White-Box Cryptography

Matthieu Rivain

Journées Nationales 2017 Pré-GDR Sécurité Informatique
Paris, 31 mai 2017

CryptoExperts

Outline

- Context:
  - white-box crypto: big trend in the industry
  - cryptographic obfuscation: big trend in the scientific literature
  - huge gap between both
- This presentation:
  - what is (cryptographic) obfuscation?
  - what is white-box cryptography?
  - white-box cryptography in practice

What is (cryptographic) obfuscation?

What is obfuscation?

Obfuscation is the deliberate act of creating obfuscated code that is **difficult for humans to understand**.

Obfuscators make reverse engineering more difficult but **do not alter the behavior** of the obfuscated application.

– wikipedia

What is obfuscation?

Obfuscation is the deliberate act of creating obfuscated code that is **difficult for humans to understand**.

Obfuscators make reverse engineering more difficult but **do not alter the behavior** of the obfuscated application.

– *wikipedia*

⇒ make a program unintelligible while preserving its functionality

Defining obfuscation

Program

- word in a formal (programming) language $P \in \mathcal{L}$
- function $\text{execute} : \mathcal{L} \times \{0,1\}^* \to \{0,1\}^*$

$$\text{execute} : (P, in) \mapsto out$$

- P implements a function $f : \mathcal{A} \to \mathcal{B}$ if

$$\forall a \in \mathcal{A} \ : \ \text{execute}(P, a) = f(a)$$

  denoted $P \equiv f$
- P_1 and P_2 are functionally equivalent if

$$P_1 \equiv f \equiv P_2 \quad \text{for some } f$$

  denoted $P_1 \equiv P_2$

Defining obfuscation

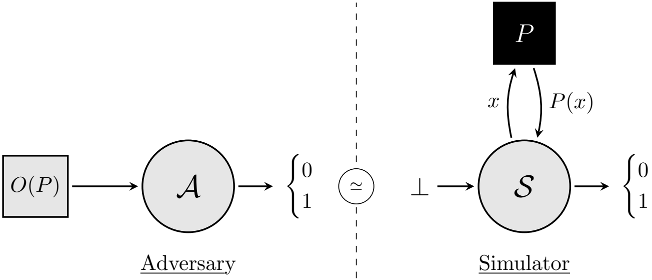
Obfuscator

- algorithm O mapping a program P to a program $O(P)$ st:
- **functionality:** $O(P) \equiv P$
- **efficiency:** $O(P)$ is efficiently executable
- **security:**
  - (informal) $O(P)$ is hard to understand
  - (informal) $O(P)$ protects its data

How to formally define the security property?

Virtual Black-Box (VBB) Obfuscation

- $O(P)$ reveals nothing more than the I/O behavior of P
- Any adversary on $O(P)$ can be simulated with a black-box access to P



$$\left|\Pr[\mathcal{A}(O(P)) = 1] - \Pr[\mathcal{S}^{P}(\perp) = 1]\right| \le \varepsilon$$

Impossibility result

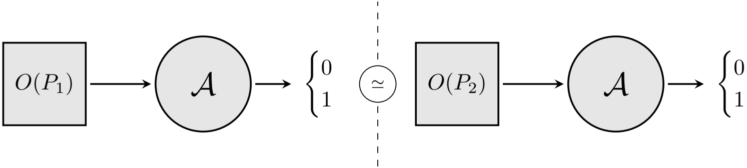
- VBB-O does not exist on general programs (CRYPTO'01)
- Counterexample:

```
uint128_t cannibal (prog P, uint128_t password)
{
    uint128_t secret1 = 0xe075b4f4eabf4377c1aa7202c8cc1ccb;
    uint128_t secret2 = 0x94ff8ec818de3bd8223a62e4cb7c84a4;

    if (password == secret1) return secret2;

    if (execute(P, null, secret1) == secret2) return secret1;

    return 0;
}
```

$$O(\texttt{cannibal})\big(O(\texttt{cannibal}), 0\big) = \texttt{secret1}$$

Indistinguishability obfuscation (iO)

