Scan-Based Attack against Elliptic Curve Cryptosystems

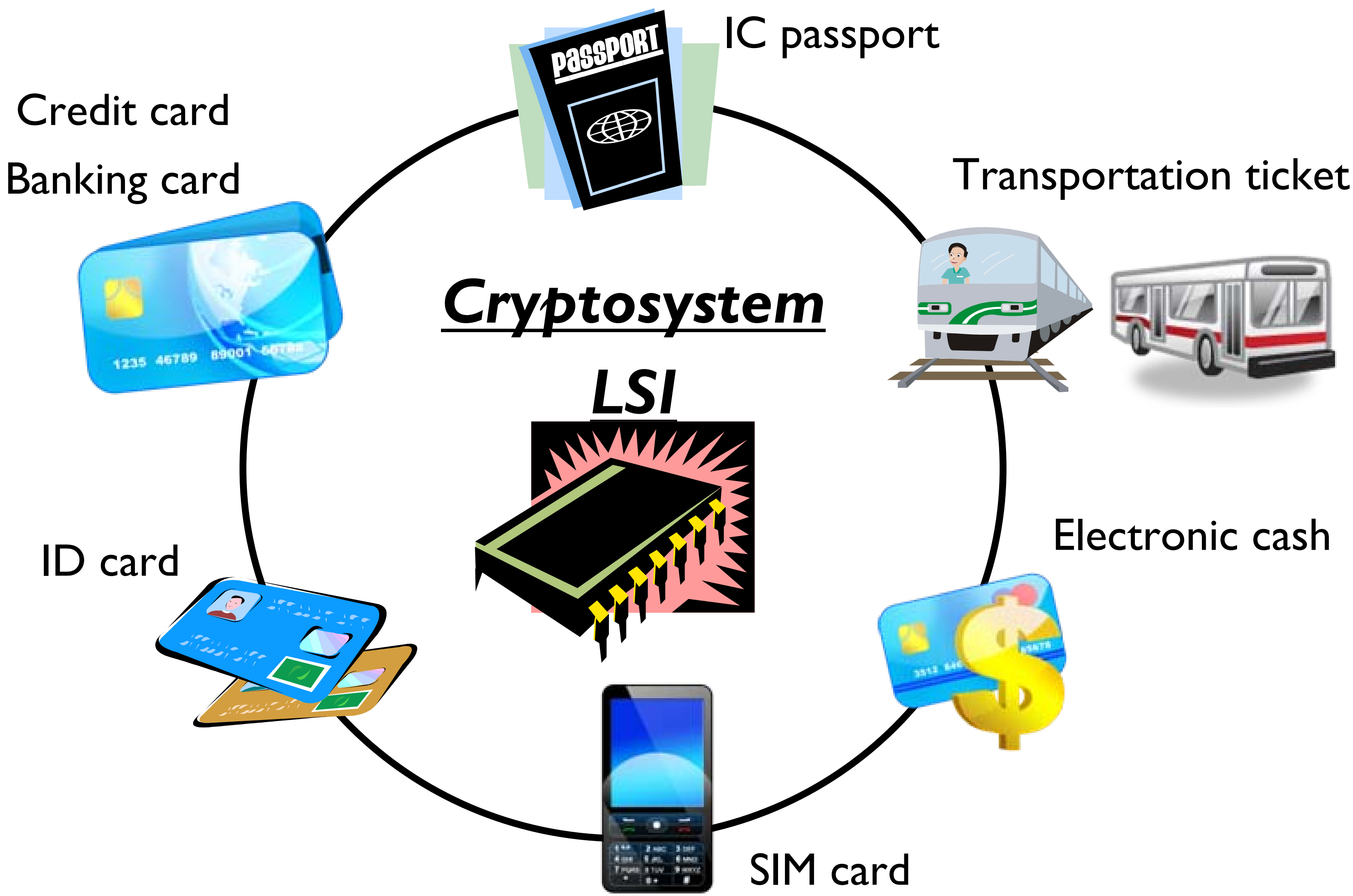
Graduate School of Fundamental Science and Engineering, Waseda University

