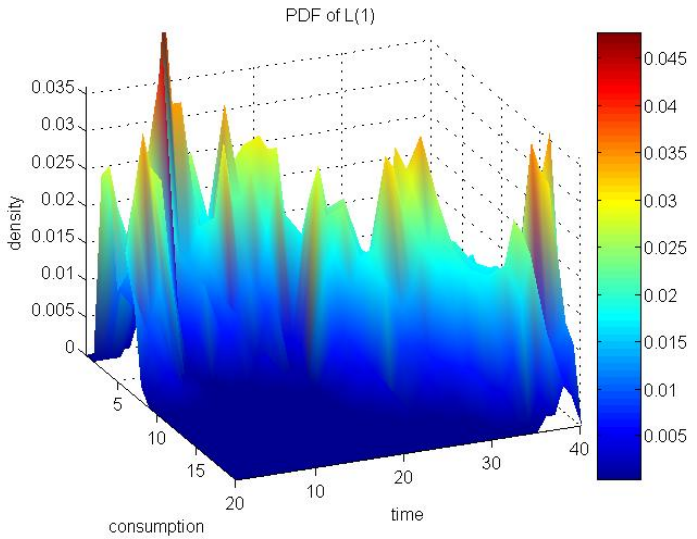
Main Observation

Example: pdf of the leakage for a device processing...

... $\texttt{AES-Sbox}(X + \mathbf{K})$ with $\mathbf{K} = 1$.

X varies uniformly
For each time (abs.) and each value ℓ in a finite interval (ord.) we plotted in z-axis:

$$Pr[\text{leakage} = \ell] \sim pdf_{\text{leakage}}(\ell)$$



Advanced Side Channel Attacks (DPA like attacks)

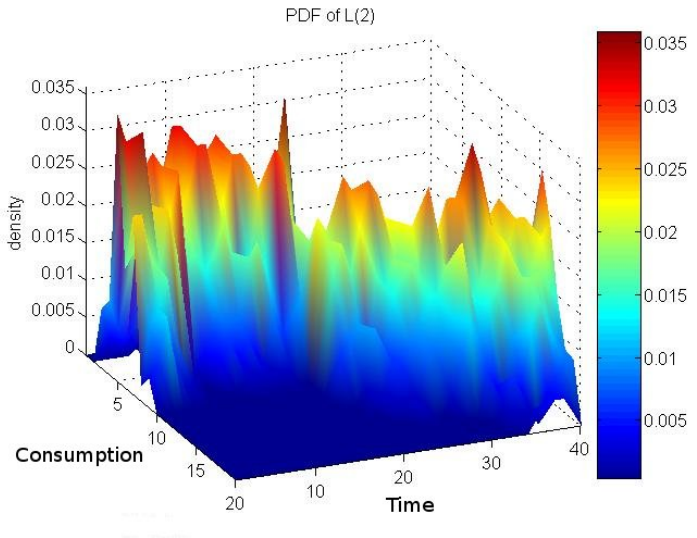
Main Observation

Example: pdf of the leakage for a device processing...

... $\texttt{AES-Sbox}(X + \mathbf{K})$ with $\mathbf{K} = 2$.

M varies uniformly
For each time (abs.) and each value ℓ in a finite interval (ord.) we plotted in z-axis:

$$Pr[\text{leakage} = \ell] \sim pdf_{\text{leakage}}(\ell)$$



Advanced Side Channel Attacks (DPA like attacks)

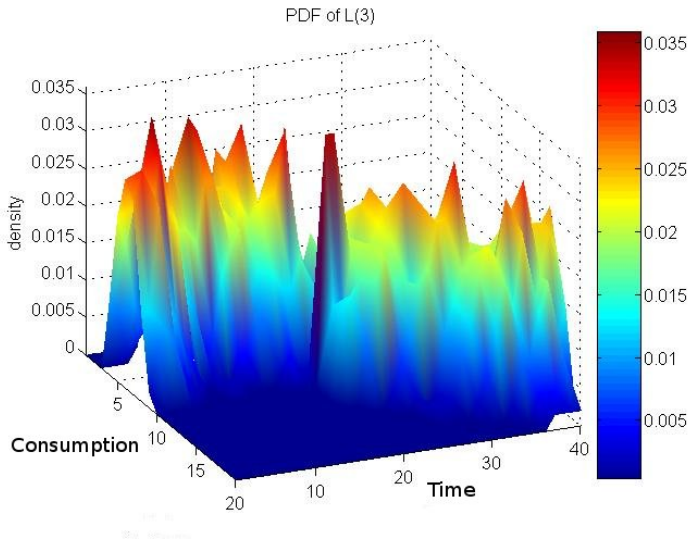
Main Observation

Example: pdf of the leakage for a device processing...

... $\texttt{AES-Sbox}(X + \mathbf{K})$ with $\mathbf{K} = 3$.

M varies uniformly
For each time (abs.) and each value ℓ in a finite interval (ord.) we plotted in z-axis:

$$Pr[\text{leakage} = \ell] \sim pdf_{\text{leakage}}(\ell)$$



Advanced Side Channel Attacks (DPA like attacks)

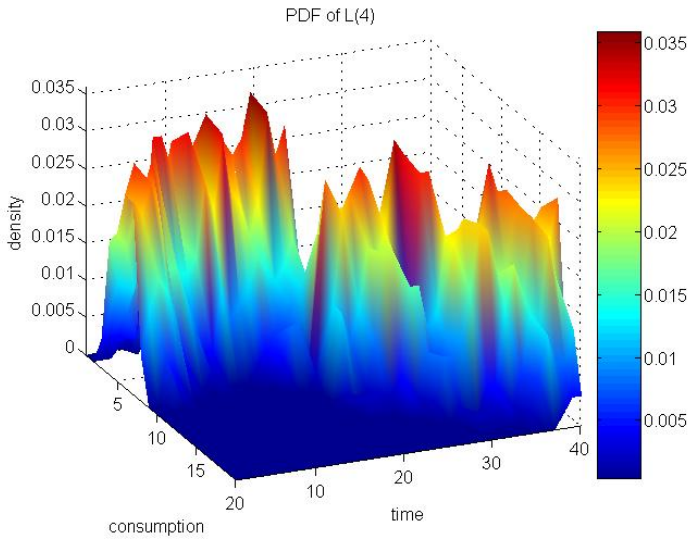
Main Observation

Example: pdf of the leakage for a device processing...

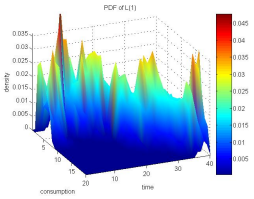
... $\texttt{AES-Sbox}(X + \mathbf{K})$ with $\mathbf{K} = 4$.

M varies uniformly
For each time (abs.) and each value ℓ in a finite interval (ord.) we plotted in z-axis:

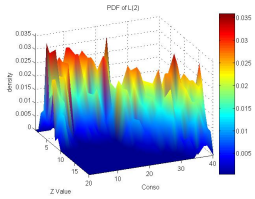
$$Pr[\text{leakage} = \ell] \sim pdf_{\text{leakage}}(\ell)$$



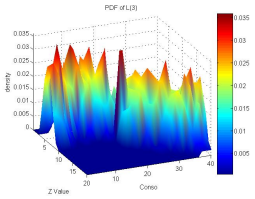
- [Pre-computation] For every possible key $k^\star$ pre-compute the pdf of the leakage $\mathbf{L}$.



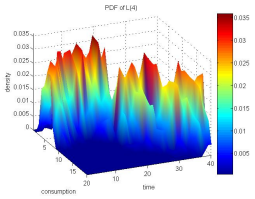
$k^\star = 1$



$k^\star = 2$



$k^\star = 3$

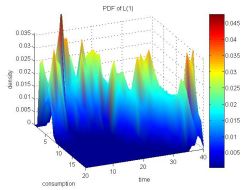


$k^\star = 4$

.....

  - [Necessary Condition] Have an open access to a copy of the target device and be able to choose the key value.

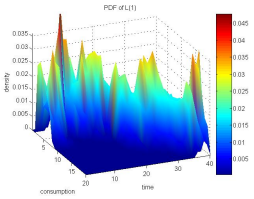
- [Measurement] Measure the consumption for the target device and estimate the pdf of $\mathbf{L}$ for this target.



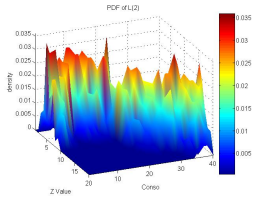
$k^\star = ?$

- [Key-recovery] Compare the pdf estimation with those pre-computed and output the most likely key candidate.

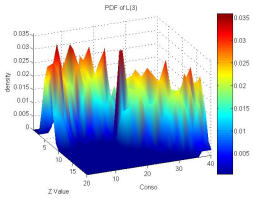
- [Pre-computation] For every possible key $k^\star$ pre-compute the pdf of the leakage $\mathbf{L}$.



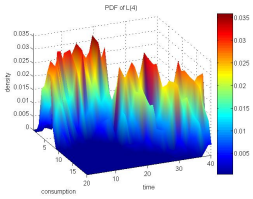
$k^\star = 1$



$k^\star = 2$



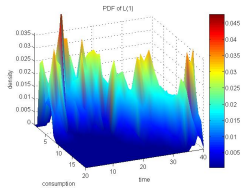
$k^\star = 3$



$k^\star = 4$ .....

  - [Necessary Condition] Have an open access to a copy of the target device and be able to choose the key value.

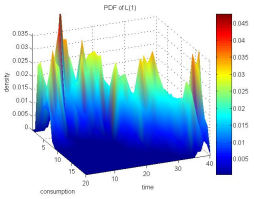
- [Measurement] Measure the consumption for the target device and estimate the pdf of $\mathbf{L}$ for this target.



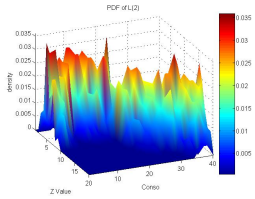
$k^\star = ?$

- [Key-recovery] Compare the pdf estimation with those pre-computed and output the most likely key candidate.

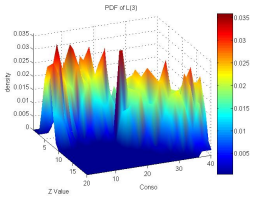
- [Pre-computation] For every possible key $k^\star$ pre-compute the pdf of the leakage **L**.



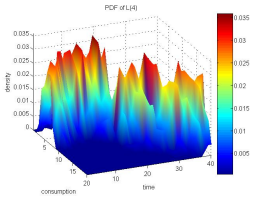
$k^\star = 1$



$k^\star = 2$



$k^\star = 3$

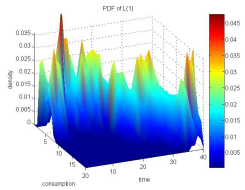


$k^\star = 4$

.....

  - [Necessary Condition] Have an open access to a copy of the target device and be able to choose the key value.

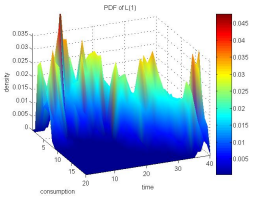
- [Measurement] Measure the consumption for the target device and estimate the pdf of **L** for this target.



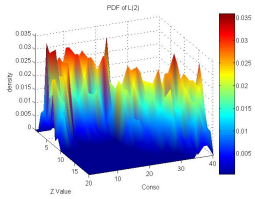
$k^\star = ?$

- [Key-recovery] Compare the pdf estimation with those pre-computed and output the most likely key candidate.

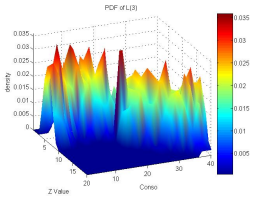
- [Pre-computation] For every possible key $k^\star$ pre-compute the pdf of the leakage **L**.



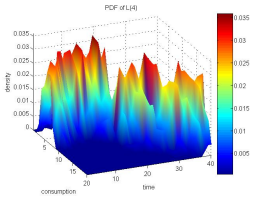
$k^\star = 1$



$k^\star = 2$



$k^\star = 3$

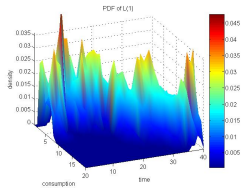


$k^\star = 4$

.....

  - [Necessary Condition] Have an open access to a copy of the target device and be able to choose the key value.

- [Measurement] Measure the consumption for the target device and estimate the pdf of **L** for this target.

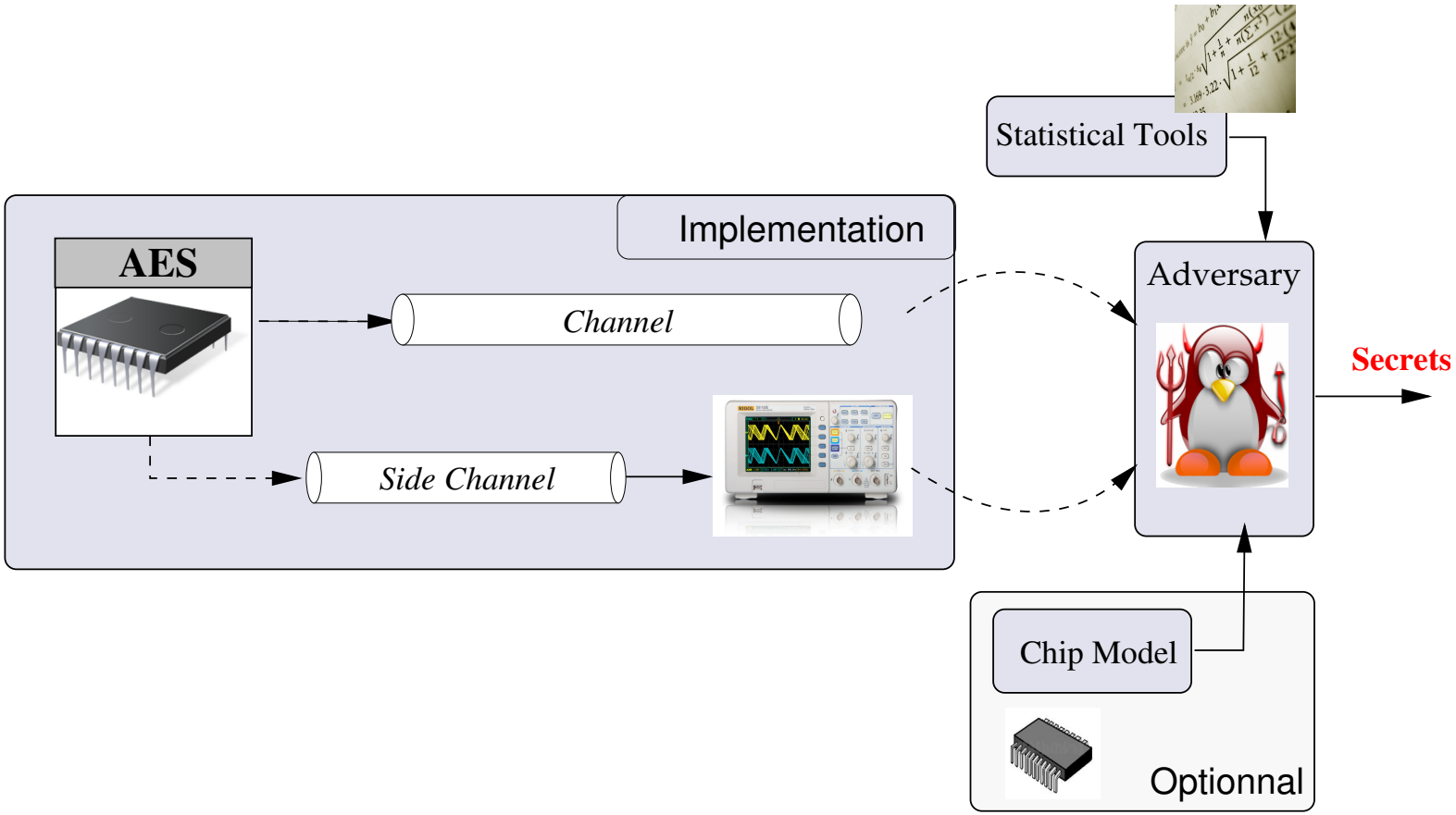


$k^\star = ?$

- [Key-recovery] Compare the pdf estimation with those pre-computed and output the most likely key candidate.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework.



Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $Z = S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $Z = S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $Z = S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $Z = S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $Z = S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $Z = S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) \quad .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $Z = S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) \quad .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: attack Description Sheet/Form

Attack Description Sheet/Form

Type of Leakage: *e.g. power consumption or electromagnetic emanation*

Model Function: *e.g. one bit of Z or its Hamming weight*

Statistical Distinguisher: *e.g. difference of means, correlation or entropy*

Key Candidate Selection: *e.g. the candidate the maximizes the scores*

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) \quad .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

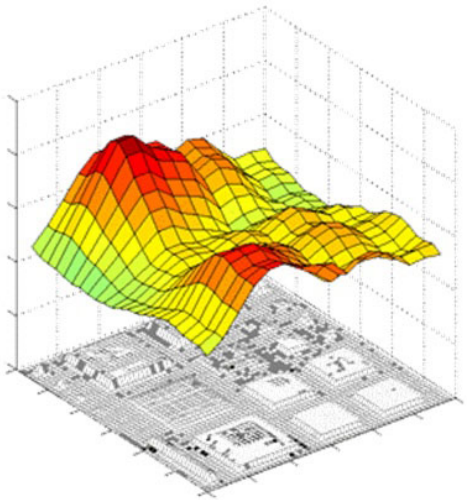
6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: define a model for the consumption.

Goal: define the kind of dependency between the manipulated data and the device behaviour.

- First solution (template/profiled attacks principle):
  - use an exact copy of the attacked device and estimate the pdf of L for every possible pair (X, k).
  - see [Chari et al at CHES 2002].
- Second solution (unprofiled attacks principle):
  - model the function $\mathbf{E}[L|\ X = x, K = k]$.
  - see Messerges PhD Thesis.

