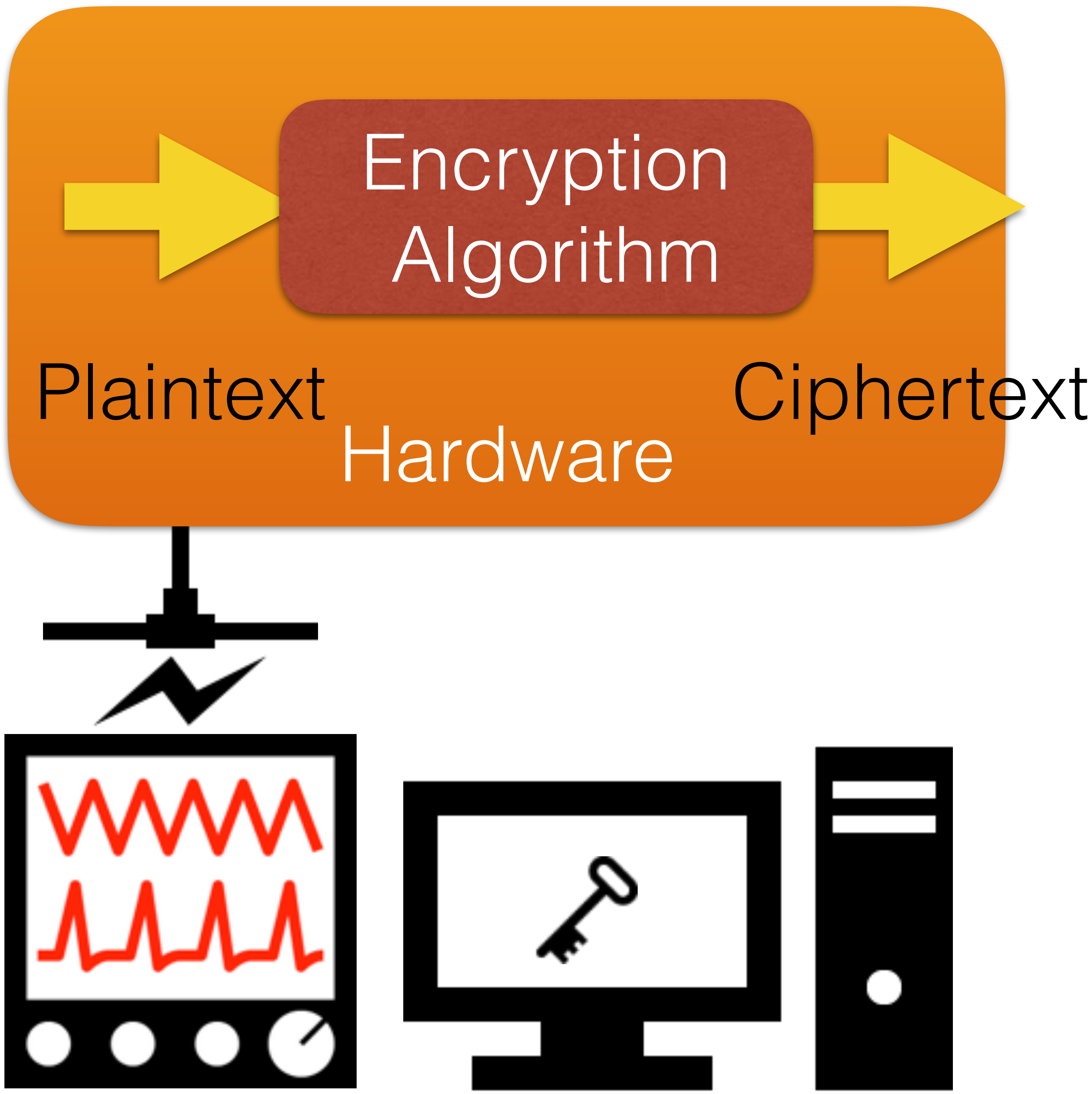
Correlation Power Analysis of AES-256 on ATmega328P

游世群 JPChen 許遠哲

Outlines

- **SCA/DPA/CPA**
- Hardware Implementation
- Demo Video
- CPA Implementation on AES Rounds
- Countermeasures
- Conclusion

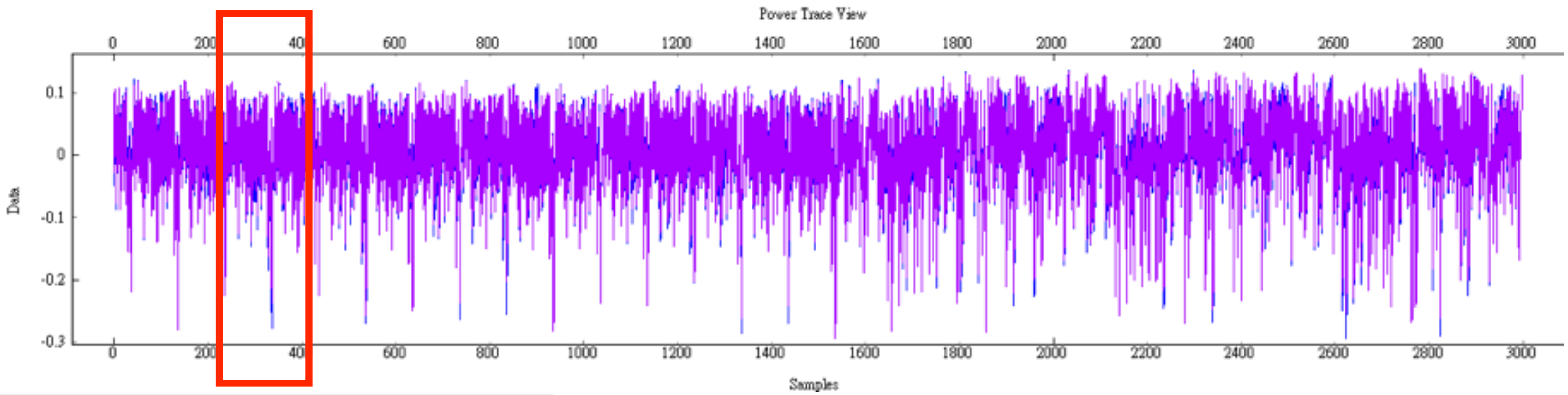
Side-Channel Analysis

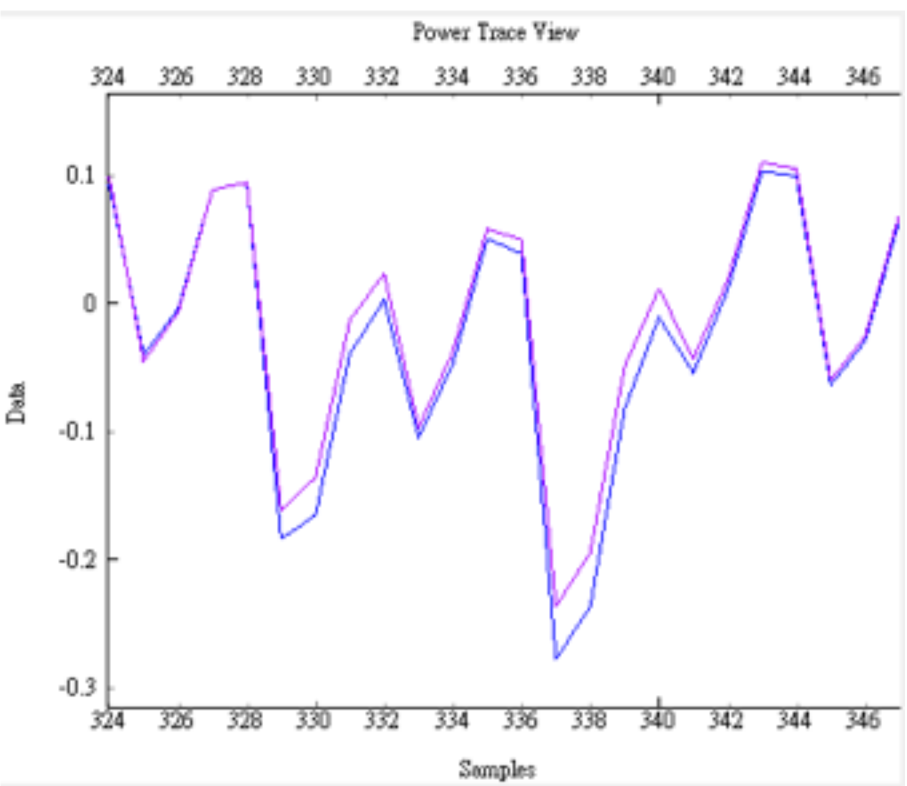


- There is a key hidden in an encryption algorithm.
- We need a hardware to implement this system.
- This hardware may leak information about the key.
- By analyzing the leakages, we can rebuild the key.

Differential Power Analysis

Compare two power traces from two different encryptions:





Differential Power Analysis (DPA)

- By analyzing the **difference between traces** from different encryptions.
- Use statistical tools to recover the key.

