



White-Box Cryptography

Security Hardware in Theory and Practice
– A Marriage of Convenience
Dagstuhl, June 2008

Bart Preneel
(joint work with Brecht Wyseur)
K.U. Leuven – COSIC
bart.preneel@kuleuven.be

2

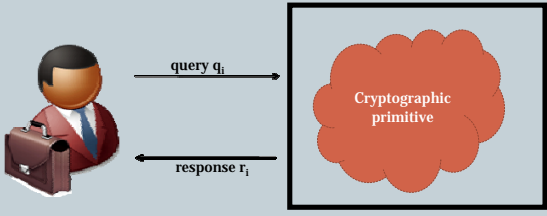
Outline

- Introduction to white-box cryptography
 - Concept and model
- White-Box DES implementations
 - Construction
 - Cryptanalysis
- Discussion

Security Notions

3

- Model an adversaries goals in terms of “games”
 - CPA, CCA, IND-CPA, NM-CPA, ... etc.

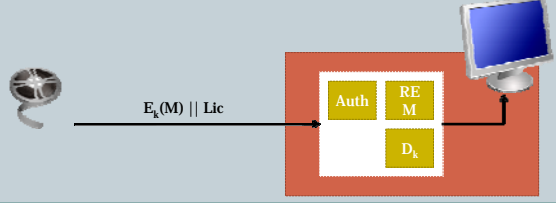


- “Black-box” security (oracle access)

Defeat of the Black-Box: DRM

4

- Digital Rights Management
 - A media player with embedded decryption key
 - Extraction of key information → compromise of DRM scheme
 - × CSS, AAC3, BD+, ... have been broken



Defeat of the Black-Box: Software Protection

5

- Online Games (e.g., World of Warcraft)
 - Adversaries have incentive to manipulate (local) state information

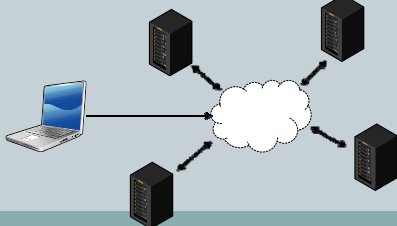



- Virtual Reality

Defeat of the Black-Box

6

- Mobile Agents
 - Mobile code, performing a task, given by its creator, without any interaction
 - Threat: compromise of task and secret information by a server



Cryptography in Untrusted endpoints

7

- **Multi-party computation/Garbled circuits**
 - Overhead in computation and communication
 - If endpoints are untrusted, why should a majority be honest?
- **Asymmetric cryptography**
 - Authenticity of public keys
 - Homomorphic encryption → rely on a trusted party for decryption of result
- **Symmetric cryptography**
 - Can one hide keys by obfuscation?

Hardware solutions

8

- **Implement keys and algorithms in hardware tokens**
 - USB dongles
 - Smart cards in set-top boxes
 - Trusted Platform Modules (TPM)

Hardware: +	Hardware: -
- secure zone for storage and computation	- flexibility (e.g., online updating)
- tamper resistance	- cost
- hard to clone	- malicious hardware
→ black box cryptography	- side-channel analysis

Side-channel cryptanalysis

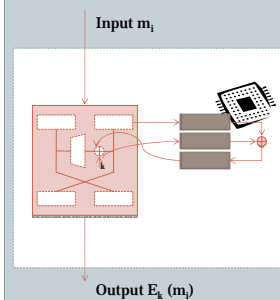
9

- **Defeat of hardware implementations**
 - Power analysis (SPA, DPA)
 - Timing analysis
 - Electromagnetic radiation
 - Fault injection
 - ...
- **Secure circuits**
 - Create model (model the information leakage)
 - Prove security in that model (reduction prove; computational proof (bounded/unbounded adversaries); ...)
 - But... what if an adversary does not comply with the model?