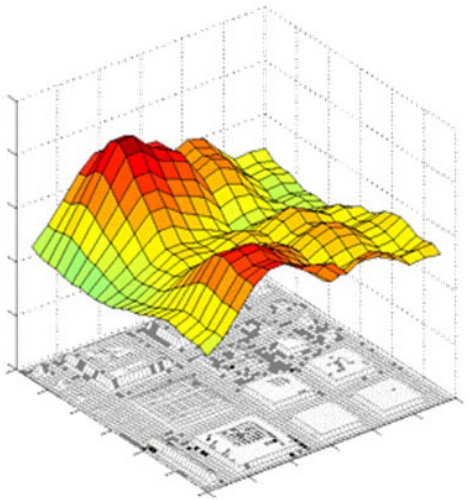


Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: define a model for the consumption.

Goal: define the kind of dependency between the manipulated data and the device behaviour.

- First solution (template/profiled attacks principle):
  - use an exact copy of the attacked device and estimate the pdf of L for every possible pair (X, k).
  - see [Chari et al at CHES 2002].
- Second solution (unprofiled attacks principle):
  - model the function $\mathbf{E}[L| X = x, K = k]$.
  - see Messerges PhD Thesis.

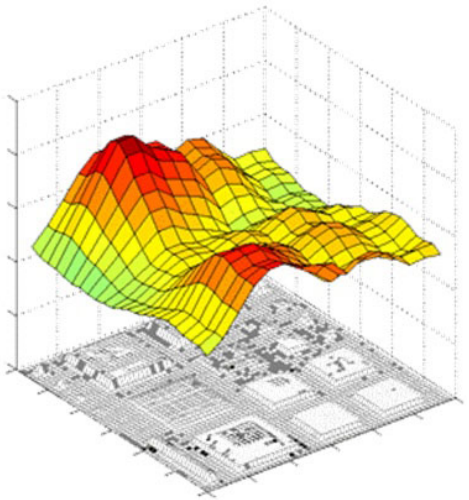


Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: define a model for the consumption.

Goal: define the kind of dependency between the manipulated data and the device behaviour.

- First solution (template/profiled attacks principle):
  - use an exact copy of the attacked device and estimate the pdf of L for every possible pair (X, k).
  - see [Chari et al at CHES 2002].
- Second solution (unprofiled attacks principle):
  - model the function $\mathbf{E}[L|\ X = x, K = k]$.
  - see Messerges PhD Thesis.



Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

Independent Noise Assumption (INA)

The random variable L related to the manipulation of Z equals $Y + B$, where Y is a function of Z and B is independent of Z.

- B is usually called the noise and is viewed as a continuous random variable.
- We usually assume $B \sim \mathcal{N}(0, \sigma^2)$. (Gaussian Noise Assumption).
- Usually, we have $Z = S(X + K)$ where
  - X is known,
  - k is the secret to recover
  - $S(\cdot)$ is a known cryptographic primitive (*e.g. an s-box*).

New problem statement

Modelling = recover the function φ s.t. $Y = \varphi(Z)$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

Independent Noise Assumption (INA)

The random variable L related to the manipulation of Z equals $Y + B$, where Y is a function of Z and B is independent of Z.

- B is usually called the noise and is viewed as a continuous random variable.
- We usually assume $B \sim \mathcal{N}(0, \sigma^2)$. (Gaussian Noise Assumption).
- Usually, we have $Z = S(X + K)$ where
  - X is known,
  - k is the secret to recover
  - $S(\cdot)$ is a known cryptographic primitive (*e.g. an s-box*).

New problem statement

Modelling = recover the function φ s.t. $Y = \varphi(Z)$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

Independent Noise Assumption (INA)

The random variable L related to the manipulation of Z equals $Y + B$, where Y is a function of Z and B is independent of Z.

- B is usually called the noise and is viewed as a continuous random variable.
- We usually assume $B \sim \mathcal{N}(0, \sigma^2)$. (Gaussian Noise Assumption).
- Usually, we have $Z = S(X + K)$ where
  - X is known,
  - k is the secret to recover
  - $S(\cdot)$ is a known cryptographic primitive (*e.g. an s-box*).

New problem statement

Modelling = recover the function φ s.t. $Y = \varphi(Z)$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

Independent Noise Assumption (INA)

The random variable L related to the manipulation of Z equals $Y + B$, where Y is a function of Z and B is independent of Z.

- B is usually called the noise and is viewed as a continuous random variable.
- We usually assume $B \sim \mathcal{N}(0, \sigma^2)$. (Gaussian Noise Assumption).
- Usually, we have $Z = S(X + K)$ where
  - X is known,
  - k is the secret to recover
  - $S(\cdot)$ is a known cryptographic primitive (*e.g. an s-box).*

New problem statement

Modelling = recover the function φ s.t. $Y = \varphi(Z)$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

Independent Noise Assumption (INA)

The random variable L related to the manipulation of Z equals $Y + B$, where Y is a function of Z and B is independent of Z.

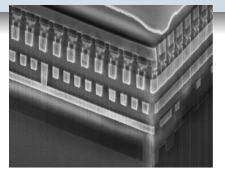
- B is usually called the noise and is viewed as a continuous random variable.
- We usually assume $B \sim \mathcal{N}(0, \sigma^2)$. (Gaussian Noise Assumption).
- Usually, we have $Z = S(X + K)$ where
  - X is known,
  - k is the secret to recover
  - $S(\cdot)$ is a known cryptographic primitive (*e.g. an s-box).*

New problem statement

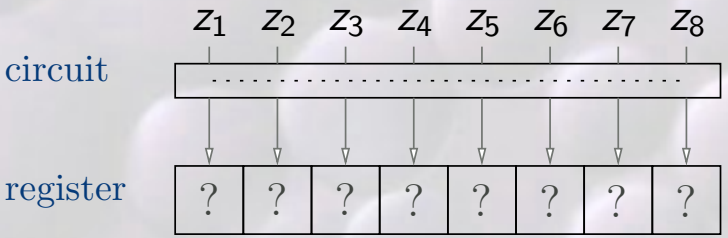
Modelling = recover the function φ s.t. $Y = \varphi(Z)$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.



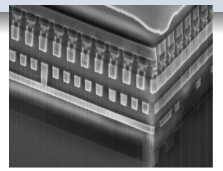
$$Z = S(X + k)$$



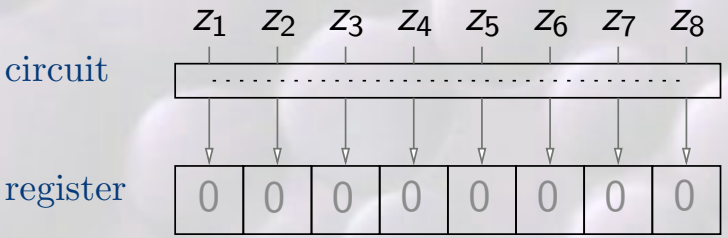
Assumption

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.



$$Z = S(X + k)$$

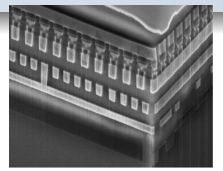


Assumption

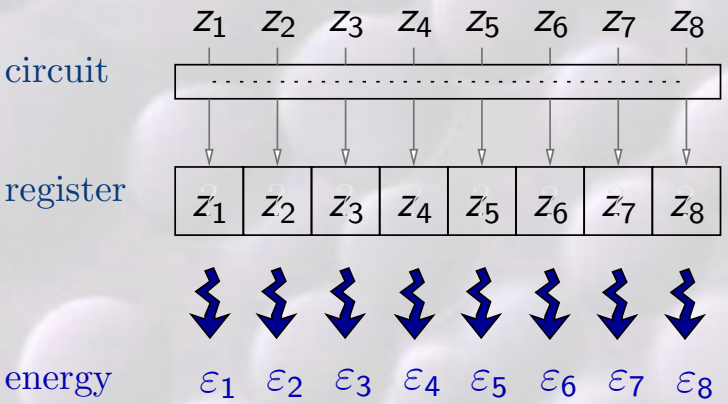
0 Precharge

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.



$$Z = S(X + k)$$



Assumption

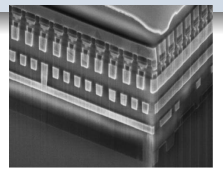
0 Precharge

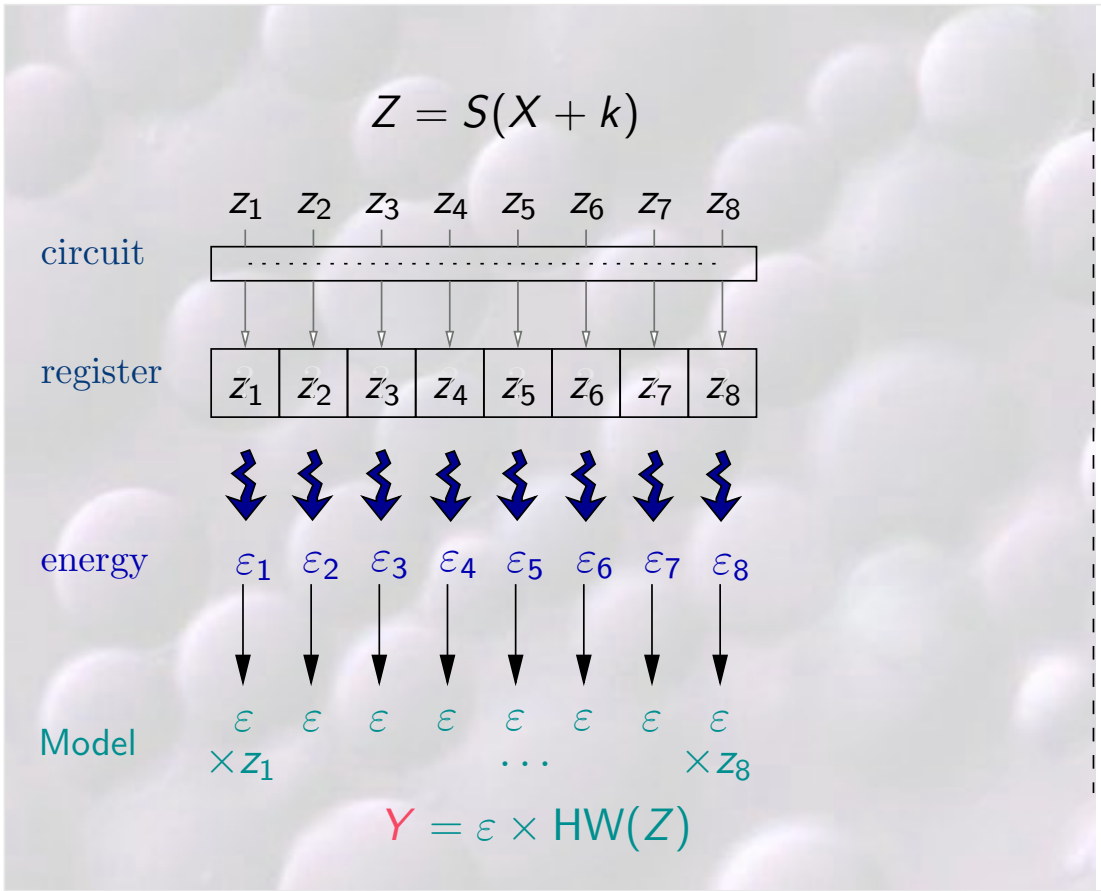
Linear Regression

the ε_i's are indep.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.





Assumption

0 Precharge

Linear Regression

the ε_i's are indep.

Hamming Weight Model

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

To sum up

- The deterministic part Y in a leakage L may be viewed as a multivariate polynomial in the bit-coordinate z_i of Z with coefficients in $\mathbb{R}$.
  - $\varphi(Z)$ is a polynomial in $\mathbb{R}[z_1, \cdots, z_n]$ and this polynomial is *a priori* unknown to the adversary.
- The modelling problem hence reduces to a problem of polynomial interpolation in noisy context:
  - from noisy observations of $\varphi(Y)$, we want to recover the coefficients ε_0, ε_1, ... such that:

$$\varphi(Z) = \underbrace{\varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots}_{\text{linear part}} + \underbrace{\varepsilon_{0,1} z_0 z_1 + \varepsilon_{0,2} z_0 z_2 + \ldots}_{\text{quadratic part}} + \underbrace{\ldots}_{\text{etc.}}$$

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

To sum up

- The deterministic part Y in a leakage L may be viewed as a multivariate polynomial in the bit-coordinate z_i of Z with coefficients in $\mathbb{R}$.
  - $\varphi(Z)$ is a polynomial in $\mathbb{R}[z_1, \cdots, z_n]$ and this polynomial is *a priori* unknown to the adversary.
- The modelling problem hence reduces to a problem of polynomial interpolation in noisy context:
  - from noisy observations of $\varphi(Y)$, we want to recover the coefficients ε_0, ε_1, ... such that:

$$\varphi(Z) = \underbrace{\varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots}_{\text{linear part}} + \underbrace{\varepsilon_{0,1} z_0 z_1 + \varepsilon_{0,2} z_0 z_2 + \ldots}_{\text{quadratic part}} + \underbrace{\ldots}_{\text{etc.}}$$

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

To sum up

- The deterministic part Y in a leakage L may be viewed as a multivariate polynomial in the bit-coordinate z_i of Z with coefficients in $\mathbb{R}$.
  - $\varphi(Z)$ is a polynomial in $\mathbb{R}[z_1, \cdots, z_n]$ and this polynomial is *a priori* unknown to the adversary.
- The modelling problem hence reduces to a problem of polynomial interpolation in noisy context:
  - from noisy observations of $\varphi(Y)$, we want to recover the coefficients ε_0, ε_1, ... such that:

$$\varphi(Z) = \underbrace{\varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots}_{\text{linear part}} + \underbrace{\varepsilon_{0,1} z_0 z_1 + \varepsilon_{0,2} z_0 z_2 + \ldots}_{\text{quadratic part}} + \underbrace{\ldots}_{\text{etc.}}$$

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

$$\varphi(Z) = \varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots$$

To sum up

- The polynomial interpolation with noise problem is usually solved thanks to linear regression techniques. See Schindler et al. at CHES 2005 or Doget et al at JCEN 2011.
- Usually, we assume the polynomial $\varphi(Z)$ is of degree 1.
- All the coefficients ε_i for degree-1 monomials are equal (to 1).
- The latter assumption (called Hamming Weight) is today pertinent for almost all smart card technologies.
- For recent ones (*e.g.* 65nm tech.), the non-linear terms must be taken into account. See Veyrat-Charvillon et al's paper at CRYPTO 2011.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

$$\varphi(Z) = \varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots$$

To sum up

- The polynomial interpolation with noise problem is usually solved thanks to linear regression techniques. See Schindler et al. at CHES 2005 or Doget et al at JCEN 2011.
- Usually, we assume the polynomial $\varphi(Z)$ is of degree 1.
- All the coefficients ε_i for degree-1 monomials are equal (to 1).
- The latter assumption (called Hamming Weight) is today pertinent for almost all smart card technologies.
- For recent ones (*e.g.* 65nm tech.), the non-linear terms must be taken into account. See Veyrat-Charvillon et al's paper at CRYPTO 2011.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

$$\varphi(Z) = \varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots$$

To sum up

- The polynomial interpolation with noise problem is usually solved thanks to linear regression techniques. See Schindler et al. at CHES 2005 or Doget et al at JCEN 2011.
- Usually, we assume the polynomial $\varphi(Z)$ is of degree 1.
- All the coefficients ε_i for degree-1 monomials are equal (to 1).
- The latter assumption (called Hamming Weight) is today pertinent for almost all smart card technologies.
- For recent ones (*e.g.* 65nm tech.), the non-linear terms must be taken into account. See Veyrat-Charvillon et al's paper at CRYPTO 2011.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

$$\varphi(Z) = \varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots$$

To sum up

- The polynomial interpolation with noise problem is usually solved thanks to linear regression techniques. See Schindler et al. at CHES 2005 or Doget et al at JCEN 2011.
- Usually, we assume the polynomial $\varphi(Z)$ is of degree 1.
- All the coefficients ε_i for degree-1 monomials are equal (to 1).
- The latter assumption (called Hamming Weight) is today pertinent for almost all smart card technologies.
- For recent ones (*e.g.* 65nm tech.), the non-linear terms must be taken into account. See Veyrat-Charvillon et al's paper at CRYPTO 2011.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

$$\varphi(Z) = \varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots$$

To sum up

- The polynomial interpolation with noise problem is usually solved thanks to linear regression techniques. See Schindler et al. at CHES 2005 or Doget et al at JCEN 2011.
- Usually, we assume the polynomial $\varphi(Z)$ is of degree 1.
- All the coefficients ε_i for degree-1 monomials are equal (to 1).
- The latter assumption (called Hamming Weight) is today pertinent for almost all smart card technologies.
- For recent ones (*e.g.* 65nm tech.), the non-linear terms must be taken into account. See Veyrat-Charvillon et al's paper at CRYPTO 2011.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: modelling for unprofiled attacks.

$$L \leftarrow Y + B = \varphi(Z) + B$$

$$\varphi(Z) = \varepsilon_0 z_0 + \varepsilon_1 z_1 + \ldots$$

To sum up

- The polynomial interpolation with noise problem is usually solved thanks to linear regression techniques. See Schindler et al. at CHES 2005 or Doget et al at JCEN 2011.
- Usually, we assume the polynomial $\varphi(Z)$ is of degree 1.
- All the coefficients ε_i for degree-1 monomials are equal (to 1).
- The latter assumption (called Hamming Weight) is today pertinent for almost all smart card technologies.
- For recent ones (*e.g.* 65nm tech.), the non-linear terms must be taken into account. See Veyrat-Charvillon et al's paper at CRYPTO 2011.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) \quad .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function HW.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathrm{HW}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: General Framework (Theoretical)

Context: attack during the manipulation of $S(X + k)$.

