Sécurité des implémentations de mécanismes cryptographiques

Adrian Thillard
adrian.thillard@ssi.gouv.fr

ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information)

Collége de l'X – 2017

Part I.

1 Side-Channel Analysis Introduction
- SCA basic principles
- History of Side-Channel Analysis
- Practical Issues
- Useful probability tools

Part II.

2 Timing attacks

Part III.

3 Sensitive Variables, Leakage Function and Model Inference
- Sensitive Variables
- Leakage Functions
- Model Inference

Part IV.

4 Distinguishers
- Perfect Leakage Model
- Partly Known Leakage Model

Part I

Introduction

Outline

Outline

Attacks against crypto primitives implementations

Attacker Models

- Classical Model: black-box model. The adversary observes only the input/output channel.
  - passively (non-adaptive attacks),
  - actively (adaptive attacks).

Attacks against crypto primitives implementations

Attacker Models

- Classical Model: black-box model. The adversary observes only the input/output channel.
  - passively (non-adaptive attacks),
  - actively (adaptive attacks).
- Opposite Model: white-box model. The adversary has a total access to the implementation.
  - The secrets are hidden in the implementation
  - The adversary must not be able to extract them.

Attacks against crypto primitives implementations

Attacker Models

- Classical Model: black-box model. The adversary observes only the input/output channel.
  - passively (non-adaptive attacks),
  - actively (adaptive attacks).
- Opposite Model: white-box model. The adversary has a total access to the implementation.
  - The secrets are hidden in the implementation
  - The adversary must not be able to extract them.
- In the middle: gray-box model. Embedded cryptography is vulnerable to other kinds of attacks through side channels.

Attacks against crypto primitives implementations

Attacker Models

- Classical Model: black-box model. The adversary observes only the input/output channel.
  - passively (non-adaptive attacks),
  - actively (adaptive attacks).
- Opposite Model: white-box model. The adversary has a total access to the implementation.
  - The secrets are hidden in the implementation
  - The adversary must not be able to extract them.
- In the middle: gray-box model. Embedded cryptography is vulnerable to other kinds of attacks through side channels.
  - the device behaviour indeed strongly depends on the value of the manipulated data.

Attacks against crypto primitives implementations

Attacker Models

- Classical Model: black-box model. The adversary observes only the input/output channel.
  - passively (non-adaptive attacks),
  - actively (adaptive attacks).
- Opposite Model: white-box model. The adversary has a total access to the implementation.
  - The secrets are hidden in the implementation
  - The adversary must not be able to extract them.
- In the middle: gray-box model. Embedded cryptography is vulnerable to other kinds of attacks through side channels.
  - the device behaviour indeed strongly depends on the value of the manipulated data.
  - active attacks: execution is disturbed during sensitive manipulations

Attacks against crypto primitives implementations

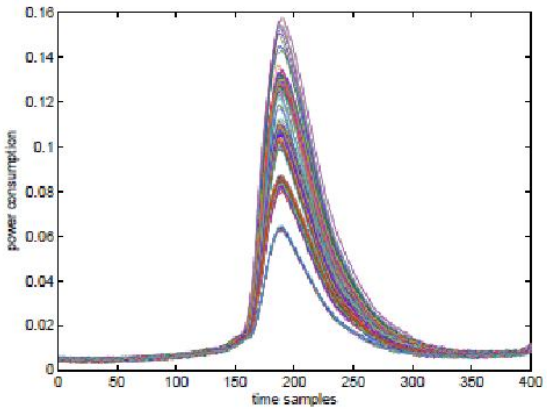
Attacker Models

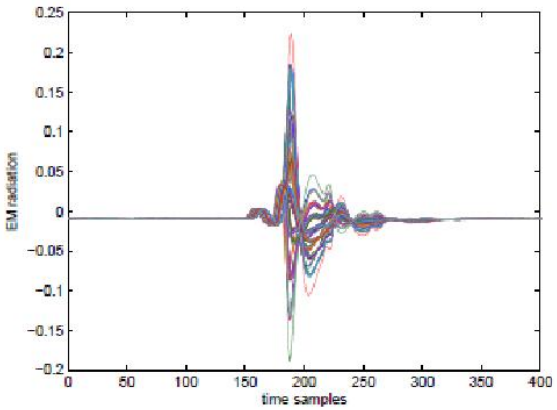
- Classical Model: black-box model. The adversary observes only the input/output channel.
  - passively (non-adaptive attacks),
  - actively (adaptive attacks).
- Opposite Model: white-box model. The adversary has a total access to the implementation.
  - The secrets are hidden in the implementation
  - The adversary must not be able to extract them.
- In the middle: gray-box model. Embedded cryptography is vulnerable to other kinds of attacks through side channels.
  - the device behaviour indeed strongly depends on the value of the manipulated data.
  - active attacks: execution is disturbed during sensitive manipulations
  - passive attacks: execution is observed to retrieve information

Side Channel 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: traces repartition does not look random.

Side Channel Leakage

What ?

The behaviour of a "processor" depends on (1) the kind of executed operations and (2) the kind of operands that are processed.

Side Channel Leakage

What ?

The behaviour of a "processor" depends on (1) the kind of executed operations and (2) the kind of operands that are processed.
Among side channels we have:

- the execution timing;
- the power consumption;
- the electromagnetic emanations;
- the sound;
- ...

Side Channel Leakage

What ?

The behaviour of a "processor" depends on (1) the kind of executed operations and (2) the kind of operands that are processed.
Among side channels we have:

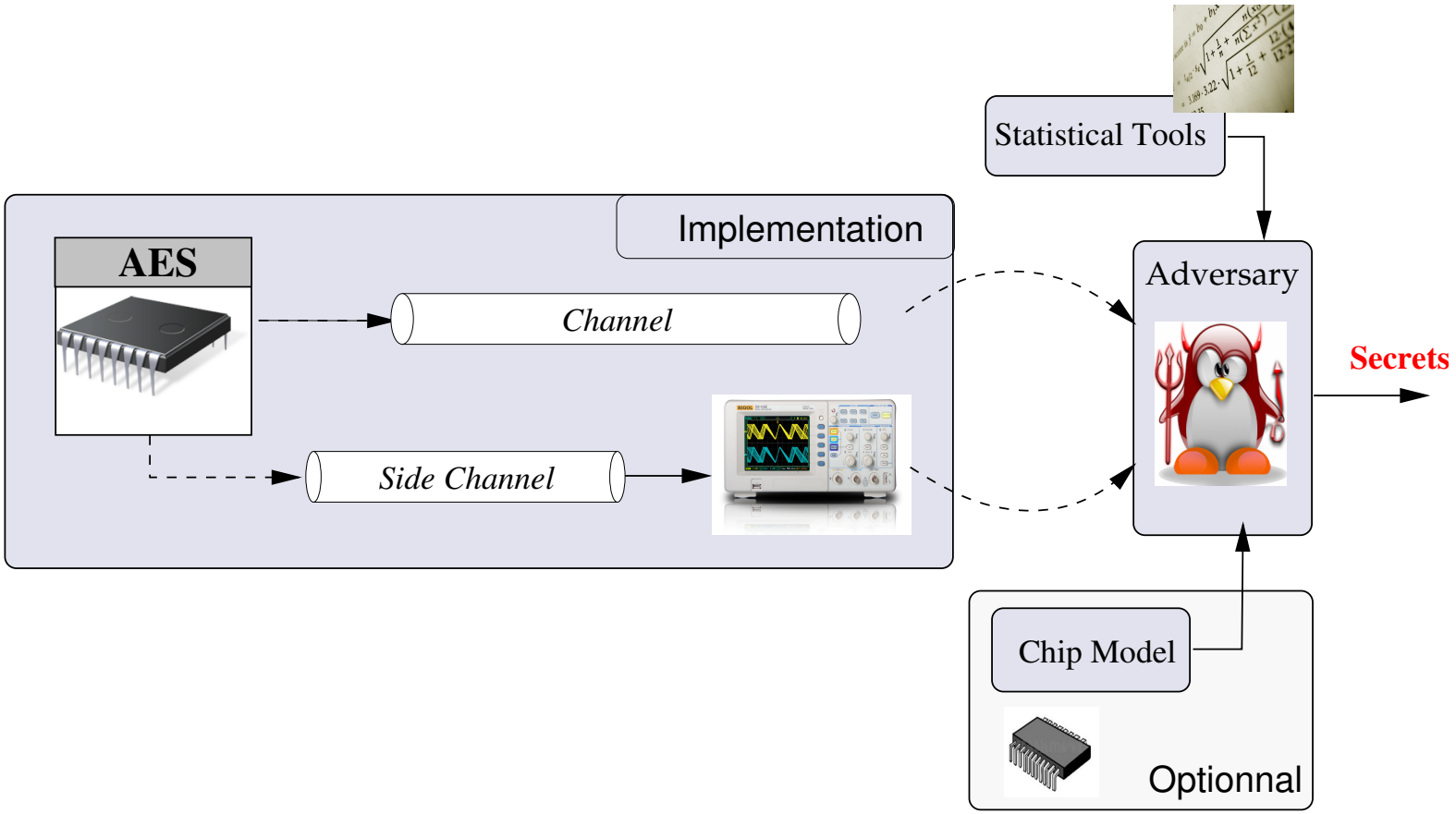
- the execution timing;
- the power consumption;
- the electromagnetic emanations;
- the sound;
- ...

They are not taken into account by the "classical" cryptanalyses.
Side channel attacks (SCA) are passives or actives. In a SCA, the adversary can:

- *observe* the behaviour of the device during the processing,
- [active SCA only] modify them.

Side Channel Attacks

Side Channel Analysis: General Framework.



Side Channel Attacks

Main targets

Small Embedded devices

- attacker model fits
- simple design and package

Recent results show

- observation attacks on smart-phones and tablets.
- observation attacks on PCs.

GenkinPipmanShamirTromer 2013-2014

State of the art...

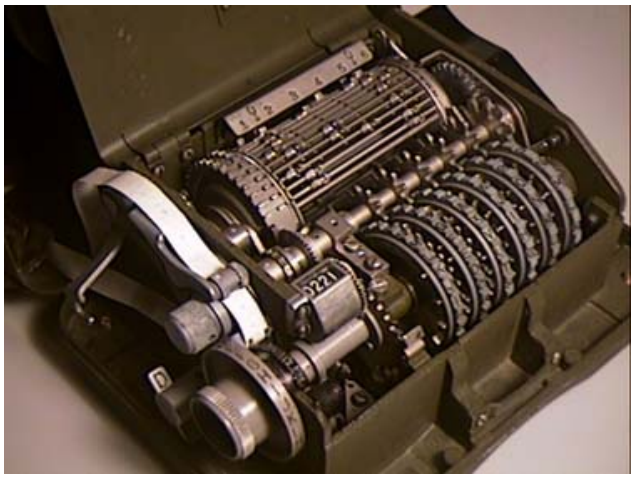
Seems legit

Outline

A brief History

1956 : MI5 breaks the encryption system used by the Egyptian Embassy at London.

- Encryption system: machine with rotors from Hagelin family.
- Peter Wright suggests to put a microphone in the cipher room.
- The sound produced by the machine when the 7 rotors are initialized indeed gives information on its initial state, then enabling to get the key and to decipher.



Kocher's success story

against the smart-card industry

- prologue
  - 1996: Timing Attacks.
  - 1998: Power Analysis.

Kocher's success story

against the smart-card industry

- prologue
  - 1996: Timing Attacks.
  - 1998: Power Analysis.
  - 1998: Get the smart card industry aware.
    $\hookrightarrow$ countermeasures implementations.

Kocher's success story

against the smart-card industry

- prologue
  - 1996: Timing Attacks.
  - 1998: Power Analysis.
  - 1998: Get the smart card industry aware.
    $\hookrightarrow$ countermeasures implementations.
  - 1998: CRI Starts writing patents on countermeasures.

Kocher's success story

against the smart-card industry

- prologue
  - 1996: Timing Attacks.
  - 1998: Power Analysis.
  - 1998: Get the smart card industry aware.
    $\hookrightarrow$ countermeasures implementations.
  - 1998: CRI Starts writing patents on countermeasures.
- tale
  - 2004: Set of patents ready $\hookrightarrow$ CRI asks VISA to pay fees
  - 2004-2009: court battle

Kocher's success story

against the smart-card industry

- prologue
  - 1996: Timing Attacks.
  - 1998: Power Analysis.
  - 1998: Get the smart card industry aware.
    $\hookrightarrow$ countermeasures implementations.
  - 1998: CRI Starts writing patents on countermeasures.
- tale
  - 2004: Set of patents ready $\hookrightarrow$ CRI asks VISA to pay fees
  - 2004-2009: court battle
  - Sept. 2009: CRI/VISA agreement $\hookrightarrow$ VISA agrees to pay
  - 2009-20xx: MasterCard, NXP, Infineon, etc. agree to pay

Kocher's success story

against the smart-card industry

- prologue
  - 1996: Timing Attacks.
  - 1998: Power Analysis.
  - 1998: Get the smart card industry aware.
    $\hookrightarrow$ countermeasures implementations.
  - 1998: CRI Starts writing patents on countermeasures.
- tale
  - 2004: Set of patents ready $\hookrightarrow$ CRI asks VISA to pay fees
  - 2004-2009: court battle
  - Sept. 2009: CRI/VISA agreement $\hookrightarrow$ VISA agrees to pay
  - 2009-20xx: MasterCard, NXP, Infineon, etc. agree to pay
- epilogue: in 2020 patents will start to fall...

Side Channel Analysis

- Side Channel Attacks (SCA) appear about 20 years ago
  - 1996: Timing Attacks
  - 1998: Power Analysis
  - 2000: Electromagnetic Analysis

Side Channel Analysis

- Side Channel Attacks (SCA) appear about 20 years ago
  - 1996: Timing Attacks
  - 1998: Power Analysis
  - 2000: Electromagnetic Analysis
- Numerous attacks
  - 1998: (single-bit) DPA — KocherJaffeJune1999
  - 1999: (multi-bit) DPA — Messerges1999
  - 2000: Higher-order SCA — Messerges2000
  - 2002: Template SCA — ChariRaoRohatgi2002
  - 2004: CPA — BrierClavierOlivier2004
  - 2005: Stochastic SCA — SchindlerLemkePaar2005
  - 2008: Mutual Information SCA — GierlichsBatinaTuyls2008
  - etc.

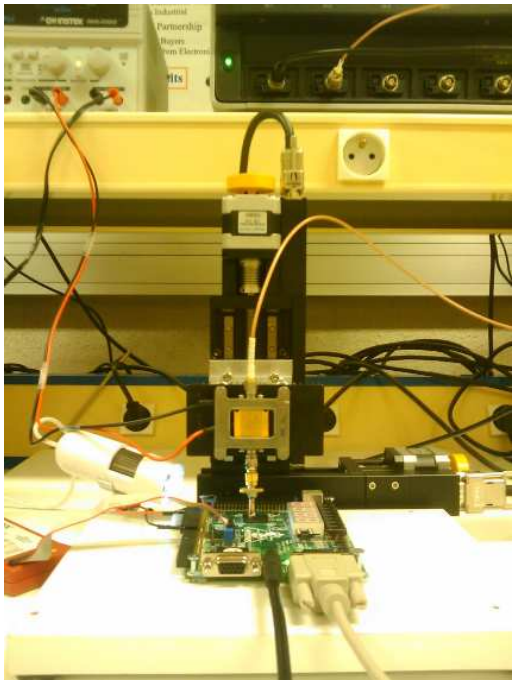
Outline

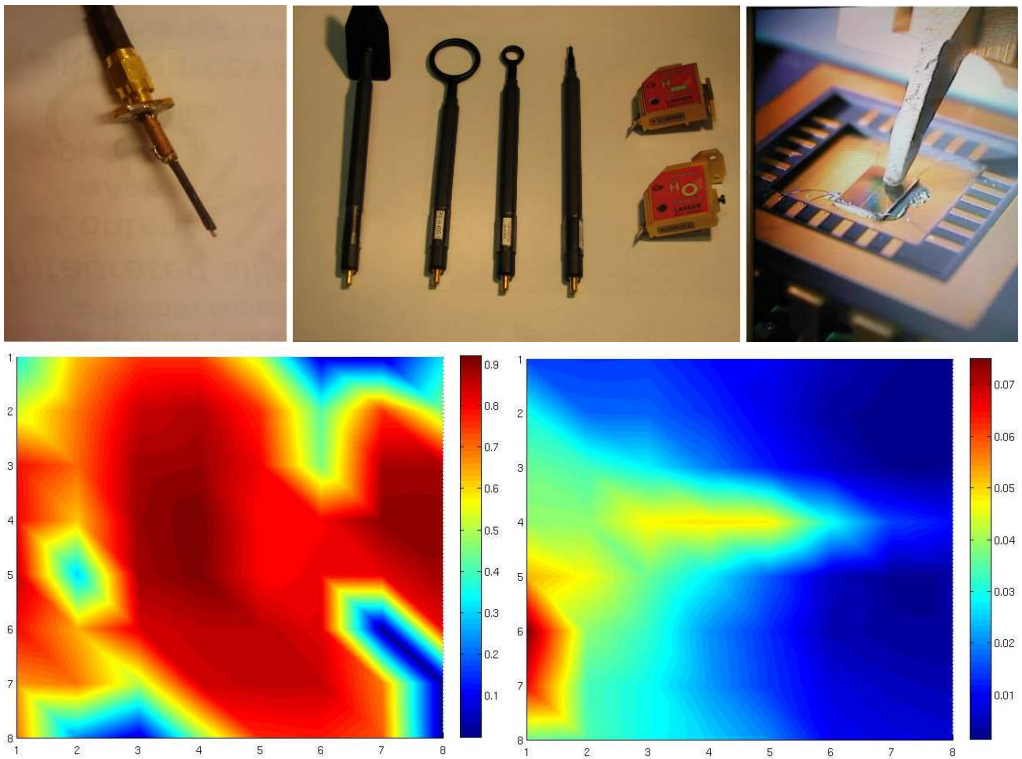
Practical Issues

Side Channel Analysis: Measurement Setup

Advanced Side Channel Attacks (DPA like attacks)

Side Channel Analysis: Measurement Setup, example of EMA

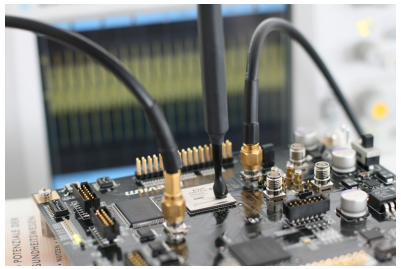




Practical Issues

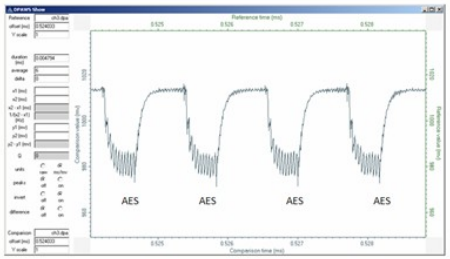
Side Channel Analysis: Measurement Step in details.

1. Which kind of measure (power consumption, electromagnetic emanations, timing, temperature)?

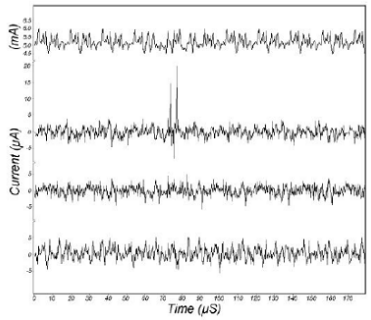


Engineers Expertise. Quality of the attacks labs.

2. Which point (time) to attack in a set of measurement traces?



Sometimes easy ...



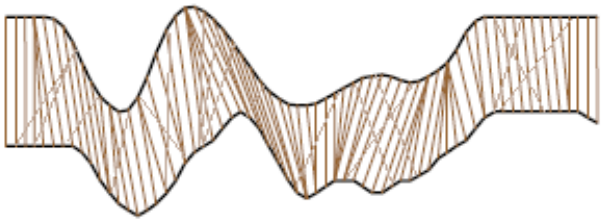
sometimes not!

Practical Issues

Side Channel Analysis: Find points of interest.

Still Engineers Expertise and use of signal processing techniques.

- First step: re-synchronization of the traces. Use of pattern matching techniques, voice synchronization techniques, etc. (*e.g.* see Muijrers et al. at CARDIS 2011).

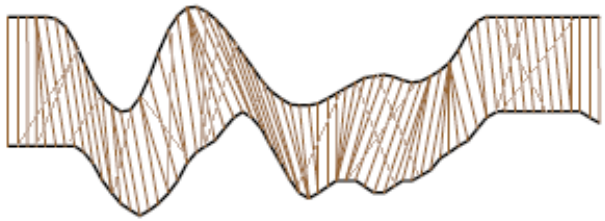


Practical Issues

Side Channel Analysis: Find points of interest.