Ryuta Nara, Nozomu Togawa,
Masao Yanagisawa, Tatsuo Ohtsuki

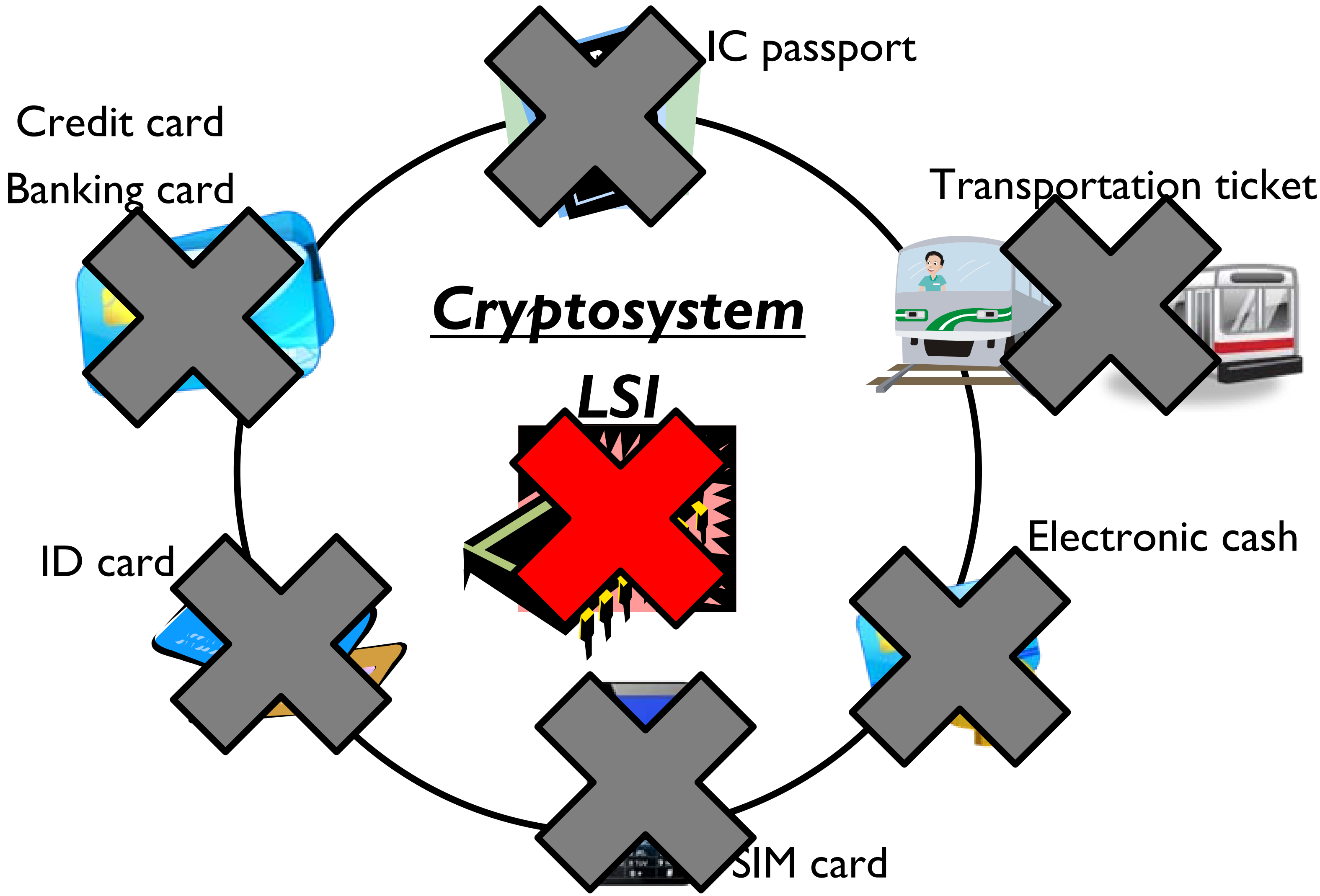
Outline

- Background
- Scan-based attacks
- Elliptic curve cryptosystem(ECC)
- Scan-based attack against ECC
- Experiments and results
- Conclusion

Background – Cryptosystem LSI –

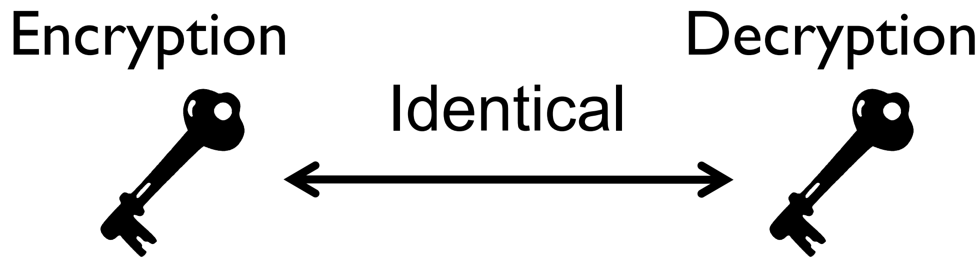


Background – Cryptosystem LSI –

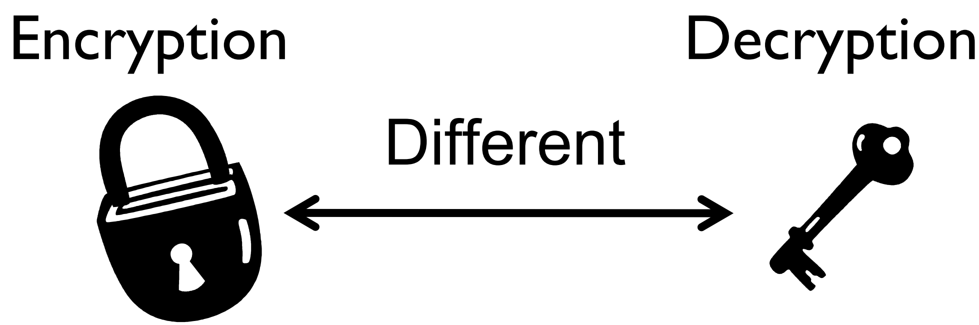


Background – Symmetric vs. Public –

- Symmetric-key cryptosystem: DES, AES



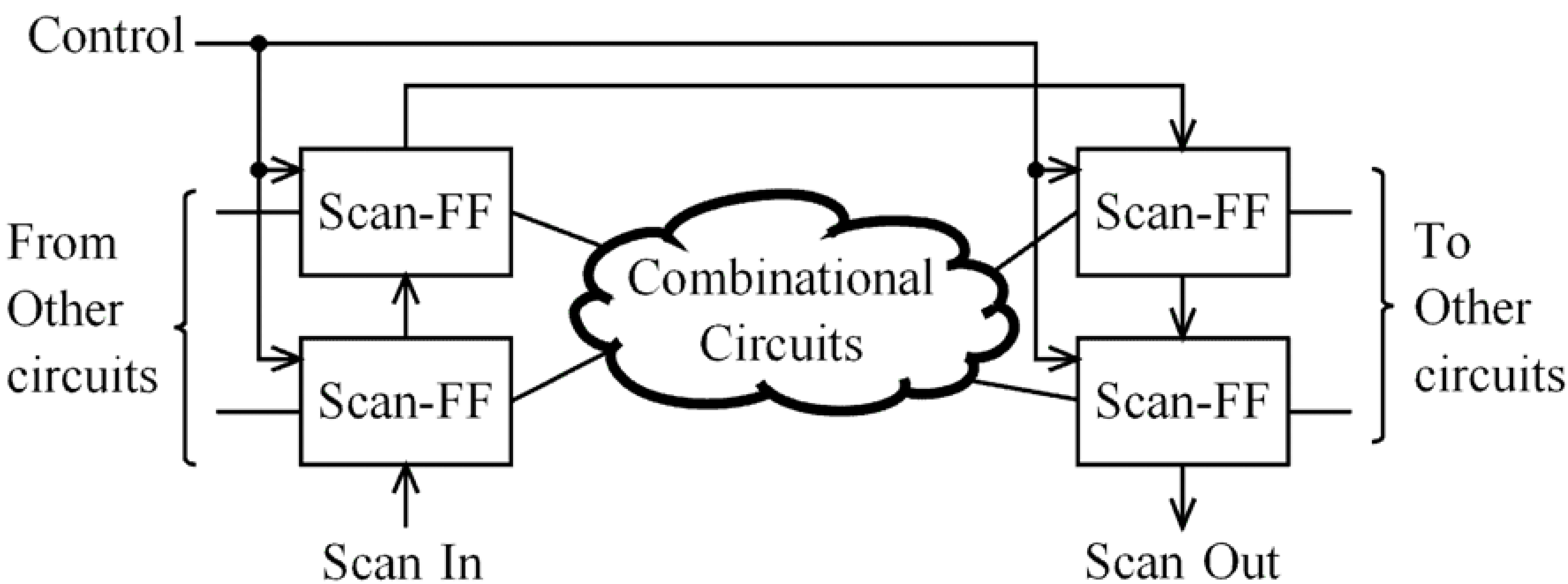
- Public-key cryptosystem: RSA, ***<u>ECC</u>****



