Fraunhofer AISEC

Localized Electromagnetic Analysis of Cryptographic Implementations

Johann Heyszl Stefan Mangard[1]
Benedikt Heinz Frederic Stumpf Georg Sigl[2]

CT-RSA 2012, San Francisco
March 1, 2012

[1] Infineon Technologies AG
[2] Technische Universität München

Outline

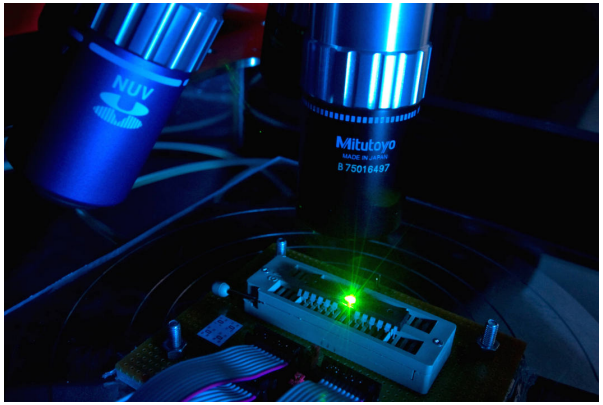
1. Motivation
2. Localized EM and Side-Channel Attacks
3. ECC Case Study - Proof-of-Concept
4. Conclusion

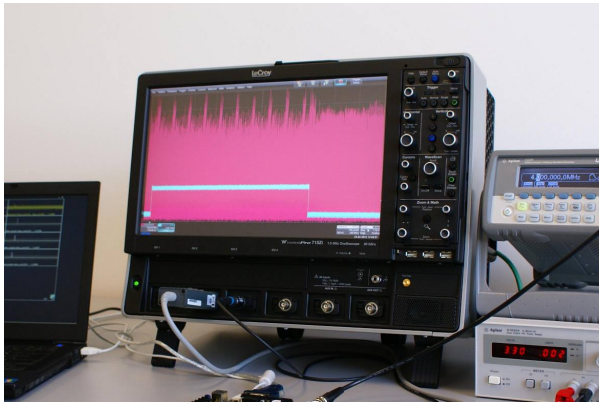
Outline

Fraunhofer AISEC

1. Motivation
2. Localized EM and Side-Channel Attacks
3. ECC Case Study - Proof-of-Concept
4. Conclusion

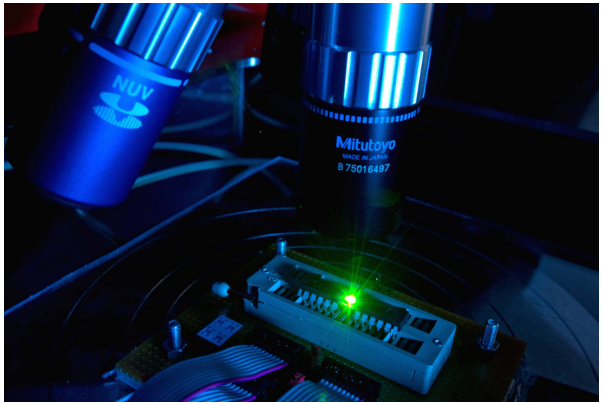
Motivation

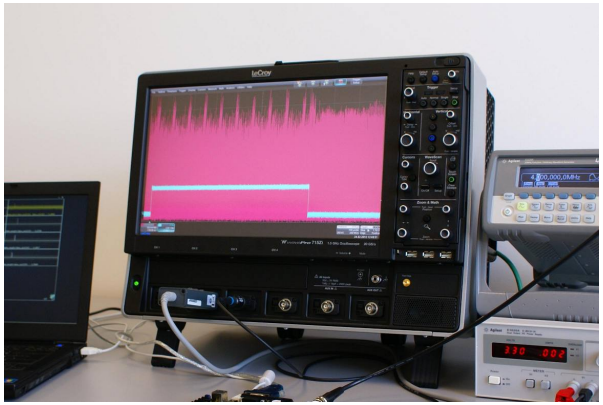




- Physical security of cryptographic implementations.
  - Information leakage through active or passive attacks.
- Passive side-channel analysis.
  - Recover secret keys through side-channel leakage.

Motivation

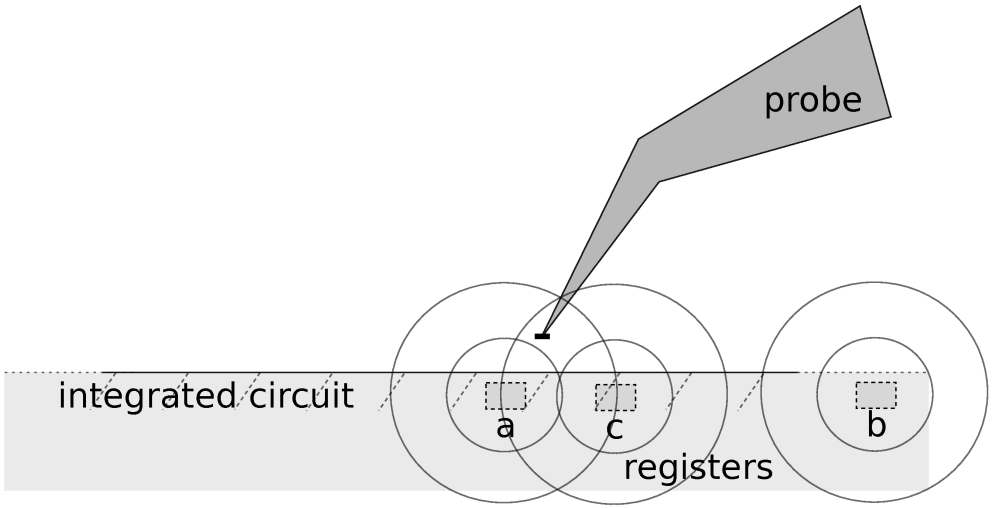




- Physical security of cryptographic implementations.
  - Information leakage through active or passive attacks.
- Passive side-channel analysis.
  - Recover secret keys through side-channel leakage.
- Electro-magnetic radiation.
  - First derivation of current consumption leakage.

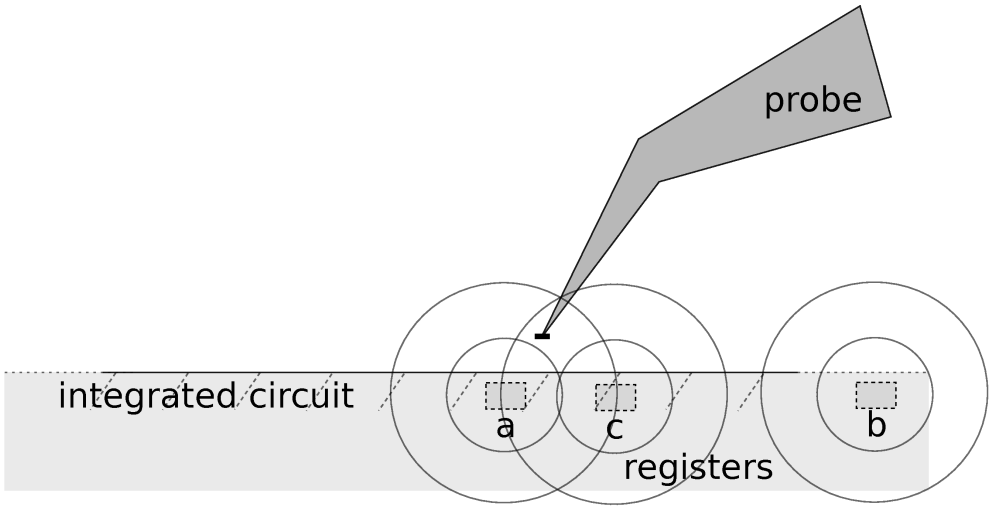
Motivation

- Localized EM analysis.
  - Spatially restrict EM measurements to parts of integrated circuit.
  - Working hypothesis → Distinguish use of registers.



Fraunhofer AISEC

- Localized EM analysis.
  - Spatially restrict EM measurements to parts of integrated circuit.
  - Working hypothesis → Distinguish use of registers.



- How to use for side-channel attacks?

Outline

Localized EM for Side-Channel Attacks

Fraunhofer AISEC

- Using localized EM analysis for side-channel attacks.
- Exploit location dependence instead of data dependence or operation dependence.
  - Depends on algorithm.
  - Location-dependence must leak information about secret.

Attacking Binary Exponentiation Algorithms

Fraunhofer AISEC

- Binary exponentiation algorithms.
- Used in public key cryptography.
  - Modular exponentiations in RSA.
  - Elliptic curve scalar multiplications in ECC.
  - E.g. square-and-multiply-always (RSA), double-and-add-always (ECC), Montgomery ladder (RSA, ECC) algorithms.

Attacking Binary Exponentiation Algorithms

Fraunhofer AISEC

- Binary exponentiation algorithms.
- Used in public key cryptography.
  - Modular exponentiations in RSA.
  - Elliptic curve scalar multiplications in ECC.
  - E.g. square-and-multiply-always (RSA), double-and-add-always (ECC), Montgomery ladder (RSA, ECC) algorithms.
- Key features.
  - Bit-wise processing of secret in loop.
  - Operation sequence uniform for each bit.
  - Register usage depends on secret bits.
    E.g. two alternately used registers, depending on current bit.

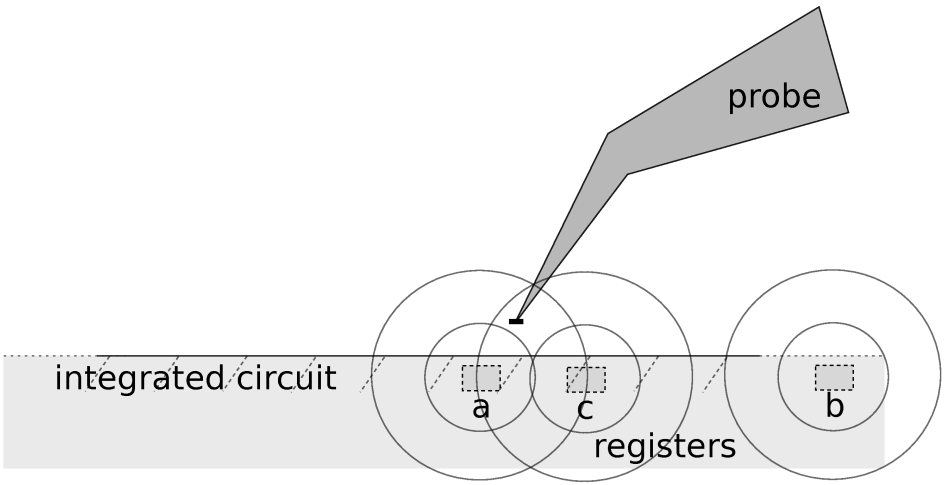
Attacking Binary Exponentiation Algorithms

Fraunhofer AISEC

- Binary exponentiation pseudo-algorithm.

Input: Secret $d = d_D d_{D-1} \ldots d_2 d_1$ with $d_i \in \{0, 1\}$

```
 1: for i = D downto 1 do
 2:     if d_i = 1 then
 3:         c ← a
 4:         c ← c^2
 5:         a ← c
 6:     else
 7:         c ← b
 8:         c ← c^2
 9:         b ← c
10:     end if
11: end for
```



- According to hypothesis,
  EM radiation from logic of e.g., *a* leads to greater amplitudes if probe is closer to *a*.

Attacking Binary Exponentiation Algorithms

Fraunhofer AISEC

- Employ established attacks.
  - E.g. template attack with known exponent.
- Detect usage sequence to recover secret.

Outline

Fraunhofer AISEC

1. Motivation
2. Localized EM and Side-Channel Attacks
3. ECC Case Study - Proof-of-Concept
4. Conclusion

ECC Case Study

Fraunhofer AISEC

- Overview.
  - Proof-of-concept.
  - Attacking elliptic curve scalar multiplication.
  - FPGA-based HW implementation.
  - High-precision EM measurement setup.
  - Template attack to exploit localized EM.

ECC Implementation

Fraunhofer AISEC

- Elliptic curve scalar multiplication $Q = d \cdot P$
- EC over binary field $GF(2^{163})$, NIST *Curve B-163* parameters.
- López and Dahab Montgomery ladder algorithm.
- Affine x- and y-coordinates as input and output.

- Fulfills requirements.
  - Bitwise processing of **163** bit scalar.
  - Register usage depends on secret bits.
  - Uniform operation sequence.

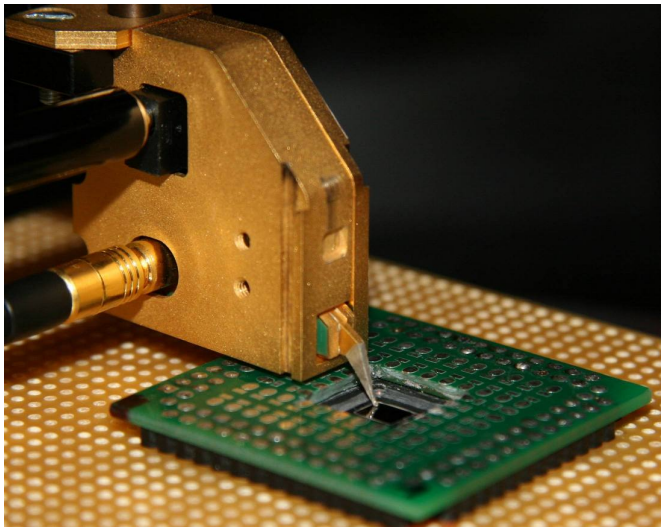
Fraunhofer AISEC

- López and Dahab Montgomery ladder.