Divide and Conquer

12	43	F5	68
77	26	54	87
A3	B3	7E	FF
9B	4A	AF	E8

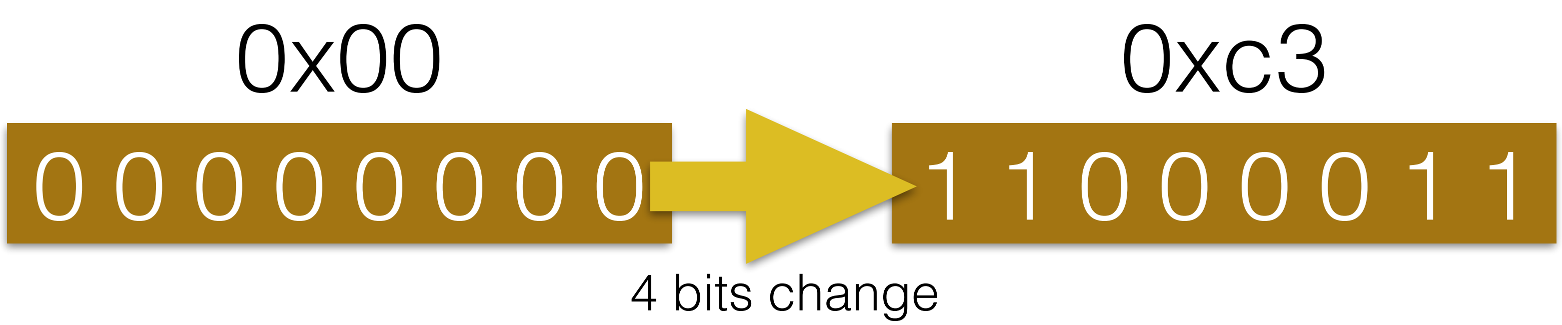
A Block of AES

- AES is a ***block cipher***.
- 1 byte as a unit.
- Plaintexts, Round Keys, Ciphertexts and Intermediate values can be regarded as 16 ***independent*** bytes.

Search Space: reduced from 2^{128} to 16×2^{8}

Power Consumption in Register

A register.



Assume that each bit changes costs the same value b, the overall power consumption y will be:

$$y = a + \mathrm{HD}(0\mathrm{x}00, 0\mathrm{xc}3) \cdot b + \mathrm{N}$$

Hamming Distance of these 2 hex-numbers

Power Consumption in Register

0x6d

0 1 **1** 0 **1** **1** **0** 1

0xc3

1 1 **0** 0 **0** **0** **1** 1

- Hamming Distance model:

$$y = a + \mathrm{HD}(0\mathrm{x}6\mathrm{c}, 0\mathrm{xc}3) \cdot b + \mathrm{N}$$

0x00

0 0 0 0 0 0 0 0

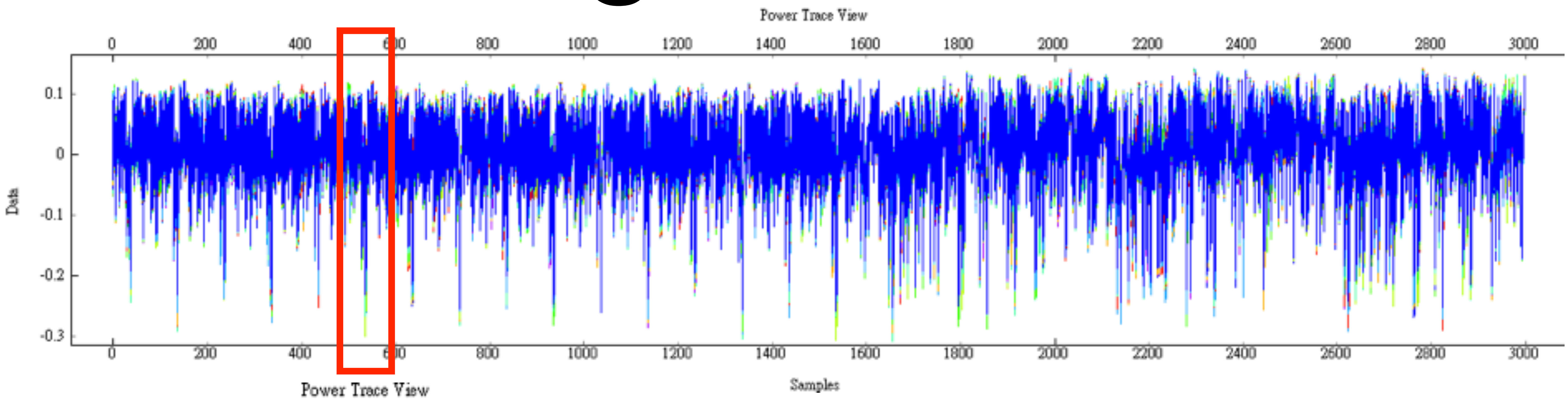
0xc3

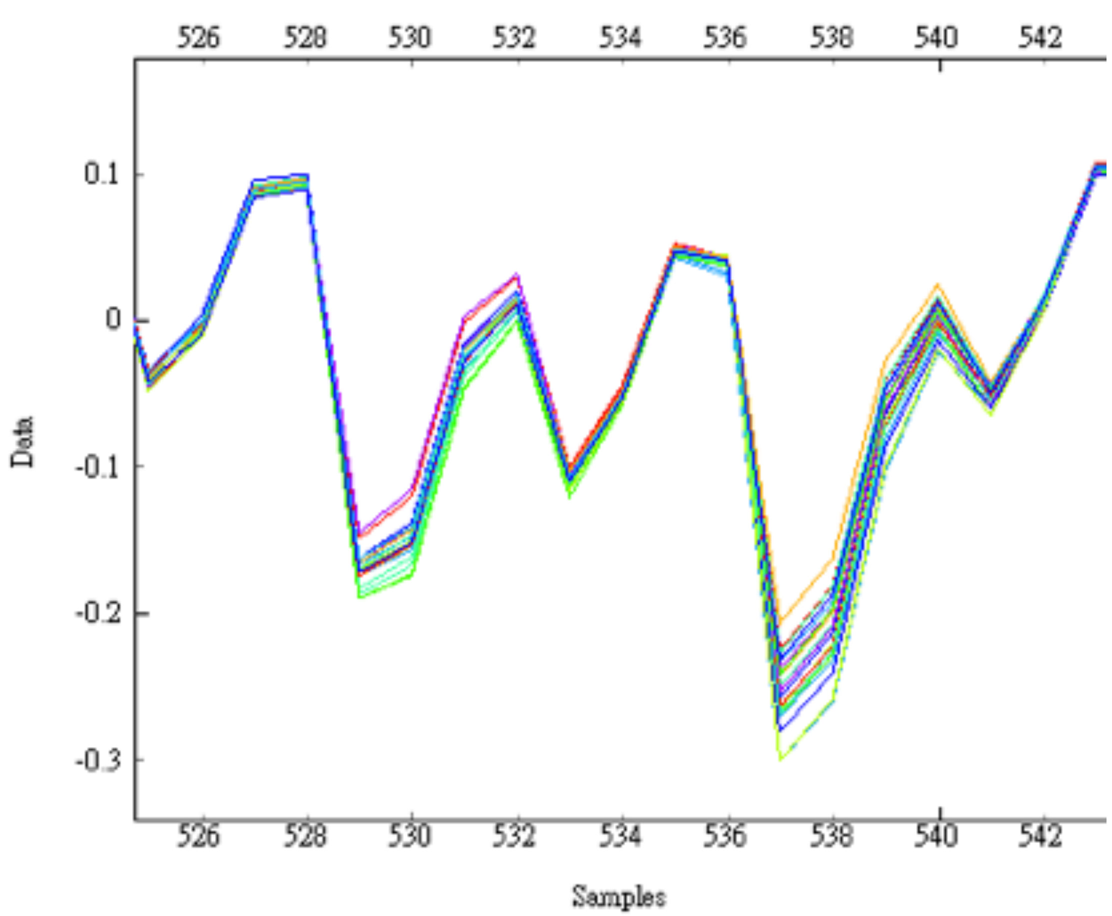
1 1 0 0 0 0 1 1

- Hamming Weight model:

$$y = a + \mathrm{HW}(0\mathrm{xc}3) \cdot b + \mathrm{N}$$

Leakages from AES

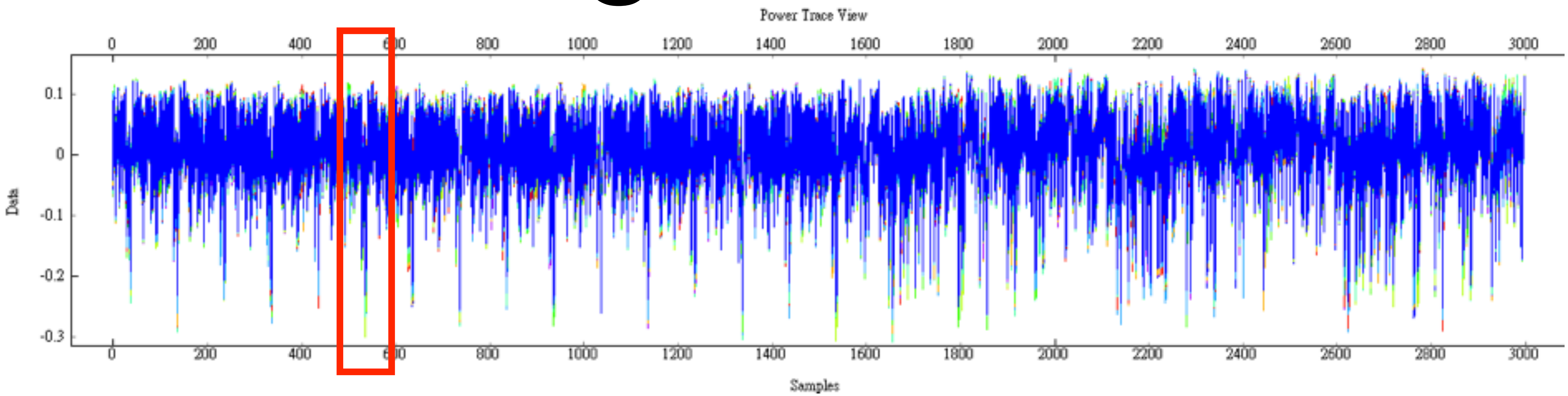




$$y = a + \mathrm{HD}(0x00, 0xc3) \cdot b + \mathrm{N}$$

$$\mathrm{T}_0\text{: } y_0 = a + \mathrm{H}(\ f_5(\mathrm{p}_0,\mathrm{k})\ ) \cdot b + \mathrm{N}$$
$$\mathrm{T}_1\text{: } y_1 = a + \mathrm{H}(\ f_5(\mathrm{p}_1,\mathrm{k})\ ) \cdot b + \mathrm{N}$$
$$\mathrm{T}_2\text{: } y_2 = a + \mathrm{H}(\ f_5(\mathrm{p}_2,\mathrm{k})\ ) \cdot b + \mathrm{N}$$
$$\vdots$$
$$\mathrm{T}_n\text{: } y_n = a + \mathrm{H}(\ f_5(\mathrm{p}_n,\mathrm{k})\ ) \cdot b + \mathrm{N}$$

Leakages from AES



T_0: $y_0 = a + \mathrm{H}(\, f_5(\mathrm{p}_0, \mathrm{k})\,) \cdot b + \mathrm{N}$

T_1: $y_1 = a + \mathrm{H}(\, f_5(\mathrm{p}_1, \mathrm{k})\,) \cdot b + \mathrm{N}$

T_2: $y_2 = a + \mathrm{H}(\, f_5(\mathrm{p}_2, \mathrm{k})\,) \cdot b + \mathrm{N}$

$\vdots$

T_n: $y_n = a + \mathrm{H}(\, f_5(\mathrm{p}_n, \mathrm{k})\,) \cdot b + \mathrm{N}$

k: key

p_i : known plaintext

f_i : the i-th Intermediate value function

H: Hamming Distance or Hamming Weight

Correlation Power Analysis

T_0: $y_0 = a + H(\,f_5(p_0,k)\,) \cdot b + N$
T_1: $y_1 = a + H(\,f_5(p_1,k)\,) \cdot b + N$
T_2: $y_2 = a + H(\,f_5(p_2,k)\,) \cdot b + N$
$\vdots$
T_n: $y_n = a + H(\,f_5(p_n,k)\,) \cdot b + N$

If our key guessing is ***right***, Cor(y, x) will be significant.

If it is ***wrong***, Cor(y, x) will be close to 0.

Pearson Correlation Coefficient:

$$Cor(y,x) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \cdot \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

Outlines

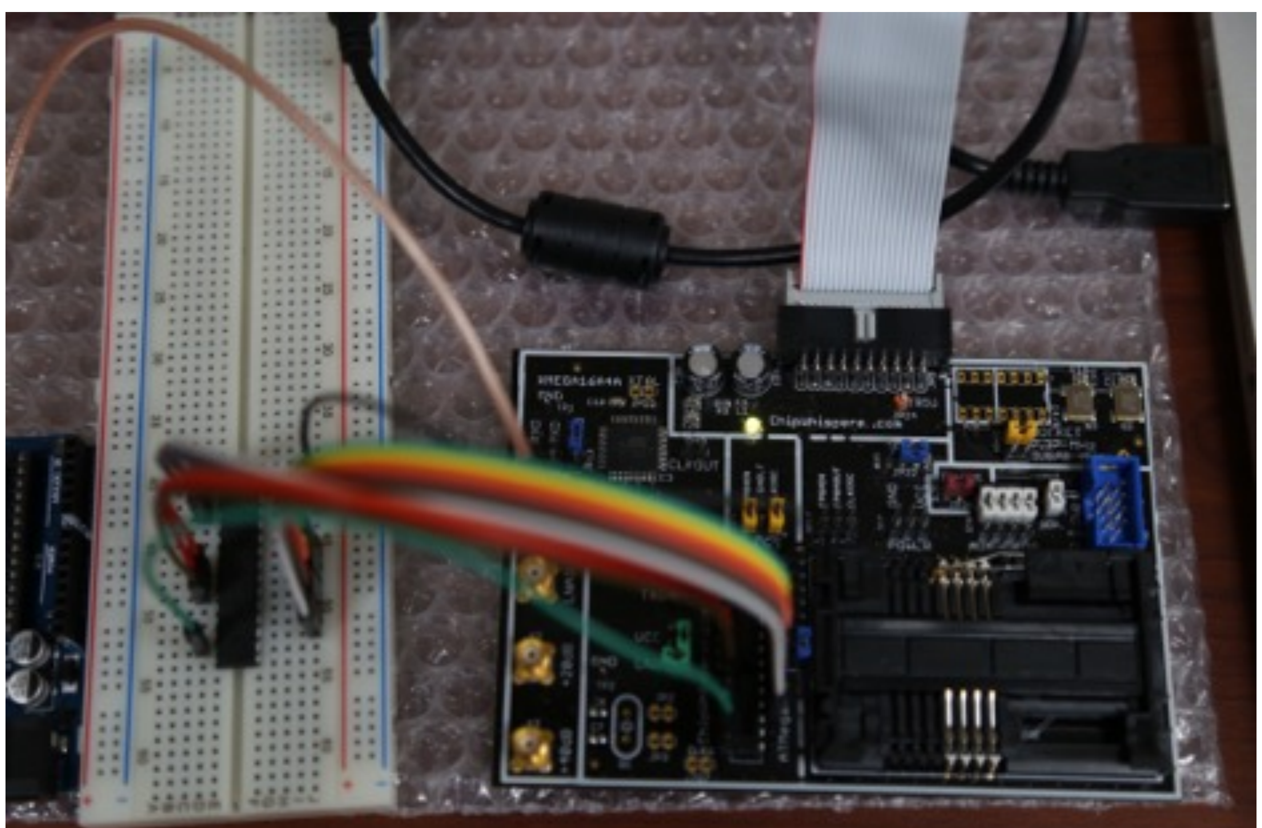
- SCA/DPA/CPA
- **Hardware Implementation**
- Demo Video
- CPA Implementation on AES Rounds
- Countermeasures
- Conclusion