******<u>***ECC: Elliptic curve cryptosystem***</u>**

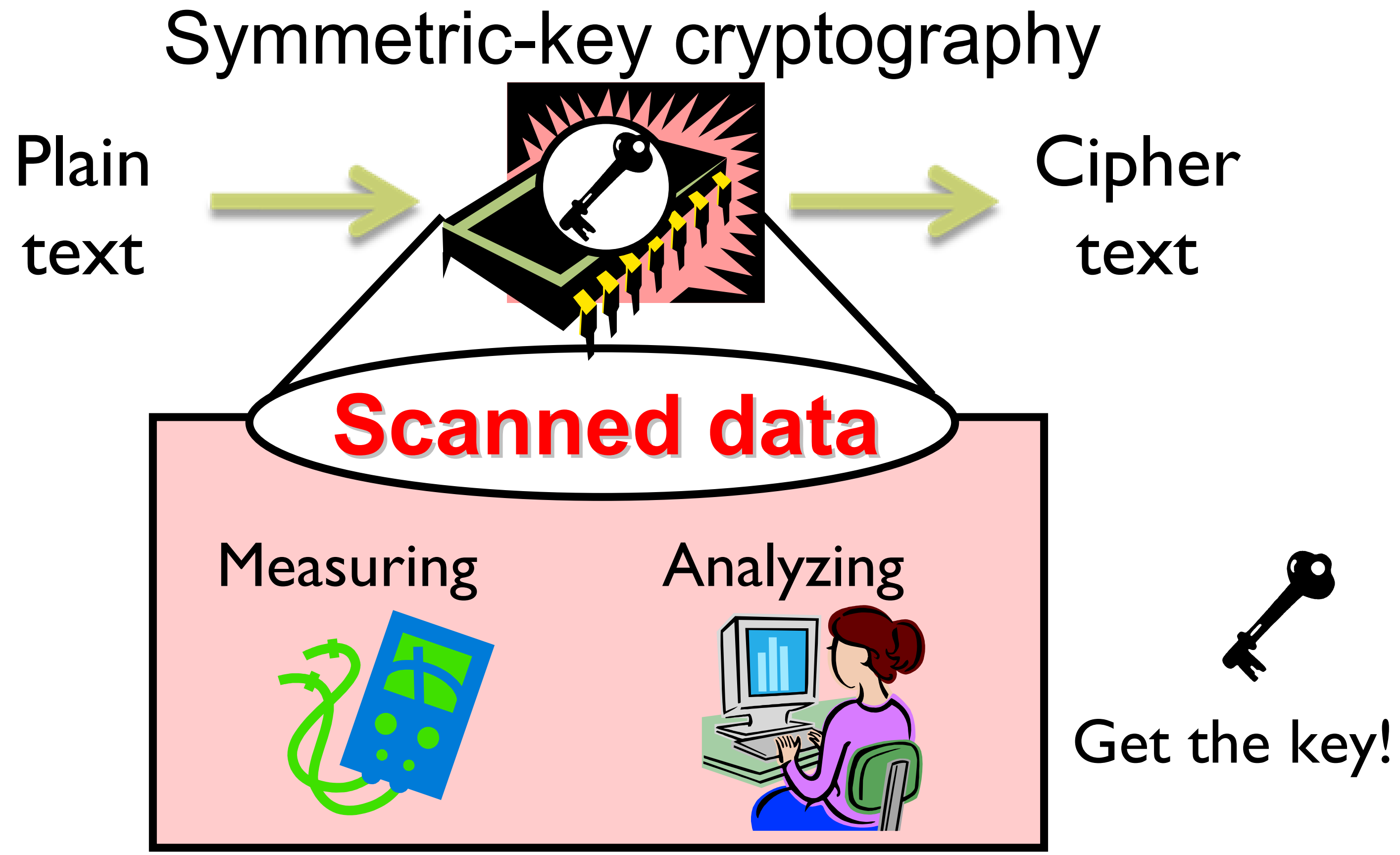
Background – Scan path test –

- High test efficiency
- Easy to implement



Scan path test

Scan-based attacks against DES[1] and AES[2,3]



[1] B. Yang, et al., International Test Conference, 2004.
[2] B. Yang, et al., Design Automation Conference(DAC), 2005.
[3] R. Nara, et al., IEICE, E92-A, No.12, Dec. 2009.