Still Engineers Expertise and use of signal processing techniques.

- First step: re-synchronization of the traces. Use of pattern matching techniques, voice synchronization techniques, etc. (*e.g.* see Muijrers et al. at CARDIS 2011).



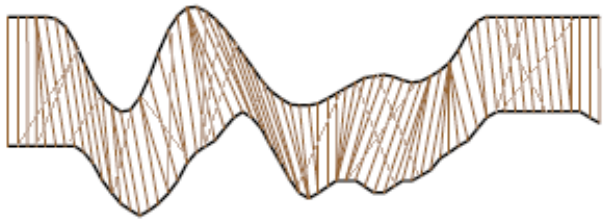
- Second step: find the points of interest. Use of signal processing techniques (*e.g.* see Bohy et al published ESmart 2003).

Practical Issues

Side Channel Analysis: Find points of interest.

Still Engineers Expertise and use of signal processing techniques.

- First step: re-synchronization of the traces. Use of pattern matching techniques, voice synchronization techniques, etc. (*e.g.* see Muijrers et al. at CARDIS 2011).



- Second step: find the points of interest. Use of signal processing techniques (*e.g.* see Bohy et al published ESmart 2003).
- Third step: select a few points of interest and apply the attack independently on each of them.

Outline

Terminology and References

Probability

mathematical theory that describes uncertainty.

Statistics

set of techniques for extracting useful information from data.

References

- *Fundamentals of Probability and Statistics For Engineers*, by T.T. Song.
- *Introduction to probability theory and its applications*, by W. Feller.
- *Mathematical Statistics*, by Jun Shao.

Useful Definitions 1/2

- Random Variable: is a variable whose value is subject to variations due to chance.
- Probability Distribution Function (PDF) of X is the function F_X defined by:

$$F_X(x) = \Pr(X \leq x)$$

  Note: $\{X \leq x\}$ is an event. A PDF is also called cumulative distribution function (cdf).
- A random variable X is discrete or continuous depending on its definition set.

Useful Definitions 2/2

- For a discrete random variable, one can define the probability mass function (pmf) p_X by:

$$p_X(x) = \Pr(X = x)$$

- For a continuous random variable X, one can define its probability density function (pdf) f_X by:

$$f_X(x) = \frac{\mathrm{d}F_X(x)}{\mathrm{d}x} \text{ or equivalently } F_X(x) = \int_{-\infty}^{t} f_X(t)\mathrm{d}x$$

Statistical Moments (discrete random variables)

- Mean

$$E[X] = \sum_x x \Pr(X = x)$$

- Variance

$$V[X] = \sum_x x^2 \Pr(X = x) - (\sum_x x \Pr(X = x))^2$$

- kth order Moment

$$M_k[X] = \sum_x x^k \Pr(X = x)$$

Note: It is possible to reconstruct the distribution of X from all its moments (its moment generating function)

Notion of Entropy

- Entropy

$$\mathrm{H}(X) = -\sum_i \Pr(X = x_i) \log(\Pr(X = x_i))$$

- Conditional Entropy

$$\mathrm{H}(X \mid Y) = \sum_{i,j} \Pr(X = x_i \text{ and } Y = y_j) \log \frac{\Pr(Y = y_j)}{\Pr(X = x_i \text{ and } Y = y_j)}$$

- Mutual Information

$$\begin{aligned} \mathrm{MI}(X, Y) &= \mathrm{H}(Y) - \mathrm{H}(Y \mid X) \\ &= \mathrm{H}(X) - \mathrm{H}(X \mid Y) \end{aligned}$$

Discrete random variable X

mean

$$\mathrm{E}(X) = \sum_i x_i \Pr[X = x_i]$$

moment of order k

$$\mathrm{m}^{(k)} = \sum_i |x_i|^k \Pr[X = x_i]$$

centralized moment of order k

$$\mathrm{m}^{(k)} = \sum_i |x_i - \mathrm{E}(X)|^k \Pr[X = x_i]$$

variance

$$\mathrm{Var}(X) = \sum_i (x_i - \mathrm{E}(X))^2 \Pr[X = x_i]$$

Entropy

$$\mathrm{H}(X) = -\sum_i \Pr[X = x_i] \log(\Pr[X = x_i])$$

Continuous random variable X

expected value

$$\mathrm{E}(X) = \int_{-\infty}^{+\infty} x f_X(x) \mathrm{d}x$$

moment of order k

$$\mathrm{m}^{(k)} = \int_{-\infty}^{+\infty} x^k f_X(x) \mathrm{d}x$$

centralized moment of order k

$$\mathrm{m}^{(k)} = \int_{-\infty}^{+\infty} (x - \mathrm{E}(X))^k f_X(x) \mathrm{d}x$$

variance

$$\mathrm{Var}(X) = \int_{-\infty}^{+\infty} (x - \mathrm{E}(X))^2 f_X(x) \mathrm{d}x$$

Entropy

$$\mathrm{H}(X) = -\int_{-\infty}^{+\infty} f_X(x) \log(f_X(x)) \mathrm{d}x$$

Normal distribution

- A normal distribution has a bell-shaped probability density function, known as the Gaussian function and noted $\mathcal{N}(\mu, \sigma)$.
- It is uniquely characterized by two parameters μ and σ, respectively called mean and standard deviation ($\sigma^2 = \text{var}$).
- Its pdf is

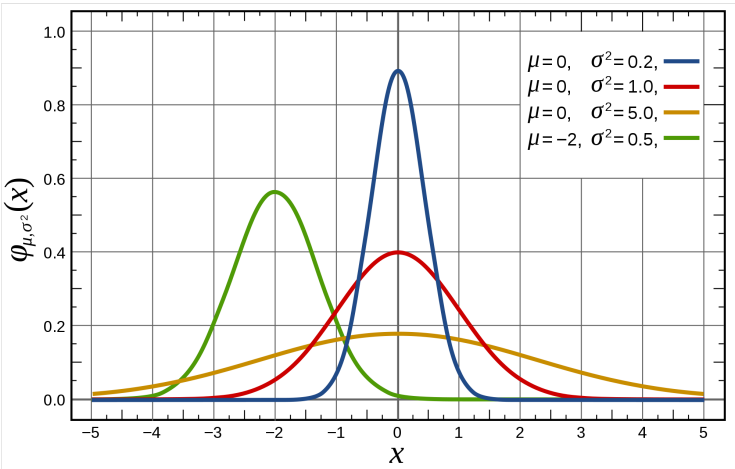
$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Its cdf is

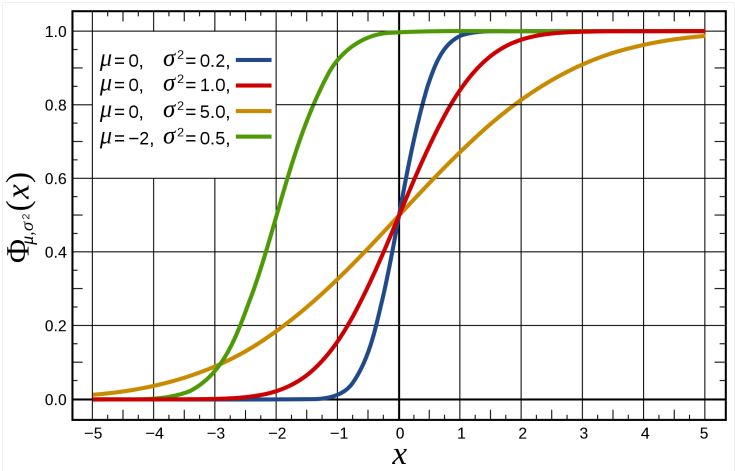
$$F_X(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$$

Normal distribution

Normal pdf



Normal cdf



Multivariate normal distribution.

- This is the natural multivariate extention of normal distribution (*i.e.* $\boldsymbol{X} = (X_1, \cdots, X_t)$), noted $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
- It is uniquely characterized by $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively the mean vector and covariance matrix.
  where $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_t)$ and $\boldsymbol{\Sigma} = \{E(X_i X_j) - E(X_i)E(X_j)\}_{i,j}$.
- Its pdf is

$$f_X(x) = \frac{1}{\sqrt{(2\pi)^t |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}$$

Independence between random variables

X and Y are independent iff

Independence between random variables

X and Y are independent iff

- $\forall(x, y)$, $\Pr(X = x, Y = y) = \Pr(X = x)\Pr(Y = y)$.
- $\forall(x, y)$, $\Pr(X = x \mid Y = y) = \Pr(X = x)$.
- $\mathrm{H}(X \mid Y) = \mathrm{H}(X)$.
- $\mathrm{MI}(X, Y) = 0$.

Independence between random variables

X and Y are independent iff

- $\forall(x, y)$, $\Pr(X = x, Y = y) = \Pr(X = x)\Pr(Y = y)$.
- $\forall(x, y)$, $\Pr(X = x \mid Y = y) = \Pr(X = x)$.
- $\mathrm{H}(X \mid Y) = \mathrm{H}(X)$.
- $\mathrm{MI}(X, Y) = 0$.

If X and Y are independent, then $\mathbf{E}(XY) = \mathbf{E}(X)\mathbf{E}(Y)$

Note: the converse is false.

Independence between random variables

X and Y are independent iff

- $\forall(x, y)$, $\Pr(X = x, Y = y) = \Pr(X = x)\Pr(Y = y)$.
- $\forall(x, y)$, $\Pr(X = x \mid Y = y) = \Pr(X = x)$.
- $\mathrm{H}(X \mid Y) = \mathrm{H}(X)$.
- $\mathrm{MI}(X, Y) = 0$.

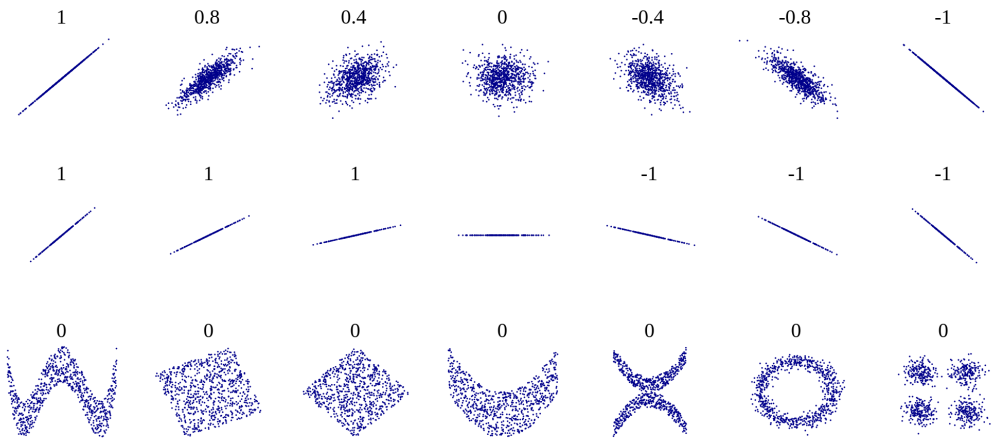
If X and Y are independent, then $\mathbf{E}(XY) = \mathbf{E}(X)\mathbf{E}(Y)$

Note: the converse is false.
Note: the value $\mathbf{E}(XY) - \mathbf{E}(X)\mathbf{E}(Y)$ is called covariance between X and Y and is denoted by $\mathbf{cov}(X, Y)$.

Linear Correlation

Linear correlation between X and Y: $\rho(X, Y) = \frac{\mathbf{cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}}$.

- It also called Pearson Coefficient.
- $\rho(X, Y) = \rho(Y, X)$ and $\rho(X, Y) \in [-1, 1]$.
- $\rho(X, Y) = 1$ implies that X and Y are dependent.

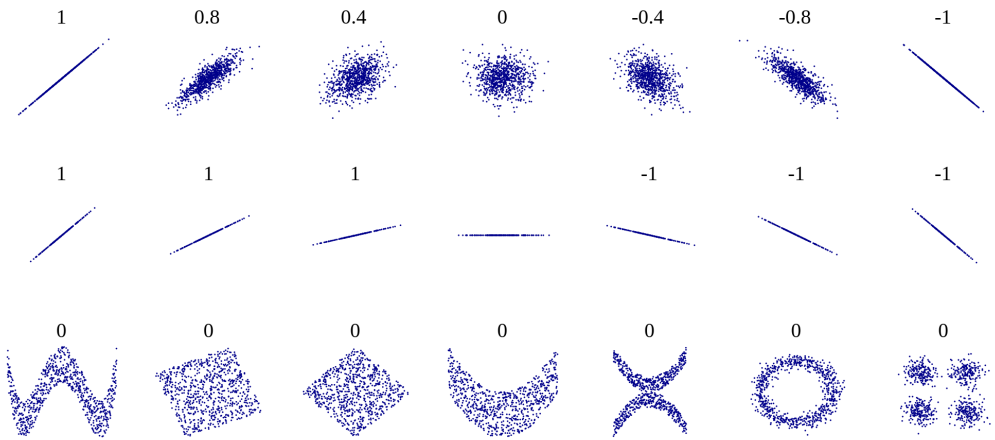


Linear Correlation

Linear correlation between X and Y: $\rho(X, Y) = \frac{\mathbf{cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}}$.

- It also called Pearson Coefficient.
- $\rho(X, Y) = \rho(Y, X)$ and $\rho(X, Y) \in [-1, 1]$.
- $\rho(X, Y) = 1$ implies that X and Y are dependent.

It detects (only!) linear dependencies between two variables.



Part II

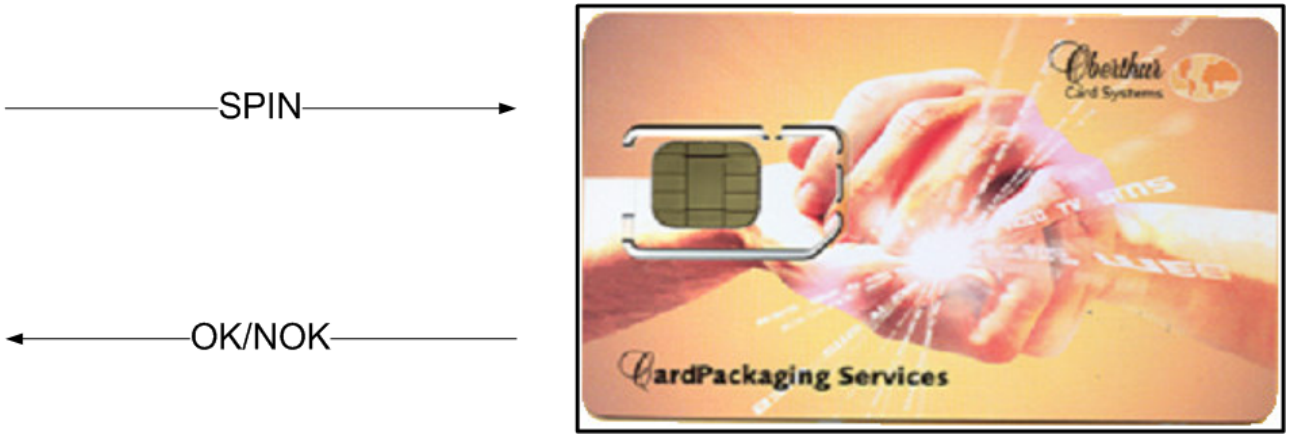
Timing

Outline

Timing attack

- Measure the timing of the operation
- Exploit this information to recover secret data

PIN verification

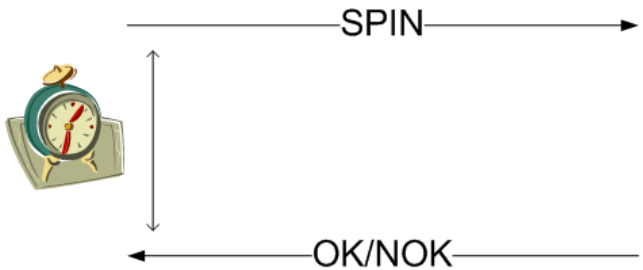


Algo PIN comparison

INPUT(S) : SPIN, PIN
OUTPUT(S) : `ok/nok`

```
1: for i = 0 to 4 do
2:     if SPIN[i] ≠ PIN[i] then
3:         return nok
4: return ok
```

PIN verification

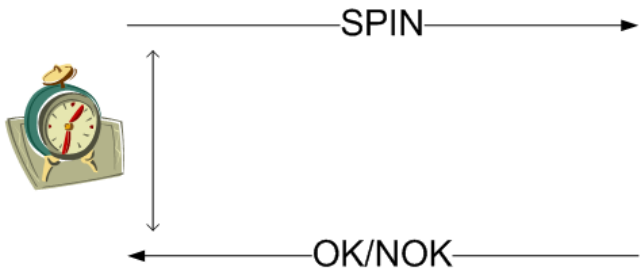


Algo PIN comparison

INPUT(S) : SPIN, PIN
OUTPUT(S) : `ok/nok`

```
1: for i = 0 to 4 do
2:     if SPIN[i] ≠ PIN[i] then
3:         return nok
4: return ok
```

PIN verification



Algo PIN comparison

INPUT(S) : SPIN, PIN
OUTPUT(S) : `ok/nok`

```
1: for i = 0 to 4 do
2:     if SPIN[i] ≠ PIN[i] then
3:         return nok
4: return ok
```

The observation of execution timing enables to retrieve PIN with 4×10 tries instead of $10^4 = 10\,000$.

AES: Brief History

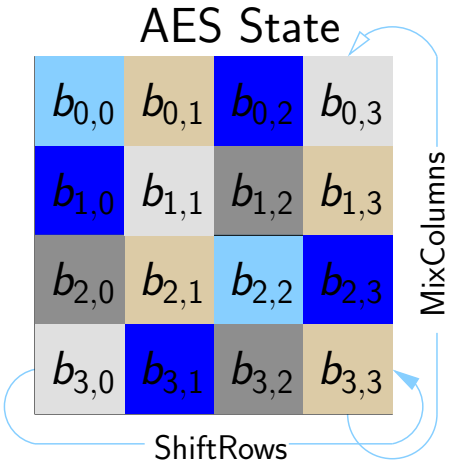
- Advanced Encryption Standard is the result of an international competition started in 1997.
- The purpose was to replace the DES standard since its key size was becoming too small.
- Some of the new requirements were: a block size equal to 128 bits (or 256 bits) and a scalable key size with possible values 128, 192 or 256.
- Among the 15 last candidates, the proposal of the Belgium researchers Daemen and Rijmen has been selected. It was called RIJNDAEL.

AES: Short Description

- Iterative block cipher (but not based on a Feistel structure).
- Substitution and Permutation Network (SPN).
  1. the state $Y(i)$ produced by the previous round is modified thanks to a non-linear substitution operation (that ensures data confusion).
  2. Then a linear permutation is applied (that ensures data diffusion).
  3. Eventually, the round key is bit-wisely added to produce $Y(i+1)$.
- The number of rounds is 10 for a 128-bits key and 14 for a 256-bits key.

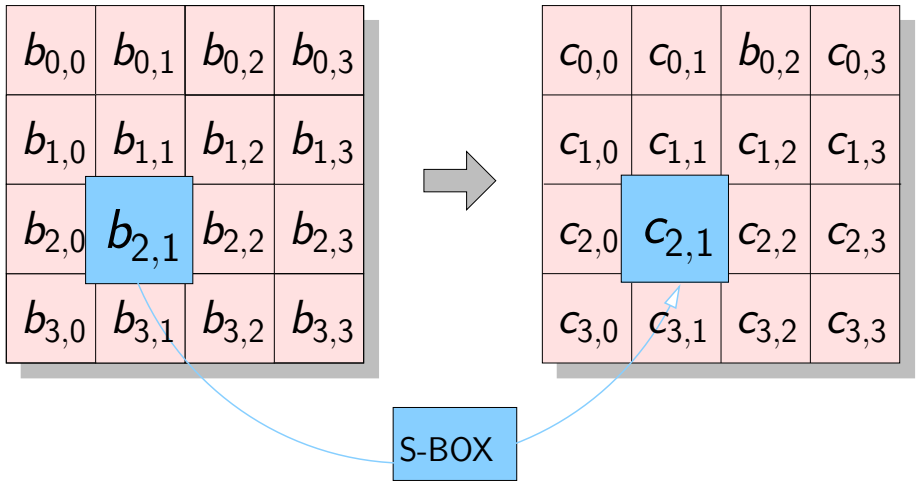
AES: details

- AES is an iterative block cipher with key sizes 128, 192 or 256 bits.
- Based on a Permutation/Substitution network. Each round is composed of:
  - A byte substitution (SubBytes).
  - A shifting of the bytes (ShiftRows)
  - A matrix product (MixColumns).
  - A bitwise addition with the round key (AddRoundKey).