ATMega328P

https://www.arduino.cc/en/Main/ArduinoBoardUno

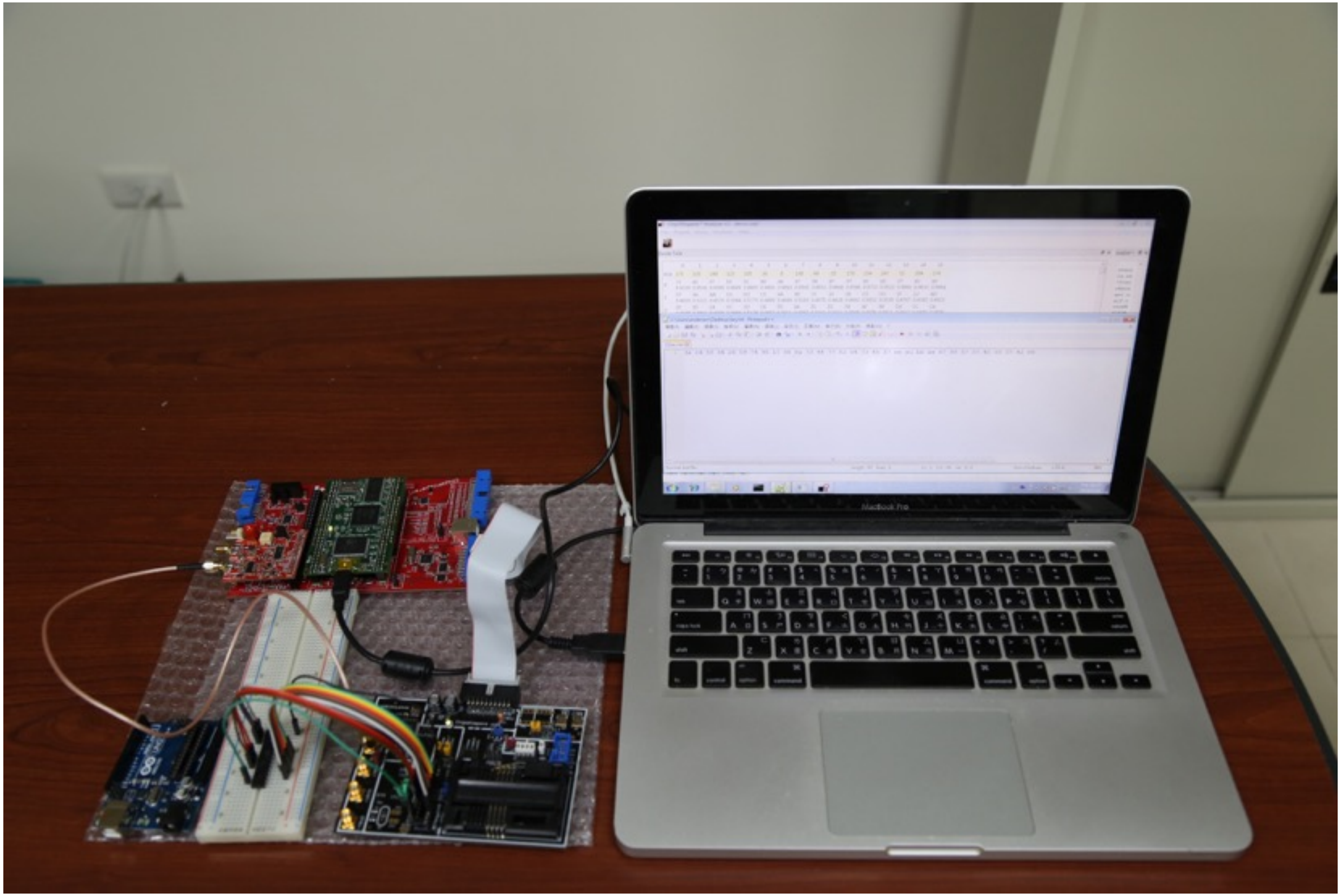
ChipWhisperer

ChipWhisperer board
1. control FPGA
2. OpenADC



MultiTarget board
1. micro controller
2. card socket
3. FPGA

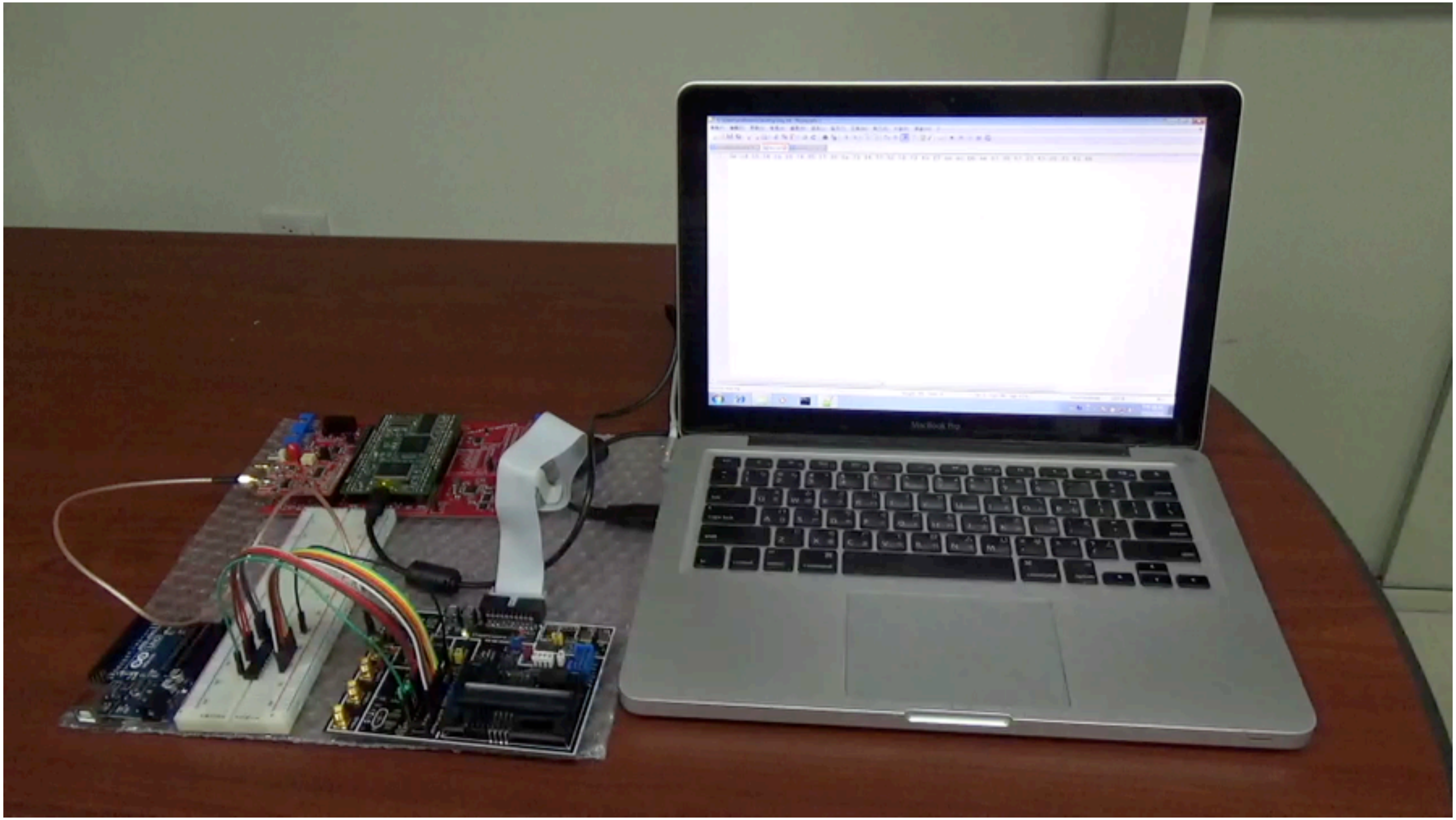
Hardware Implementation



Outlines

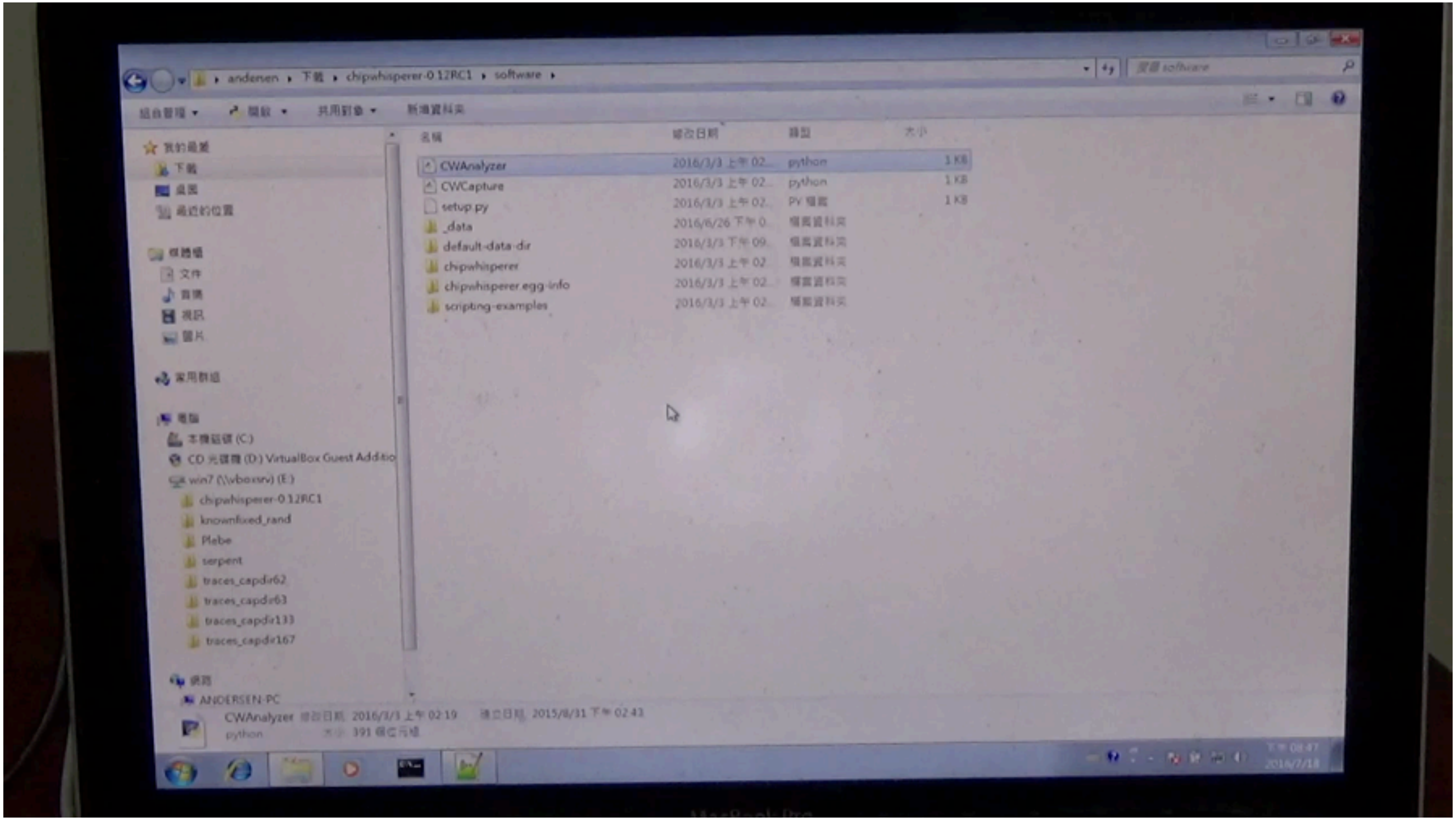
- SCA/DPA/CPA
- Hardware Implementation
- **Demo Video**
- CPA Implementation on AES Rounds
- Countermeasures
- Conclusion

