

Analysis and Improvement of Differential Computation Attacks against Internally-Encoded White-Box Implementations

Junwei Wang

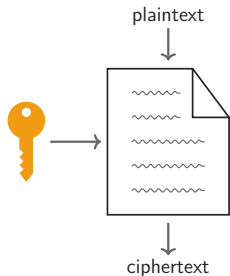
Joint work with Matthieu Rivain

LACS seminar, Feb 20, 2019

Overview

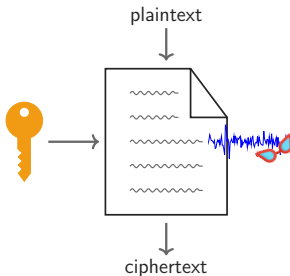
- 1 ■ White-Box Context
- 2 ■ DCA Analysis against Internal Encodings
- 3 ■ *Collision Attack*
- 4 ■ Can We Go Deeper?

White-Box Threat Model



black-box model

knowing the cipher
observing I/O behavior
e.g. linear/differential cryptanalysis



gray-box model

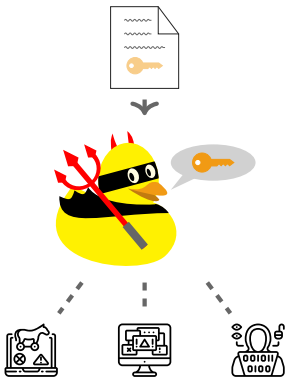
+ side-channel leakages
(power/EM/time/...)
e.g. differential power analysis



white-box model [\[SAC02\]](#)

owning the binary
controlling the environment

White-Box Threat Model



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

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

Typical Applications

Digital Content Distribution

videos, music, games, e-books, ...

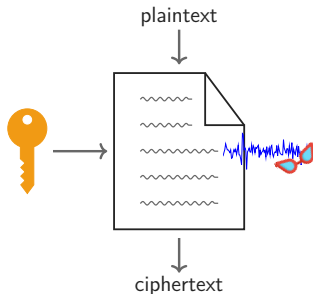


Host Card Emulation

mobile payment without a secure element



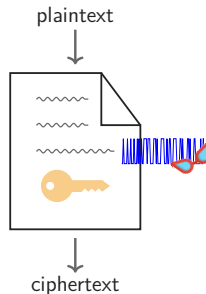
Differential Computation Analysis [CHES16]



gray-box model

side-channel leakages (*noisy*)

e.g. power/EM/time/...



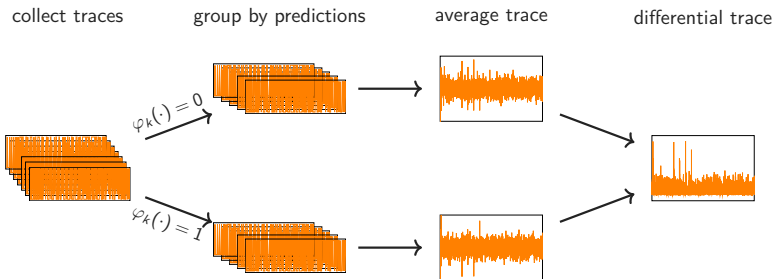
white-box model

computational leakage (*perfect*)

e.g. registers/accessed memory/...

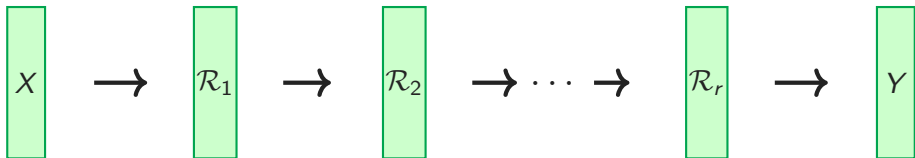
Differential Computation Analysis [CHES16]

Differential power analysis techniques on computational leakages



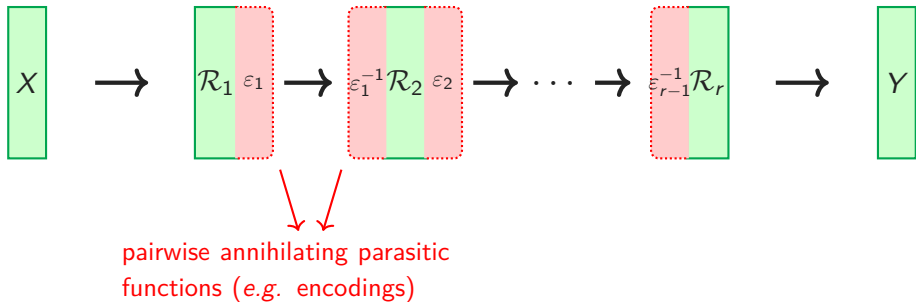
Implying strong *linear correlation* between the sensitive variables and the leaked samples in the computational traces.