Input: Scalar $d = d_D d_{D-1} \ldots d_2 d_1$ with $d_i \in \{0, 1\}$,
Point $P = (x_P, y_P) \in E$, Curve Parameter b
Output: Point $Q = d \cdot P = (x_Q, y_Q)$

1: $X_0 \leftarrow 1, Z_0 \leftarrow 0, X_1 \leftarrow x_P, Z_1 \leftarrow 1$
2: **for** $i = D$ downto 1 **do**
3: $\quad T \leftarrow Z_{1-d_i}$
4: $\quad Z_{1-d_i} \leftarrow (X_{1-d_i} \cdot Z_{d_i} + X_{d_i} \cdot Z_{1-d_i})^2$
5: $\quad X_{1-d_i} \leftarrow x_P \cdot Z_{1-d_i} + X_{1-d_i} \cdot X_{d_i} \cdot T \cdot Z_{d_i}$
6: $\quad T \leftarrow X_{d_i}$
7: $\quad X_{d_i} \leftarrow X_{d_i}^4 + b \cdot Z_{d_i}^4$
8: $\quad Z_{d_i} \leftarrow T^2 \cdot Z_{d_i}^2$
9: **end for**
10: $(x_Q, y_Q) \leftarrow Mxy(X_0, Z_0, X_1, Z_1, x_P, y_P)$ {Computation of affine coordinates.}
11: **return** (x_Q, y_Q)

Measurement Setup



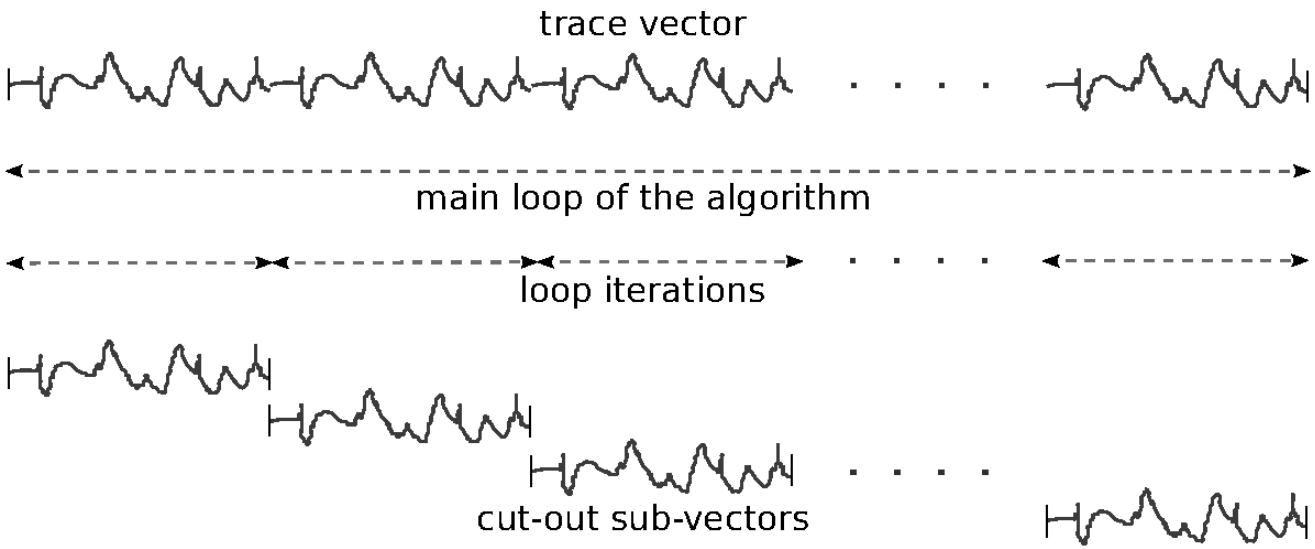
- Backside-decapsulated Xilinx Spartan 3 FPGA.
- x-y-table with step length of **50** μm.
- Inductive, near-field probe with **100** μm resolution.
- **5** GS/s sampling.
- Compressed using one peak to peak distance sample per cycle.

Template Attack - Process Steps

- Observe public ECC operation multiple times.
  - Known exponent (e.g., signature verification).

Template Attack - Process Steps

- Observe public ECC operation multiple times.
  - Known exponent (e.g., signature verification).
- Record traces and split into sub-vectors.
  - Same operation sequence in all loop iterations.
  - Each loop iteration sub-vector, different secret bit.



Template Attack - Process Steps, cont'd

Fraunhofer AISEC

- Group into two sets according to known exponent bits.
  - Difference-of-means between bit-**0** and bit-**1** set.

Template Attack - Process Steps, cont'd

Fraunhofer AISEC

- Group into two sets according to known exponent bits.
  - Difference-of-means between bit-**0** and bit-**1** set.
- Find eligible location on die.
  - Highest difference-of-means between bit-**0** and bit-**1** set.

Template Attack - Process Steps, cont'd

Fraunhofer AISEC

- Group into two sets according to known exponent bits.
  - Difference-of-means between bit-**0** and bit-**1** set.
- Find eligible location on die.
  - Highest difference-of-means between bit-**0** and bit-**1** set.
- Build templates.
  - Different to template attacks on data-dependent leakage.
  - Only two templates each covering one loop iteration.
  - Public operation can be used (regardless of different base).
  - Two reduced templates: means of each sub-vector sets.

Template Attack - Process Steps, cont'd

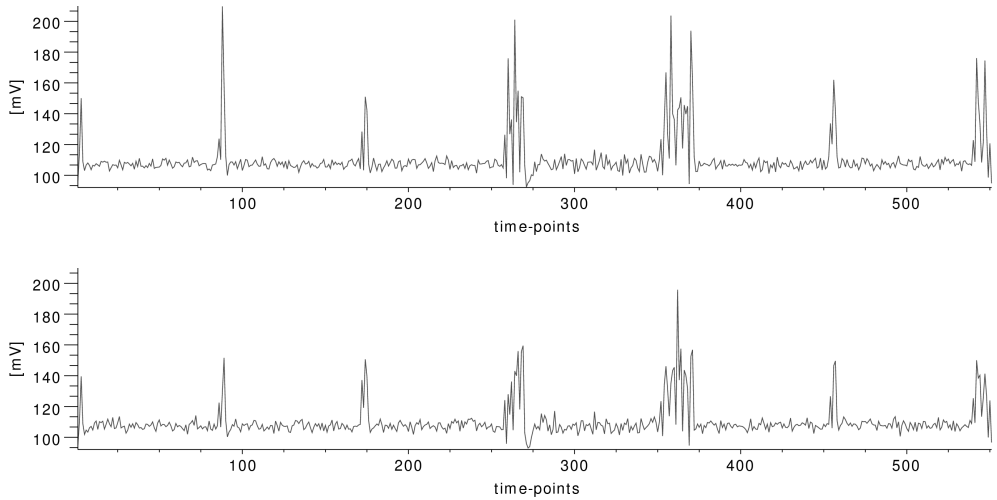
Fraunhofer AISEC

- Group into two sets according to known exponent bits.
  - Difference-of-means between bit-**0** and bit-**1** set.
- Find eligible location on die.
  - Highest difference-of-means between bit-**0** and bit-**1** set.
- Build templates.
  - Different to template attacks on data-dependent leakage.
  - Only two templates each covering one loop iteration.
  - Public operation can be used (regardless of different base).
  - Two reduced templates: means of each sub-vector sets.
- Attack private ECC operation using single trace.
  - On best location.
  - Using built templates.
  - Least-square matching.

Group into Two Sets & Difference-of-Means

Known Exponent Analysis

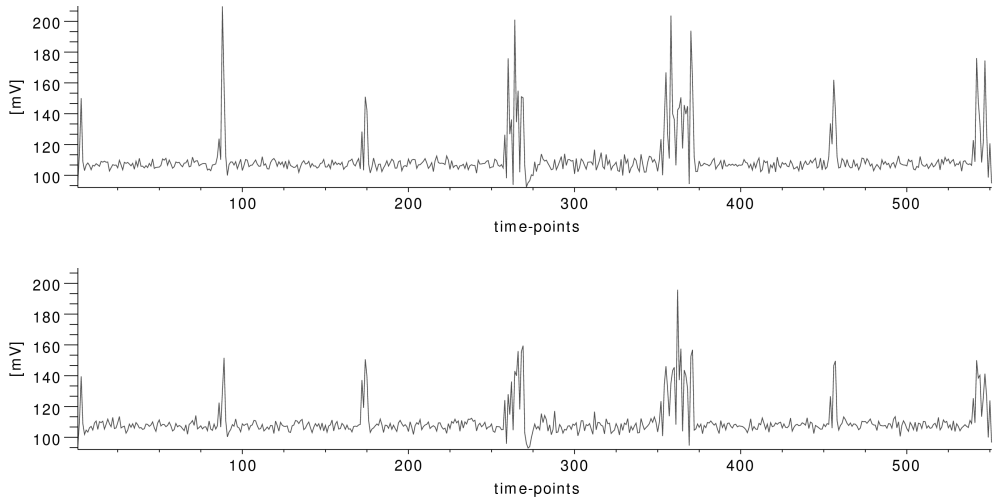
- Mean vectors for bit-**0** and bit-**1** set (one loop iteration).



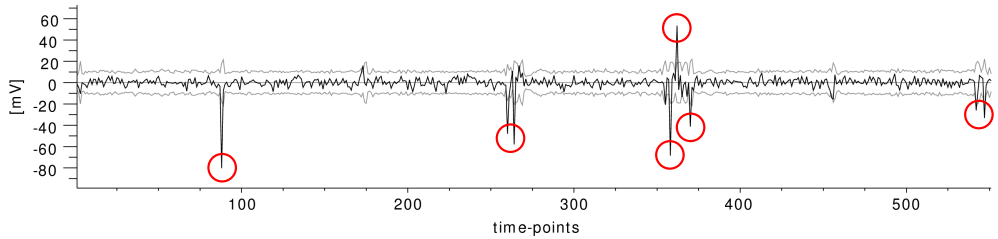
Group into Two Sets & Difference-of-Means

Known Exponent Analysis

- Mean vectors for bit-**0** and bit-**1** set (one loop iteration).



- Difference-of-means. Test using confidence interval.



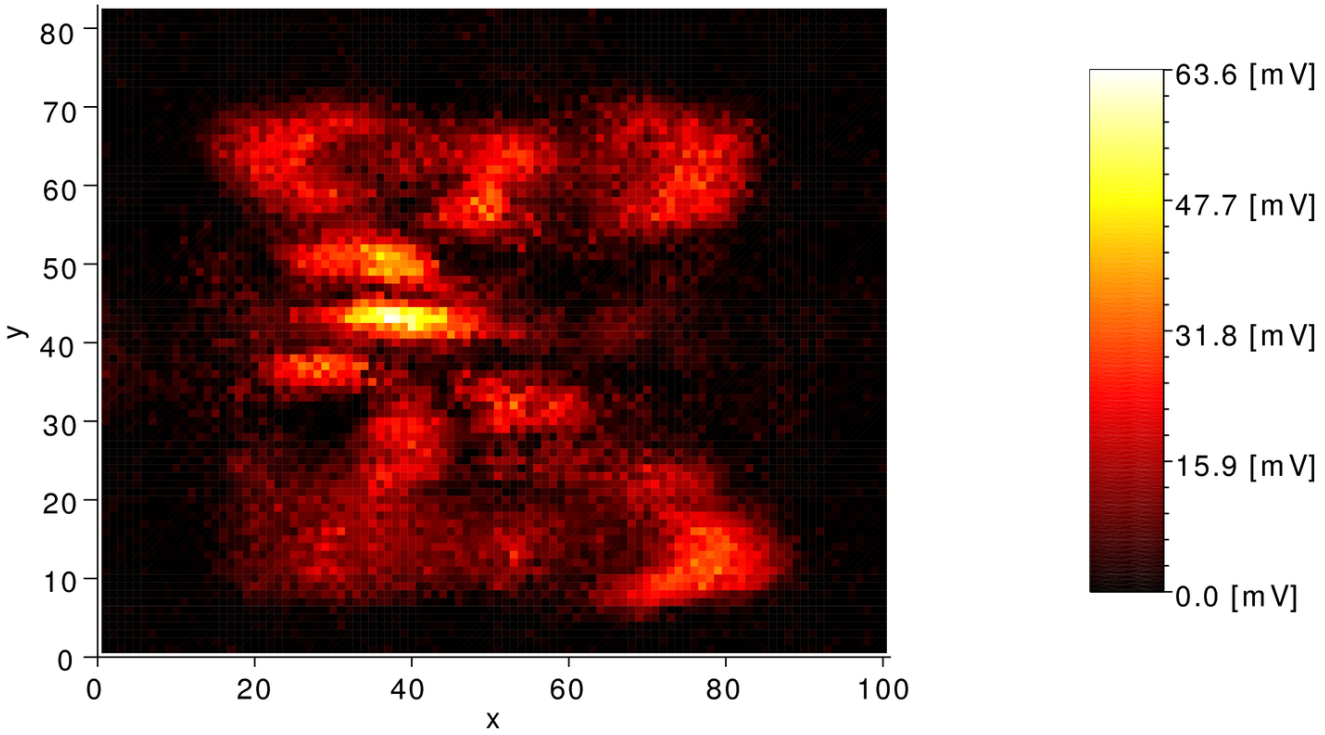
  - Significant difference in multiple cycles (e.g., **88**).