1. Measurement :
   - get a leakages sample $(\ell_{k,i})_i$ related to a sample $(x_i)_i$ of plaintexts.
2. Model Selection :
   - Design/Select a function HW.
3. Prediction :
   - For every $\hat{k}$, compute $m_{\hat{k},i} = \mathrm{HW}(S(x_i + \hat{k}))$.
4. Distinguisher Selection :
   - Choose a statistical distinguisher Δ.
5. Key Discrimination :
   - For every $\hat{k}$, compute the distinguishing value $\Delta_{\hat{k}}$:

$$\Delta_{\hat{k}} = \Delta\left((\ell_{k,i})_i, (m_{\hat{k},i})_i\right) .$$

6. Key Candidate Selection :
   - Deduce $\hat{k}$ from all the values $\Delta_{\hat{k}}$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Under INA assumption, the pdf f_L of L is a Gaussian Mixture:

$$f_L(\ell) = \sum_i \Pr[\varphi(Z) = i] \times \mathcal{N}(i, \sigma^2)$$

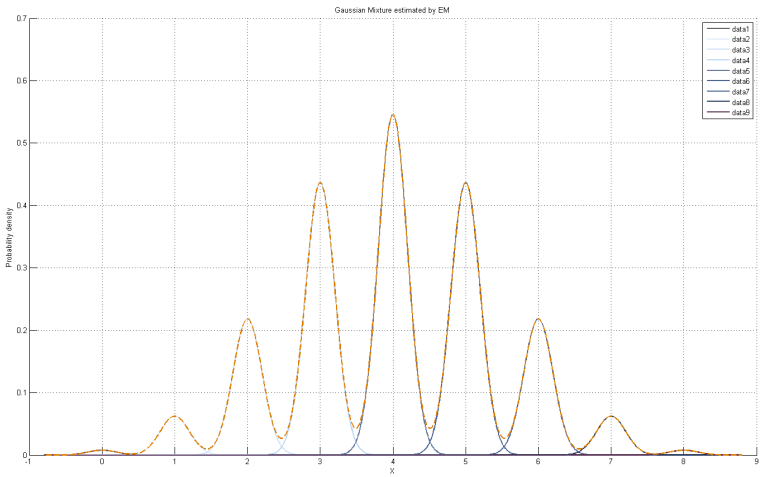


Figure: No noise ($\sigma = 0.2$)

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Under INA assumption, the pdf f_L of L is a Gaussian Mixture:

$$f_L(\ell) = \sum_i \Pr[\varphi(Z) = i] \times \mathcal{N}(i, \sigma^2)$$

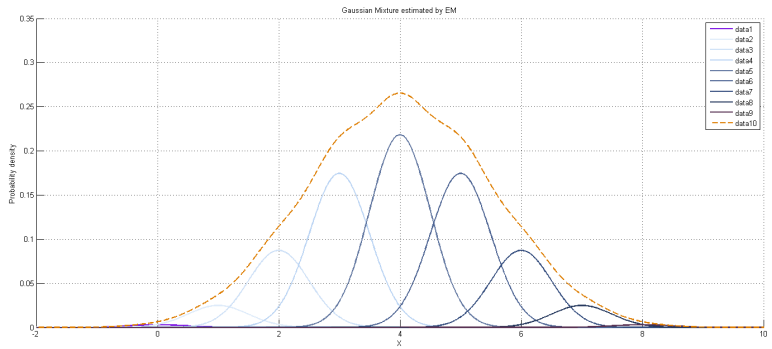


Figure: Small noise ($\sigma = 0.5$)

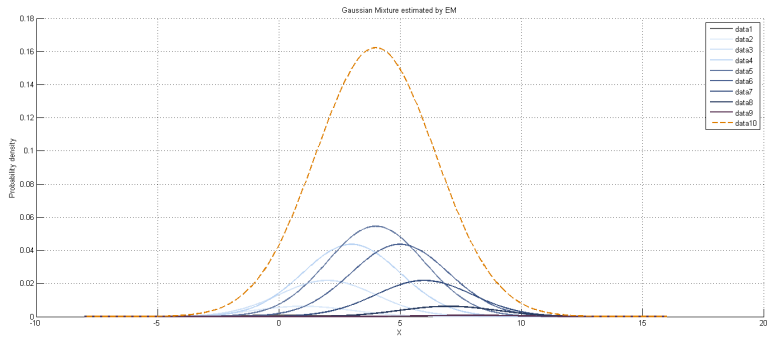


Figure: Medium noise ($\sigma = 2$)

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Question: which property of this mixture depends on the secret k?
Note: difficult question since the adversary does not know φ but a model $\mathbf{m}$ for it!
Many proposals have been done in the literature:

- DPA Kocher et al at CRYPTO 96,
- Multi-bit DPA Messerges in his PhD Thesis,
- CPA Brier et al at CHES 2004,
- Stochastic Attacks Schindler et al at CHES 2006
- or the MIA Gierlichs et al at CHES 2008.
- etc.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Question: which property of this mixture depends on the secret k?

Note: difficult question since the adversary does not know φ but a model $\mathbf{m}$ for it!

Many proposals have been done in the literature:

- DPA Kocher et al at CRYPTO 96,
- Multi-bit DPA Messerges in his PhD Thesis,
- CPA Brier et al at CHES 2004,
- Stochastic Attacks Schindler et al at CHES 2006
- or the MIA Gierlichs et al at CHES 2008.
- etc.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Question: which property of this mixture depends on the secret k?
Note: difficult question since the adversary does not know φ but a model $\mathbf{m}$ for it!
Many proposals have been done in the literature:

- DPA Kocher et al at CRYPTO 96,
- Multi-bit DPA Messerges in his PhD Thesis,
- CPA Brier et al at CHES 2004,
- Stochastic Attacks Schindler et al at CHES 2006
- or the MIA Gierlichs et al at CHES 2008.
- etc.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Question: which property of this mixture depends on the secret k?
Note: difficult question since the adversary does not know φ but a model $\mathbf{m}$ for it!
Many proposals have been done in the literature:

- DPA Kocher et al at CRYPTO 96,
- Multi-bit DPA Messerges in his PhD Thesis,
- CPA Brier et al at CHES 2004,
- Stochastic Attacks Schindler et al at CHES 2006
- or the MIA Gierlichs et al at CHES 2008.
- etc.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

DPA attack Kocher et al at CRYPTO 96.

Attack Description Sheet/Form: DPA

Type of Leakage: no restriction.
Model Function: the function $\mathbf{m} : Z \mapsto z_i$ for some index i.
Statistical Distinguisher: difference of means Test.
Key Candidate Selection: the candidate the maximizes the scores.

Score value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \mathbf{E}(L \mid M_{\hat{k}} = 1) - \mathbf{E}(L \mid M_{\hat{k}} = 0)$$

with $M_{\hat{k}}$ equal to the ith bit of $Z = S(X + \hat{k})$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

DPA attack Kocher et al at CRYPTO 96.

Attack Description Sheet/Form: DPA

Type of Leakage: no restriction.
Model Function: the function $\mathbf{m} : Z \mapsto z_i$ for some index i.
Statistical Distinguisher: difference of means Test.
Key Candidate Selection: the candidate the maximizes the scores.

Score value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \mathbf{E}(L \mid M_{\hat{k}} = 1) - \mathbf{E}(L \mid M_{\hat{k}} = 0)$$

with $M_{\hat{k}}$ equal to the ith bit of $Z = S(X + \hat{k})$.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

DPA attack Kocher et al at CRYPTO 96. Why does it work?

$$\begin{aligned}\Delta_{\hat{k}} &= \mathbf{E}(L \mid M_{\hat{k}} = 1) - \mathbf{E}(L \mid M_{\hat{k}} = 0) \\ &= \mathbf{E}(\varphi(Z) + B \mid M_{\hat{k}} = 1) - \mathbf{E}(\varphi(Z) + B \mid M_{\hat{k}} = 0)\end{aligned}$$

Since the noise B is independent of Z,

$$\begin{aligned}\Delta_{\hat{k}} &= \mathbf{E}(\varphi(Z) \mid M_{\hat{k}} = 1) - \mathbf{E}(\varphi(Z) \mid M_{\hat{k}} = 0) \\ = \mathbf{E}(\varepsilon_i z_i + (\varphi(Z) - \varepsilon_i z_i) \mid M_{\hat{k}} = 1) &- \mathbf{E}(\varepsilon_i z_i + (\varphi(Z) - \varepsilon_i z_i) \mid M_{\hat{k}} = 0)\end{aligned}$$

Let us assume that $(\varphi(Z) - \varepsilon_i z_i)$ is independent of z_i and $M_{\hat{k}}$ (true in practice).

$$\Delta_{\hat{k}} = \varepsilon_i \left(\mathbf{E}(z_i \mid M_{\hat{k}} = 1) - \mathbf{E}(z_i \mid M_{\hat{k}} = 0)\right)$$

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

$$\Delta_{\hat{k}} = \varepsilon_i \left(\mathbf{E}(z_i \mid M_{\hat{k}} = 1) - \mathbf{E}(z_i \mid M_{\hat{k}} = 0) \right)$$

where

- z_i is the ith bit of $S(M + k)$
- $M_{\hat{k}}$ is the ith bit of $S(M + \hat{k})$

If $k = \hat{k}$, then $z_i = M_{\hat{k}}$ and :

$$\Delta_{\hat{k}} = \varepsilon_i \left(1 - 0\right) = \varepsilon_i$$

If $k = \hat{k}$, then z_i and $M_{\hat{k}}$ are independent (due to properties of S) and

$$\Delta_{\hat{k}} = \varepsilon_i \left(\mathbf{E}(z_i) - \mathbf{E}(z_i) \right) = 0$$

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

DPA attack Kocher et al at CRYPTO 96.

- Pros: no need for assumption on the device properties, quite efficient in practice.
- Cons: does not use all the information in the trace and attack each bit of the target separately.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Multi-bit DPA attack Messerges in his PhD Thesis.

Attack Description Sheet/Form: Multi-bit DPA

Type of Leakage: no restriction.
Model Function **m**: the Hamming weight function.
Statistical Distinguisher: difference of means for a parameter τ.
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \mathbf{E}(L \mid M_{\hat{k}} \leq \tau) - \mathbf{E}(L \mid M_{\hat{k}} > \tau)$$

with $M_{\hat{k}}$ equal to the HW$[S(X + \hat{k})]$.

- Pros: exploit more information than the DPA.
- Cons: need assumption (Hamming weight) on the device behaviour.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Multi-bit DPA attack Messerges in his PhD Thesis.

Attack Description Sheet/Form: Multi-bit DPA

Type of Leakage: no restriction.
Model Function **m**: the Hamming weight function.
Statistical Distinguisher: difference of means for a parameter τ.
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \mathbf{E}(L \mid M_{\hat{k}} \leq \tau) - \mathbf{E}(L \mid M_{\hat{k}} > \tau)$$

with $M_{\hat{k}}$ equal to the HW$[S(X + \hat{k})]$.

- Pros: exploit more information than the DPA.
- Cons: need assumption (Hamming weight) on the device behaviour.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

Multi-bit DPA attack Messerges in his PhD Thesis.

Attack Description Sheet/Form: Multi-bit DPA

Type of Leakage: no restriction.
Model Function **m**: the Hamming weight function.
Statistical Distinguisher: difference of means for a parameter τ.
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \mathbf{E}(L \mid M_{\hat{k}} \leq \tau) - \mathbf{E}(L \mid M_{\hat{k}} > \tau)$$

with $M_{\hat{k}}$ equal to the HW$[S(X + \hat{k})]$.

- Pros: exploit more information than the DPA.
- Cons: need assumption (Hamming weight) on the device behaviour.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

CPA attack Brier et al at CHES 2004.

Attack Description Sheet/Form: CPA

Type of Leakage: no restriction.
Model Function **m**: possibly any function (in practice HW).
Statistical Distinguisher: linear correlation coefficient.
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \rho(L, M_{\hat{k}})$$

- Pros: exploit more information than the previous ones and is more powerful
- Cons: need assumption (Hamming weight) on the device behaviour.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

CPA attack Brier et al at CHES 2004.

Attack Description Sheet/Form: CPA

Type of Leakage: no restriction.
Model Function **m**: possibly any function (in practice HW).
Statistical Distinguisher: linear correlation coefficient.
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \rho(L, M_{\hat{k}})$$

- Pros: exploit more information than the previous ones and is more powerful
- Cons: need assumption (Hamming weight) on the device behaviour.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

CPA attack Brier et al at CHES 2004.

Attack Description Sheet/Form: CPA

Type of Leakage: no restriction.
Model Function **m**: possibly any function (in practice HW).
Statistical Distinguisher: linear correlation coefficient.
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \rho(L, M_{\hat{k}})$$

- Pros: exploit more information than the previous ones and is more powerful
- Cons: need assumption (Hamming weight) on the device behaviour.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

MIA attack Gierlichs et al at CHES 2008.

Attack Description Sheet/Form: MIA

Type of Leakage: no restriction.
Model Function **m**: any non-injective function (in practice HW).
Statistical Distinguisher: mutual information (MI).
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = MI(L; M_{\hat{k}}) = \text{entropy}(L) - \text{entropy}(L \mid M_{\hat{k}})$$

- Pros: theoretically able to detect any kind of dependency whatever the quality of the model if the function $x \mapsto \mathbf{m} \circ S(x + k)$ is non-injective!
- Cons: need for efficient estimators of the entropy (currently less efficient than the CPA) Batina *et al*, *Journal of Cryptology 2011*.

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: the statistical distinguisher

MIA attack Gierlichs et al at CHES 2008.

Attack Description Sheet/Form: MIA

Type of Leakage: no restriction.
Model Function **m**: any non-injective function (in practice HW).
Statistical Distinguisher: mutual information (MI).
Key Candidate Selection: the candidate the maximizes the scores.

Distinguishing value $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = MI(L; M_{\hat{k}}) = \text{entropy}(L) - \text{entropy}(L \mid M_{\hat{k}})$$

- Pros: theoretically able to detect any kind of dependency whatever the quality of the model if the function $x \mapsto \mathbf{m} \circ S(x + k)$ is non-injective!
- Cons: need for efficient estimators of the entropy (currently less efficient than the CPA) Batina *et al, Journal of Cryptology 2011*.

Advanced Side Channel Attacks (DPA like attacks)

Other attacks

- Stochastic attacks: See Schindler et al. at CHES 2005 or Doget et al at JCEN 2011.
  - Good alternative when classical (*e.g. HW) models fail.*
  - *Amounts to process an Euclidean distance between the leakage values and the estimations in the regressed model.*
- Kolmogorov-Smirnov Based attacks: Whitnall et al. at CARDIS 2011.
  - Good alternative to the MIA.
- PPA, EPA, VPA, etc: other attacks exist **but are often very ad hoc ones with no clear advantage to the "classical" ones**.
- Works comparing the attacks:
  - "How to Compare Profiled Side-Channel Attacks?" Standaert *et al, ACNS 2009*.
  - "A fair evaluation framework for comparing side-channel distinguishers" by Withnall *et al, JCEN 2011*.
  - "Univariate Side Channel Attacks and Leakage Modeling" by Doget *et al, JCEN 2011*.

Regression Linear Based (a.k.a. stochastic) Attacks

Attack Description Sheet/Form: **Linear Regression**

Type of Leakage: no restriction.
Model Function: a set of basis functions $\mathrm{m}^{(i)}(\cdot,\cdot)$ s.t. φ can be approximated as linear combination of them.
Statistical Distinguisher: Euclidean Distance (a.k.a. sum of squares difference).
Key Candidate Selection: the candidate that maximizes the goodness of fit coefficient.

Regression Linear Based (a.k.a. stochastic) Attacks

- Goal: recover Z or at least a good approximation of it!
- Note: ... from noisy observations L of Z and from the corresponding plaintexts X.
- Idea: for each $\hat{k}$ find the best approximation $\hat{Z}$ of Z as a linear combination of the variables $\mathrm{m}^{(i)}(X, \hat{k})$.
  - If the distance $||Z - \hat{Z}||_2$ is small $\Longrightarrow \hat{k} = k$...
  - otherwise $\hat{k} \neq k$.

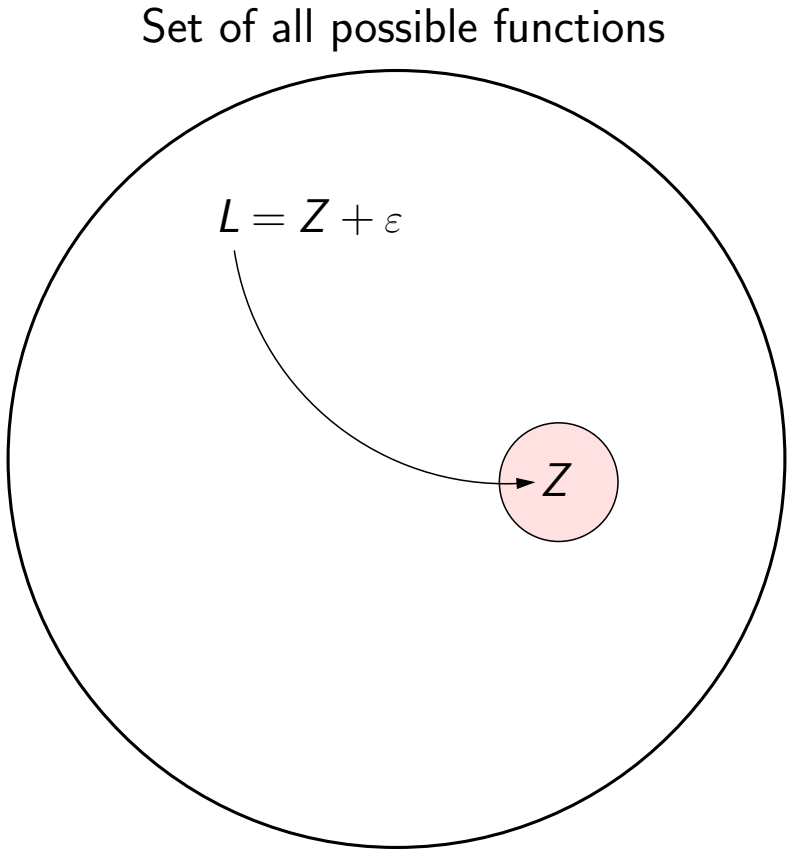
Regression Linear Based (a.k.a. stochastic) Attacks

- Goal: recover Z or at least a good approximation of it!
- Note: ... from noisy observations L of Z and from the corresponding plaintexts X.
- Idea: for each $\hat{k}$ find the best approximation $\hat{Z}$ of Z as a linear combination of the variables $\mathrm{m}^{(i)}(X, \hat{k})$.
  - If the distance $||Z - \hat{Z}||_2$ is small $\Longrightarrow \hat{k} = k$...
  - otherwise $\hat{k} \neq k$.

Regression Linear Based (a.k.a. stochastic) Attacks

- Goal: recover Z or at least a good approximation of it!
- Note: ... from noisy observations L of Z and from the corresponding plaintexts X.
- Idea: for each $\hat{k}$ find the best approximation $\hat{Z}$ of Z as a linear combination of the variables $\mathrm{m}^{(i)}(X, \hat{k})$.
  - If the distance $||Z - \hat{Z}||_2$ is small $\Longrightarrow \hat{k} = k$...
  - otherwise $\hat{k} \neq k$.

