Higher-Order DCA against Standard Side-Channel Countermeasures

Andrey Bogdanov [1] Matthieu Rivain [2] Philip S. Vejre [1] Junwei Wang [2,3,4]

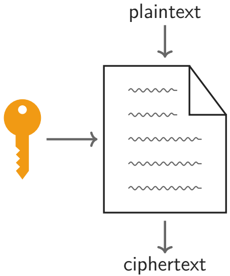
[1]Technical University of Denmark [2]CryptoExperts
[3]University of Luxembourg [4]University Paris 8

COSADE 2019, 4 April, 2019

Overview

1. White-Box Context
2. Differential Computation Analysis
3. Side-Channel Countermeasures
4. Higher-Order DCA
5. Multivariate Higher-Order DCA

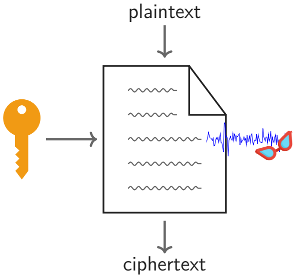
White-Box Threat Model



black-box model

knowing the specification
observing I/O behavior

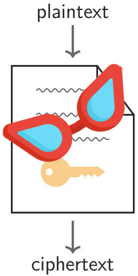
e.g. linear/differential cryptanalysis



gray-box model

+ side-channel leakages
(power/EM/time/· · ·)

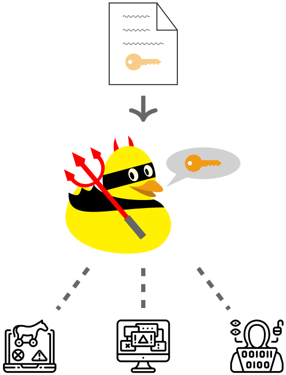
e.g. differential power analysis



white-box model [SAC02]

fully controlling the binary
and its execution environment

White-Box Adversary



- **Goal:** to extract a cryptographic key, $\cdots$
- **Where:** from a software impl. of the cipher
- **Who:** malwares, co-hosted applications, user themselves, $\cdots$
- **How:** (*by all kinds of means*)
  - analyze the code
  - spy on the memory
  - interfere the execution
  - $\cdots$

No provably secure white-box scheme for standard block ciphers.

Typical Applications

Digital Content Distribution

videos, music, games, e-books, ···

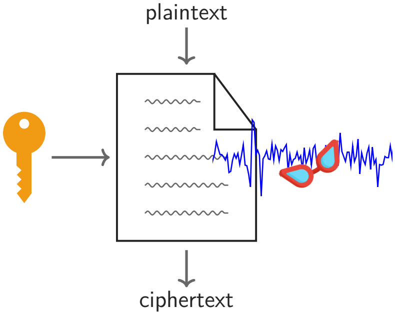
Host Card Emulation

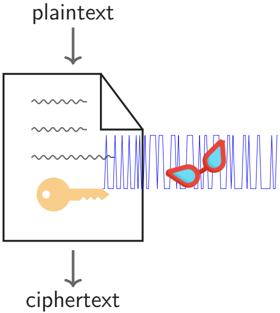
mobile payment without a *secure element*

Overview

1. White-Box Context
2. Differential Computation Analysis
3. Side-Channel Countermeasures
4. Higher-Order DCA
5. Multivariate Higher-Order DCA

Differential Computation Analysis [CHES16]





gray-box model

side-channel leakages (*noisy*)

e.g. power/EM/time/⋯

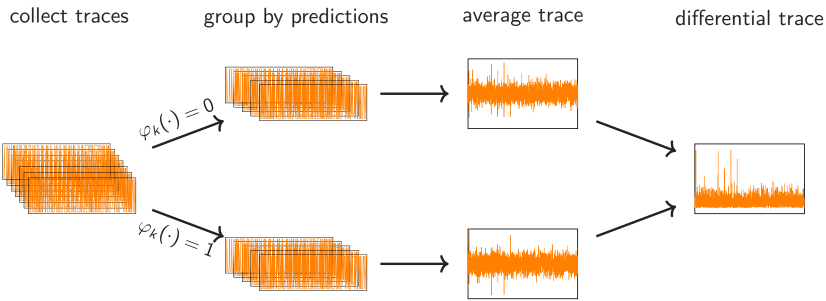
white-box model

computational leakage (*perfect*)

e.g. registers/accessed memory/⋯

Differential Computation Analysis [CHES16]

Differential power analysis techniques on computational leakages



Implying strong *linear correlation* between the sensitive variables and the leaked samples in the computational traces.

Overview

1. White-Box Context
2. Differential Computation Analysis
3. Side-Channel Countermeasures
4. Higher-Order DCA
5. Multivariate Higher-Order DCA

Masking

- Split a sensitive variable x in d shares *s.t.*

$$x = x_1 \oplus x_2 \oplus \cdots \oplus x_d$$

- Any combination of $d - 1$ shares is independent with x.