Find Eligible Locations

Known Exponent Analysis

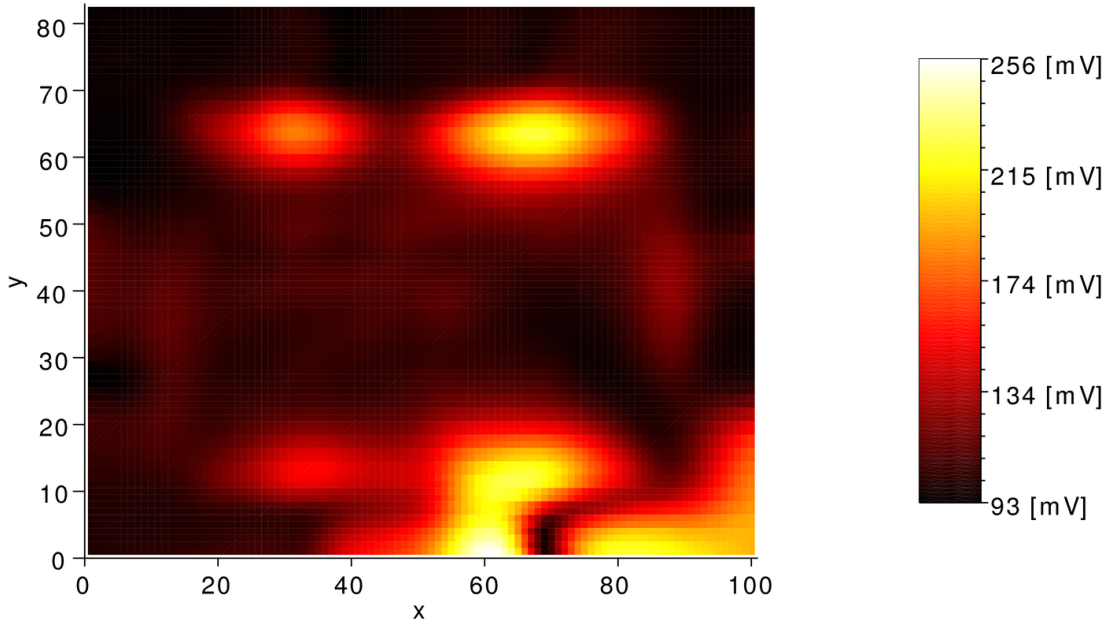
Fraunhofer AISEC

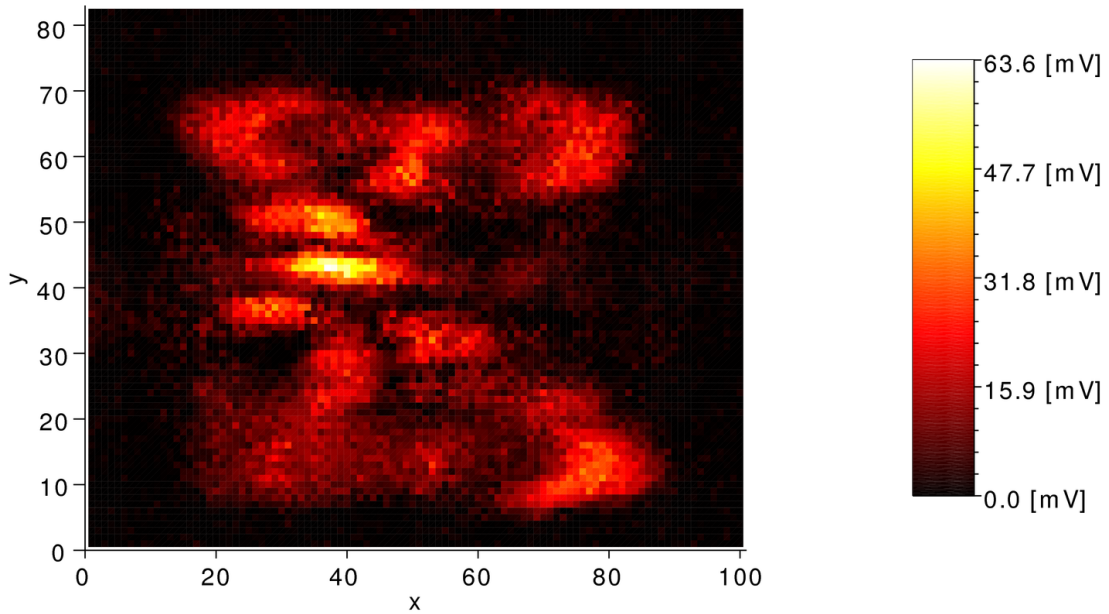
- Greatest absolute difference-of-means on die.

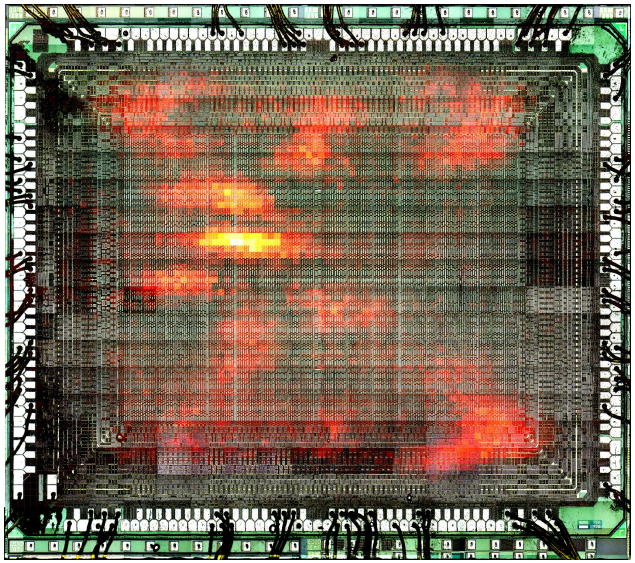


- Regions with significant difference-of-means!

Known Exponent Analysis

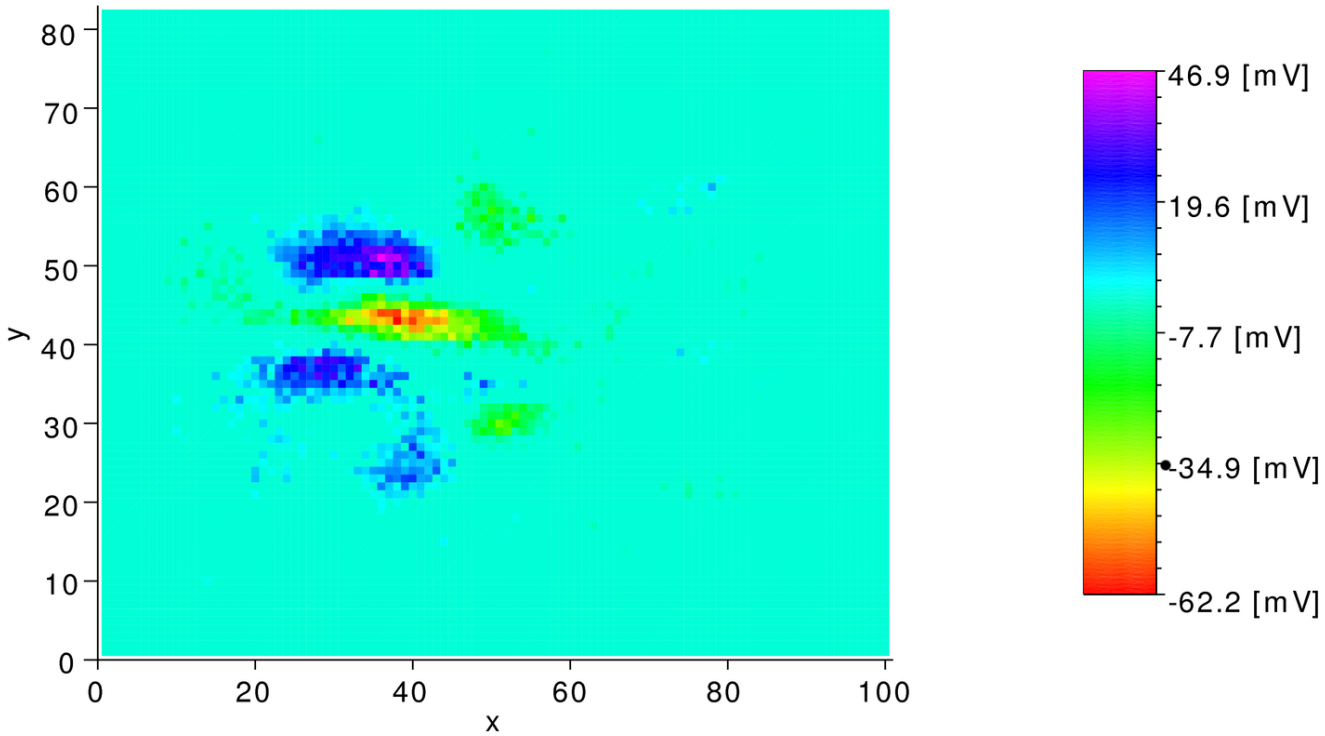






Fraunhofer AISEC

- Example for better understanding.
  - Only cycle **88** analyzed ($X_{1-d_i} \leftarrow X_{1-d_i} \cdot Z_{d_i}$).
  - Signed difference-of-means.



- Positive difference → closer to **0**-registers.
- Negative difference → closer to **1**-registers.

Template Attack

- Choose location with greatest difference-of-means.

Template Attack

- Choose location with greatest difference-of-means.
- Record single trace of private operation at this location.

Template Attack

- Choose location with greatest difference-of-means.
- Record single trace of private operation at this location.
- Segment into sub-vectors.

Template Attack

- Choose location with greatest difference-of-means.
- Record single trace of private operation at this location.
- Segment into sub-vectors.
- Match to templates using least-square distance.

Template Attack

Fraunhofer AISEC

- Choose location with greatest difference-of-means.
- Record single trace of private operation at this location.
- Segment into sub-vectors.
- Match to templates using least-square distance.
- **Results from our case study:**
  - Correct classification of **161** of **163** bits.
- **Proves high significance of location-based leakage.**

Countermeasures

- Many countermeasures do not prevent location-based leakage.
  - Montgomery ladder.
  - Projective coordinate randomization.
  - Base point blinding.
  - Exponent blinding.
    - Prevents template attacks.
    - Does not prevent collision attacks.
- Location-based leakage **only** prevented by randomizing physical locations of registers.

Our Countermeasure

For López-Dahab Montgomery Ladder

Fraunhofer AISEC

- Randomization of physical location of variables.
- At end of every iteration in main loop, perform:

9: $r \leftarrow random \in [0, 1]$
10: $c \leftarrow swap\_state \oplus r$
11: $T \leftarrow X_0 + X_1$ {*swap* X_0 *and* X_1 *if* $c = 1$}
12: $X_0 \leftarrow T - X_{1-c}$
13: $X_1 \leftarrow T - X_c$
14: $T \leftarrow Z_0 + Z_1$ {*swap* Z_0 *and* Z_1 *if* $c = 1$}
15: $Z_0 \leftarrow T - Z_{1-c}$
16: $Z_1 \leftarrow T - Z_c$
17: $swap\_state \leftarrow r$

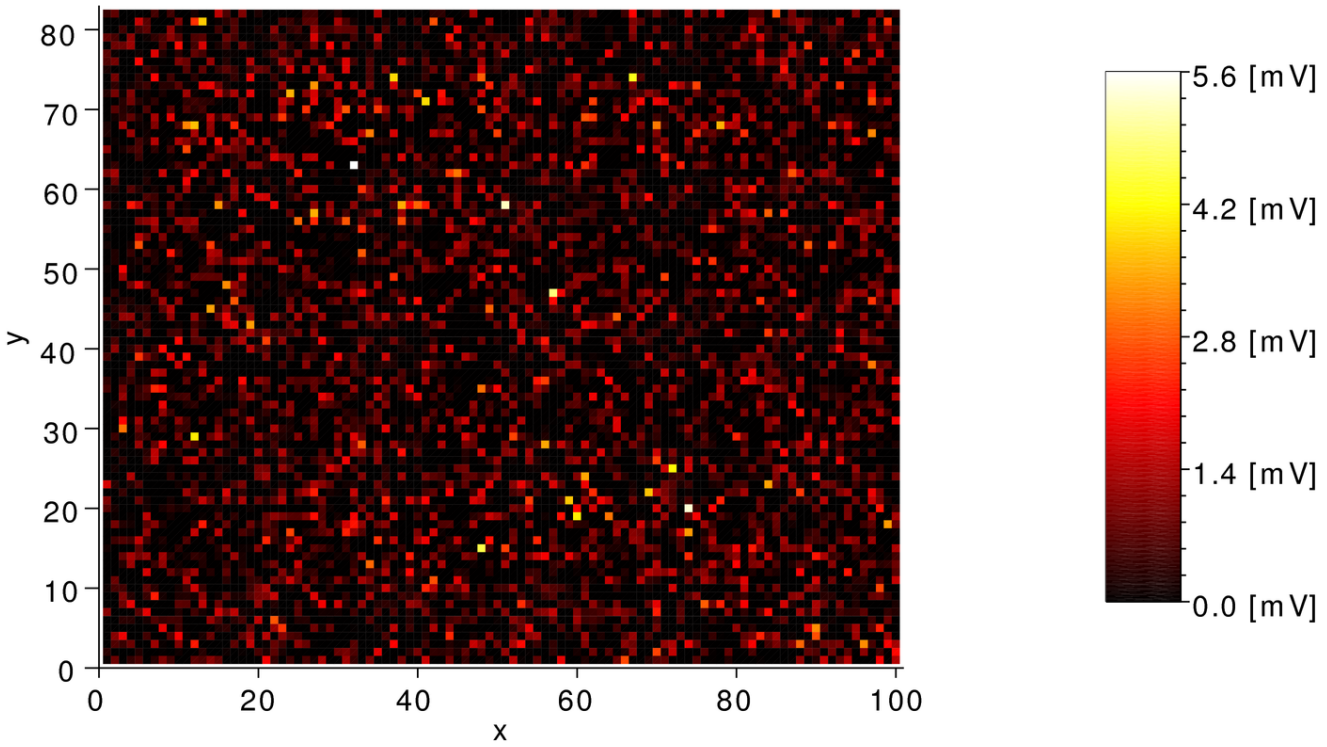
  - Uniform operation sequence.
  - $\sim 4\%$ computation overhead.
  - No hardware overhead (T re-used).

Our Countermeasure

For López-Dahab Montgomery Ladder

Fraunhofer AISEC

- Difference-of-means analysis when using countermeasure.



- Random appearance.
- No significant regions.
- Small amplitudes.

Outline

Conclusion

Fraunhofer AISEC

- Proved working hypothesis.
  - Cryptographic designs leak location-based information.
  - Exploitable for side-channel-attacks.
- Prevent location-based information leakage.
  - Repeatedly randomize assignment of algorithm variables to physical locations throughout cryptographic algorithm.

Towards Different Flavors of Combined Side Channel Attacks.