AES

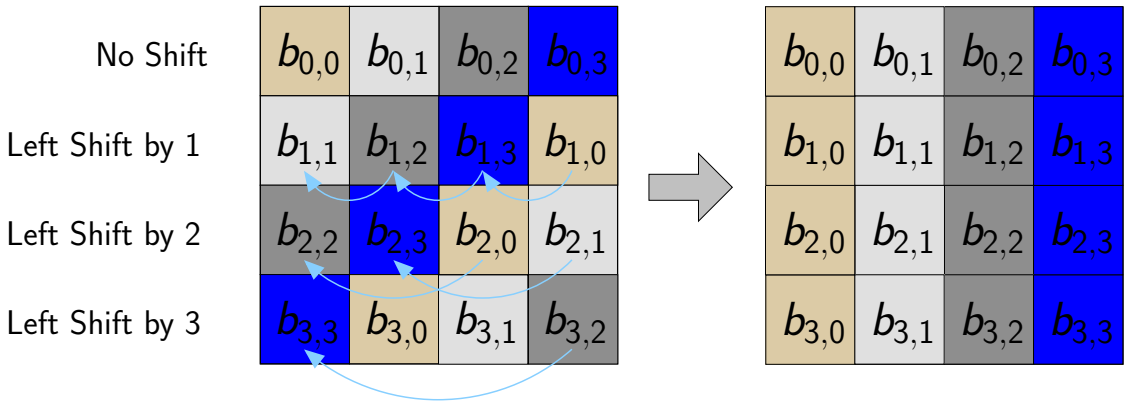
SubBytes



- Each byte is replaced by a value stored in a table of 256 bytes. This table is called s-box.

AES

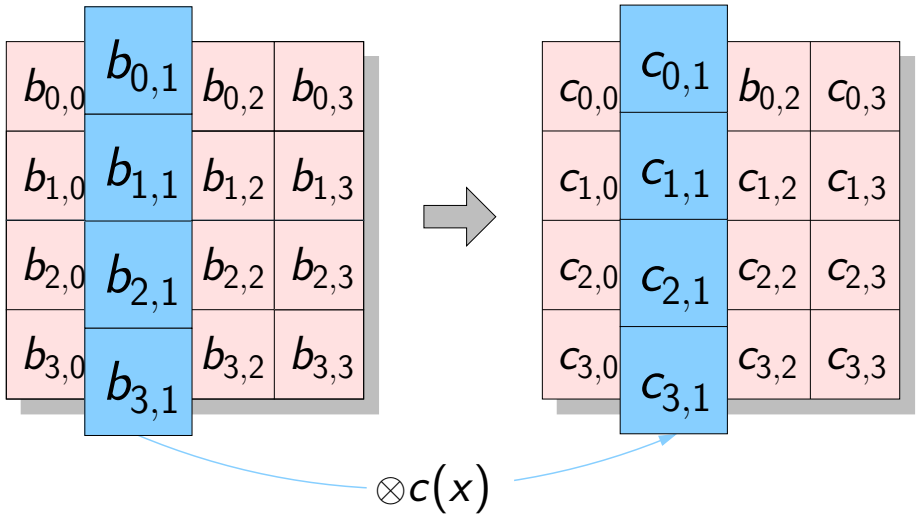
ShiftRows



- The bytes of each row are permuted. The permutation is simply left cycling.

AES

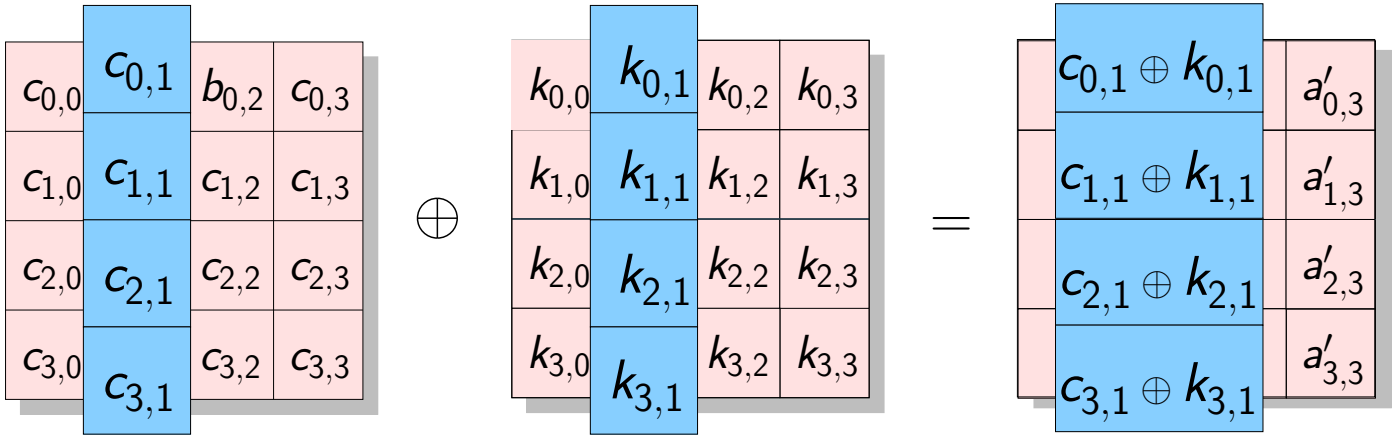
MixColumns



- The columns of the state are considered as polynomials over GF(256)
- Each column is multiplied by the polynomial defined on GF(256) by $c(X) = 3X^3 + X^2 + X + 2 \mod X^4 + 1$.
- For the operations on GF(256), the representation $GF(2)[X]/(X^8 + X^4 + X^3 + X + 1)$ is used.

AES

AddRoundKey



- The round key is bitwisely added.
- The four previous steps are repeated 10 times (for a 128-bits key).
- For the last round, the Mixcolumns step is not performed.

Mixcolumns Implementation

- Each column is multiplied by the polynomial defined on GF(256) by $c(X) = 3X^3 + X^2 + X + 2 \mod X^4 + 1$.
- For the operations on GF(256), the representation $GF(2)[X]/(X^8 + X^4 + X^3 + X + 1)$ is used.
- Gotta implement a multiplication of two elements of GF(256)!

Implementation method

Logtable

- Gotta implement a multiplication:
- if $(a.b)$
  - return $Alogtable[(Logtable[a] + Logtable[b]) mod 255]$;
- else
  - return 0

Chosen plaintext

$K = 3224BE84E16CD6AE529049F1F1BBE9EB$

- $P = 00000000000000000000000000000000$
- First MC: $(32, 24, BE, 84)$

Chosen plaintext

$K = 3224BE84E16CD6AE529049F1F1BBE9EB$

- $P = 00000000000000000000000000000000$
- First MC: $(32, 24, BE, 84)$
- $P = 01000000000000000000000000000000$
- First MC: $(33, 24, BE, 84)$

Chosen plaintext

$K = 3224BE84E16CD6AE529049F1F1BBE9EB$

- $P = 00000000000000000000000000000000$
- First MC: $(32, 24, BE, 84)$
- $P = 01000000000000000000000000000000$
- First MC: $(33, 24, BE, 84)$
- $P = 32000000000000000000000000000000$
- First MC: $(00, 24, BE, 84)$

Conclusion

- Information about secret values can be retrieved.
- Countermeasure: *returnmultTab*[*a*][*b*]

Part III

Sensitive Variables, Leakage Functions and Model Inference

Outline

Outline

Crypto Primitives and Implementations

Introduction

Different abstraction levels:

- Mathematical level
- Algorithmic level
- Hardware level

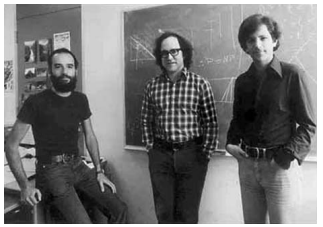
First Example

Modular exponentiation

- The modular exponentiation is the main operation during the RSA processing and other algorithms:
  - Diffie-Helmann
  - DSA
  - primality tests
- This is an operation costly both in terms of memory and timing:
  $\Longrightarrow$ Need for optimisation
- This is an operation that manipulates secret data:
  $\Longrightarrow$ Need for security

RSA algorithm

Mathematical level



RSA primitive

Let p, q be two prime numbers and let $N = pq$.
Let $e, d \in \mathbb{Z}_{\phi(N)}$ be such that $ed = 1 \mod \phi(N)$.

Public key $PK = (N, e)$	modulus N, *public exponent* e
Private key $SK = (N, d)$	private exponent d

Encryption	Decryption
$c = m^e \mod N$	$m = c^d \mod N$
Signature	**Verification**
$s = m^d \mod N$	$m \stackrel{?}{=} s^e \mod N$

RSA algorithm

Algorithmic level, *e.g. Square-and-Multiply*

Observation :

$$d = \sum_{i=0}^{i=n} d_i 2^i = d_0 + 2(d_1 + 2(d_2 + \ldots + 2(d_{n-1} + 2(d_n))) \ldots)$$

$$m^d = m^{d_0} \times (m^{d_1} \times (m^{d_2} \times \ldots (m^{d_{n-1}} \times (m^{d_n})^2)^2 \ldots)^2$$

RSA algorithm

Algorithmic level, *e.g.* *Square-and-Multiply*

Algo S&M binary from left to right

INPUT(S) : $m, d = (d_n d_{n-1} \dots d_1 d_0)_2$
OUTPUT(S) : m^d

1: $t \leftarrow 1$
2: **for** $i = n$ to 0 **do**
3: $\quad t \leftarrow t \times t$
4: $\quad$ **if** $d_i = 1$ **then**
5: $\quad\quad t \leftarrow t \times m$
6: **return** t

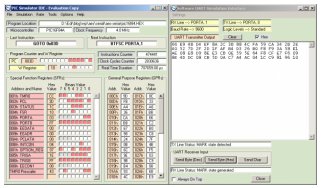
Operations :

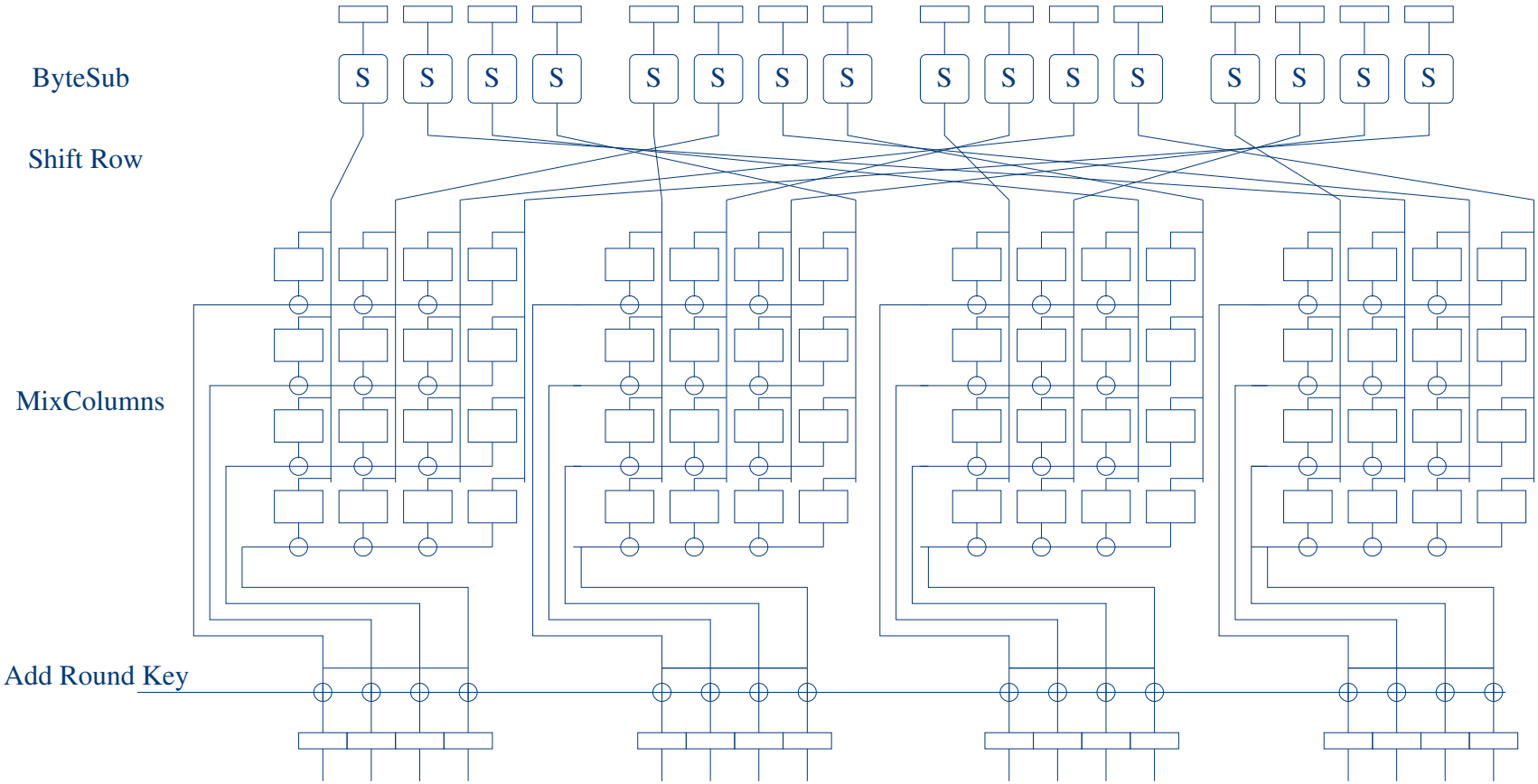
- n SQ
- $HW(d) - 1$ MULT.

Memory : m, t

AES algorithm

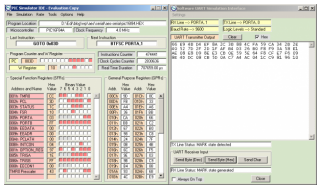
Algorithmic level, *e.g. 8-bit Software Implementation*

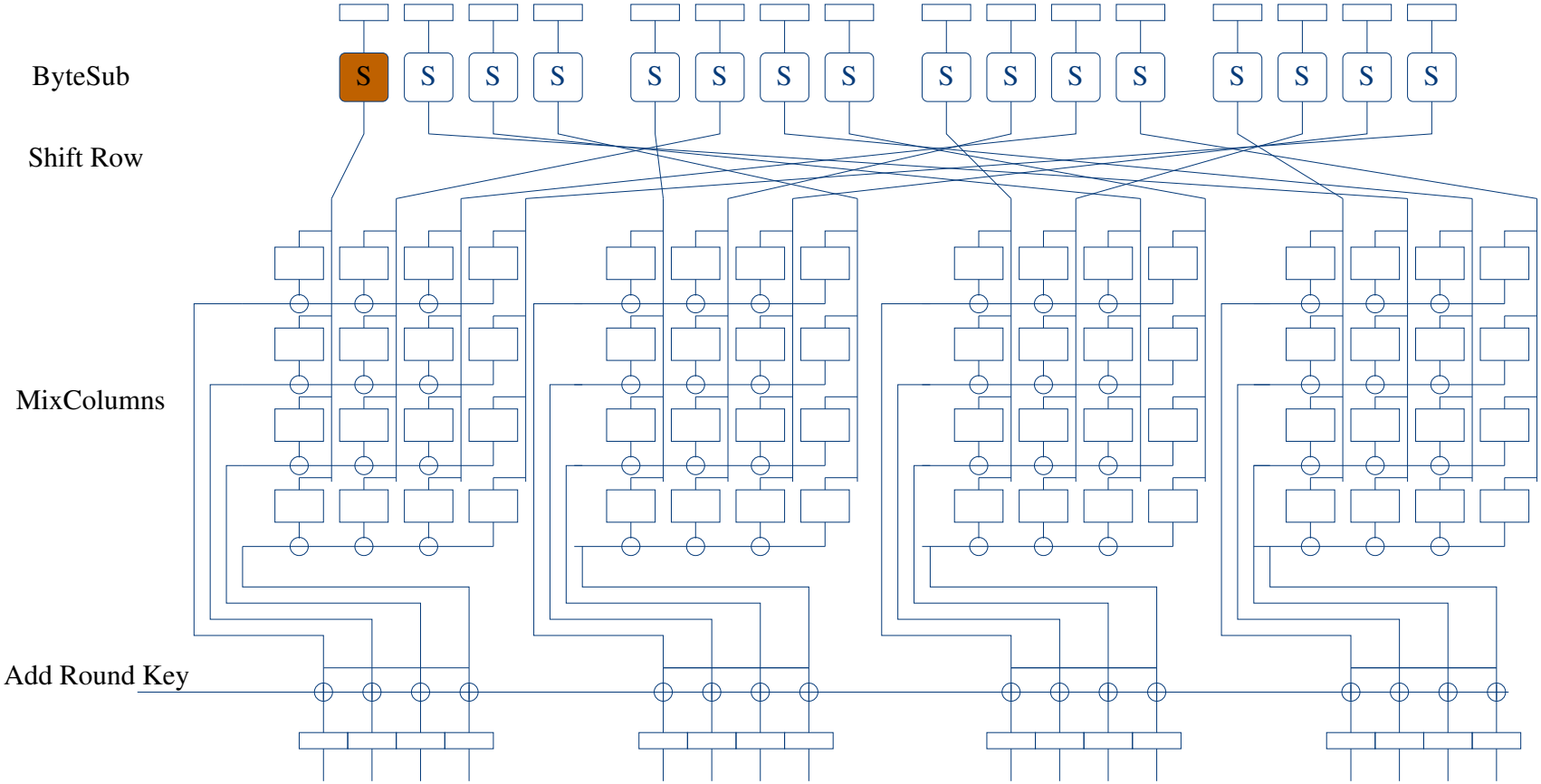




AES algorithm

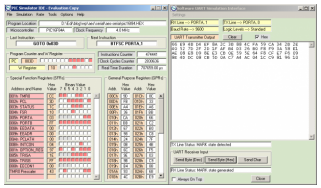
Algorithmic level, *e.g. 8-bit Software Implementation*

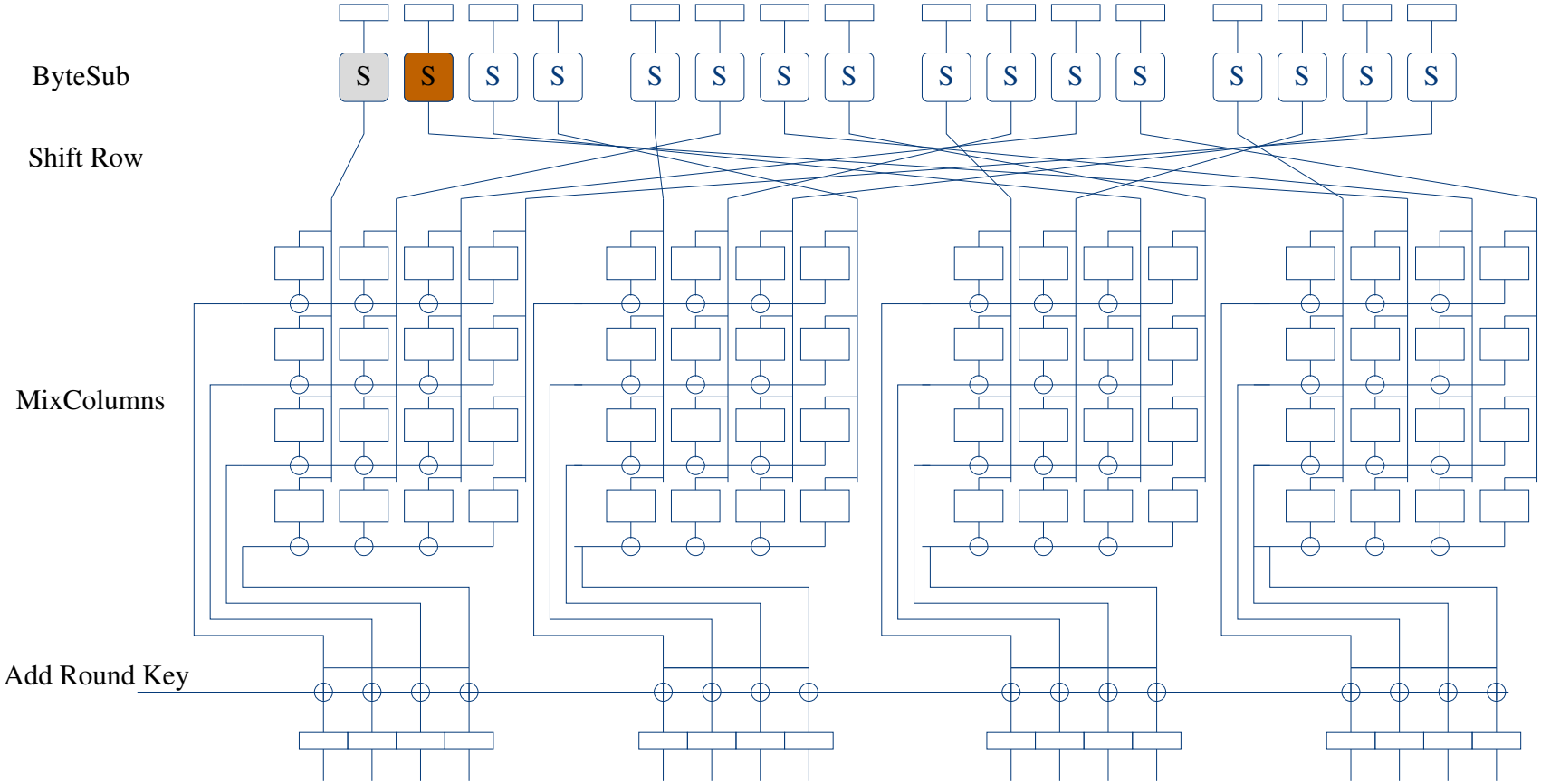




AES algorithm

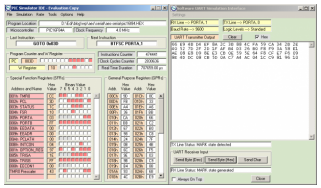
Algorithmic level, *e.g. 8-bit Software Implementation*