Shuffling

- **Time shuffling**: randomize the order of computations

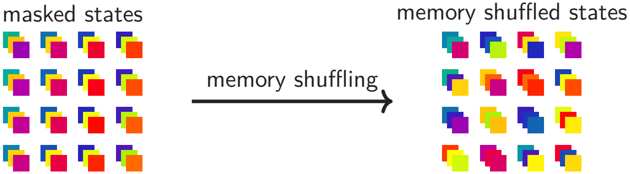
iteration in *normal* order

iteration in *randomized* order

Shuffling

- **Time shuffling**: randomize the order of computations

- But no enough: traces can be *memory* aligned
- **Memory shuffling**: randomize the memory locations of shares



Masking and Shuffling: Security

- No external random source
- Security requirements (informally) for PRNG (seeded by the input plaintext):
  - *Pseudorandomness*: unpredictable outputs
  - *Obscurity*: hiding design
  - *Obfuscation*: preventing reverse-engineering

- Masking is good enough to prevent DCA.
- However, still vulnerable to *linear decoding analysis* (LDA)
  [ia.cr/2018/098; AC18]
- Necessary to introduce *noise*

What about masking + shuffling?

This Work

- We quantify the security brought by masking and shuffling for a passive adversary by introducing
  - the higher-order variant of DCA attack
  - and an optimized multivariate version
- We analyze both attacks and verify our results by simulations
- We showcase the masking and shuffling orders that should be taken in practice

Overview

1. White-Box Context
2. Differential Computation Analysis
3. Side-Channel Countermeasures
4. Higher-Order DCA
5. Multivariate Higher-Order DCA

DCA: a Formal Description

- $N \times t$ matrix $(v_{i,j})_{i,j}$: N computational traces of t time slots
- $\varphi_k(x)$: key dependent predictions
- C: correlation measurement

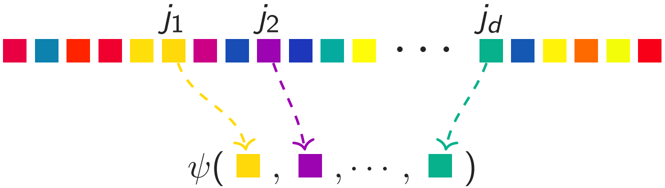
$$\gamma_k = \max_{1 \le j \le t} \mathrm{C}\left(\, (v_{i,j})_i \;\; , \;\; \left(\varphi_k(x_i)\right)_i \right)$$

- Success probability:

$$p_{\mathrm{succ}} = \Pr\left(\mathrm{argmax}_{k \in \mathcal{K}} \gamma_k = k^*\right) \; .$$

Introducing *Higher-Order* DCA

- Trace **pre-processing**: a d-*th order traces* contains $q = \binom{t}{d}$ points:



- Perform DCA attacks on the higher-order traces

Higher-Order DCA against Masking

If only using *masking*:

- $\exists$ fixed $j_1^* < \cdots < j_d^*$ $\quad s.t. \quad \varphi_{k^*}(x) = v_{j_1^*} \oplus \cdots \oplus v_{j_d^*}$ for all traces
- Hence, the natural *combination function* is

$$\psi(v_{j_1}, \cdots, v_{j_d}) = v_{j_1} \oplus \cdots \oplus v_{j_d}$$

- Correlation measurement

$$C_k = \#\text{traces} \quad s.t. \quad \varphi_k(x) = v_{j_1} \oplus \cdots \oplus v_{j_d}$$

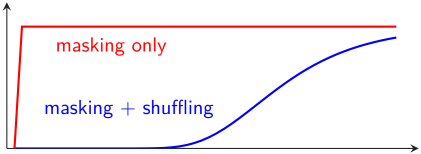
- Even for small N,

$$\gamma_k = \max_j C_k \quad \text{satisfys} \quad \begin{cases} = N & \text{if } k = k^* \\ < N & \text{if } k = k^\times \end{cases}$$

HO-DCA against Masking and Shuffling

If using both *masking* and *shuffling*:

- $\nexists$ fixed $j_1^* < \cdots < j_d^* \quad s.t. \quad \varphi_{k^*}(x) = v_{j_1^*} \oplus \cdots \oplus v_{j_d^*}$ for all traces
- More traces are required to be successful:



- Limitation: each sample in the higher-order traces is considered *independently*

Overview

1. White-Box Context
2. Differential Computation Analysis
3. Side-Channel Countermeasures
4. Higher-Order DCA
5. **Multivariate Higher-Order DCA**

Multivariate Higher-Order DCA

- The multivariate attack optimizes the analysis by exploiting joint information of the higher-order samples on the secrets
- Our proposal is based on a maximum likelihood distinguisher

$$\gamma_k = \Pr\left(K = k | (\boldsymbol{V}_i)_i = (\boldsymbol{v}_i)_i \wedge (X_i)_i = (x_i)_i\right)$$

