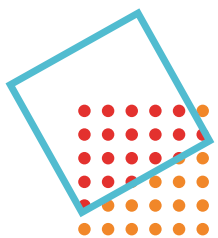
Attacking Randomized Exponentiations Using Unsupervised Learning

Guilherme Perin, Laurent Imbert, Lionel Torres, Philippe Maurine

Constructive Side-Channel Analysis and Secure Design
COSADE, April 14th and 15th, 2014 - Paris



Laboratoire
Informatique
Robotique
Microélectronique
Montpellier

UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

Side-Channel Attacks on Exponentiations

- Multi-trace attacks: DPA [KJJ1999], CPA [BCO2004], Collisions [FV2003], templates [CRR2002], etc.
- Single-execution of an exponentiation (Horizontal Attacks);

Horizontal Attacks

- Single-execution of an exponentiation (single trace);
- Threat known or masked inputs;
- Get advantages of long-integer multiplications;

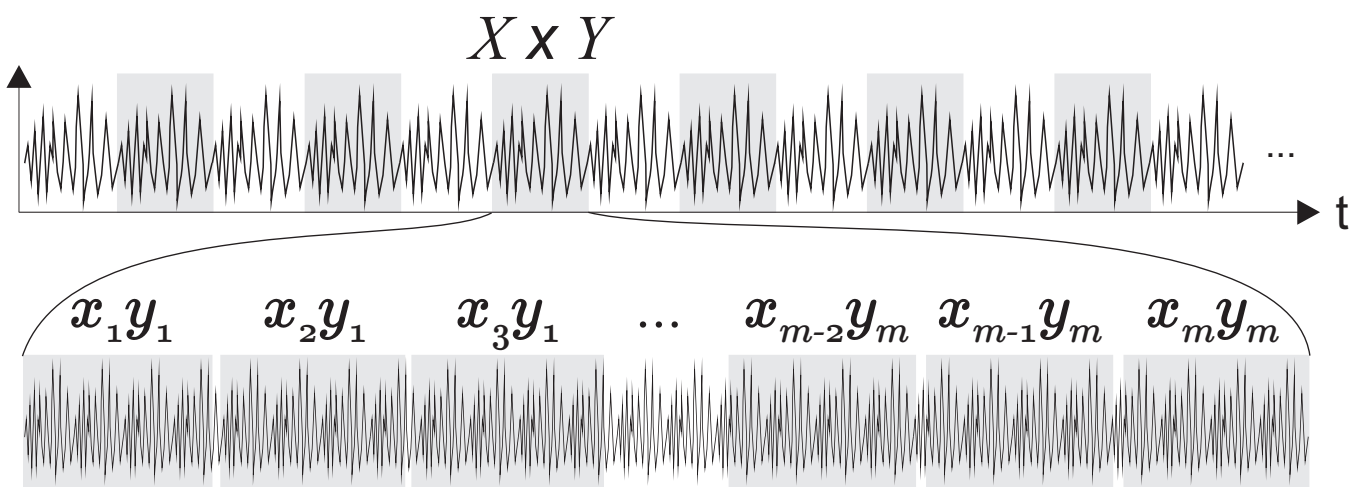
- CHES 2001: Colin Walter presented the *Big Mac Attack*;
- ICICS 2010: Clavier *et al* presented the *Horizontal correlation attack*;
- INDOCRYPT 2012: Clavier *et al* presented the *Triangular Trace Analysis*;
- COSADE 2012: Sven Bauer presented template-based attacks on RSA exponent blinding;
- INDOCRYPT 2013: Aurelie Bauer *et al* presented correlation-based attacks on the exponent blinding (single traces);
- SAC 2013: Aurelie Bauer *et al* presented horizontal collision-correlation attacks on Elliptic Curves;

Horizontal Correlation Attacks

Data: $X = (x_1, ..., x_m)$ and $Y = (y_1, ..., y_m)$.

Result: $W = X \times Y$

for $i = 1$ **to** m **do**
    $c = 0$
    for $j = 1$ **to** m **do**
        $(uv) = (w_{i+j} + x_i y_j) + c$
        $w_{i+j} = v$
        $c = u$
    end
    $w_{i+n+1} = c$
end



Related Work

Johann Heyszl, Andreas Ibing, Stefan Mangard, Fabrizio De Santis, and Georg Sigl. "**Clustering algorithms for non-profiled single-execution attacks on exponentiations**" IACR Cryptology ePrint Archive, 2013.

- Device Under Test: ECC Co-processor;
- Multi-probe EM acquisition;
- The authors simulated the acquisition of 9 measurements on 9 different probe positions;
- The leakage of each measurement (single-execution of exponentiations) is combined to retrieve the secret.

Our contributions:
- Non-profiled attack;
- Single-probe, single-execution of an exponentiation;
- Horizontal attack;
- Statistical classifiers (majority decision, PDF, bayesian classifier) of clustered data;

Outline

1. Device under test
2. Unsupervised learning and clustering algorithms
3. The unsupervised attack
4. Improvements
5. Countermeasures

Device Under Test

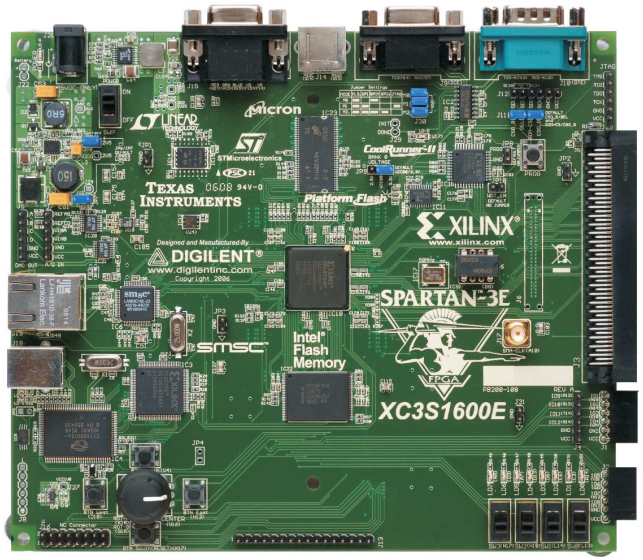
RSA-512

Countermeasures:

- Montgomery ladder using Residue Number System (RNS)
- Exponent blinding ($d_r = d + r\phi(N)$)
- Leak Resistant Arithmetic (RNS)

Exploited leakages:

- Single traces
- SPA-leakages:
  - Memory addresses
  - Control decisions



Device Under Test

RNS Montgomery Ladder

Data: x in $\mathcal{A} \cup \mathcal{B}$, where $\mathcal{A} = (a_1, a_2, \ldots, a_k)$, $\mathcal{B} = (b_1, b_2, \ldots, b_k)$, $A = \prod_{i=1}^{k} a_i$, $B = \prod_{i=1}^{k} b_i$, $\gcd(A, B) = 1$, $\gcd(B, N) = 1$ and $d = (d_\ell \ldots d_2 d_1)_2$.

Result: $z = x^d \bmod N$ in $\mathcal{A} \cup \mathcal{B}$

Pre-Computations: $|AB \bmod N|_{\mathcal{A} \cup \mathcal{B}}$

randomize($\mathcal{A}$, $\mathcal{B}$)
$d_r = d + r.\phi(N)$
$A_0 = MM(1, AB \bmod N, N, \mathcal{A}, \mathcal{B})$ (in $\mathcal{A} \cup \mathcal{B}$)
$A_1 = MM(x, AB \bmod N, N, \mathcal{A}, \mathcal{B})$ (in $\mathcal{A} \cup \mathcal{B}$)
for $i = \ell$ **to** 1 **do**
  if $d_i = 1$ **then**
    $A_0 = MM(A_0, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
    $A_1 = MM(A_1, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
  else
    $A_1 = MM(A_0, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
    $A_0 = MM(A_0, A_0, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
  end
end
$z = MM(A_0, 1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)

Known-input attacks can not be mounted (collisions, DPA, CPA).