White-Box Model [Chow-Eisen-Johnson-Van Oorschot 02]

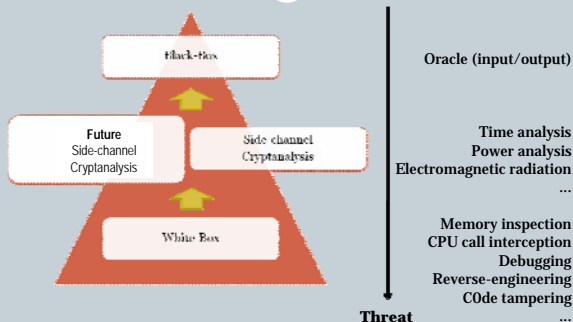
10



- **Threat model (strong: unlimited access)**
 - Read memory/registers
 - Cache attacks
 - Inserting break-points
 - Force a system crash
 - Tamper code
 - Modification of internal variables
 - Dynamic analysis of the implementation
 - ...
- **Adversary's goal (weaker)**
 - Extract "key" information

White-Box vs. Black-Box

11



Software Attacks

12

- **Entropy attack**
 - Use of randomness properties of keys, in contrast to surrounding code
 - Memory/binary dump:



key information

- "Cold reboot" attacks* on full disk encryption

* [http://citp.princeton.edu/memory/, 2008]

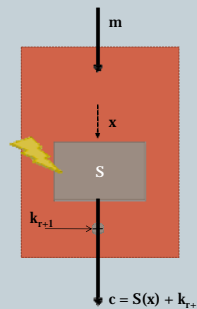
Software

13

- Existing 'solutions'
 - Splitting the cryptographic key into pieces stored in different locations in memory [Aucksmith et al.]
 - Make linear transformations to data values [Collberg et al.]
 - Problem in cryptography: need to "undo" transformations before any non-linear operation
 - Split key into different subkeys, under some relation f (e.g., XOR)
 - $k_2 = f(K, k_1)$
- But: vulnerable to static and dynamic analysis
 - Tracing of program execution (e.g., with IDA-Pro)
 - Entropy analysis [Shamir and Van Someren, 1998]
- Pre-'White-Box Crypto' era: "Cryptographic keys for reasonably secure ciphers cannot be securely hidden in software"

Software Attacks

14



- Key Whitening Attack
 - Attack target: SPN block ciphers with *key whitening* and static S-boxes
 - An easy way to mount an attack on software binaries
 - Identify and overwrite S-boxes in static binary
 - $c_t = (S_t(x)=0) + k_{t+1}$

[Kerins and Kursawe, WISec 2006]

White-Box Cryptography

15

The art of implementing a cryptographic primitive in a "secure" way, albeit under attack in a white-box attack context

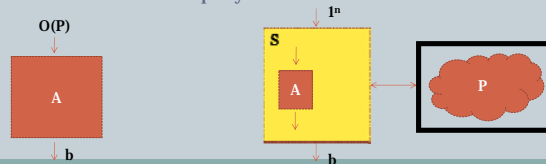
Obfuscation [Barak+01]

16

- Produce a distribution
- Satisfy a relation
- Compute a function or a predicate

- Obfuscation: an adversary does not gain any knowledge when having white-box information (i.e., the implementation) at hand, as compared to having oracle access

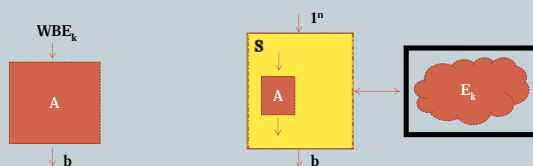
- $\forall A, \exists S$, such that $\Pr[A(O(P)) = 1] - \Pr[S^P = 1] < \text{negl}(n)$
- "Virtual Black-Box Property"



Theoretical WBC

17

- Towards (im)possibility results on WBC
- Simulation based proofs, inspired by research on provable security and obfuscation



- BB security notions \rightarrow WB security notions?

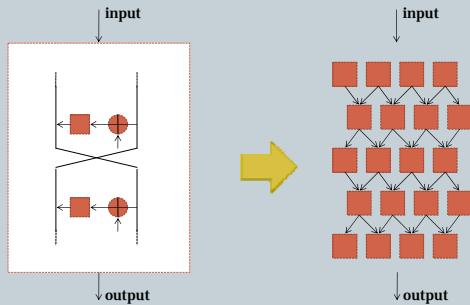
Approach?