Demo Video



3a cd 58 34 26 59 74 95 17 98 8a 73 44 77 52 54 73 45 f7 ee ec bb ae 67 98 87 07 45 00 37 42 66

Demo Video



3a cd 58 34 26 59 74 95 17 98 8a 73 44 77 52 54 73 45 f7 ee ec bb ae 67 98 87 07 45 00 37 42 66

Outlines

- SCA/DPA/CPA
- Hardware Implementation
- Demo Video
- **CPA Implementation on AES Rounds**
- Countermeasures
- Conclusion

CPA on One Round of AES Encryption

Known Input

12	43	F5	
77	26	54	87
A3	B3	7E	FF
9B	4A	AF	E8

Choose a byte

Key Guess

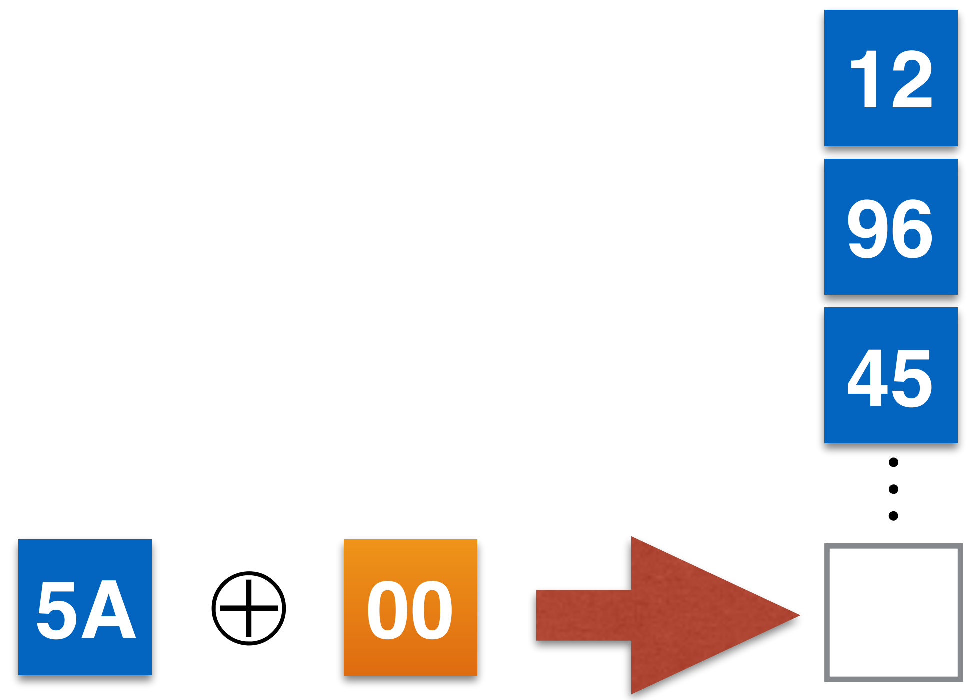
Choose a key guession

00
01
02
03
⋮
FF

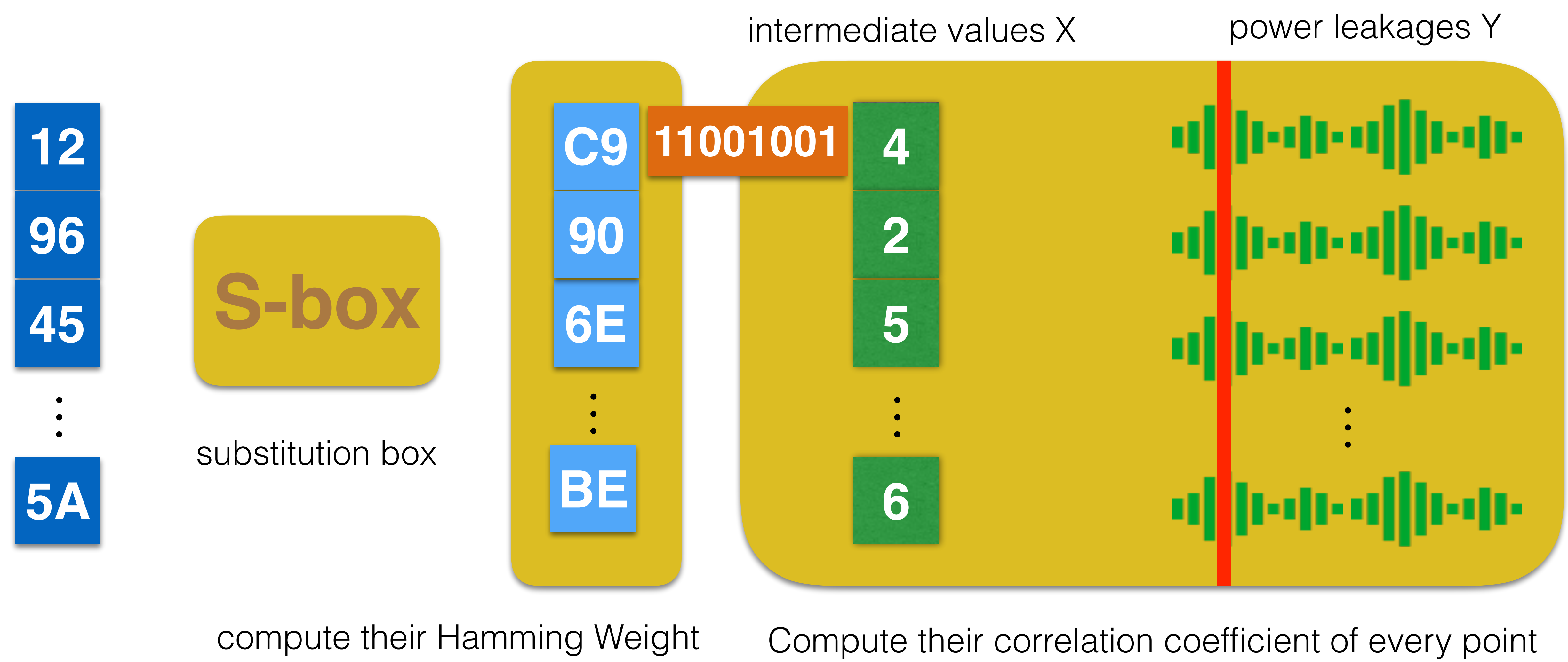
CPA on One Round of AES Encryption

Input N

5A	0C	6C	FC
67	BE	AF	60
42	FF	C3	51
6E	23	0A	A9

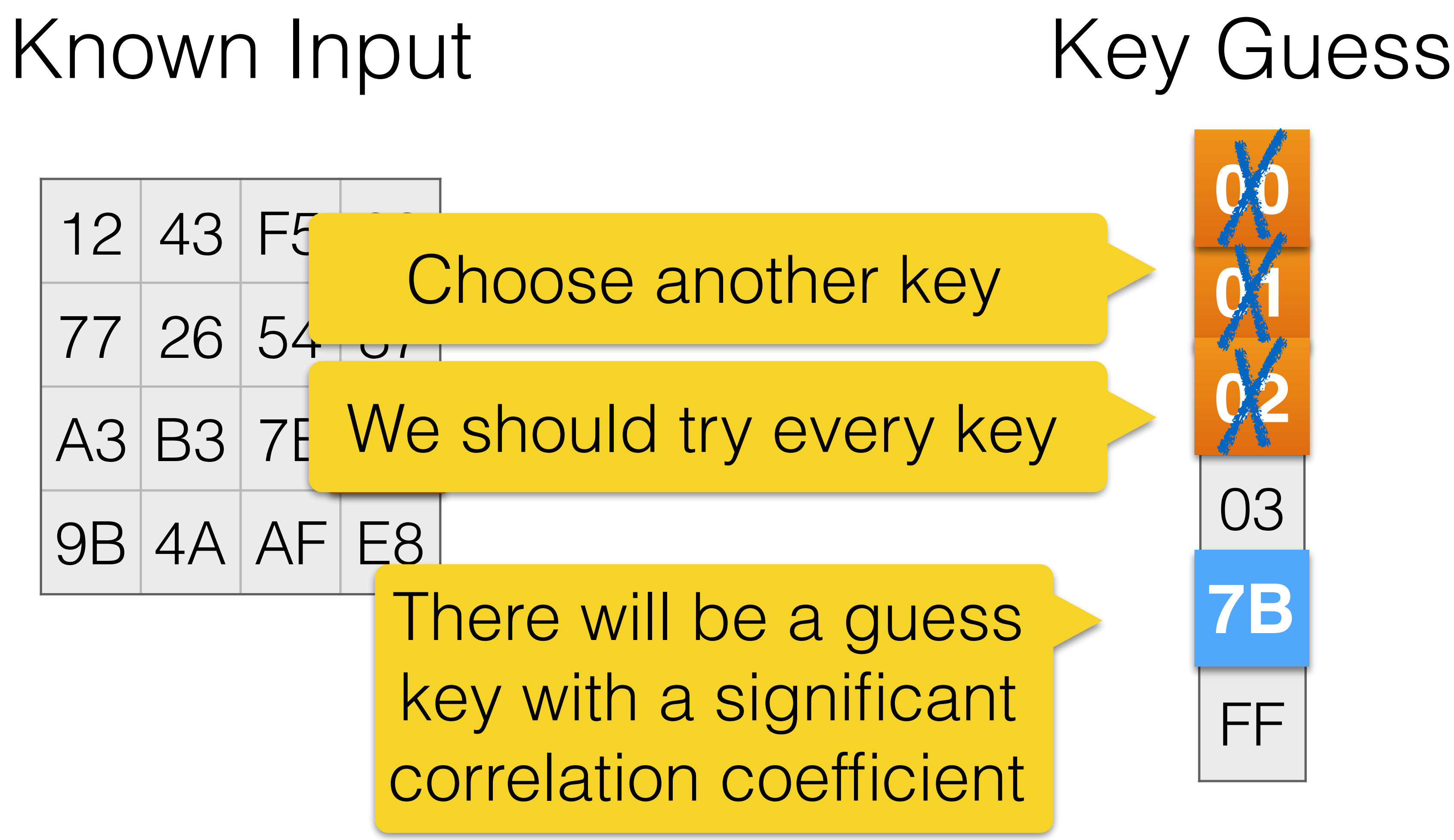


CPA on One Round of AES Encryption



- If there are any points with a significant Correlation Coefficient value, the guessing key might be correct.