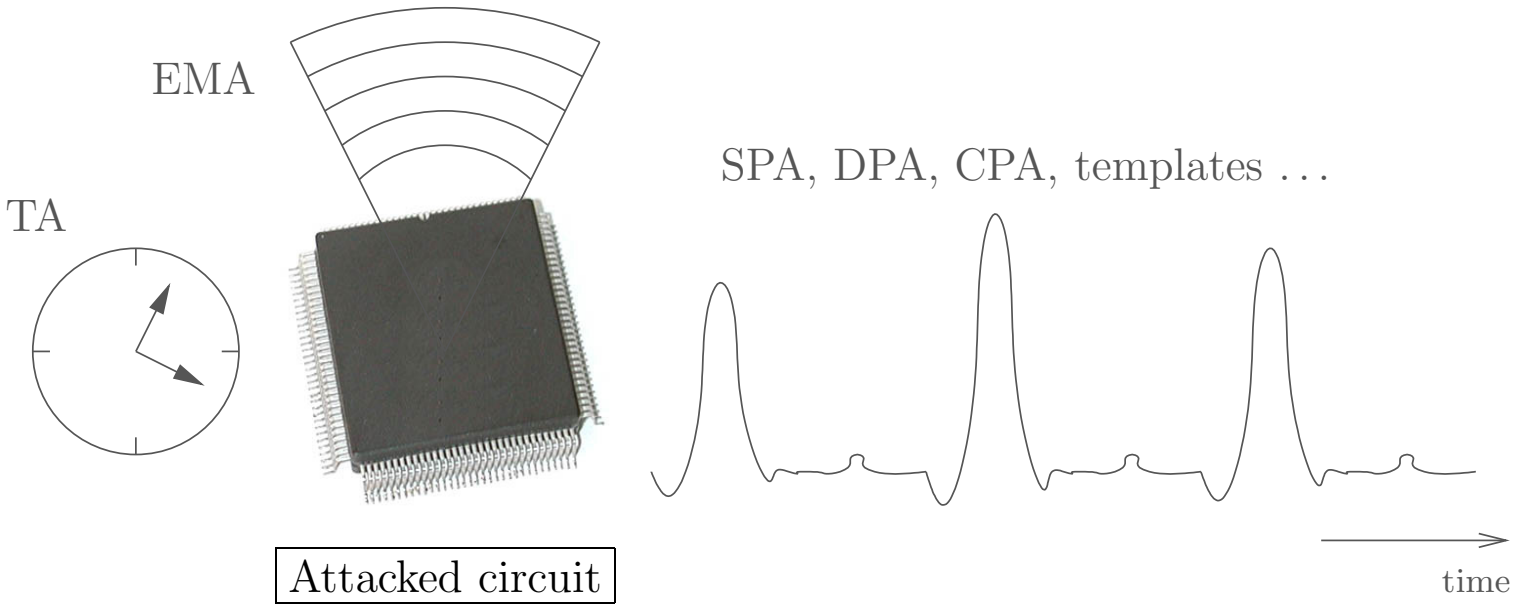
Shivam Bhasin Youssef Souissi Sylvain Guilley
Maxime Nassar Jean-Luc Danger
<shivam.bhasin@TELECOM-ParisTech.fr>

Thursday, March 1st, 2012

Presentation Outline

Presentation Outline

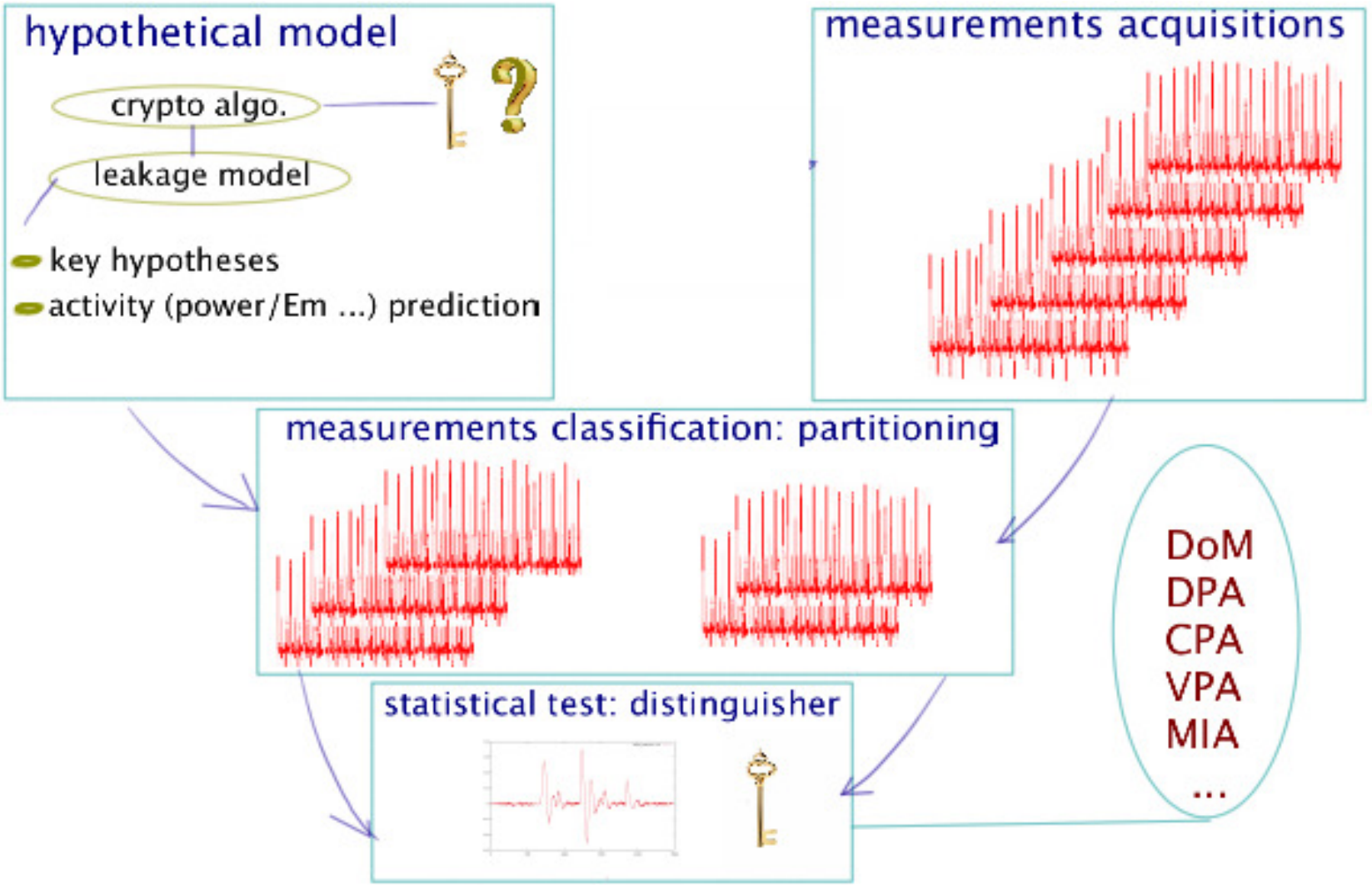
Side-Channel Attacks (SCA)



Different Types of SCA

- Timing Attacks.
- Power Analysis Attacks.
- Electromagnetic Attacks.

SCA: Basic Algorithm



Motivations

- Countermeasures make measurements a scarce resource.
- There is a need for accelerating SCA.

Motivations

- Countermeasures make measurements a scarce resource.
- There is a need for accelerating SCA.

How to Accelerate SCA?

Our Idea:

- The right key in different attacks is always ranked higher.
- But the false key candidates differ from one attack to another.
- **Combining different attack** will negate the wrong key candidates faster and **accelerate** the attacks.

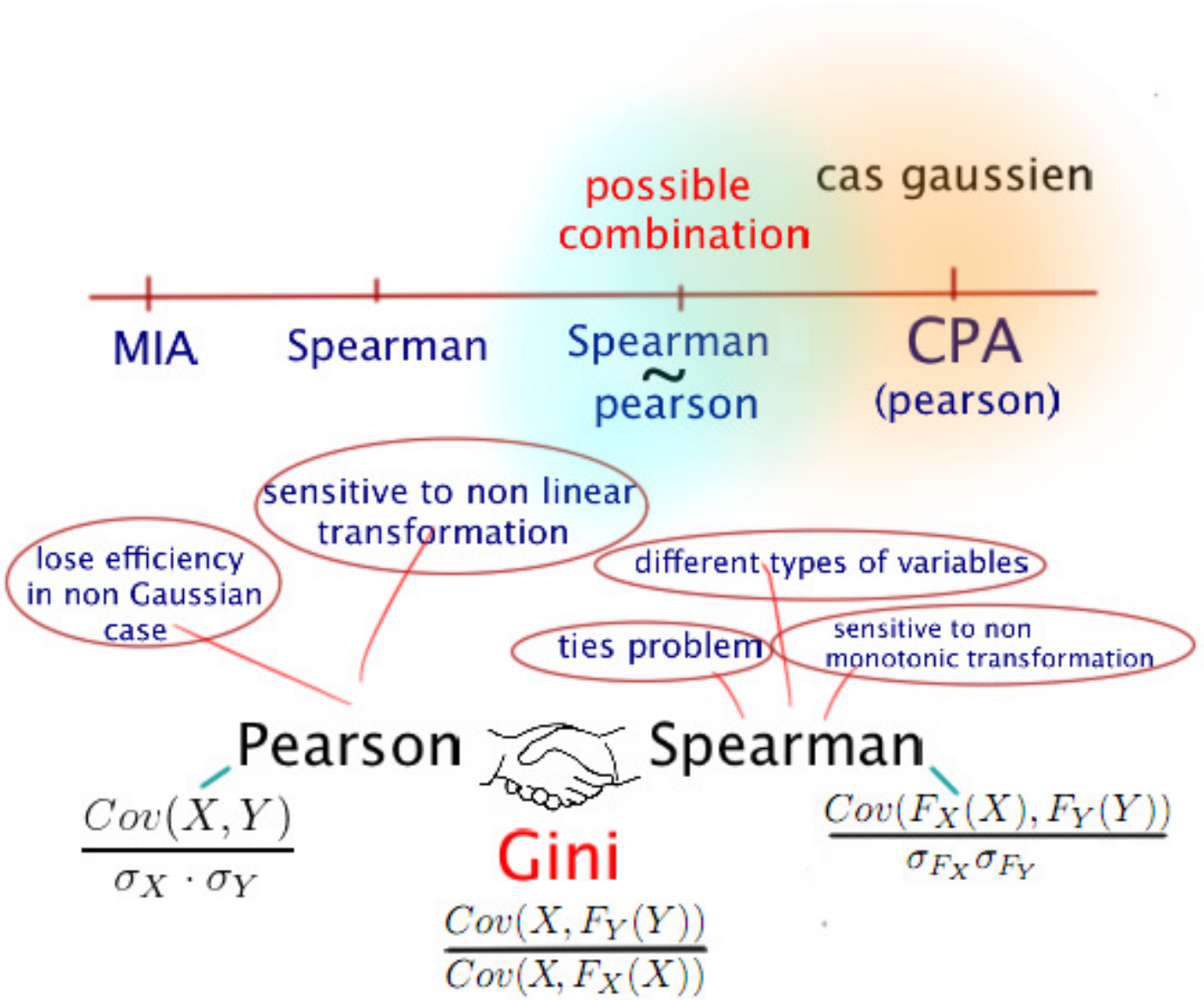
Combined Side-Channel Attacks

Two Approaches

1. Combination of Distinguishers.
2. Combination of Measurements.

Presentation Outline

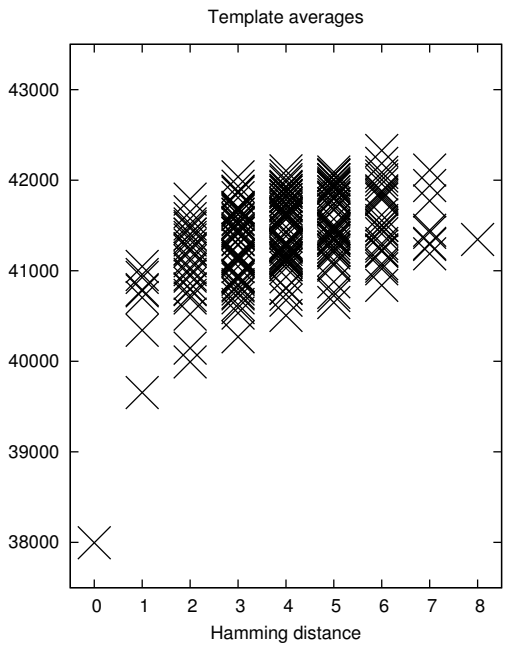
Combining SCA Distinguishers

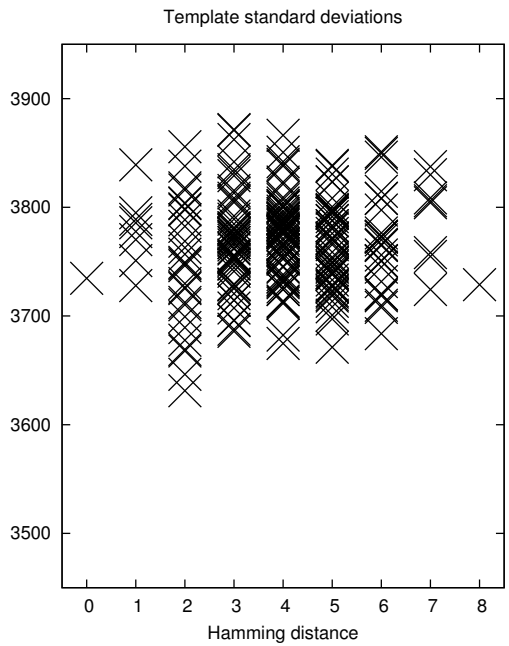


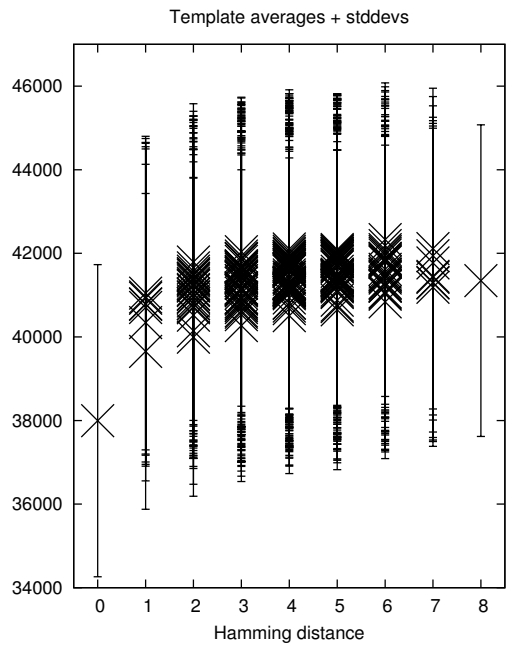
Comparing Distinguishers

$$\mathscr{L}(x) = \mathrm{HW}(x) + \alpha \cdot \delta(x)$$

where Kronecker symbol $\delta(x) = 1$ when $x = 0$ else 1.

