Improving Non-Profiled Attacks on Exponentiations Based on Clustering and Extracting Leakage from Multi-Channel High-Resolution EM Measurements

Robert Specht Johann Heyszl Martin Kleinsteuber[a] Georg Sigl[a] ,

[a]Technische Universität München, Munich, Germany

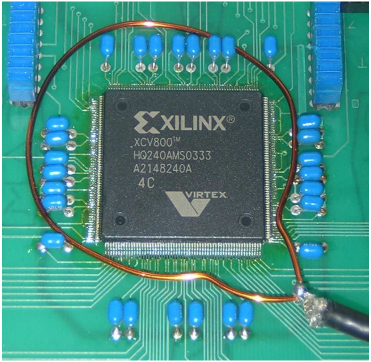
Fraunhofer AISEC

Motivation

- Asymmetric ciphers (e.g. ECC)
- Attackers only have single trace
- Profiling is often prevented

- How could attackers still exploit leakage in the best way?
- Will multiple probes help attackers?

Different Coils for EMA

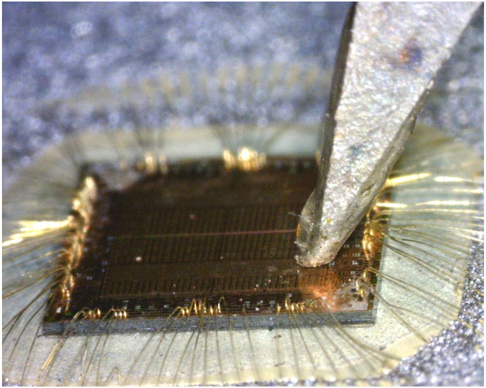


Source: De Mulder et al. [a]

- Simple EM measurements are roughly as good as current measurements

[a] De Mulder, E.; Örs, S. B.; Preneel, B. & Verbauwhede, I. Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems Comput. Electr. Eng., Pergamon Press, Inc., 2007, 33, 367-382

Different Coils for EMA



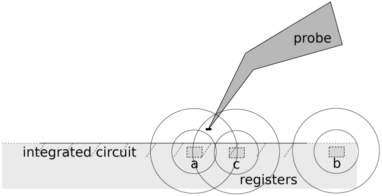
- Tiny coils
  - Closer to circuit parts → Better SNR
  - Also: Location-dependent leakage of asymmetric crypto

Exponentiations in Asymmetric Ciphers
Heyszl et al. 2012

- (Previous work)
- Typical algorithm structure in asymmetric crypto:

Input: Secret $d = d_N d_{N-1} \dots d_2 d_1$ with $d_i \in \{0, 1\}$

for $i = N$ downto 1 **do**
    if $d_i = 1$ **then**
        $c \leftarrow c^2 + a$
        $a \leftarrow c$
    else
        $c \leftarrow c^2 + b$
        $b \leftarrow c$
    end if
end for



- Iteration based algorithm: 1 Iteration = 1 Bit
- Similarities for the two values of d_i is what attackers may exploit

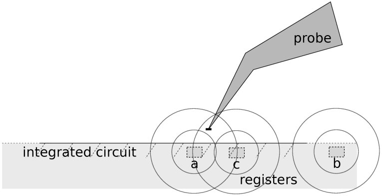
- Registers are spread over die (registers hold multiple bytes)
- Location-based information leakage from high-precision probe

Exponentiations in Asymmetric Ciphers
Heyszl et al. 2012

- (Previous work)
- Typical algorithm structure in asymmetric crypto:

Input: Secret $d = d_N d_{N-1} ... d_2 d_1$ with $d_i \in \{0, 1\}$

for $i = N$ downto 1 **do**
    if $d_i = 1$ **then**
        $c \leftarrow c^2 + a$
        $a \leftarrow c$
    else
        $c \leftarrow c^2 + b$
        $b \leftarrow c$
    end if
end for



- Iteration based algorithm: 1 Iteration = 1 Bit
- Similarities for the two values of d_i is what attackers may exploit
- Registers are spread over die (registers hold multiple bytes)
- Location-based information leakage from high-precision probe

Our Practical Investigation

- ECC (Elliptic Curve Cryptography) engine on FPGA
- Measurement setup with three probes on die
- No profiling for good positions
- Repeat measurements on 400 positions → 400 tests

1. Analyse measurements of probes separately - Improve algorithms
2. Compare single to combined probes outcome - Evaluate advantage

