



Looking into the White Box

Pascal Junod
@cryptopathe



Talk Roadmap

- A Standard Scenario
- White-Box Cryptography
- Security Models
- The Academic Viewpoint
- Building Secure-Enough White-Box Primitives
- Cryptographic Perspective





A Standard Scenario




A Standard Scenario - Remote attestation




GET /api/v1.0/signup/



... you look like a  ...

GET /api/v1.0/signup/




... you look like a  ...

A Standard Scenario - Remote attestation



GET /api/v1.0/signup/ and btw,
here is a MAC-signed random
challenge




... but what if your MAC  get
stolen?



GET /api/v1.0/signup/ and btw,
here is a MAC-signed random
challenge



... is it a stolen MAC  ?



A Standard Scenario - Remote attestation



GET /api/v1.0/signup/ and btw,
here is a WBC-AES-encrypted
random challenge



... sure, but what if your WBC
crypto is code-lifted ?



GET /api/v1.0/signup/ and btw,
here is a WBC-AES-encrypted
random challenge



... sure, but is it
code-lifted WBC crypto ?



A Standard Scenario - Remote attestation



GET /api/v1.0/signup/ and btw,
here is a WBC-AES-encrypted
random challenge



GET /api/v1.0/signup/ and btw,
here is a WBC-AES-encrypted
random challenge

Well done, you have managed to
carve the WBC. 🍑



White-Box Cryptography



White-Box Cryptography

White-box cryptography deals with **implementations** of **cryptographic algorithms** running in the **most hostile** computing environments, i.e., in the white-box security model.

White-Box Cryptography - Why ?

White-box security is THE relevant security model in many real-world scenarios

White-Box Cryptography - Why ?

Non-trivial gap between the academic state-of-the-art about white-box cryptography and industry practices

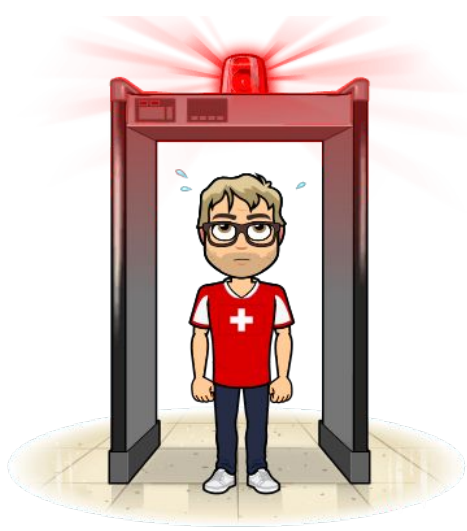
- Pressure from market, see e.g., Host-Card Emulation (HCE)
- Pressure from real-world adversaries (DRM)

White-Box Cryptography - Why ?

You think that cryptography is magic ? Then white-box cryptography is $(magic + sorcery + wizardry)^2$

- Allows to transform AES in RSA encryption
- Allows to transform HMAC-SHA256 in RSA signature





Security Models

Security Models - Black-Box Security

- Crypto primitives abstracted by black boxes (aka “oracles”)
- Well-defined API, which the adversary respects
- Various attack models considered by cryptographers
 - Encryption schemes
 - Ciphertext-only
 - Known plaintext
 - Chosen plaintext (adaptive vs. non-adaptive variants)
 - Chosen plaintext and ciphertext
 - Signature schemes
 - Existential/selective/universal forgery

Security Models - Grey-Box Security

- Model considered only since the mid 90's by cryptographers
- Strict superset of black-box security
 - All capabilities of black-box adversaries
 - + additional exploitation of some (physical) information about the scheme's implementation
 - Time
 - Power consumption
 - EM leakage
 - Sound leakage
 - Faults injection