Device Under Test

RNS Montgomery Ladder

Data: x in $\mathcal{A} \cup \mathcal{B}$, where $\mathcal{A} = (a_1, a_2, \ldots, a_k)$, $\mathcal{B} = (b_1, b_2, \ldots, b_k)$, $A = \prod_{i=1}^{k} a_i$, $B = \prod_{i=1}^{k} b_i$, $\gcd(A, B) = 1$, $\gcd(B, N) = 1$ and $d = (d_\ell \ldots d_2 d_1)_2$.

Result: $z = x^d \bmod N$ in $\mathcal{A} \cup \mathcal{B}$

Pre-Computations: $|AB \bmod N|_{\mathcal{A} \cup \mathcal{B}}$

$randomize(\mathcal{A}, \mathcal{B})$
$d_r = d + r.\phi(N)$
$A_0 = MM(1, AB \bmod N, N, \mathcal{A}, \mathcal{B})$ (in $\mathcal{A} \cup \mathcal{B}$)
$A_1 = MM(x, AB \bmod N, N, \mathcal{A}, \mathcal{B})$ (in $\mathcal{A} \cup \mathcal{B}$)
for $i = \ell$ **to** 1 **do**
  if $d_i = 1$ **then**
    $A_0 = MM(A_0, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
    $A_1 = MM(A_1, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
  else
    $A_1 = MM(A_0, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
    $A_0 = MM(A_0, A_0, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
  end
end
$z = MM(A_0, 1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)

Leakage must be exploited from single-traces.

Device Under Test

RNS Montgomery Ladder

Data: x in $\mathcal{A} \cup \mathcal{B}$, where $\mathcal{A} = (a_1, a_2, \ldots, a_k)$, $\mathcal{B} = (b_1, b_2, \ldots, b_k)$, $A = \prod_{i=1}^{k} a_i$, $B = \prod_{i=1}^{k} b_i$, $\gcd(A, B) = 1$, $\gcd(B, N) = 1$ and $d = (d_\ell \ldots d_2 d_1)_2$.

Result: $z = x^d \bmod N$ in $\mathcal{A} \cup \mathcal{B}$

Pre-Computations: $|AB \bmod N|_{\mathcal{A}\cup\mathcal{B}}$

$randomize(\mathcal{A}, \mathcal{B})$
$d_r = d + r.\phi(N)$
$A_0 = MM(1, AB \bmod N, N, \mathcal{A}, \mathcal{B})$ (in $\mathcal{A} \cup \mathcal{B}$)
$A_1 = MM(x, AB \bmod N, N, \mathcal{A}, \mathcal{B})$ (in $\mathcal{A} \cup \mathcal{B}$)
for $i = \ell$ **to** 1 **do**
  if $d_i = 1$ **then**
    $A_0 = MM(A_0, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
    $A_1 = MM(A_1, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
  else
    $A_1 = MM(A_0, A_1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
    $A_0 = MM(A_0, A_0, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)
  end
end
$z = MM(A_0, 1, N, \mathcal{B}, \mathcal{A})$ (in $\mathcal{A} \cup \mathcal{B}$)

Remaining leakages: memory addresses, control decisions (address-bit attacks).

Outline

1. Device under test
2. Unsupervised learning and clustering algorithms
3. The unsupervised attack
4. Improvements
5. Countermeasures

Basics about clustering

- Clustering is a:
  - data mining technique;
  - unsupervised learning process for partitioning a data set into c sub-groups;
- Instances of sub-group c_1 are similar to each other and dissimilar to the instances of other sub-groups c_i.
- Clustering algorithms are divided in:
  - partitional clustering techniques (k-means, fuzzy c-means, k-medoids, etc.);
  - distribution-based clustering (expectation-maximization algorithm);
  - hierarchical clustering techniques;
  - density-based clustering techniques;
  - grid-based clustering techniques.

Basics about clustering

- Clustering is a:
  - data mining technique;
  - unsupervised learning process for partitioning a data set into c sub-groups;
- Instances of sub-group c_1 are similar to each other and dissimilar to the instances of other sub-groups c_i.
- Clustering algorithms are divided in:
  - partitional clustering techniques (k-means, fuzzy c-means, k-medoids, etc.);
  - distribution-based clustering (expectation-maximization algorithm);
  - hierarchical clustering techniques;
  - density-based clustering techniques;
  - grid-based clustering techniques.

K-Means

$\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$
$\mu_i = x_{r_i}$ (initialize the center with a random sample)
c classes

---

begin initialize $\mathbf{x}, n, c, \mu_1, \ldots, \mu_c$
  do classify n samples x_j according to nearest μ_i by computing $ED_{i,j}$
    recompute μ_i
  until no change in μ_i
 return $\mu_1, \ldots, \mu_c$
end

Euclidean Distance: $ED_{ij} = \| x_j - \mu_i \|^2$

Fuzzy k-Means

$\sum_{i=1}^{c} P(\omega_i|x_j) = 1$

---

begin initialize $n, c, \mu_1, \ldots, \mu_c, P(\omega_i|x_j)$
  normalize probabilities of cluster memberships
  do classify n samples according to nearest μ_i
    recompute μ_i by Eq. 1
    recompute $P(\omega_i|x_j)$ by Eq. 2
  until no change in μ_i and $P(\omega_i|x_j)$
 return $\mu_1, \ldots, \mu_c$
end

$$\mu_j = \frac{\sum_{j=1}^{n}[P(\omega_i|x_j)]^b x_j}{\sum_{j=1}^{n}[P(\omega_i|x_j)]^b} \tag{1}$$

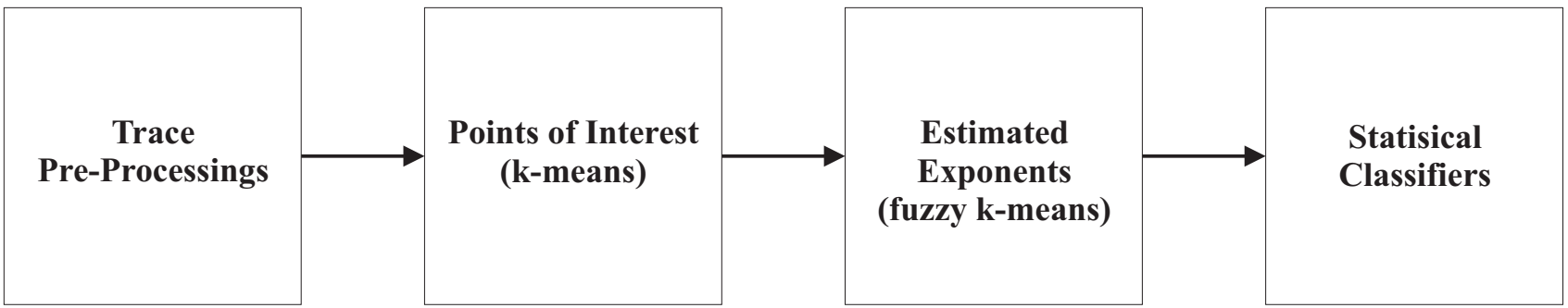
$$P(\omega_i|x_j) = \frac{(1/ED_{ij})^{1/(b-1)}}{\sum_{r=1}^{c}(1/ED_{rj})^{1/(b-1)}}, \quad ED_{ij} = \| x_j - \mu_i \|^2 \tag{2}$$

Outline

1. Device under test
2. Unsupervised learning and clustering algorithms
3. The unsupervised attack
4. Improvements
5. Countermeasures