- Restricted to circuits *i.e.* programs without branches/loops
- For any two programs P_1 and P_2 st $P_1 \equiv P_2$ and $|P_1| = |P_2|$, the obfuscated programs $O(P_1)$ and $O(P_2)$ are indistinguishable


$O(P_1)$
$\mathcal{A}$
$\begin{cases}0\\1\end{cases}$
$\simeq$
$O(P_2)$
$\mathcal{A}$
$\begin{cases}0\\1\end{cases}$

$$|\Pr[\mathcal{A}(O(P_1)) = 1] - \Pr[\mathcal{A}(O(P_2)) = 1]| \leq \varepsilon$$

- Best possible obfuscation in some sense

What is white-box cryptography?

What is white-box cryptography?

"the attacker is assumed to have full access to the encrypting software and control of the execution environment"

"Our main goal is to make key extraction difficult."

"While an attacker can clearly make use of the software itself, forcing an attacker to use the installed instance is often of value to DRM systems providers."

– *Chow* et al. *(DRM 2002)*

What is white-box cryptography?

"the attacker is assumed to have full access to the encrypting software and control of the execution environment"

⇒ obfuscation restricted to encryption (or another crypto primitive)

"Our main goal is to make key extraction difficult."

"While an attacker can clearly make use of the software itself, forcing an attacker to use the installed instance is often of value to DRM systems providers."

– *Chow* et al. *(DRM 2002)*

What is white-box cryptography?

“the attacker is assumed to have full access to the encrypting software and control of the execution environment”

⇒ obfuscation restricted to encryption (or another crypto primitive)

“Our main goal is to make key extraction difficult.”

⇒ relaxed security requirements

“While an attacker can clearly make use of the software itself, forcing an attacker to use the installed instance is often of value to DRM systems providers.”

– *Chow* et al. *(DRM 2002)*

What is white-box cryptography?

"the attacker is assumed to have full access to the encrypting software and control of the execution environment"

⇒ obfuscation restricted to encryption (or another crypto primitive)

"Our main goal is to make key extraction difficult."

⇒ relaxed security requirements

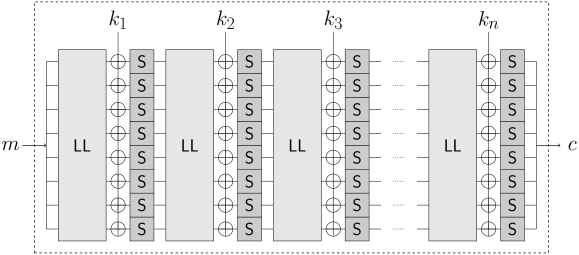
"While an attacker can clearly make use of the software itself, forcing an attacker to use the installed instance is often of value to DRM systems providers."

⇒ encryption software ≠ secret key

– *Chow* et al. *(DRM 2002)*

What is white-box cryptography?

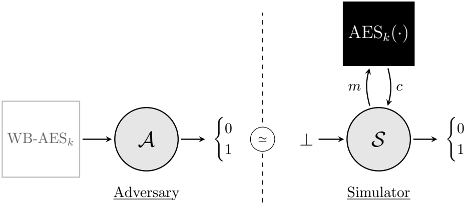
- Obfuscation restricted to a specific class of crypto primitives
- Typically, SPN ciphers:



- Strong structure, could enable specific obfuscation techniques
- Running example: $\left\{\mathrm{AES}_k(\cdot)\,|\,k \in \{0,1\}^{128}\right\}$

VBB-obfuscated AES

- Strongest form of WBC



- Impossibility result does not apply
- The AES-LUT program achieves VBB
  - but does not fit into $10^9 \cdot 10^9 \cdot 10^9$ TB
- How to build a compact VBB AES implementation?
  - could be impossible to achieve

iO-obfuscated AES

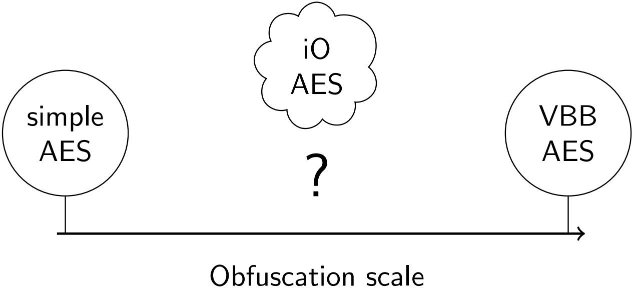
- Is this a good obfuscator?

```
1. $k \leftarrow$ extract-key$(P_k)$
2. return reference implem $\mathrm{AES}_k$
```

  - This is an iO AES obfuscator!