Security Models - White-Box Security

- Model considered only since beginning of 00's by academics
- Worst conditions to do crypto
 - All black-box capabilities
 - + all grey-box capabilities
 - + full control of implementation and its environment
 - Static reverse engineering (disassemblers, decompilers, etc.)
 - Dynamic reverse engineering (debuggers, code instrumentation, emulators, hypervisors, symbolic/concolic execution, etc.)
 - Arbitrary fault injection capabilities in code and data
 - Arbitrary inspection of registers, memory and storage

Security Models - Summary

Black-box security

- Cryptography operated in trusted environments
- Remote and properly secured API, e.g. signing oracle for a CA
- “Mathematical insurance”



Security Models - Summary

Grey-box security

- Secure hardware environments
- CPUs, smartcards, USB dongles, TPMs, secure STB chipsets, etc.



Security Models - Summary

White-box security

- Software-only environments, when no secure HW element is available
- Untrusted endpoints (laptop, mobile phone, etc.)
- Aka “man-at-the-end” security model



The Academic Viewpoint



Academic Viewpoint - Design and Attack Times

- White-box cryptography model proposed by Chow et al. in 2002
 - Implementations of DES and AES “securely” embedding a hard-coded key
 - Supposed to **resist** to **key extraction**
 - Relying on internal secret bijective encodings, expressed as table lookups
 - Implementations consist of about 100’s to 1000’s kB of precomputed tables
 - Quickly broken using different types of attack strategies (black- and grey-box)
- Several other designs proposed, some relying on multivariate cryptography
- Currently, **all published designs have been broken**



Cryptanalysis of White-Box DES
Implementations with Arbitrary External
Encodings

Brecht Wyseur¹, Wil Michiels², Paul Gorissen², and Bart Preneel¹

Attacking an obfuscated cipher by injecting faults

Matthias Jacob
mjacob@cs.princeton.edu
Dan Boneh
dabo@cs.stanford.edu
Edward Felten
felten@cs.princeton.edu

Cryptanalysis of a White Box
AES Implementation

Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi*

Differential Computation Analysis:
Hiding your White-Box Designs is Not Enough

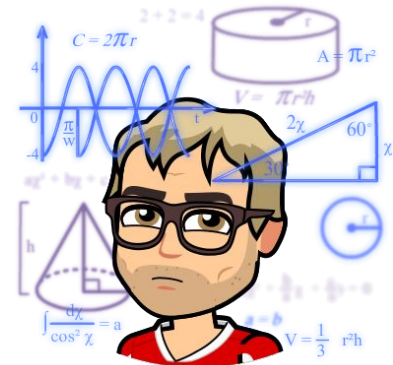
Joppe W. Bos¹, Charles Hubain^{2*}, Wil Michiels¹, and Philippe Teuwen¹

Academic Viewpoint - Theoretical WB Security

Several formal notions of white-box security have been formalized.

- Virtual Black-Box Property
- Indistinguishability Obfuscation
- One-Wayness
- Incompressibility

NB: the white-box compiler is always assumed to be **public !**

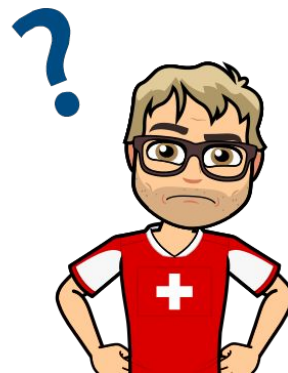


Academic Perspective - Theoretical WB Security

Virtual Black-Box Property [BGI+01]

"Given a VBB obfuscator $\mathbf{O}()$, everything that can be computed from $\mathbf{O}(P)$ can also be computed given an oracle to the program P ."

