SIDE CHANNEL ATTACKS AGAINST IOS CRYPTO LIBRARIES AND MORE

DR. NAJWA AARAJ

HACK IN THE BOX
13 APRIL 2017
WHAT WE DO

What we do

• Robust and Efficient Cryptographic Protocols
• Research in Cryptography and Cryptographic Implementations
• Key Management and Trusted Platforms Research
• Operating Systems Security Research
• Frameworks for Secure Communication and Practical Encryption

Our aim

• Secure and Authenticated Communication Channels
• Data at Rest and Data in Transit Security
• Operating Systems and Software Stack Security
• Hardware Security
Secure protocols are needed...

- Most Secure Algorithms
- Robust Authentication and Localized PKI
- Perfect Forward and Future Secrecy
- Anonymity vs. Non-Repudiation

Diagram:
1. Authentication
2. Key Exchange Using Diffie Hellman
3. Encrypt messages with AES Keys and send over TLS
4. Hash messages and generate digests
5. Authentication
6. Verify message digests
7. Decrypt messages

Public Key Crypto × Private Key Crypto × Hashing

Public Key Crypto
Private Key Crypto
Hashing
... SO IS KERNEL AND HARDWARE LEVEL SECURITY

- Data at Rest Encryption
- Real-time Integrity Monitor
- Hardened Cryptographic Library
- Hardware-Rooted Key Management
- Hardened OS and Kernel

- Secure Boot Loader
- Hardware-Based Root of Trust
CRYPTO POSES CHALLENGES FOR COMMUNICATION SOLUTIONS

• Encryption should be intractable by theoretical (Crypt)-analysis
  • E.g., Intractable properties of factoring large numbers

• Systems can be broken by inadequate bit-level security and crypto parameters

• Inadequate Root of Trust for the Cryptographic System

• Systems can be weakened by inappropriate implementations
  • E.g., side-channel attacks; memory attacks

“By Attacking a Crypto Device, the adversary hopes to subvert its security correctness properties by extracting some secret”
SIDE CHANNEL ANALYSIS

PlainText $\rightarrow$ Crypto Algorithm $\rightarrow$ CipherText

Keystream

Computations Timing
Power Consumption
EM Radiation
Temperature Variations
• Power consumption depends on computation performed
• Point multiplication (K*Q) occur during key and signature generation (Double-and-add algorithm)
• double and add are very different ops (key bit ==1)
• Simple Power Analysis (SPA): Leakage during point multiplication leaks information about secret (K)

PRACTICAL EXAMPLES

• Power Analysis on Smart Cards – Retrieval of smart card PINs
• Smart Phones – Leakage of keys used to secure communications, financial transactions, SSL traffic, data stored on phones

Most Common Side Channel Attacks

• **Power analysis** – this type of side channel attacks has been demonstrated in many cryptosystems; emanate from circuits which typically consume differing amounts of power based on operations and data

• **EM** – Electromagnetic emissions and signals (e.g., through near-field inductive and capacitive coupling, or far field antennas)

• **Timing** – Data-dependent differences in process time and delay in cryptographic operations based on data input and relevant operation.
**TYPICAL TARGETS**

- **Most Common Targets:**
  - Smart Cards / FPGAs - Microcontrollers/ Custom Processors

- **Other Common Targets:**
  - Smart Phones / Embedded Devices / CPU Boards
  - USBs / Consoles /

- **Modern Targets**
  - Charging patterns of devices
  - Interfaces of software / hardware co-design interactions
  - Servers

- **When interception is feasible (active / passive), simple or differential side channel attacks will aim at breaking modern crypto**
EXPERIMENTAL SET-UP

Signal Acquisition
- Signal Acquisition Card, triggered automatically

For FPGA-based implementations
- Smart Card Reader based on an FPGA controller
- Reader output antenna amplified and feeding into input channel of Signal Acquisition Card

For Mobile Applications
- Near Field Probe set

* Equipment shown is part of CRI DPAWS toolkit
EXPERIMENT 1: RSA SIGNATURE USING CRT ATTACK DESCRIPTION

Signature = \( s = m^d \mod n \)  
\( d = \text{private exponent}, p \) and \( q \) are primes, \( m \) is the message to sign

A CRT implementation of Signature is as follows:

**Pre-computation**

\[ d_p = d \mod (p-1) \]
\[ d_q = d \mod (q-1) \]
\[ k = p^{-1} \mod q \]

**Reduction**

\[ m_p = m \mod p \]
\[ m_q = m \mod q \]

**Exponentiation**

\[ S_p = m_p^{d_p} \mod p \]
\[ S_q = m_q^{d_q} \mod q \]

**Recombination**

\[ s = ( ( (S_q - S_p) \times K ) \mod q ) \times p + S_p \]
**EXPERIMENT 1: ACQUIRING SIGNALS FROM THE SMART CARD**