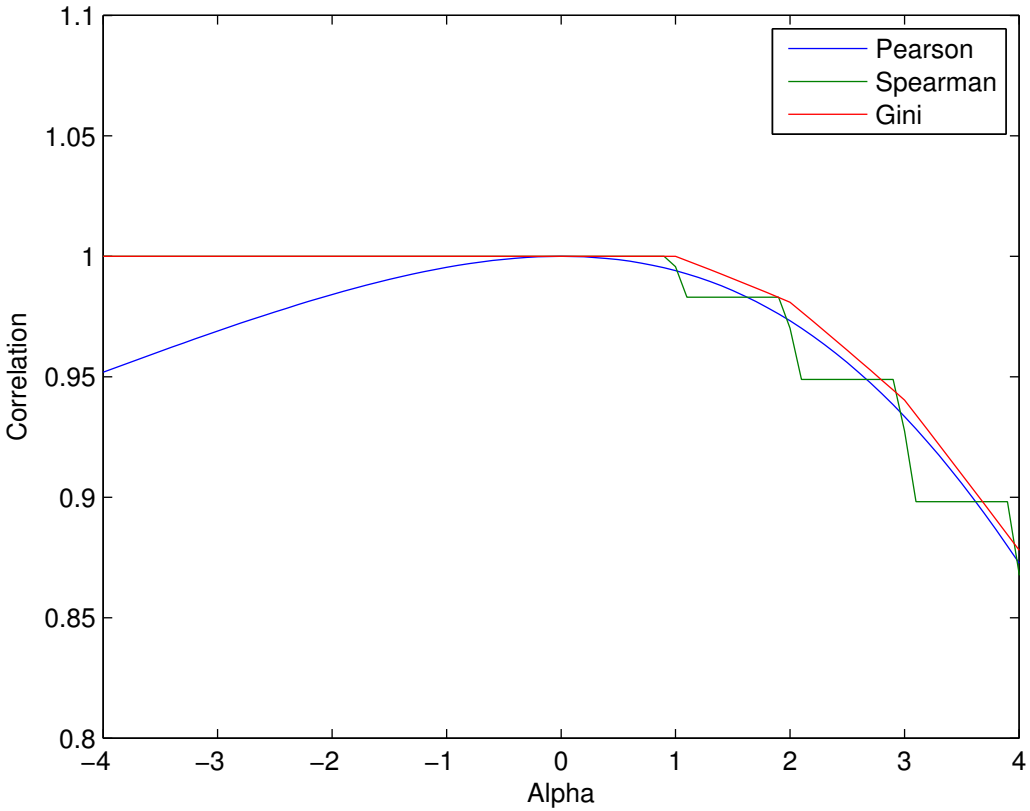






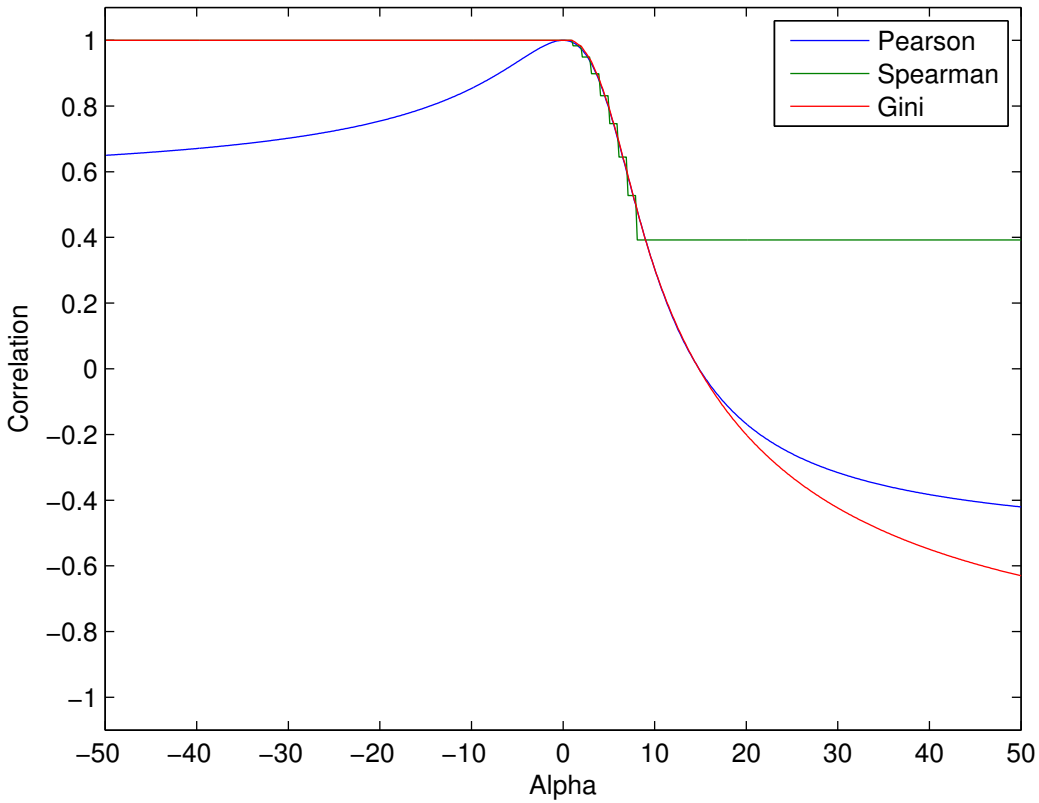
Comparing Distinguishers

$$\mathscr{L}(x) = \mathrm{HW}(x) + \alpha \cdot \delta(x)$$



Comparing Distinguishers

$$\mathscr{L}(x) = \mathrm{HW}(x) + \alpha \cdot \delta(x)$$



Combined SCA Distinguishers: Empirical Combination

Observations for empirical combination

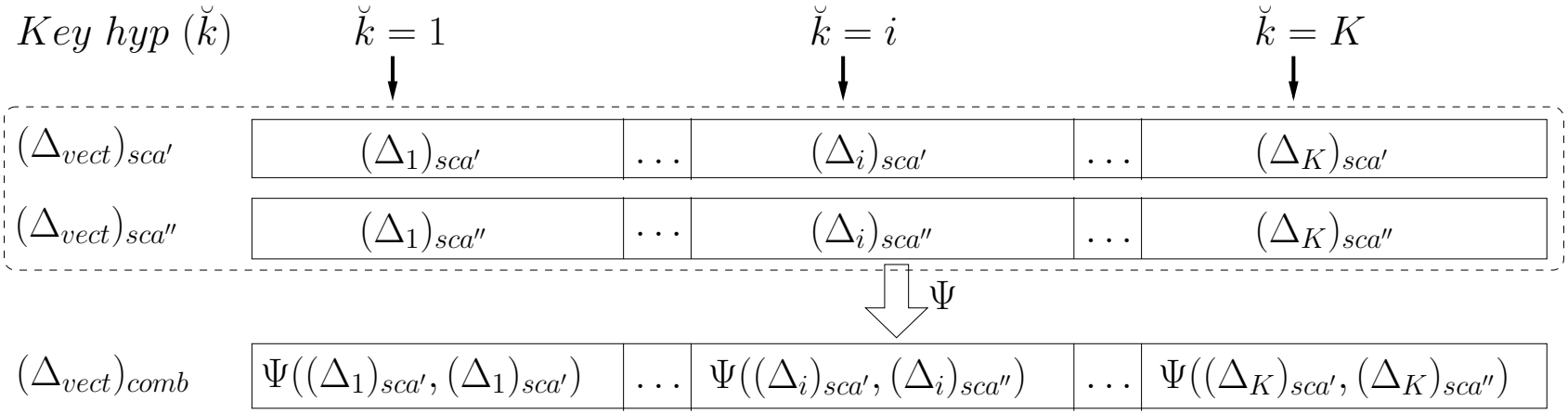
Both distinguishers have:

- Similar evolution in term of evaluation metrics,
- Same temporal positions for secret key unlike false keys,
- Not the same predicted key for each iteration,
- Secret key always ranked among the first ranks.

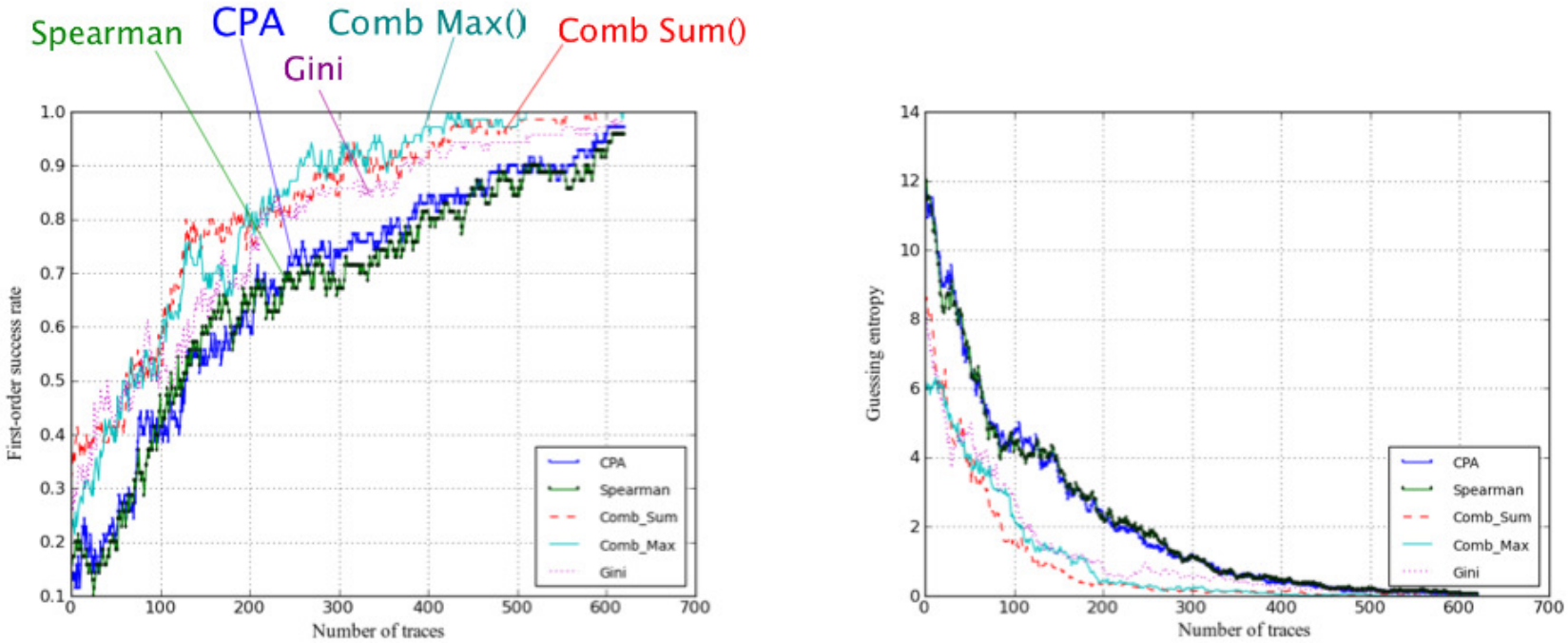
Empirical combination process

- Both attacks should be performed in parallel.
- Apply, in real time (*i.e.* for each iteration), an aggregate function (*e.g* the Max() or the Sum()) on the values returned by CPA and Spearman distinguishers, respectively.