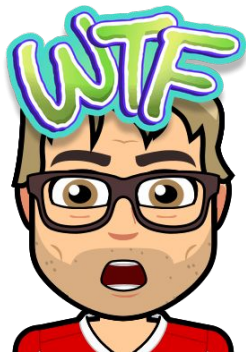
- Known results
 - "A generic obfuscator **does not exist**, i.e., there exist programs that cannot be VBB-obfuscated."
 - VBB obfuscators have been published for some very specific classes of functions



Academic Viewpoint - Theoretical WB Security

Indistinguishability Obfuscation [BGI+01]

“Given an indistinguishability obfuscator $\mathbf{iO}()$ and two equivalent circuits C_1 and C_2 , the two distributions $\mathbf{iO}(C_1)$ and $\mathbf{iO}(C_2)$ are indistinguishable”



- Known results
 - First (inefficient) candidate published in [GGH+13]
 - Several cryptographic primitives have been derived from an iO obfuscator

Academic Viewpoint - Theoretical WB Security

One-Wayness (aka strong white-box) [DLPR13, BBK14]

"Given the implementation of an encryption scheme, it is infeasible to decrypt."

- Known results
 - Some proposals exist, however based on public-key techniques



Academic Viewpoint - Theoretical WB Security

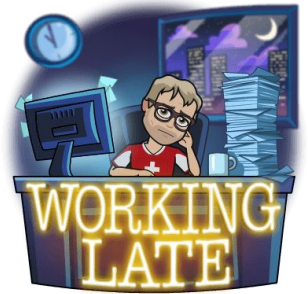
Incompressibility (aka weak white-box, space hardness)
[DLPR13, BBK14, BI15]

"Given an implementation of a white-boxed primitive with a certain size, it is infeasible to derive a smaller implementation thereof."

- Known results
 - Some proposals exist, that typically use large pseudo-random precomputed tables.



Building “Secure-Enough” White-Box Primitives



Resistance to Key Extraction

- Let's assume that one is looking for a good resistance to key extraction
 - Sufficient (but not always necessary !) to break one-wayness
- What are the requirements behind “robust-enough” white-box crypto?
 - Black-box adversaries
 - Grey-box adversaries
 - White-box adversaries



Resistance to Black-Box Attacks

- First of all, we need a secure crypto primitive
- Many engineering details to define
 - Static or dynamic key ?
 - Implementation updatability
 - What is the impact of a broken WBC instance on subsequent WBC instances ?
 - Crypto primitive ? Mode of operation ?
 - AES only, implementation of mode left “outside” ?
 - Authenticated-encryption primitive ?
 - Standard algorithm ?
 - AES ?
 - Custom and secret algorithm ?
 - How to derive randomness on an untrusted terminal ?



Resistance to Grey-Box Attacks - Timing

WBC implementations must be time-constant.

- Depending on the algorithm nature, time-constantness can be tricky.
- Standard (time-) blinding techniques use randomness
 - In a white-box scenario, randomness coming from the system cannot be trusted
- Interactions with code obfuscators
 - Existing time dependences can be amplified by obfuscating compilers
 - E.g., code virtualization
 - Higher sensitivity to cache misses
 - Time dependences can sometimes be accidentally introduced by obfuscating compilers

Resistance to Grey-Box Attacks - Leakage

WBC implementations must be leakage-free.

- Leakage prevention
 - Probes of which order ?
 - Splitting secret data in multiple statistically uncorrelated shares
 - Use blinding techniques
 - (Implement leakage resilient cryptography)
- Main challenge
 - Most leakage prevention mechanisms are supposed to use “secure” randomness

Resistance to Grey-Box Attacks - Faults

WBC implementations must resist faults injection.

- Faults injection prevention
 - Redundant computations
 - Use of internal integrity checks
 - Use of standard software tamper-proofing techniques
- Main challenge
 - Final performances

Resistance to White-Box Attacks

- As of today, we have no choice but accept to use a pragmatic approach
 - Efficient cryptographic obfuscation is not really here
 - Size and performance matter in practice
- Goal is making the adversary's job as costly as possible
 - Leverage custom, secret algorithms and secret white-box compilers
 - Defend against code-lifting attacks
 - Defend against software reverse engineering

Resistance to White-Box Attacks - Custom Algos

In a white-box context, one can and should, whenever possible, get rid of Kerckhoff's principle.