Our Practical Investigation

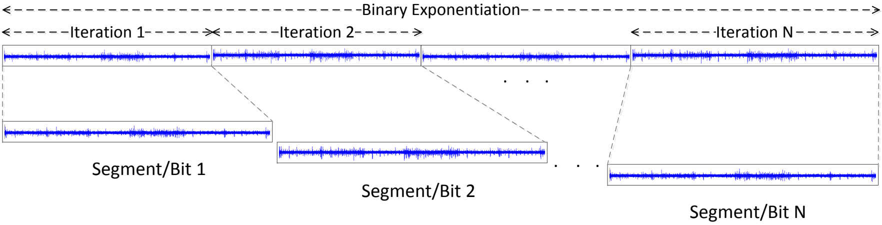
- ECC (Elliptic Curve Cryptography) engine on FPGA
- Measurement setup with three probes on die
- No profiling for good positions
- Repeat measurements on 400 positions $\rightarrow$ 400 tests

1. Analyse measurements of probes separately - Improve algorithms
2. Compare single to combined probes outcome - Evaluate advantage

Algorithmic Approach
Overview over Attack Analysis

- For each trace:
    1. Cut trace into segments corresponding to one bit
    2. Reduce amount of data
    3. Perform cluster classification
    4. Check how well the classification matches secret exponent

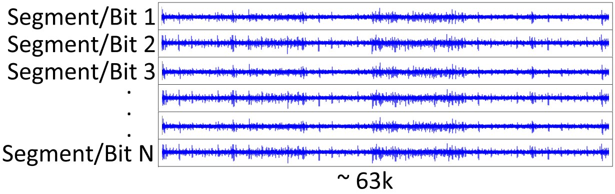
Algorithmic Approach
(1) Split Trace into Segments



- 1 loop iteration = 1 segment = 1 bit
- Split whole measurement trace into segments
- Rearrange to matrix

Fraunhofer AISEC

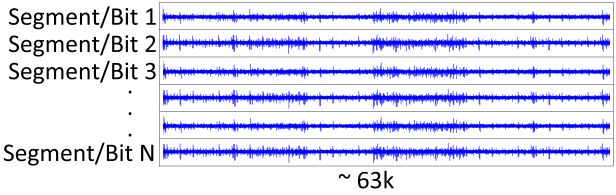
Algorithmic Approach
(2) Reduce Data Amount



- A lot of dimensions does not contain useful information → Reduce
  - Ideally, reduced to leakage and remove noise?
  - Earlier, simple trace compression techniques were used in this context
- Principal component analysis (PCA)

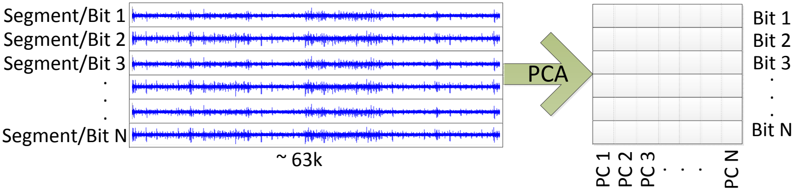
Fraunhofer AISEC

Algorithmic Approach
(2) Reduce Data Amount



- A lot of dimensions does not contain useful information → Reduce
  - Ideally, reduced to leakage and remove noise?
  - Earlier, simple trace compression techniques were used in this context
- Principal component analysis (PCA)

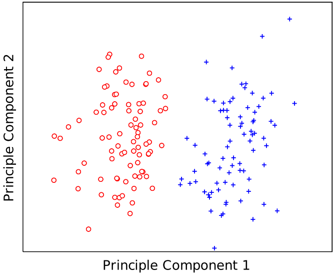
Algorithmic Approach
(2) Principal Component Analysis (PCA)



- PCA projects data to maximize variance
- Every PC is different projection
- Segments with length of about 63k

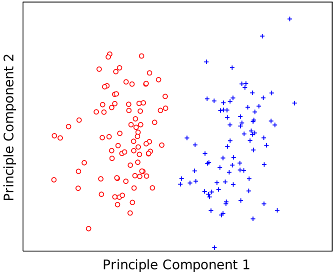
Algorithmic Approach
(2) Principal Component Analysis (PCA)

- Example data from a measurement:
  - Some principal components contain useful information (PC 1)
  - Others only noise (PC 2)