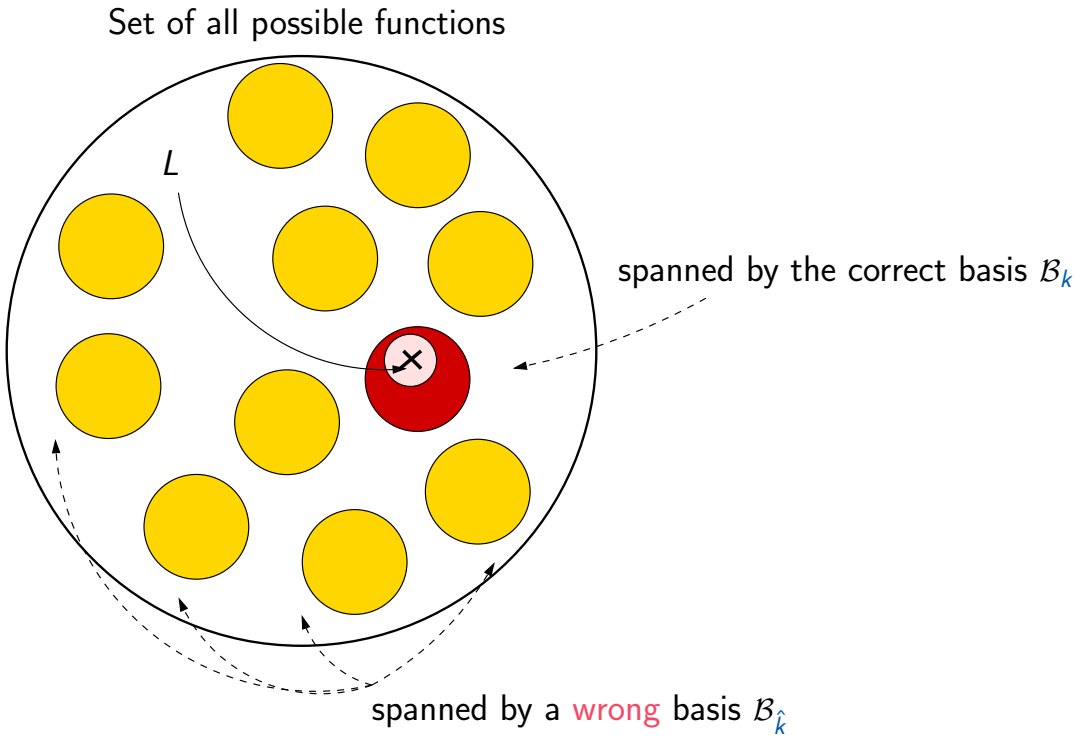
Regression Linear Based (a.k.a. stochastic) Attacks



- For each $\hat{k}$, define a basis $\mathcal{B}_{\hat{k}} = (\mathrm{m}^{(i)}(X, \hat{k}))_i$.
- Compute the distance between the e.v. spanned $\mathcal{B}_{\hat{k}}$ and L
- Choose the key that minimizes the distance.

Regression Linear Based (a.k.a. stochastic) Attacks

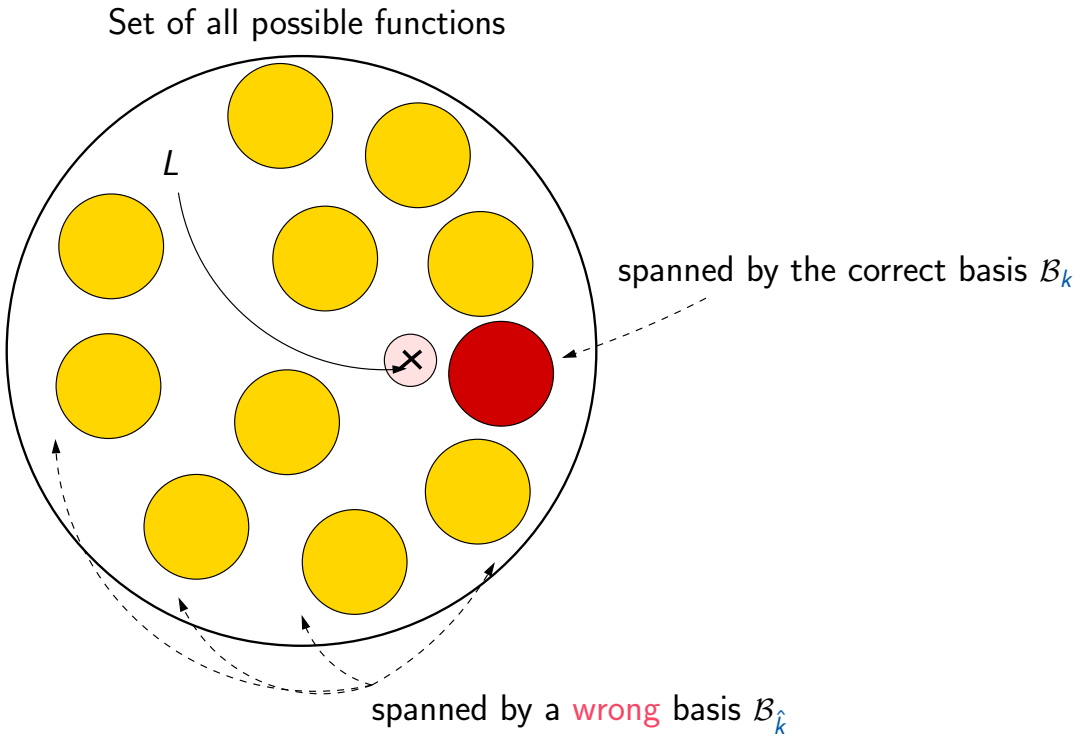
Distance null for the correct hypothesis!



- For each $\hat{k}$, define a basis $\mathcal{B}_{\hat{k}} = (\mathrm{m}^{(i)}(X, \hat{k}))_i$.
- Compute the distance between the e.v. spanned $\mathcal{B}_{\hat{k}}$ and L
- Choose the key that minimizes the distance.

Regression Linear Based (a.k.a. stochastic) Attacks

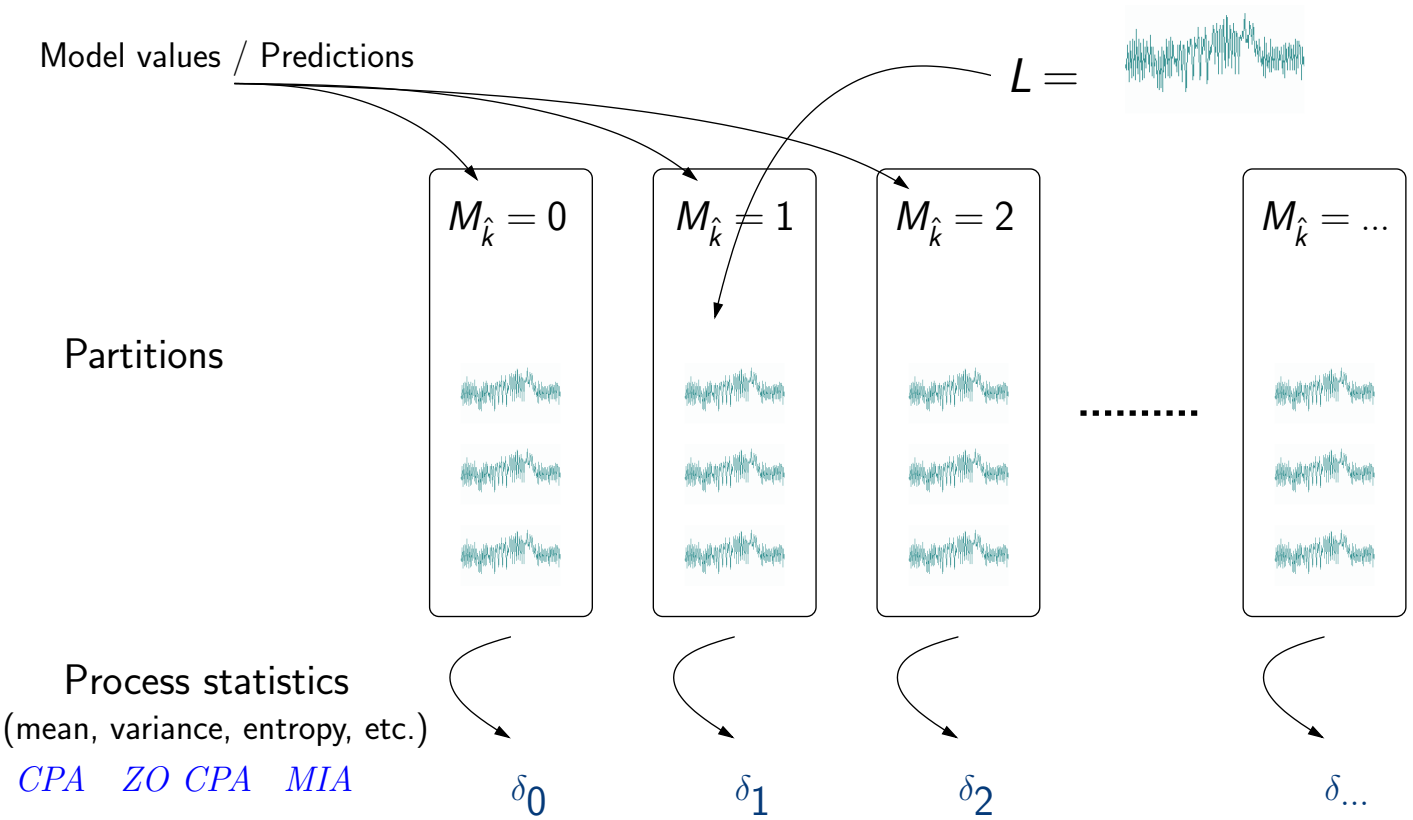
Distance small for the correct hypothesis!



- For each $\hat{k}$, define a basis $\mathcal{B}_{\hat{k}} = (\mathrm{m}^{(i)}(X, \hat{k}))_i$.
- Compute the distance between the e.v. spanned $\mathcal{B}_{\hat{k}}$ and L
- Choose the key that minimizes the distance.

Distinguishers Processing

... a partitioning description.



Combine the statistics

$$\Delta_{\hat{k}} = \sum_i \delta_i \times \mathrm{P}[M_{\hat{k}} = i]$$

Advanced Side Channel Attacks (DPA like attacks)

Attack Efficiency Consideration

Attack Efficiency

The efficiency of an SCA given a success rate β is the smallest value N such that:

$$\Pr(\text{Attack succeeds in recovering } k \text{ with } N \text{ measurements}) \geq \beta \ .$$

Particular case: the attack involves correlation coefficient ($i.e. \Delta = \rho$):

$$\Pr\left(\hat{\rho}_k(N) > \max_{\hat{k} \neq k} \hat{\rho}_{\hat{k}}(N)\right) \geq \beta \ .$$

where $\hat{\rho}_k(N)$ denotes the estimation of ρ_k based on N.

Advanced Side Channel Attacks (DPA like attacks)

Attack Efficiency Consideration

Attack Efficiency

The efficiency of an SCA given a success rate β is the smallest value N such that:

$$\Pr(\text{Attack succeeds in recovering } k \text{ with } N \text{ measurements}) \geq \beta \ .$$

Particular case: the attack involves correlation coefficient ($i.e. \Delta = \rho$):

$$\Pr\left(\hat{\rho}_k(N) > \max_{\hat{k} \neq k} \hat{\rho}_{\hat{k}}(N)\right) \geq \beta \ .$$

where $\hat{\rho}_k(N)$ denotes the estimation of ρ_k based on N.

Advanced Side Channel Attacks (DPA like attacks)

Attack Efficiency Consideration

Attack Efficiency

The efficiency of an SCA given a success rate β is the smallest value N such that:

$$\Pr(\text{Attack succeeds in recovering } k \text{ with } N \text{ measurements}) \geq \beta \ .$$

Particular case: the attack involves correlation coefficient (*i.e.*$\Delta = \rho$):

$$\Pr\left(\hat{\rho}_k(N) > \max_{\hat{k} \neq k} \hat{\rho}_{\hat{k}}(N)\right) \geq \beta \ .$$

where $\hat{\rho}_k(N)$ denotes the estimation of ρ_k based on N.

Advanced Side Channel Attacks (DPA like attacks)

Attack Efficiency Consideration

- Fisher: when $\hat{\rho}_{\hat{k}}(N)$ is computed between samples that have a joint normal distribution, $Z_{N,\hat{k}} = \frac{1}{2} \ln \left(\frac{1+\hat{\rho}_k(N)}{1-\hat{\rho}_{\hat{k}}(N)} \right)$ has a normal distribution with parameters

$$\mathbf{E}(Z_{N,\hat{k}}) = \frac{1}{2} \ln \left(\frac{1+\rho_k}{1-\rho_{\hat{k}}} \right) \text{ and } \mathrm{Var}(Z_{N,\hat{k}}) = (N-3)^{-2}.$$

- [Mangard at CT-RSA 2004] So, $\Pr(\hat{\rho}_k(N) > \hat{\rho}_{\hat{k}}(N)) = \beta$ implies:

$$N = 3 + 8 \left(\frac{\Phi^{-1}(\beta)}{\ln \left(\frac{1+\rho_k}{1-\rho_{\hat{k}}} \right)} \right)^2,$$

where Φ denotes the pdf of $\mathcal{N}(0,1)$.

Advanced Side Channel Attacks (DPA like attacks)

Attack Efficiency Consideration

- Fisher: when $\hat{\rho}_{\hat{k}}(N)$ is computed between samples that have a joint normal distribution, $Z_{N,\hat{k}} = \frac{1}{2} \ln \left(\frac{1+\hat{\rho}_k(N)}{1-\hat{\rho}_{\hat{k}}(N)} \right)$ has a normal distribution with parameters

$$\mathbf{E}(Z_{N,\hat{k}}) = \frac{1}{2} \ln \left(\frac{1+\rho_k}{1-\rho_{\hat{k}}} \right) \text{ and } \mathrm{Var}(Z_{N,\hat{k}}) = (N-3)^{-2}.$$

- [Mangard at CT-RSA 2004] So, $\Pr(\hat{\rho}_k(N) > \hat{\rho}_{\hat{k}}(N)) = \beta$ implies:

$$N = 3 + 8 \left(\frac{\Phi^{-1}(\beta)}{\ln \left(\frac{1+\rho_k}{1-\rho_{\hat{k}}} \right)} \right)^2,$$

where Φ denotes the pdf of $\mathcal{N}(0,1)$.

Advanced Side Channel Attacks (DPA like attacks)

Attack Efficiency Consideration

- Fisher: when $\hat{\rho}_{\hat{k}}(N)$ is computed between samples that have a joint normal distribution, $Z_{N,\hat{k}} = \frac{1}{2} \ln \left(\frac{1+\hat{\rho}_k(N)}{1-\hat{\rho}_{\hat{k}}(N)} \right)$ has a normal distribution with parameters

$$\mathbf{E}(Z_{N,\hat{k}}) = \frac{1}{2} \ln \left(\frac{1+\rho_k}{1-\rho_{\hat{k}}} \right) \text{ and } \mathrm{Var}(Z_{N,\hat{k}}) = (N-3)^{-2}.$$

- [Mangard at CT-RSA 2004] Assuming $\rho_{\hat{k}}(N) = 0$ we get:

$$N \approx 8 \times \Phi^{-1}(\beta)^2 \times \rho_k^{-2} \ ,$$

since $\ln(1+x) \approx x$ if $|x| < 1$.

Advanced Side Channel Attacks (DPA like attacks)

Link between Efficiency and Signal to Noise Ratio (SNR)

- Let us define the SNR by:

$$\text{SNR} = \frac{\mathbf{Var}[\mathbf{L}] - \mathbf{E}[\mathbf{Var}[\mathbf{L} \mid Z]]}{\mathbf{E}[\mathbf{Var}[\mathbf{L} \mid Z]]} = \frac{\mathbf{Var}[\varphi(Z)]}{\mathbf{E}[\mathbf{Var}[\mathbf{L} \mid Z]]}$$

  Note: can be computed without knowing φ!

- [Mangard at CT-RSA 2004] If $\text{SNR} \ll 1$, we have

$$\rho_{\hat{k}}(N) = \text{SNR} \times \rho^0_{\hat{k}}(N)$$

  where $\rho^0_{\hat{k}}(N)$ denotes the correl. when there is no stoch. noise.

- Consequently,

$$N \sim \frac{1}{SNR}$$

$SNR = 0.01$	$\rightarrow$	around 100 traces	$\rightarrow$	*few seconds*
$SNR = 0.001$	$\rightarrow$	around 1000 traces	$\rightarrow$	*less than $1/4$ hour*
$SNR = 0.0001$	$\rightarrow$	around $1O^5$ traces	$\rightarrow$	*several hours*
$SNR = 10^{-6}$	$\rightarrow$	several millions of traces	$\rightarrow$	*several days*

Advanced Side Channel Attacks (DPA like attacks)

Link between Efficiency and Signal to Noise Ratio (SNR)

- Let us define the SNR by:

$$\text{SNR} = \frac{\mathbf{Var[L]} - \mathbf{E[Var[L \mid Z]]}}{\mathbf{E[Var[L \mid Z]]}} = \frac{\mathbf{Var}[\varphi(Z)]}{\mathbf{E[Var[L \mid Z]]}}$$

Note: can be computed without knowing φ!

- [Mangard at CT-RSA 2004] If $\text{SNR} \ll 1$, we have

$$\rho_{\hat{k}}(N) = \text{SNR} \times \rho^0_{\hat{k}}(N)$$

where $\rho^0_{\hat{k}}(N)$ denotes the correl. when there is no stoch. noise.

- Consequently,

$$N \sim \frac{1}{SNR}$$

$SNR = 0.01$	$\rightarrow$	around 100 traces	$\rightarrow$	*few seconds*
$SNR = 0.001$	$\rightarrow$	around 1000 traces	$\rightarrow$	*less than $1/4$ hour*
$SNR = 0.0001$	$\rightarrow$	around $1O^5$ traces	$\rightarrow$	*several hours*
$SNR = 10^{-6}$	$\rightarrow$	several millions of traces	$\rightarrow$	*several days*

Advanced Side Channel Attacks (DPA like attacks)

Link between Efficiency and Signal to Noise Ratio (SNR)

- Let us define the SNR by:

$$\text{SNR} = \frac{\mathbf{Var[L]} - \mathbf{E[Var[L \mid Z]]}}{\mathbf{E[Var[L \mid Z]]}} = \frac{\mathbf{Var}[\varphi(Z)]}{\mathbf{E[Var[L \mid Z]]}}$$

Note: can be computed without knowing φ!