CPA on One Round of AES Encryption



Repeat 16 times for each bytes!

Compare AES-256 with AES-128

Similarities:

- Block size is 128 bits, so as Round Key size.

Differences:

- 256-bit Master Key.
- 14 rounds while 10 rounds in AES-128.
  k_0: the first half (128 bits) of master key.
  k_1: the second half (128 bits) of master key.

2b	28	ab	09	a0	88	23	2a	f2	7a	59	73	3d	47	1e	6d
7e	ae	f7	cf	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	80	16	23	7a
15	de	15	4f	fe	2c	39	76	95	b9	80	f6	47	fe	7e	88
16	a6	88	3c	17	b1	39	05	f2	43	7a	7f	7d	3e	44	3b

First Round is enough

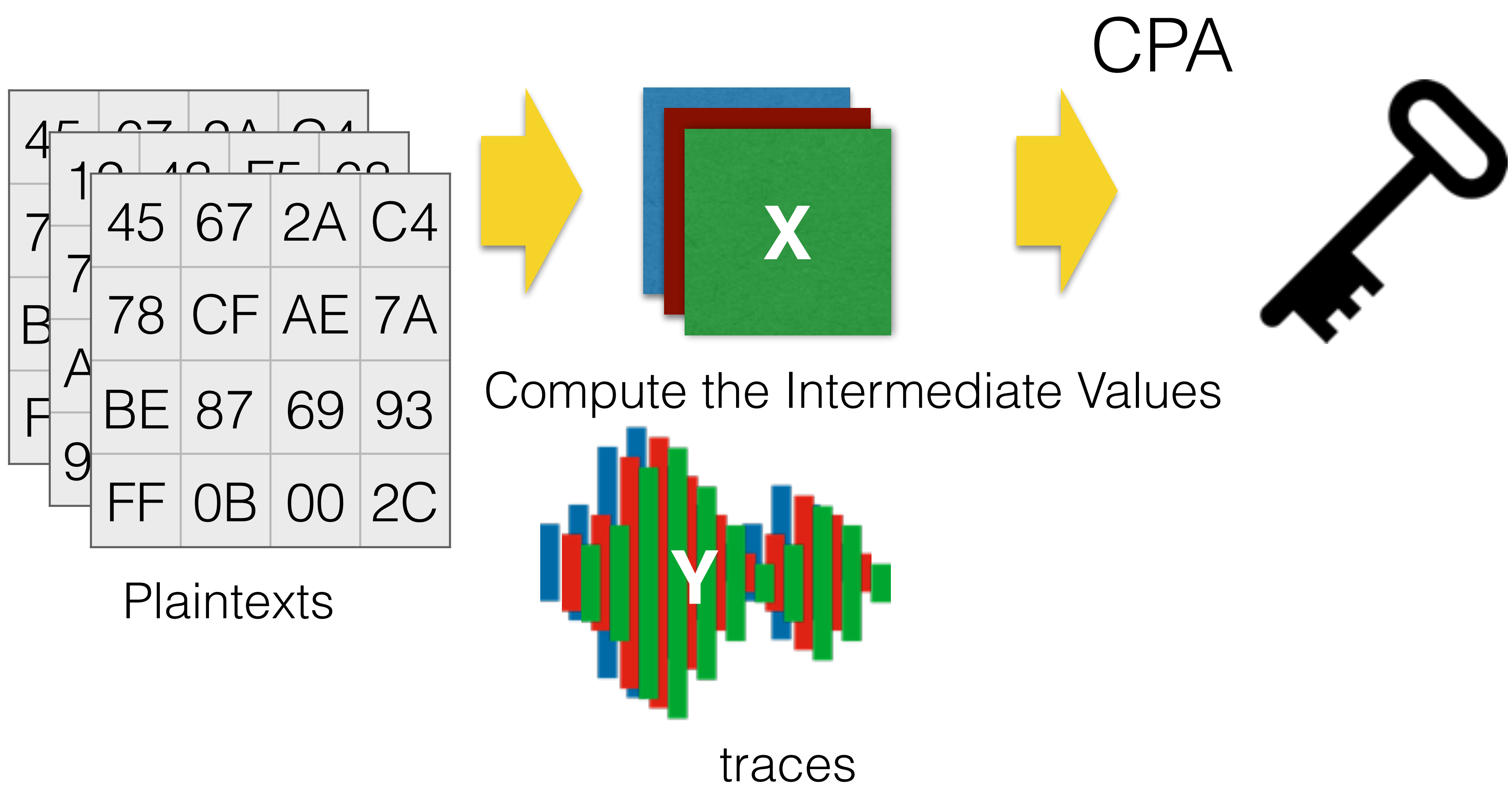
Key Schedule of AES-128

2b	28	ab	09	11	7a	44	4a	f6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3f	75
7e	ae	f7	cf	02	93	8f	93	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	a7
15	de	15	4f	4e	0c	ee	13	51	83	96	d9	7b	77	99	8a
16	a6	88	3c	f6	be	27	86	c0	66	ee	d2	43	fd	da	5c