Purpose of our presentation

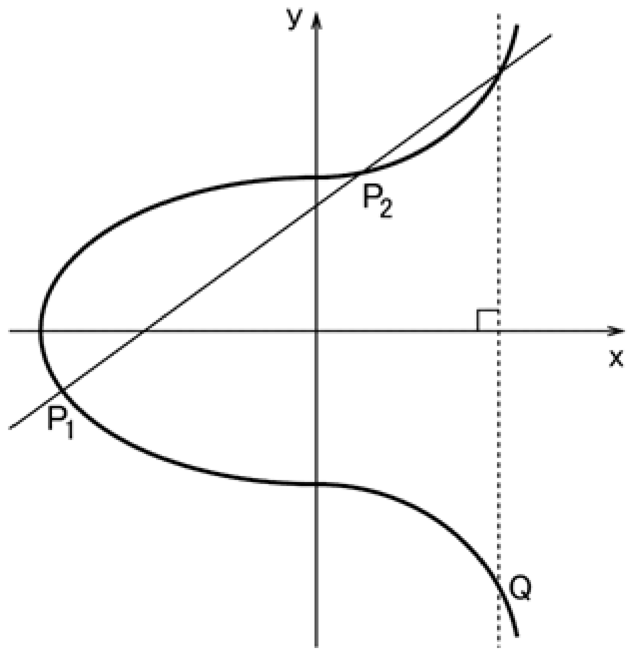
Scan-based attack

against

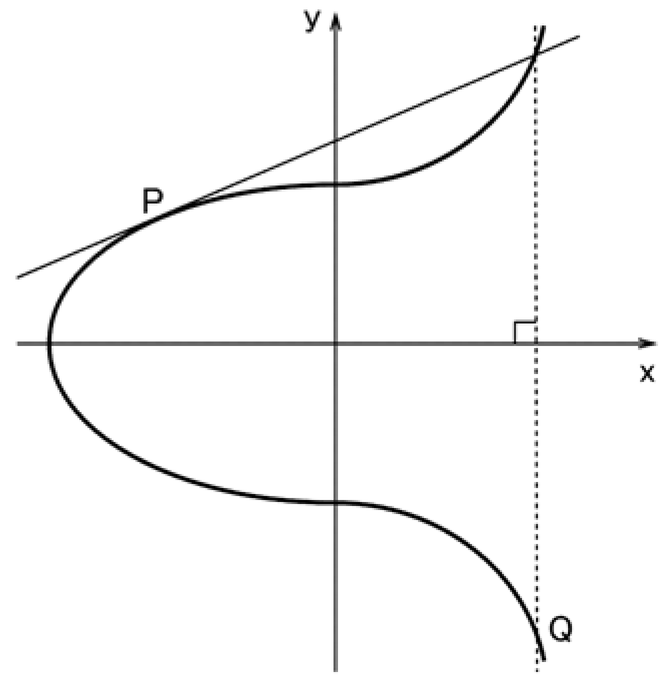
elliptic curve cryptosystem

Elliptic curve cryptosystem

- Basic operation



$$Q = P_1 + P_2$$



$$Q = 2P$$

Elliptic curve cryptosystem: Q=kP

- Montgomery's method[15]

$$k = (\underbrace{k_{m-1}\, k_{m-2} \ldots k_{i+1}}\, \boxed{k_i} \ldots k_1\, k_0),\ k_j \in \{0,1\}$$

$[lP,\ (l+1)P] \rightarrow$ $[\ 2lP,\ (2l+1)P]$ if $k_i=0$

$*l = k_{m-1}\, k_{m-2} \ldots k_{i+1}$ $\searrow$ $[(2l+1)P,\ 2(l+1)P]$ if $k_i=1$

$k_3=1$	P, 2P	
$k_2=1$	3P, 4P	P+2P = 3P, 2 × 2P = 4P
$k_1=0$	6P, 7P	2 × 3P = 6P, 3P+4P = 7P,
$k_0=1$	13P, 14P	6P+7P = 13P(1101_2P), 2 × 7P = 14P

[15] P. L. Montgomery, Mathematics of Computation, 1987.

Intermediate values vs. secret key k[18]

If $k = (k_{m-1}\, k_{m-2} \ldots k_{i+1}\, k_i \ldots k_1\, k_0)$, $k_j \in \{0,1\}$

$k_{m-1} \ldots k_{i+1}$: already-known; k_i: unknown

iff $\boldsymbol{k_i = 0}$, V(i)P ∈ a set of intermediate values

$$V(i)P = \left(\sum_{j=i}^{m-1} k_j 2^{j-i+1} + 1 \right) P$$

$$\boldsymbol{k_i \Leftrightarrow V(i)P}$$

[18] L. Goubin, PKC, 2003.

Sample

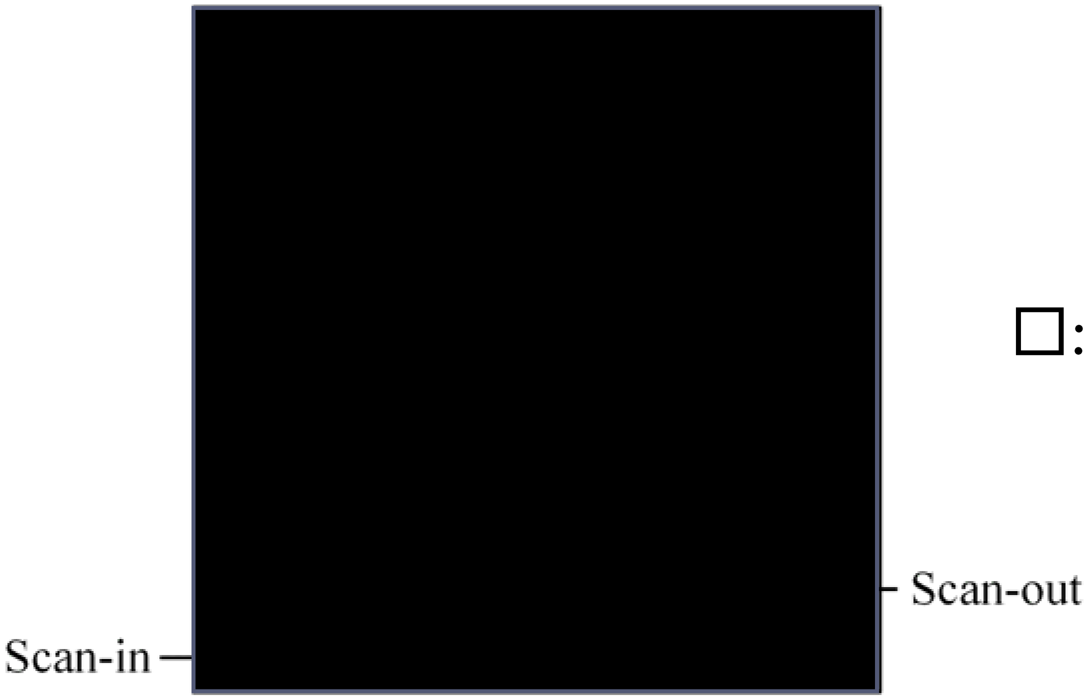
Intermediate values V(i)P (4 bits)