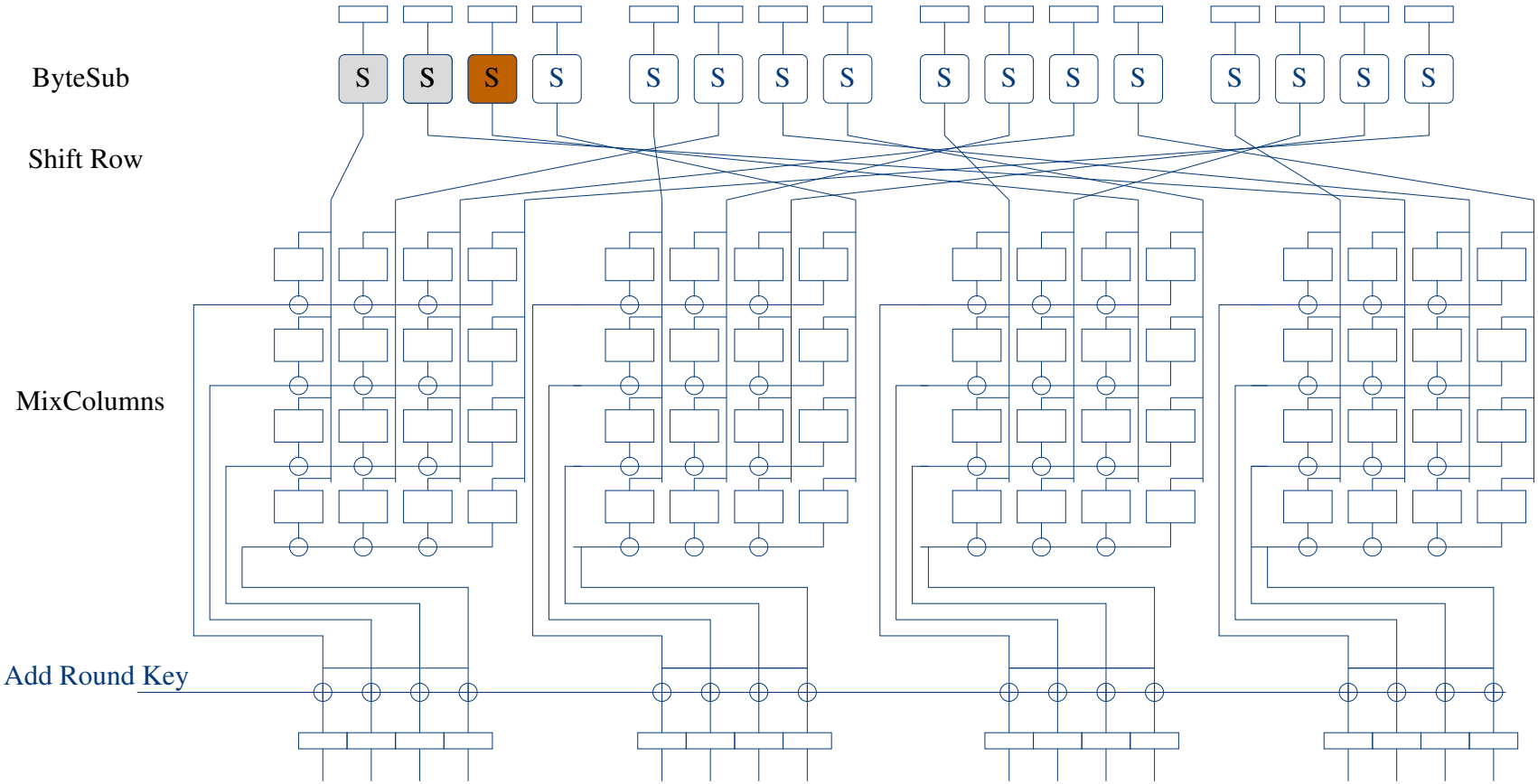




AES algorithm

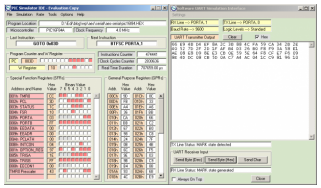
Algorithmic level, *e.g. 8-bit Software Implementation*

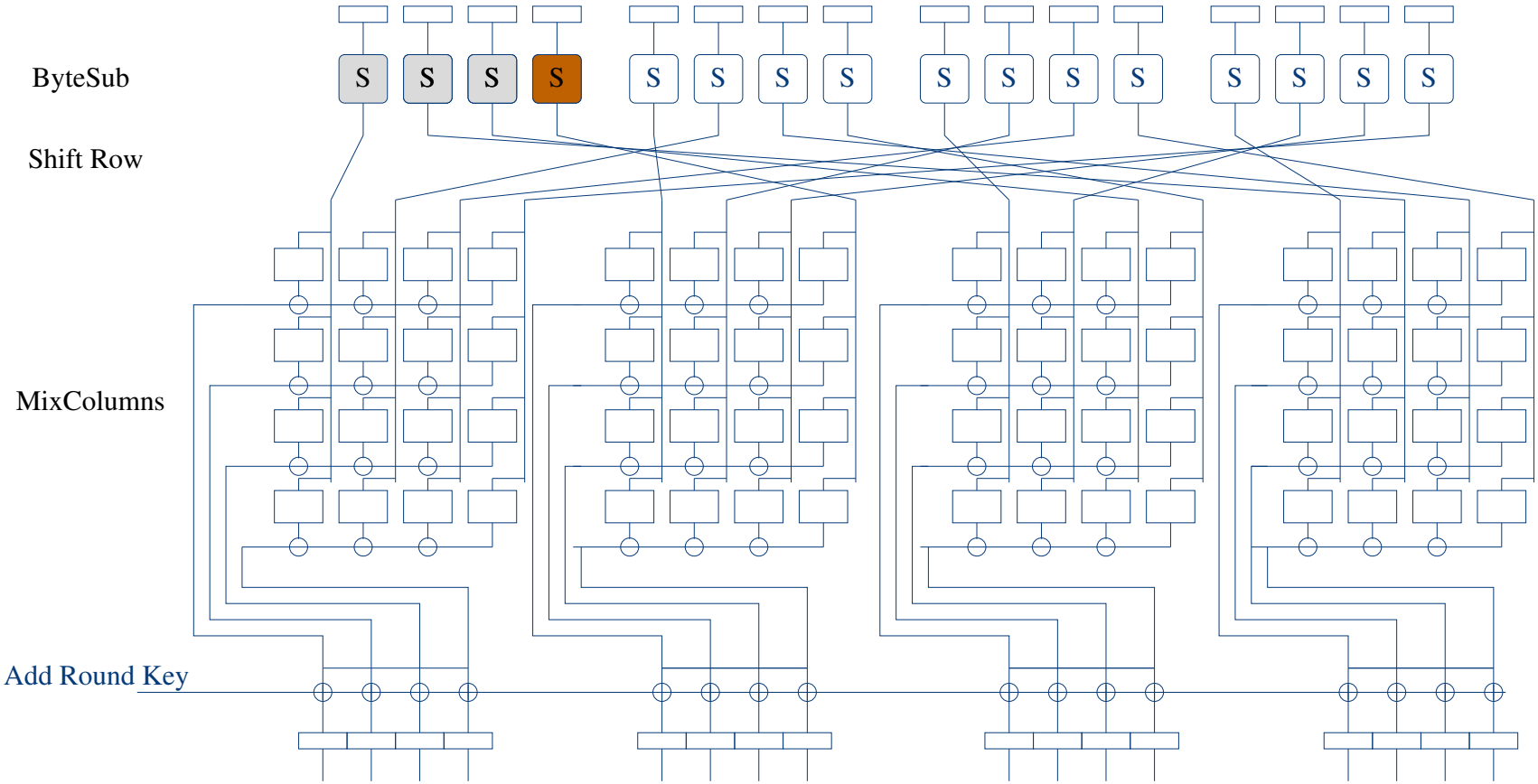




AES algorithm

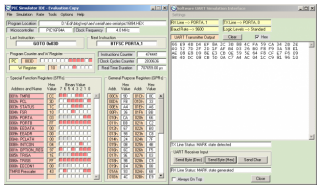
Algorithmic level, *e.g. 8-bit Software Implementation*

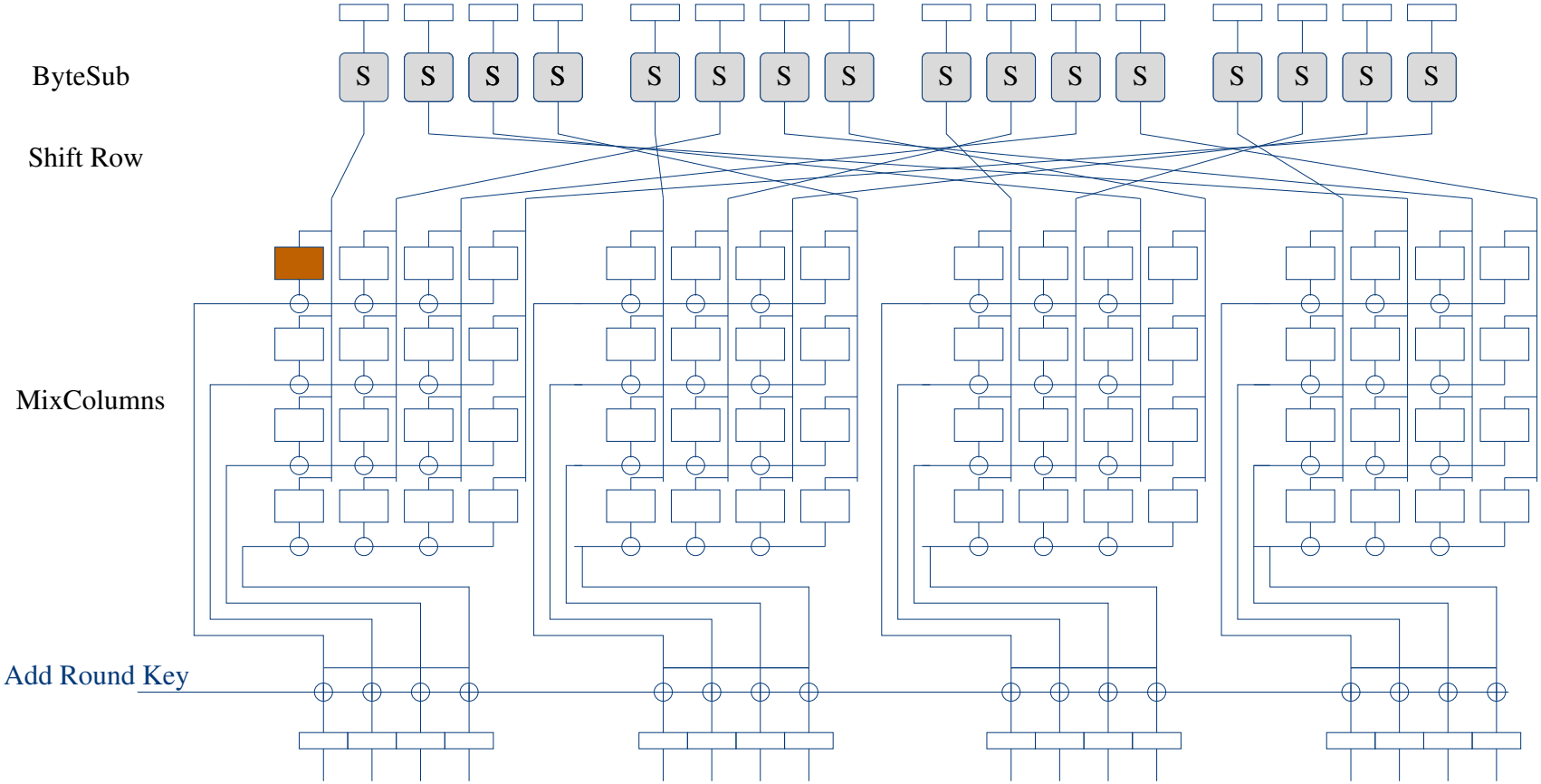




AES algorithm

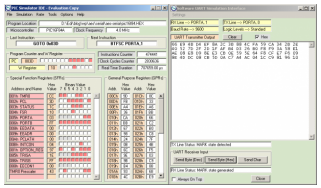
Algorithmic level, *e.g. 8-bit Software Implementation*

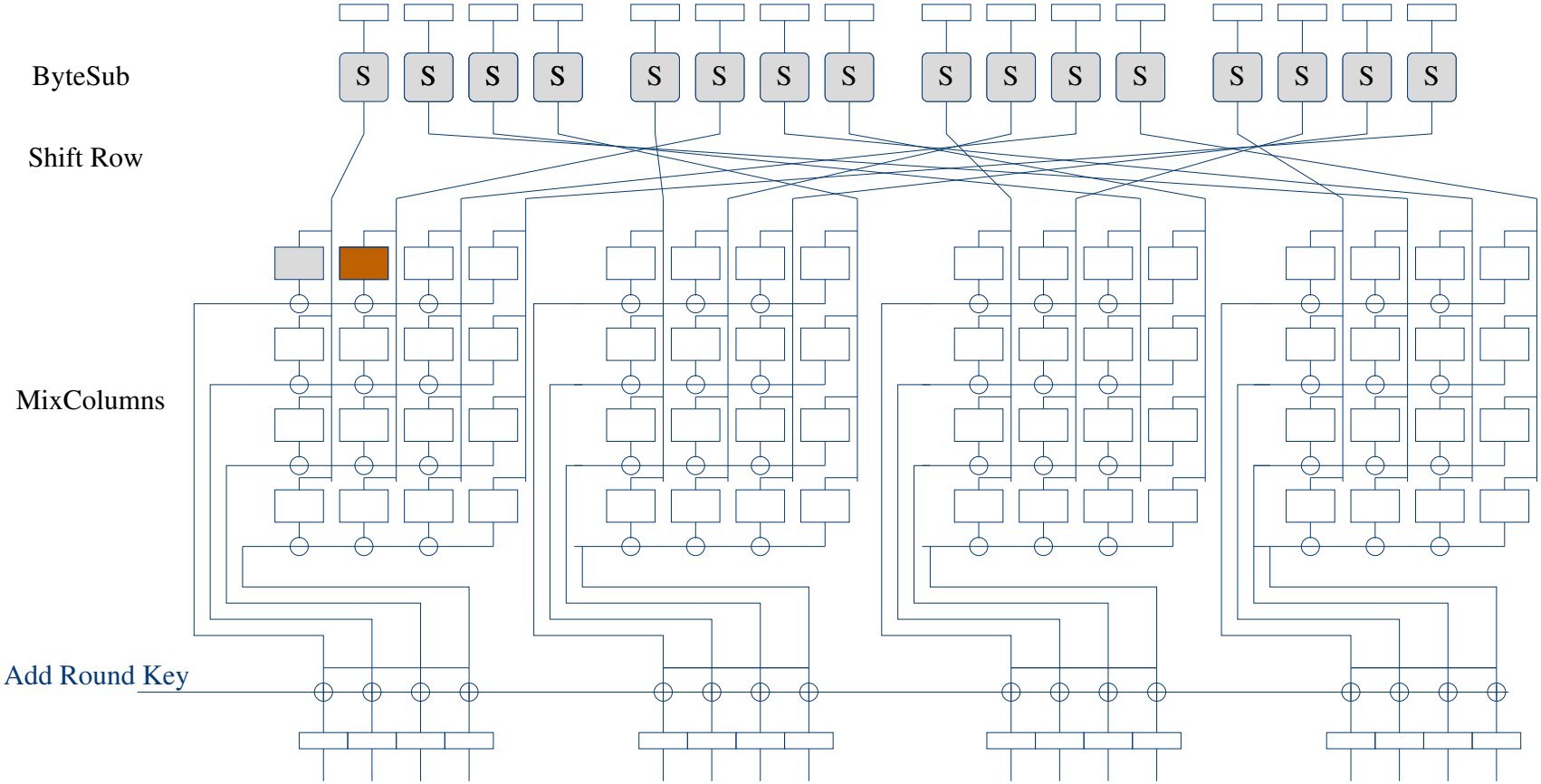




AES algorithm

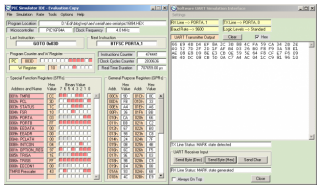
Algorithmic level, *e.g. 8-bit Software Implementation*

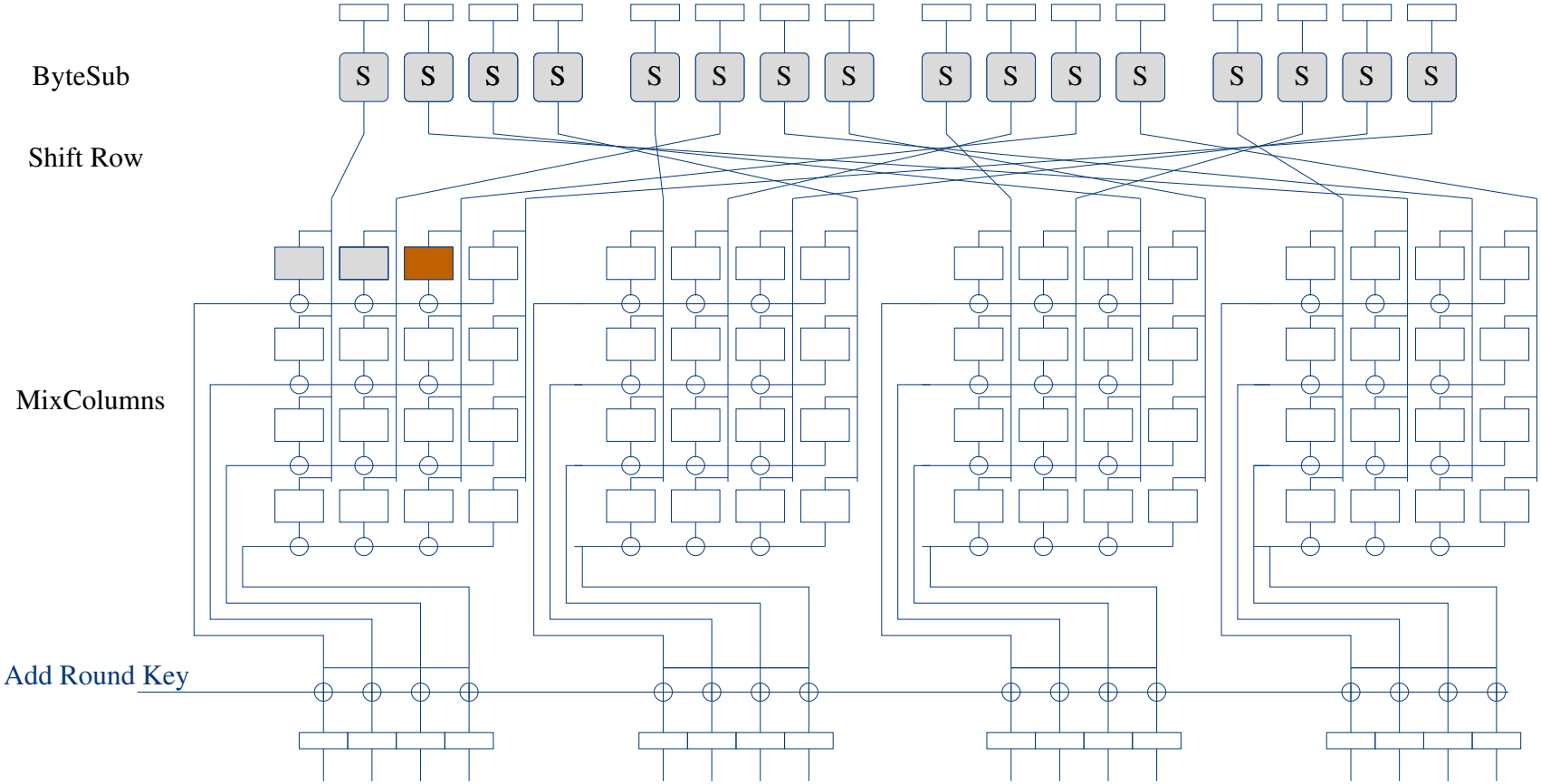




AES algorithm

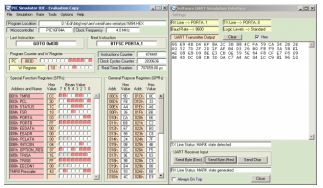
Algorithmic level, *e.g. 8-bit Software Implementation*

