Something that is little can often have great power

Key Bit-dependent Attack on Scalar Multiplication using a Single-Trace

2018.09.10

Bo-Yeon Sim[*,**], Junki Kang [†], and Dong-Guk Han[*,**]

* Kookmin University

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

† The Affiliated Institute of ETRI

SICADA Side Channel Analysis Design Academy

Previously proposed attacks on PKCs were

Algorithm. Left to Right Scalar Multiplication	
INPUT	$P,\ d = (d_{n-1}, d_{n-2}, \cdots, d_0)_2$
OUTPUT	dP

Step 1. $R_0 = O$

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

2.1. $R_0 = 2R_0$

2.2. $R_{1-d_i} = R_0 + P$

Step 3. Return R_0

Algorithm. Left to Right Scalar Multiplication	
INPUT	$P,\ d = (d_{n-1}, d_{n-2}, \cdots, d_0)_2$
OUTPUT	dP

Step 1. $R_0 = O$

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

Countermeasure

Step 3. Return R_0

- 1996 Timing Attacks
- 1998 Simple Power Analysis
  - Differential Power Analysis
- 2004 Correlation Power Analysis
- 2001 Big Mac Attack
  - Various Collision Attacks
  - Various Template Attacks

Countermeasure

➔ apply more than one randomization method to make attack complexity higher

- DPA Countermeasure
  - DPA Countermeasure
    - SPA Countermeasure
    - Mathematical proof

Is there no vulnerability on key bit check phase?

- **At the beginning of each loop,**
  the key bit value is extracted from an n-bit key string $d = (d_{n-1}, d_{n-2}, \cdots, d_0)_2$
  and stored in a d_i variable

Algorithm. Left to Right Scalar Multiplication (Addition Always)	
INPUT	$P,\ d = (d_{n-1}, d_{n-2}, \cdots, d_0)_2$
OUTPUT	dP
Step 1. $R_0 = O$ **Step 2. For $i = n-1$ down to 0 do** **Countermeasure** Step 3. Return R_0	**+ point / scalar blinding**