<table>
<tr><td>k_3</td><td colspan="8">1</td></tr>
<tr><td rowspan="2">k_2</td><td colspan="4">5P</td><td colspan="4">7P</td></tr>
<tr><td colspan="4">0(10)</td><td colspan="4">1(11)</td></tr>
<tr><td rowspan="2">k_1</td><td colspan="2">9P</td><td colspan="2">11P</td><td colspan="2">13P</td><td colspan="2">15P</td></tr>
<tr><td colspan="2">0(100)</td><td colspan="2">1(101)</td><td colspan="2">0(110)</td><td colspan="2">1(111)</td></tr>
<tr><td rowspan="2">k_0</td><td>8P</td><td>9P</td><td>10P</td><td>11P</td><td>12P</td><td>13P</td><td>14P</td><td>15P</td></tr>
<tr><td>1000</td><td>1001</td><td>1010</td><td>1011</td><td>1100</td><td>1101</td><td>1110</td><td>1111</td></tr>
</table>

P, 2P
3P, 4P
6P, 7P
13P, 14P

1. $k_3 = 1$
2. V(2)P = **5P** does not exit → $k_2 = 1$
3. V(1)P = **13P** exits → $k_1 = 0$
4. k_0=0? or k_0=1? → **13(1101)P**

How to find the intermediate values?

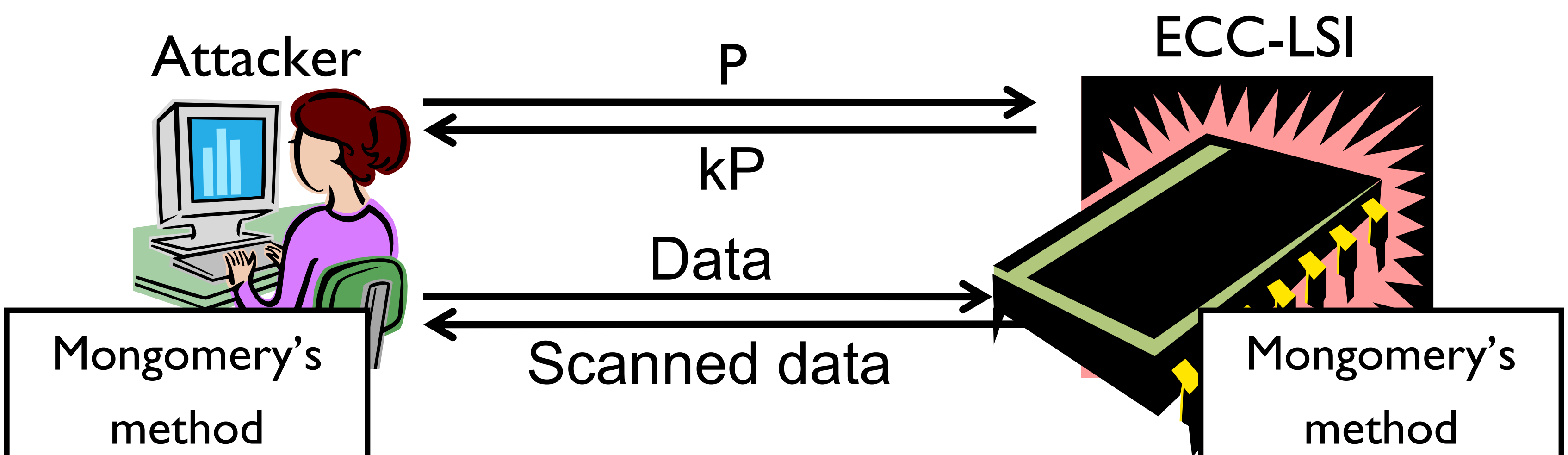


Layout of ECC LSI
□:Cell gate, ■:Register

- Problem 1: Which registers store intermediate values?
- Problem 2: When do intermediate values appear?

Assumption

- Attackers can
  - compute kP with any P by using an ECC LSI
  - access the scan path
- Known
  - kP algorithm used in an ECC LSI (e.g. Montgomery's method)

