Side Channel Attacks -- Power Analysis

Cybersecurity Specialization
-- Hardware Security

Simple Power Analysis (SPA)

- Visual examination of graphs of the current used by a device over time to deduce information about data/operation.
  - Variations in power consumption occur as the device performs different operations or input.
  - Oscilloscopes can show the data-induced variations.
  - Frequency filters and averaging functions are used to filter out high-frequency components.
- Measuring power/current
  - Simple: e.g. read from terminal in smart cards.
  - Equipment: relatively inexpensive, high precision

Square and Multiply Algorithm

- Goal: Compute a^e (mod n)
  1. convert e to binary: $k_s k_{s-1} \ldots k_1 k_0$
  2. b = 1;
  3. for (i=s; i>=o; i--)
  4. { b = b*b (mod n);
  5. if (k_i == 1)
  6. b = b * a (mod n)
  7. }
  8. return b;

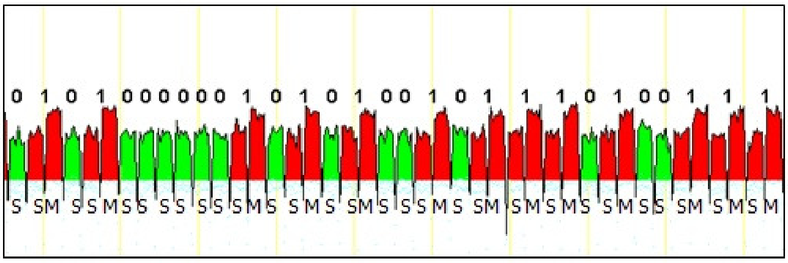
Side channel attacks!

Observable side channel info during hardware execution: current, power, timing, …

The value of bit k_i determines whether this non-trivial operation will be required.

SPA on Modular Exponentiation

- There are fast implementations for the square operation, so it takes less time and power to compute b^2 than b*a.



SECRET EXPONENT

S = Square

M = Multiply

http://www.eetimes.com/document.asp?doc_id=1278081

SPA Features and Variations

- Directly deduces information (key, round, etc.) from power/current trace
- Relies on small number of traces during normal execution (high accuracy required/preferred)
- Needs precise understanding of the crypto algorithm/protocol and its implementation
- Non-invasive, no trace of attack
- Passive, but can be more effective with
  - control of the normal execution
  - fault injection to cause abnormal execution

  in order to obtain and verify the deduced information

Differential Power Analysis (DPA)

- Procedure
  - Collect a large amount of power/current waveforms with a scope
  - Build a model or hypothesis about the secret key or information
  - Apply (advanced) data processing methods (e.g. hypothesis test) to reveal the key
- DPA can be performed in any algorithm that has the operation $\beta=S(\alpha\oplus K)$

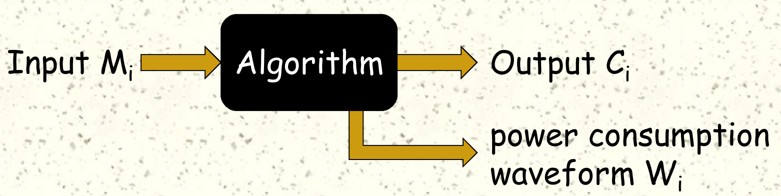
  where α is known and K is the segment key

DPA Pros and Cons

- Requirements
  - Need to know the crypto algorithm/protocol under attack
  - Needs a large amount of power traces (this implies that the attack needs to have control of the device for some time)
  - Some tools/skills on statistical analysis
- Advantages
  - No need to know implementation details
  - No need to have accurate traces

DPA: Data Collection

- For an input data M_i
- Run the algorithm to obtain output C_i
- Measure the power consumption W_i
- Repeat this N times



DPA: Data Partition & DPA Value

- (M_i, C_i, W_i): i=1, 2, …, N
- Assume that the algorithm performs a known function f
- Compute $L_i = f(M_i) = L_{i1}L_{i2}\ldots$ for a key K
- Select a bit position j in L_i
- Data partition
  - $S_0 = \{(M_i, C_i, W_i): L_{ij}=0\}$
  - $S_1 = \{(M_i, C_i, W_i): L_{ij}=1\}$
- DPA value calculation:

$$\Delta = \frac{\sum_{w_i \in S_0} w_i}{|S_0|} - \frac{\sum_{w_i \in S_1} w_i}{|S_1|}$$

DPA: Hypothesis Testing

- If an incorrect K is used
  - "independent" L_{ij} and C_{ij}
  - $\frac{\sum_{w_i \in S_0} w_i}{|S_0|} \approx \frac{\sum_{w_i \in S_1} w_i}{|S_1|}$
  - DPA Δ is close to 0
- If the correct K is used
  - $L_{ij}= C_{ij}$
  - Δ = average (0) – average(1)
  - a peak value in DPA

$$\Delta = \frac{\sum_{w_i \in S_0} w_i}{|S_0|} - \frac{\sum_{w_i \in S_1} w_i}{|S_1|}$$

DPA Example: Break a Key

- AES: 128-bit key, 16 bytes K_i (i=0, …, 15)
- For each K_i, test all 256 possible values
- A total of 256*16 = 4096 DPAs (<< 2^{128})