- [Mangard at CT-RSA 2004] If $\text{SNR} \ll 1$, we have

$$\rho_{\hat{k}}(N) = \text{SNR} \times \rho^0_{\hat{k}}(N)$$

where $\rho^0_{\hat{k}}(N)$ denotes the correl. when there is no stoch. noise.

- Consequently,

$$N \sim \frac{1}{SNR}$$

$SNR = 0.01$	$\rightarrow$	around 100 traces	$\rightarrow$	*few seconds*
$SNR = 0.001$	$\rightarrow$	around 1000 traces	$\rightarrow$	*less than $1/4$ hour*
$SNR = 0.0001$	$\rightarrow$	around $1O^5$ traces	$\rightarrow$	*several hours*
$SNR = 10^{-6}$	$\rightarrow$	several millions of traces	$\rightarrow$	*several days*

Advanced Side Channel Attacks (DPA like attacks)

Link between Efficiency and Signal to Noise Ratio (SNR)

- Let us define the SNR by:

$$\text{SNR} = \frac{\mathbf{Var}[\mathbf{L}] - \mathbf{E}[\mathbf{Var}[\mathbf{L} \mid Z]]}{\mathbf{E}[\mathbf{Var}[\mathbf{L} \mid Z]]} = \frac{\mathbf{Var}[\varphi(Z)]}{\mathbf{E}[\mathbf{Var}[\mathbf{L} \mid Z]]}$$

  Note: can be computed without knowing φ!

- [Mangard at CT-RSA 2004] If $\text{SNR} \ll 1$, we have

$$\rho_{\hat{k}}(N) = \text{SNR} \times \rho^0_{\hat{k}}(N)$$

  where $\rho^0_{\hat{k}}(N)$ denotes the correl. when there is no stoch. noise.

- Consequently,

$$N \sim \frac{1}{SNR}$$

$SNR = 0.01$	$\rightarrow$	around 100 traces	$\rightarrow$	*few seconds*
$SNR = 0.001$	$\rightarrow$	around 1000 traces	$\rightarrow$	*less than $1/4$ hour*
$SNR = 0.0001$	$\rightarrow$	around $1O^5$ traces	$\rightarrow$	*several hours*
$SNR = 10^{-6}$	$\rightarrow$	several millions of traces	$\rightarrow$	*several days*

Advanced Side Channel Attacks (DPA like attacks)

More accurate efficiency evaluations

- Core Idea: relax the assumption $\rho_{\hat{k}}(N) = 0$ for any $\hat{k} \neq 0$.
- Note: this assumption contradicts the ghost Peaks phenomenon ... which is however observed in practice!
- Recent works on this subject: Rivain, SAC 2008, Fei, Luo, Ding, CHES 2012, Thillard, Prouff, Roche, CHES 2013.

Advanced Side Channel Attacks (DPA like attacks)

More accurate efficiency evaluations

- Core Idea: relax the assumption $\rho_{\hat{k}}(N) = 0$ for any $\hat{k} \neq 0$.
- Note: this assumption contradicts the ghost Peaks phenomenon ... which is however observed in practice!
- Recent works on this subject: Rivain, SAC 2008, Fei, Luo, Ding, CHES 2012, Thillard, Prouff, Roche, CHES 2013.

Advanced Side Channel Attacks (DPA like attacks)

More accurate efficiency evaluations

- Core Idea: relax the assumption $\rho_{\hat{k}}(N) = 0$ for any $\hat{k} \neq 0$.
- Note: this assumption contradicts the ghost Peaks phenomenon ... which is however observed in practice!
- Recent works on this subject: Rivain, SAC 2008, Fei, Luo, Ding, CHES 2012, Thillard, Prouff, Roche, CHES 2013.

Advanced Side Channel Attacks (DPA like attacks)

More accurate efficiency evaluations

- Core Idea: relax the assumption $\rho_{\hat{k}}(N) = 0$ for any $\hat{k} \neq 0$.
- Note: this assumption contradicts the ghost Peaks phenomenon ... which is however observed in practice!
- Recent works on this subject: Rivain, SAC 2008, Fei, Luo, Ding, CHES 2012, Thillard, Prouff, Roche, CHES 2013.

Advanced Side Channel Attacks (DPA like attacks)

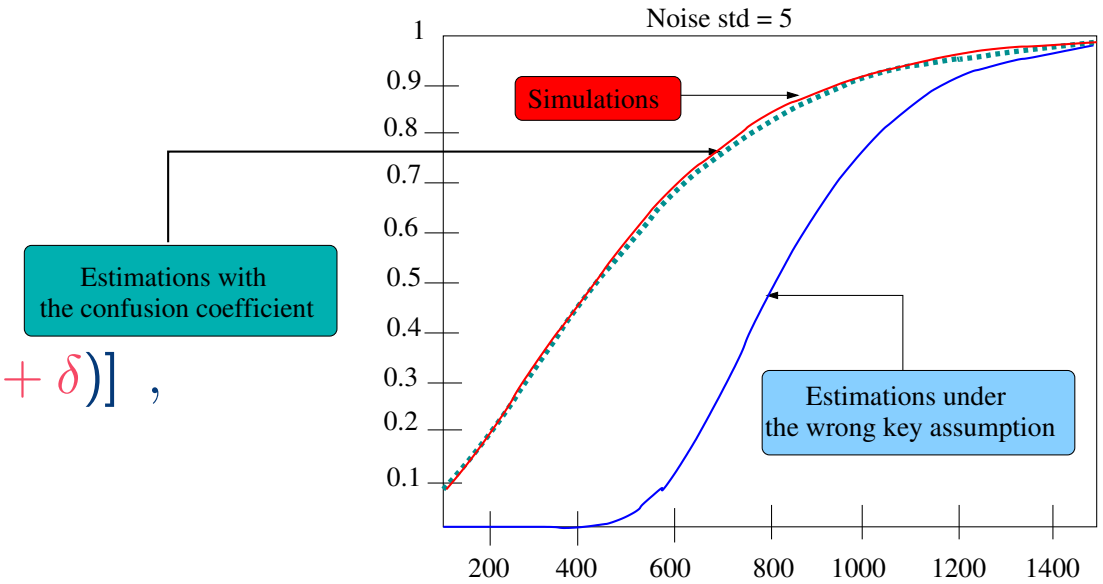
More accurate efficiency evaluations

- Core Idea: relax the assumption $\rho_{\hat{k}}(N) = 0$ for any $\hat{k} \neq 0$.
- Note: this assumption contradicts the ghost Peaks phenomenon ... which is however observed in practice!
- Recent works on this subject: Rivain, SAC 2008, Fei, Luo, Ding, CHES 2012, Thillard, Prouff, Roche, CHES 2013.

Use the notion of Confusion Coefficient, defined for every δ by:

$$\kappa_\delta = \mathbf{E}[\mathrm{m} \circ S(X+k) \times \mathrm{m} \circ S(X+k+\delta)] \ ,$$

where m is the model used in the attack.



Advanced Side Channel Attacks (DPA like attacks)

Efficiency of Other Attacks (MIA, Templates, etc.)

- When provided with the same *a priori* information about the leakage, CPA, MIA, DPA and Gaussian template attacks are asymptotically equivalent Mangard *et al*, *IET Information Security 2011*.
  - $\Longrightarrow$ Efficiency formula $N \approx 8 \times \Phi^{-1}(\beta)^2 \times \Delta_k^{-2}$ stays true for the corresponding distinguishers.
- Note: for Template attacks, the cost of the on-line phase may be constant but the cost of the off-line templates building will be linear in SNR^{-1}.
- In conclusion, adding security consists in finding efficient way(s) to decrease Δ_k as much as possible.
  - *i.e.*specify the algorithm implementation such that for any instantaneous leakage L, for any key part k and for any function g:

$$\Delta(L, g(X, k)) < \varepsilon \ ,$$

where X is some plaintext part and ε is a security parameter.

Advanced Side Channel Attacks (DPA like attacks)

Efficiency of Other Attacks (MIA, Templates, etc.)

- When provided with the same *a priori* information about the leakage, CPA, MIA, DPA and Gaussian template attacks are asymptotically equivalent Mangard *et al*, *IET Information Security 2011*.
  - $\Longrightarrow$ Efficiency formula $N \approx 8 \times \Phi^{-1}(\beta)^2 \times \Delta_k^{-2}$ stays true for the corresponding distinguishers.
- Note: for Template attacks, the cost of the on-line phase may be constant but the cost of the off-line templates building will be linear in SNR^{-1}.
- In conclusion, adding security consists in finding efficient way(s) to decrease Δ_k as much as possible.
  - *i.e.*specify the algorithm implementation such that for any instantaneous leakage L, for any key part k and for any function g:

$$\Delta(L, g(X, k)) < \varepsilon \ ,$$

where X is some plaintext part and ε is a security parameter.

Advanced Side Channel Attacks (DPA like attacks)

Efficiency of Other Attacks (MIA, Templates, etc.)

- When provided with the same *a priori* information about the leakage, CPA, MIA, DPA and Gaussian template attacks are asymptotically equivalent Mangard *et al, IET Information Security 2011*.
  - $\Longrightarrow$ Efficiency formula $N \approx 8 \times \Phi^{-1}(\beta)^2 \times \Delta_k^{-2}$ stays true for the corresponding distinguishers.
- Note: for Template attacks, the cost of the on-line phase may be constant but the cost of the off-line templates building will be linear in SNR^{-1}.
- In conclusion, adding security consists in finding efficient way(s) to decrease Δ_k as much as possible.
  - *i.e.*specify the algorithm implementation such that for any instantaneous leakage L, for any key part k and for any function g:

$$\Delta(L, g(X, k)) < \varepsilon \ ,$$

where X is some plaintext part and ε is a security parameter.

Advanced Side Channel Attacks (DPA like attacks)

Efficiency of Other Attacks (MIA, Templates, etc.)

- When provided with the same *a priori* information about the leakage, CPA, MIA, DPA and Gaussian template attacks are asymptotically equivalent Mangard *et al, IET Information Security 2011*.
  - $\Longrightarrow$ Efficiency formula $N \approx 8 \times \Phi^{-1}(\beta)^2 \times \Delta_k^{-2}$ stays true for the corresponding distinguishers.
- Note: for Template attacks, the cost of the on-line phase may be constant but the cost of the off-line templates building will be linear in SNR^{-1}.
- In conclusion, adding security consists in finding efficient way(s) to decrease Δ_k as much as possible.
  - *i.e.*specify the algorithm implementation such that for any instantaneous leakage L, for any key part k and for any function g:

$$\Delta(L, g(X, k)) < \varepsilon \ ,$$

where X is some plaintext part and ε is a security parameter.

Advanced Side Channel Attacks (DPA like attacks)

Efficiency of Other Attacks (MIA, Templates, etc.)

- When provided with the same *a priori* information about the leakage, CPA, MIA, DPA and Gaussian template attacks are asymptotically equivalent Mangard *et al, IET Information Security 2011*.
  - $\Longrightarrow$ Efficiency formula $N \approx 8 \times \Phi^{-1}(\beta)^2 \times \Delta_k^{-2}$ stays true for the corresponding distinguishers.
- Note: for Template attacks, the cost of the on-line phase may be constant but the cost of the off-line templates building will be linear in SNR^{-1}.
- In conclusion, adding security consists in finding efficient way(s) to decrease Δ_k as much as possible.
  - *i.e.*specify the algorithm implementation such that for any instantaneous leakage L, for any key part k and for any function g:

$$\Delta(L, g(X, k)) < \varepsilon \ ,$$

where X is some plaintext part and ε is a security parameter.

Advanced Side Channel Attacks (DPA like attacks)

Efficiency of Other Attacks (MIA, Templates, etc.)

- When provided with the same *a priori* information about the leakage, CPA, MIA, DPA and Gaussian template attacks are asymptotically equivalent Mangard *et al, IET Information Security 2011*.
  - $\Longrightarrow$ Efficiency formula $N \approx 8 \times \Phi^{-1}(\beta)^2 \times \Delta_k^{-2}$ stays true for the corresponding distinguishers.
- Note: for Template attacks, the cost of the on-line phase may be constant but the cost of the off-line templates building will be linear in SNR^{-1}.
- In conclusion, adding security consists in finding efficient way(s) to decrease Δ_k as much as possible.
  - *i.e.*specify the algorithm implementation such that for any instantaneous leakage L, for any key part k and for any function g:

$$\Delta(L, g(X, k)) < \varepsilon \ ,$$

where X is some plaintext part and ε is a security parameter.

Advanced Side Channel Attacks (DPA like attacks)

Efficiency of Other Attacks (MIA, Templates, etc.)

- When provided with the same *a priori* information about the leakage, CPA, MIA, DPA and Gaussian template attacks are asymptotically equivalent Mangard *et al, IET Information Security 2011*.
  - $\Longrightarrow$ Efficiency formula $N \approx 8 \times \Phi^{-1}(\beta)^2 \times \Delta_k^{-2}$ stays true for the corresponding distinguishers.
- Note: for Template attacks, the cost of the on-line phase may be constant but the cost of the off-line templates building will be linear in SNR^{-1}.
- In conclusion, adding security consists in finding efficient way(s) to decrease Δ_k as much as possible.
  - *i.e.*specify the algorithm implementation such that for any instantaneous leakage L, for any key part k and for any function g:

$$\Delta(L, g(X, k)) < \varepsilon \ ,$$

where X is some plaintext part and ε is a security parameter.

Plan

SCA Countermeasures

- Masking [IBM Team at CRYPTO 1999].
  - Efficient against SCA in practice.
  - Difficult to implement for non-linear transformations.
- Shuffling [Researchers from Graz University at ACNS 2006].
  - Less efficient against SCA in practice.
  - Easy to implement for every transformation.
- Whitening [Kocher Jaffe June, CRYPTO 1999].
  - Less efficient than masking when used alone and costly in Hardware.
  - Easy to implement for every transformation.

SCA Countermeasures

- Masking [IBM Team at CRYPTO 1999].
  - Efficient against SCA in practice.
  - Difficult to implement for non-linear transformations.
- Shuffling [Researchers from Graz University at ACNS 2006].
  - Less efficient against SCA in practice.
  - Easy to implement for every transformation.
- Whitening [Kocher Jaffe June, CRYPTO 1999].
  - Less efficient than masking when used alone and costly in Hardware.
  - Easy to implement for every transformation.

SCA Countermeasures

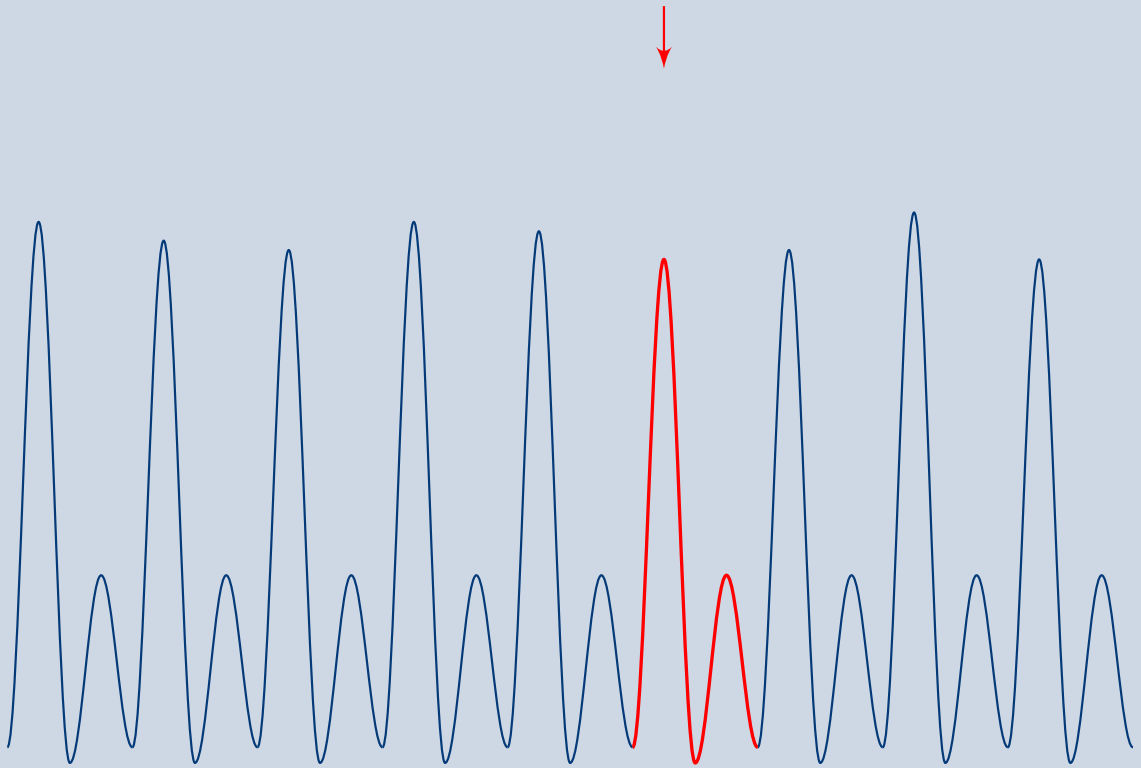
- Masking [IBM Team at CRYPTO 1999].
  - Efficient against SCA in practice.
  - Difficult to implement for non-linear transformations.
- Shuffling [Researchers from Graz University at ACNS 2006].
  - Less efficient against SCA in practice.
  - Easy to implement for every transformation.
- Whitening [Kocher Jaffe June, CRYPTO 1999].
  - Less efficient than masking when used alone and costly in Hardware.
  - Easy to implement for every transformation.

SCA Countermeasures

- Masking [IBM Team at CRYPTO 1999].
  - Efficient against SCA in practice.
  - Difficult to implement for non-linear transformations.
- Shuffling [Researchers from Graz University at ACNS 2006].
  - Less efficient against SCA in practice.
  - Easy to implement for every transformation.
- Whitening [Kocher Jaffe June, CRYPTO 1999].
  - Less efficient than masking when used alone and costly in Hardware.
  - Easy to implement for every transformation.