18

- Ideal block cipher for a fixed key: one huge random lookup table ($2^n \cdot n$ bits)
- n -bit block cipher with a k -bit key spans only a fraction of size 2^k of the space of $2^{n!}$ permutations
- Goal: approximate ideal cipher with tables of much smaller size; make 'internal' information ambiguous
- Transform into a **randomized** network of key-dependent lookup tables
 - fixed key implementations

Main idea of WBC

19



Internal Encodings

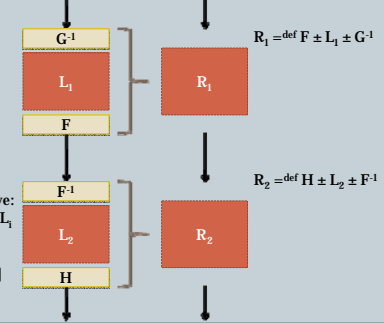
20

$L_1, L_2: GF(2^n) \rightarrow GF(2^n)$

$F: GF(2^n) \rightarrow GF(2^n)$
a random bijection

$LF = \text{def } F \pm L_1$
In the case that LF is bijective:
 $\forall L_1; \exists F_1$ such that $LF = F_1 \pm L_1$

Information theoretical
local security [Shannon '49]



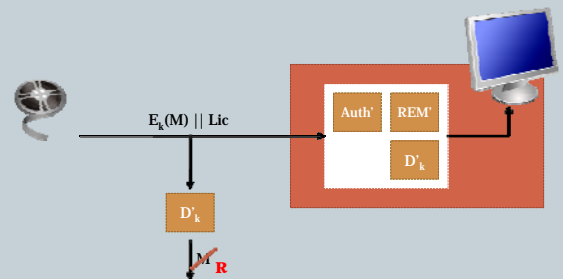
Key?

21

- Issue: what is a key?
- Adversary can still
 - Attempt to isolate the entire “oracle implementation”, and use this as some sort of key
 - Implement a functional equivalent implementation (without ‘unfriendly’ subroutines)
- Goal: force an adversary to **execute** the implementation in order to encrypt/decrypt/...
 - Watermark software, add traceability [BG'03]
 - Hook white-box implementations into the containing application (\rightarrow enable deployment of authentication code)

Back to DRM

22



External Encodings

23

- Implement $G \circ D_k \circ F$, instead of D_k
- Issue: not original scheme any more
 - Pre and post-processing on input and output at other components of system (local/remote)
 - Local: **interlock** implementation into software container, extending the cryptographic boundaries
 - Remote: effective against “global cracks”
- Second motivation: prevent attacks on first and last round
- Security relies on “cryptographic strength” of underlying cipher, when external encodings are chosen independently at random
 - Search space of functions that the cipher might compute is at least as large as the original cipher’s search space

Metrics

24

- Diversity
 - # ‘encodings’ of an implementation due to injection of randomness (# distinct constructions)
- Ambiguity
 - # alternative interpretations of a specific instance
 - Instance: lookup table \rightarrow “local security”
 - Instance: cipher such as DES \rightarrow related keys (e.g., via the DES complementation property)

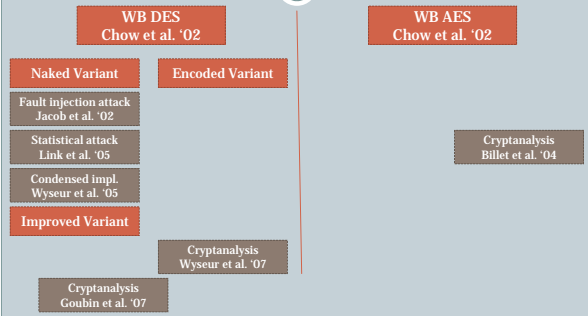
Results and Observations

25

- A **publicly known** transformation of a cryptographic cipher into a randomized network of lookup tables
 - Bulkier and slower than original (unsecure) implementations (but in certain applications, this can be justified)
 - White-Box Cryptography \neq security by obscurity
- **WBC as toolbox for asymmetric crypto: public encryption key: $WB(E_k)$; private decryption key: k**
 - However, stronger security requirements (invertibility)
- **Many other observations: generic tool for software diversification; enable tamper resistant code; prevent side-channel cryptanalysis; ...**
- **Challenge: reduction proofs of white-box security to black-box security**

State-of-the-art (constructions)

26

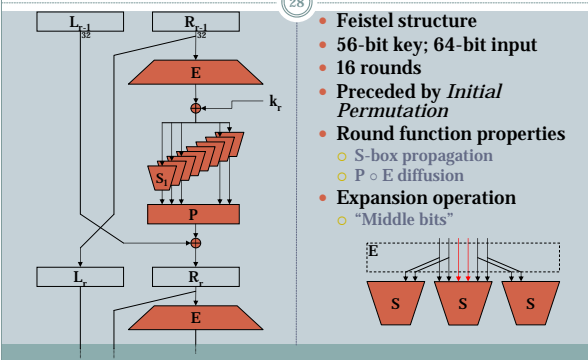


White-Box DES Implementations

- CONSTRUCTION
- CRYPTANALYSIS
- CONCLUSIONS

Data Encryption Standard (DES)