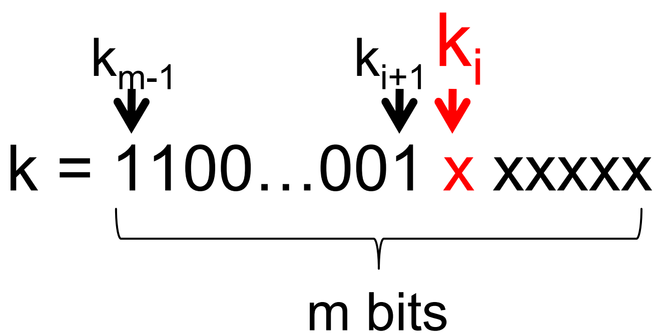


Proposed method

- **Discriminator to V(i)P erated from *<u>1-bit register value</u>***

1-bit register is

- Independent of the connection of registers
- Independent of timing

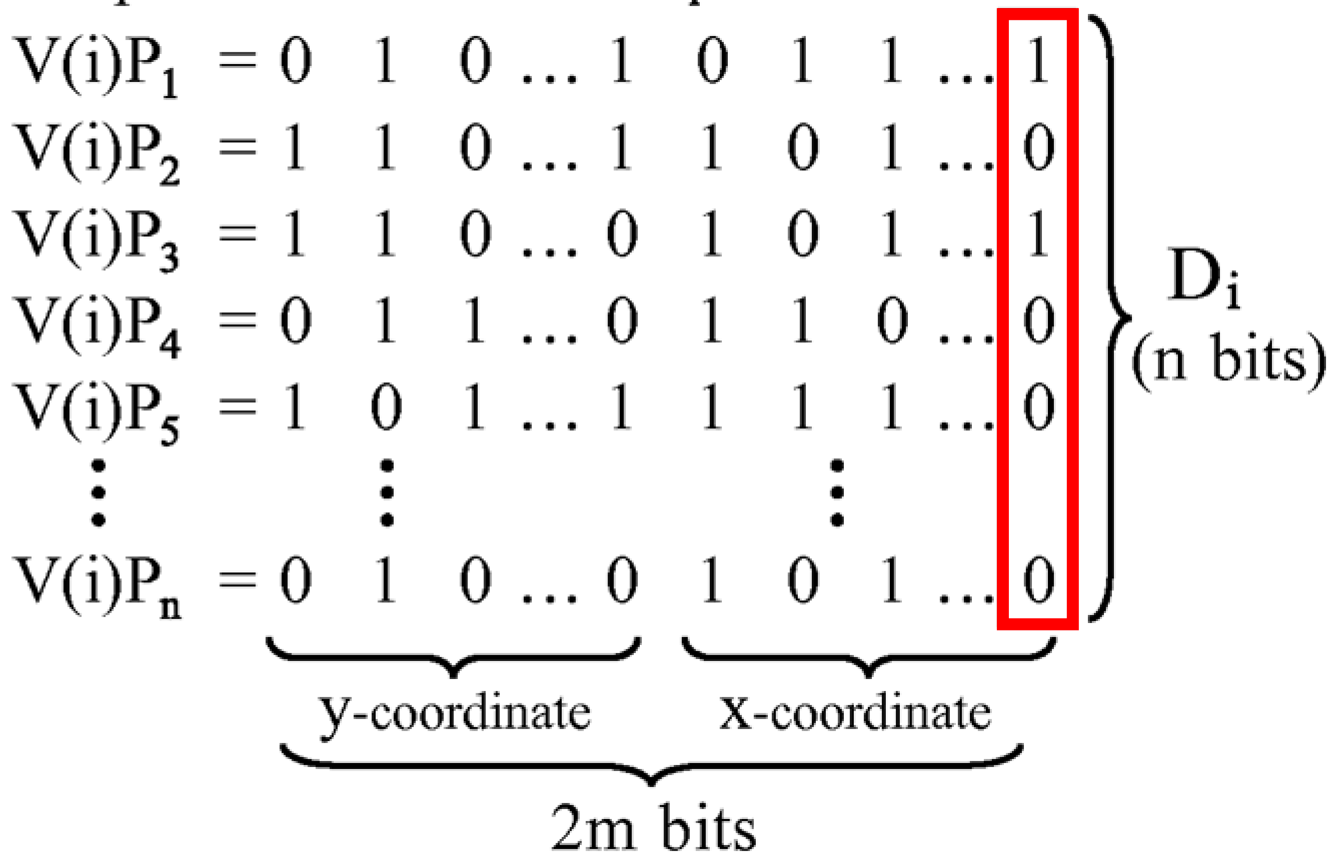


Let's find k_i !

