It was right under your nose

2017.09.26

SICADA(**S**ide **C**hannel **A**nalysis **D**esign **A**cademy)

Dept. of Information Security, Cryptology, and Mathematics, Kookmin University

Bo-Yeon Sim and Dong-Guk Han

Previously proposed attacks on PKCs were based on

- **the patterns of data-dependent conditional branches**

Algorithm. Left to Right Binary Method	
INPUT	$M, N, k = (k_{n-1}, k_{n-2}, \cdots, k_0)_2$
OUTPUT	$M^k \; mod \; N$

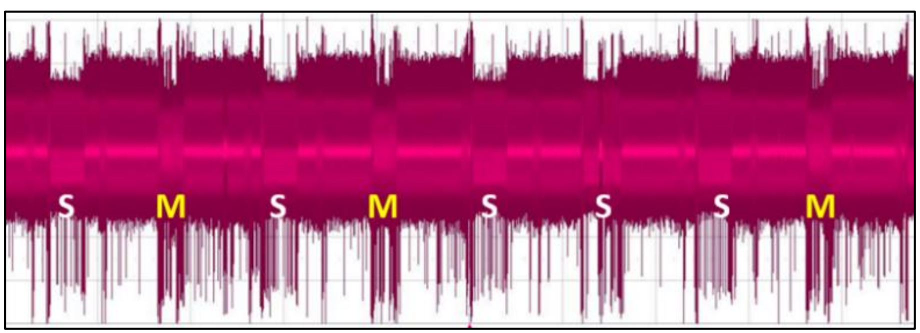
Step 1. $R = 1$

Step 2. For $i = n - 1$ down to 0 do

  2.1. $R = R \times R \; mod \; N$

  2.2. IF $k_i = 1$ then $R = R \times M \;\; mod \; N$

Step3. Return R



Countermeasure ➔ make it regular

1996 Timing Attacks

1998 Simple Power Analysis

Mathematical proof

Previously proposed attacks on PKCs were based on

- **statistical characteristic according to intermediate values**

Algorithm. Left to Right Square and Multiply Always	
INPUT	$M, N, k = (k_{n-1}, k_{n-2}, \cdots, k_0)_2$
OUTPUT	$M^k \ mod\ N$

Step 1. $R_0 = 1$

Step 2. For $i = n - 1$ down to 0 do

2.1. $R_0 = R_0 \times R_0 \ mod\ N$

2.2. $R_{1-k_i} = R_0 \times M \ mod\ N$

Step4. Return R_0

n traces

Countermeasure ➔ apply a randomization method

1996 Timing Attacks

1998 Simple Power Analysis

Differential Power Analysis

SPA Countermeasure

Mathematical proof

Previously proposed attacks on PKCs were based on

- the interrelationship between data, and etc.

Algorithm. Left to Right Square and Multiply Always	
INPUT	$M, N, k = (k_{n-1}, k_{n-2}, \cdots, k_0)_2$
OUTPUT	$M^k \ mod \ N$

Step 1. $R_0 = 1$

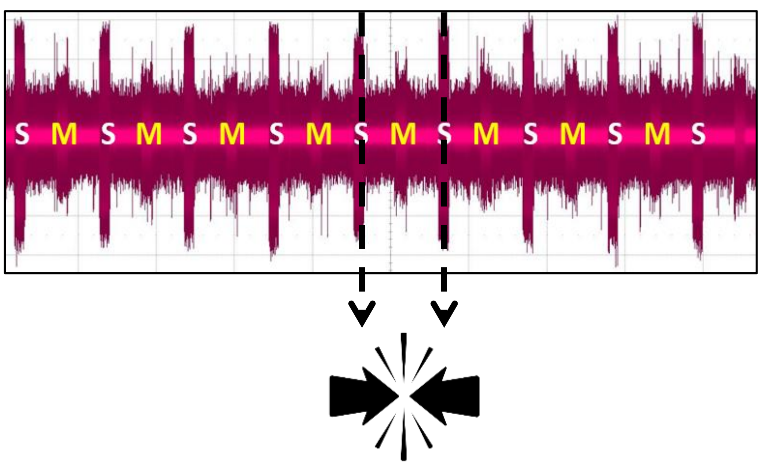
Step 2. For $i = n - 1$ down to 0 do

2.1. $R_0 = R_0 \times R_0 \ mod \ N$

2.2. $R_{1-k_i} = R_0 \times M \ mod \ N$

Step4. Return R_0

+ data / exponent blinding



1996 Timing Attacks

1998 Simple Power Analysis

Differential Power Analysis

2001 Big Mac Attack

Various Collision Attacks

Various Template Attacks

DPA Countermeasure

SPA Countermeasure

Mathematical proof

Previously proposed attacks on PKCs were based on

Algorithm. Left to Right Square and Multiply Always	
INPUT	$M, N, d = (d_{n-1}, d_{n-2}, \cdots, d_0)_2$
OUTPUT	$M^d \ mod\ N$