Internal Encoding Countermeasure [SAC02]



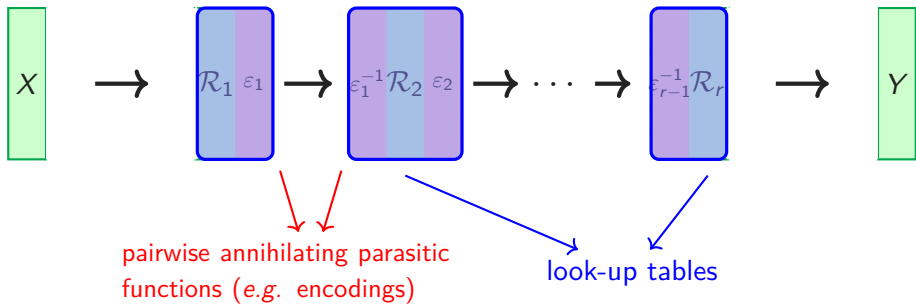
1. Represent the cipher into a *network* of **transformations**
2. Obfuscate the network by **encoding** *adjacent* transformations
3. Store the encoded transformations into **look-up tables**

Internal Encoding Countermeasure [SAC02]



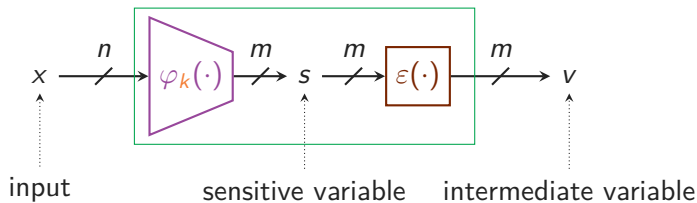
1. Represent the cipher into a *network* of transformations
2. Obfuscate the network by encoding adjacent transformations
3. Store the encoded transformations into look-up tables

Internal Encoding Countermeasure [SAC02]



1. Represent the cipher into a *network* of **transformations**
2. Obfuscate the network by **encoding** *adjacent* transformations
3. Store the encoded transformations into **look-up tables**

Internal Encoding Leakage



- A *key-dependent* (n, m) selection function φ_k in a block cipher
- A *random* selected m -bit bijection ε
- $\varepsilon \circ \varphi_k$, as a result of some **table look-ups**, is **leaked in the memory**
- To exploit the leakage of $\varepsilon \circ \varphi_k$, it is necessary that $n > m$

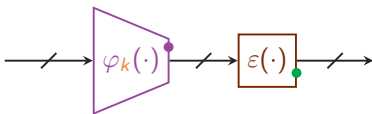
DCA Attack Limitations

1. The seminal work [CHES16] lacks in-depth understanding of DCA
2. The follow-up analysis [ACNS18] is
 - ▶ *partly* experimental (in particular for wrong key guesses)
 - ▶ *Only* known to work on **nibble** encodings
 - ▶ *Only* known to work on the **first** and **last** rounds
 - ▶ Success probability is unknown
3. The computational traces are only sub-optimally exploited

DCA Analysis against Internal Encoding

Based on well-established theory – *Boolean correlation*, instead of *difference of means*: for any key guess k

$$\rho_k = \text{Cor}\left(\varphi_k(\cdot)[i] , \varepsilon \circ \varphi_{k^*}(\cdot)[j] \right)$$



DCA success (roughly) requires:

$$|\rho_{k^*}| \geq \max_{k \neq k^*} |\rho_k|$$

ρ_{k^*} and ρ_{k^\times} : Distributions

- **Ideal** assumption: $(\varphi_k)_k$ are mutually independent random (n, m) functions

Correct key guess k^* ,

$$\rho_{k^*} = 2^{2-m} N^* - 1$$

where

$$N^* \sim \mathcal{HG}(2^m, 2^{m-1}, 2^{m-1}) .$$

Only depends on m .

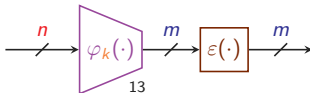
Incorrect key guess k^\times ,

$$\rho_{k^\times} = 2^{2-n} N^\times - 1$$

where

$$N^\times \sim \mathcal{HG}(2^n, 2^{n-1}, 2^{n-1}) .$$

Only depends on n .



Lemma

Lemma

Let $\mathcal{B}(n)$ be the set of balanced n -bit Boolean function. If $f \in \mathcal{B}(n)$ and $g \xleftarrow{\$} \mathcal{B}(n)$ independent of f , then the balanceness of $f + g$ is $B(f + g) = 4 \cdot N - 2^n$ where $N \sim \mathcal{HG}(2^n, 2^{n-1}, 2^{n-1})$ denotes the size of $\{x : f(x) = g(x) = 0\}$.