Needs 2 Rounds

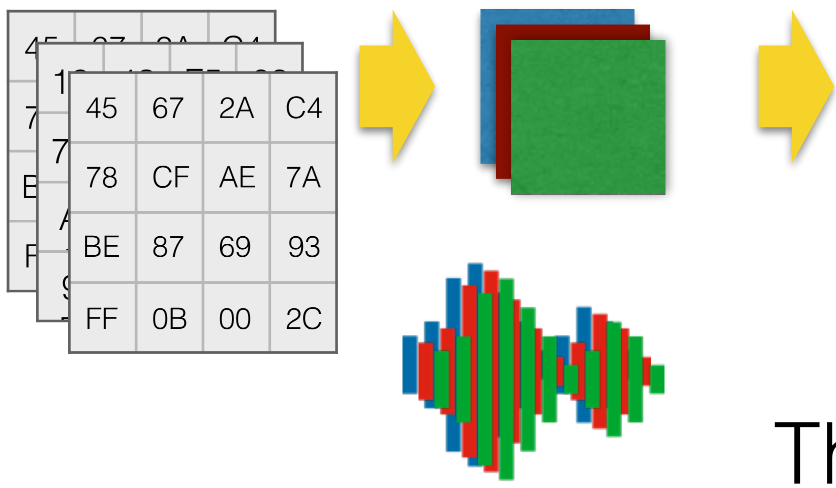
Key Schedule of AES-256

Compare AES-256 with AES-128

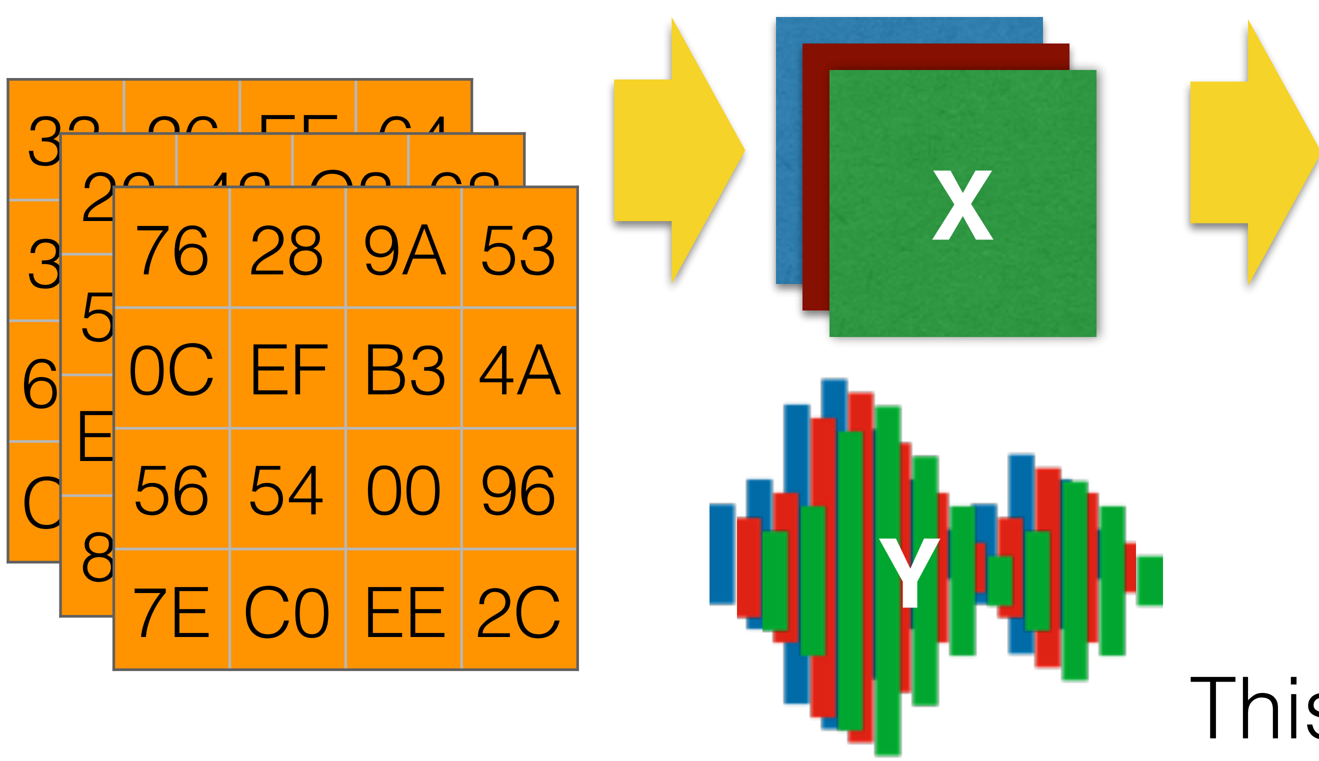


Attacks on AES-128

Compare AES-256 with AES-128



This round key is the first-half key
Use it to compute the input of the next round

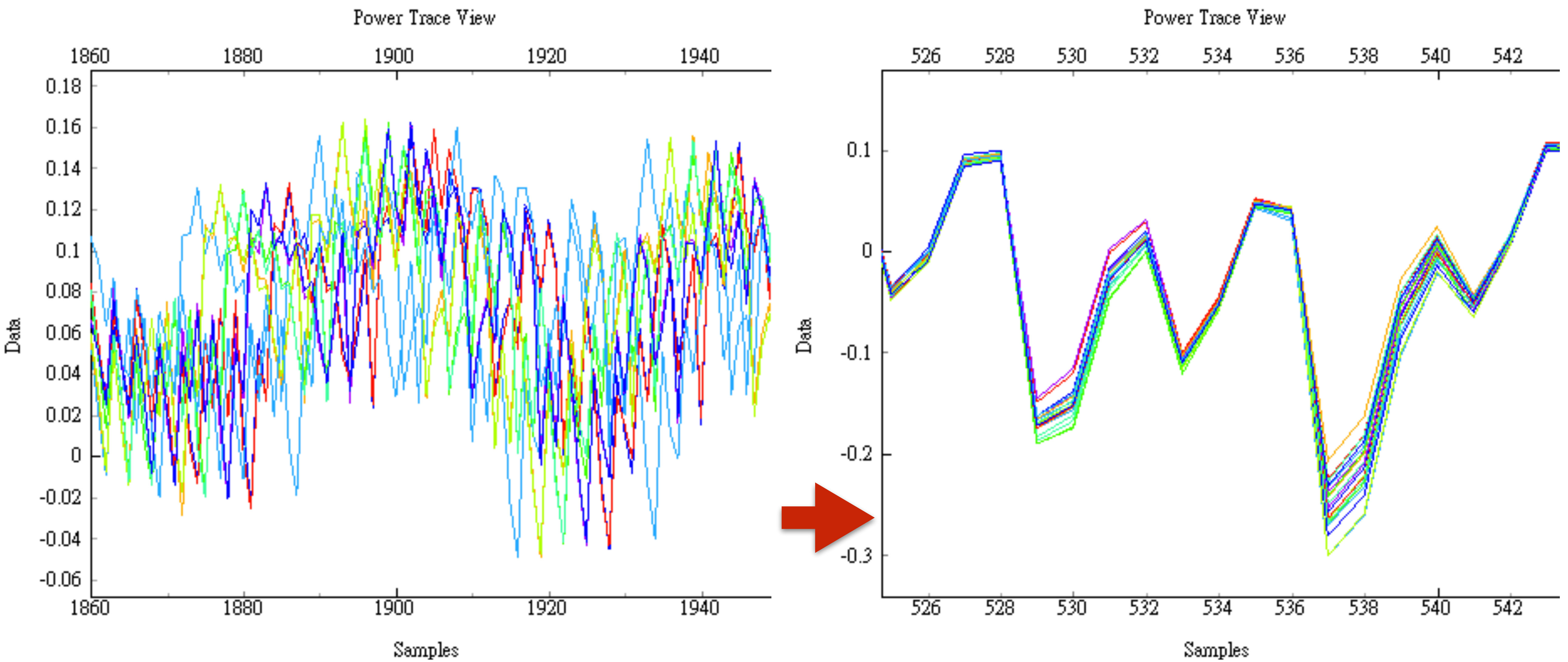


1 Round Encryption Results

traces

This round key is the second-half key

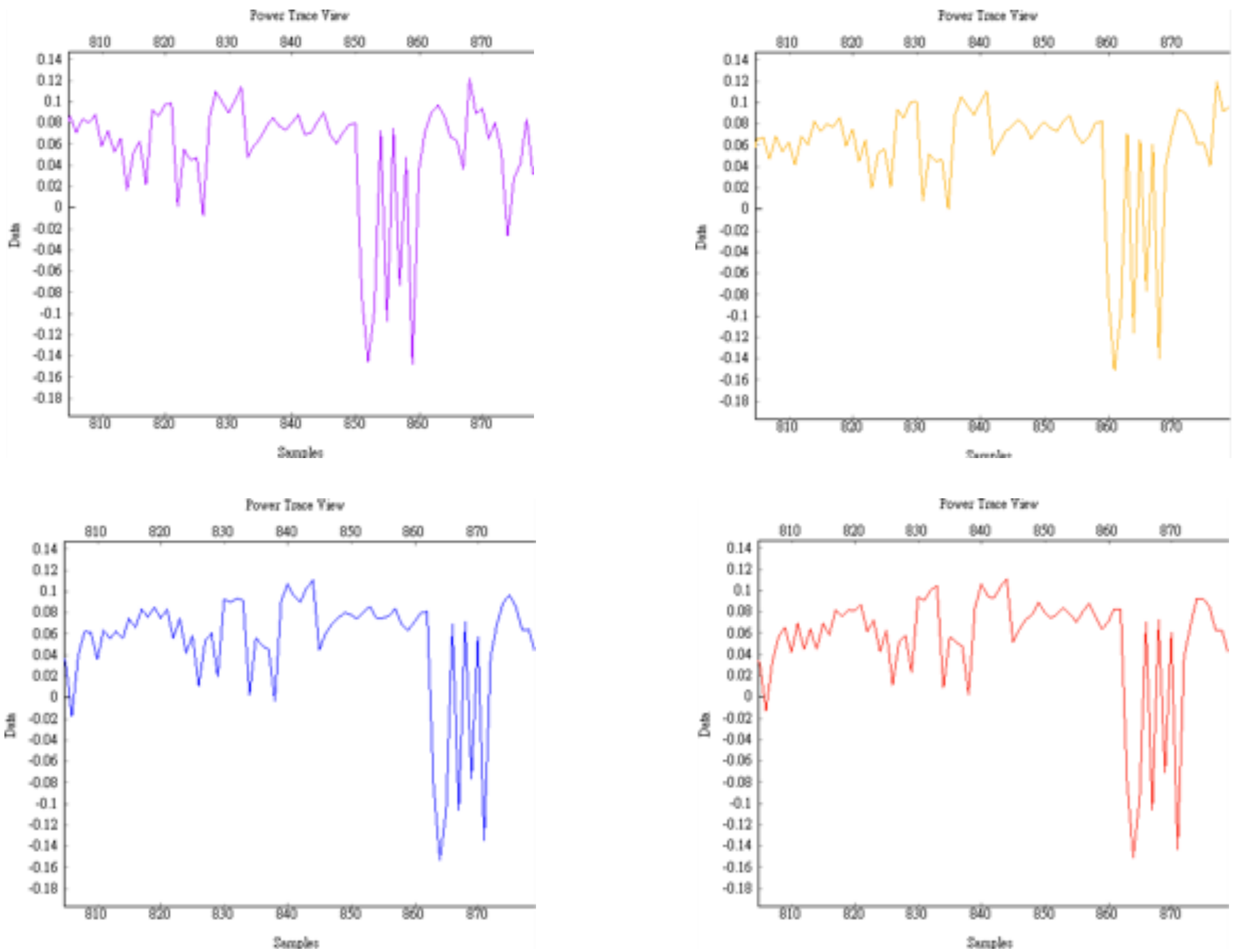
Resynchronization and Alignment



- The variables we concern change ***vertically***.
- Those ***horizontal*** shifts could be disturbances.

Resynchronization and Alignment

- Use some special pattern to align.



Call this special signal $h[n]$

Resynchronization and Alignment

- Method 1: Sum of Absolute Difference (SAD).

$$\text{SAD} = \sum_{i=0}^{N-1} |(h[i] - x[i])|$$

- If two N-points signals are similar, SAD will be small.
- Align the traces by ***minimizing*** the SAD.