Step 1. $R_0 = 1$

Step 2. For $i = n-1$ down to 0 do

2.1. $R_0 = R_0 \times R_0 \ mod\ N$

2.2. $R_{1-d_i} = R_0 \times M \ mod\ N$

Step4. Return R_0

various countermeasures have been proposed

DPA Countermeasure

DPA Countermeasure

SPA Countermeasure

Mathematical proof

Algorithm. Left to Right Square and Multiply Always	
INPUT	$M, N, d = (d_{n-1}, d_{n-2}, \cdots, d_0)_2$
OUTPUT	$M^d \ mod\ N$

Step 1. $R_0 = 1$

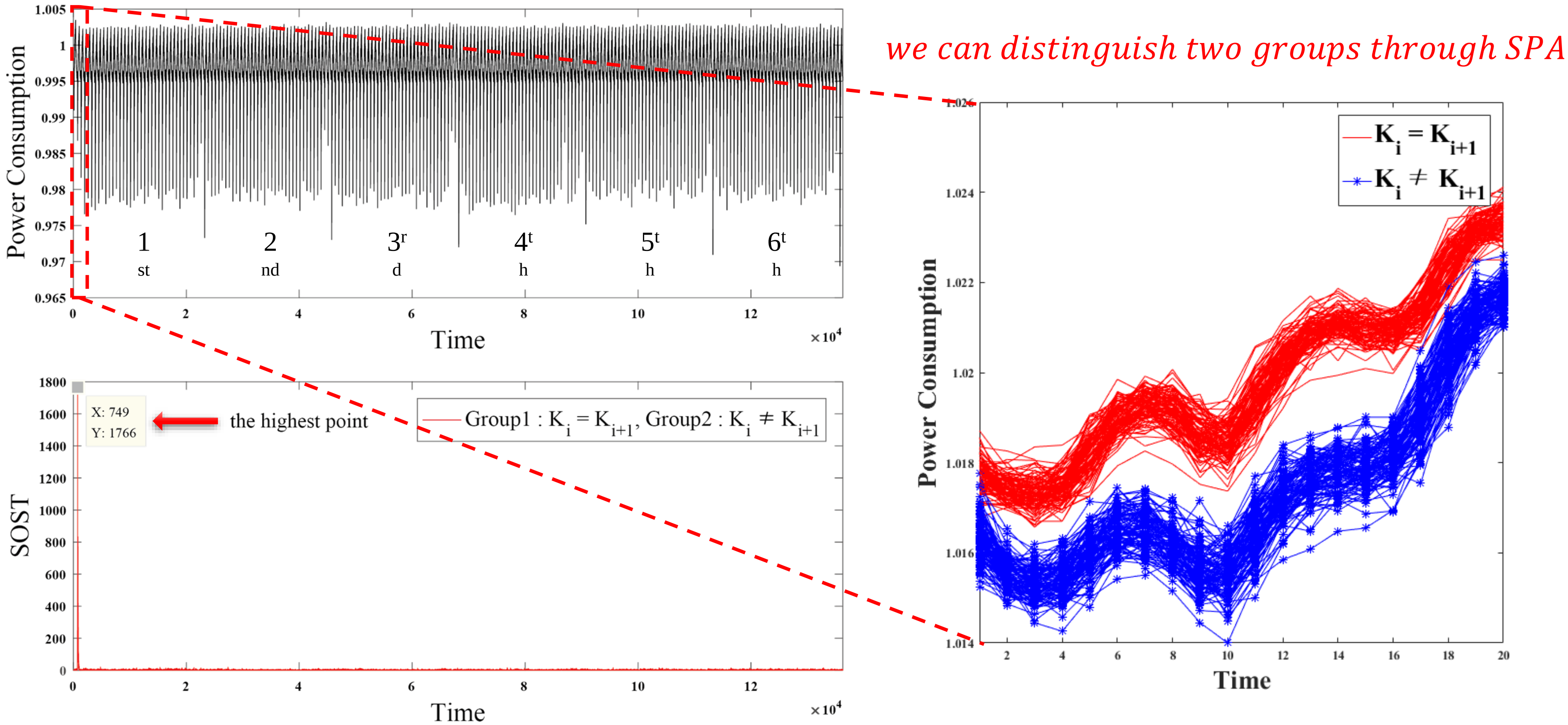
Step 2. For $i = n-1$ down to 0 do

Countermeasure

Step4. Return R_0

Do you think is it secure?

Attack on Protected PKC using a Single Trace



The attack does not require sophisticated pre-processing

such as decapsulation, localization, multi-probe, and principle component analysis

The power consumption is related to the k_i value

Algorithm. Left to Right Square and Multiply Always	
INPUT	$M, N, k = (k_{n-1}, k_{n-2}, \cdots, k_0)_2$
OUTPUT	$M^k \ mod \ N$
Step 1. $R_0 = 1$	
Step 2. For $i = n-1$ down to 0 do	
2.1. $R_0 = R_0 \times R_0 \ mod \ N$	
2.2. $R_{1-k_i} = R_0 \times M \ mod \ N$	
Step4. Return R_0	**+ data / exponent blinding**

$$\boldsymbol{k = (k_{n-1} k_{n-2} \cdots k_0)_2}$$

$$k_i \uparrow \ k_i \uparrow \ \cdots \ \uparrow$$

Private key bits are directly loaded during the check phase,

but no countermeasures have been considered to protect this phase

I am going to present our paper at ISPEC 2017.

If you have any questions, let's see you there.