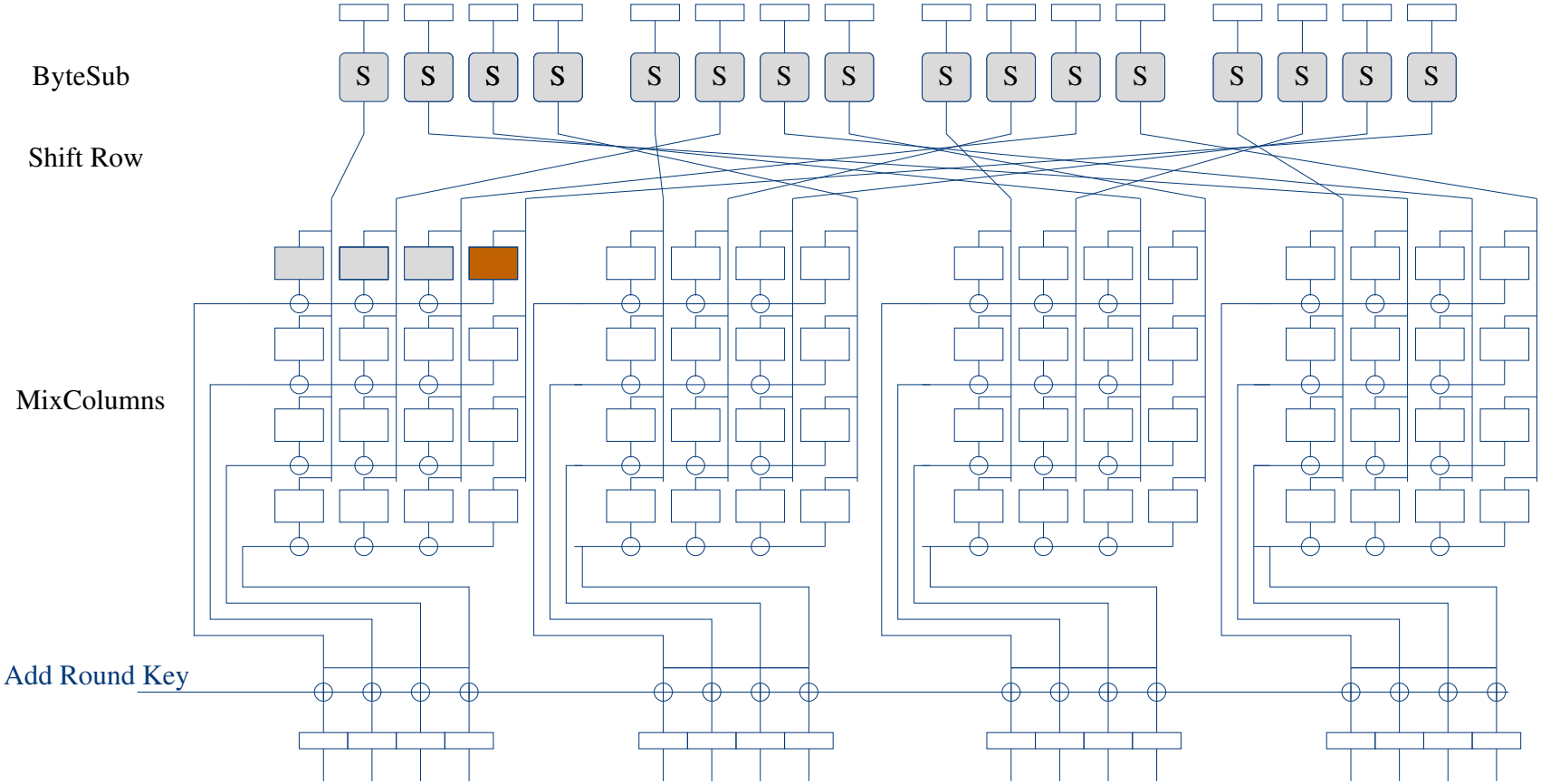




AES algorithm

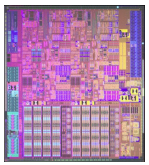
Algorithmic level, *e.g. 8-bit Software Implementation*

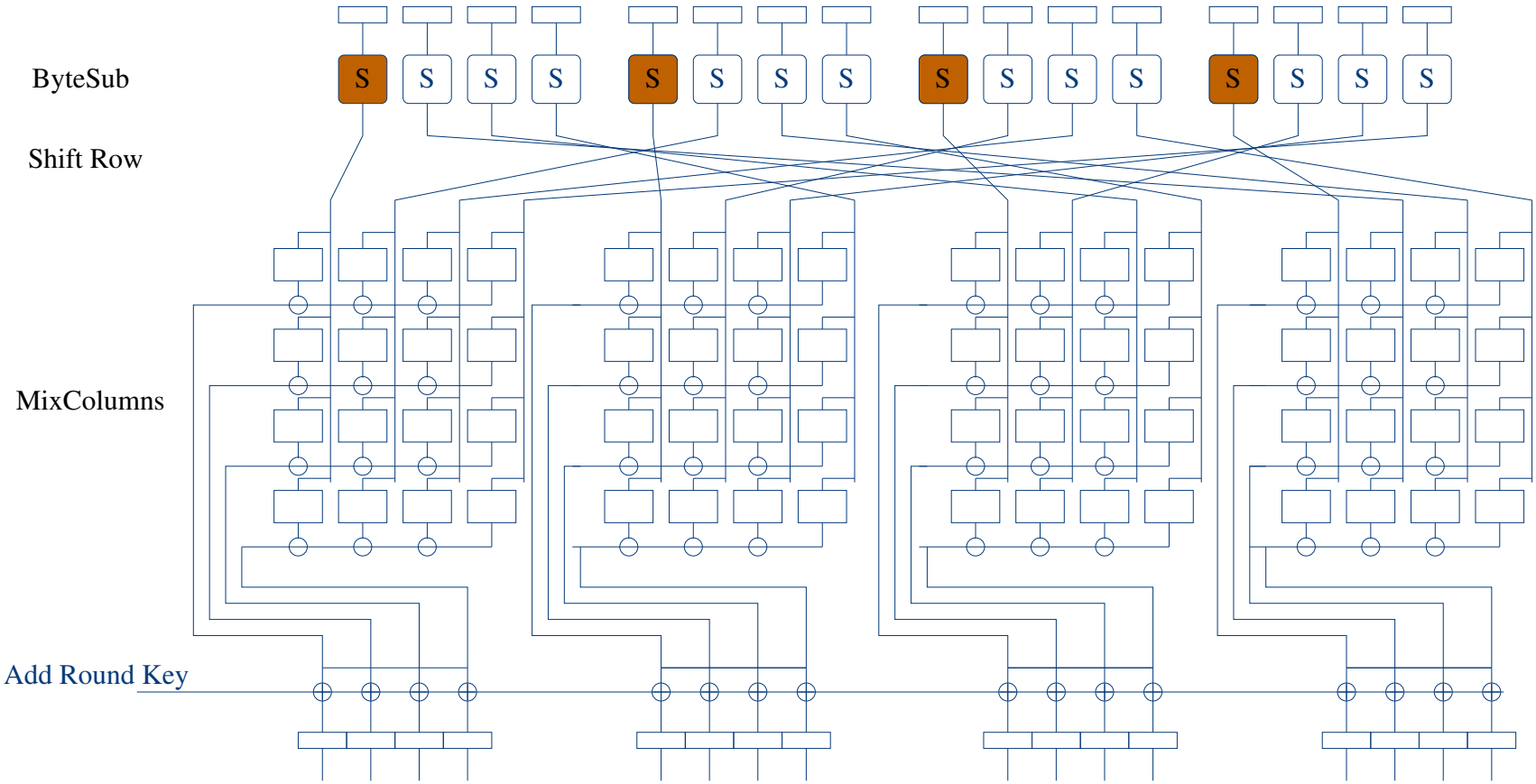




AES algorithm

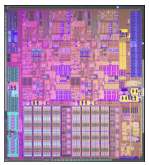
Algorithmic level, *e.g.* *Hardware Implementation*

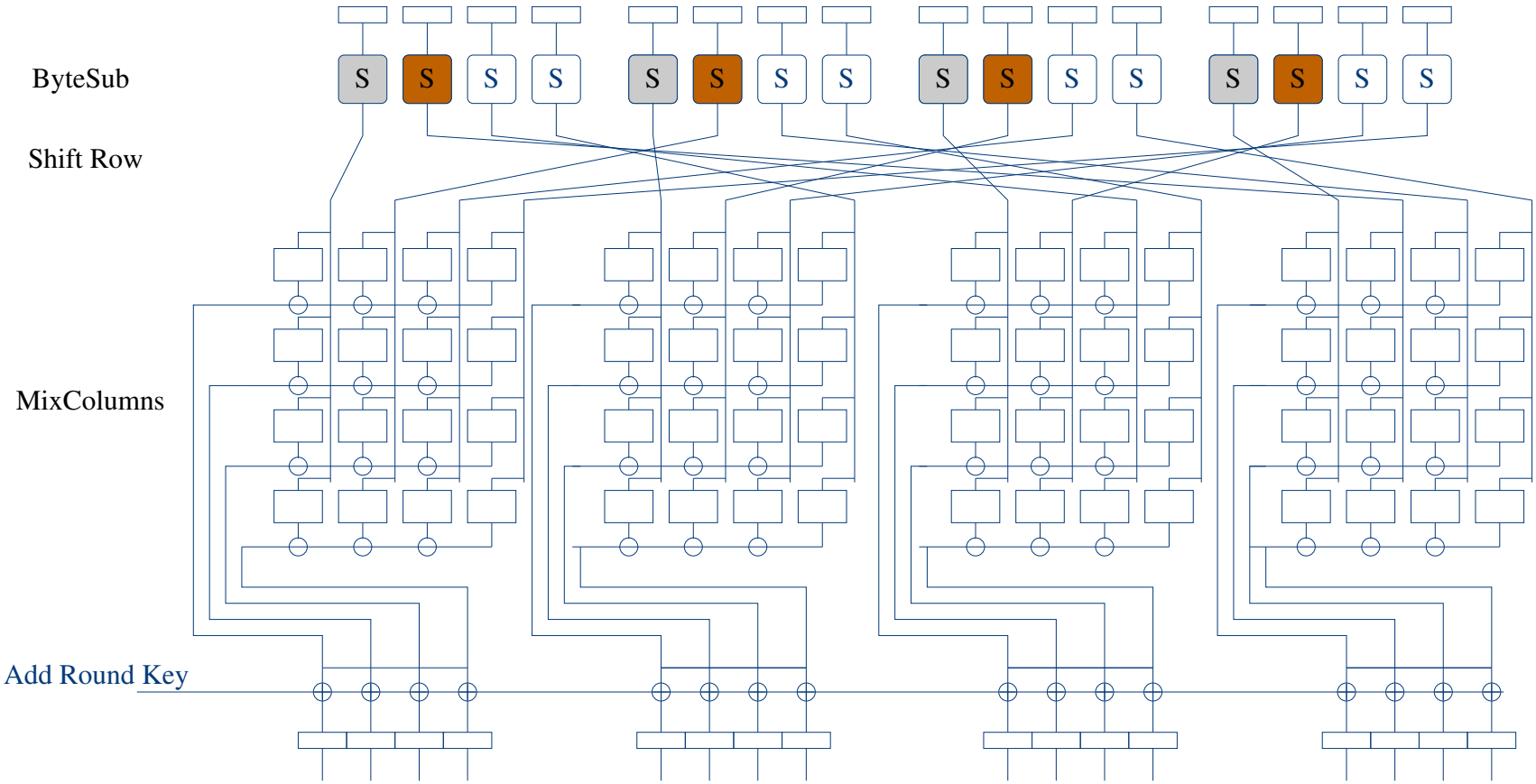




AES algorithm

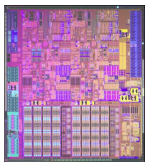
Algorithmic level, *e.g.* *Hardware Implementation*

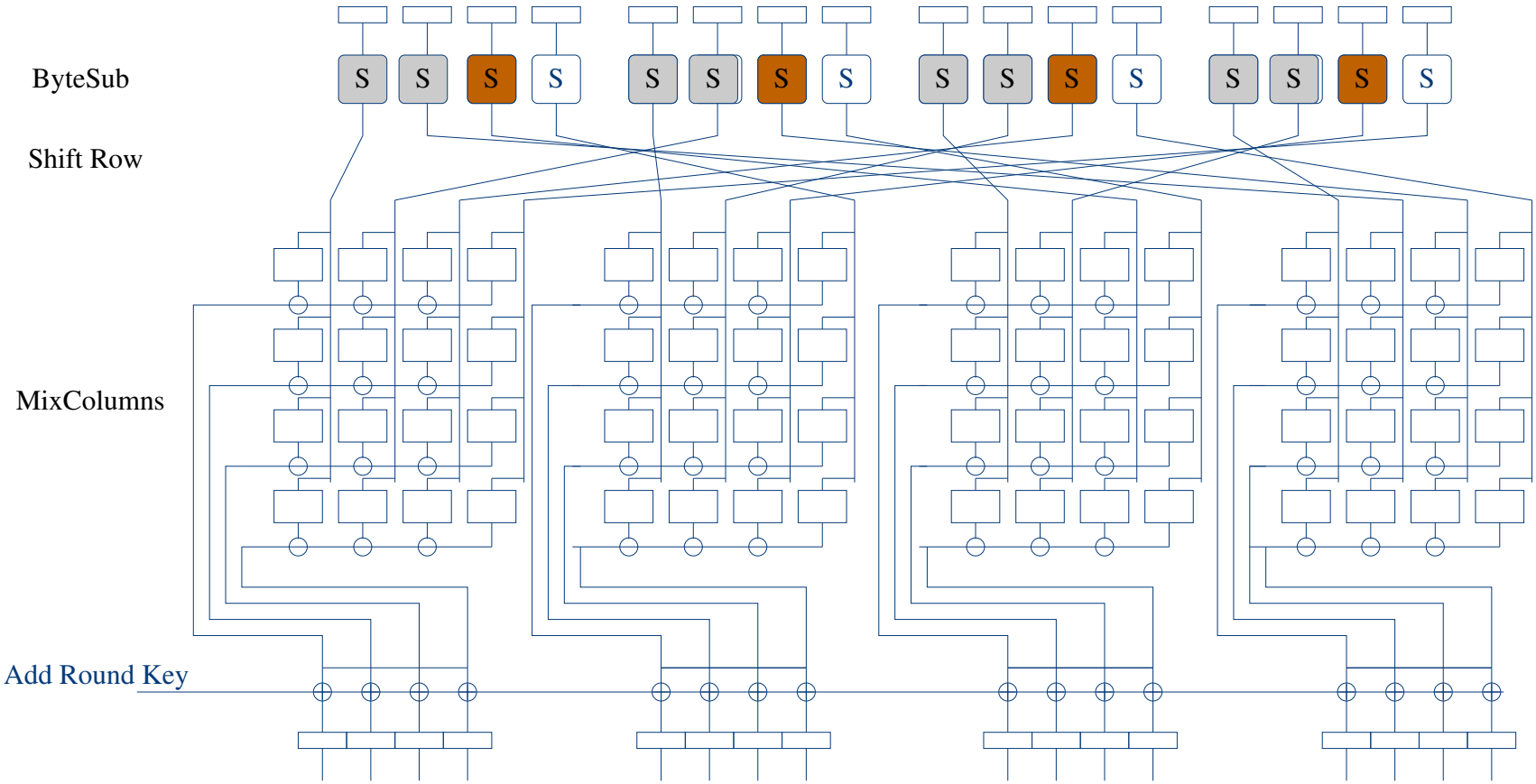




AES algorithm

Algorithmic level, *e.g.* *Hardware Implementation*





Crypto Primitives and Implementations

Intermediate Variables Recapitulation

Different kind of intermediate variables:

- Constants;

  e.g. number of rounds, key sizes

- *Public variables;*

  e.g. Plaintext or Ciphertexts

- Secrets: (part of) the secret key;

  e.g. d in RSA

- *Mix between Secret and Public variables.*

  e.g. $m^{d_{|\ell}}$ in RSA or $P \oplus K$ in AES

Crypto Primitives and Implementations

Intermediate Variables Recapitulation

Different kind of intermediate variables:

- Constants;

  e.g. number of rounds, key sizes

- *Public variables;*

  e.g. Plaintext or Ciphertexts

- Secrets: (part of) the secret key;

  e.g. d in RSA

- *Mix between Secret and Public variables.*

  e.g. $m^{d_{|\ell}}$ in RSA or $P \oplus K$ in AES

Crypto Primitives and Implementations

Limitations of Sensitive Variables

sensitive variables: $\mathrm{MI}(V, P, K) \neq 0$

we have $V = f(K, P, \cdots)$

- then the key often cannot be retrieved entirely from a single realization of (V, P)
  $\hookrightarrow$ several observations may improve the key coverage depending on f.

  e.g. AES AddKey/Sbox output.
- distinguishing effort

  e.g. AES SBox uniform differentiality

Quick Recap

Sensitive Variables

Best choice for f:

- as simple as possible;
- involving public variables;
- good distinguishing properties.

Classical choices $\log_2(\text{Secret}) \leq 32$

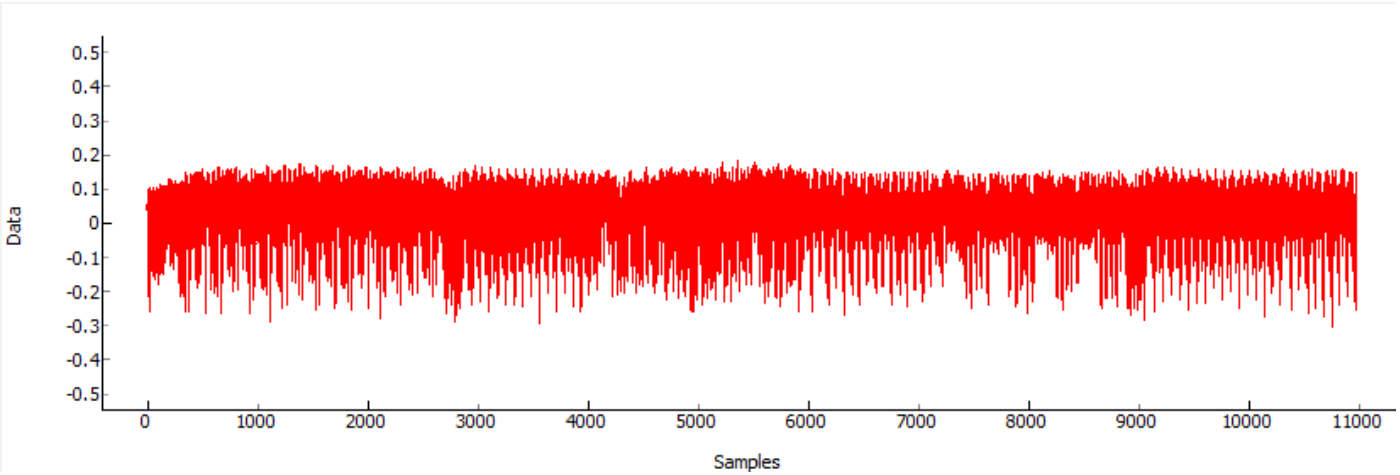
- AES: $K \oplus P$, $\text{Sbox}(K \oplus P)$.
- RSA: d, $m^{d_{|\ell}}$.