Aggregate Function



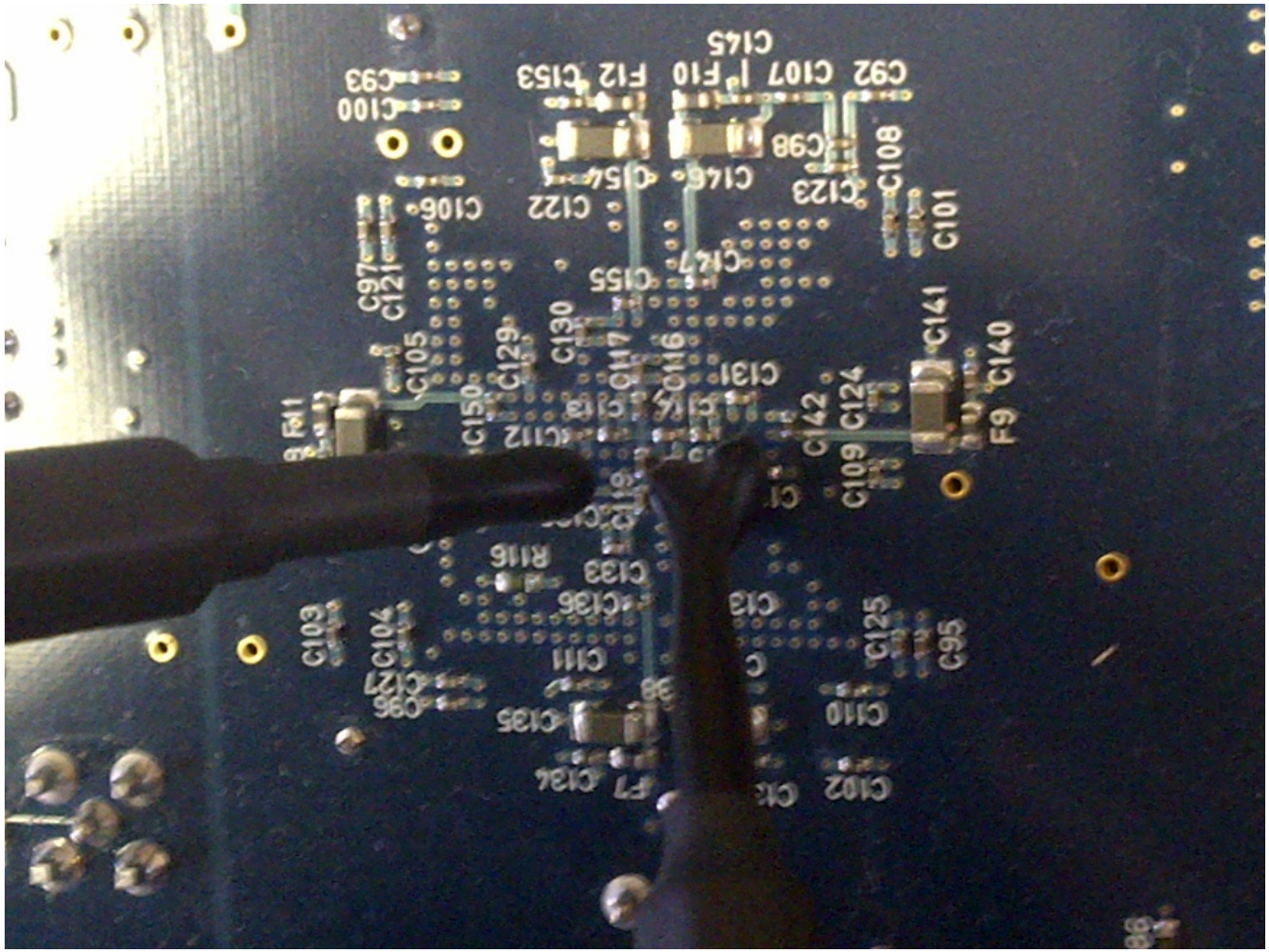
Combined SCA Distinguishers: Results



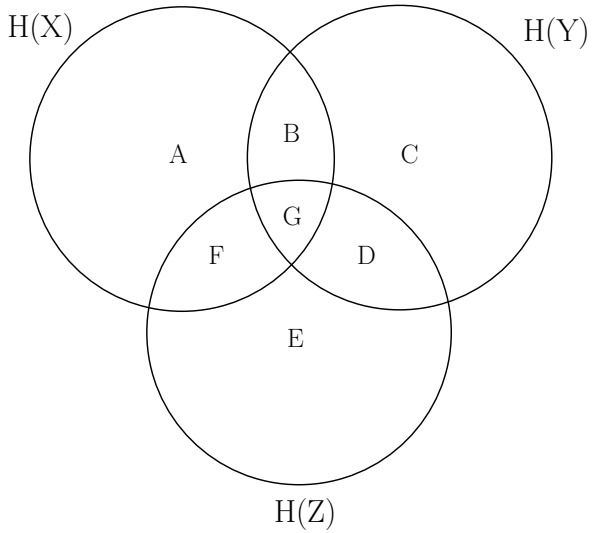
SR and GE of Combinations based CPA *vs* basic CPA (unprotected DES).

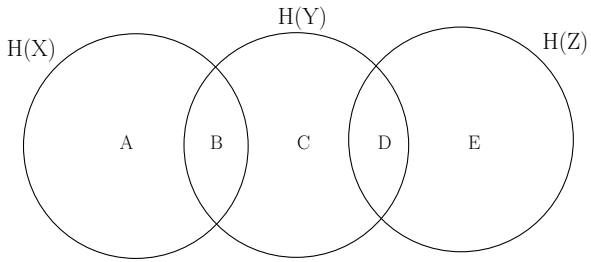
Presentation Outline

Combination of Measurements



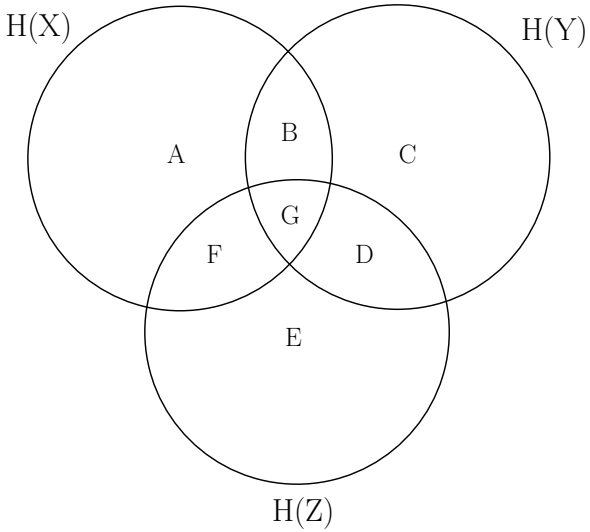
Theoretical Background

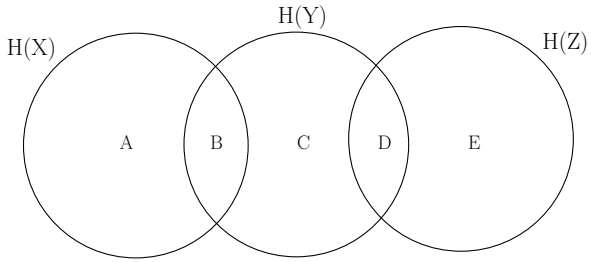




$$I(X;Y;Z) = I(X,Y;Z) - I(X;Z) - I(Y;Z)$$
$$I(X;Y;Z) = (D+F+G) - (F+G) - (D+G) = \textbf{-G}$$

Theoretical Background





I(X,Y;Z)= INTERACTION GAIN

$$I(X;Y;Z) = I(X,Y;Z) - I(X;Z) - I(Y;Z)$$
$$I(X;Y;Z) = (D+F+G) - (F+G) - (D+G) = \text{-}\mathbf{G}$$

Experimental Demonstration

Setup Phase

- Leakage points are chosen by cartography or trial-and-error method.
- Two traces corresponding to the same encryption are recorded using EM probes.

Attack Phase

- Traces from the 2 probes are concatenated.
- Normalization of traces may be required.
- CPA is launched on the concatenated trace.
- The co-efficient of the two section of traces are combined using aggregate function.

Experimental Results on DES

S-box No.	0	1	2	3	4	5	6	7
C_1	350	943	733	400	410	320	548	592
C_2	432	1073	720	980	176	281	551	192
Comb_sum	212	750	397	251	165	270	448	184
Percent Gain	39.42	20.46	44.86	37.25	6.25	3.96	18.24	4.16

Average result of 30 CPA

Presentation Outline

Conclusions & Perspectives

Conclusions

- Proposed two new methodologies of combined attacks.
- Gini is a theoretical combination Pearson and Spearman.
- Aggregate function like Sum and Max can be used to combine distinguishers and measurements.
- Observed up to 50% gain in terms of number of traces.

Perspectives

- Application of these methodologies to profiled SCA.
- Combining sub-processes in parallel execution of an algorithm.

Thank you for your attention

Towards Different Flavors of Combined Side Channel Attacks.

Shivam Bhasin Youssef Souissi Sylvain Guilley
Maxime Nassar Jean-Luc Danger
<shivam.bhasin@TELECOM-ParisTech.fr>

Thursday, March 1st, 2012

Towards Different Flavors of Combined Side Channel Attacks.

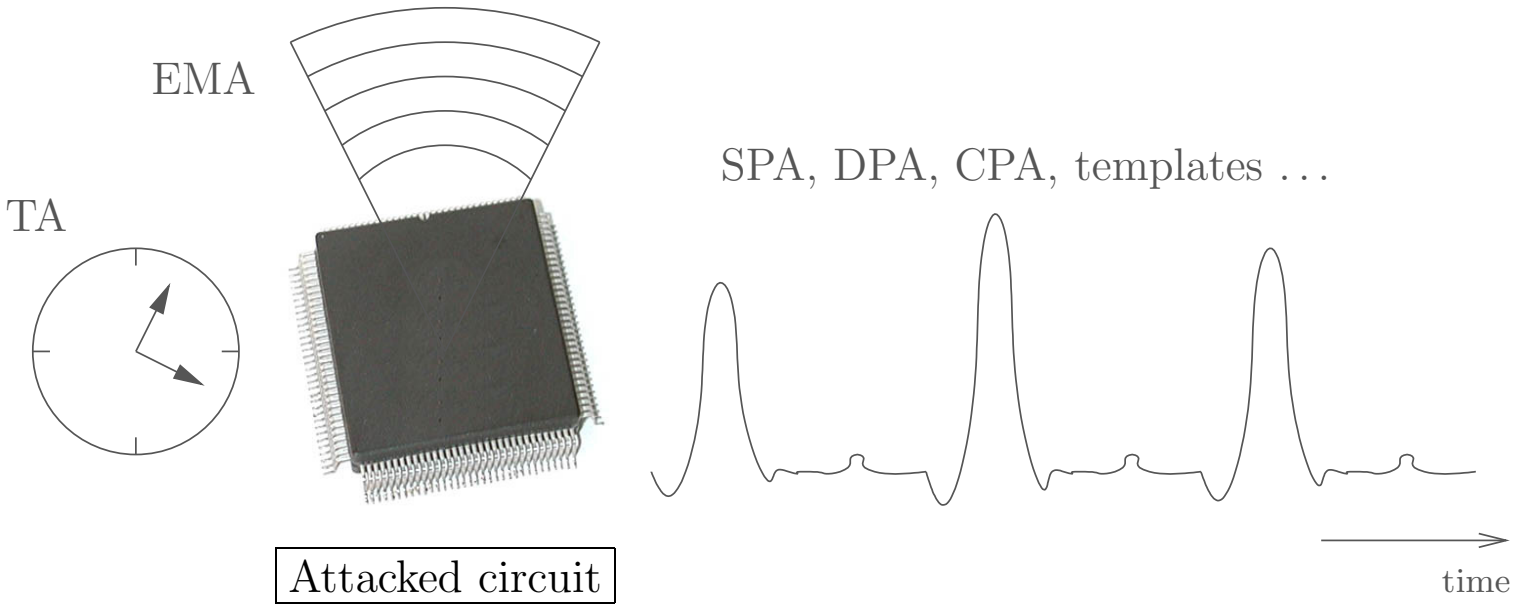
Shivam Bhasin Youssef Souissi Sylvain Guilley
Maxime Nassar Jean-Luc Danger
<shivam.bhasin@TELECOM-ParisTech.fr>

Thursday, March 1st, 2012

Presentation Outline

Presentation Outline

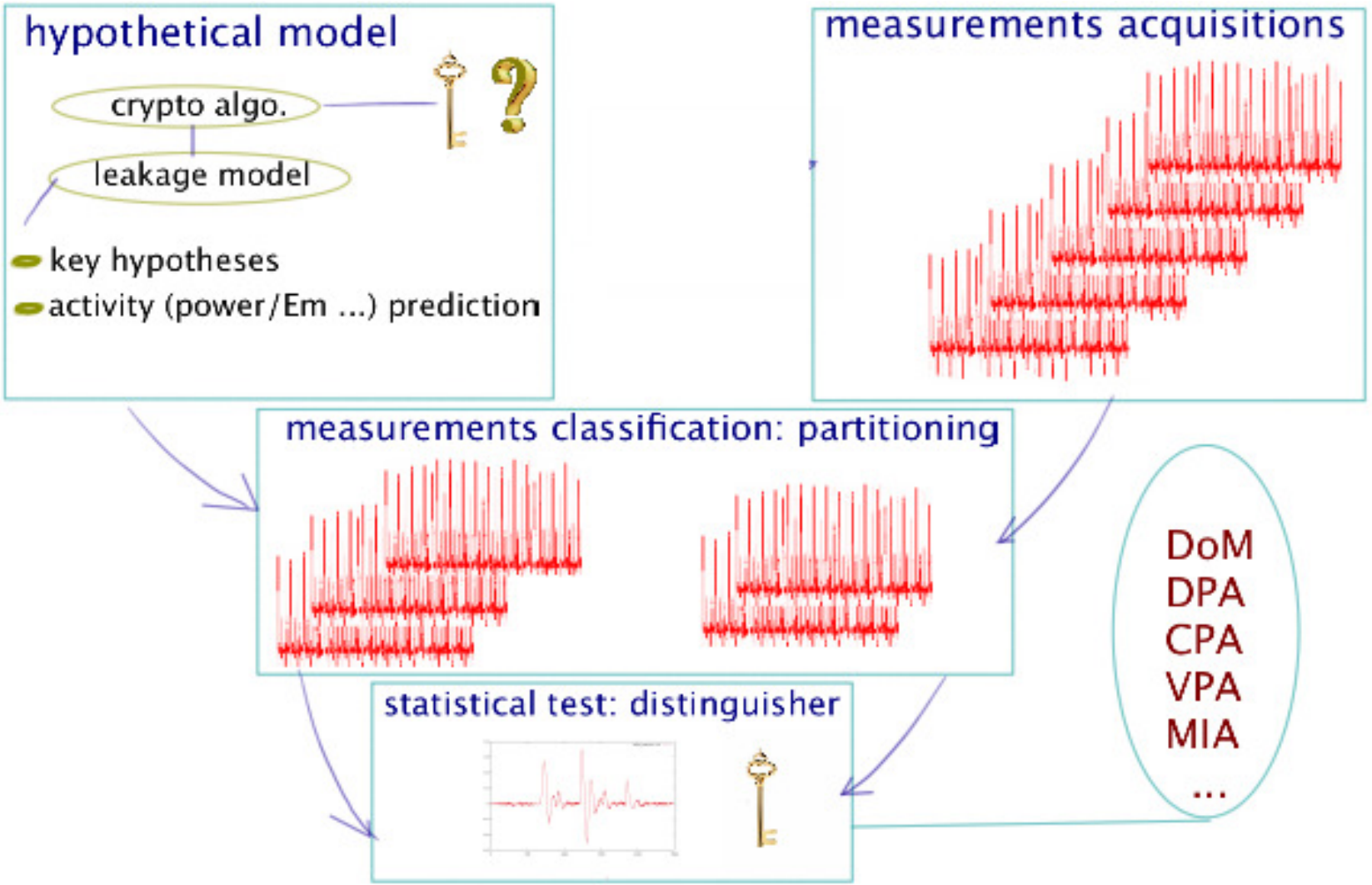
Side-Channel Attacks (SCA)



Different Types of SCA

- Timing Attacks.
- Power Analysis Attacks.
- Electromagnetic Attacks.

SCA: Basic Algorithm



Motivations

- Countermeasures make measurements a scarce resource.
- There is a need for accelerating SCA.