- We show that

$$\gamma_k \propto \prod_{i=1}^{N} \mathrm{C}_k(\boldsymbol{v}_i, x_i)$$

where *the counter*

$$\mathrm{C}_k(\boldsymbol{v}, x) := \#d\text{-tuples} \quad s.t. \quad v_{j_1} \oplus \cdots \oplus v_{j_d} = \varphi_k(x) \quad \text{in one trace.}$$

Analysis of Multivariate HO-DCA

- **Goal**: to compute the success rate

$$\Pr(\forall k^{\times} \neq k^{*},\ \gamma_{k^*} > \gamma_{k^\times}) = \Pr(\gamma_{k^*} > \gamma_{k^\times})^{|\mathcal{K}|-1}$$

- **Assumption**: each shuffled trace consists of d shares $+$ uniform variables elsewhere

- We define the *zero-counter* event

$$\mathcal{Z}_k = \{\exists \text{ a trace } \quad \text{s.t. } \mathsf{C}_k(\boldsymbol{v}, x) = 0\}$$

- By the law of total probability

$$\Pr(\gamma_{k^*} > \gamma_{k^\times}) = \Pr(\gamma_{k^*} > \gamma_{k^\times} | \mathcal{Z}_{k^\times}) + \Pr(\gamma_{k^*} > \gamma_{k^\times} | \neg\mathcal{Z}_{k^\times})$$

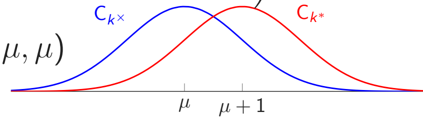
  - $\mathcal{Z}_{k^\times}$ happens $\quad\Longrightarrow\quad \gamma_{k^*} > \gamma_{k^\times} = 0$

$\mathcal{Z}_{k^\times}$ does not Happen

- It is easy to show that

$$\Pr(\gamma_{k^*} > \gamma_{k^\times} | \neg \mathcal{Z}_{k^\times}) = \Pr\left(\frac{1}{N}\sum_{i=1}^{N}(\log \mathsf{C}_{k^*} - \log \mathsf{C}_{k^\times}) > 0 | \neg \mathcal{Z}_{k^\times}\right)$$

- Approximately, $\mathsf{C}_{k^*} \sim \mathcal{N}(\mu+1, \mu)$ and $\mathsf{C}_{k^\times} \sim \mathcal{N}(\mu, \mu)$



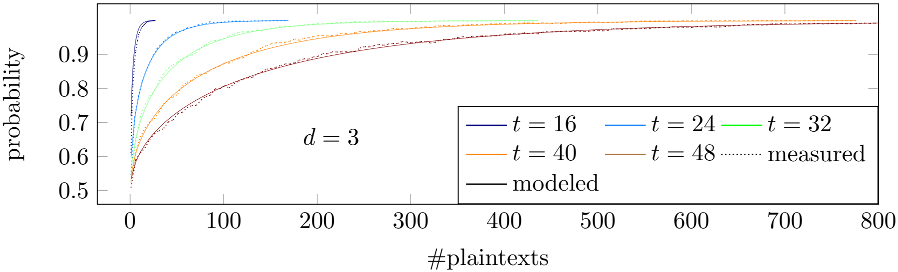
- Thanks to central limit theorem and Taylor expansion

$$p_{\text{succ}} = \Theta\left(erf\left(\frac{1}{2}\sqrt{\frac{N}{\binom{t}{d}}}\right)\right)$$

- Implying the trace complexity $N = \mathcal{O}\left(\binom{t}{d}\right)$

Experimental Verification

- The analysis involves approximations, *e.g.*:
  - ideal assumption on the traces
  - Gaussian approximations of the counters
  - Taylor expansion truncation, etc
- The accuracy is verified by simulations.



Attacking Complexity

- Trace complexity: $N = \mathcal{O}\left(\binom{t}{d}\right)$.
- Computation complexity: $\mathcal{O}\left(|\mathcal{K}| \cdot N \cdot \binom{t}{d}\right) = \mathcal{O}\left(|\mathcal{K}| \cdot \binom{t}{d}^2\right)$.
- A 7-th order masking will bring approximately 85-bit security.

Table: d-th order attacks to achieve 90% success probability, where $|\mathcal{K}| = 256$.

d	$\log_2 N$	$\log_2$ time	d	$\log_2 N$	$\log_2$ time	d	$\log_2 N$	$\log_2$ time
3	10.6	32.7	5	21.0	53.5	7	31.6	74.6
4	15.8	43.1	6	26.3	64.1	8	36.9	85.3

Conclusion

- DCA is an adaption of DPA attack
- It is natural to adapt classical DPA countermeasures
- We propose to higher-order DCA attacks to analyze the effectiveness
- We give close formulae for their success rates and we verify them by simulations
- The security level of this approach is quantified:
  - trace complexity: $N = \mathcal{O}\left(\binom{t}{d}\right)$
  - computation complexity: $\mathcal{O}\left(|\mathcal{K}| \cdot \binom{t}{d}^2\right)$
- Attackers are forced to perform active attack / reverse engineering

Thank You !