Outline

Leakage Function

The link b/w side-channel traces and sensitive variables

Consider a side-channel trace (from EM observation):

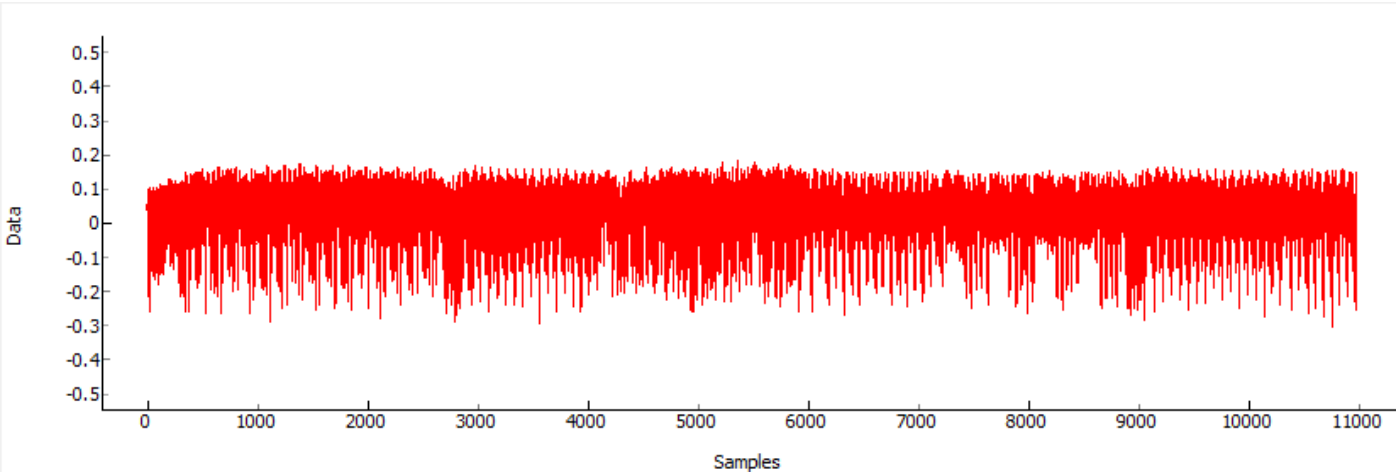


http://hackaday.io/project/956/log/10108-aes256-is-not-enough-breaking-a-bootloader

Leakage Function

The link b/w side-channel traces and sensitive variables

Consider a side-channel trace (from EM observation):



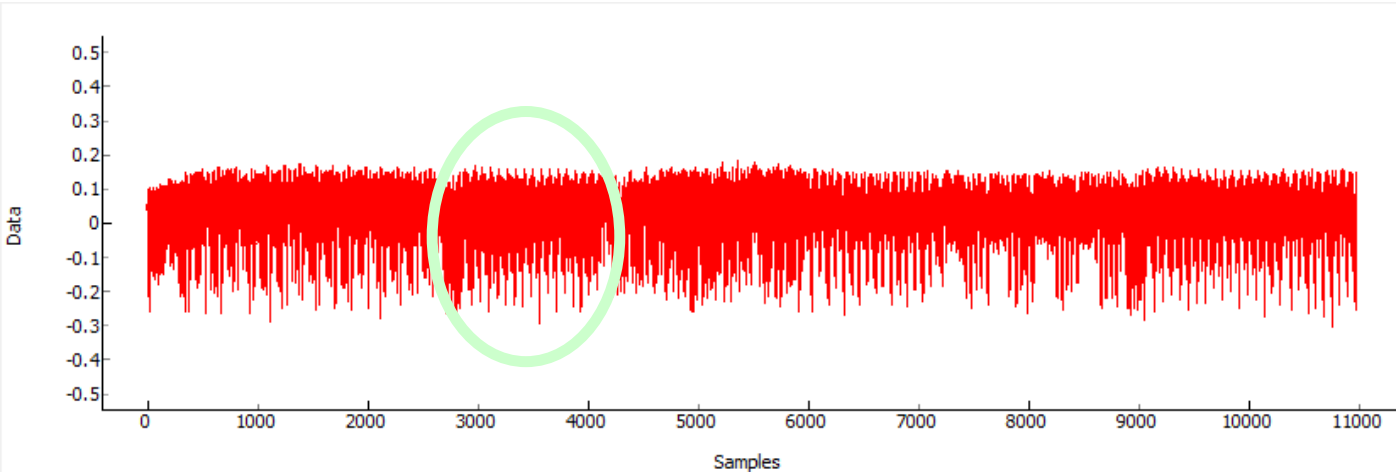
http://hackaday.io/project/956/log/10108-aes256-is-not-enough-breaking-a-bootloader

$$L[?] = \ell(\text{Sbox}(P \oplus K))$$

Leakage Function

The link b/w side-channel traces and sensitive variables

Consider a side-channel trace (from EM observation):



http://hackaday.io/project/956/log/10108-aes256-is-not-enough-breaking-a-bootloader

$$L[i] = \ell(\text{Sbox}(P \oplus K))$$

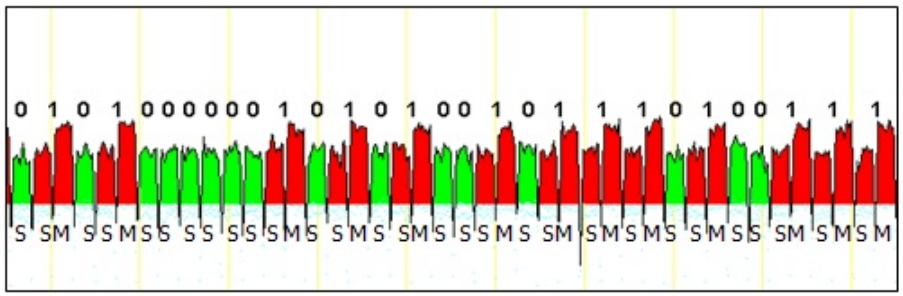
Leakage Function

A simple case: Unprotected Square-and-Multiply Algorithm

Algo S&M binary from left to right

INPUT(S) :
$m, d = (d_n d_{n-1} \dots d_1 d_0)_2$
OUTPUT(S) : m^d

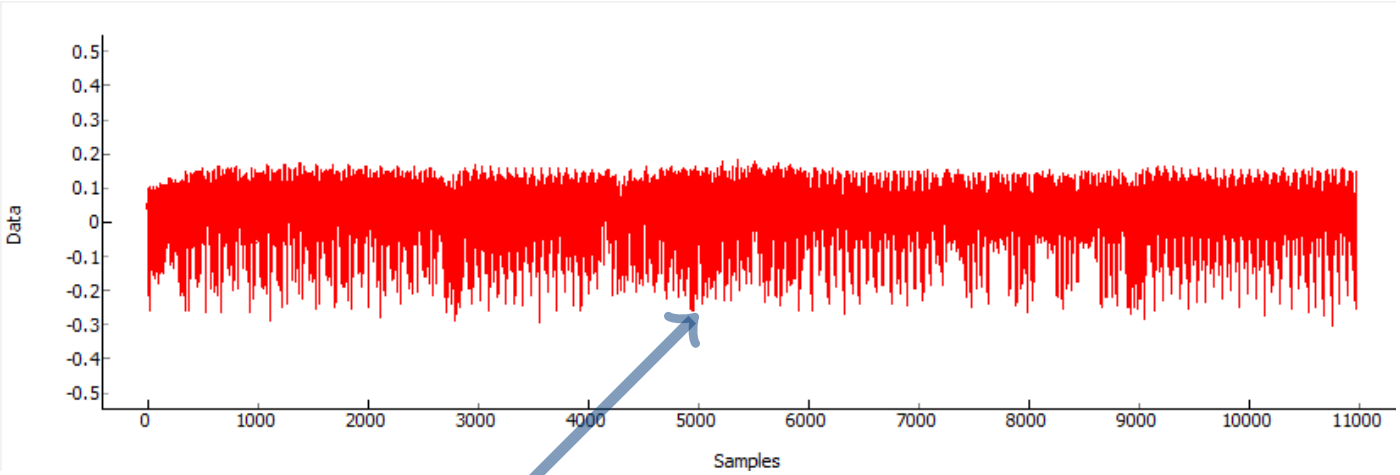
1: $t \leftarrow 1$
2: **for** $i = n$ to 0 **do**
3: $\quad t \leftarrow t \times t$
4: $\quad$ **if** $d_i = 1$ **then**
5: $\quad\quad t \leftarrow t \times m$
6: **return** t

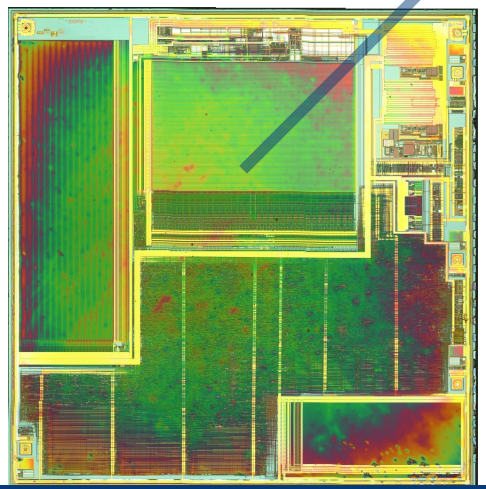


Remark/Assumption: squaring and multiplication have different leakage patterns.

Leakage Function

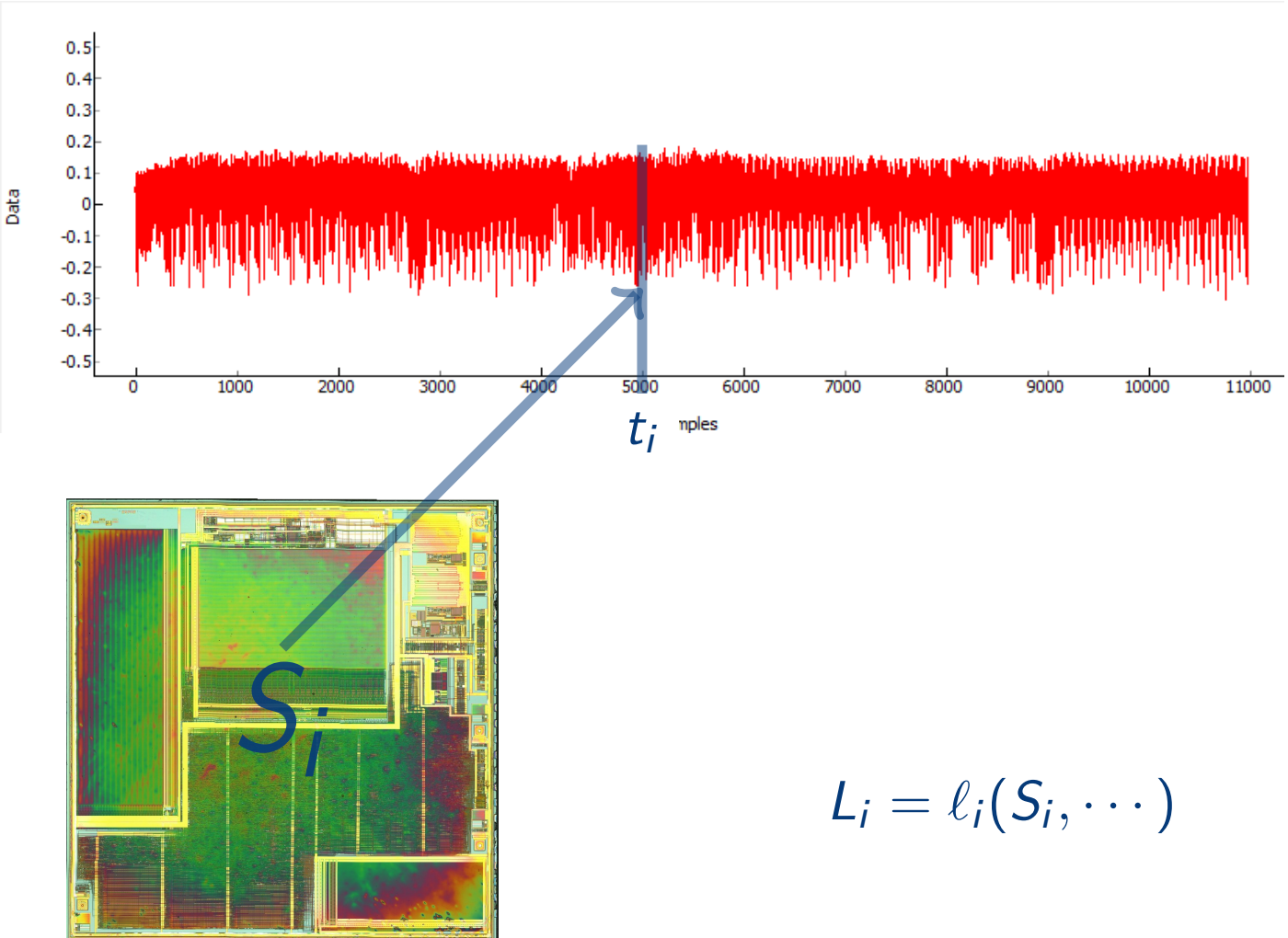
Formalism





Leakage Function

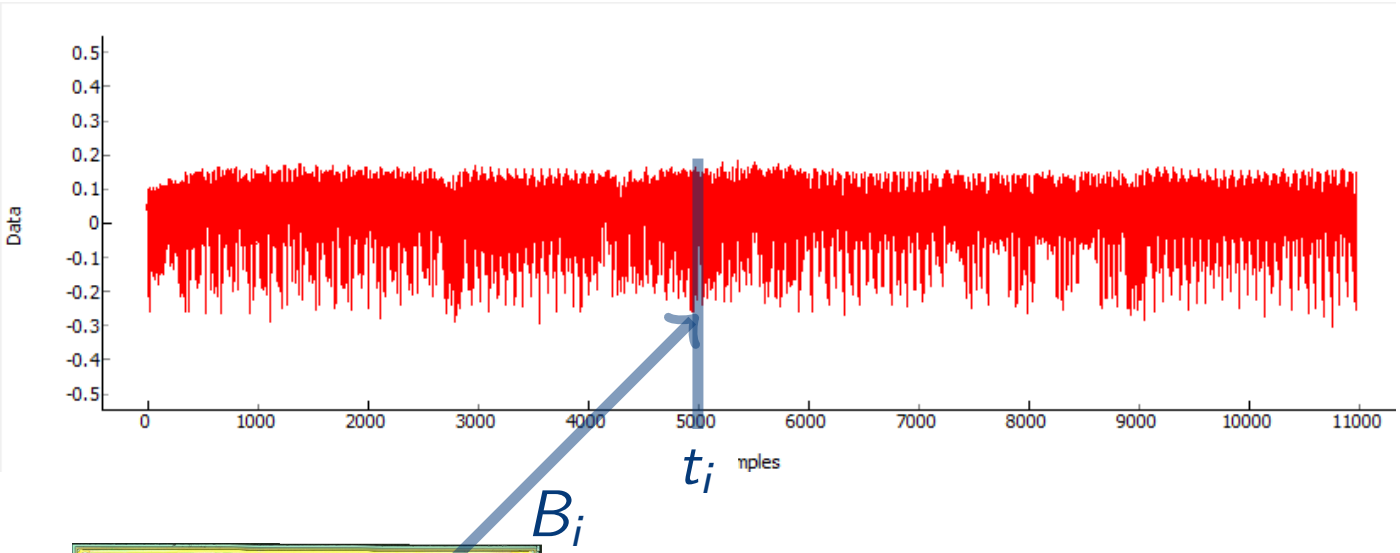
Formalism



$$L_i = \ell_i(S_i, \cdots)$$

Leakage Function

Formalism

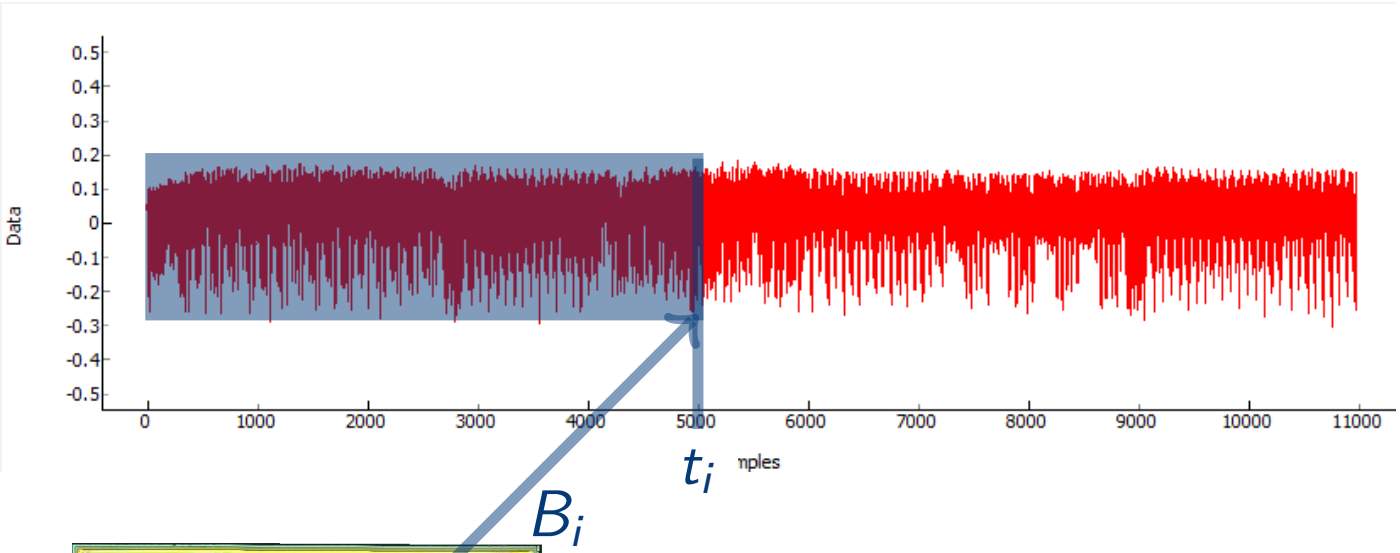


$$L_i = \ell_i(S_i, B_i, \cdots)$$

B_i follows an unknown distribution law $\mathcal{B}_{S_i}$.

Leakage Function

Formalism



$$L_i = \ell_i(S_i, B_i, \{S_j, B_j\}_{j<i})$$

B_j follows an unknown distribution law $\mathcal{B}_{S_j}$.

Leakage Function

Only Computation Leaks Assumption MicaliReyzin2004

Not a perfect model of the reality MoradiMischke2013 but quite a good approximation.

$$L_i = \ell_i(S_i, B_i)$$

Leakage Function

Only Computation Leaks Assumption MicaliReyzin2004

Not a perfect model of the reality MoradiMischke2013 but quite a good approximation.

$$L_i = \ell_i(S_i, B_i)$$

Then, by sending all useless information from S_i to B_i

$$L_i = \ell_i(V, B_i)$$

where B_i follows an unknown distribution law $\mathcal{B}_{S_i}$.

B_i is the sum of the measurement and algorithmic noise.

Leakage Function

Independent additive Gaussian Noise Assumption

When more precise model is needed

e.g. for security proofs or theoretical evaluation of an attack.

$$L_i = \varphi_i(V) + B_i$$

with $B_i \hookleftarrow \mathcal{N}(\mu_i, \sigma_i)$.