$$d = (d_{n-1}d_{n-2}\cdots d_0)_2$$

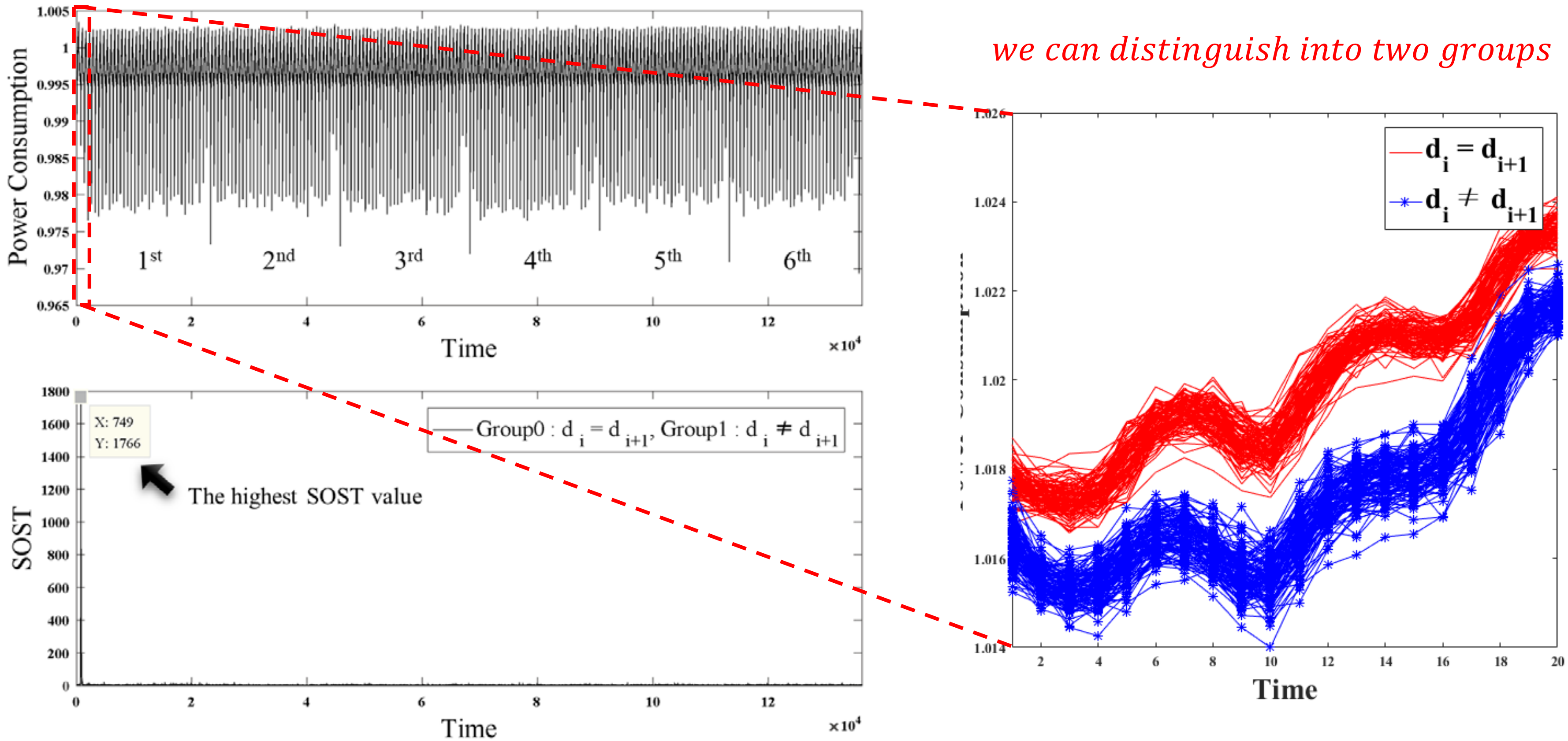
d_i ↑ d_i ↑ $\cdots$ ↑

Private key bits are directly loaded during the check phase,

but no countermeasures have been considered to protect this phase

In hardware implementation

- **The power consumption is related to the d_i value**

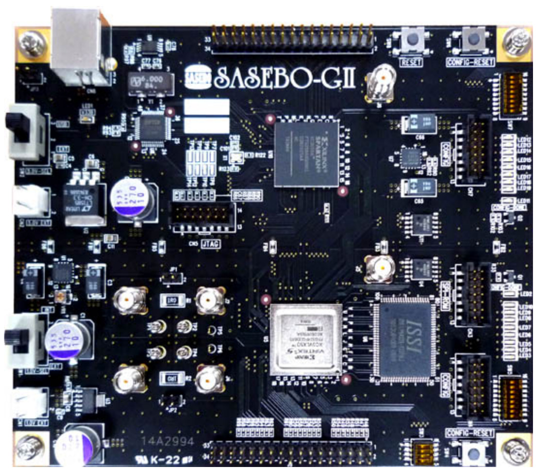
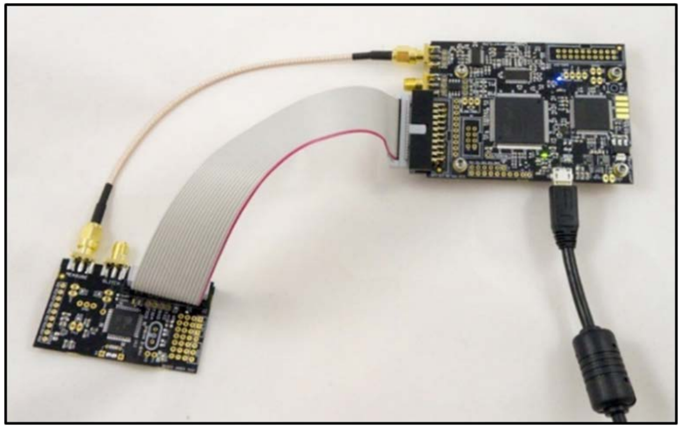
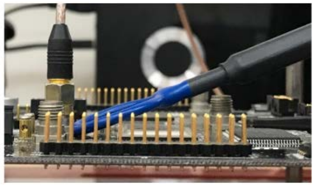


