White-Box Cryptography

3rd ECRYPT PhD Summer School
May 2008, Crete

Wyseur Brecht
KU Leuven – ESAT/SCD – COSIC
brecht.wyseur@esat.kuleuven.be
This afternoon

14:50 – 16:00
Brecht WYSEUR

16:20 – 17:30
Oliver BILLET

• Introduction to white-box cryptography
  ○ Concept and model

• White-Box DES implementations
  ○ Construction
  ○ Cryptanalysis

• White-Box AES implementations
  ○ Construction
  ○ Cryptanalysis

• Theoretical aspects
Security Notions

3

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

- “Black-box” (oracle) security
Defeat of the Black-Box

• **Mobile Agents**
  - Mobile code, performing a task, given by its creator, without any interaction.
  - Threat: compromise of task and secret information by a server.
Defeat of the Black-Box

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

\[ E_k(M) || \text{Lic} \]
What if the end-points cannot be trusted?

Asymmetric cryptography
- Public key ... not really a threat... or is it?
- Multi-party computation (specifically designed asymmetric primitives, and protocols)
- Garbled circuits $\rightarrow$ communication overhead
- Homomorphic Encryption $\rightarrow$ rely on a trusted party for decryption of result

Symmetric cryptography
- Put keys in hardware (back to oracle access to a trusted end-point)
- Obfuscation
## Hardware solutions

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

<table>
<thead>
<tr>
<th>Pro Hardware</th>
<th>Con Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>- “easy” safe heaven</td>
<td>- Flexibility (e.g., online updating)</td>
</tr>
<tr>
<td>- Tamper resistant (up to some level)</td>
<td>- “Expensive”</td>
</tr>
<tr>
<td>- Not that easy to clone</td>
<td>- Malicious?</td>
</tr>
<tr>
<td>→ means for authenticity</td>
<td>- Side-channel analysis</td>
</tr>
</tbody>
</table>

- Tamper resistant (up to some level)
- Not that easy to clone → means for authenticity
- “easy” safe heaven
- Tamper resistant (up to some level)
- Not that easy to clone → means for authenticity
- “easy” safe heaven
Side-channel cryptanalysis

- 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 to the model?
White-Box Model

- **Threats**
  - Read memory
  - Cache attacks
  - Inserting break-points
  - Force a crash
  - Tamper code
  - Modification of internal variables
  - Dynamic analysis of the implementation
  - ...

- **Adversary’s goal**
  - Extract “key” information
White-Box vs. Black-Box

Oracle (input/output)

Time analysis
Power analysis
Electromagnetic radiation
...
Memory inspection
CPU call interception
Debugging
Reverse-engineering
Code tampering
...

Future
Side-channel Cryptalaysis

White-Box

Black-Box

Side-channel Cryptanalysis
Software Attacks

- 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

- **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 cryptographic ciphers: 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

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

[Kerins and Kursawe, WISSec 2006]
White-Box Cryptography

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

- 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
  - ∀ A, ∃ S, such that Pr[A(O(P)) = 1] − Pr[S^P = 1] < neg(n)
  - “Virtual Black-Box Property”
Towards (im)possibility results on WBC

Simulation based proofs, inspired by provable security research, and theoretical obfuscation research

BB security notions $\Rightarrow$ WB security notions?
Approach?

- Impossible ideal: implement a cipher as one lookup table
  - Huge size ($2^n \cdot n$ bits)
- But, class of encryption schemes spans only a fraction of the full space of $2^n!$ permutations (namely, $2^k$ members).
- Goal: approximation of the impossible ideal using tables of much smaller size, and make ‘internal’ information ambiguous.
- Transform into a randomized network of key-instantiated lookup tables $\rightarrow$ fixed key implementations.
Main idea of WBC

input

output

input

output
Internal Encodings

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

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

$L_F = \text{def } F \circ L_1$

In the case that $L_F$ is bijective:

$\forall L_i : \exists F_i$ such that $L_F = F_i \circ L_i$

↓

Information theoretical local security [Shannon ‘49]

$R_1 = \text{def } F \circ L_1 \circ G^{-1}$

$R_2 = \text{def } H \circ L_2 \circ F^{-1}$
 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 (enable deployment of authentication code)
External Encodings

- 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

- **Diversity**
  - Nr of possible ‘encodings’ of an implementation due to injection of randomness (nr of distinct constructions)

- **Ambiguity**
  - Number of alternative interpretations of a specific instance.
  - Instance: lookup table → “local security”
  - Instance: cipher such as DES → related keys (e.g., via the DES implementation property)
Results and Observations

- **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 ≠ security by obscurity
- **WBC as toolbox for asymmetric crypto**: public encryption key: \( \text{WB}(E_k) \); private decryption key: \( k \)
  - However, stronger security requirements (invertability)
- 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)

WB DES
Chow et al. 2002

Naked Variant
Fault injection attack
Jacob et al. 2002
Statistical attack
Link et al. 2005
Condensed impl.
Wyseur et al. 2005
Improved Variant

Encoded Variant

WB AES
Chow et al. 2002

Cryptanalysis
Wyseur et al. 2007

Cryptanalysis
Billet et al. 2004

Cryptanalysis
Goubin et al. 2007
White-Box DES Implementations

CONSTRUCTION

CRYPTANALYSIS

CONCLUSIONS
Data Encryption Standard (DES)

- Feistel structure
- 56-bit key; 64-bit input
- 16 rounds
- Preceded by *Initial Permutation*
- Round function properties
  - S-box propagation
  - P ○ E diffusion
- Expansion operation
  - “Middle bits”
T-Boxes

- Embed key information into bijective primitives
- Construction of T-boxes
  - Partial evaluation
  - Split-path encoding
  - By-pass encoding
- \( T: \text{GF}(2^n) \rightarrow \text{GF}(2^n) \)
  - Size: \( 2^n \cdot n \) bit
  - Bijective \( \rightarrow \) suitable for obtaining ‘local security’
Data Encryption Standard

[Chow et al. 2002]
Matrix Decomposition

- Transform a linear operation into a network of LUTs

\[
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
\vdots \\
y_n \\
\end{bmatrix} = \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_m \\
\end{bmatrix}
\]

- 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 et al. 2002]
• 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-dependant, randomized lookup tables.
  ○ Known structure [Kerckhoffs; no “security through obscurity”]
  ○ Link et al.: 2.25 MB
Cryptanalysis of White-Box DES Implementations

2 CRYPTOGRAPHY RESULTS

Goubin et al. SAC’07

Wyseur et al. SAC’07
Truncated differential attack

On “naked-DES”

Procedure:

- $X := IP^{-1}(L_0 || R_0)$ random
- $R'_0 := R_0 + \Delta_R$ with $\Delta_R$ 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]
Attack on “nonstandard-DES”

- Block-level analysis of IP \( o \) F
- Recovery of columns of F (by finding \( \Delta \) 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)
Diffential 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

[B. Wyseur, W. Michiels, P. Gorissen, B. Preneel; SAC’07]
• Recover differences representing $R_{r-1}$ flips
Dataflow between the rounds random, but ...

- Propagation of differences leaks information
  \[\rightarrow\] 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) \[\rightarrow\] (partial) recovery of input to the S-box (after key addition)

- Recovery of key information
Key recovery

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

- 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
- Reduced round attacks on ciphers
- DES cryptanalysis based on properties that are very typical to Feistel ciphers
  - Open question: possibility...
Q&A

MORE INFORMATION

http://whiteboxcrypto.com

brecht.wyseur@esat.kuleuven.be