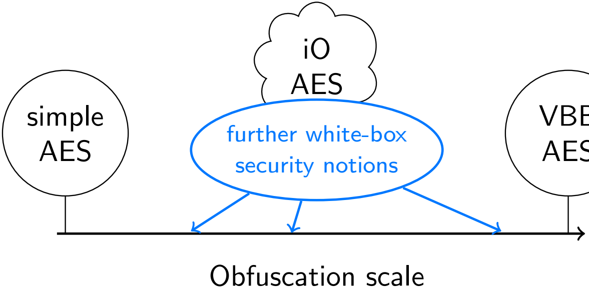
- So what does iO-AES means?
  - not clear!

Defining WBC



- We need something
  - relaxed compared to VBB
  - meaningful compared to iO

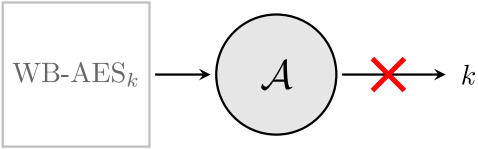
Defining WBC



- We need something
  - relaxed compared to VBB
  - meaningful compared to iO ⇒ further notions
- SAC 2013: *"White-Box Security Notions for Symmetric Encryption Schemes"*

What could we expect?

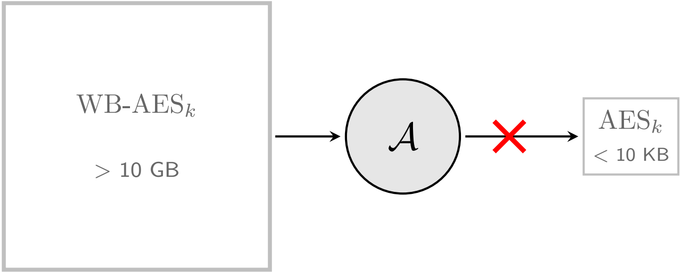
- The least requirement: key extraction must be difficult



- Code-lifting cannot be avoided
- It should be different to have $\mathrm{WB\text{-}AES}_k$ and k
- Using the software should constrain the adversary

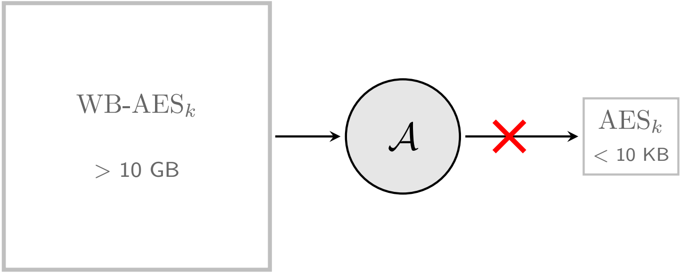
Incompressibility

- Make the implementation huge and **incompressible**



Incompressibility

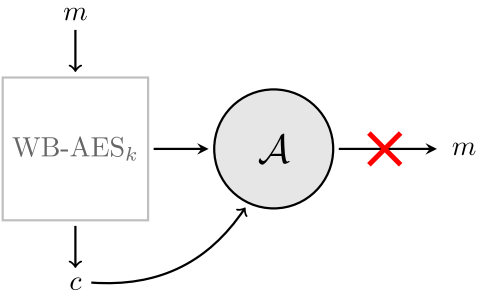
- Make the implementation huge and **incompressible**