Algorithmic Approach
(3) Cluster Classification

- Clustering means finding a “label” for the segments

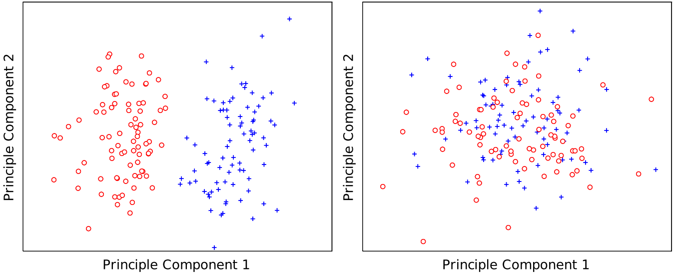


- Expectation-Maximization algorithm trains a Gaussian mixture model
  - Data should consists of 2 Gaussian distributions
  - Difficult to separate, because some “overlap“

Fraunhofer AISEC

Algorithmic Approach
(3) Cluster Classification

- Clustering means finding a “label” for the segments



- Expectation-Maximization algorithm trains a Gaussian mixture model
  - Data should consists of 2 Gaussian distributions
  - Difficult to separate, because some “overlap“

Algorithmic Approach
(4) Result Evaluation

- How well did the attack recover the secret?
- Success metric Brute Force Complexity
  - Estimates the number of bits, which an attacker must test to get the correct key
  - The lower the brute force complexity, the easier is the key to recover (E.g. < 32 bits is very easy)
  - Ranges from 1 to 163 bits

Part I: Analyzing Probes Separately

- As first investigation, we analyzed every probe separately
- Every probe has been put on 400 positions $\rightarrow$ 400 tests for each probe

- For each position and probe:
  - Analyze different components after PCA
  - Perform clustering
  - Calculate brute force complexity as result

Part I: Analyzing Probes Separately
Selecting Principal Components

- We select only few principal components before clustering
  - Useful information concentrated on few principal components
  - remove noise

–> IF right ones are selected → Difficult

- We found that selecting specific single principal components leads to best results
  - We also tested using multiple ones, but this led to worse results average
  - Only the topmost 20 components are useful

- For evaluation:
- Calculate the brute force complexity for each measurement position
- Count the number of tests (measurement positions) which led to each brute force complexity range (similar to histogram)

Part I: Analyzing Probes Separately
Selecting Principal Components

- We select only few principal components before clustering
  - Useful information concentrated on few principal components
  - remove noise

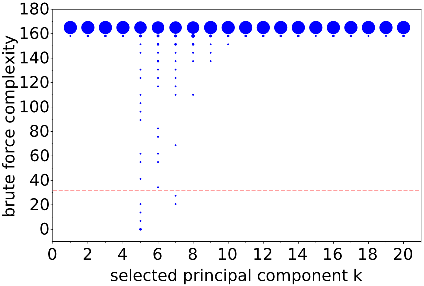
–> IF right ones are selected → Difficult

- We found that selecting specific single principal components leads to best results
  - We also tested using multiple ones, but this led to worse results average
  - Only the topmost 20 components are useful

- For evaluation:
- Calculate the brute force complexity for each measurement position
- Count the number of tests (measurement positions) which led to each brute force complexity range (similar to histogram)

Part I: Analyzing Probes Separately

Results Probe 1 (250 µm)

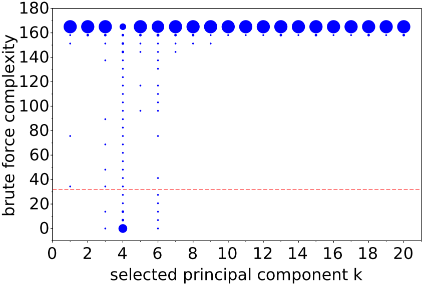


- Only few tests (positions) led to low complexities:
  - 3 % of 400 measurement points below 32 bits when using component number 5
- First components do not contain much leakage, despite highest contained signal variance
  - Most leakage in components 5 to 7

Fraunhofer AISEC

Part I: Analyzing Probes Separately

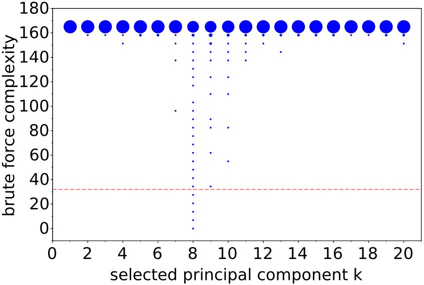
Results Probe 2 (150 µm)



- Much better results than first probe:
  - 56 % of 400 measurement points below 32 bits when using component number 4

Part I: Analyzing Probes Separately

Results Probe 3 (100 µm)