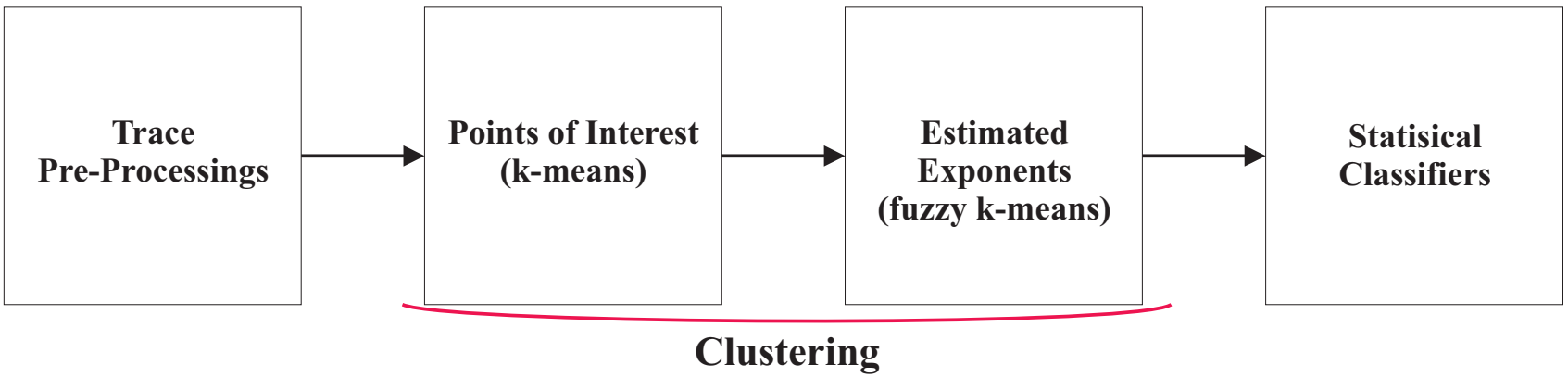
The Unsupervised Attack

- It is based on a single-execution of an exponentiation;
- It requires minimal knowledges about implementation aspects (modular multiplication algorithm);
- The proposed attack is divided into four phases:



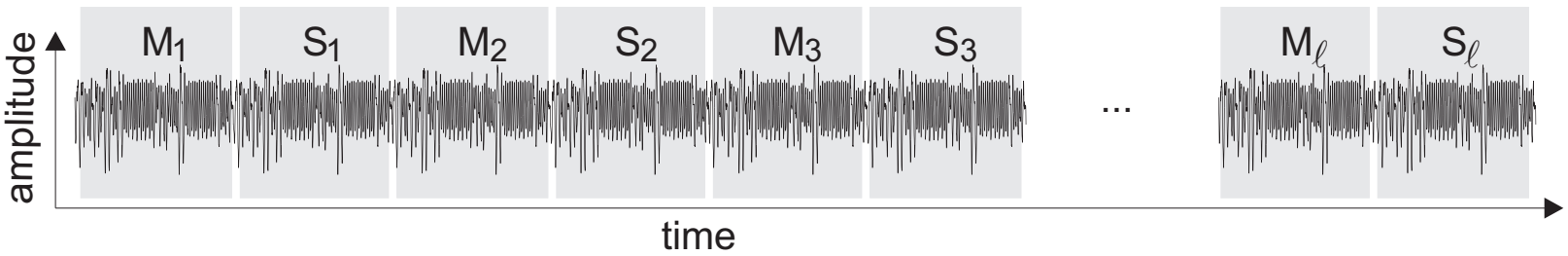
The Unsupervised Attack

- It is based on a single-execution of an exponentiation;
- It requires minimal knowledges about implementation aspects (modular multiplication algorithm);
- The proposed attack is divided into four phases:



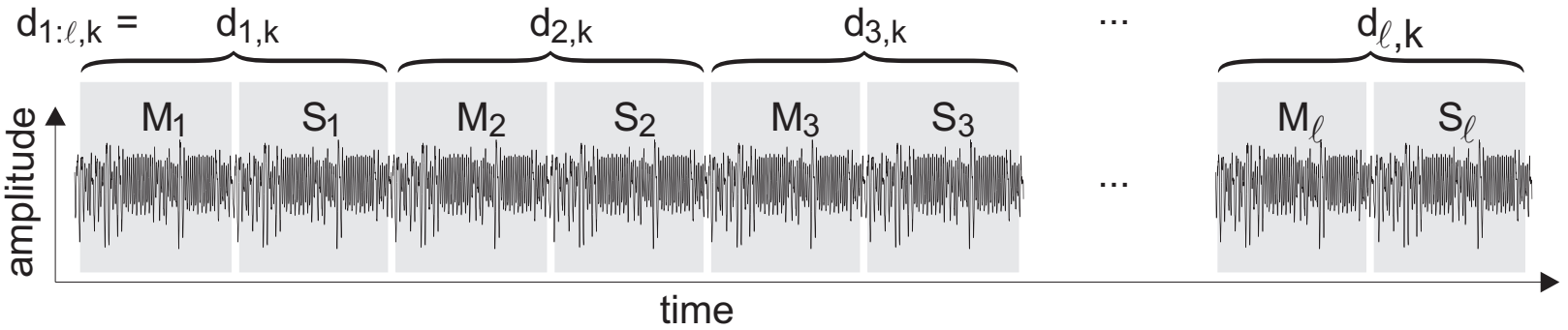
Trace Pre-Processings

Montgomery ladder exponentiation trace:



Trace Pre-Processings

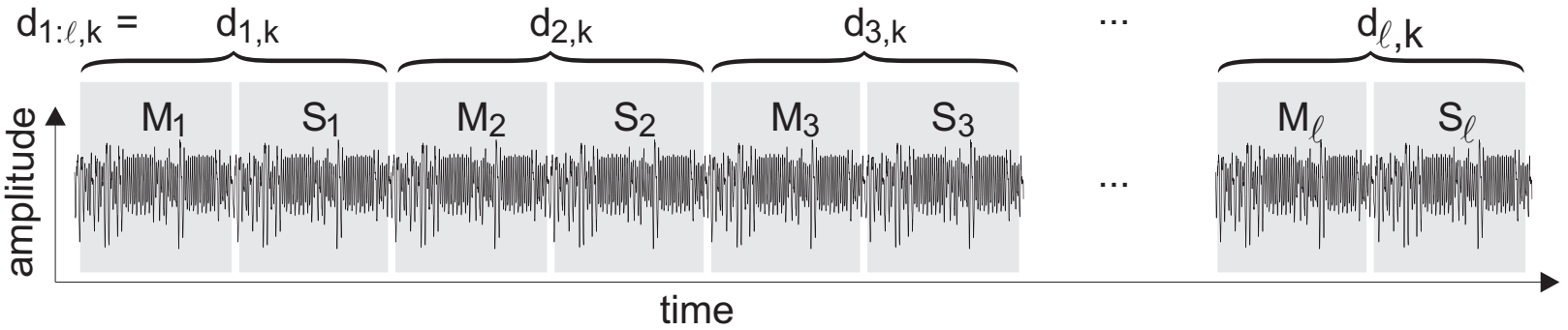
Montgomery ladder exponentiation trace:



$d_{i,k}$ = i-th exponent bit of k-th exponent

Trace Pre-Processings

Montgomery ladder exponentiation trace:



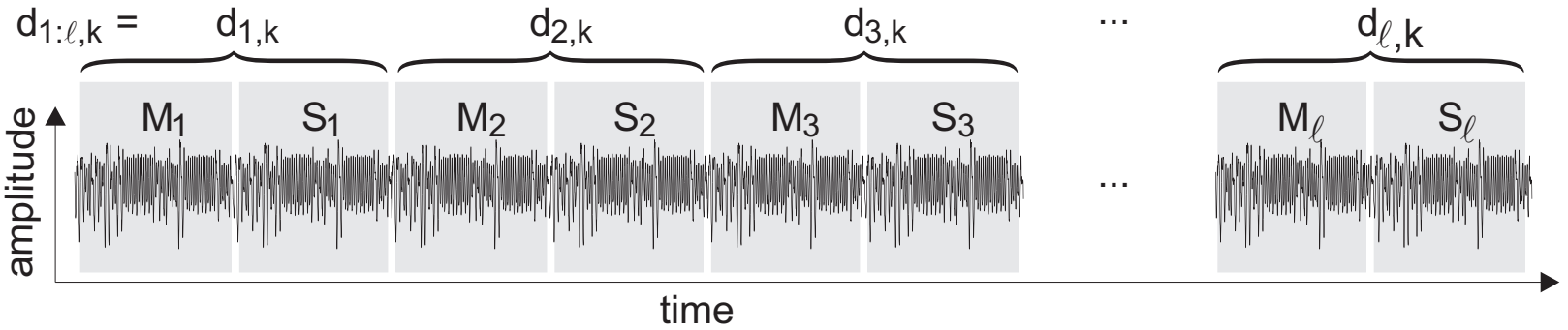
$d_{i,k}$ = i-th exponent bit of k-th exponent

$\mathrm{M}_i\mathrm{S}_i$ = Multiply-Squaring at $d_{i,k}$, 59200 samples (20GS/s, 50MHz)

$\langle \mathrm{MS} \rangle_i$ = 592 concatenated samples (average 100 into 1).

