White-Box Cryptography

ASCURE ACADEMY EVENT
MAY 2008, BRUSSELS

WYSEUR Brecht
KULeuven – ESAT/SCD – COSIC
Leuven, Belgium

brecht.wyseur@esat.kuleuven.be
KULeuven – ESAT/SCD - COSIC

- Katholieke Universiteit Leuven
  - Department of Electronics (ESAT)
    - Computer Security and Industrial Cryptography
      - Headed by Prof. Bart Preneel, Prof. Vincent Rijmen, and Prof. Ingrid Verbauwhede
      - 7 Post doctoral, and 38 Doctoral researchers
COSIC Research Activities

- **Cryptographic algorithms**
  - Design (AES, RIPEMD-160, MQ-IP)
  - Cryptanalysis
  - Secure implementations: hardware, software, HW/SW co-design, side-channel attacks, white-box cryptography

- **Protocols**: key establishment, anonymous communication, broadband encryption

- **Fundamental research**: boolean functions, secret sharing, algebraic curves, multiparty computation
COSIC Research Activities

Privacy & Identity Management

Security in Ubiquitous/Pervasive Systems

Software Security

Trusted Platforms and Embedded Systems

Document Security, Watermarking and Perceptual Hashing
Outline

- Cryptography: key concepts
- Towards practice: implementation issues
- Beyond the black-box
- White-Box Attack Context
- White-Box Cryptography
Cryptography

- Cryptography
  - Cryptology: design of ciphers
  - Cryptanalysis: cipher-breaking

- Kerckhoffs principle [Kerckhoffs 1883]
  - The cryptographic algorithm (code) is publicly known; A cipher is instantiated with a secret key.
  - Avoid ‘security by obscurity’
  - Research as an open process: competition, challenges, prices, conferences, reputation, ...

- Security
  - Provably-secure for unbounded adversary $\rightarrow |\text{key}| = |\text{data}|$ [Shannon]
  - Provably-secure for polynomial-time adversaries $\rightarrow P \neq NP$
  - Failure to cryptanalyze
Symmetric Cryptography

- A secret key shared by Alice and Bob
- Eve, an adversary ‘on the wire’ (→ passive adversary)
  - Goal: gain knowledge on \( m_i \) and/or \( k \); ‘fake’ \( m_i \)’s (in the case of Authenticated Encryption)
Security Notions

- Security notions in conventional cryptography
  - CPA, CCA, CCA2, IND-CPA, NM-CPA, etc.

- Adversary’s game (→ active adversary):

- 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 $\rightarrow$ compromise of DRM scheme
    - CSS, AACS, BD+, ... have been broken
Defeat of the Black-Box

- Other examples in practice:
  - Online games (e.g., World of Warcraft)
    - Adversaries have an incentive to manipulate (local) state information.
  - Cold reboot attacks*

- How to address these issues?
  - Put cryptographic primitives in hardware
  - Deploy ‘special’ software protection techniques

* [http://citp.princeton.edu/memory/, Feb 2008]
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>
Side-channel cryptanalysis

- **Defeat of hardware implementations**
  - Power analysis (SPA, DPA)
  - Timing analysis
  - Electromagnetic radiation
  - Fault injection
  - ... 

- **Prove security**
  - Create model (model the information leakage)
  - Prove security in that model (reduction prove; computational proof (bounded/unbounded adversaries); ...)
  - But... what if an adversary steps out of 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
Software Attacks

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

    ![Entropy Attack Diagram]

    key information

  - “Cold reboot” attacks on disk encryption schemes
• 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 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 can not 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
  - $c_t = (S_t(x)=0) \oplus k_{r+1}$

[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
  - \( \forall A, \exists S, \text{ such that } \Pr[A(O(P)) = 1] - \Pr[S^P = 1] < \text{neg}(n) \)
Approach?

- Impossible ideal: implement a cipher as one lookup table
  - Huge size ($\rightarrow 2^n \cdot n$ 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
$L_1, L_2 : \text{GF}(2^n) \rightarrow \text{GF}(2^n)$

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

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

$\forall L_i : \exists F_i \text{ such that } L F = F_i \circ L_i$

Information theoretical local security [Shannon '49]

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

$R_2 = \text{def} H \circ L_2 \circ F^{-1}$
Key?

- **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 [Billet et al.]
  - Hook white-box implementations into the containing application (enable deployment of authentication code)
Back to DRM
External Encodings

- Implement $G \circ E_k \circ F$, instead of $E_k$
- 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”
- 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
**State-of-the-art (constructions)**

- **WB DES**
  - Naked Variant
    - Fault injection attack
      - Jacob et al. 2002
  - Encoded Variant
    - Statistical attack
      - Link et al. 2005
    - Condensed impl.
      - Wyseur et al. 2005
  - Improved Variant
    - Cryptanalysis
      - Goubin et al. 2007

- **WB AES**
  - Naked Variant
    - Cryptanalysis
      - Billet et al. 2004
  - Encoded Variant
    - Cryptanalysis
      - Wyseur et al. 2007
State-of-the-art in constructions

Black-Box Security
- Algebraic Structure (prove strength)
- Simplicity and Kerckhoffs Principle

White-Box Security
- Algebraic Analysis
- Reduced Rounds Cryptanalysis
- Speed and Implementation Size
Applications

- Given secure white-box constructions, what can one do?
  - Software diversity
  - Essential building block for DRM applications
  - Substitute for (expensive, inflexible) hardware approaches
  - Tamper resistances (due to ‘smart’ choice of randomness injection [Michiels et al., DRM’07])
  - A new class of asymmetric constructions
    - Note: stronger requirements (invertability of network of LUTs)
  - ...

...
Conclusions

- Key information is **extremely hard** to hide in software implementations.
- Theoretically: it’s feasible* 😊, but how?
- Practical:
  - Constructions of the DES and the AES have been proposed.
  - But... broken – seems difficult to hide differential behavior and algebraic structure of cryptographic ciphers.
  - The mindset of cryptology (design of ciphers) needs to be changed.
- Applications
  - **Very** promising technology for a wide set of applications.

* [A. Saxena and B. Wyseur, in submission phase, 2008]
Q&A

MORE INFORMATION
http://whiteboxcrypto.com
brecht.wyseur@esat.kuleuven.be