It’s Not the Size of Your Keys, It’s How You Use Them

Cryptography in a White-Box World

A Presentation for the New Directions in Cryptography Workshop

Phil Eisen, Cloakware Corporation

June 27, 2008
The Cryptographer’s Dream

> Many people who become cryptographers do so for one of two reasons
  – To develop an unbreakable cipher
  – To break a well-known cipher

> Cipher cracking contests are very popular, involving thousands of people

> People continue to make machines to break DES, an already broken cipher
  – They want to break it better!
The Rules of the Game

> Everyone who takes an introductory cryptography course learns that there are rules for cipher designers, and rules for cryptanalysts

> To have a cipher design taken seriously, you must
  – Publish your algorithm in complete detail
  – Provide test vectors
  – Show that your cipher resists known attacks

> History has borne out the soundness of these rules
  – Security through obscurity doesn’t work for very long
The Rules of the Game (2)

> To break a cipher, here’s what you get:
  - Full algorithmic details
  - Access to an implementation that encrypts under the key of interest
  - The ability to pass any plaintext you want to this implementation, and to see the resulting ciphertext (adaptive chosen plaintext attacks)

> What you don’t get, however, is access to the internals of the implementation while it’s running
  - This is the black-box attack model
  - Almost all new ciphers proposed today are described and attacked under this model
Where did the cryptanalyst’s rules come from?
In secure hardware (an ever changing entity), the black-box attack model is a realistic one

- Question: when was the last time you used secure hardware?
Times Have Changed

> Software is easier (and therefore cheaper) to
  - design
  - implement (fabricate)
  - test
  - distribute
  - diversify
  - revoke
  - update
  - retire

> Overall, these factors outweigh the security considerations
Times Have Changed (2)

> 1977 – DES
  – Optimized for hardware implementations
  – Standard did not allow for software implementations until 1988

> 2000 – AES
  – Evaluation criteria explicitly discussed performance in software
  – Hardware performance was not considered until the 2\textsuperscript{nd} round

> We live in a software world
Who’s the attacker?

Alice

000110101001...

Bob

Eve

Fred

Harry

Gord

Isaac

www.stolenmusic.org
White-Box Attacks

> Let’s visit this new attack context
  – Software implementations
  – Environment is untrusted
  – Attacker has direct access to the machine while it’s running

> What’s meant by direct access? The attacker can
  – Trace every program instruction
  – View the contents of memory and cache at any granularity
  – Stop execution at any point and run an off-line process
    • Reduced round attacks are no longer theoretical
  – Alter code or memory at will
    • Fault attacks are real and trivial to execute
  – and can do all this for as long as they want, whenever they want, in collusion with as many other people as they can find
White-Box Attacks (2)

**White-box**
- Debuggers
- Emulators
- Other attack tools

**Grey Box**
- Timing analysis
- Power analysis
- Fault injection

**Black-box**
- Input / output
> Recall the *always double and add* method described by Prof. Miri as a defence against side channel attacks on elliptic curve scalar multiplication

**Input:** $d = d_1d_2...d_n$ (the scalar), $P$ (the elliptic curve point)

**Output:** $Q = dP$ (another elliptic curve point)

1. $Q = P$
2. for $i$ from 2 to $n$
   - $T_1 = 2Q$
   - $T_2 = T_1 + P$
   - if $(d_i = 1)$
     - $Q = T_2$
   - else
     - $Q = T_1$
3. return $Q$
Attacking ECC (2)

> The black-box attacker sees only d, P and dP
  – Always double and add is overkill in this case

> The grey-box attacker sees

\[ \begin{align*}
  & \text{if } (d_i = 1) \\
  & \quad Q = T_2 \\
  & \text{else} \\
  & \quad Q = T_1
\end{align*} \]

  – A consistent power trace leaks no information

> The white-box attacker sees

  – They can trace the execution and extract the key
White-Box Attacks (3)

> The security proofs from the black-box attack context simply don’t carry over to the white-box context
  - NB: the proofs are not invalid, they just consider a different attack model

> We are now forced to consider a white-box attacker; they are strictly more powerful than our classic black-box attacker
White-Box Cryptography

> A short-form for cryptographic implementations that provide security against a white-box attacker

> Even more so than with side-channel attacks, the *implementation* becomes as important as the algorithm itself
White-Box Cryptography (2)

> This is still a relatively untapped field, with a lot of fundamental unanswered questions

- What is a formal definition for the white-box attack context?
- What’s meant by “security” in a white-box attack context?
  - What are we trying to defend? For how long?
- Is practical white-box cryptography possible?
  - This almost certainly depends on answers to the first two questions
- Are existing algorithms, designed for the black-box attack context only, a good starting point, or should we start from scratch?
White-Box Cryptography and Obfuscation

> There are several models of obfuscation, but all involve the hiding of certain properties of a program.

> The value of the key is one such (very important) property.

> Thus, if we could create an obfuscator, we could apply it to cryptographic algorithms and increase security against white-box attackers.
Some Results

> We do know that it’s possible to implement a cipher in such a way that the best attack is a black-box attack

> Consider AES, with key K

  – It can be described as a function that takes a 128-bit input and produces a 128-bit output
  
  – Such a function can be “implemented” as a lookup table with $2^{128}$ entries
  
  – Such an implementation has no internals, so it can only be attacked as a black box

> Obviously, this isn’t practical

> Open question: can we do any better?
Barak et al – “On the (Im)possibility of Obfuscating Programs

- Proposed a definition for an obfuscator, and showed that there existed contrived programs that could not be obfuscated under this model
- No claims made regarding the obfuscatability of programs in general
- Their result applies equally well to hardware implementations, so doesn’t quite match the real world
Other models for obfuscation:

- Canetti et al (2008) showed that it is possible to obfuscate point functions under their model
- Hohenberger et al (2007) were able to obfuscate re-encryption under a security-oriented model
- Goldwasser et al (2008) introduced *best-possible obfuscation*, with various positive and negative results
Some Results (4)

> Proposed implementations of AES:
  
    • Presented the first implementation of AES that took white-box attacks into account
  
    • An attack on the Chow et al implementation
  
    • Another attack
Proposed implementations of DES:

  - The first implementation of DES that took white-box attacks into account
- Jacob et al (2002), “Attacking an Obfuscated Cipher by Injecting Faults”
  - An attack on one variant of white-box DES proposed by Chow et al
  - An improved implementation
  - Powerful attacks on the Chow et al and Link et al implementations
What’s Next?

> A “white-box friendly” cipher design
  - Design a cipher from the ground up to be secure in a white-box attack context
  - This would require both a cipher design, with demonstrable black-box security properties, and a description of a white-box implementation
Conclusions

> The model we have used for analyzing ciphers needs updating
> Software implementations and legitimate users as attackers push us towards a white-box attack context
> The implementation of a cipher is as important as the cipher itself
> There is a ton of opportunity to do seminal work in white-box cryptography
Contact information

Phil Eisen
Senior Cryptomathematician
phil.eisen@cloakware.com

Cloakware Corporation
84 Hines Road, Suite 300
Ottawa, ON, Canada
K2K 3G3
Tel: +1 613.271.9446

Cloakware Ltd.
33-35 Daws Lane
London NW7 4SD
United Kingdom
Tel: +44 (0) 1189.340940

Cloakware Inc.
8320 Old Courthouse Road
Suite 201
Vienna, VA, U.S.A.
22182
Tel: +1 703.847.3611

www.cloakware.com