Trace Pre-Processings

Montgomery ladder exponentiation trace:



$d_{i,k}$ = i-th exponent bit of k-th exponent

$\mathrm{M}_i\mathrm{S}_i$ = Multiply-Squaring at $d_{i,k}$, 59200 samples (20GS/s, 50MHz)

$\langle \mathrm{MS} \rangle_i$ = 592 concatenated samples (average 100 into 1).

- Increase SNR;
- Less misalignment between $\mathrm{M}_i\mathrm{S}_i$ frames;
- Less computational burden.

Points of Interest

Now, the ℓ operations $\langle \mathrm{MS} \rangle_i$ are represented by a matrix T and we apply the *k*-means clustering on all columns of matrix T, every $\{t_{i:\ell,j}\}$, $1 \leq j \leq 592$ (in our experiments, $\ell = 512$):

$$T = \begin{bmatrix} \langle \mathrm{MS} \rangle_1 \\ \langle \mathrm{MS} \rangle_2 \\ \vdots \\ \langle \mathrm{MS} \rangle_\ell \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,592} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,592} \\ \vdots & \vdots & \ddots & \vdots \\ t_{\ell,1} & t_{\ell,2} & \cdots & t_{\ell,592} \end{bmatrix}$$

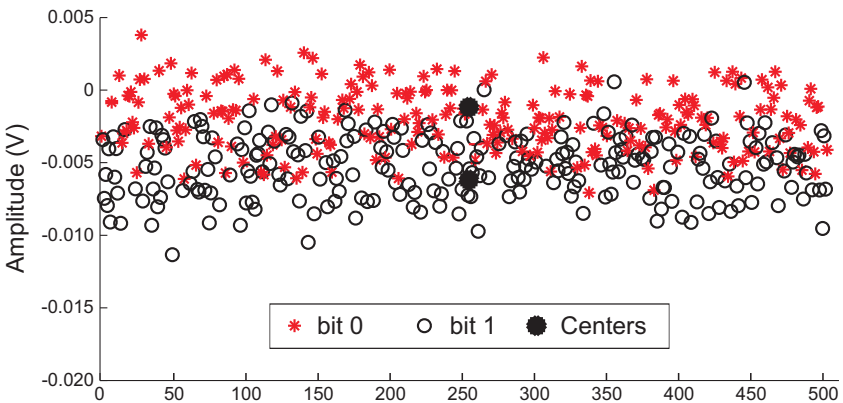
$$\downarrow \qquad \downarrow \qquad \qquad \downarrow$$

$$\{t_{i:\ell,1}\} \; \{t_{i:\ell,2}\} \qquad \{t_{i:\ell,592}\}$$

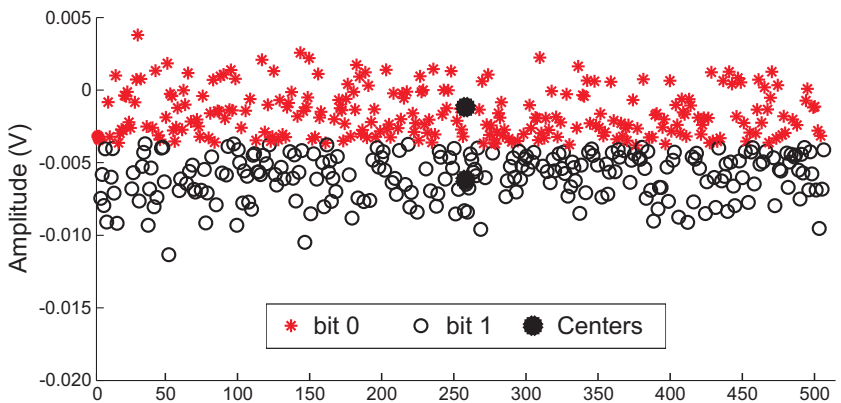
Points of Interest

Now, the ℓ operations $\langle \mathrm{MS} \rangle_i$ are represented by a matrix T and we apply the *k*-means clustering on all columns of matrix T, every $\{t_{i:\ell,j}\}$, $1 \leq j \leq 592$ (in our experiments, $\ell = 512$):

$$T = \begin{bmatrix} \langle \mathrm{MS} \rangle_1 \\ \langle \mathrm{MS} \rangle_2 \\ \vdots \\ \langle \mathrm{MS} \rangle_\ell \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,592} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,592} \\ \vdots & \vdots & \ddots & \vdots \\ t_{\ell,1} & t_{\ell,2} & \cdots & t_{\ell,592} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \downarrow \\ \{t_{i:\ell,1}\} \end{matrix} \begin{matrix} \\ \\ \\ \\ \downarrow \\ \{t_{i:\ell,2}\} \end{matrix} \begin{matrix} \\ \\ \\ \\ \downarrow \\ \{t_{i:\ell,592}\} \end{matrix}$$



Correct Classification



Cluster classification

Points of Interest

Now, the ℓ operations $\langle \mathrm{MS} \rangle_i$ are represented by a matrix T and we apply the k-means clustering on all columns of matrix T, every $\{t_{i:\ell,j}\}$, $1 \leq j \leq 592$ (in our experiments, $\ell = 512$):

$$T = \begin{bmatrix} \langle \mathrm{MS} \rangle_1 \\ \langle \mathrm{MS} \rangle_2 \\ \vdots \\ \langle \mathrm{MS} \rangle_\ell \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,592} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,592} \\ \vdots & \vdots & \ddots & \vdots \\ t_{\ell,1} & t_{\ell,2} & \cdots & t_{\ell,592} \end{bmatrix}$$

$$\downarrow \qquad \downarrow \qquad \qquad \downarrow$$

$$\{t_{i:\ell,1}\} \quad \{t_{i:\ell,2}\} \qquad \{t_{i:\ell,592}\}$$

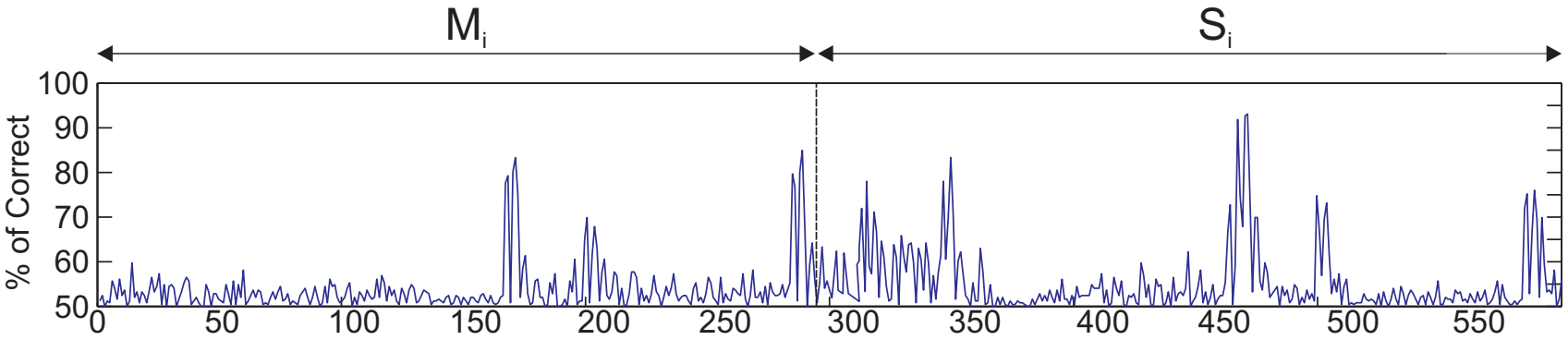
$$\downarrow \qquad \downarrow \qquad \qquad \downarrow$$

$$\widehat{d_{i:\ell,1}} \quad \widehat{d_{i:\ell,2}} \qquad \widehat{d_{i:\ell,592}} = \text{estimated exponents}$$