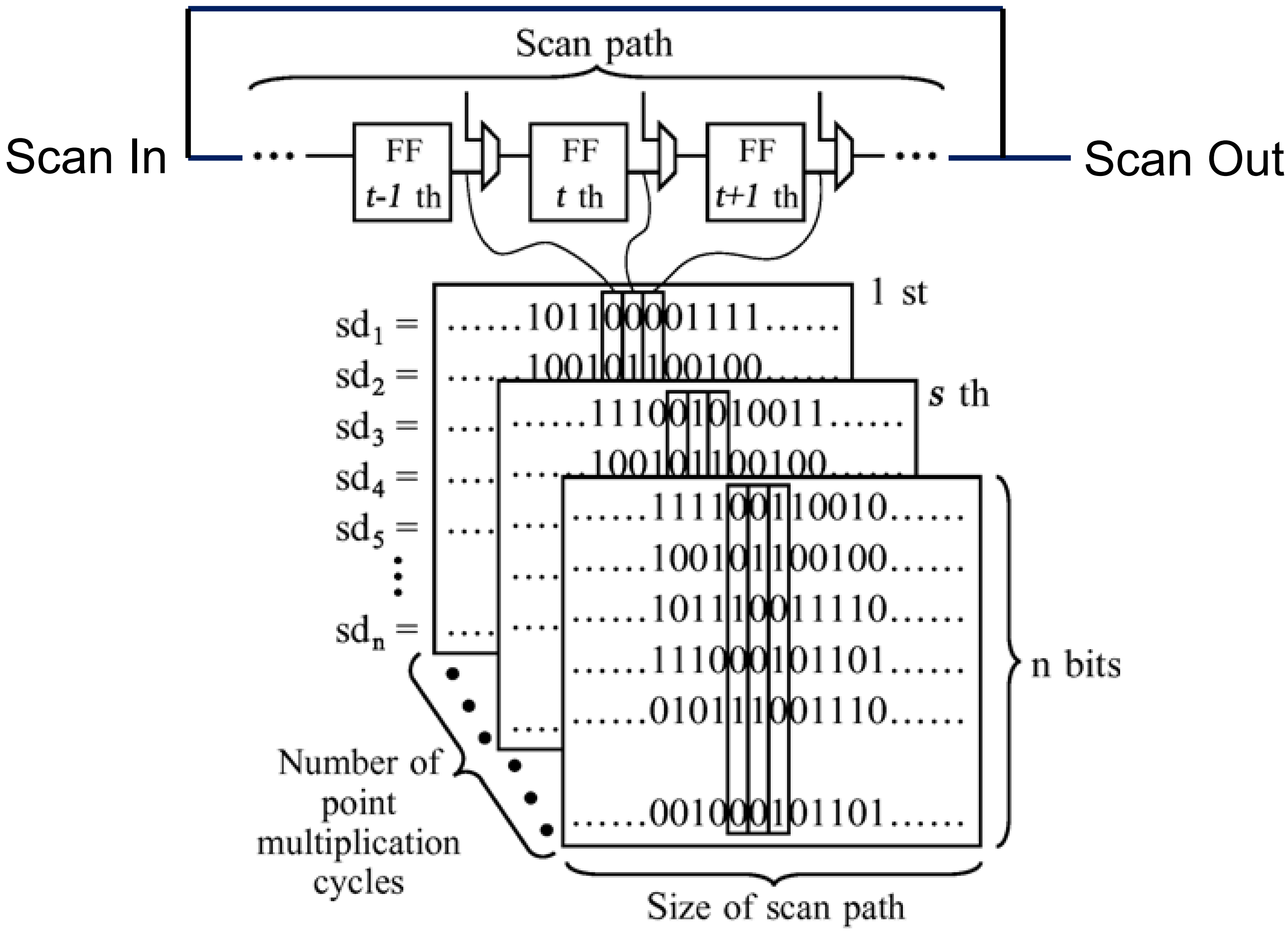
D_i: Discriminator to V(i)P

Input: $P_r \in E(F_{2^m})$ $(1 \leq r \leq n)$, V(i)

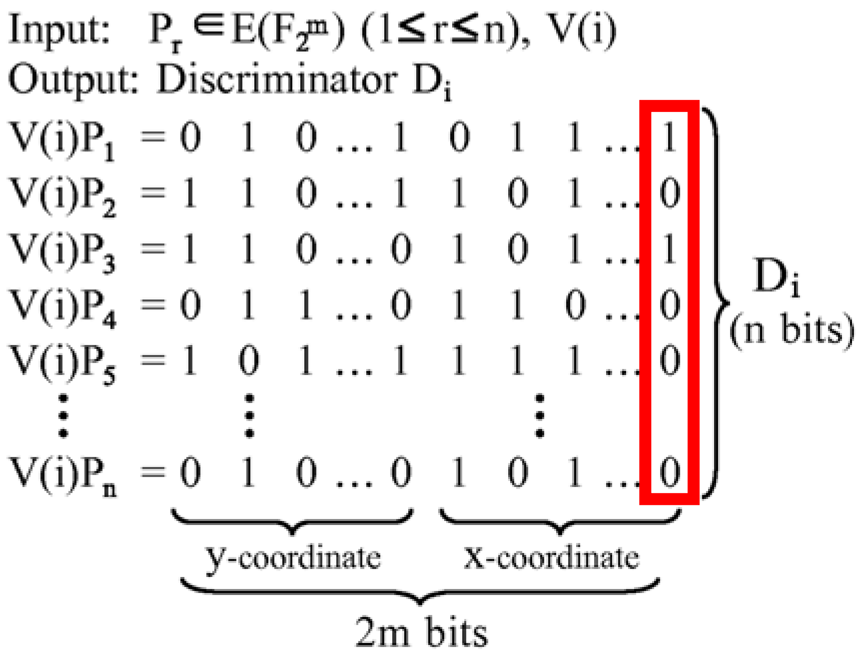
Output: Discriminator D_i

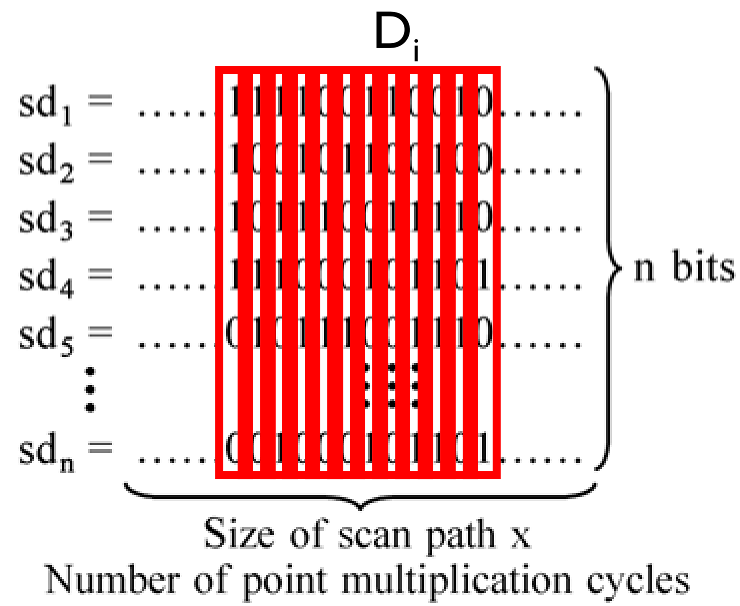


Scanned data



D_i exits?