“Security by obscurity”

vs.

“Obscurity on top of security”

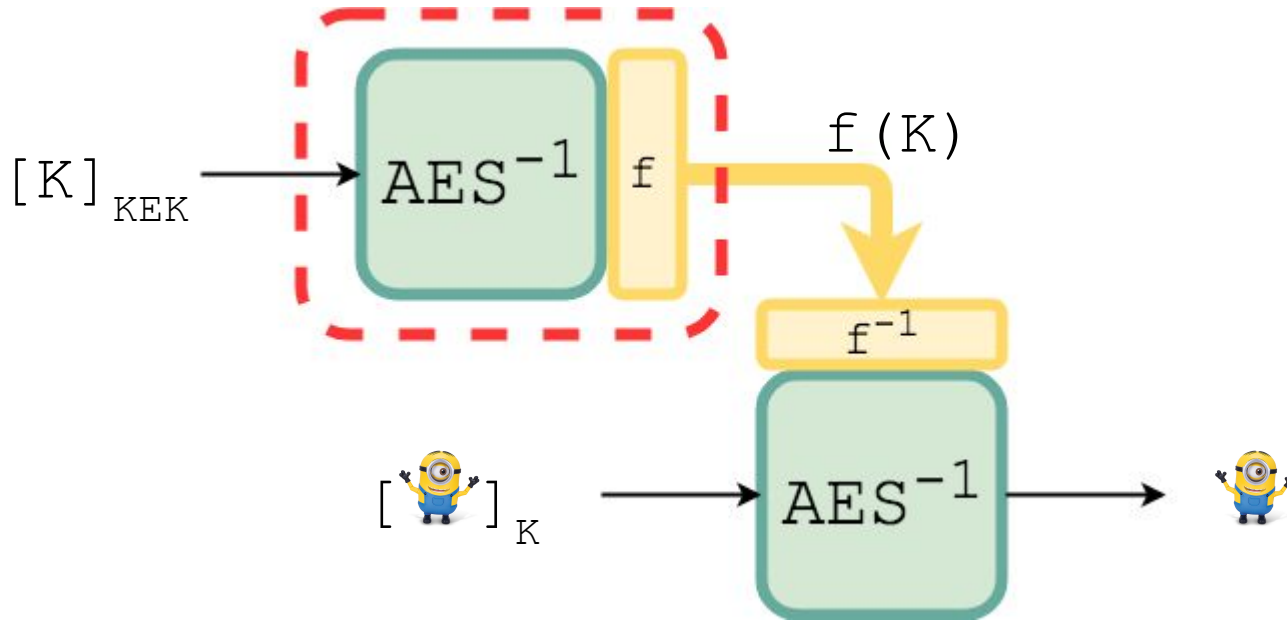


Caveat emptor: don't design your own crypto if you are not a black belt cryptographer.

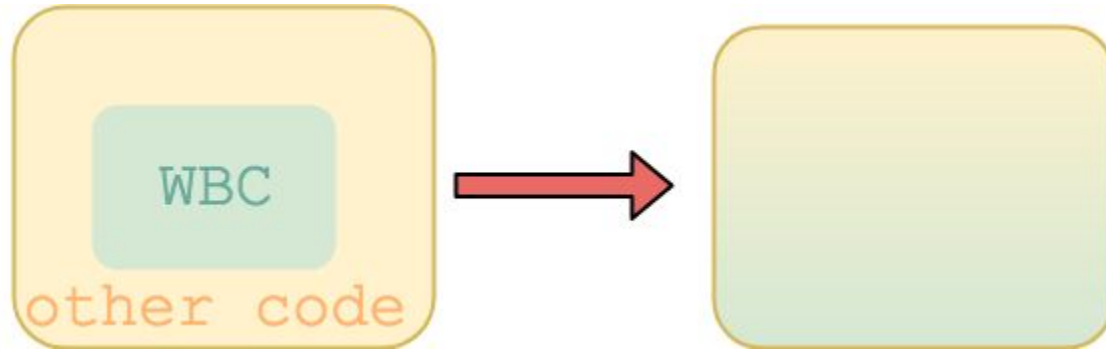
Resistance to White-Box Attacks - Code Lifting