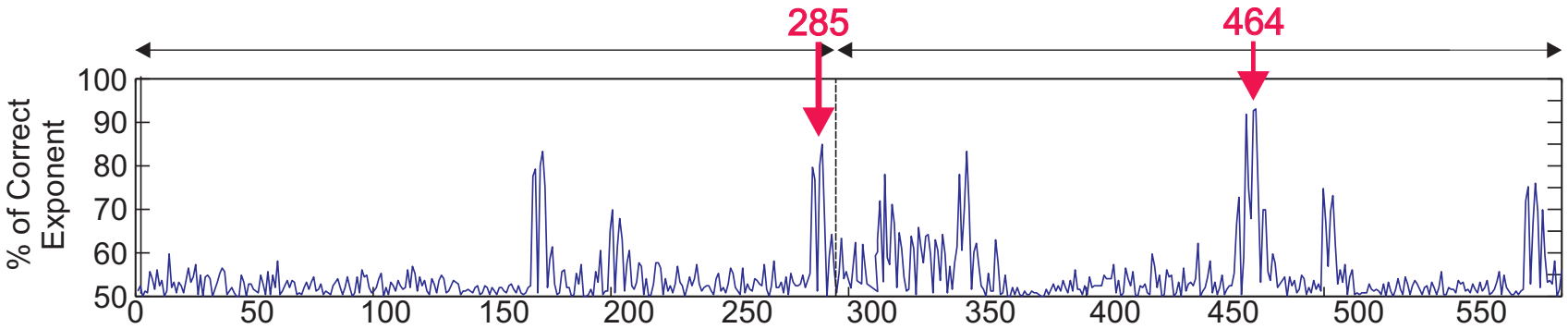
For each estimated exponent $\widehat{d_{i:\ell,j}}$, an estimated (approximated) difference of means $\widehat{D_j}$ is computed:

$$\widehat{D_j} = \frac{1}{N_0} \sum_{i=1}^{N_0} \langle \mathrm{MS} \rangle_{\widehat{d_{i,j}}=0} - \frac{1}{N_1} \sum_{i=1}^{N_1} \langle \mathrm{MS} \rangle_{\widehat{d_{i,j}}=1} \tag{3}$$

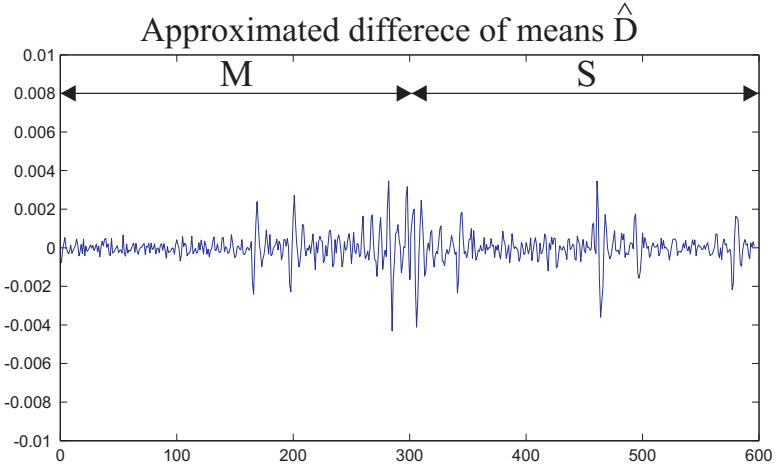
Points of Interest



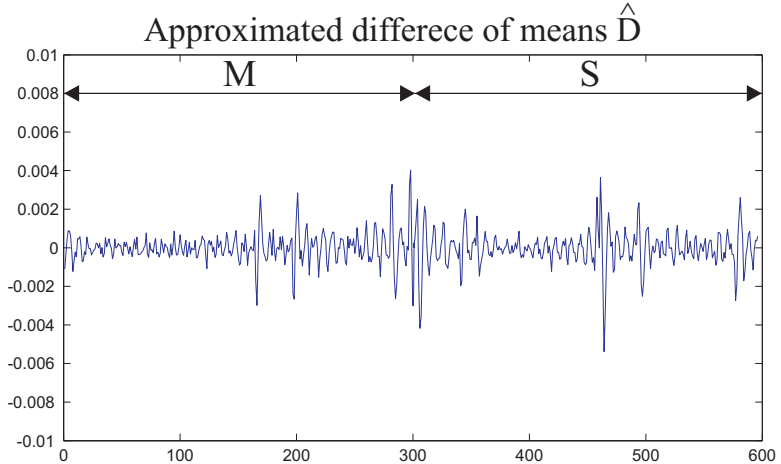
Points of Interest



For the most likely estimated exponents, the estimated difference of means are similar:

