Information Security – Theory vs. Reality

0368-4474, Winter 2015-2016

Lecture 3:
Power analysis, correlation power analysis

Lecturer:
Eran Tromer
Power Analysis

Simple Power Analysis
Correlation Power Analysis
Differential Power Analysis
Power analysis

- Power analysis: measure device’s power consumption.
The AES Cipher

Plaintext -> AES -> Ciphertext

Key
AES symmetric cipher: Lookup-based implementation

```c
char p[16], k[16];                     // plaintext and key
int32 Col[4];                          // intermediate state
const int32 T0[256],T1[256],T2[256],T3[256]; // lookup tables
...

/* Round 1 */
...
```
AES symmetric cipher: “Algebraic” implementation

AES symmetric cipher: “Algebraic” implementation in code

```c
void AESEncrypt( u8 input[16], u8 output[16] ) {
    [...]  
    for (r=1; r<=9; r++)
    {
        ByteSub(state);
        ShiftRow(state);
        MixColumn(state);
        KeyAdd(state, roundKeys, r);
    }
    [...] 

Source: http://users.ece.utexas.edu/~gerstl/ee382v-ics_f09/soc/tutorials/System_C_Code_Examples_2/date04_examples/cosimulate/sw_only/```
Theory of power analysis

- Power consumption is **variable**
- Power consumption depends on **instruction**
- Power consumption depends on **data**
The power consumption of a CMOS gate depends on the data:
- q: 0->0  almost no power consumption
- q: 1->1  almost no power consumption
- q: 0->1  high power consumption (proportional to C2)
- q: 1->0  high power consumption (proportional to C1)
Data-dependent power consumption: wire capacitance

Capacitance of last cells in this inverter, outgoing wire, and first cells in next gate

Source: DPA Book
Power Depends on Instruction, Locally

LCALL SET_ROUND_TRIGGER
MOV A,ASM_input + 0 ; load a0
XRL A,ASM_key + 0 ; add k0
MOVC A,@A + DPTR ; S-box look-up
MOV ASM_input, A ; store a0
LCALL CLEAR_ROUND_TRIGGER
Power Depends on Instruction, Locally

Source: DPA Book
Power Depends on Instruction, Globally

Example: AES

[Joye Olivier 2011]
Power Depends on Data

Source: DPA Book
Power Analysis

- Simple Power Analysis
- Differential Power Analysis
  - Warm-up Correlation Power Analysis
  - Full Correlation Power Analysis
Plaintexts and ciphertexts may be chosen, known or unknown.
Theory of power analysis

- Power consumption is **variable**
- Power consumption depends on **instruction**
- Power consumption depends on **data**
Simple Power Analysis (SPA)

Example: square-and-multiply RSA exponentiation.

- **Pros:**
  - Single trace (or average of a few) may suffice
- **Cons:**
  - Detailed reverse engineering
  - Long manual part
  - Hard to handle bad signal-to-noise ratio, especially for small events, (e.g., AES)
Differential Power Analysis (DPA)

- Use statistical properties of traces to recover key
- Pros:
  - Very limited reverse engineering
  - Harder to confuse
- Cons:
  - Large amount of traces
- Two main types of DPA:
  - Difference of means (traditional DPA)
  - Correlation power analysis (CPA)
Mean, variance, correlation

Low Variance ⇐
High Variance

Low Correlation ⇐
High Correlation

攻坚战
CPA Basics

- We want to discover the correct key value ($c_k$) and when it is used ($c_t$)

- Idea:
  - On the correct time, the power consumption of all traces is correlated with the correct key
  - On other times and other keys the traces should show low correlation
Warm-up CPA

- Assume plaintext and correct key are known but correct time is unknown
- Form hypothesis and test it
- Good hypothesis function \( f(p, k) \)
  - Depends on known plaintext and
  - Depends on small amount of key bits
  - Deterministic
  - Sufficiently random (e.g., non-linear), sensitive to small changes in \( p \) and \( k \)
- Maps to power consumption using a power consumption model
Warm-up CPA in Numbers

- 1000 traces, each consisting of 1 million points
- Each trace uses a different known plaintext – 1000 plaintexts
- 1 known key
- Hypothesis is vector of 1000 hypothetical power values
- Output of warm-up CPA: vector of 1 million correlation values with peak at $c_t$
Warm-up CPA in Pictures
Full CPA

- Plaintext is known, but correct key and correct time unknown
- Idea: run warm-up CPA many times in parallel
- Create many competing hypotheses
Full CPA in Numbers

- **1000 traces**, each consisting of **1 million points**
- Each trace uses a different known plaintext – **1000 plaintexts**
- Key is unknown – **256 guesses** for first byte
- Hypothesis is matrix of **1000X256** hypothetical power values
- Output of full CPA: matrix of **1,000,000X256 correlation values** with peak at \((c_k, c_t)\)
Full CPA in Pictures

corr_traces = result(dpa_obj);

(5x10) matrix 'traces'

Trace 1
Trace 2
Trace 3
Trace 4
Trace 5

(5x3) matrix 'hypotheses'

(3x10) matrix 'corr_traces'

Hypothesis 1
Hypothesis 2
Hypothesis 3
Full attack

- Acquire traces
- For every key fragment and corresponding hypothesis function:
  - Compute matrix of hypotheses
  - Compute correlation (score) matrix
  - Find maximum in correlation matrix and deduce value of key fragment
Discussion: effects and tradeoffs of parameters

- Number of traces
- Trace length
- Number of hypotheses (i.e., number of relevant key bits affecting the hidden state being modeled)
Assumptions/complications/mitigation

ASSUMPTIONS/BOTTLENECKS

- Alignment (shifts, fragmentation)
  - Handling algorithmically (e.g., cross correlation, string alignment)
  - Physical synchronization via triggering
- Data size: #traces, trace length
- Assuming fixed over all traces:
  - Key
  - Computation progress and flow
- Measurement quality
  - Physical access, noise, transmission
  - Equipment cost
  - Expertise

MITIGATIONS

- Timing
  - Unstable clock
  - Randomize control/instruction flow
  - Insert random/key/plaintext-dependent delays
- Change protocol to roll keys often
- Add power noise

Example: probing an “decoupling capacitor” (SMD 0402) close to the microcontroller chip. [O’Flynn 2013]
Further Reading

http://www.dpabook.org
http://www.springerlink.com/content/go1q1k