With

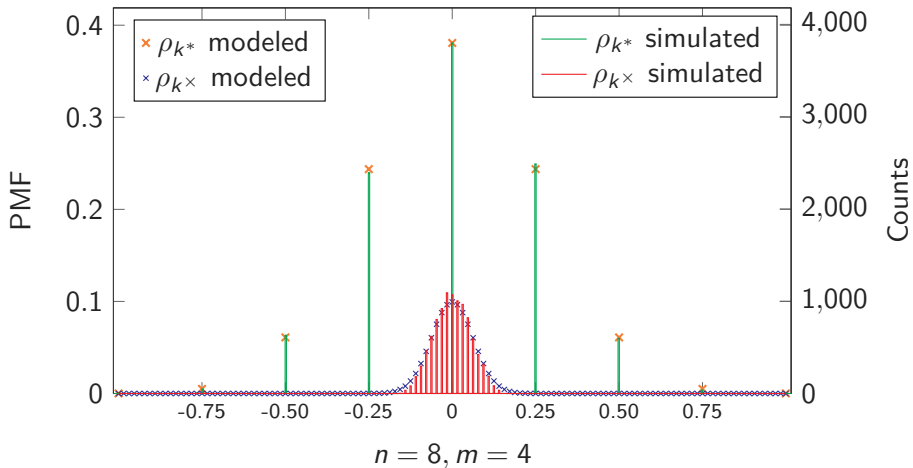
$$\text{Cor}(f + g) = \frac{1}{2^n} B(f + g)$$

\Rightarrow

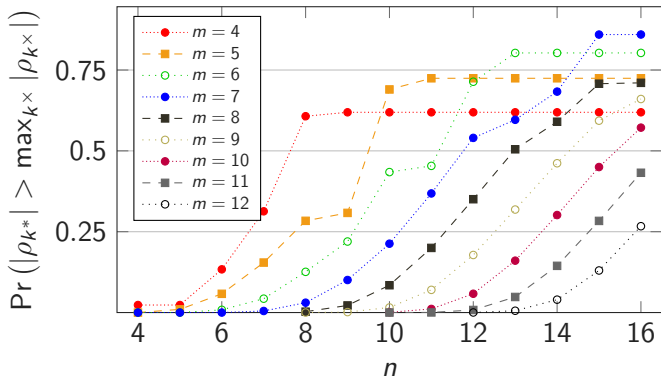
$$\rho_{k^*} = 2^{2-m} N^* - 1 \quad \text{and} \quad \rho_{k^\times} = 2^{2-n} N^\times - 1$$

where $N^* \sim \mathcal{HG}(2^m, 2^{m-1}, 2^{m-1})$ and $N^\times \sim \mathcal{HG}(2^n, 2^{n-1}, 2^{n-1})$.

ρ_{k^*} and ρ_{k^\times} : Distributions



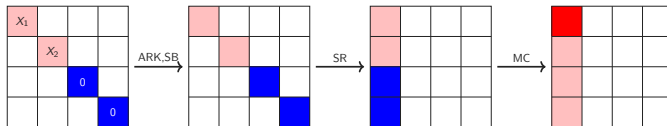
DCA Success Rate: $|\rho_{k^*}| > \max_{k \neq k^*} |\rho_k|$



DCA success probability converges towards $\approx 1 - \Pr_{N^*}(2^{m-2})$ for $n \geq 2m + 2$.

Attack a NSC Variant: a White-Box AES

- Byte encoding protected
- DCA has failed to break it *before this work*
- Our approach: target a output byte of MixColumn in the first round



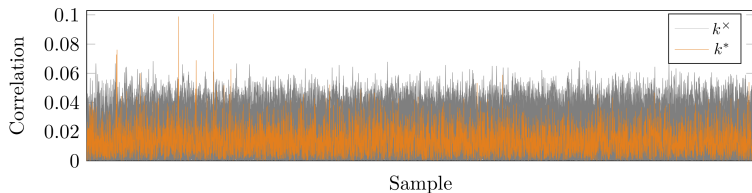
$$\varphi_{k_1||k_2}(x_1||x_2) = 2 \cdot \mathbf{Sbox}(x_1 \oplus k_1) \oplus 3 \cdot \mathbf{Sbox}(x_2 \oplus k_2) \oplus \mathbf{Sbox}(k_3) \oplus \mathbf{Sbox}(k_4)$$