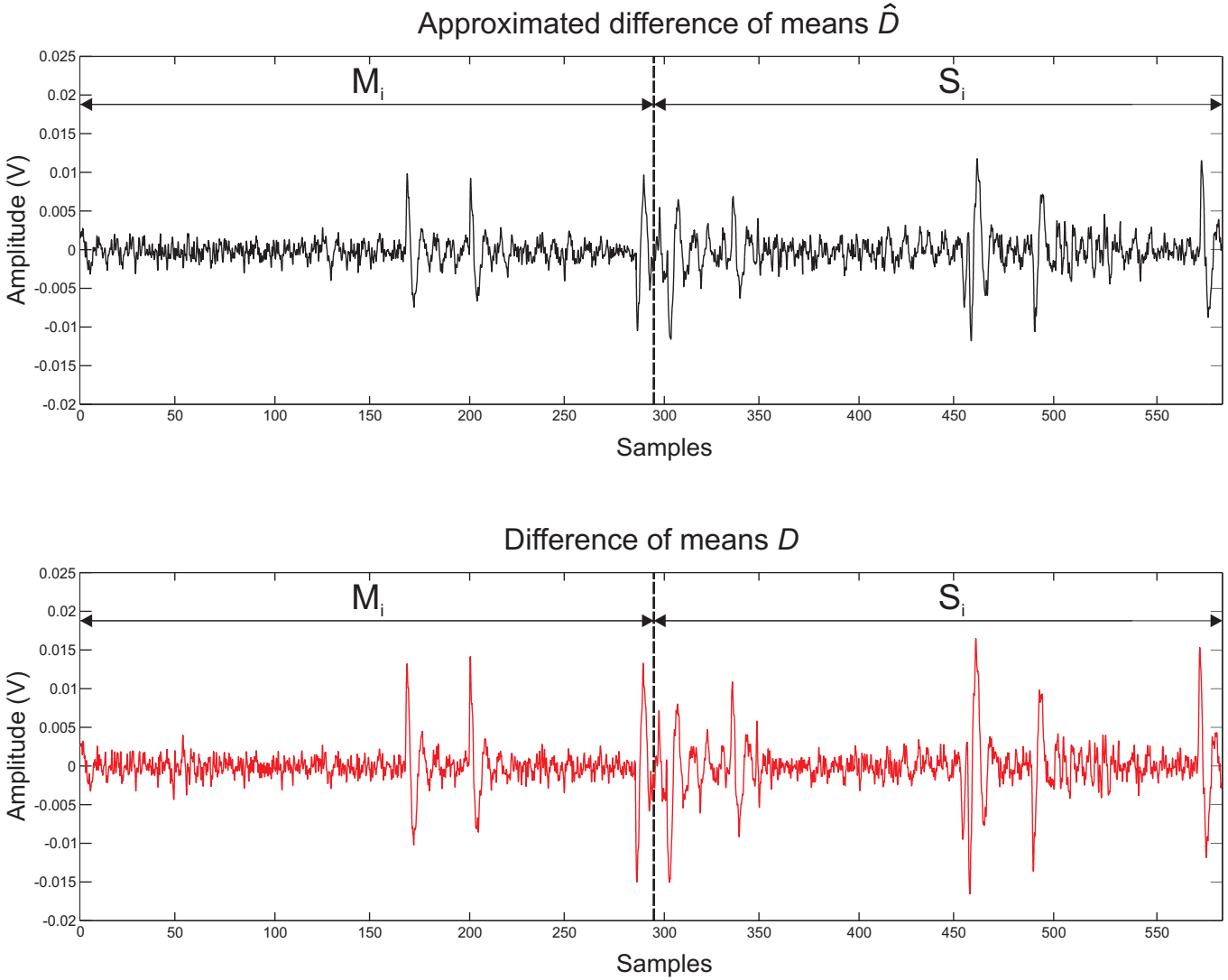


$\{t_{i:\ell,285}\}$

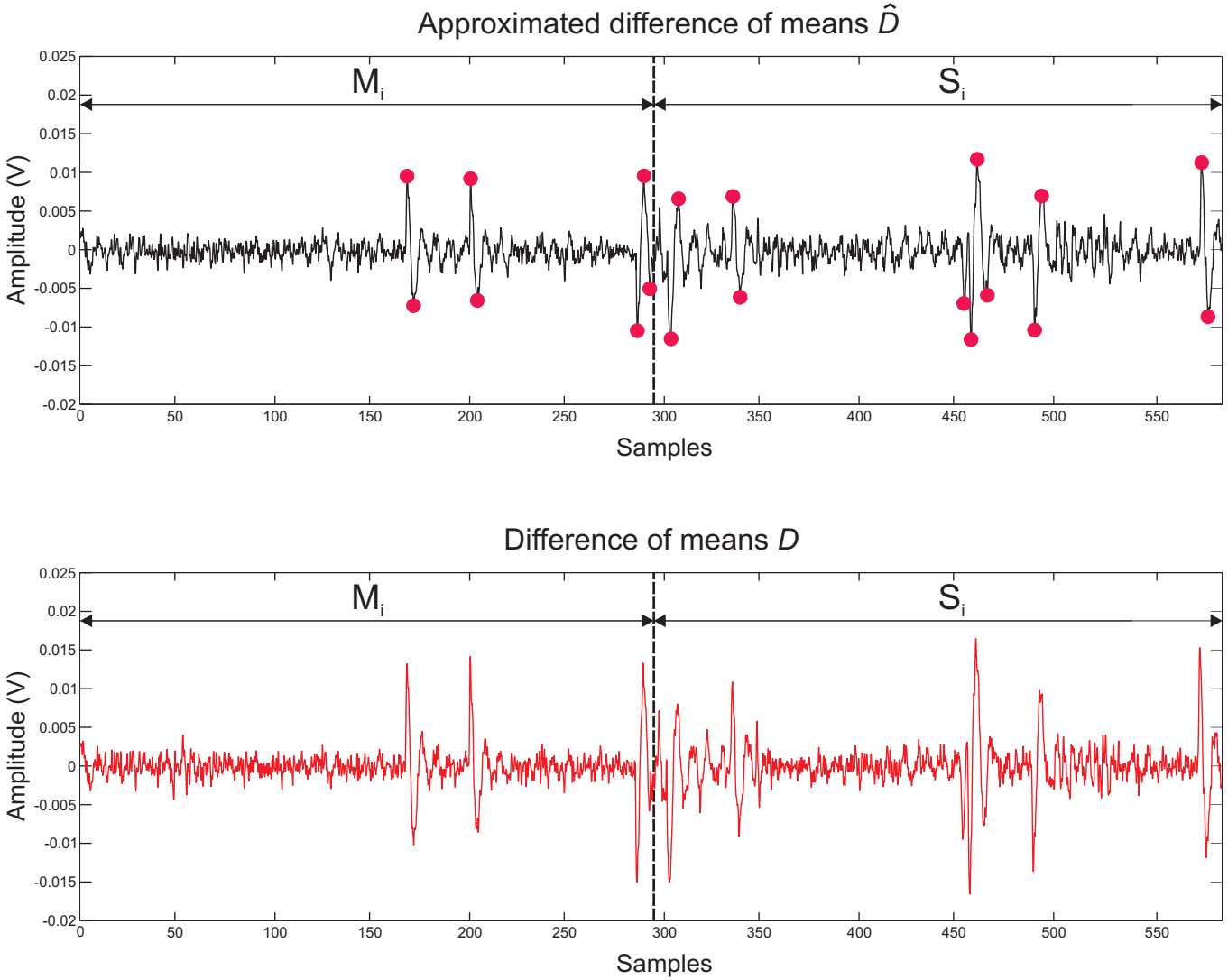


$\{t_{i:\ell,464}\}$

Points of Interest



Points of Interest



Clustering

- We selected $u = 17$ points of interest among 592 possibilities ($p_j =$ 165, 166, 169, 281, 282, 285, 285, 342, 461, 462, 464, 465, 497, 498, 577, 580, 581)
- The *fuzzy* k-means returns two centers μ_1 and μ_2, representing two classes (0,1)
- With $u = 17$ points there will be 2^{17} different possibilities

Clustering

- We selected $u = 17$ points of interest among 592 possibilities ($p_j =$ 165, 166, 169, 281, 282, 285, 285, 342, 461, 462, 464, 465, 497, 498, 577, 580, 581)
- The *fuzzy* k-means returns two centers μ_1 and μ_2, representing two classes (0,1)
- With $u = 17$ points there will be 2^{17} different possibilities

Classes association:

1. Initialize $\mu_1 = min\{x\}$, $\mu_2 = max\{x\}$.
   - Then, $\mu_1 < \mu_2$ at the end of clustering.
   - Compare μ_1 and μ_2 with $\widehat{D}$
2. Take one recovered exponent $\widehat{d_{1:\ell,v}}$, $v \in \{p_j\}$:
   - $h_{1:\ell} = \text{XOR}(\widehat{d_{1:\ell,v}}, \widehat{d_{1:\ell,p_j}})$, $\forall p_j$.
   - If $\sum_{i=1}^{\ell} h_i < \ell/2$ then $\text{NOT}(d_{1:\ell,p_j})$, otherwise keep unchanged.

Clustering

- We selected $u = 17$ points of interest among 592 possibilities ($p_j =$ 165, 166, 169, 281, 282, 285, 285, 342, 461, 462, 464, 465, 497, 498, 577, 580, 581)
- The *fuzzy* k-means returns two centers μ_1 and μ_2, representing two classes (0,1)
- With $u = 17$ points there will be 2^{17} different possibilities

Classes association:

1. Initialize $\mu_1 = min\{x\}$, $\mu_2 = max\{x\}$.
   - Then, $\mu_1 < \mu_2$ at the end of clustering.
   - Compare μ_1 and μ_2 with $\widehat{D}$
2. Take one recovered exponent $\widehat{d_{1:\ell,v}}$, $v \in \{p_j\}$:
   - $h_{1:\ell} = \text{XOR}(\widehat{d_{1:\ell,v}}, \widehat{d_{1:\ell,p_j}})$, $\forall p_j$.
   - If $\sum_{i=1}^{\ell} h_i < \ell/2$ then $\text{NOT}(d_{1:\ell,p_j})$, otherwise keep unchanged.

Ex: $\ell = 10$, $d_{1:10,1} = 100000100$, $d_{1:10,2} = 0111011011$
$h_{1:10} = 1111011111$, $\sum_{i=1}^{\ell} h_i = 9$

Exponent Recovery - Majority Decision

exponent	classified exponent bits $\widehat{d_{i,k}}$																																							correct
$\widehat{d_{1:40,497}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	76.02%
$\widehat{d_{1:40,498}}$	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	1	76.42%
$\widehat{d_{1:40,166}}$	1	1	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	1	76.42%
$\widehat{d_{1:40,462}}$	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	0	1	0	0	1	1	1	1	0	0	1	0	76.42%
$\widehat{d_{1:40,580}}$	1	1	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	1	1	1	1	1	1	0	0	0	1	1	1	1	0	0	1	0	0	1	0	0	0	1	78.86%
$\widehat{d_{1:40,342}}$	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	0	0	1	0	1	0	0	1	0	1	79.27%
$\widehat{d_{1:40,282}}$	1	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1	1	1	1	1	1	0	0	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	78.86%
$\widehat{d_{1:40,281}}$	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	1	80.08%
$\widehat{d_{1:40,169}}$	1	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1	0	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	80.08%
$\widehat{d_{1:40,165}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1	80.89%
$\widehat{d_{1:40,581}}$	1	0	1	0	0	0	0	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	0	1	82.52%
$\widehat{d_{1:40,577}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1	0	0	1	1	1	0	1	1	0	1	1	0	0	0	0	0	1	82.52%
$\widehat{d_{1:40,284}}$	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	0	1	83.74%
$\widehat{d_{1:40,285}}$	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1	89.43%
$\widehat{d_{1:40,465}}$	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	1	90.65%
$\widehat{d_{1:40,461}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	1	91.25%
$\widehat{d_{1:40,464}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	93.06%
$\widehat{d_{1:40,k}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	1	100%
$d_{1:40,k}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	1	100%