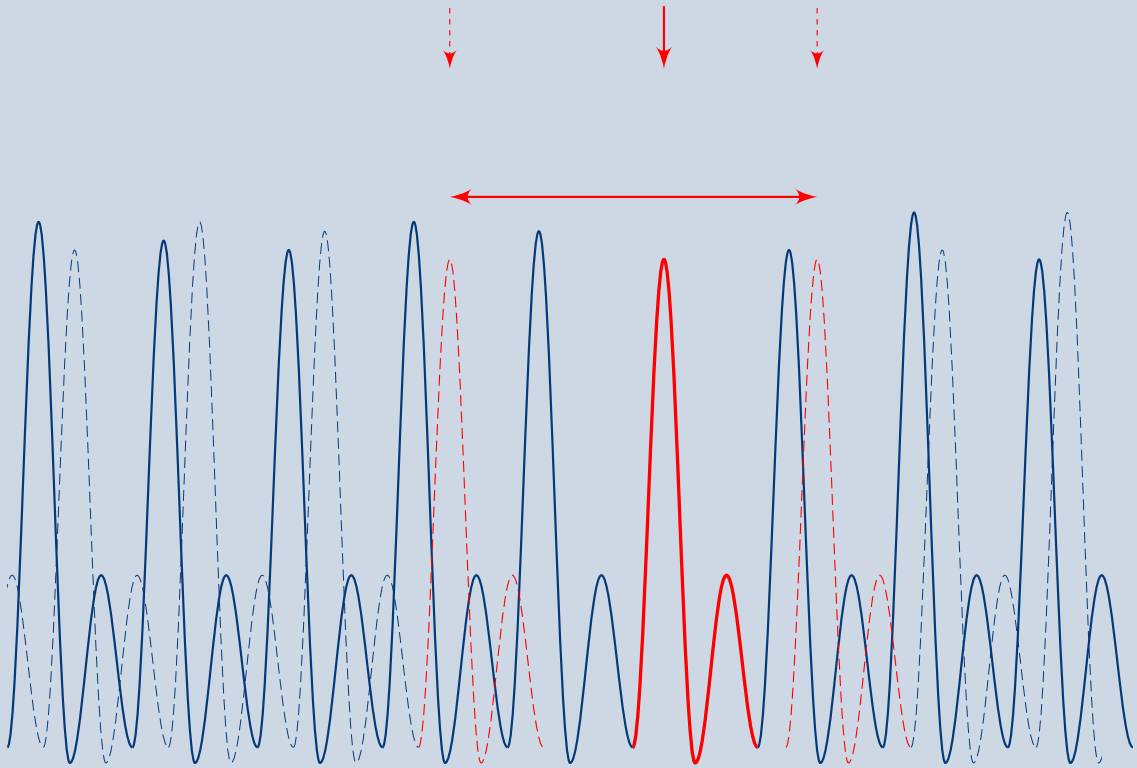
SCA Countermeasures

Core Ideas



SCA Countermeasures

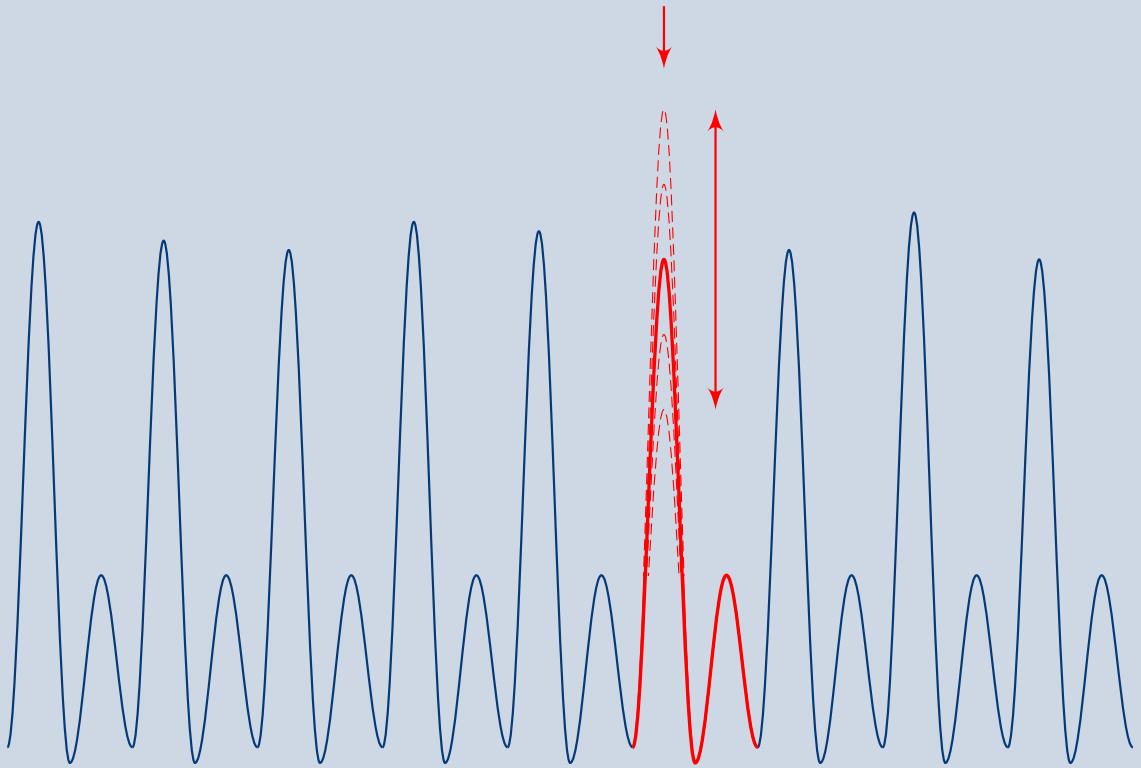
Core Ideas



Desynchronisation = different points, same amplitutde

SCA Countermeasures

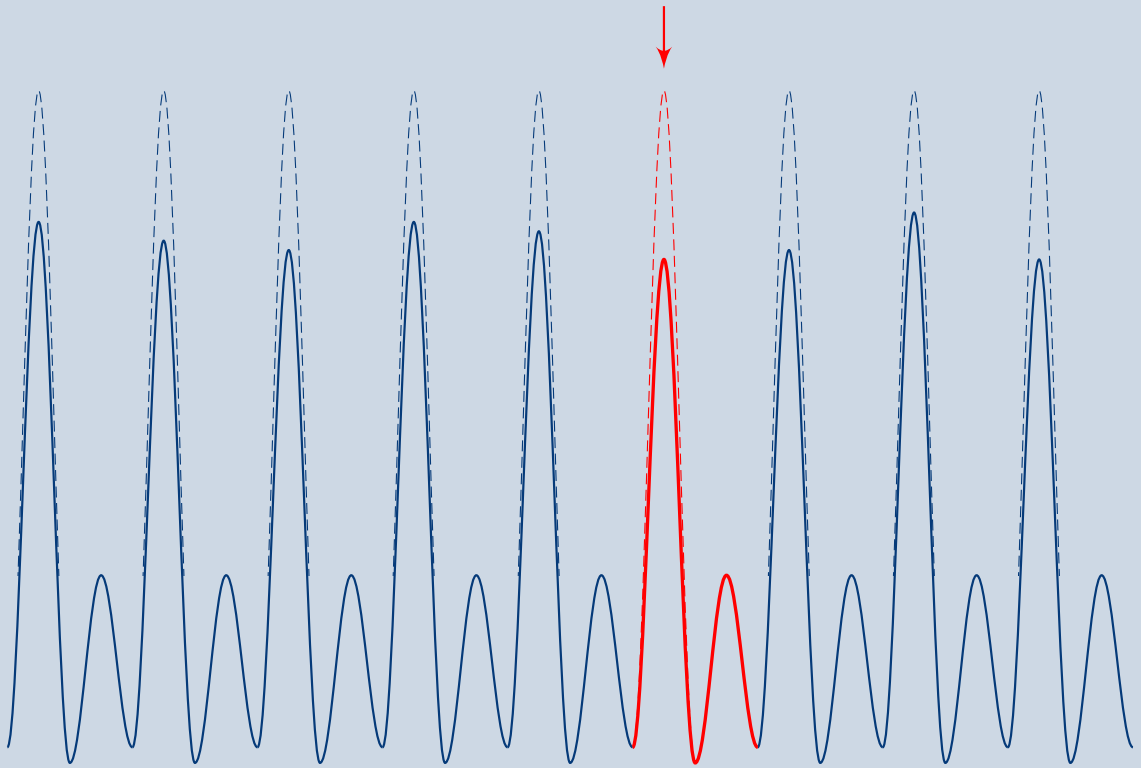
Core Ideas



Masking = same point, random amplitude

SCA Countermeasures

Core Ideas

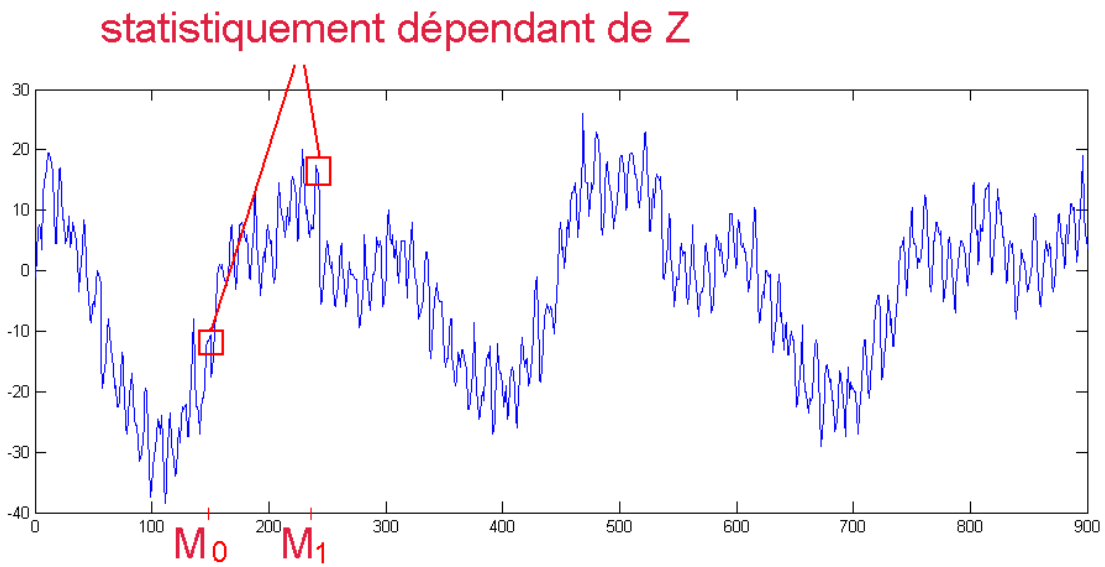


Balanced Logic = same point, constant amplitude

Higher Order Side Channel Attacks

Core Principle

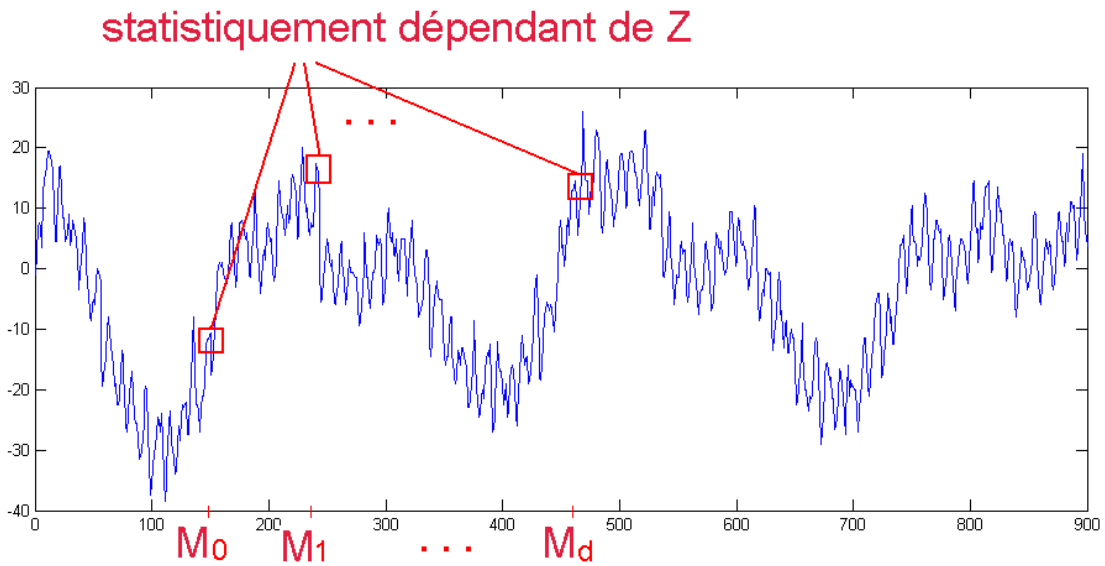
- First Order Masking: $M_0 = Z \oplus M_1$
- $\Longrightarrow$ Second Order SCA:



Higher Order Side Channel Attacks

Core Principle

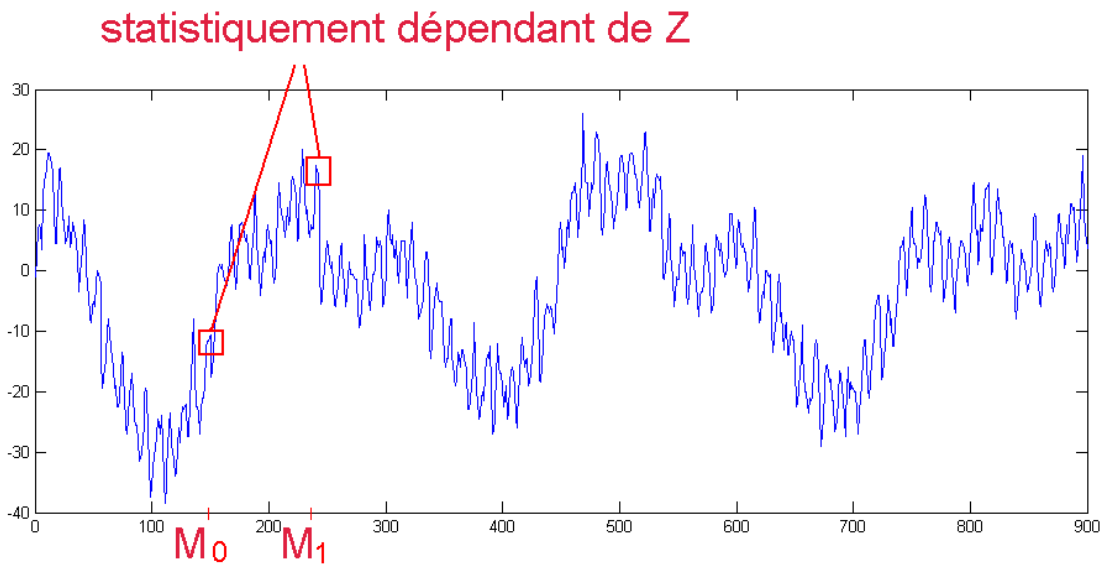
- Masking of order d: $M_0 = Z \oplus M_1 \oplus \cdots \oplus M_d$
- Attack of order $d + 1$:



Higher Order Side Channel Attacks

Core Principle

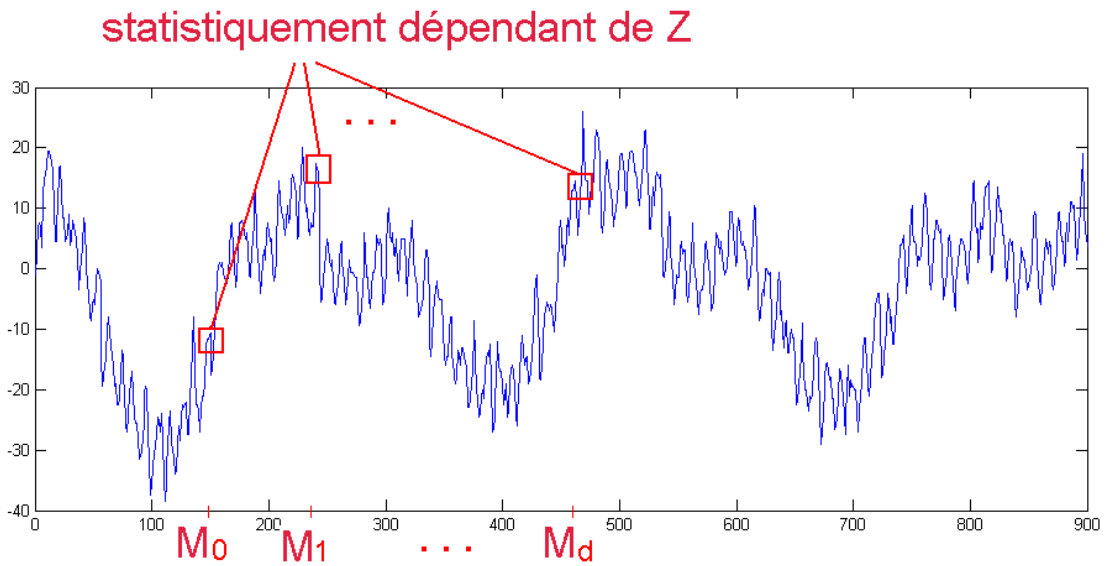
- First Order Masking: $M_0 = Z \oplus M_1$
- $\Longrightarrow$ Second Order SCA:



Higher Order Side Channel Attacks

Core Principle

- Masking of order d: $M_0 = Z \oplus M_1 \oplus \cdots \oplus M_d$
- Attack of order $d + 1$:



Higher Order Side Channel Attacks

Introduction

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA
  - [Messerges in his PhD Thesis]
  - Improved latter in Prouff et al at IEEE TC 2009 or in Gierlichs et al at Journal of Cryptology 2011
- t^{th}-order Shuffling: Integrated Attacks
  - [Clavier et al at CHES 2000]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA
  - [Tillich et al at ACNS 2007]
  - Improved in [Rivain et al at CHES 2009]

Higher Order Side Channel Attacks
Introduction

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA
  - [Messerges in his PhD Thesis]
  - Improved latter in Prouff et al at IEEE TC 2009 or in Gierlichs et al at Journal of Cryptology 2011
- t^{th}-order Shuffling: Integrated Attacks
  - [Clavier et al at CHES 2000]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA
  - [Tillich et al at ACNS 2007]
  - Improved in [Rivain et al at CHES 2009]

Higher Order Side Channel Attacks
Introduction

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA
  - [Messerges in his PhD Thesis]
  - Improved latter in Prouff et al at IEEE TC 2009 or in Gierlichs et al at Journal of Cryptology 2011
- t^{th}-order Shuffling: Integrated Attacks
  - [Clavier et al at CHES 2000]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA
  - [Tillich et al at ACNS 2007]
  - Improved in [Rivain et al at CHES 2009]

Higher Order Side Channel Attacks
Introduction

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA
  - [Messerges in his PhD Thesis]
  - Improved latter in Prouff et al at IEEE TC 2009 or in Gierlichs et al at Journal of Cryptology 2011
- t^{th}-order Shuffling: Integrated Attacks
  - [Clavier et al at CHES 2000]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA
  - [Tillich et al at ACNS 2007]
  - Improved in [Rivain et al at CHES 2009]

Higher Order Side Channel Attacks
Introduction

Advanced SCA have been defined to target each CM

- d^{th}-order Masking: HO-SCA
  - [Messerges in his PhD Thesis]
  - Improved latter in Prouff et al at IEEE TC 2009 or in Gierlichs et al at Journal of Cryptology 2011
- t^{th}-order Shuffling: Integrated Attacks
  - [Clavier et al at CHES 2000]
- (d^{th}-order Masking)-and-(t^{th}-order shuffling): Integrated HO-SCA
  - [Tillich et al at ACNS 2007]
  - Improved in [Rivain et al at CHES 2009]

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $M_{\hat{k}}$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(M_{\hat{k}}, f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $M_{\hat{k}}$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(M_{\hat{k}}, f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Example: if $Z = S(M + k)$ and $M_{\hat{k}} = \mathrm{HW}[S(M + \hat{k})]$, we have ...