Motivations

- Countermeasures make measurements a scarce resource.
- There is a need for accelerating SCA.

How to Accelerate SCA?

Our Idea:

- The right key in different attacks is always ranked higher.
- But the false key candidates differ from one attack to another.
- **Combining different attack** will negate the wrong key candidates faster and **accelerate** the attacks.

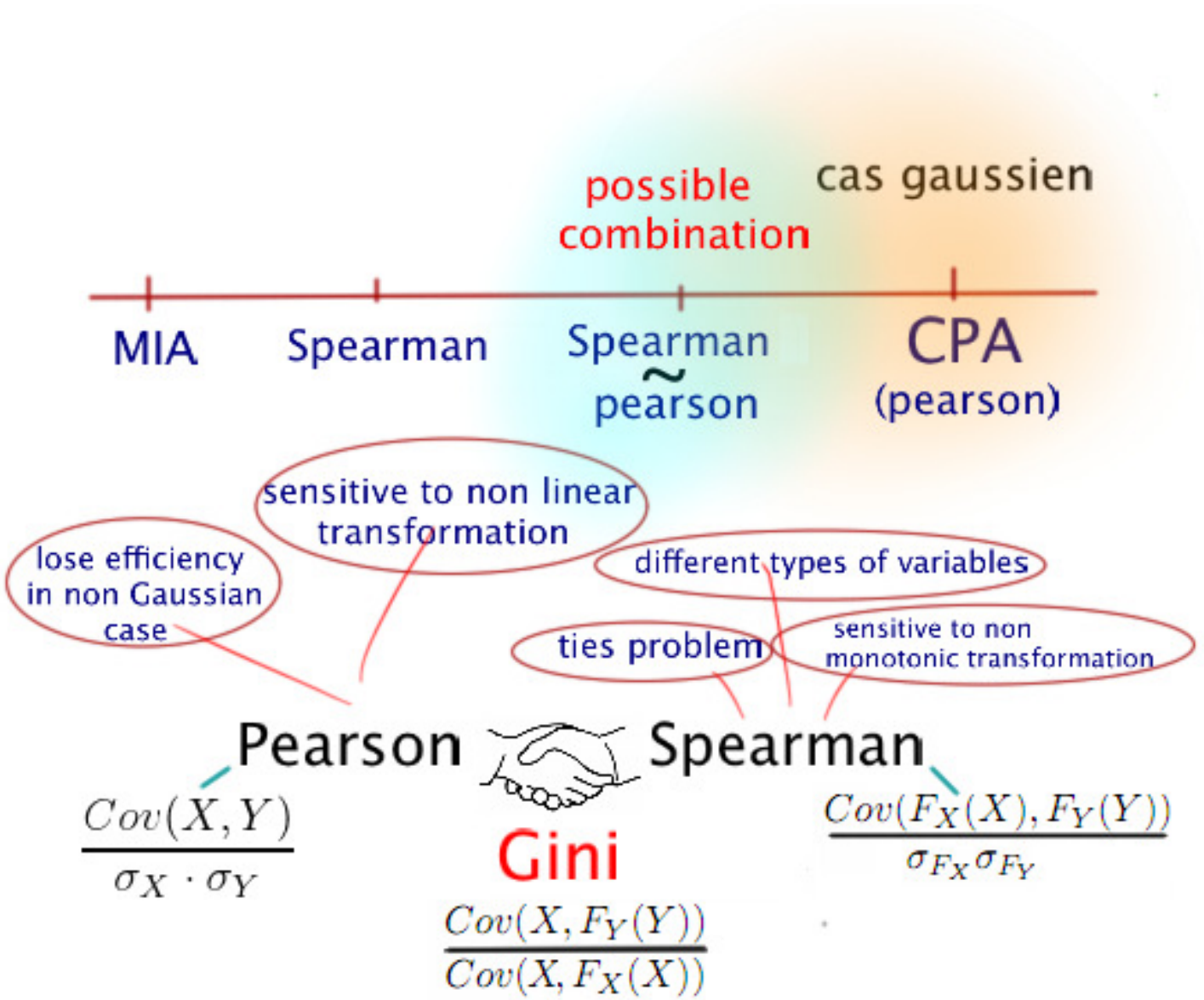
Combined Side-Channel Attacks

Two Approaches

1. Combination of Distinguishers.
2. Combination of Measurements.

Presentation Outline

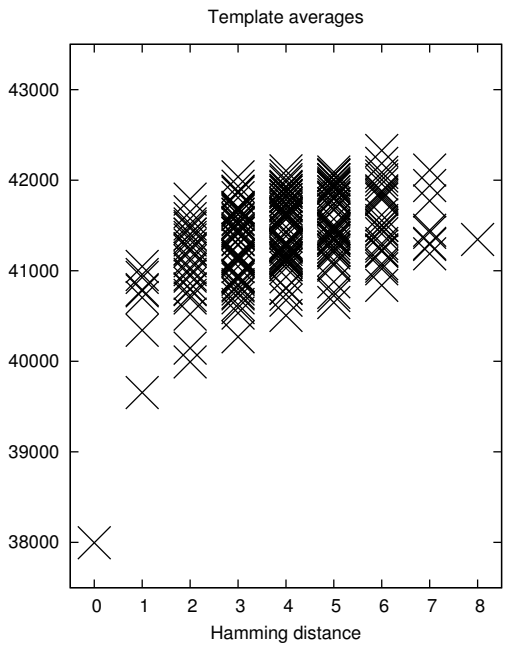
Combining SCA Distinguishers

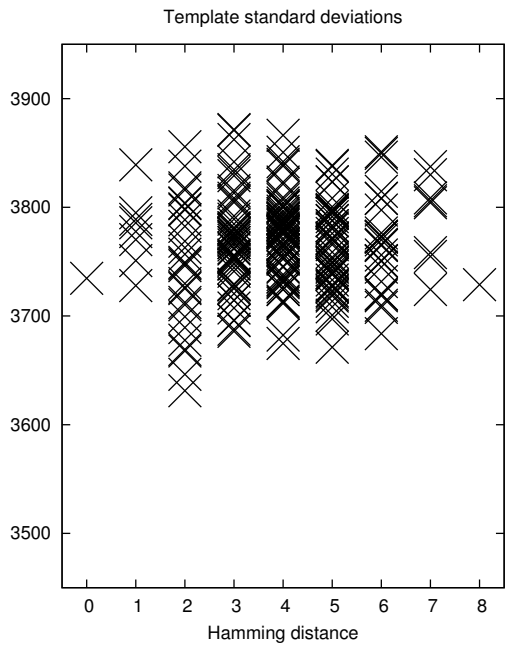


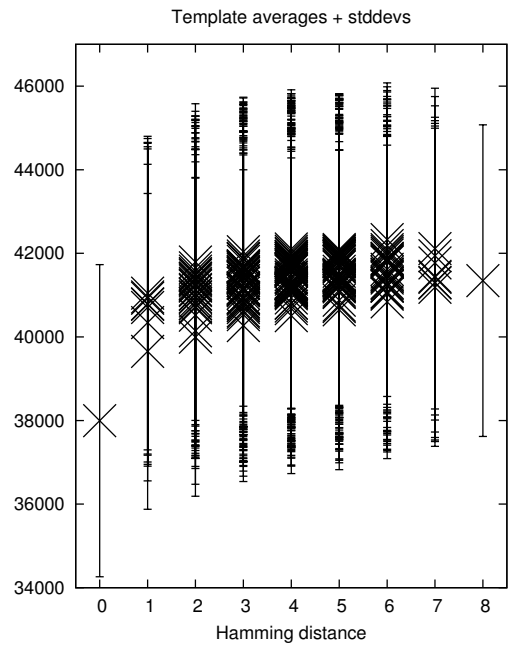
Comparing Distinguishers

$$\mathscr{L}(x) = \mathrm{HW}(x) + \alpha \cdot \delta(x)$$

where Kronecker symbol $\delta(x) = 1$ when $x = 0$ else 0.

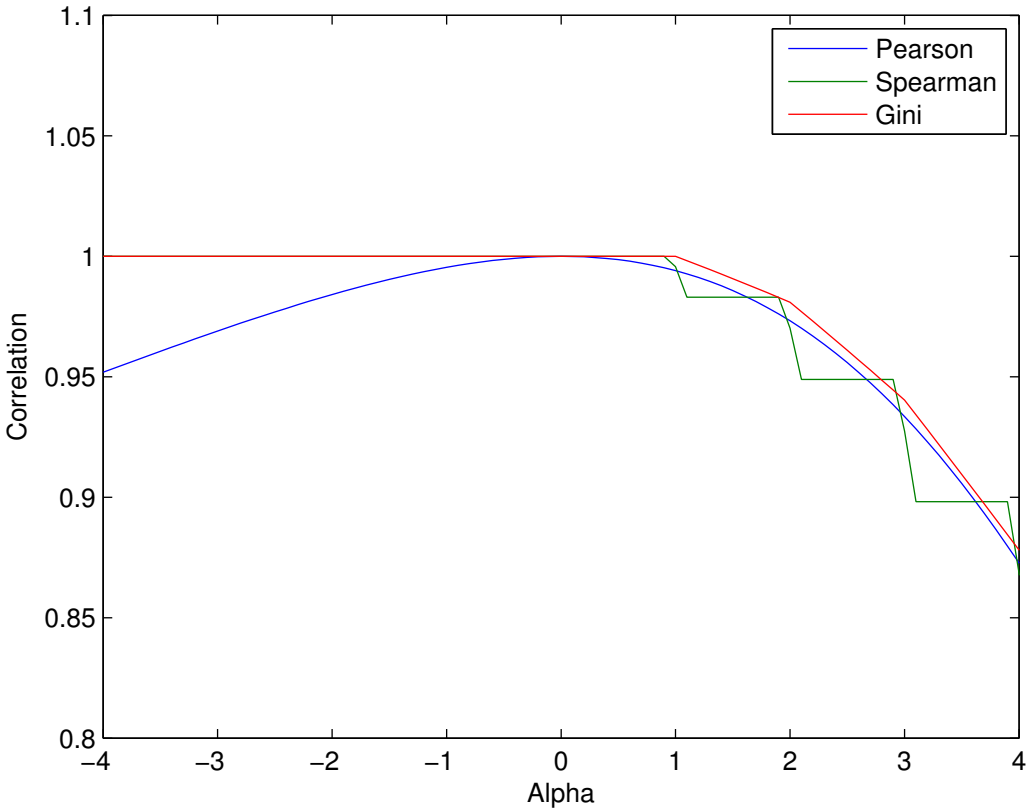






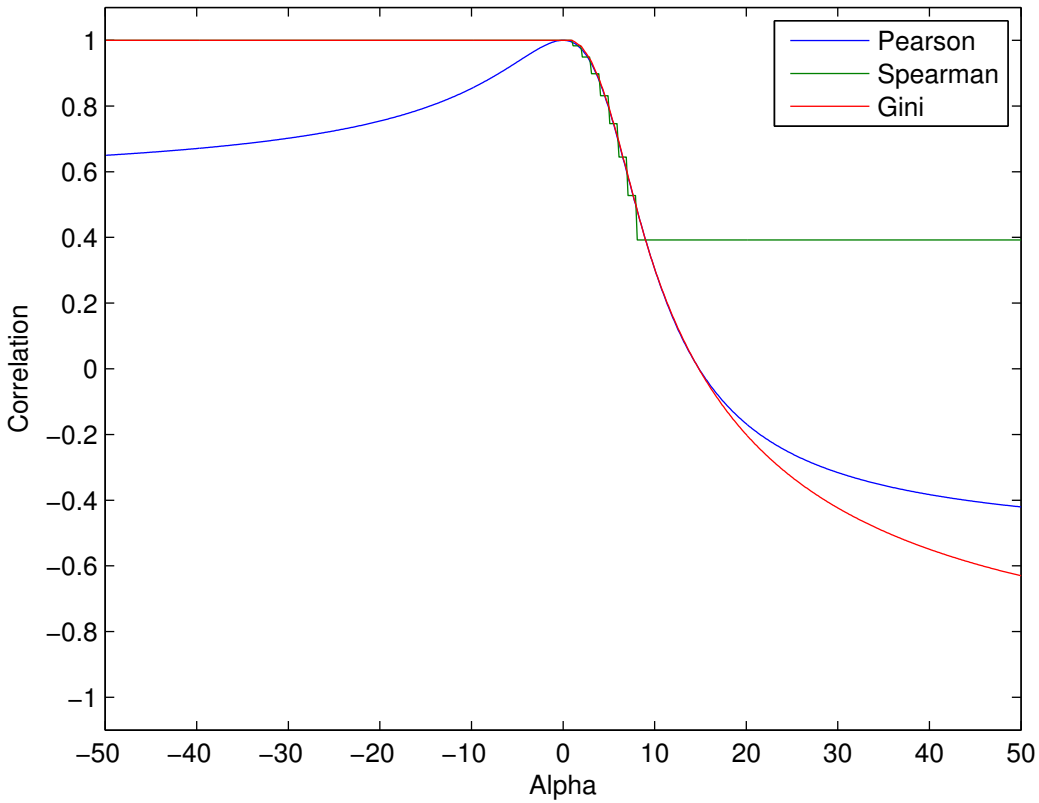
Comparing Distinguishers

$$\mathscr{L}(x) = \mathrm{HW}(x) + \alpha \cdot \delta(x)$$



Comparing Distinguishers

$$\mathscr{L}(x) = \mathrm{HW}(x) + \alpha \cdot \delta(x)$$



Combined SCA Distinguishers: Empirical Combination

Observations for empirical combination

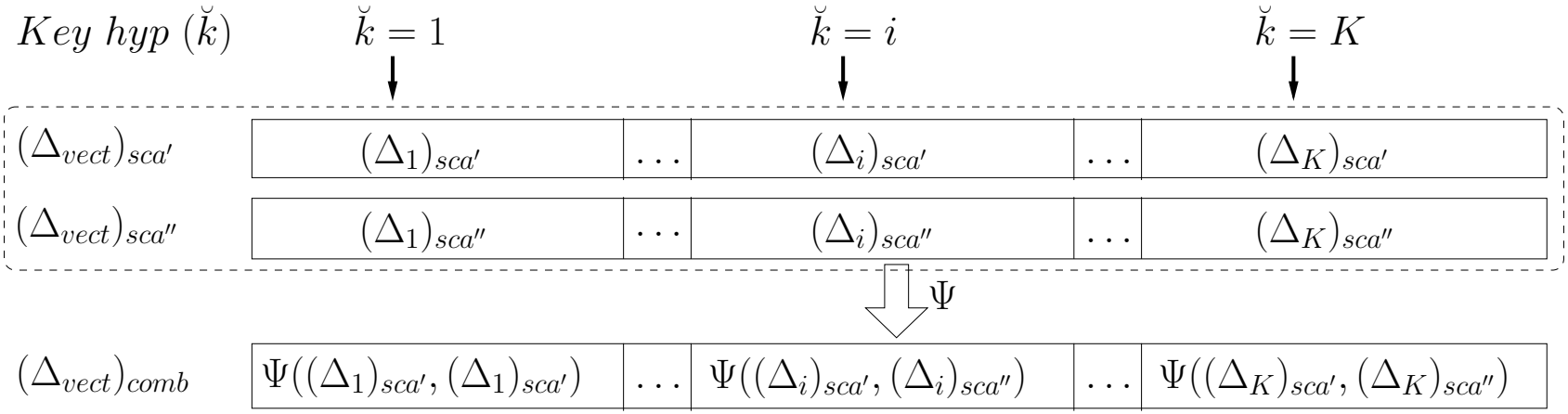
Both distinguishers have:

- Similar evolution in term of evaluation metrics,
- Same temporal positions for secret key unlike false keys,
- Not the same predicted key for each iteration,
- Secret key always ranked among the first ranks.