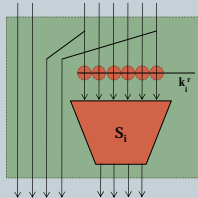
28



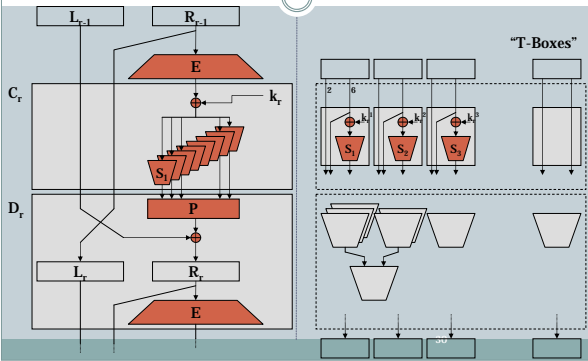
T-Boxes

29

- **Embed key information into bijective primitives**
- **Construction of T-boxes**
 - Partial evaluation
 - Split-path encoding
 - By-pass encoding
- **T: $GF(2^n) \rightarrow GF(2^n)$**
 - Size: $2^n \cdot n$ bit
 - Bijective \rightarrow suitable for to obtain 'local security'



Data Encryption Standard [Chow+'02]

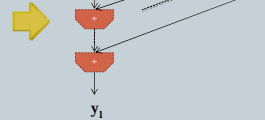


Matrix Decomposition

31

- Transform a linear operation into a network of LUTs

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_m \end{bmatrix}$$



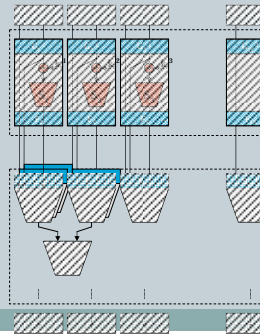
- Sparse matrices

- Leakage of information of internal encodings
- Transform $M \rightarrow B \circ (B^{-1} \circ M)$, with B a mixing bijection

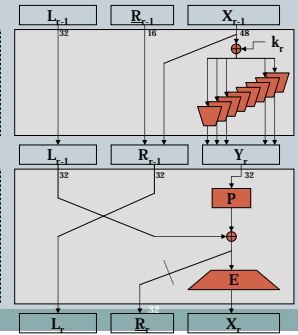
White-Box DES Implementations [Chow+'02]

32

Lookup table implementation

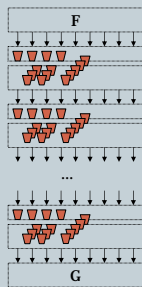


Semantic representation



Result

33



- External encodings

- Against extraction of the full implementation from the containing software
- Against attacks on the first and/or last round

- Result

- A network of key-dependent, randomized lookup tables
- Known structure [Kerckhoffs; no "security through obscurity"]
- Link et al.: 2.25 MB

Cryptanalysis of White-Box DES Implementations

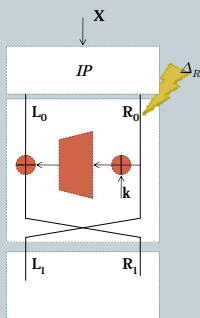
2 CRYPTANALYSIS RESULTS

Goubin et al.
SAC'07

Wyseur et al.
SAC'07

WB DES Cryptanalysis (1)

35



- Truncated differential attack

- On "naked-DES"

- Procedure:

- $X := IP^{-1}(L_0 || R_0)$ random
- $R_0' := R_0 + \Delta_R$ with Δ_R flip on 2 middle bits
- Guess k , and compute L_0' such that $R_1' = R_1$ (on simulated DES instance)
- Compute difference propagation at end of round 1 on WB DES instance
- Verify

[L. Goubin, J-M Masereel, M. Quisquater, SAC'07]

WB DES Cryptanalysis (1)

36

- Attack on "nonstandard-DES"

- Block-level analysis of IP \circ F
- Recovery of columns of F (by finding Δ such that $F(\Delta) = e_i$)
- Assumption: linear external encodings