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.

In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

Higher Order Side Channel Attacks

Introduction - General Principle

All the previous SCA follow the same outlines.

1. Input: set of observations for the signals $(L_i)_i$ related to a sensitive datum Z
2. Choose a statistical distintguisher Δ and a pre-processing function f
3. From the observations, estimate $f(L_i)$
4. For every hypothesis $\mathrm{HW}[S(M + \hat{k})]$ on Z, estimate

$$\Delta_{\hat{k}} = |\Delta(\mathrm{HW}[S(M + \hat{k})], f((L_i)_i))| \ .$$

5. Select the hypothesis that maximizes the estimation of $\Delta_{\hat{k}}$.

Note: if the mutual information is used instead of the correlation coefficient, there is not need for a pre-processing function f.
In other cases, the single difference is the function f.

HO-SCA against Higher Order Masking

Illustration with Δ being Pearson' Correlation Coefficient

Context: sensitive variable Z split into $d+1$ shares M_0, ..., M_d
Notation: L_i is the signal related to M_i.

Function f is a normalized product:

$$f(L_0, \cdots, L_d) = \prod_{i=0}^{d} (L_i - \mathbf{E}(L_i)) \ .$$

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{cst_1}{\left(\sqrt{1 + cst_2 \cdot \sigma^2}\right)^{d+1}} \ .$$

It is denoted by $\rho(d, \sigma)$.

HO-SCA against Higher Order Masking

Illustration with Δ being Pearson' Correlation Coefficient

Context: sensitive variable Z split into $d+1$ shares M_0, ..., M_d

Notation: L_i is the signal related to M_i.

Function f is a normalized product:

$$f(L_0, \cdots, L_d) = \prod_{i=0}^{d} (L_i - \mathbf{E}(L_i)) \ .$$

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{cst_1}{\left(\sqrt{1 + cst_2 \cdot \sigma^2}\right)^{d+1}} \ .$$

It is denoted by $\rho(d, \sigma)$.

HO-SCA against Higher Order Masking

Illustration with Δ being Pearson' Correlation Coefficient

Context: sensitive variable Z split into $d+1$ shares M_0, ..., M_d
Notation: L_i is the signal related to M_i.

Function f is a normalized product:

$$f(L_0, \cdots, L_d) = \prod_{i=0}^{d} (L_i - \mathbf{E}(L_i)) \ .$$

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{cst_1}{\left(\sqrt{1 + cst_2 \cdot \sigma^2}\right)^{d+1}} \ .$$

It is denoted by $\rho(d, \sigma)$.

HO-SCA against Higher Order Masking

Illustration with Δ being Pearson' Correlation Coefficient

Context: sensitive variable Z split into $d+1$ shares M_0, ..., M_d
Notation: L_i is the signal related to M_i.

Function f is a normalized product:

$$f(L_0, \cdots, L_d) = \prod_{i=0}^{d} (L_i - \mathbf{E}(L_i)) \ .$$

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{cst_1}{\left(\sqrt{1 + cst_2 \cdot \sigma^2}\right)^{d+1}} \ .$$

It is denoted by $\rho(d, \sigma)$.

HO-SCA against Higher Order Masking

Illustration with Δ being Pearson' Correlation Coefficient

Context: sensitive variable Z split into $d+1$ shares M_0, ..., M_d
Notation: L_i is the signal related to M_i.

Function f is a normalized product:

$$f(L_0, \cdots, L_d) = \prod_{i=0}^{d} (L_i - \mathbf{E}(L_i)) \ .$$

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{cst_1}{\left(\sqrt{1 + cst_2 \cdot \sigma^2}\right)^{d+1}} \ .$$

It is denoted by $\rho(d, \sigma)$.

Integrated SCA Against Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: the signal S containing information about Z is randomly spread over t different signals L_1, ..., L_t.

Function f is an Integrated signal:

$$f(L_1, \cdots, L_t) = L_1 + L_2 + ... + L_t$$

Note: the sum always contains the term S.
In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{t}\sqrt{1 + cst_2 \cdot \sigma^2}} \cdot$$

Integrated SCA Against Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: the signal S containing information about Z is randomly spread over t different signals L_1, ..., L_t.

Function f is an Integrated signal:

$$f(L_1, \cdots, L_t) = L_1 + L_2 + ... + L_t$$

Note: the sum always contains the term S.
In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{t}\sqrt{1 + cst_2 \cdot \sigma^2}} \cdot$$

Integrated SCA Against Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: the signal S containing information about Z is randomly spread over t different signals L_1, ..., L_t.

Function f is an Integrated signal:

$$f(L_1, \cdots, L_t) = L_1 + L_2 + ... + L_t$$

Note: the sum always contains the term S.

In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{t}\sqrt{1 + cst_2 \cdot \sigma^2}} \cdot$$

Integrated SCA Against Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: the signal S containing information about Z is randomly spread over t different signals L_1, ..., L_t.

Function f is an Integrated signal:

$$f(L_1, \cdots, L_t) = L_1 + L_2 + ... + L_t$$

Note: the sum always contains the term S.
In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{t}\sqrt{1 + cst_2 \cdot \sigma^2}} \cdot$$

Advanced SCA *vs* Masking-and-Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: X is split into $d+1$ shares M_0, M_1, ..., M_d whose manipulations are randomly spread over t different times.

Function f is a Combined-and-Integrated Signal:

$$f((L_i)_i) = \sum_{(i_0,\ldots,i_d)\in I} \prod_{j=0}^{d} \left(L_{i_j} - \mathbf{E}(L_{i_j})\right) \ .$$

Note: the sum always contains the term $\prod_{i=0}^{d}(M_i - \mathbf{E}(M_i))$.
In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{\#I}}\rho(d,\sigma) \ .$$

Advanced SCA *vs* Masking-and-Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: X is split into $d+1$ shares M_0, M_1, ..., M_d whose manipulations are randomly spread over t different times.

Function f is a Combined-and-Integrated Signal:

$$f((L_i)_i) = \sum_{(i_0,\ldots,i_d)\in I} \prod_{j=0}^{d} \left(L_{i_j} - \mathbf{E}(L_{i_j})\right) \ .$$

Note: the sum always contains the term $\prod_{i=0}^{d}(M_i - \mathbf{E}(M_i))$.
In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{\#I}}\rho(d,\sigma) \ .$$

Advanced SCA *vs* Masking-and-Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: X is split into $d+1$ shares M_0, M_1, ..., M_d whose manipulations are randomly spread over t different times.

Function f is a Combined-and-Integrated Signal:

$$f((L_i)_i) = \sum_{(i_0,\ldots,i_d)\in I} \prod_{j=0}^{d} \left(L_{i_j} - \mathbf{E}(L_{i_j})\right) \quad .$$

Note: the sum always contains the term $\prod_{i=0}^{d}(M_i - \mathbf{E}(M_i))$.
In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{\#I}}\rho(d,\sigma) \quad .$$

Advanced SCA *vs* Masking-and-Shuffling

Illustration with Δ being Pearson' Correlation Coefficient

Context: X is split into $d+1$ shares M_0, M_1, ..., M_d whose manipulations are randomly spread over t different times.

Function f is a Combined-and-Integrated Signal:

$$f((L_i)_i) = \sum_{(i_0,\ldots,i_d)\in I} \prod_{j=0}^{d} \left(L_{i_j} - \mathbf{E}(L_{i_j})\right) \quad .$$

Note: the sum always contains the term $\prod_{i=0}^{d}(M_i - \mathbf{E}(M_i))$.
In the Hamming Weight Model, the efficiency satisfies:

$$\rho_k = \frac{1}{\sqrt{\#I}}\rho(d,\sigma) \quad .$$

Plan

RSA Key Generation

RSA keys

- 2 primes: (p, q) (between 512 and 4092 bits)
- $N = pq, e$
- $d = e^{-1} \bmod \phi(N)$

RSA strong keys

- 2 primes: (p, q) (between 512 and 4092 bits)
- $p - 1, p + 1, q - 1$ and $q + 1$ possess a large factor (100 bits)
- $N = pq, e$
- $d = e^{-1} \bmod \phi(N)$

Large prime numbers are easy to find [Gordon'84]

For r large, the r^{th} prime number is about $r \log r$
$\hookrightarrow$ by randomly picking r between x and $x/2$

$$\Pr(r \text{ is prime}) \simeq 1/(\log x)$$

Incremental Search

1. pick a random odd seed $v \in [x/2, x]$
2.
3. verify that it is a prime with good probability
4. if yes, return.
5. otherwise increase v by 2 and go back to step 2.

[Brant *et al* 91, 92] and norms ANSI X9.31 and FIPS 186-4

Large prime numbers are easy to find [Gordon'84]

For r large, the r^{th} prime number is about $r \log r$
$\hookrightarrow$ by randomly picking r between x and $x/2$

$$\Pr(r \text{ is prime}) \simeq 1/(\log x)$$

Incremental Search

1. pick a random odd seed $v \in [x/2, x]$
2.
3. verify that it is a prime with good probability
4. if yes, return.
5. otherwise increase v by 2 and go back to step 2.

[Brant *et al* 91, 92] and norms ANSI X9.31 and FIPS 186-4

Large prime numbers are easy to find [Gordon'84]

For r large, the r^{th} prime number is about $r \log r$
$\hookrightarrow$ by randomly picking r between x and $x/2$

$$\Pr(r \text{ is prime}) \simeq 1/(\log x)$$

Incremental Search with sieving process

1. pick a random odd seed $v \in [x/2, x]$
2. test the divisibility by small primes (< 256)
3. verify that it is a prime with good probability
4. if yes, return.
5. otherwise increase v by 2 and go back to step 2.

[Brant *et al* 91, 92] and norms ANSI X9.31 and FIPS 186-4

Large prime numbers are easy to find [Gordon'84]

For r large, the r^{th} prime number is about $r \log r$
$\hookrightarrow$ by randomly picking r between x and $x/2$

$$\Pr(r \text{ is prime}) \simeq 1/(\log x)$$

Incremental Search with sieving process

1. pick a random odd seed $v \in [x/2, x]$
2. test the divisibility by small primes (< 256)
3. verify that it is a prime with good probability
4. if yes, return.
5. otherwise increase v by 2 and go back to step 2.

[Brant *et al* 91, 92] and norms ANSI X9.31 and FIPS 186-4

Primes Generation by Incremental search

Input : A bit-length ℓ, an even constant τ, the set $\mathcal{S} = \{s_0, \cdots, s_{52}\}$ of all odd primes lower than 256 (stored in ROM), a number t of Miller-Rabin tests to perform
Output: A probable prime p

```
   /* Generate a seed                                                    */
1  randomly generate an odd ℓ-bit integer v_0;
   /* Prime Sieve                                                        */
2  v ← v_0; s ← s_0; j = 0;
3  while (v mod s ≠ 0) and (j < 53) do
4  |   j = j + 1;
5  |   s ← s_j;
6  if (j ≠ 53) then
7  |   v = v + τ;
8  |   goto Step 3;
   /* Probabilistic primality tests                                      */
9  else
10 |   i = 0;
   |   /* Process t Miller-Rabin's tests (stop if one fails)            */
11 |   while (Miller-Rabin( ) = ok) and (i < t) do
12 |   |   i = i + 1;
   /* Process 1 Lucas' test[a]                                           */
13 if (i = t) and (Lucas( ) = ok) then
14 |   return v;
15 else
16 |   v = v + τ;
17 |   goto Step 3;
```

Primes Generation by Incremental search

Input : A bit-length ℓ, an even constant τ, the set $\mathcal{S} = \{s_0, \cdots, s_{52}\}$ of all odd primes lower than 256 (stored in ROM), a number t of Miller-Rabin tests to perform
Output: A probable prime p

/* Generate a seed */
randomly generate an odd ℓ-bit integer v_0;
/* Prime Sieve */
$v \leftarrow v_0$; $s \leftarrow s_0$; $j = 0$;
while $(v \mod s \neq 0)$ *and* $(j < 53)$ **do**
    $j = j + 1$;
    $s \leftarrow s_j$;
if $(j \neq 53)$ **then**
    $v = v + \tau$;
    goto Step 3;
/* Probabilistic primality tests */
else
    $i = 0$;
    /* Process t Miller-Rabin's tests (stop if one fails) */
    while $(Miller\text{-}Rabin(\,) = ok)$ *and* $(i < t)$ **do**
        $i = i + 1$;
/* Process 1 Lucas' test[a] */
if $(i = t)$ *and* $(Lucas(\,) = ok)$ **then**
    return v;
else
    $v = v + \tau$;
    goto Step 3;

Primes Generation by Incremental search

Input : A bit-length ℓ, an even constant τ, the set $\mathcal{S} = \{s_0, \cdots, s_{52}\}$ of all odd primes lower than 256 (stored in ROM), a number t of Miller-Rabin tests to perform
Output: A probable prime p

/* Generate a seed */
1 randomly generate an odd ℓ-bit integer v_0;

/* Prime Sieve */
2 $v \leftarrow v_0$; $s \leftarrow s_0$; $j = 0$;
3 **while** $(v \mod s \neq 0)$ *and* $(j < 53)$ **do**
4 $\quad j = j + 1$;
5 $\quad s \leftarrow s_j$;
6 **if** $(j \neq 53)$ **then**
7 $\quad v = v + \tau$;
8 $\quad$ **goto** Step 3;

/* Probabilistic primality tests */
9 **else**
10 $\quad i = 0$;
$\quad$ /* Process t Miller-Rabin's tests (stop if one fails) */
11 $\quad$ **while** $(Miller\text{-}Rabin(\,) = ok)$ *and* $(i < t)$ **do**
12 $\quad\quad i = i + 1$;

/* Process 1 Lucas' test[a] */
13 **if** $(i = t)$ *and* $(Lucas(\,) = ok)$ **then**
14 $\quad$ **return** v;
15 **else**
16 $\quad v = v + \tau$;
17 $\quad$ **goto** Step 3;

SPA on key generations

[Finke *et al* 09]

- Attacker model: knows when the sieving is stopping for a candidate (*i.e.*when $v_0 + \tau i = 0 \bmod s_j$), where
  - i is the index of the tested candidate: assumed to be known.
  - s_j is the j^{th} sieving element: assumed to be known.
- From $p = v_0 + \tau n$ (with n the number of tested candidates), build the equations $p - \tau(n - i) = 0 \bmod s_j$, where
  - n is the final number of tested candidates: assumed to be known.
- Using the CRT, part of $p = v_0 + \tau n$ may be retrieved (including $p \bmod \prod_j s_j$)
- Finish using Coppersmith factorisation method
  $\hookrightarrow$ need at least $\frac{\log p}{2}$ bits

SPA on key generations

[Finke *et al* 09]

- Attacker model: knows when the sieving is stopping for a candidate (*i.e.*when $v_0 + \tau i = 0 \bmod s_j$), where
  - i is the index of the tested candidate: assumed to be known.
  - s_j is the j^{th} sieving element: assumed to be known.
- From $p = v_0 + \tau n$ (with n the number of tested candidates), build the equations $p - \tau(n - i) = 0 \bmod s_j$, where
  - n is the final number of tested candidates: assumed to be known.
- Using the CRT, part of $p = v_0 + \tau n$ may be retrieved (including $p \bmod \prod_j s_j$)
- Finish using Coppersmith factorisation method
  $\hookrightarrow$ need at least $\frac{\log p}{2}$ bits

. . . there exist a very convenient countermeasure

Prime Sieve

```
   /* Prime Sieve for v_0                                                    */
1  for j = 0 to 52 do
2  |   R[j] ← v_0 mod s_j;   /* costly modular reduction over ℓ-bit integers */
3  |   ;

   /* Prime Sieve for v_i with i > 0                                         */
4  v ← v_0;
5  while (R does not contain a null remainder) do
6  |   v = v + τ;
7  |   for j = 0 to 52 do
   |   |   /* efficient modular reduction over 8-bit integers               */
8  |   |   R[j] ← R[j] + τ mod s_j;
   |   |
```

Algorithm 1: Improved Prime Sieve

Prime Sieve

```
   /* Prime Sieve for v_0                                                    */
1  for j = 0 to 52 do
2  |   R[j] ← v_0 mod s_j;   /* costly modular reduction over ℓ-bit integers */
3  |   ;
   /* Prime Sieve for v_i with i > 0                                         */
4  v ← v_0;
5  while (R does not contain a null remainder) do
6  |   v = v + τ;
7  |   for j = 0 to 52 do
   |   |   /* efficient modular reduction over 8-bit integers               */
8  |   |   R[j] ← R[j] + τ mod s_j;
```

Algorithm 2: Improved Prime Sieve

Prime Sieve

```
   /* Prime Sieve for v_0                                                    */
1  for j = 0 to 52 do
2  |   R[j] ← v_0 mod s_j;   /* costly modular reduction over ℓ-bit integers */
3  |   ;
   /* Prime Sieve for v_i with i > 0                                         */
4  v ← v_0;
5  while (R does not contain a null remainder) do
6  |   v = v + τ;
7  |   for j = 0 to 52 do
   |   |   /* efficient modular reduction over 8-bit integers              */
8  |   |   R[j] ← R[j] + τ mod s_j;
```

Algorithm 3: Improved Prime Sieve

Differential SCA on the balanced prime sieve

For every $j \leq \lambda$ and every $i \leq n$, the attacker may observe the manipulation of $v_0 + \tau i \bmod s_j$ by the device.
Then, he can try to recover v_0 which is the sole unknown value.