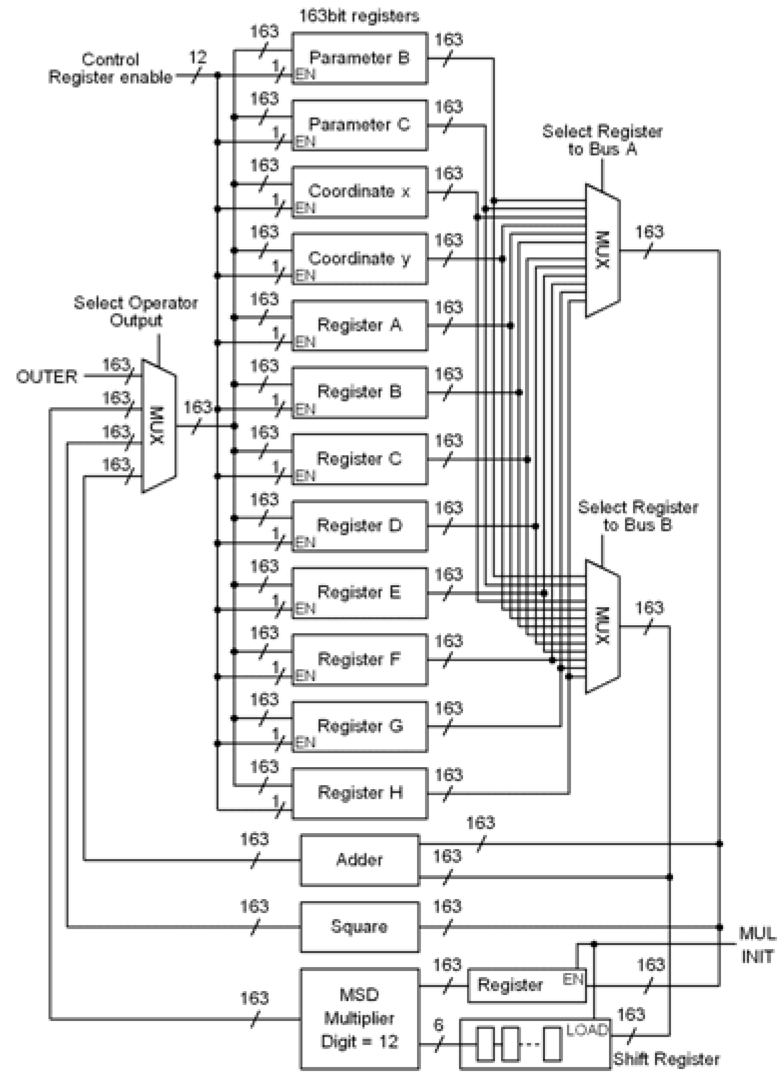


D_i exists ⇔ k_i=0

D_i does not exist ⇔ k_i=1

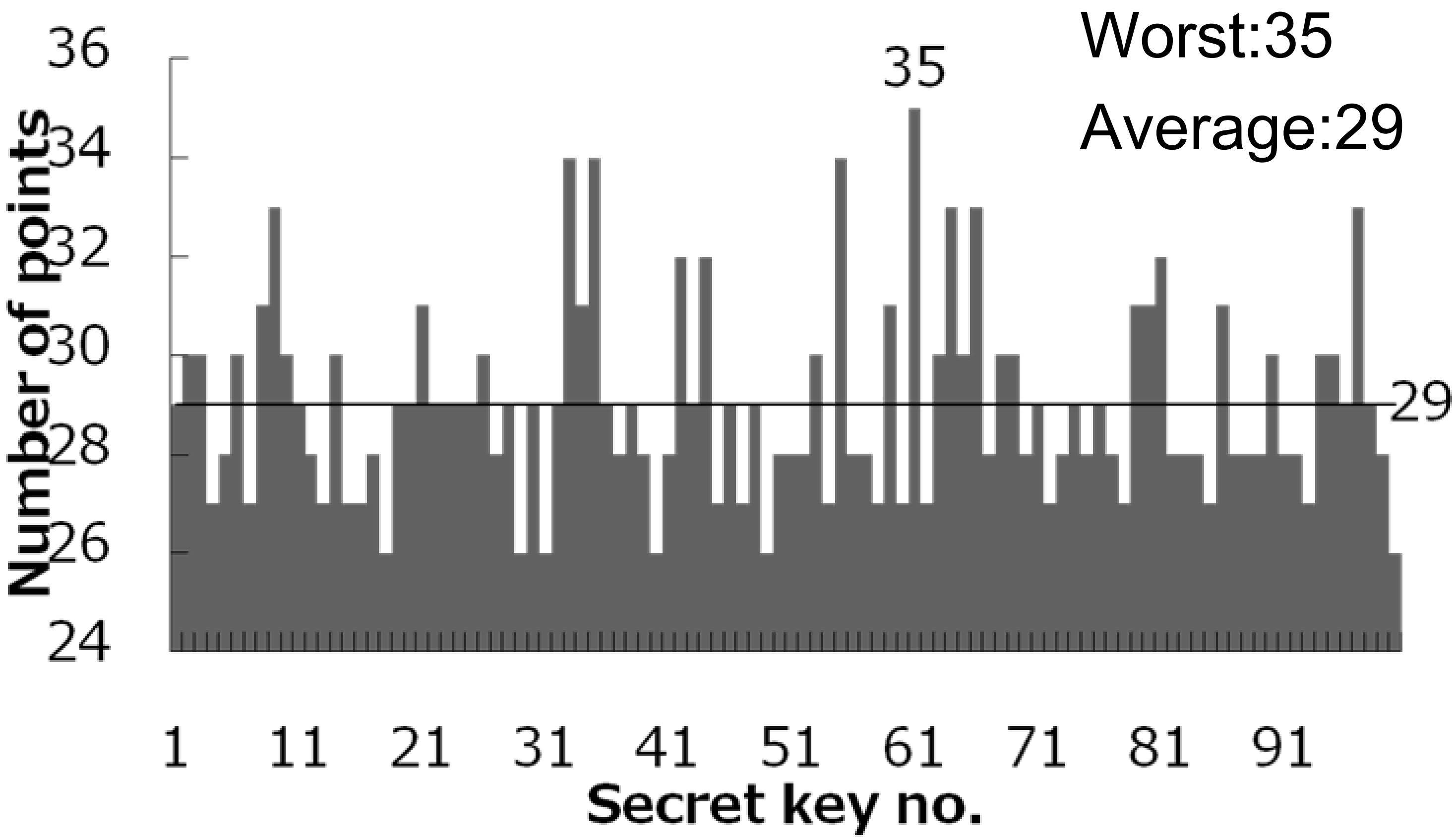
Experiments

- Key length: **<u>163</u>** bits
- Size of scan path : **<u>2,520</u>** bits
- Q=kP : **<u>15,137</u>** cycles



ECC-LSI architecture

Results


Worst:35
Average:29
35
29
36
34
32
30
28
26
24
Number of points
1
11
21
31
41
51
61
71
81
91
Secret key no.

Conclusion

- **Scan-based attack** against **elliptic curve cryptosystem**
- Deciphering a secret key k
  at **40 seconds** by using **29 input**

Future works

- Attacks :
  - Method for accessing scan path
  - Compactor
- Defense :
  - Efficiency defense method