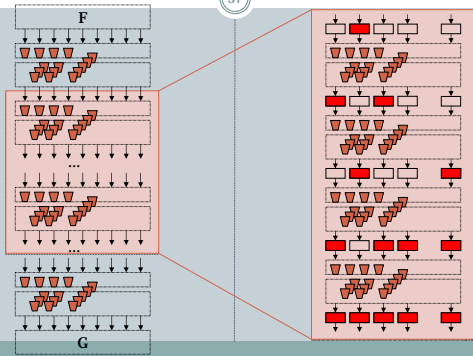
- Deploy the "naked-DES" attack

- Result

- In 95% of the cases: key recovery in below 50 seconds (on a "standard" PC)
- Works only with linear external encodings...

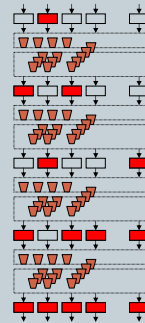
WBDES Cryptanalysis (2)

37



WBDES Cryptanalysis (2)

38

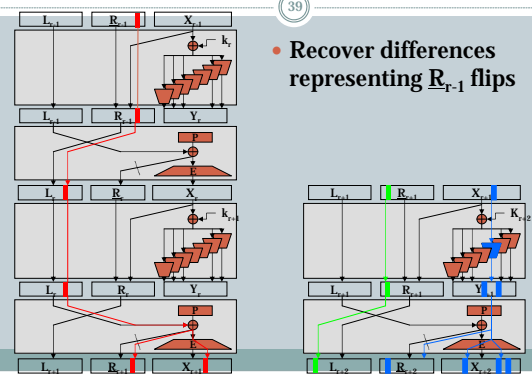


- **Differential cryptanalysis on obfuscated rounds**
 - Independent of external encodings
- **Procedure:**
 - Random input X
 - Inject faults at input of round r
 - Study difference propagation at inputs of rounds r+1, r+2, ...
 - Distinguish flips of S-box input bits
 - Identify S-boxes in T-boxes, and study their difference propagation (which is input dependent) → recover S-box input
 - Recovery of key information

[Wyseur-Michiels-Gorissen-Preneel'07]

WBDES Cryptanalysis (2)

39



- **Recover differences representing R_{r-1} flips**

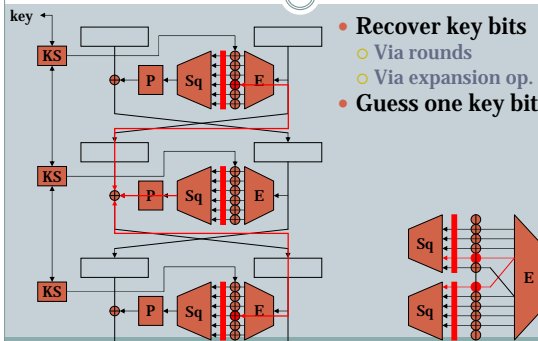
WBDES Cryptanalysis (2)

40

- **Dataflow between the rounds random, but ...**
- **Propagation of differences leaks information**
 - Find differences on the input of T-boxes, that represent flips on the internal S-boxes
- **Difference propagation of an S-boxes depends on the original input to this S-box (which was fixed when X was chosen) → (partial) recovery of input to the S-box (after key addition)**
- **Recovery of key information**

Key recovery

41



- **Recover key bits**
 - Via rounds
 - Via expansion op.
- **Guess one key bit**

Conclusions

42

- **Differential properties are difficult to hide in white-box implementations**
 - Internal encodings cannot exceed the boundaries of lookup tables
 - Implement several S-boxes together, and addition of random data paths would make it a bit harder
- **DES cryptanalysis based on properties that are very typical to Feistel ciphers**
 - Open question: can one hide these?
- **Can exploit attacks on ciphers reduced to a few rounds**

Perspectives

- **No secure whitebox implementation known today**
 - new block cipher?
 - relax definition: impossible to obtain short description
- **Relation to side-channel analysis**
 - Side-channel model gives access that is limited in time and scope (rather than unlimited)
 - But learning any information on key or I/O behavior is an attack
- **Relation to obfuscation**
 - Find obfuscators for specific programs, e.g., point functions
 - Relax definition, e.g. [Goldwasser+'07][Hofheinz+'07] (TCC)

Q&A

44

MORE INFORMATION

<http://whiteboxcrypto.com>

bart.preneel@kuleuven.be