**Signal Acquisition at a Frequency of 1.5MHZ (0.6 sec trace size)**

- We capture signal in the time domain
- Calculate Fast Fourier Transform
- **FFT Frequency bin at which frequency peaks is our crypto signal**
- We apply a **band pass filter** to isolate crypto traces (bandpass center frequency = $F = 0.625$Mhz and bandwidth = $2F$)

* Software used is part of CRI DPAWS toolkit
EXPERIMENT 1: SIMPLE POWER ATTACK AGAINST RSA USING CRT

Initial Reduction

\[ m_p = m \mod p \]
\[ m_q = m \mod q \]

\[ S_p = m_p^{d_p} \mod p \]
\[ S_q = m_q^{d_q} \mod q \]

- P and Q half the size of n (2048 / 2)
- CRT (RSA) operation, the exponentiation with dp uses the binary algorithm (SPA by timing differences between multiplication and squaring)

Simple Power analysis enables us to recover all RSA 2048 bits – 2 days to capture signal on one card. Automated scripting for remaining cards
EXPERIMENT 1: FINDING THE SIGN. SIGNAL – RSA SIGNATURE (IOS) USING OPENSSL

- SPA is not effective anymore
- Lots of signal processing to reduce noise and non-crypto signals
- We resort to differential Power Analysis on modular reduction to recover prime $p$

$$m_p = m \mod p$$
$$m_q = m \mod q$$

We can discover the bits of $p$ sequentially (1024); fix $p$ to 1 or 0, calculate intermediate traces and confirm bit guessing if curves average is not NULL
EXPERIMENT 1: SPA AND DPA FAILURE - RSA SIGNATURE (IOS) USING IOS APIS

- Uses iOS Unit Test using Apple APIs
- SPA and DPA are not effective in capturing CRT or non-CRT RSA operations
- Next Steps include Locating the CPU by setting the CPU to a constant clock frequency and De-capsulating the back cover chip
EXPERIMENT 2: AES ENCRYPTION ATTACK DESCRIPTION

No diffusion in Last Round
C = XOR(Round Key, Value)

If final round key is extracted, intermediate Keys can be recovered by reversing Key schedule (and S-Box\(^{-1}\))

Attack one byte at a time: \(2^{8} \times 32\)
For each 2\(^8\) combination, guess of Round Key is correct when output traces are correlated to guessed data
EXPERIMENT 2: FINDING THE AES ENCRYPTION SIGNAL

14 Round; We start Here
EXPERIMENT 2: DIFFERENTIAL POWER ATTACK AGAINST AES LAST ROUND

- No difference of power consumption across rounds
- However, the following is to be observed:
  - Last round does not have a Mix Column operation
  - Byte Substitution (S-box) operated on each byte separately
- Given storage structure, an intermediate bit will be chosen for analysis and keys bits will be guessed one byte at a time

- Number of equations to solve:
  - In one round: $2^8 \times 32 = 8,192$
  - Total Traces: 114,688,000
EXPERIMENT 2: DPA – AES ENCRYPTION (IOS) USING OPENSSL

- Same attack apply
- Same linear shift registers structure is used to store intermediate values
- Except iOS makes it harder to capture signals and isolate crypto frequencies
WHAT MADE IT POSSIBLE

• For **RSA implementation on Smart card**, Simple Power Analysis was possible due to:
  - No hardening or countermeasures in FPGA-based implementation
  - Non constant time operations when exponent bit is 0 vs. exponent bit is 1

• For **RSA implementation on iOS application using custom OpenSSL implementation**
  - Simple Power Analysis was not possible given OpenSSL hardening of CRT
  - iOS does not make it easy to capture signals from the phone
  - Differential Power Analysis was rendered possible because Application Master Key was constantly
    used instead of using ephemeral keys for One time signatures (OTS) that are derived from Master
    Key using a KDF

• For **AES implementation [Card + iOS]**
  - No countermeasures against differential power analysis
  - Differential Power analysis rendered possible because of Lack of Key Agility
COUNTERMEASURES

Algorithm-level Countermeasures

- Randomness (masking / blinding)
- Constant Time implementations
- Pre-computations and Leak Reduction
- Noise based countermeasures
- Increase dependencies on Boolean ops (e.g. keccak)
- Randomize in-algorithm structures between rounds
Protocol-level Countermeasures:

- Reduce the amount of leakage to less than the minimum required for key(s) recovery using SPA / DPA / EM-based leakage
- Reduce interim states that could lead to leakage
- Key Agility (per session / per call / per message)
- Layered Security
- Increased Overall bit level security
- Redundant crypto operations to reduce leakage and temp values
- Smart choice of the cryptosystem
NEXT STEPS – WORK WITH CRYPTO THAT USES NATIVE IOS APIS

- Locate the CPU by setting the CPU to a constant clock frequency
- De-capulate the back cover chip

**Set-up 1:**

- Use the Probing Set-up to scan surface of the phone
- Try to capture relevant signals to recover keys and understand practicality of such an attack

**Set-up 2:**

- Use the Probing Set-up to intercept interfaces to secure enclave through dedicated hardware buses
THANK YOU