Exponent Recovery - Probability Density Function

- According to [1], *"single bits are never known with certainty* [...] *and an SPA attacker* [...] *can only give a probability that any particular operation is a squaring or a multiplication"*;
- So, we obtained the probability that each exponent bit is zero or one;
- To compute the pdf, we need the means μ and σ of classes:
  - Means $\longrightarrow$ cluster centers μ_1, μ_2
  - Variance $\longrightarrow$ standard deviation of all set of samples $\sigma_{t_{i,p_j}}$

$$P(t_{i,p_j}, \mu_1) = \frac{e^{-\frac{1}{2}(t_{i,p_j}-\mu_1)^2/2\sigma^2}}{e^{-\frac{1}{2}(t_{i,p_j}-\mu_1)^2/2\sigma^2} + e^{-\frac{1}{2}(t_{i,p_j}-\mu_2)^2/2\sigma^2}}, \quad 1 \leq i \leq \ell, 1 \leq j \leq u$$

---

[1] Sven Bauer, "Attacking exponent blinding in RSA without CRT", COSADE 2012.

Exponent Recovery - Probability Density Function

$$S_{0,1:u} = \frac{1}{u}\sum_{j=1}^{u} P(t_{i,p_j}, \mu_1) \quad 1 \le i \le \ell \quad \widehat{d_{i,k}} = \begin{cases} 0, & \text{if} \quad S_{0,1:u} \ge 0.5) \\ 1, & \text{if} \quad S_{0,1:u} < 0.5) \end{cases}$$

point	probabilities $p(t_{i,p_j}, \mu_1)$																				correct
$\widehat{d_{1:20,p_1}}$	0.1	1.0	0.5	0.6	0.7	0.9	0.9	0.4	1.0	0.7	0.6	0.9	0.0	0.8	0.0	0.2	0.3	0.7	0.5	0.1	76.02%
$\widehat{d_{1:20,p_2}}$	0.3	1.0	0.8	0.6	0.8	0.5	0.4	0.1	0.9	0.7	0.9	0.5	0.7	0.9	0.5	0.2	0.8	0.4	0.1	0.2	76.42%
$\widehat{d_{1:20,p_3}}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	76.83%
$\widehat{d_{1:20,p_4}}$	0.2	0.9	0.9	0.8	0.6	0.3	0.5	0.2	0.5	0.3	0.7	0.9	0.6	0.2	0.1	0.2	0.8	0.2	0.1	0.1	76.42%
$\widehat{d_{1:20,p_5}}$	0.5	0.5	0.9	0.9	0.9	0.6	0.8	0.3	0.9	0.1	0.8	0.3	0.2	0.6	0.7	0.1	0.8	0.0	0.3	0.3	78.86%
$\widehat{d_{1:20,p_6}}$	0.1	0.7	0.5	0.4	0.8	0.7	0.7	0.9	0.5	1.0	0.8	0.6	0.2	0.9	0.5	0.3	0.9	0.6	0.7	0.4	78.86%
$\widehat{d_{1:20,p_7}}$	0.5	0.8	0.9	0.8	0.9	0.8	0.8	0.0	0.9	0.6	0.9	0.5	0.1	0.6	0.2	0.1	0.9	0.0	0.3	0.1	78.86%
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
$\widehat{d_{1:20,p_{16}}}$	0.2	0.8	1.0	0.8	1.0	0.9	0.9	0.3	0.5	0.9	0.7	0.7	0.0	0.8	0.1	0.7	0.8	0.2	0.0	0.2	90.65%
$\widehat{d_{1:20,p_{17}}}$	0.3	0.7	0.6	0.6	1.0	0.7	1.0	0.1	0.8	1.0	0.8	0.7	0.3	0.8	0.3	0.1	0.6	0.5	0.3	0.1	93.06%
$S_{0,1:17}$	0.3	0.6	0.7	0.6	0.7	0.6	0.6	0.3	0.7	0.7	0.7	0.6	0.2	0.6	0.3	0.3	0.7	0.3	0.2	0.3	100%
$\widehat{d_{1:20,k}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	100%
$d_{1:20,k}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	100%

Exponent Recovery - Bayesian Classifier

Means $\longrightarrow$ cluster centers μ_1, μ_2
Variance $\longrightarrow$ standard deviation of all set of samples $\sigma_{t_{i,p_j}}$

$$p(t_{i,p_j}, \mu_1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t_{i,p_j}-\mu_1)^2}{2\sigma^2}}$$

$$p(t_{i,p_j}, \mu_2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t_{i,p_j}-\mu_2)^2}{2\sigma^2}}$$

$$P(\mu_1|t_{i,p_j}) = \frac{p(t_{i,p_j}, \mu_1)P(\mu_1)}{p(t_{i,p_j}, \mu_1)P(\mu_1) + p(t_{i,p_j}, \mu_2)P(\mu_2)}$$

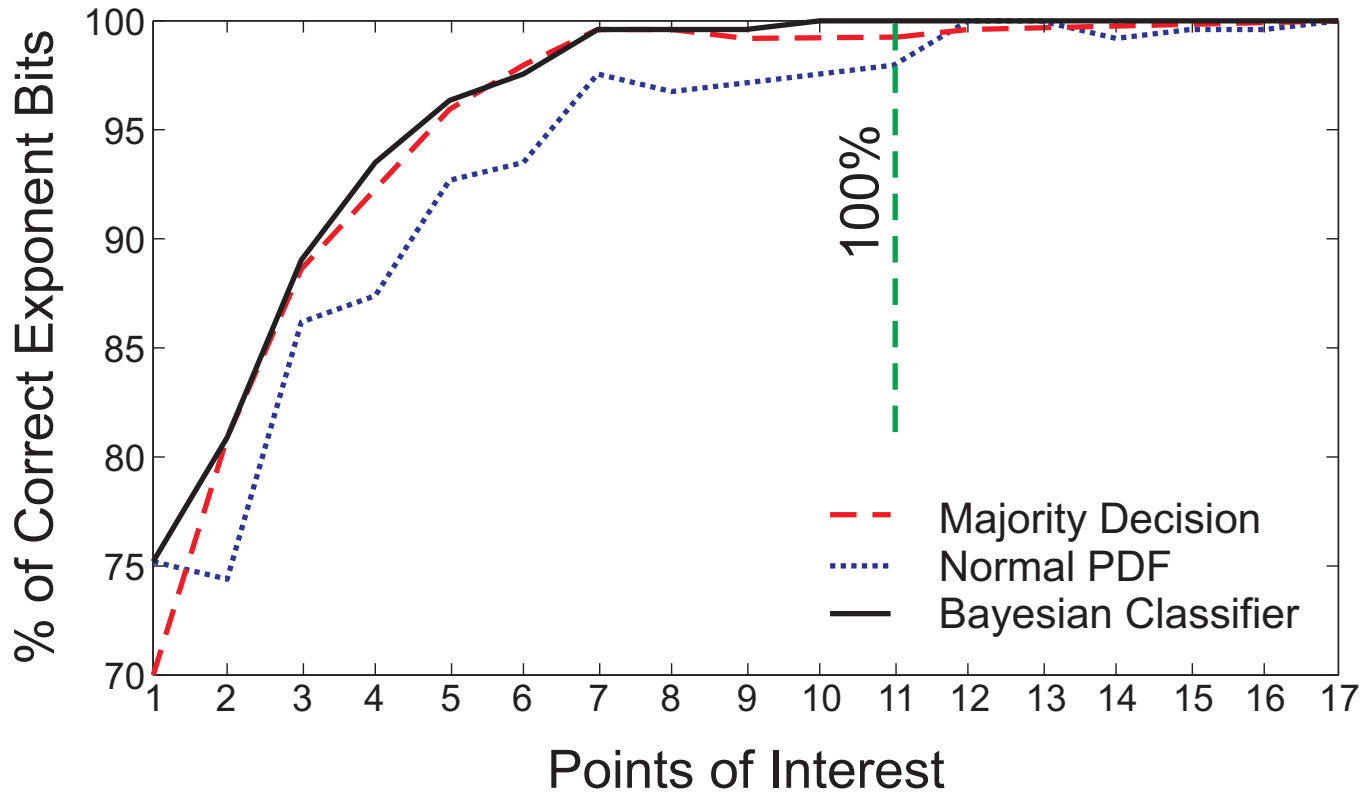
$$P(\mu_2|t_{i,p_j}) = \frac{p(t_{i,p_j}, \mu_2)P(\mu_2)}{p(t_{i,p_j}, \mu_1)P(\mu_1) + p(t_{i,p_j}, \mu_2)P(\mu_2)}$$