- Code lifting attack
 - Use of a WBC implementation as an encryption/decryption/signature oracle
 - No need to understand its inner workings
 - Requires reverse engineering of WBC API boundaries
 - Easy: dynamic libraries
 - Less easy: code carving in a native binary
- Solutions
 - External encodings
 - Dissolving in other, neighbour executable code

Resistance to White-Box Attacks - Encodings



Resistance to White-Box Attacks - Code Dissolving



- Code dissolving, thanks to a software obfuscator
 - Functions merging
 - Functions splitting



Cryptographic Perspective

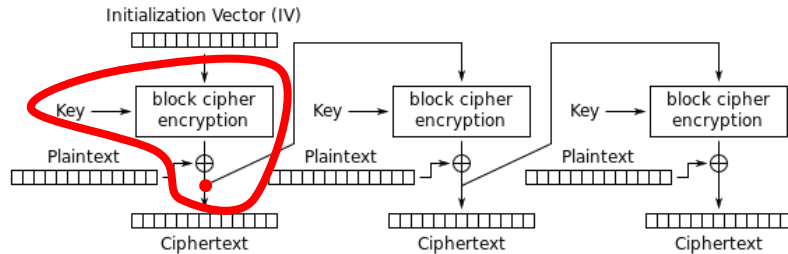
Cryptographic Functionalities

- Many subtleties hide into the use of white-boxed cryptographic primitives
- Examples:
 - CTR mode
 - MACs
 - AES-GCM
 - RSA-OAEP

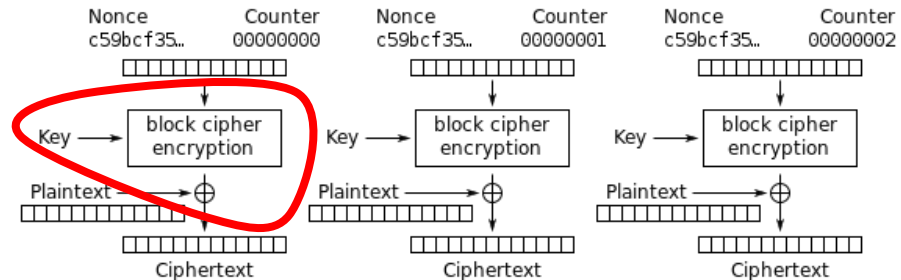


Symmetric Mode of Operations - CTR/OFB/CFB

- CTR and OFB modes do not provide any resistance to inversion!
 - Given an encryption (decryption) oracle, it is trivial to derive a decryption (encryption) oracle.
 - Up to nonce generation mechanism
- CFB is a quasi-symmetric mode
 - How costly is it to identify the red point in the WBC code?