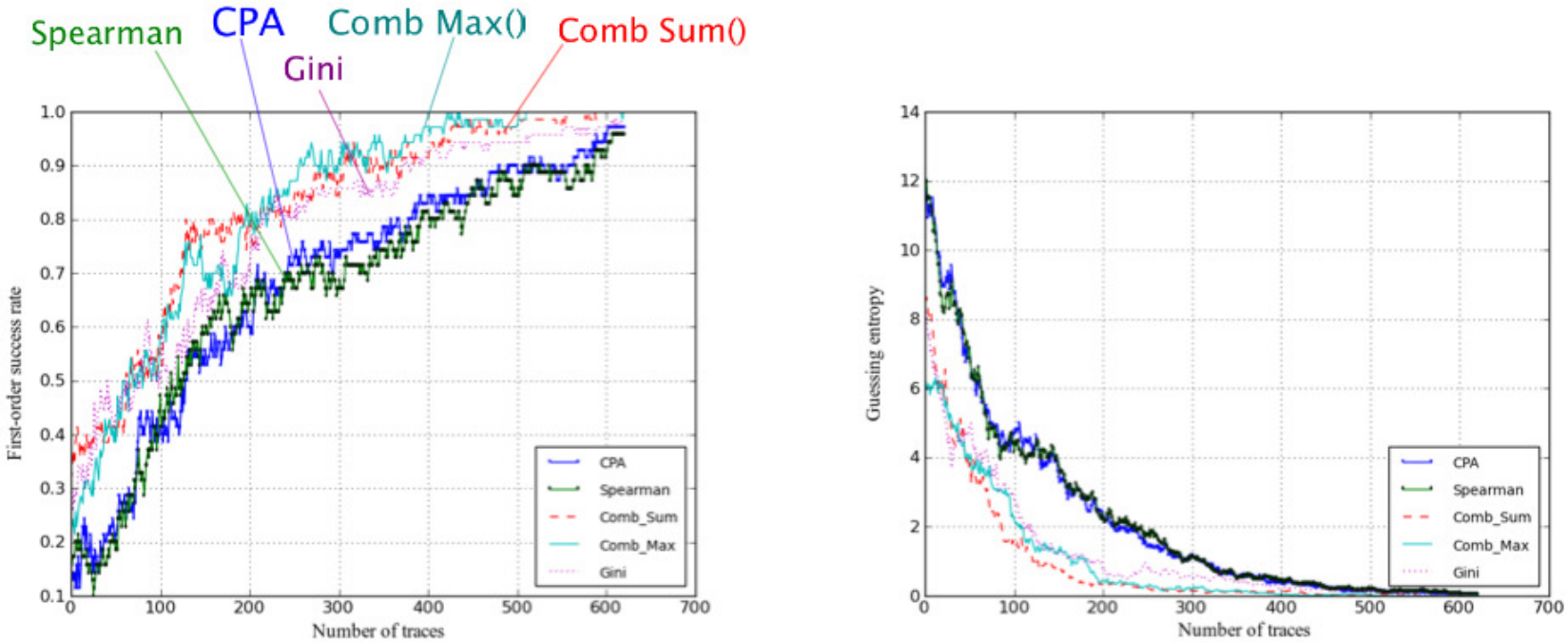
Empirical combination process

- Both attacks should be performed in parallel.
- Apply, in real time (*i.e.* for each iteration), an aggregate function (*e.g* the Max() or the Sum()) on the values returned by CPA and Spearman distinguishers, respectively.

Aggregate Function



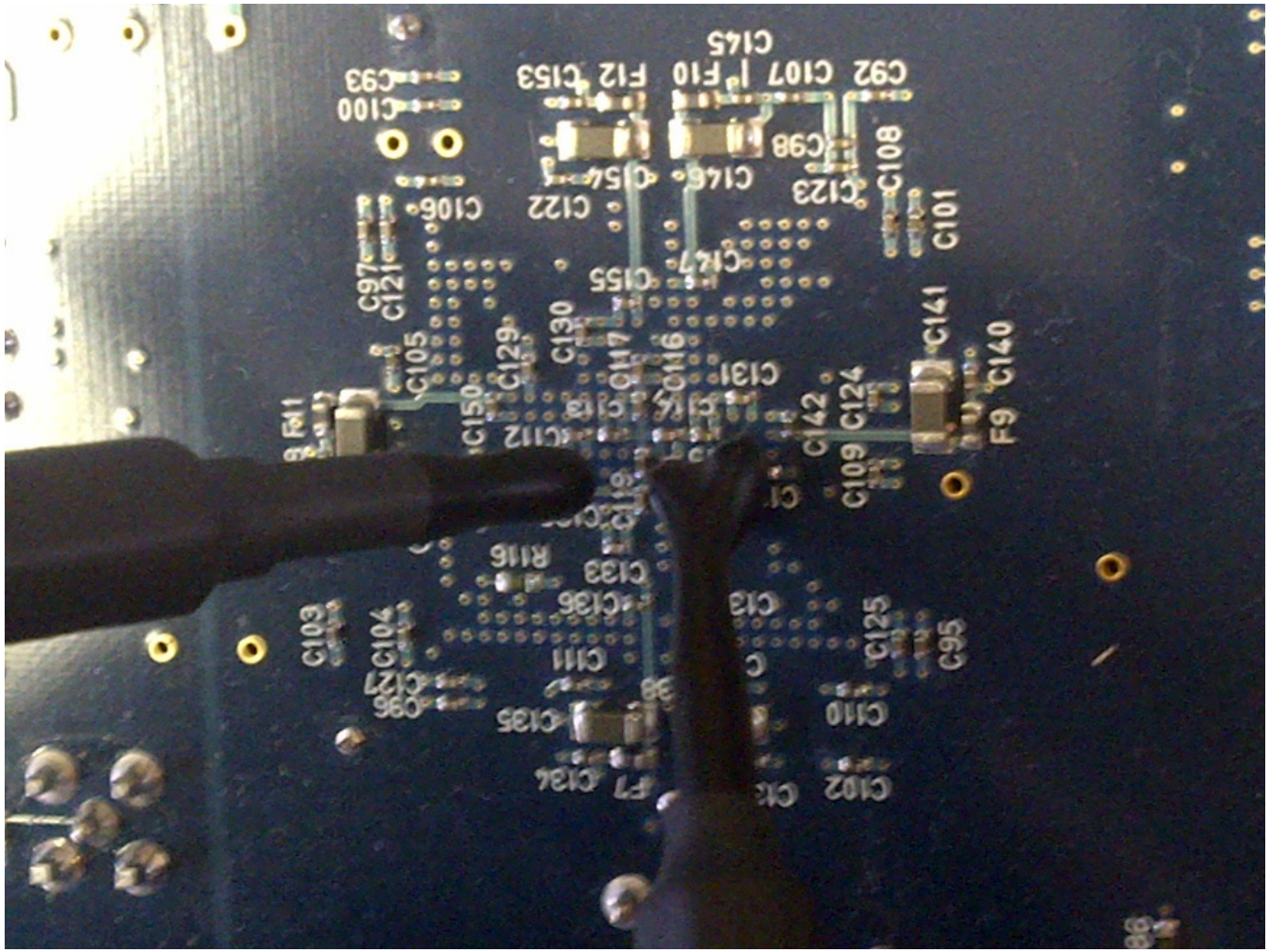
Combined SCA Distinguishers: Results



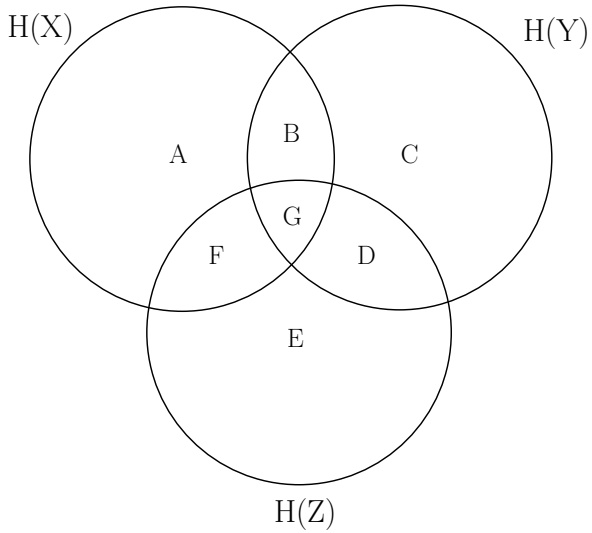
SR and GE of Combinations based CPA *vs* basic CPA (unprotected DES).

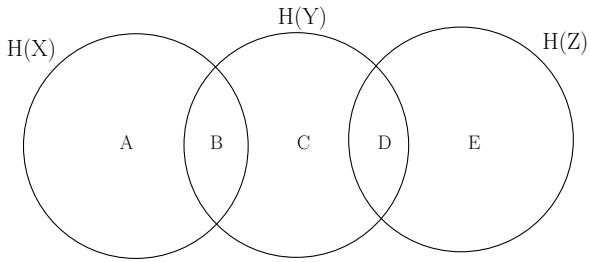
Presentation Outline

Combination of Measurements



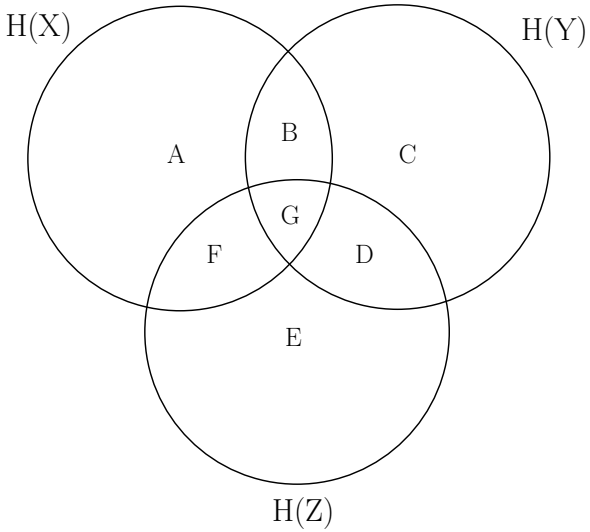
Theoretical Background

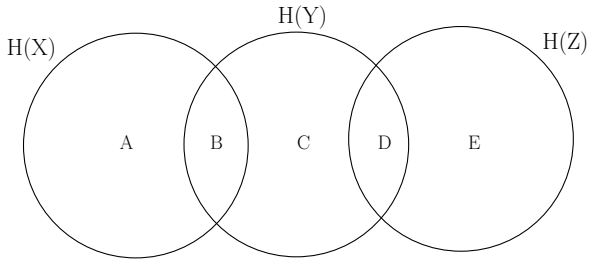




$$I(X;Y;Z) = I(X,Y;Z) - I(X;Z) - I(Y;Z)$$
$$I(X;Y;Z) = (D+F+G) - (F+G) - (D+G) = \textbf{-G}$$

Theoretical Background





I(X,Y;Z)= INTERACTION GAIN

$$I(X;Y;Z) = I(X,Y;Z) - I(X;Z) - I(Y;Z)$$
$$I(X;Y;Z) = (D+F+G) - (F+G) - (D+G) = \text{-}\mathbf{G}$$

Experimental Demonstration

Setup Phase

- Leakage points are chosen by cartography or trial-and-error method.
- Two traces corresponding to the same encryption are recorded using EM probes.

Attack Phase

- Traces from the 2 probes are concatenated.
- Normalization of traces may be required.
- CPA is launched on the concatenated trace.
- The co-efficient of the two section of traces are combined using aggregate function.

Experimental Results on DES

S-box No.	0	1	2	3	4	5	6	7
C_1	350	943	733	400	410	320	548	592
C_2	432	1073	720	980	176	281	551	192
Comb_sum	212	750	397	251	165	270	448	184
Percent Gain	39.42	20.46	44.86	37.25	6.25	3.96	18.24	4.16

Average result of 30 CPA

Presentation Outline

Conclusions & Perspectives

Conclusions

- Proposed two new methodologies of combined attacks.
- Gini is a theoretical combination Pearson and Spearman.
- Aggregate function like Sum and Max can be used to combine distinguishers and measurements.
- Observed up to 50% gain in terms of number of traces.

Perspectives

- Application of these methodologies to profiled SCA.
- Combining sub-processes in parallel execution of an algorithm.

Thank you for your attention

Towards Different Flavors of Combined Side Channel Attacks.

Shivam Bhasin Youssef Souissi Sylvain Guilley
Maxime Nassar Jean-Luc Danger
<shivam.bhasin@TELECOM-ParisTech.fr>

Thursday, March 1st, 2012