$\varphi()$ *denotes the deterministic part of* $\ell()$

Leakage Function

Independent additive Gaussian Noise Assumption

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

$$F_L(x) = \sum_i \Pr[\varphi(V) = i]\mathcal{N}(i, \sigma^2)(x)$$

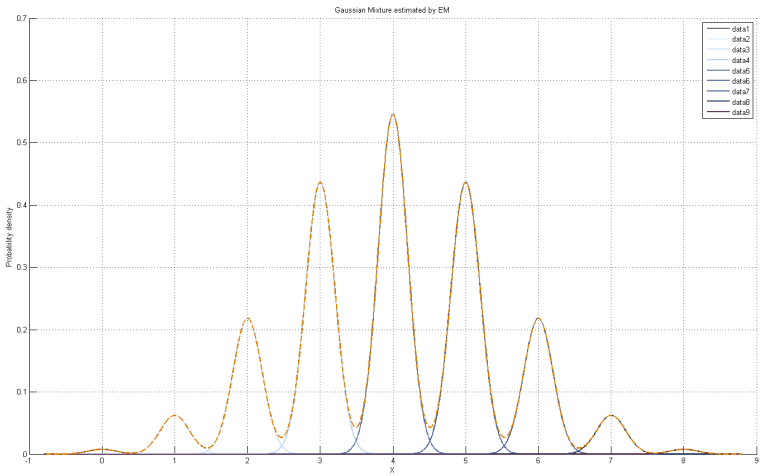


Figure: ($\sigma = 0.2$)

Leakage Function

Independent additive Gaussian Noise Assumption

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

$$F_L(x) = \sum_i \Pr[\varphi(V) = i]\mathcal{N}(i, \sigma^2)(x)$$

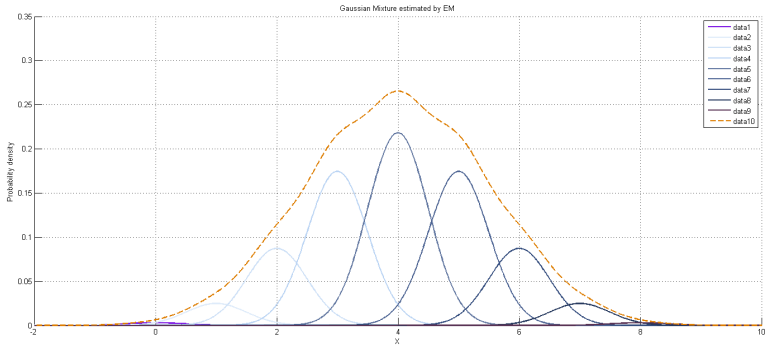


Figure: Small noise ($\sigma = 0.5$)

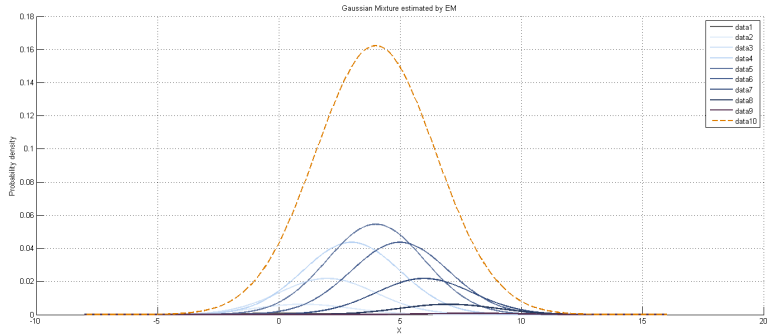


Figure: Medium noise ($\sigma = 2$)

Quick Recap

Leakage Functions

In practice very little knowledge about it but accurate models are needed for

- some side-channel attacks;
- side-channel resistance proofs.

Quick Recap

Leakage Functions

In practice very little knowledge about it but accurate models are needed for

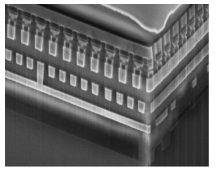
- some side-channel attacks;
- side-channel resistance proofs.

$\hookrightarrow$ How to infer a leakage model?

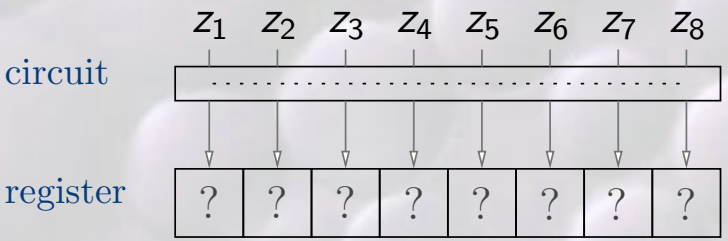
Outline

Electrical engineer intuition

CMOS technology behaviour



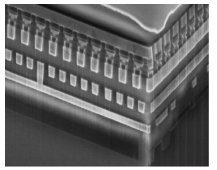
$$Z = S(M + k)$$



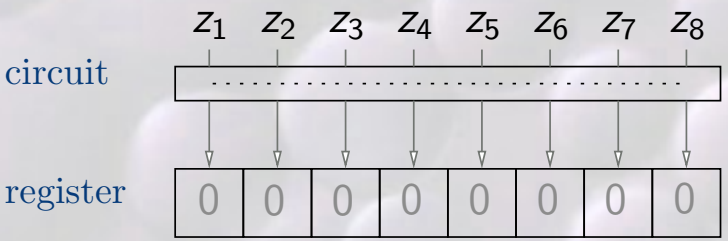
Assumption

Electrical engineer intuition

CMOS technology behaviour



$$Z = O(M + k)$$

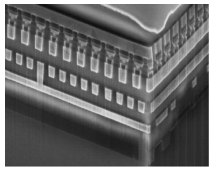


Assumption

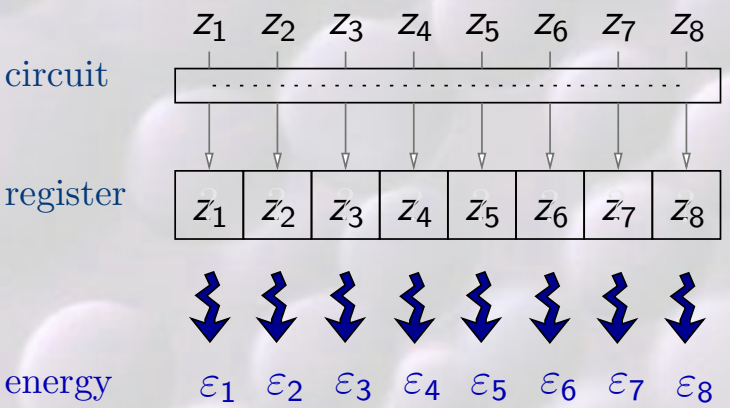
0 Precharge

Electrical engineer intuition

CMOS technology behaviour



$Z = O(M + k)$



Assumption

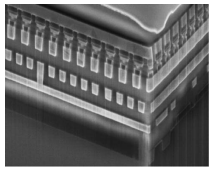
0 Precharge

Linear Regression

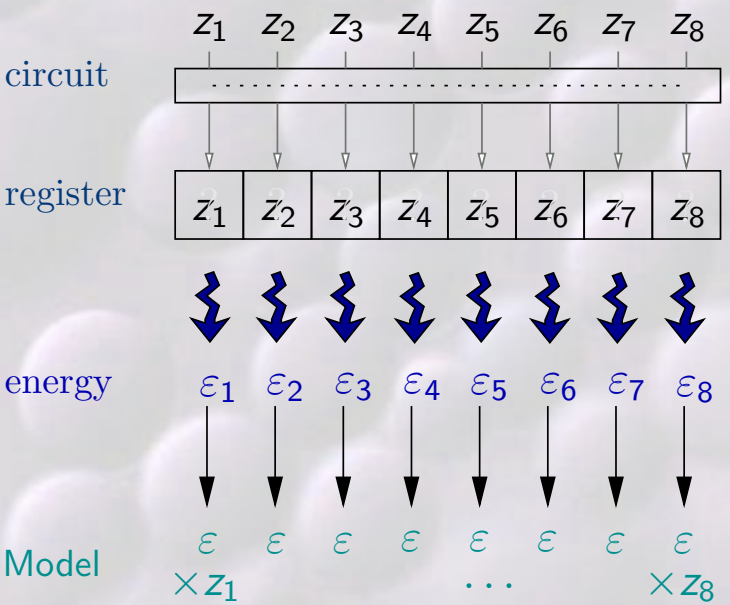
the ε_i's are indep.

Electrical engineer intuition

CMOS technology behaviour



$Z = O(M + k)$



$Y = \varepsilon \times \mathrm{HW}(Z)$

Assumption

0 Precharge

Linear Regression

the ε_i's are indep.

Hamming Weight Model

Electrical engineer intuition

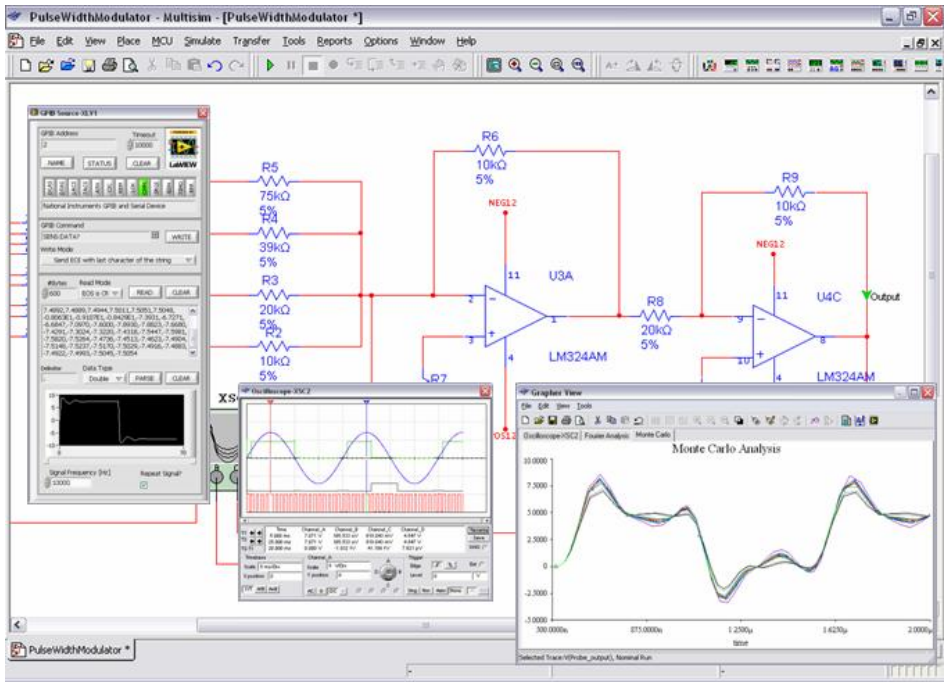
CMOS technology behaviour

The Hamming Weight Model (and its counterpart Hamming Distance Model) is widely used in the field due to its simplicity and the broad range of devices where it somehow fits.

Electrical engineer intuition

CMOS technology behaviour and Hardware design

e.g. simulations of electromagnetic emanations or power consumption from the design



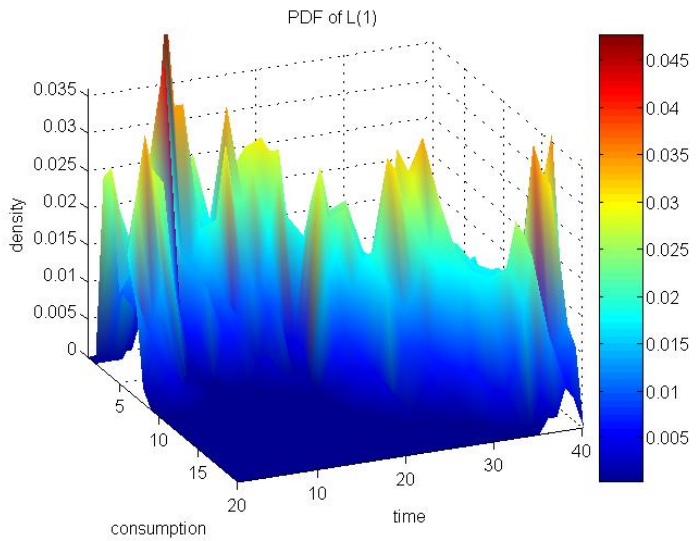
Model Inference

Through Side-Channel Observations

pdf inference of $L = \ell_i(V + \mathcal{B}_{S_i})$ for $V = 0$ from observations

For each time (abs.) and each side-channel value x in a finite interval (ord.) we plotted in z-axis:

$$Pr[L = x] \sim pdf_L(x)$$



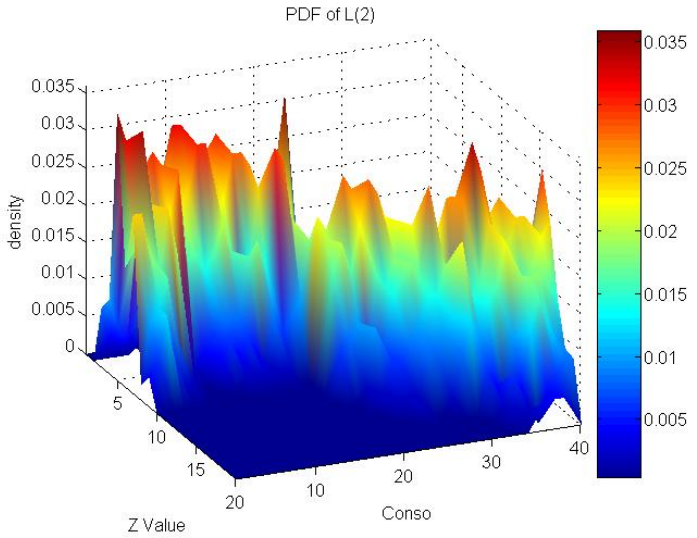
Model Inference

Through Side-Channel Observations

pdf inference of $L = \ell_i(V + \mathcal{B}_{S_i})$ for $V = 1$ from observations

For each time (abs.) and each side-channel value x in a finite interval (ord.) we plotted in z-axis:

$$Pr[L = x] \sim pdf_L(x)$$



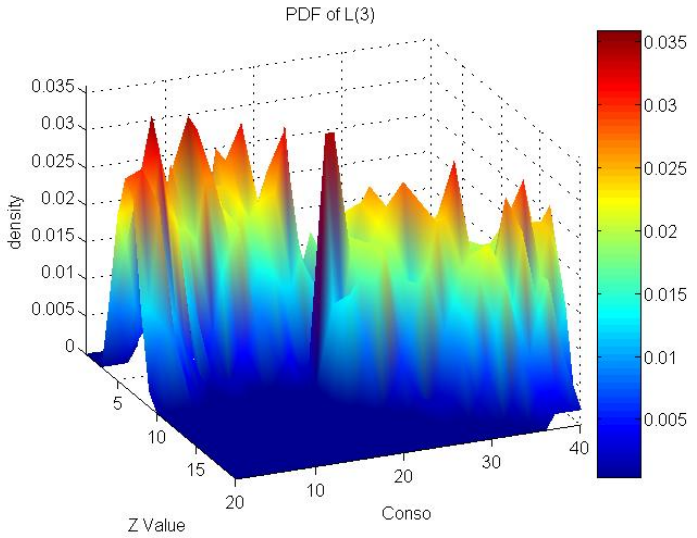
Model Inference

Through Side-Channel Observations

pdf inference of $L = \ell_i(V + \mathcal{B}_{S_i})$ for $V = 2$ from observations

For each time (abs.) and each side-channel value x in a finite interval (ord.) we plotted in z-axis:

$$Pr[L = x] \sim pdf_L(x)$$



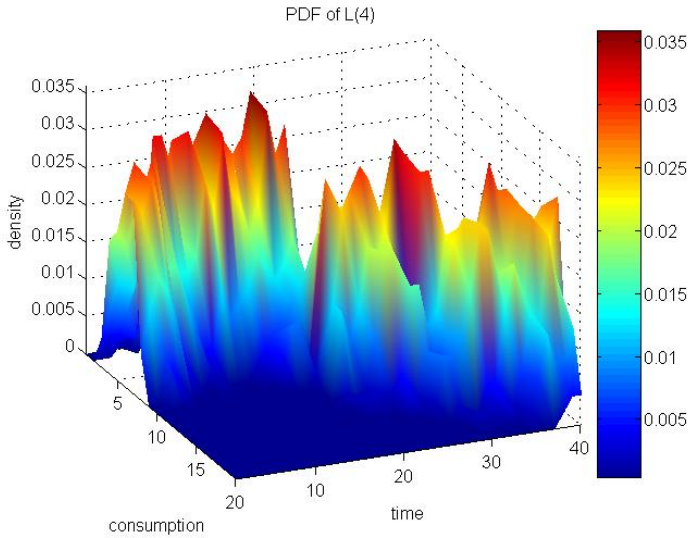
Model Inference

Through Side-Channel Observations

pdf inference of $L = \ell_i(V + \mathcal{B}_{S_i})$ for $V = \cdots$ from observations

For each time (abs.) and each side-channel value x in a finite interval (ord.) we plotted in z-axis:

$$Pr[L = x] \sim pdf_L(x)$$



Model Inference

Through Side-Channel Observations

pdf inference of $L = \ell_i(V + \mathcal{B}_{S_i})$ for $V = v$ from observations

Two main methods:

- Histograms: compute the pdf from integration over the partitioned space Time $\times$ Leakages.
  $\hookrightarrow$ prone to estimations errors
- Estimate the statistical moments of the distribution:

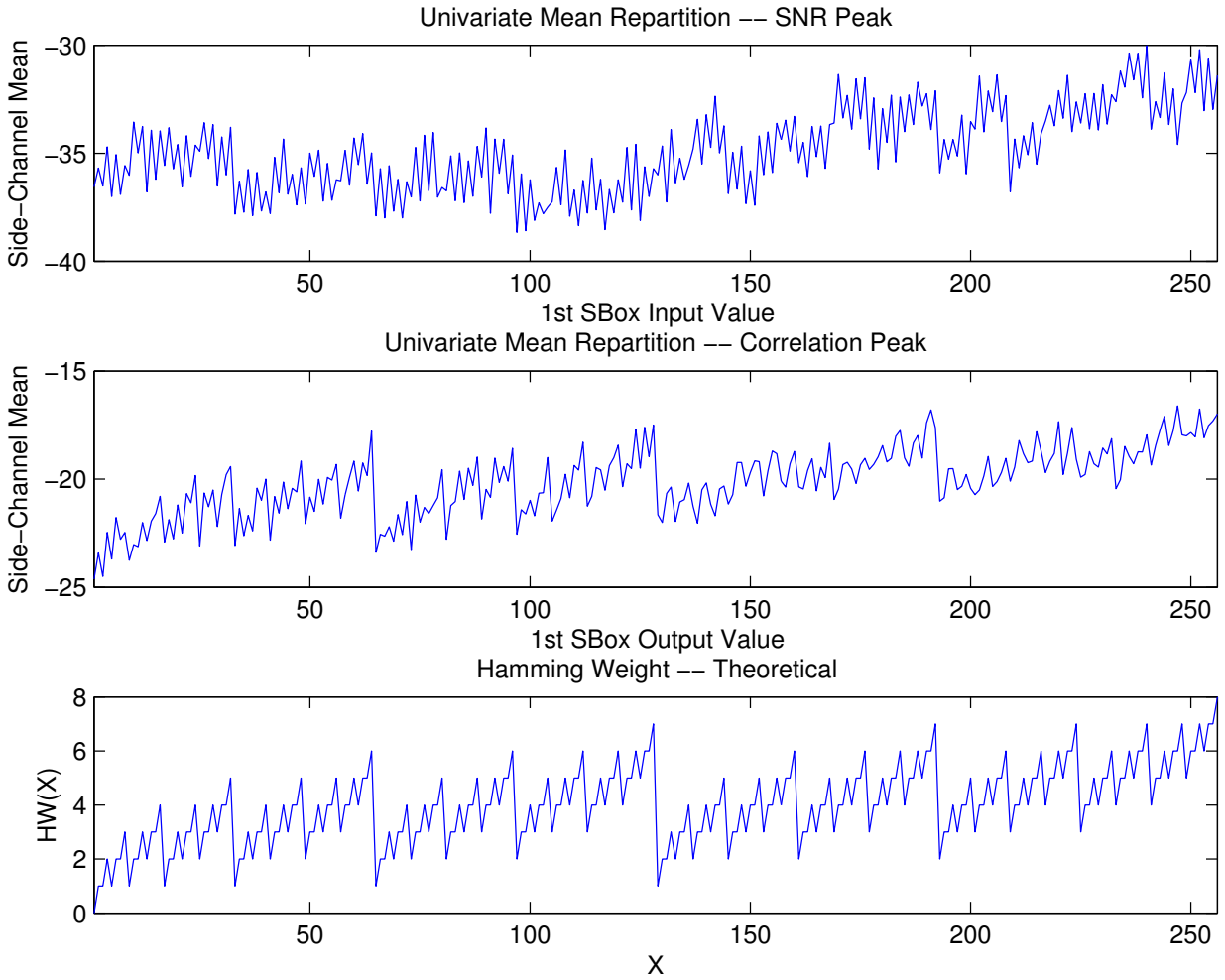
$$M_s = E[L^s]$$

$\hookrightarrow$ prone to estimations errors when s grows

In practice, one often assume that $\mathcal{B}_{S_i}$ follows a multivariate Gaussian distribution, in this case only the two first moments (mean vector and covariance matrix) have to be estimated.

Model Inference

Through Side-Channel Observations: Examples on real SCA traces



Model Inference

Through Side-Channel Observations

Inference of $L = \varphi(V) + B$ from Stochastic analysis SchindlerLemkePaar2005.

Let $(V_1, V_2, \cdots, V_t)$ be the binary decomposition of V.

$$\varphi(V) = \sum_i \alpha_i \cdot V_i + \sum_{i<j} \alpha_{i,j} \cdot V_i V_j + \cdots + \sum_{i_1<i_2<\cdots<i_t} \alpha_{i_1,i_2,\cdots,i_t} \cdot V_{i_1} V_{i_2} \cdots V_{i_t}$$

The inference problem can hence be reduced to a polynomial interpolation problem in a noisy context.