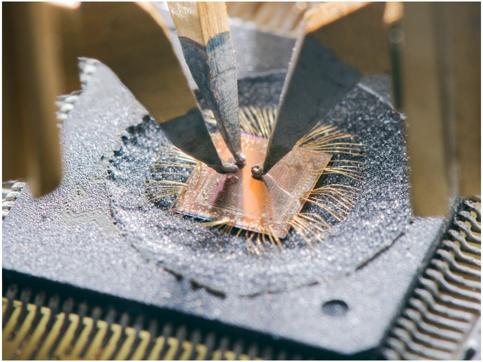
- Again, only few tests (positions) led to low complexities:
  - 3 % of 400 measurement points below 32 bits when using component number 8
- This time in components 7-10

Part I: Analyzing Probes Separately
Summary

- The 150 µm probe led to the best results
  - May be due to individual quality, little closer distance, or FPGA and design
- Selecting single principal components after PCA worked best for clustering
- Not the highest-ranked ones contain most leakage, but ≈ the 3rd to 8th ones
- Comparison to previous method using same measurements (simple trace compression + *k*-means from Heyszl et al., 2012)
  - Clear improvement: 0 % of tests below 32 bits with previous method
- As expected: Profiled template attack still performs better

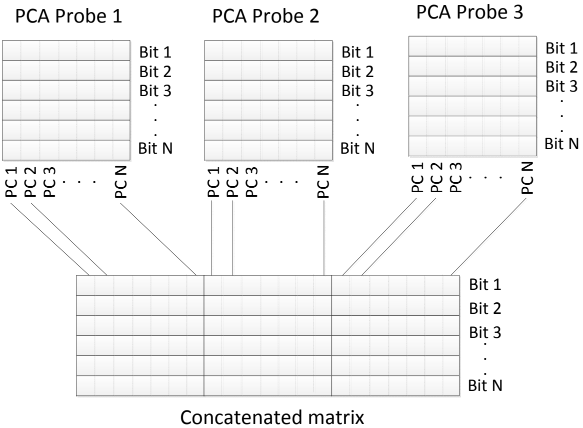
Part II: Combining Multiple Probes

- Available leakage is always limited and Profiling is prevented in many cases
- Goal of possible attackers
  - Use 3 probes instead of one → Combine leakage
  - No need to know positions → Test multiple positions at once



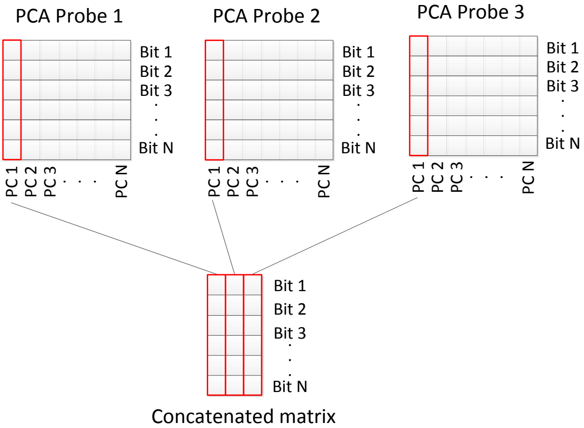
Part II: Combining Multiple Probes

Concatenation after PCA



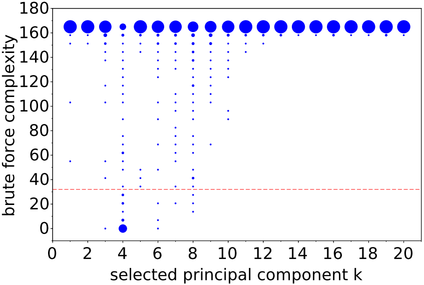
- As before: Only selected principal components are combined

Part II: Combining Multiple Probes
Concatenation after PCA



- As before: Only selected principal components are combined

Part II: Combining Multiple Probes
Results from Combined 3 Probes



- Overall, more tests resulted in complexities < 163 bits
- But combination led to slightly worse results than best single probe:
  - 52 % instead of max. 56 % of 400 measurement points below 32 bits when using component number 4

Part II: Combining Multiple Probes

Summary

- No actual improvement from combining multiple probes for clustering attack
  - Maybe the algorithms are still not perfect
- But: Profiled template attack showed improved results
  - Improvement from best single probe in 82 % of cases
  - 66 % instead of 62 % of 400 measurement points below 32 bits

Conclusions

- Algorithmic improvement for clustering-based, non-profiled attack against asymmetric crypto
  - Using PCA (which is also done in other SCAs)
  - Use selection strategy for single principal components
- No improvement from multiple probes in case of this non-profiled clustering attack
- But: Improvement observed in case of profiled template attack