The attack does not require sophisticated pre-processing

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

SPA and DPA resistant algorithm

- ex) Montgomery-López-Dahab ladder algorithm + scalar randomization

Hardware Implementation	Software Implementation
	
Power Consumption	Power Consumption
Electromagnetic  	Electromagnetic  

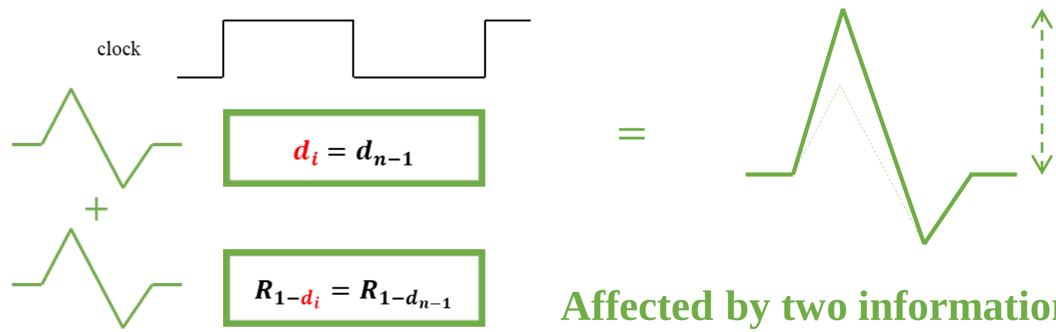
SICADA
Side Channel Analysis Design Academy

Property 1	$i = 4$	$i = 3$	$i = 2$	$i = 1$	$i = 0$
	$d_i = 1$	$d_i = 0$	$d_i = 0$	$d_i = 1$	$d_i = 1$

$\boldsymbol{d_4 \oplus d_3 = 1}$ $\boldsymbol{d_3 \oplus d_2 = 0}$ $\boldsymbol{d_2 \oplus d_1 = 1}$ $\boldsymbol{d_1 \oplus d_0 = 0}$

- **The operations are executed in parallel**
- **Registers to be accessed and the d_i value are determined simultaneously**

Step 2. For $i = n-1$ down to 0 do
  2.1. $R_0 = 2R_0$
  2.2. $R_{1-d_i} = R_0 + P$