Model Inference

Linear Regression

Let $Y = \sum_{i \leq t} \alpha_i X_i + \varepsilon$ and let $\{y^j, x_1^j, x_2^j, \cdots, x_t^j\}_{j \leq N}$ be N realizations of the random variables.

$$\begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{pmatrix} = \begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_t^1 \\ x_1^2 & x_2^2 & \cdots & x_t^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_t^N \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_t \end{pmatrix} + \begin{pmatrix} \varepsilon^1 \\ \varepsilon^2 \\ \vdots \\ \varepsilon^N \end{pmatrix}$$

Under the assumption that ε is uncorrelated with the X_is, the least square method gives us an estimate of the coefficients α_i:

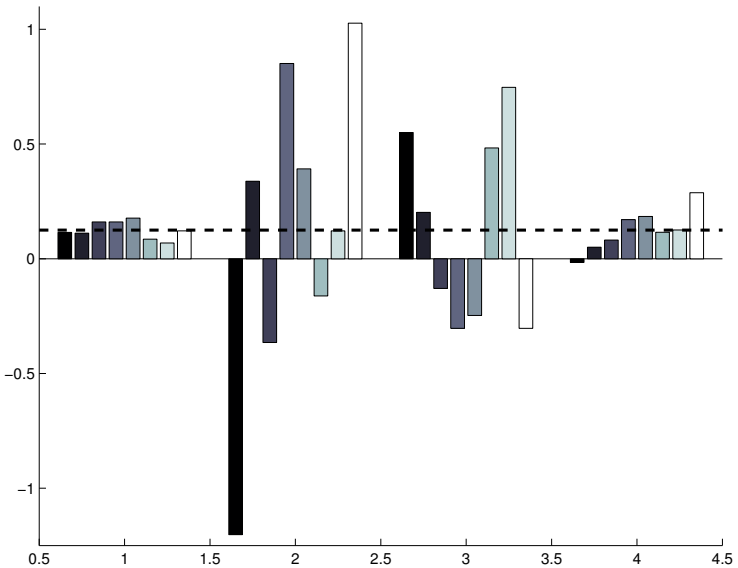
$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{x}^T \boldsymbol{x})^{-1} \boldsymbol{x}^T \boldsymbol{y} \ ,$$

which minimizes the objective function

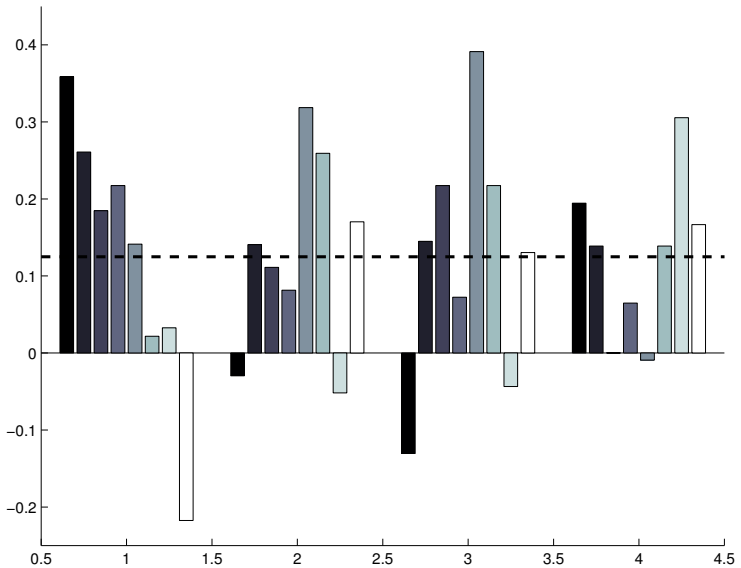
$$||\boldsymbol{y} - \boldsymbol{x}\hat{\boldsymbol{\alpha}}||_2$$

Model Inference

Through Side-Channel Observations: Examples on real SCA traces



(a) Device A



(b) Device B

Figure: LRA weights for two devices

Model Inference

Through Side-Channel Observations

Inference of $\ell_i(v + \mathcal{B}_{S_i})$ for any value v through machine learning approaches.

Machine Learning is the generic term to design "learning from data" for a computer. Recent publications propose to use algorithm from this field as an alternative to Stochastic or pdf inference. Among the best algorithms:

- k-means
- Support Vector Machines
- Random Forests

HospodarGierlichsMulderVerbauwhede2011, HeuserZohner2012, LermanMedeirosBontempiMarkowitch2013.

Quick Recap

Leakage Functions and Leakage Model

Many theoretical models in the litterature, in practice we often use the simplest form:

$$L \hookleftarrow \varphi(V) + \mathcal{N}(\mu, \sigma)$$

The leakage function is infered by

- knowledge on the technology: HW.
- knowledge on the technology and Hardware design.
- when possible, ineference from side-channel observations.

What do we have so far?

Let's build a Side-Channel Attack!

Hypothesis

- Let $V = f(P, K)$ be a sensitive variable.
- Let $\mathbf{m}$ be an accurate model of the leakage function ℓ.
- Let $\{\ell_{p_i}\}_i$ be a set of side-channel observations with respective known inputs $\{p_i\}_i$ and unknown secret k.

What do we have so far?

Let's build a Side-Channel Attack!

Hypothesis

- Let $V = f(P, K)$ be a sensitive variable.
- Let $\mathbf{m}$ be an accurate model of the leakage function ℓ.
- Let $\{\ell_{p_i}\}_i$ be a set of side-channel observations with respective known inputs $\{p_i\}_i$ and unknown secret k.

Predictions and statistical distance evaluation

- For each key candidate $\hat{k}$, evaluate $\{\text{pred}_i\}_i = \{\mathbf{m}(v_i = f(p_i, \hat{k}))\}_i$.
- For each key candidate $\hat{k}$, estimate a statistical distance between $\{\ell_{p_i}\}_i$ and $\{\text{pred}_i\}_i$.
- Output the closest key candidate $\hat{k}$.

What do we have so far?

Let's build a Side-Channel Attack!

Hypothesis

- Let $V = f(P, K)$ be a sensitive variable.
- Let $\mathbf{m}$ be an accurate model of the leakage function ℓ.
- Let $\{\ell_{p_i}\}_i$ be a set of side-channel observations with respective known inputs $\{p_i\}_i$ and unknown secret k.

Predictions and statistical distance evaluation

- For each key candidate $\hat{k}$, evaluate $\{\text{pred}_i\}_i = \{\mathbf{m}(v_i = f(p_i, \hat{k}))\}_i$.
- For each key candidate $\hat{k}$, estimate a statistical distance between $\{\ell_{p_i}\}_i$ and $\{\text{pred}_i\}_i$.
- Output the closest key candidate $\hat{k}$.

Part IV

Distinguishers

Outline

Distinguishers

Introduction

Predictions and statistical distance evaluation

- For each key candidate $\hat{k}$, evaluate $\{\text{pred}_i\}_i = \{\mathbf{m}(v_i = f(p_i, \hat{k}))\}_i$.
- For each key candidate $\hat{k}$, estimate a statistical distance between $\{\ell_{p_i}\}_i$ and $\{\text{pred}_i\}_i$.
- Output the closest key candidate $\hat{k}$.

In Side-Channel, a distinguisher is a statistical tool to estimate the coherence between predictions and observations.

Distinguishers

Introduction

Predictions and statistical distance evaluation

- For each key candidate $\hat{k}$, evaluate $\{\text{pred}_i\}_i = \{\mathbf{m}(v_i = f(p_i, \hat{k}))\}_i$.
- For each key candidate $\hat{k}$, estimate a statistical distance between $\{\ell_{p_i}\}_i$ and $\{\text{pred}_i\}_i$.
- Output the closest key candidate $\hat{k}$.

In Side-Channel, a distinguisher is a statistical tool to estimate the coherence between predictions and observations.

- Robustness against the noise B.
- Robustness against the error in the model $\mathbf{m}(\cdot)$.

Outline

Maximum likelihood

Optimal... when the model is perfectly known

Hypothesis

We have, for any value $v = f(p, k)$ taken by the sensitive variable

$$\Pr[L_p = x \mid K = k]$$

For each value of p, the family of pdf is denoted $\{F(L_p \mid k)\}_{k \in \mathcal{K}}$. After n observations $\{\ell_{p_i}\}_{i \leq n}$ that we assume independent and identically distributed with constant unknown parameter k, then their joint density function (also called likelihood) is:

$$F(\ell_{p_1}, \ell_{p_2}, \cdots, \ell_{p_n} \mid k) = F(\ell_{p_1} \mid k) \times F(\ell_{p_2} \mid k) \times \cdots \times F(\ell_{p_n} \mid k)$$

Maximum likelihood

Optimal... when the model is perfectly known

The maximum likelihood estimator of k is then

$$k^\star = \text{argmax}_{k \in \mathcal{K}} F(\ell_{p_1}, \ell_{p_2}, \cdots, \ell_{p_n} \mid k)$$

Maximum likelihood

Optimal... when the model is perfectly known

The maximum likelihood estimator of k is then

$$k^\star = \text{argmax}_{k \in \mathcal{K}} F(\ell_{p_1}, \ell_{p_2}, \cdots, \ell_{p_n} \mid k)$$

In practice:

- the *log-likelihood* is used: $\sum_{i \leq n} \log(F(\ell_{p_i} \mid k))$.
- the leakage is often assumed to follow a *multivariate gaussian distribution*:

$$F(\ell_p \mid k) = \frac{1}{(2\pi)^{t/2}\sqrt{\det(\Sigma_v)}} \exp\left(-\frac{1}{2}(\ell_p - \mu_v) \cdot \Sigma_v^{-1} \cdot (\ell_p - \mu_v)^T\right),$$

where $v = f(p, k)$ and (μ_v, Σ_v) are respectively the mean vector and covariance matrix of dimension t.

Maximum likelihood

Optimal... when the model is perfectly known

Remarks

- This approach works even when intermediate variables are secret-only.
- $F(\ell \mid k)$ for all $k \in \mathcal{K}$ must be known.

Outline

Partly Known Leakage Model

Often, the attacker has some knowledge on the deterministic part φ of the leakage function

the noise distribution is then assumed to be independent from V

For instance

- The leakage model comes from the study of the technology/design.
- The leakage model comes from a linear regression leakage inference.

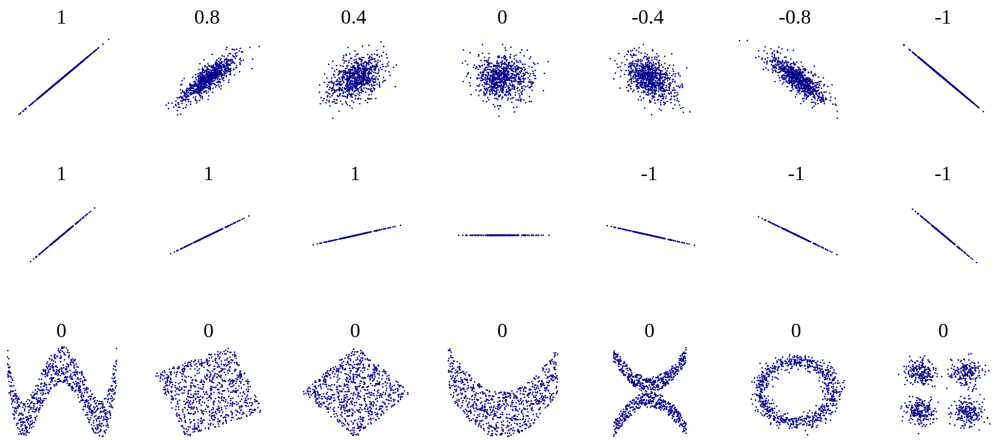
The deterministic part of $\ell()$ is partly known

Pearson Correlation

Linear correlation between X and Y: $\rho(X, Y) = \frac{\mathbf{cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}}$.

- It also called Pearson Coefficient.
- $\rho(X, Y) = \rho(Y, X)$ and $\rho(X, Y) \in [-1, 1]$.
- $\rho(X, Y) = 0$ implies that X and Y are dependent.

It detects (only!) linear dependencies between two variables.



The deterministic part of $\ell()$ is partly known

Pearson Correlation

In the case of an intuition-based leakage model (*e.g.* Hamming weight model), it is often assumed that this model is linearly related to the deterministic part of the leakage function, *i.e.* $\varphi(V) = \alpha \cdot HW(V)$.
We then have:

$$k^\star = \text{argmax}_{k \in \mathcal{K}} \hat{\rho}(\{\ell_{p_i}\}, \{\mu_{v_i}\}),$$

where $v_i = f(p_i, k)$ and μ_{v_i} is the value of the leakage model applied to v_i (*i.e.* they are the predictions $\{\text{pred}_i\}$).

Quick Recap

Distinguishers

Various statistical tools with their advantages and disadvantage.

In practice:

- Difference of Means and Correlation are still the most common distinguishers.
- Techniques based on model inference are rising as the attacker model becomes more and more realistic.

Part V

Side-Channel Attacks

Outline

Side-Channel Attacks

Introduction

Simple Power Analysis

- for secret-only sensitive variables.
- simple and (partly) known leakage functions.
- simple test for distinguishing keys.

Advanced Power Analysis

- for secret+public sensitive variables.
- divide and conquer strategy: $v = f(p, k)$ with $log_2(k) \leq 32$.
- a statistical distinguisher adapted to the attacker model.

Outline

Simple Power Analysis

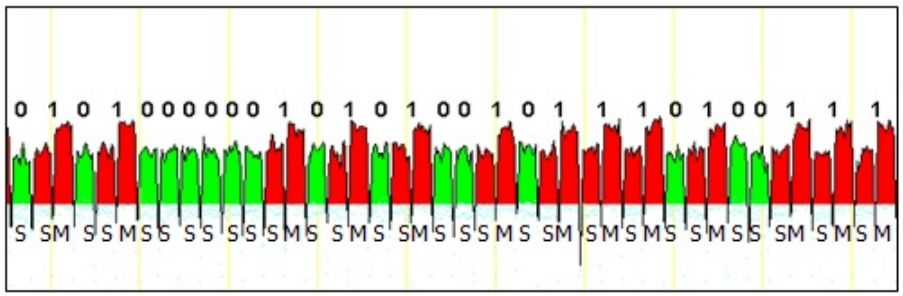
Kocher *et al.* 1996

Classical Example: Unprotected Square-and-Multiply Algorithm

Algo S&M binary from left to right

INPUT(S) :
$m, d = (d_n d_{n-1} \dots d_1 d_0)_2$
OUTPUT(S) : m^d

1: $t \leftarrow 1$
2: **for** $i = n$ to 0 **do**
3: $\quad t \leftarrow t \times t$
4: $\quad$ **if** $d_i = 1$ **then**
5: $\quad\quad t \leftarrow t \times m$
6: **return** t



Remark/Assumption: squaring and multiplication have different leakage patterns.

Outline

Advanced Power Analysis

Kocher *et al.* 1996

General Framework (Theoretical)

Context: attack during the manipulation of $V = (P + K)$.

1. Measurement : get a sample $(\ell_{p_i})_i$ related to a sample $(p_i)_i$ of plaintexts.

Advanced Power Analysis

Kocher *et al.* 1996

General Framework (Theoretical)

Context: attack during the manipulation of $V = (P + K)$.

1. Measurement : get a sample $(\ell_{p_i})_i$ related to a sample $(p_i)_i$ of plaintexts.
2. Model Selection : Design/Select a function $\mathbf{m}(\cdot)$.

Advanced Power Analysis

Kocher *et al.* 1996

General Framework (Theoretical)

Context: attack during the manipulation of $V = (P + K)$.

1. Measurement : get a sample $(\ell_{p_i})_i$ related to a sample $(p_i)_i$ of plaintexts.
2. Model Selection : Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction : For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(f(p_i + \hat{k}))$.

Advanced Power Analysis

Kocher *et al.* 1996

General Framework (Theoretical)

Context: attack during the manipulation of $V = (P + K)$.

1. Measurement : get a sample $(\ell_{p_i})_i$ related to a sample $(p_i)_i$ of plaintexts.
2. Model Selection : Design/Select a function $\mathbf{m}(\cdot)$.
3. Prediction : For every $\hat{k}$, compute $m_{\hat{k},i} = \mathbf{m}(f(p_i + \hat{k}))$.
4. Distinguisher Selection : Choose a statistical distinguisher Δ.

Advanced Power Analysis

Kocher *et al.* 1996

General Framework (Theoretical)

Context: attack during the manipulation of $V = (P + K)$.

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

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

The random variable associated to $m_{\hat{k},i}$ is denoted by $M_{\hat{k}}$.

Advanced Power Analysis

Kocher *et al.* 1996

General Framework (Theoretical)

Context: attack during the manipulation of $V = (P + K)$.

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

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

   The random variable associated to $m_{\hat{k},i}$ is denoted by $M_{\hat{k}}$.

6. Key Candidate Selection : Output $\hat{k}$ maximizing $\Delta_{\hat{k}}$.

Advanced Power Analysis

Side Channel Analysis: attack Description Sheet

Attack Description Sheet

Type of Leakage: *e.g. power consumption or electromagnetic emanation*
Model Function: *e.g. one bit of V 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 Power Analysis

Litterature

Many proposals in the literature:

- DPA KocherJaffeJun 1996-98
- Multi-bit DPA Messerges 2000;
- Template ChariRaoRohatgi 2002;
- CPA BrierClavierOlivier 2004;
- Stochastic Attacks SchindlerLemkePaar2005;
- or the MIA GierlichsBatinaTuylsPreneel 2008;
- ...

Advanced Power Analysis

DPA attack

Kocher *et al.* 1996.

Attack Description Sheet: Differential Power Analysis

Type of Leakage: no restriction.
Model Function: binary model (*e.g. 1 bit from an sbox output).*
Statistical Distinguisher: Absolute Difference of Means.
Key Candidate Selection: the candidate that maximizes the DoM.

- 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 Power Analysis

MPA attack — Messerges 2000.

Attack Description Sheet: **Multi-bit Differential Power Analysis**

Type of Leakage: no restriction.
Model Function: the Hamming weight function.
Statistical Distinguisher: $\Delta_{\hat{k}}$: a statistical estimator of

$$\Delta_{\hat{k}} = \mathbf{E}[L \mid \mathbf{m}(\hat{v}) \leq \tau] - \mathbf{E}[L \mid \mathbf{m}(\hat{v}) > \tau]$$

Key Candidate Selection: the candidate that maximizes the Δ.

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

Advanced Power Analysis

CPA attack

BrierClavierOlivier 2004.

Attack Description Sheet: **Correlation Power Analysis**

Type of Leakage: no restriction.
Model Function: the Hamming weight function.
Statistical Distinguisher: Absolute Pearson Correlation. Key Candidate Selection: the candidate that maximizes the Pearson coefficient.

- Pros: Exploit more information than the DPA or MPA. Low algorithm complexity.
- Cons: Need assumption (*e.g. Hamming weight) on the device behaviour.*

Advanced Power Analysis

Template Attacks

ChariRaoRohatgi 2002

Attack Description Sheet: **Template**

Type of Leakage: no restriction.
Model Function: model *inference* from a copy of the target device.
Statistical Distinguisher: Maximum Likelihood.
Key Candidate Selection: the candidate that maximizes the probability of being correct.

- Pros: Optimal attack when the profiling phase (inference) is accurate enough.
- Cons: Need a practice device.

Advanced Power Analysis

Linear Regression Based (a.k.a. stochastic) Attacks SchindlerLemkePaar2005

Attack Description Sheet: **Stochastic attacks**

Type of Leakage: no restriction.
Model Function: model *inference* from a copy of the target device using linear regression.
Statistical Distinguisher: Euclidean Distance (a.k.a. sum of squares difference) or Linear Correlation.
Key Candidate Selection: the candidate that minimizes the Euclidean Distance or maximizes the Pearson Coefficient.

- Pros: Efficient model inference.
- Cons: Need a practice device. Does not infer the noise distribution.

Advanced Power Analysis

Linear Regression Attacks

DogetProuffRivainStandaert 2011

Attack Description Sheet: **Linear Regression attacks**

Type of Leakage: no restriction.
Model Function: a set of basis functions $\mathbf{m}^{(i)}(\cdot,\cdot)$ s.t. $\varphi()$ 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.

- Pros: Efficient model inference.
- Cons: Does not infer the noise distribution.

Advanced Power Analysis

MIA attack GierlichsBatinaTuylsPreneel 2008

Attack Description Sheet: **Mutual Information Analysis**

Type of Leakage: no restriction.
Model Function: None is needed.
Statistical Distinguisher: Mutual Information.
Key Candidate Selection: the candidate that maximizes the Mutual Information.

- Pros: theoretically able to detect any kind of dependency whatever the quality of the model
- Cons: need for efficient estimators of the entropy.

Advanced Power Analysis

Other attacks

- Collisions-Based attacks: SchrammWollingerPaar2003, MoradiMischkeEisenbarth2010, Moradi2012.
  - Good alternative when no leakage information is provided.
- Kolmogorov-Smirnov Based attacks: WhitnallOswaldMather2011.
  - 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.

Part VI

Evaluation