6.857: Computer and Network Security (Spring 2009)

Guest lecturer: Eran Tromer

Lecture 6: Side-channel attacks

February 23, 2009

Traditional attacks

- Bad specifications
- Insecure algorithm
- Implementation bugs
- Hardware intrusion
- Software intrusion

Inadvertent information flows

- Side channels
- Covert channels
  - Violate Mandatory Access Controls
  - Avoiding detection: steganography
- Violation of standard "platform stack" abstraction
  - Across machines
  - Across processes
  - Across chroot "jails"
  - Across virtual machines (e.g., patent 6922774)

Channels

- Electromagnetic (TEMPEST) [Kuhn 2003]
  - CRT monitors and LCD monitors [Kuhn 2004]
  - CPUs, smartcards
  - Keyboard [Vuagnoux Pasini 2008, `http://lasecwww.epfl.ch/keyboard`]

- Power
  - Smartcards
  - RFID (via EM backscatter) [Oren Shamir 2006, `http://www.wisdom.weizmann.ac.il/~yossio/rfid`]
- Timing [Kocher 1996]
  - Branches
    - Modular exponentiation via square-and-multiply
      $c^d \bmod n = \{x \leftarrow 1;\ \text{for } i\text{=1023,...,0: } \{x \leftarrow x^2 \bmod \text{n; if } d_i = 1{:}x \leftarrow x \cdot c \bmod n\ \}\}$
    - Long-integer multiplication: plain vs. Karatsuba
  - CPU ops whose timing depends on operands (e.g., shifts, multiplications, division)
  - S-box access cache collisions (see below)
  - Protocol-level (SSL)
  - Incoming input (e.g., ssh keystrokes) [Song Wagner Tian 2001]
  - Local (same machine) or remote (over a network)
- Diffuse visible light from CRT screens [Kuhn 2002]
- Acoustic
  - CPUs [Shamir Tromer 2004, `http://people.csail.mit.edu/tromer/acoustic`]
  - Keyboards [Asonov Agrawal 2004]
  - Printers
- Cache
  - Shared resource across local processes. "Protected memory" is for data; this attacks *meta*data (addresses).
  - Observation methods:
    - Power trace [Page 2002]
    - Covert channels [Hu 1991]
    - Collision (timing) [Lauradoux 2005][Bonneau Mironov 2006]
    - Eviction as input (timing) [Bernstein 06][Shamir Tromer Osvik 2006]
    - Eviction as output (prime+probe) [Shamir Tromer Osvik 2006][Percival 2006]
- Other microarchitectural channels
  - Instruction cache / trace cache [Aciicmez 2007]
  - Branch prediction [Aciicmez Schindler Koc 2006]
  - Functional units (e.g., floating-point multiplier) [Aciicmez Seifert 2007]
- Faults survey: [Bar-El Choukri Nacacche Runstall Whelan 2004]

- Triggers: EM, power, clock skew, neutrons, camera flash , luck [Skorobogatov Anderson 2002]
- RSA via Chinese Remainder Theorem
- Differential Fault Analysis of arbitrary ciphers [Biham Shamir 1996]
- Single memory error suffices to break out of Java VM [Govindavajhala Appel 2003]

- Multi-spectral / multi-modal (e.g., power+timing)

Analysis

- Simple (power) analysis
  - Observe a single trace (e.g., current low vs. high→ bit is 0 or 1)
- Differential (power) analysis [Kocher Jaffe Jun 1999]
  - Focus on one key-dependent value
  - Build a model of the device given known input and (partial) key
  - To test a key hypothesis: compare model to measurements for many different inputs, to average away noise
  - "High-order" variant: compare multiple points in time/space
- Template attack
- Stochastic model

Countermeasures

- Goals
  - Preserves functionality
  - Secure
  - Efficient
  - Generic
- Degrading the channel (Faraday cages, opaque partitions, sound mufflers, power filters...)
- Degrading the signal by injecting noise (randomizing delays, timing, power, memory accesses..)
- Eliminating the signal by making it deterministic or random (more generally: key-independent)
  - Eliminate branches

  - ∗ Exponentiation:
    c^d mod n={$x \leftarrow 0$; for i=1023,...,0: {$x \leftarrow x^2 \cdot (dc + (1-d))$ mod n)} (33% worse).
  - ∗ Elliptic curve formulas
- – Cache access normalization
- – Bitslicing

- Program obfuscation
  - Virtual black box: any circuit C is transformed into C' such that anything you can efficiently compute by looking at C' could also be efficiently computed given just black-box access to C.
  - Extremely powerful [Hofheinz Malone-Lee Stam 2006]
    - ∗ Private key encryption ⟶ public key encryption
    - ∗ MACs ⟶ electronic signatures
  - Known obfuscators: just a few extremely simple cases (e.g., point functions) [Canetti 1997]
  - Generic obfuscation is impossible [Barak Goldreich Impagliazzo Rudich Sahai Vadhan Yang 2001]
  - Heuristic
    - ∗ "Jumble" code by stripping identifiers, moving code/data around, randomly choosing equivalent sequence, etc.
    - ∗ Typically broken manually or by "decompiler" tools

— The following was not covered in class —

- Oblivious RAM — a compiler such that adversary can't distinguish real execution from that of a fake CPU which run an idle loop for the same duration and magically outputs the same. [Goldreich Ostrovsky 1995]
- Encoding for leakage reduction and error detection
  - Leakage-resistant logic and fault-resistant logic styles (e.g., balanced)
  - Masking
- Cryptographic transformations and models
  - Security against bounded-#wired measurements [Ishai Sahai Wagner 2003]
  - Side-channel-aware reductions [Micali Reyzin 03]
  - "Algorithmic Tamper-Proof ": part tamper-proof , part secret [Gennaro Lysyanskaya Malkin Micali Rabin 04]
  - Stream cipher assuming half-readable memory [Dziembowski Pietrzak 2008]

For a survey of many of these topics, see Chapter 17 in Ross Anderson, *Security Engineering*, 2nd ed., Wiley, 2008.