- Several new primitives based on this idea
  - Toy example (SAC'13): RSA with large public exponent
  - Block-ciphers based on large tables (CCS'15, ASIACRYPT'16)
  - Big-key cipher (CRYPTO'16)

One-wayness

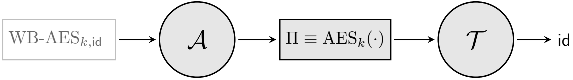
- Make the implementation **one-way**



- Namely: turning AES into a public-key cryptosystem
- PK crypto with light-weight private operations

Traceability

- Include a tracing mechanism

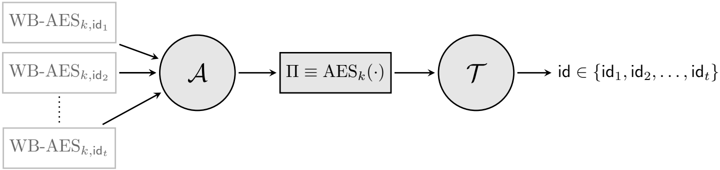

$\text{WB-AES}_{k,\mathsf{id}}$
$\mathcal{A}$
$\Pi \equiv \text{AES}_k(\cdot)$
$\mathcal{T}$
id

$$\exists\, \mathcal{T} \text{ st } \forall\, \mathcal{A} : \text{WB-AES}_{k,\mathsf{id}} \mapsto \Pi \equiv \text{AES}_k(\cdot) \;\Rightarrow\; \mathcal{T}(\Pi) = \mathsf{id}$$

- Possible use case: pay-TV

Traceability

- Include a tracing mechanism



$\mathrm{WB\text{-}AES}_{k,\mathsf{id}_1}$
$\mathrm{WB\text{-}AES}_{k,\mathsf{id}_2}$
$\mathrm{WB\text{-}AES}_{k,\mathsf{id}_t}$
$\mathcal{A}$
$\Pi \equiv \mathrm{AES}_k(\cdot)$
$\mathcal{T}$
$\mathsf{id} \in \{\mathsf{id}_1, \mathsf{id}_2, \ldots, \mathsf{id}_t\}$

$$\exists\, \mathcal{T} \text{ st } \forall\, \mathcal{A} : \mathrm{WB\text{-}AES}_{k,\mathsf{id}} \mapsto \Pi \equiv \mathrm{AES}_k(\cdot) \;\Rightarrow\; \mathcal{T}(\Pi) = \mathsf{id}$$

- Possible use case: pay-TV

White-box cryptography in practice

Original white-box AES

- SAC 2002: *"White-Box Cryptography and an AES Implementation"* (Chow *et al.*)

- First step: represent AES as a network of look-up tables
  - Each AES round composed of 4 *sub-rounds* of the form:

$$(y_0, y_1, y_2, y_3) = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \otimes \begin{pmatrix} S(x_0 \oplus k_0) \\ S(x_5 \oplus k_5) \\ S(x_{10} \oplus k_{10}) \\ S(x_{15} \oplus k_{15}) \end{pmatrix}$$

Original white-box AES

- This rewrites:

$$(y_0, y_1, y_2, y_3) \ = \ T_0[x_0] \oplus T_5[x_5] \oplus T_{10}[x_{10}] \oplus T_{15}[x_{15}]$$

  where $T_i : 8$ bits → 32 bits defined as

$$\begin{aligned} T_0[x] &= S(x \oplus k_0) \times (02\ 01\ 01\ 03)^T \\ T_5[x] &= S(x \oplus k_5) \times (03\ 02\ 01\ 01)^T \\ T_{10}[x] &= S(x \oplus k_{10}) \times (01\ 03\ 02\ 01)^T \\ T_{15}[x] &= S(x \oplus k_{15}) \times (01\ 01\ 03\ 02)^T \end{aligned}$$