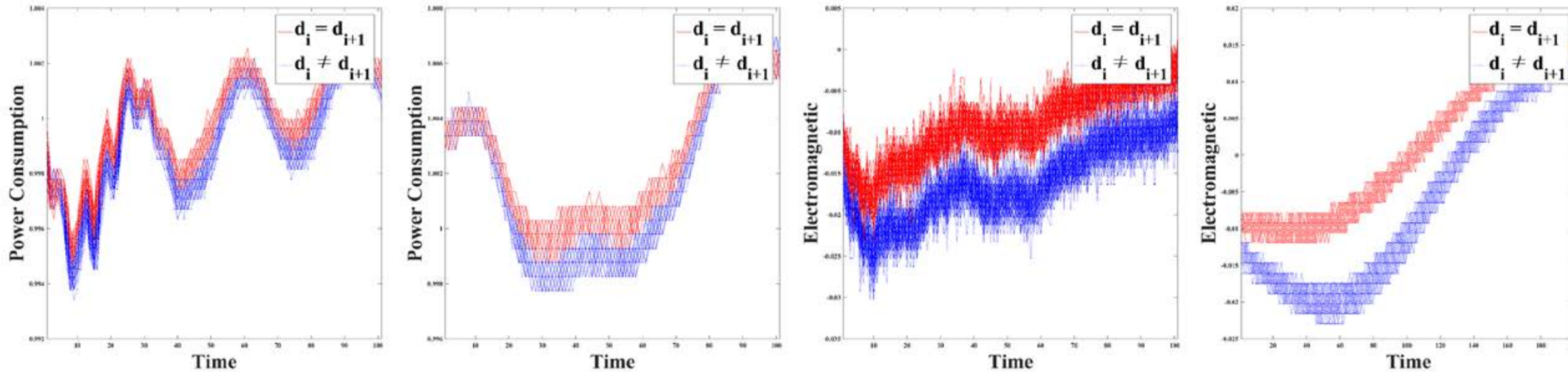
Affected by two information

Property 3 (include Property 1)	$R_0 = R_0 \times M$	$R_1 = R_0 \times M$	$R_1 = R_0 \times M$	$R_1 = R_0 \times M$	$R_1 = R_0 \times M$

$\boldsymbol{R_0 \oplus R_1 \neq 0}$ $\boldsymbol{R_1 \oplus R_1 = 0}$ $\boldsymbol{R_0 \oplus R_1 \neq 0}$ $\boldsymbol{R_1 \oplus R_1 = 0}$

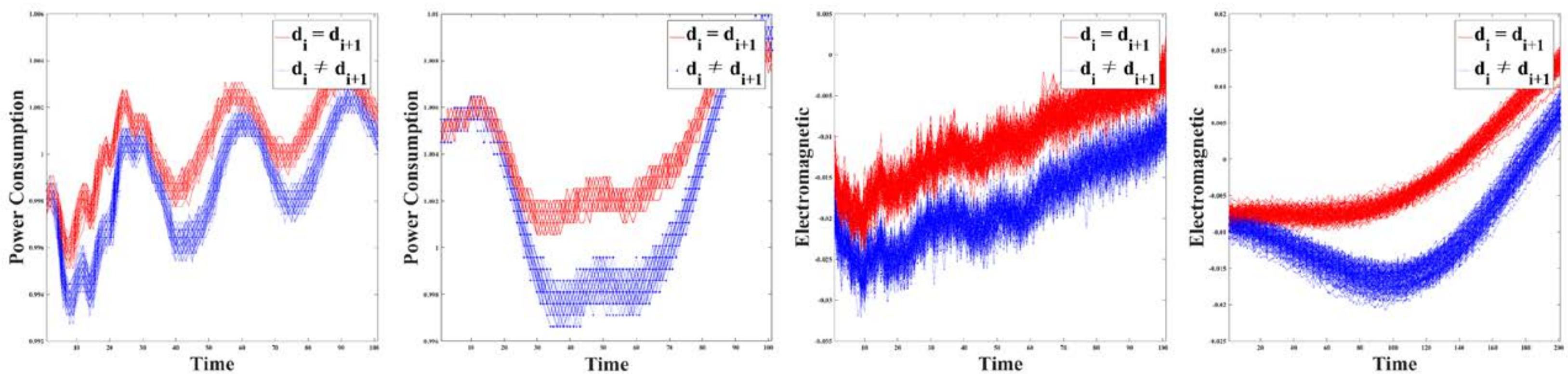
Key Bit-dependent Attack using a Single-Trace

Montgomery-López-Dahab ladder algorithm + scalar randomization



new result

Hardware	Power Consumption		Electromagnetic	
	None	Low Pass Filter	None	Low Pass Filter
	K-MEANS	K-MEANS	K-MEANS	K-MEANS
Property 1	97.74 %	97.71 %	100 %	100 %
Property 3	100 %	100 %	100 %	100 %



In software implementation

Key bit check function of mbedTLS (openSSL)