$$\varepsilon' = \varepsilon \circ \oplus_c,$$

$$n = 16, m = 8, |\mathcal{K}| = 2^{16}.$$

Attack a NSC Variant: a White-Box AES

- Attack results: ~ 1800 traces

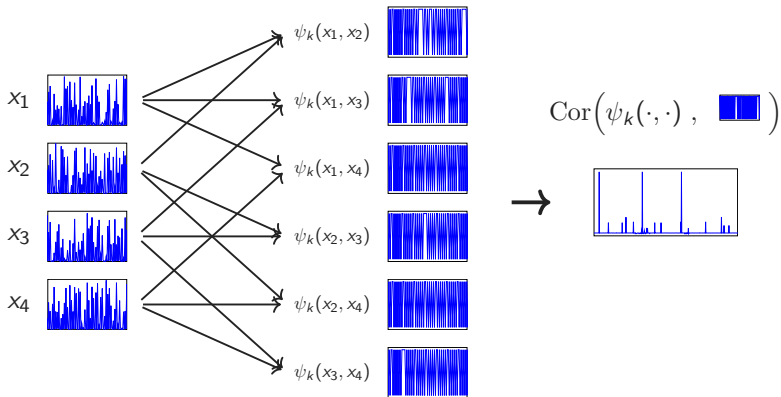


- Similar attack can be applied to a “masked” white-box implementation, which intends to resist DCA.

Collision Attack

N inputs & raw traces

$\binom{N}{2}$ collision predictions & traces



$$\psi_k(x_1, x_2) := (\varphi_k(x_1) = \varphi_k(x_2))$$

Collision Attack: Explanation





















Based on the principle:

$$\varphi_k(x_1) = \varphi_k(x_2) \Leftrightarrow \varepsilon \circ \varphi_k(x_1) = \varepsilon \circ \varphi_k(x_2)$$

Trace Complexity:

$$N = O\left(2^{\frac{m}{2}}\right)$$

Collision Attack: Explanation

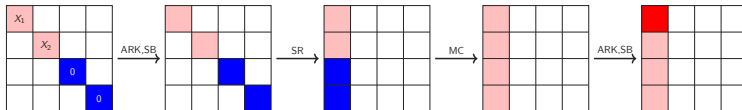
	Predictions					
	1	2	3	4	5	6
key guesses	k_1					
	k_2					
	k_3					
	k_4					

k^* “collides” $\wedge \forall k^\times, k^*$ and k^\times are not “isomorphic”

$$\Rightarrow N = O\left(2^{\frac{m}{2}}\right)$$

Attack the NSC Variant

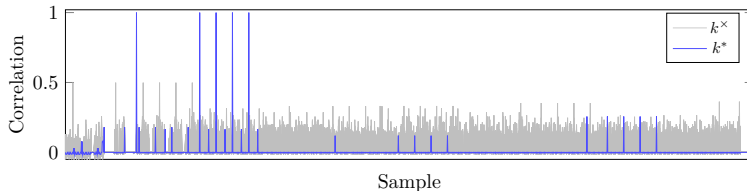
- Same target: a first round MixColumn output byte



$$\varphi_{k_1 || k_2}(x_1 || x_2) = 2 \cdot \mathbf{Sbox}(x_1 \oplus k_1) \oplus 3 \cdot \mathbf{Sbox}(x_2 \oplus k_2)$$

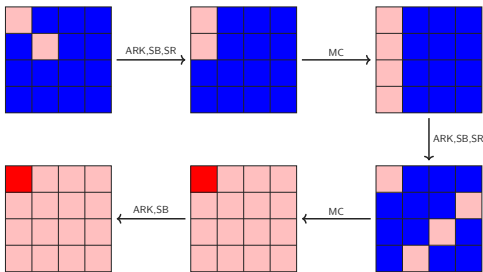
$$\varepsilon' = \varepsilon \circ \oplus_c \quad \text{or} \quad \varepsilon'' = \varepsilon \circ \mathbf{Sbox} \circ \oplus_{c \oplus k'_1}$$

- Attack results: 60 traces



Can We Go Deeper

- YES, WE CAN !!!



$$\varphi_{k_1||k_2||c}(x_1||x_2) = 2 \cdot \mathbf{Sbox}\left(2 \cdot \mathbf{Sbox}(x_1 \oplus k_1) \oplus 3 \cdot \mathbf{Sbox}(x_2 \oplus k_2) \oplus c\right)$$

with $\varepsilon' = \varepsilon \circ \oplus_{c'}$ and $n = 16, m = 8, |\mathcal{K}| = 2^{24}$ where

$$c = \mathbf{Sbox}(k_3) \oplus \mathbf{Sbox}(k_4) \oplus k'_1 \quad \text{and} \quad c' = 3 \cdot \mathbf{Sbox}(\dots) \cdot \mathbf{Sbox}(\dots) \cdot \mathbf{Sbox}(\dots) .$$

Conclusion

- DCA against internal encodings has been analysed in depth
 - ▶ Allows to attack wider encodings
- Computation traces have been further exploited
 - ▶ Showcase to attack variables beyond the first round of the cipher
 - ▶ New class of collision attack with very low trace complexity
- Hence, protecting AES with internal encodings in the beginning rounds is insufficient

Thank You !

ia.cr/2019/076