- XORs performed with a table $T_{\text{xor}} : 8$ bits → 4 bits

$$T_{\text{xor}}[x_0 || x_1] = x_0 \oplus x_1$$

Original white-box AES

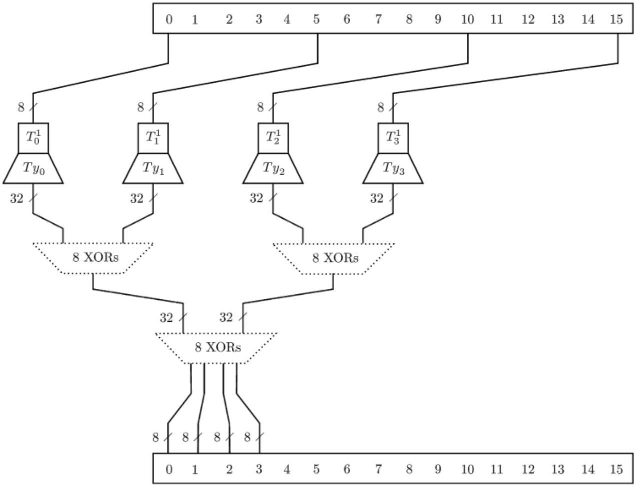


Illustration: J. Muir *"A Tutorial on White-box AES"* (ePrint 2013)

Original white-box AES

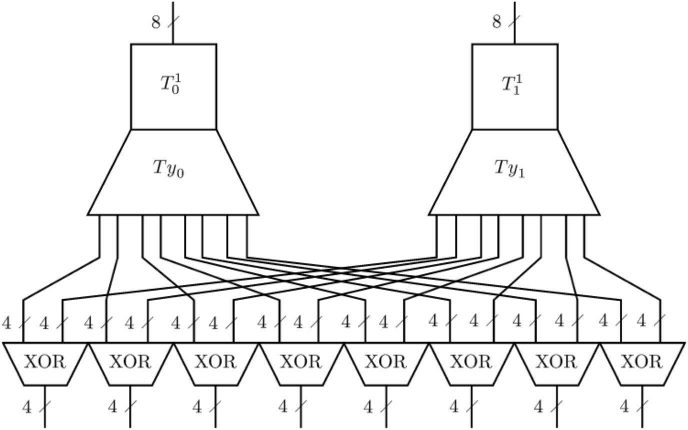


Illustration: J. Muir *"A Tutorial on White-box AES"* (ePrint 2013)

Original white-box AES

- Second step: randomize the look-up tables
  - Each table T is replaced by

    $$T' = g \circ T \circ f^{-1}$$

    where f, g are **random encodings**
  - For two *connected* tables T, R

    $$\begin{array}{l} T' = g \circ T \circ f^{-1} \\ R' = h \circ R \circ g^{-1} \end{array} \quad \Rightarrow \quad R' \circ T' = h \circ (R \circ T) \circ f^{-1}$$

- Intuition: encoded tables bring no information on the key
  - true for a single table
  - true for a chain $g \circ T_n \circ T_{n-1} \circ \cdots \circ T_1 \circ f^{-1}$
  - not true for the larger picture

Original white-box AES

- Consider the *encoded sub-round* (32 bits → 32 bits):

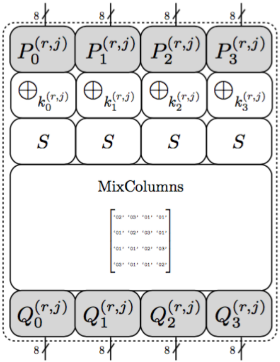
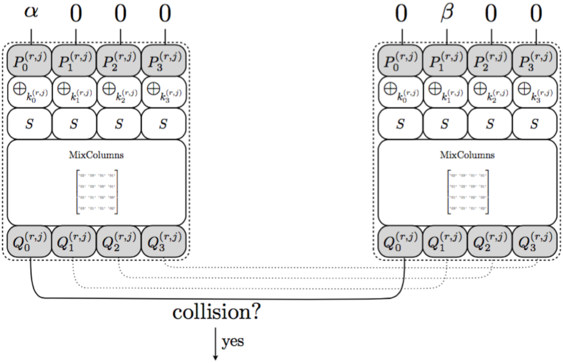


Illustration: Y. De Mulder (presentation SAC 2013)

- The four key bytes can be easily extracted from it

Original white-box AES

- For instance, a simple collision attack:



α 0 0 0
0 β 0 0
$P_0^{(r,j)}$ $P_1^{(r,j)}$ $P_2^{(r,j)}$ $P_3^{(r,j)}$
$\oplus_{k_0^{(r,j)}}$ $\oplus_{k_1^{(r,j)}}$ $\oplus_{k_2^{(r,j)}}$ $\oplus_{k_3^{(r,j)}}$
S S S S
MixColumns
$Q_0^{(r,j)}$ $Q_1^{(r,j)}$ $Q_2^{(r,j)}$ $Q_3^{(r,j)}$
collision?
yes

$$02 \cdot S_0(\alpha) \oplus 03 \cdot S_1(0) = 02 \cdot S_0(0) \oplus 03 \cdot S_1(\beta)$$

where $S_0(x) = S(P_0(x) \oplus k_0)$ and $S_1(x) = S(P_1(x) \oplus k_1)$

Illustration: Y. De Mulder (presentation SAC 2013)

A white-box cemetery

- Many attacks on the original design:
  - First break: BGE attack (Billet *et al.* SAC 2004)
  - Generic attack on WB SPN ciphers (Michiels *et al.* SAC 2008)
  - Improved BGE attack (Lepoint *et al.* SAC 2013)
  - Collision attack (Lepoint *et al.* SAC 2013)
  - Classical fault attacks (*e.g.* Piret-Quisquater, CHES 2003)
  - Differential Computational Analysis (Bos *et al.* CHES 2016)
- Other attempts:
  - Perturbed WB-AES using MV crypto (Bringer *et al.* ePrint 2006)
    ⇒ broken (De Mulder *et al.* INDOCRYPT 2010)
  - WB-AES based on wide linear encodings (Xiao-Lai *et al.* CSA 2009)
    ⇒ broken (De Mulder *et al.* SAC 2012)
  - WB-AES based on dual AES ciphers (Karroumi, ICISC 2010)
    ⇒ broken (Lepoint *et al.* SAC 2013)
- Not better for white-box DES...

White-box crypto today

- All published WB implementations have been broken
- Yet the industrial need exists
  - Digital content protection (pay-TV, DRM)
  - Mobile payments
  - Software protection
- Advent of the secret specification paradigm
  - mix of WBC techniques and code obfuscation
  - attackers don't know how the implementation was generated
  - this is called **security through obscurity**
  - Kerckhoffs / Shannon must be turning in their graves!
- Development of generic attacks against WBC
  - Differential Computational Analysis (DCA)
  - Differential Fault Analysis (DFA)

Differential Computational Analysis

- Suggested by NXP / Riscure
    - Presentation at BalckHat 2015
    - Best paper award CHES 2016

- Dynamically record all the data-dependent information during an execution
    - memory writes / loads (addresses + values)
    - logical instructions (operands + result)
    - program counter ⇒ re-synchronization

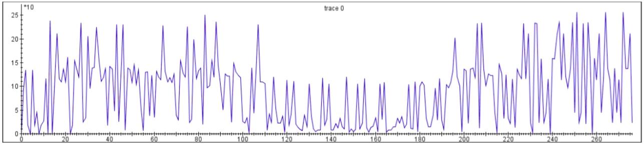
  ⇒ computational trace

- Apply DPA techniques to computational traces

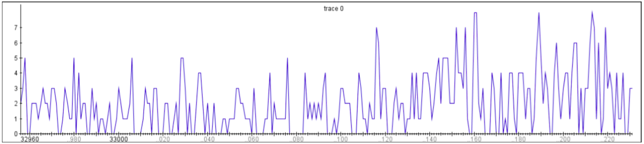
$$\rho(\text{bit computed if } k = \hat{k},\ \text{computational trace}_k)$$

Differential Computational Analysis

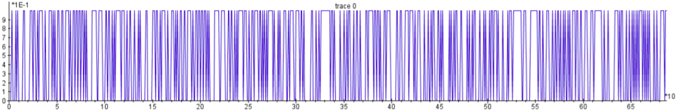
- Example – trace build from 8-bit data reads:



- Corresponding trace with Hamming weights:



- Corresponding bits:



Traces: J. Bos (presentation CHES 2016)

Differential Computational Analysis

- Advantages of DCA:
  - generic / not specific to some WB design
  - does not require reverse engineering efforts
  - defeats all attempts of code-obfuscation
  - seems to work remarkably well in practice

- Why does it work?
  - in most WB implementations, randomization is static
  - encodings are generated once and hardcoded
  - we might have

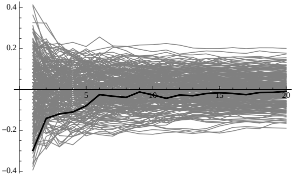
$$\rho(x_i, \mathsf{Enc}[x]_j) \gg 0$$

  - especially with 4-bit encodings

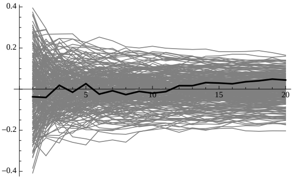
$$\mathsf{Enc}[x_0 \,\|\, x_1] = \mathsf{Enc}[x_0] \,\|\, \mathsf{Enc}[x_1]$$

Differential Computational Analysis

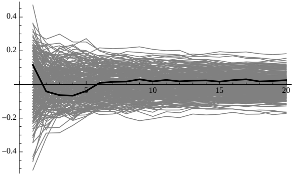
- Simple correlation experiment (1000 samples)
  - Target variable: Enc$[S(x \oplus k)]$ (4-bit encoding)
  - Model: MSB of $S(x \oplus \hat{k})$



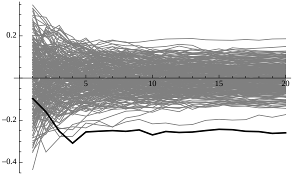
Bit 0



Bit 1



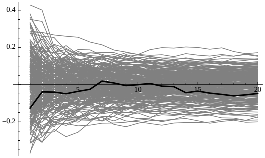
Bit 2



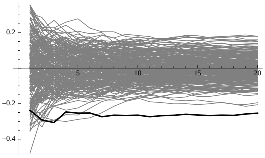
Bit 3

Differential Computational Analysis

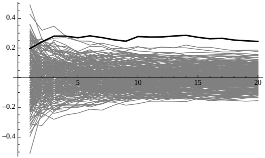
- With another (4-bit) encoding:



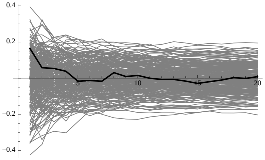
Bit 0



Bit 1



Bit 2



Bit 3

- Most of the time 1, 2, or 3 bits leak
- For 8-bit encodings: correlation to weak

Countermeasures?

- We know a lot of countermeasures against DPA / DFA
- It might be hard to translate them to the white-box context
- Attackers have a full control of the implementation
  - any external RNG can be set to 0
  - easy synchronization of traces (all can be monitored)
  - dummy operations can be easily detected and removed
  - coherence checks can be bypassed
- Only pseudorandomness based on the plaintext is possible
- Countermeasures must be obfuscated to be hard to remove

Conclusion

Conclusion

- WBC can be define as a restriction of cryptographic obfuscation
    - subset of programs (*e.g.* given crypto primitive)
    - relaxed security notions
- More work is needed to
    - refine / define alternative security notions
    - build new candidate white-box implementations
    - analyze DCA / DFA against WBC
    - translate existing countermeasures to the white-box context
- Open question: who beats who?
    - secret-spec designer *vs.* state-of-the-art cryptanalyst

Interested in white-box crypto?

- WhibOx 2016, white-box cryptography and obfuscation
  14 Aug. 2016, UCSB

  `https://www.cryptoexperts.com/whibox2016/`

- Ecrypt white-box challenge, coming soon
  - submit your (secret-design) white-box AES
  - try to break submitted implementations

- Visit CryptoExperts website:

  `https://www.cryptoexperts.com/technologies/white-box/`

Questions ?