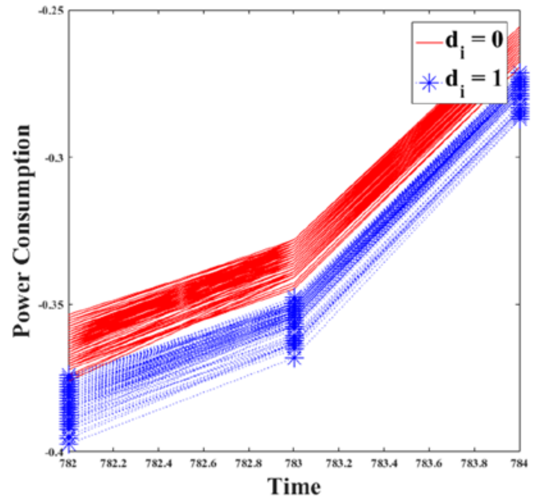
according to Hamming Weight of d_i

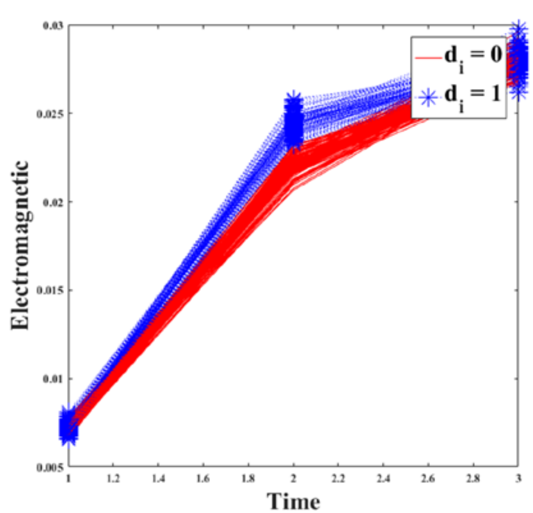
Software	Power Consumption	
	None	
	DIFF	K-MEANS
Property 2	97.60 %	97.60 %
Property 4	100 %	100 %

new result

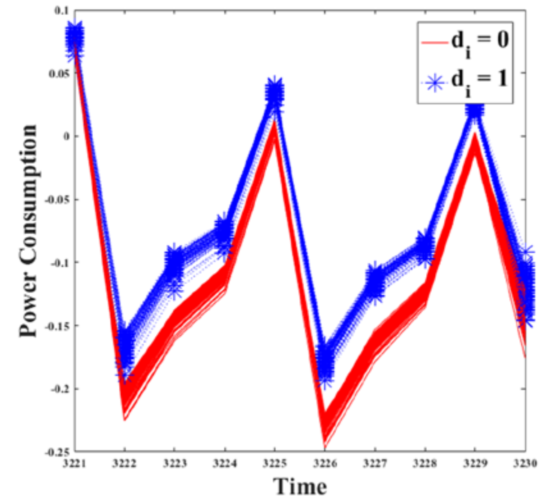
Software	Electromagnetic	
	Low Pass Filter	
	DIFF	K-MEANS
Property 2	93.72 %	94.17 %
Property 4	94.17 %	95.96 %

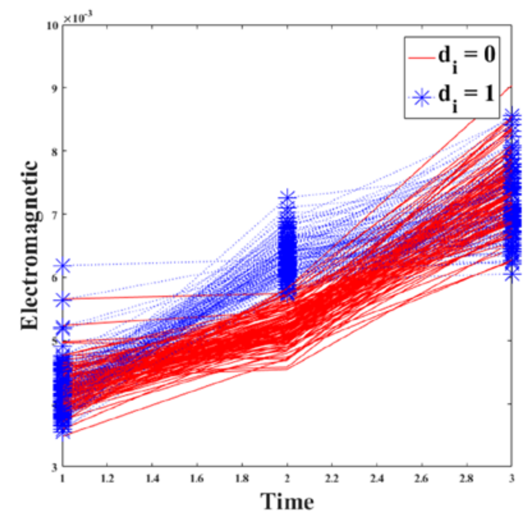
✓ **d_i value (key bit check phase)**





✓ **referred register address R_{d_i}**





Related article

- **If you have any question, refer to following article**

https://doi.org/10.1007/978-3-319-72359-4_10