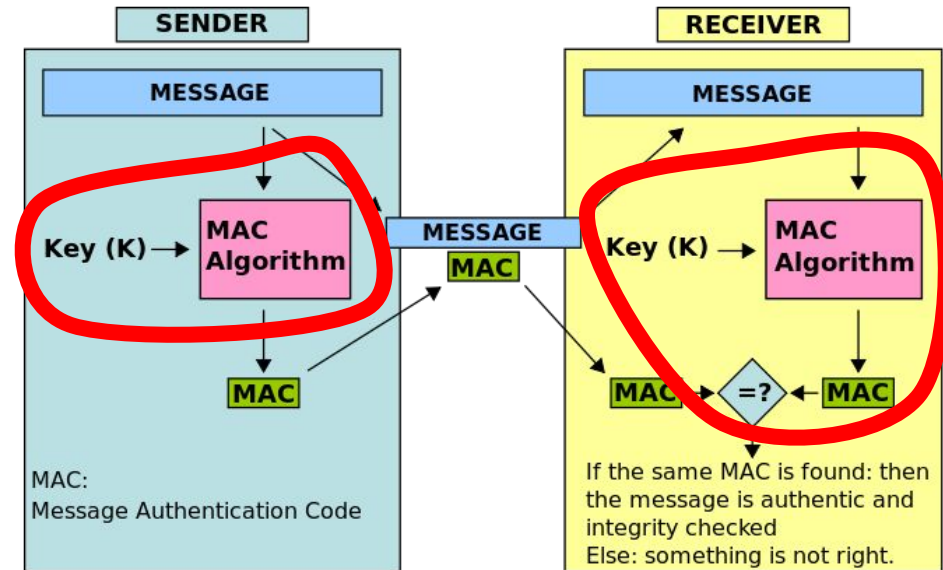
Cipher Feedback (CFB) mode encryption



Counter (CTR) mode encryption

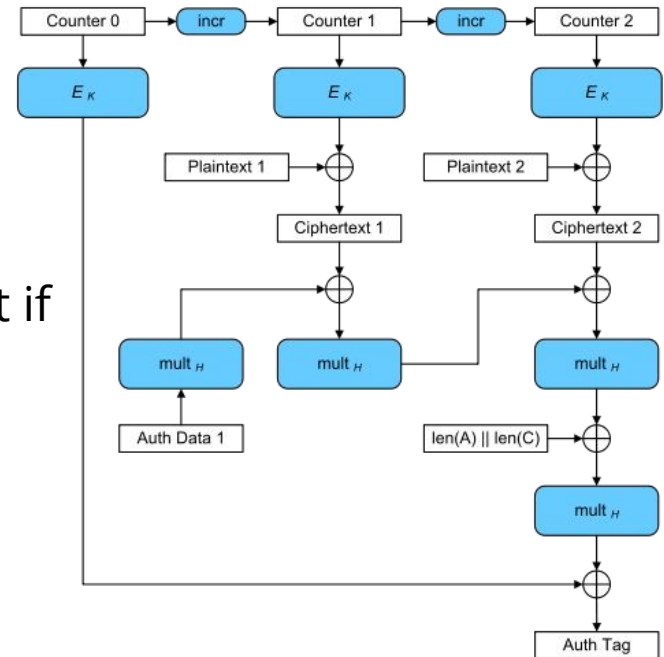
Message Authentication Codes

- In most MACs, the tag generation and verification procedures are identical, up to the tag comparison part.
 - HMAC-SHA256
 - (Encrypted) CBC-MAC
 - Poly1305-AES
 - ...



Authenticated Encryption - AES-GCM

- Like CTR mode, the encryption and decryption directions are very similar
- Strong resistance to inversion is unlikely
- Only difference:
 - Tag generation mechanism
- In an ideal world, the decryption implementation should return the plaintext if and only if the authentication tag is valid.



Public-Key Encryption: RSA-OAEP

- Is it possible to recover a public modulus N out of a white-boxed RSA-OAEP encryption routine, assuming public exponent $e = 65537$?
- Possible solution:
 - Stick the randomness to a known constant
 - Given $\text{pad}(M)$, compute
 - $C = \text{RSA-OAEP}(\text{pad}(M)) = \text{pad}(M)^e \pmod{N}$
 - $C' = \text{pad}(M)^e$
 - NB: for $\text{sizeof}(N) == 2048$ bits and $e = 65537$, C' will be around 2^{27} bits.
 - $\text{gcd}(C, C')$ which is N , or a very small multiple thereof

Time to Conclude

- We barely know how to implement secure cryptography in the white-box model
- Academic research still at the start of the journey
- Still, WBC is useful in practice and many non-published designs are deployed in the wild

THANKS