Focusing on $v \bmod s_j$

With a small hypothesis on $v_0 \bmod s_j$, for every $i \leq n$ one can guess

$$v_0 + \tau i \bmod s_j \ .$$

$\hookrightarrow$ select a leakage model and apply a DPA, CPA, MIA, ... to test the hypotheses $\hookrightarrow$ (hopefully) leads to the recovery of $v_0 \bmod s_j$.

Finish like [Finke *et al* 09]

Differential SCA on the balanced prime sieve

For every $j \leq \lambda$ and every $i \leq n$, the attacker may observe the manipulation of $v_0 + \tau i \bmod s_j$ by the device.
Then, he can try to recover v_0 which is the sole unknown value.

Focusing on $v \bmod s_j$

With a small hypothesis on $v_0 \bmod s_j$, for every $i \leq n$ one can guess

$$v_0 + \tau i \bmod s_j \ .$$

$\hookrightarrow$ select a leakage model and apply a DPA, CPA, MIA, ... to test the hypotheses $\hookrightarrow$ (hopefully) leads to the recovery of $v_0 \bmod s_j$.

Finish like [Finke *et al* 09]

- Size of n?
- Success Rate of this attack with common leakage model?
- Limitations in practice?

Simulation Results

Size of n

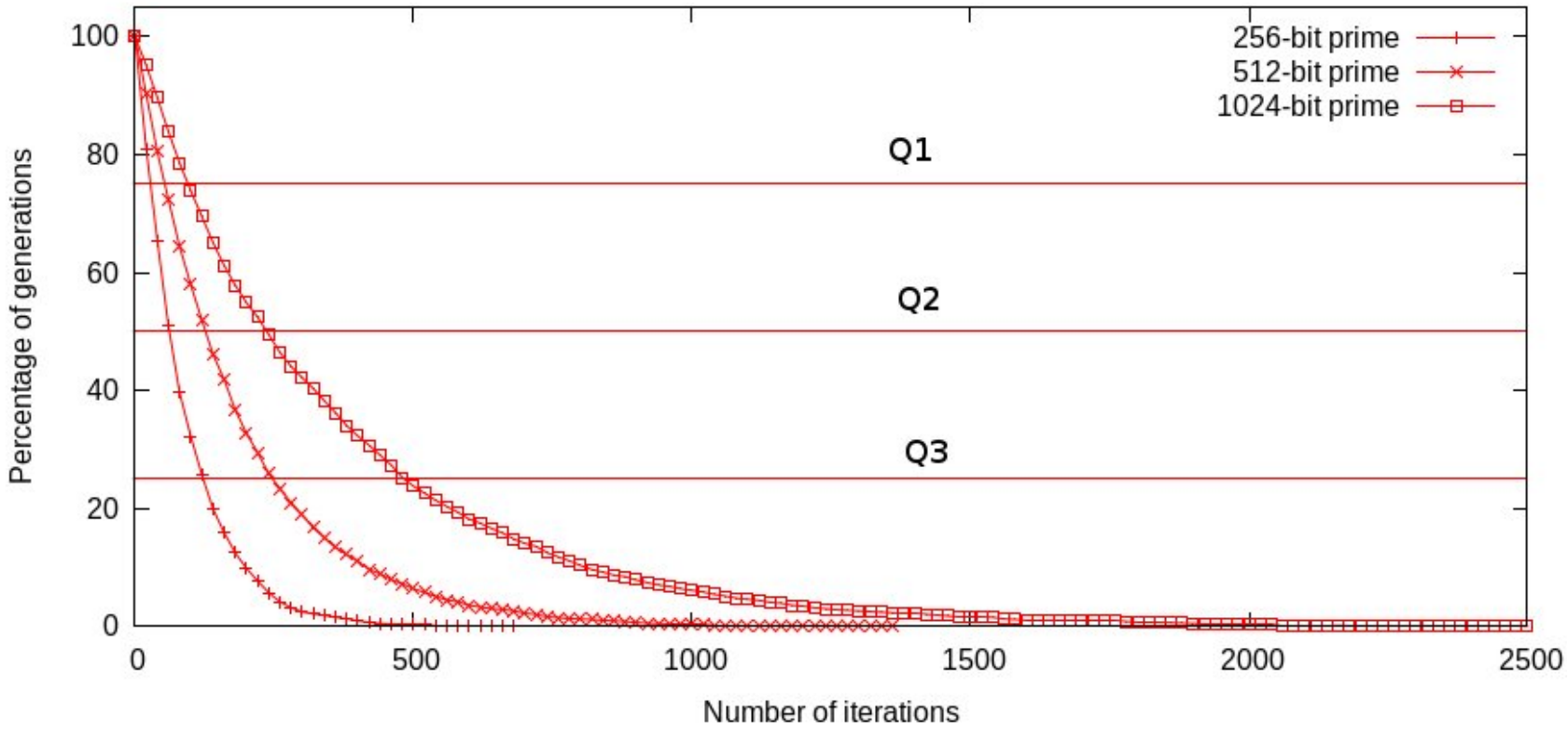
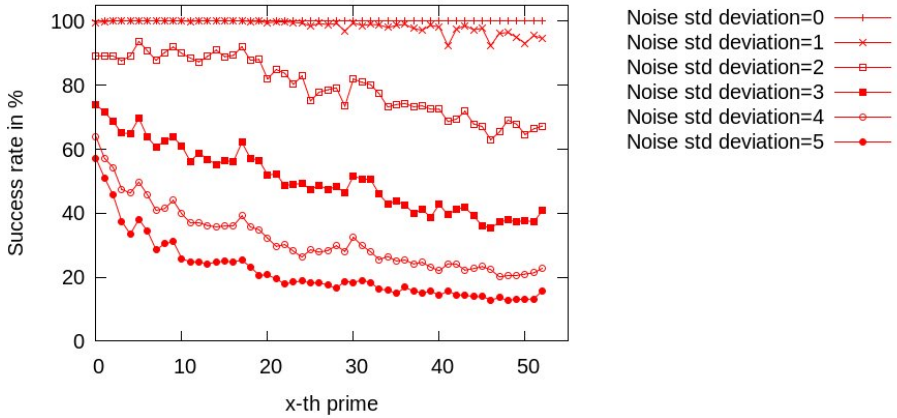


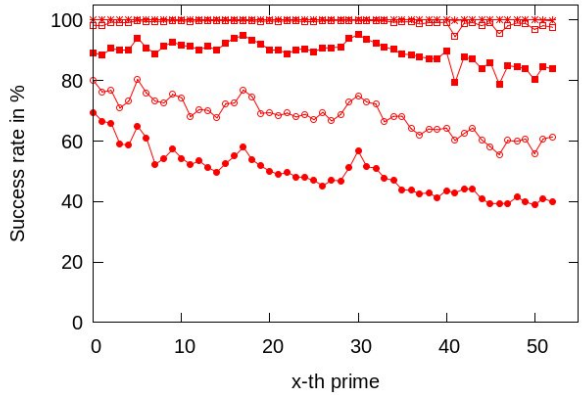
Figure: Cumulative distribution function of n for different prime bit-lengths ℓ

Simulation Results

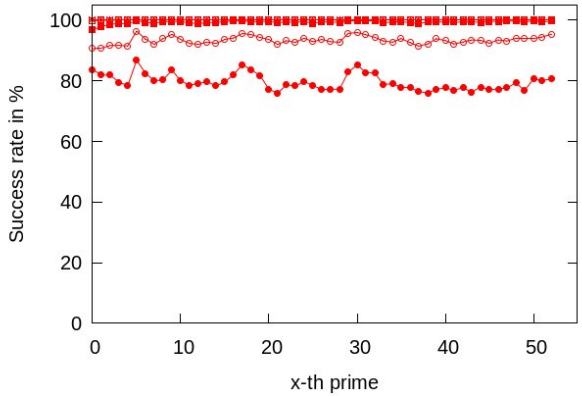
success rate



(a) 53 iterations (Q_1)



(b) 126 iterations (Q_2)

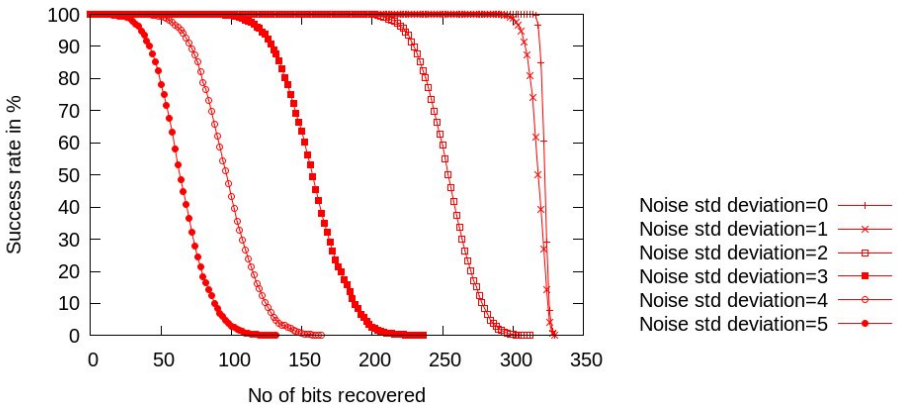


(c) 246 iterations (Q_3)

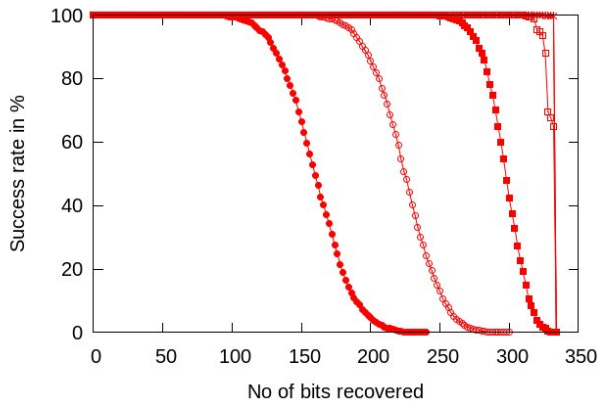
Figure: Success rate for the first F3 ...

Simulation Results

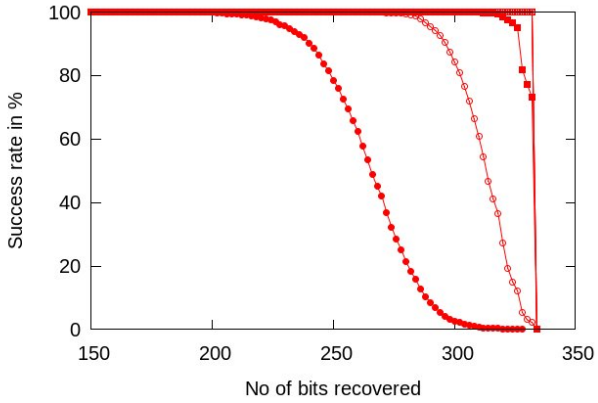
success rate



(a) 53 iterations (Q_1)



(b) 126 iterations (Q_2)



(c) 246 iterations (Q_3)

σ	Q_1	Q_2	Q_3
0	1	1	1
1	1	1	1
2	0.46	1	1
3	0	0.99	1
4	0	0.08	1
5	0	0	0.7

(d) Success rate to recover 256 bits

Traces acquisition in practice

Parameters

- 512 primes generations
- 8-bit CPU running at $\sim$ 50 MHz
- sieving size: 53
- primality tests: 10 Miller-Rabins

modular arithmetic co-processor

In a worst case scenario the generation can take 2.5s

Acquisition Oscilloscope

- 250 MSamples of memory
- 100 MSamples per seconds

Traces acquisition in practice

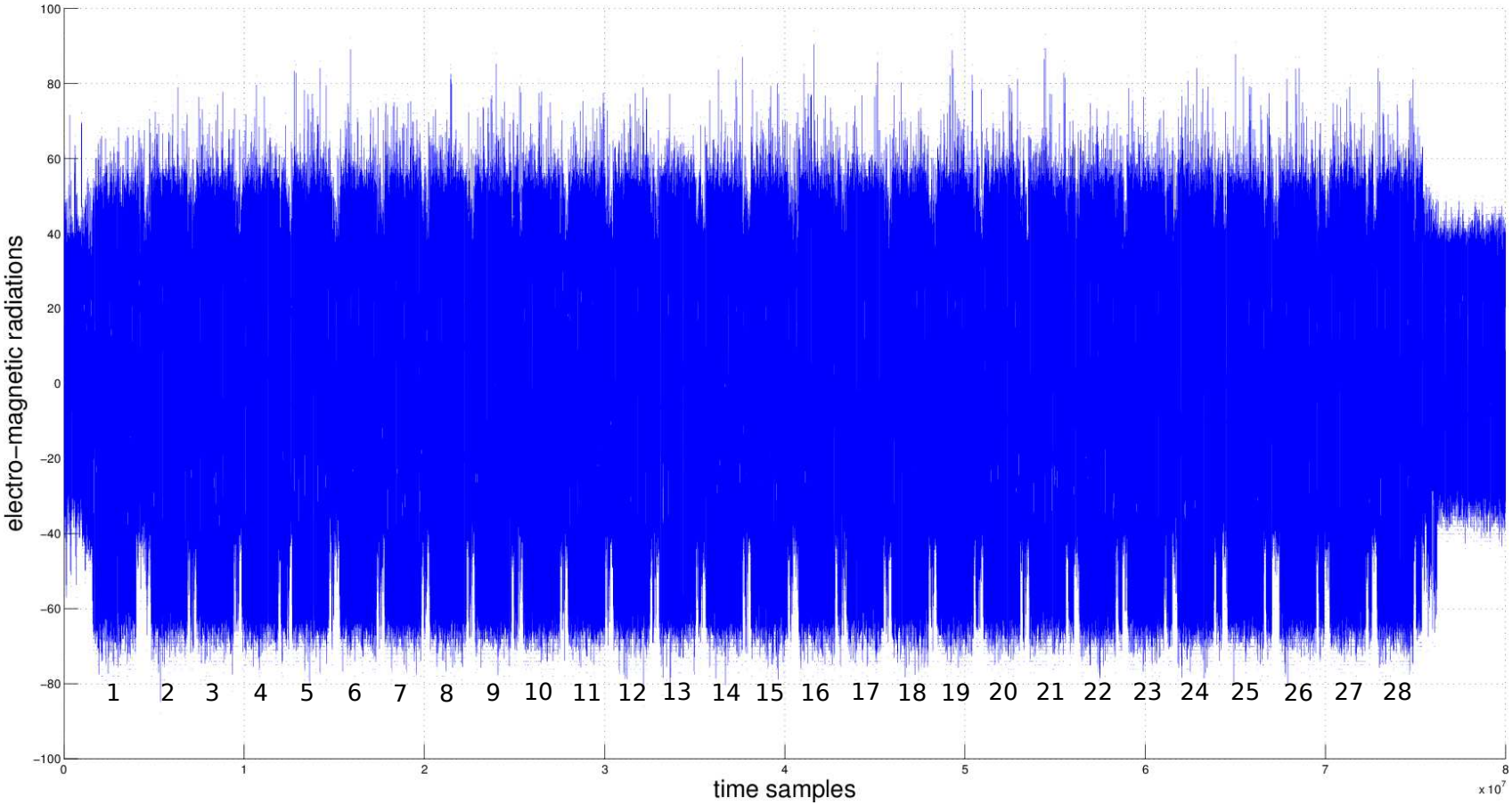
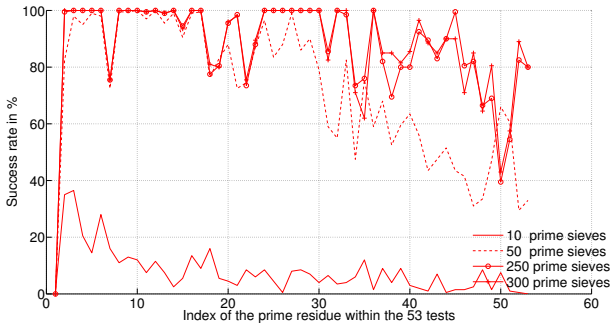


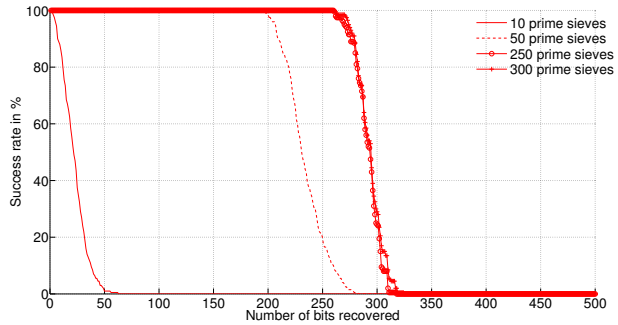
Figure: Electro-magnetic radiations measured during a prime number generation computation on a commercial smartcard. Pattern 1 corresponds to the initial costly prime sieve, whereas patterns 2 to 28 correspond to Miller-Rabin tests.

Results on a toy example

- focus on the sieving process on a 8-bit CPU
- varying number n of iterations



(a) Success rates for each prime sieve elements (over 200 attack experiments)



(b) Success rates for recovering x bits of information on the generated prime

Attack Improvements

Find the correct bits of p

- Key Enumeration Algorithms [Veyrat *et al* 12]
- Coherency check using RSA equation $N = pq$

Change the standards

Existing Solutions

- Fouque-Tibouchi 2011
  Inject randomness at each iteration
- Clavier *et al* 2012
  Efficient provable prime generation