Exponent Recovery - Bayesian Classifier

point	probabilities $P(\mu_1 \mid t_{1:20,p_j})$																				correct
$P(\mu_1 \mid t_{1:20,p_1})$	0.7	0.4	0.6	0.8	0.7	0.8	0.6	0.4	0.8	0.8	0.9	0.8	0.4	0.7	0.4	0.3	0.6	0.2	0.4	0.6	75.23%
$P(\mu_1 \mid t_{1:20,p_2})$	0.5	1.0	0.8	0.9	0.7	0.9	0.4	0.2	1.0	1.0	0.9	0.8	0.2	0.9	0.0	0.0	0.8	0.2	0.1	0.6	82.89%
$P(\mu_1 \mid t_{1:20,p_3})$	0.5	1.0	0.8	0.9	0.7	1.0	0.5	0.1	1.0	1.0	0.9	0.8	0.1	0.9	0.0	0.1	0.9	0.3	0.1	0.6	89.02%
$P(\mu_1 \mid t_{1:20,p_4})$	0.1	1.0	0.9	1.0	0.7	0.9	0.9	0.0	1.0	1.0	1.0	0.8	0.0	1.0	0.0	0.0	1.0	0.1	0.0	0.1	93.45%
$P(\mu_1 \mid t_{1:20,p_5})$	0.4	1.0	1.0	1.0	0.9	1.0	0.8	0.0	0.9	1.0	1.0	0.6	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.3	96.34%
$P(\mu_1 \mid t_{1:20,p_6})$	0.4	1.0	1.0	1.0	0.9	1.0	0.8	0.0	0.9	1.0	1.0	0.6	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.3	97.56%
$P(\mu_1 \mid t_{1:20,p_7})$	0.4	1.0	0.9	1.0	1.0	1.0	0.7	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.3	99.59%
$P(\mu_1 \mid t_{1:20,p_8})$	0.0	1.0	1.0	1.0	1.0	1.0	0.7	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	99.59%
$P(\mu_1 \mid t_{1:20,p_9})$	0.0	1.0	1.0	1.0	1.0	1.0	0.8	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	99.59%
$P(\mu_1 \mid t_{1:20,p_{10}})$	0.0	1.0	1.0	1.0	1.0	1.0	0.9	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	99.18%
$P(\mu_1 \mid t_{1:20,p_{11}})$	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	100.00%
$P(\mu_1 \mid t_{1:20,p_{12}})$	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	100.00%
$P(\mu_1 \mid t_{1:20,p_{13}})$	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	100.00%
$P(\mu_1 \mid t_{1:20,p_{14}})$	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	100.00%
$P(\mu_1 \mid t_{1:20,p_{15}})$	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	100.00%
$P(\mu_1 \mid t_{1:20,p_{16}})$	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	100.00%
$P(\mu_1 \mid t_{1:20,p_{17}})$	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	100.00%
$\widehat{d_{1:20,k}}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	100%
$d_{1:20,k}$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	100%

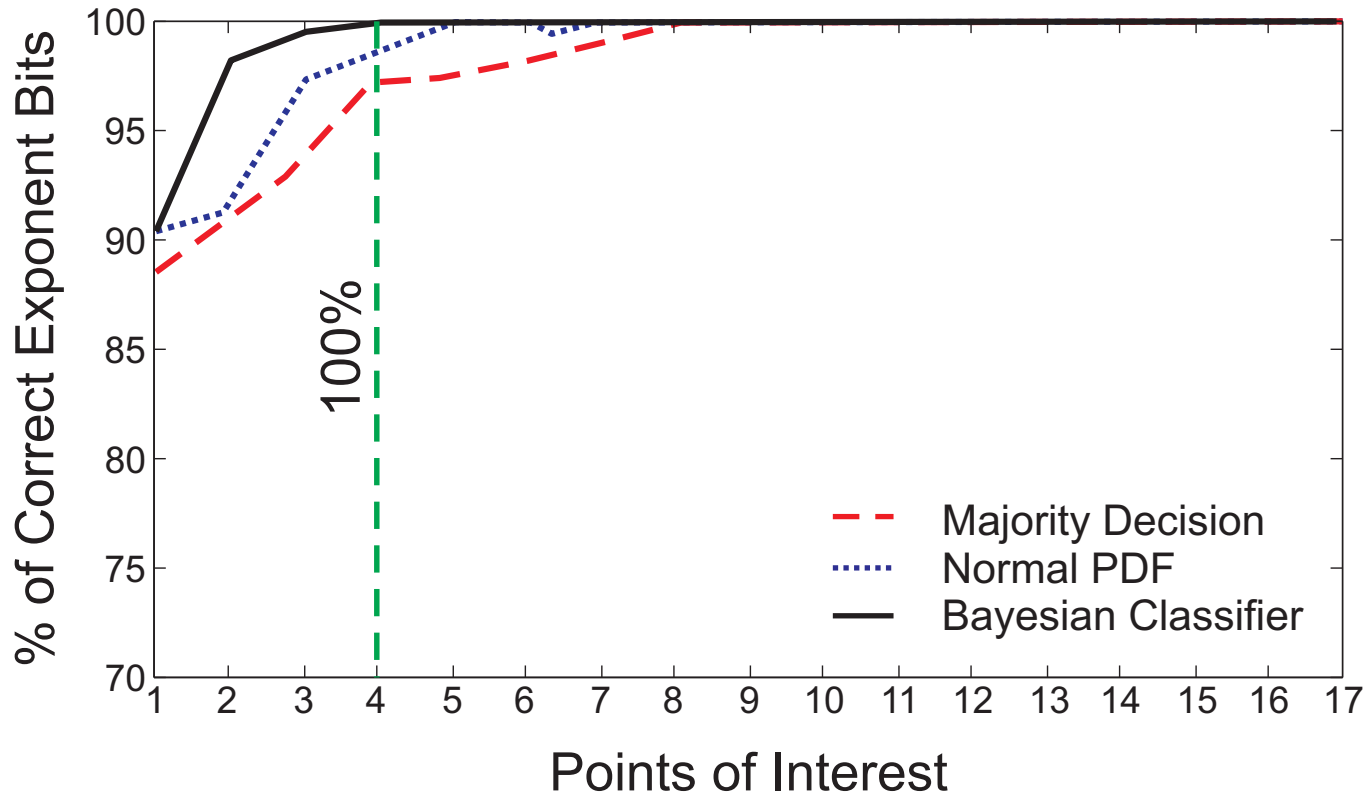
Exponent Recovery

From the least to the most leaking point



Exponent Recovery

From the most to the least leaking point



Conclusions

- The presented attack can recover the entire exponent from a **single trace** by exploring the leakage through a **horizontal** mode using **clustering algorithms**;
- It explores the **SPA-leakages** related to address-bit;
- Countermeasures: hardware countermeasures based on **RAM addressing randomization** (by doing so, 80% of exponent was recovered using the Bayesian classifier);
- The unsupervised attack can be applied in ECC and CRT-RSA;

Thank you for your attention

Questions?

perin@lirmm.fr