Resynchronization and Alignment

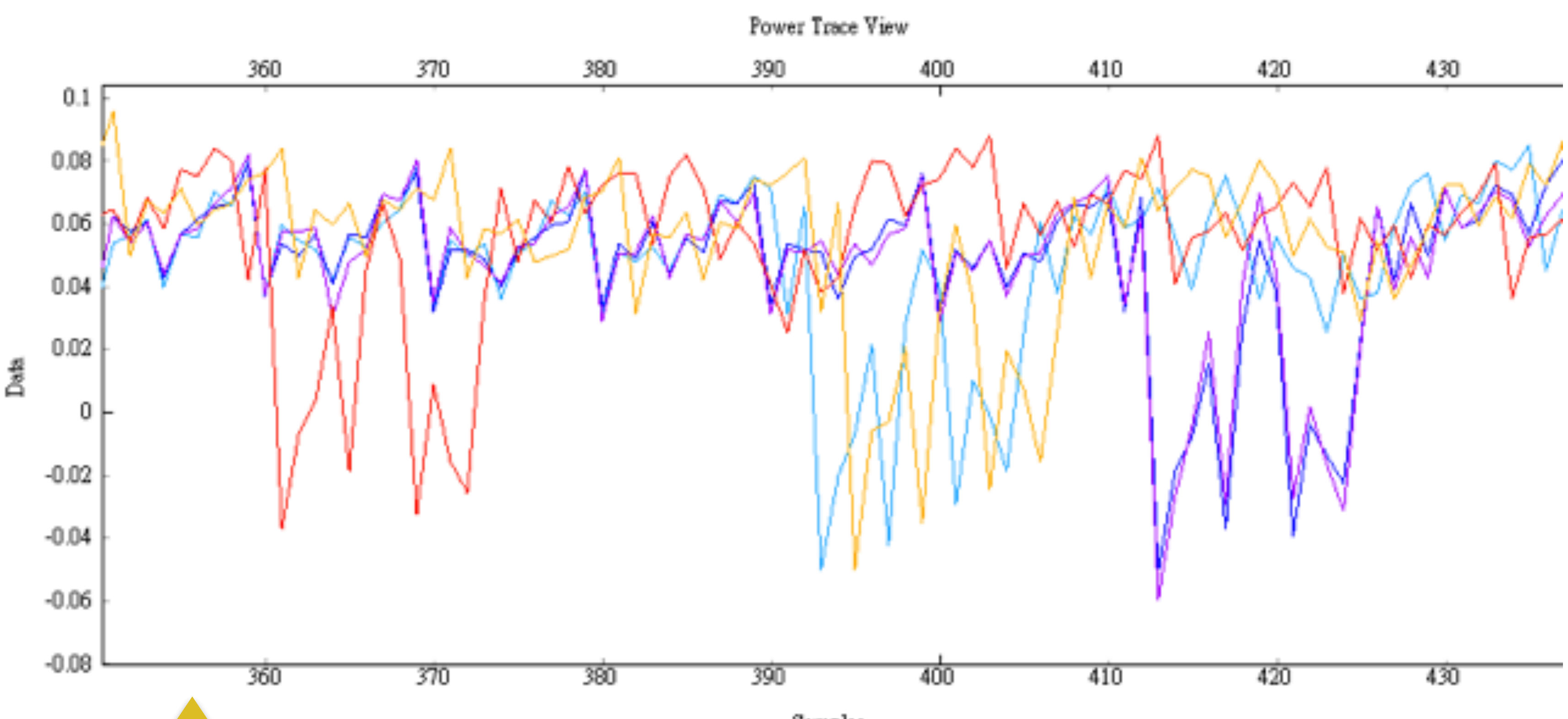
- Method 2: Correlation based method.

$$\mathrm{Cor}(h,x) = \frac{\sum_{i=0}^{N-1}(h[i]-\bar{h})(x[i]-\bar{x})}{\sqrt{\sum_{i=0}^{N-1}(h[i]-\bar{h})^2}\cdot\sqrt{\sum_{i=0}^{N-1}(x[i]-\bar{x})^2}}$$

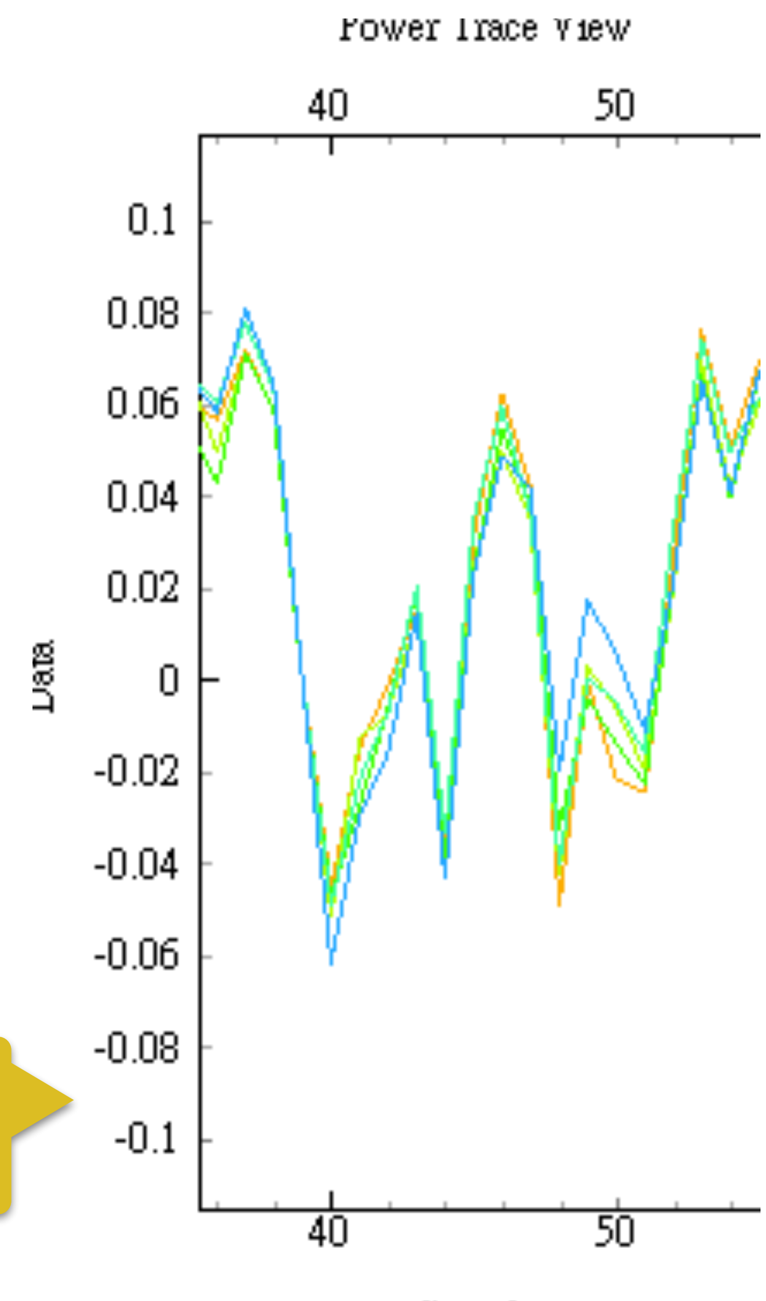
- If two N-points signals are similar, correlation coefficient will near to 1.

- Align the traces by ***maximizing*** the correlation coefficient.

Resynchronization and Alignment



Before resynchronization



After resynchronization

Outlines

- SCA/DPA/CPA
- Hardware Implementation
- Demo Video
- CPA Implementation on AES Rounds
- **Countermeasures**
- Conclusion

CHES 2016 CTF


chesworkshop.org

CHES 2016

CHES Home
CHES 2016
PROGRAM
STUDENT STIPENDS
ACCEPTED PAPERS
CALL FOR PAPERS
CHES CHALLENGE
CALL FOR POSTERS
CALL FOR TUTORIALS
CALL FOR SPONSORS
LOCAL INFORMATION
REGISTRATION

Welcome

Welcome to the home page of the 18th Conference on Cryptographic Hardware and Embedded Systems 2016. CHES 2016 will take place Santa Barbara, CA, USA, from Wednesday, August 17th - Friday, August 19th, 2016. The annual CHES conference highlights new results in the design and analysis of cryptographic hardware and software implementations. CHES provides a valuable connection between the research and cryptographic engineering communities and attracts participants from industry, academia, and government organizations.

News

Jul 14, 2016: Poster Submission is still open until **July 31**.
Jun 26, 2016: The Conference Program is now available.
Jun 23, 2016: Registration is now open. Early registration ends on **July 31, 2016**.
Jun 16, 2016: Stipends for students: Please send your application email to the General Chairs.
Jun 13, 2016: The list of Accepted Papers is now online.
Jun 3, 2016: Interested in supporting CHES? See our Call for Sponsors.
Jun 2. 2016: Call for Posters and Tutorials is now available.

PLATINUM SPONSOR
Rambus Cryptography Research
GOLD SPONSORS
riscure
SPONSORS
ALPhA NOV
eshard
NewAE Technology

http://www.chesworkshop.org/ches2016/start.php

CHES 2016 CTF

ctf.newae.com

CHES 2016 Capture the Flag | Scoreboard | Challenges | FAQ/Rules | Your Submissions | Hardware Lab | Sign out

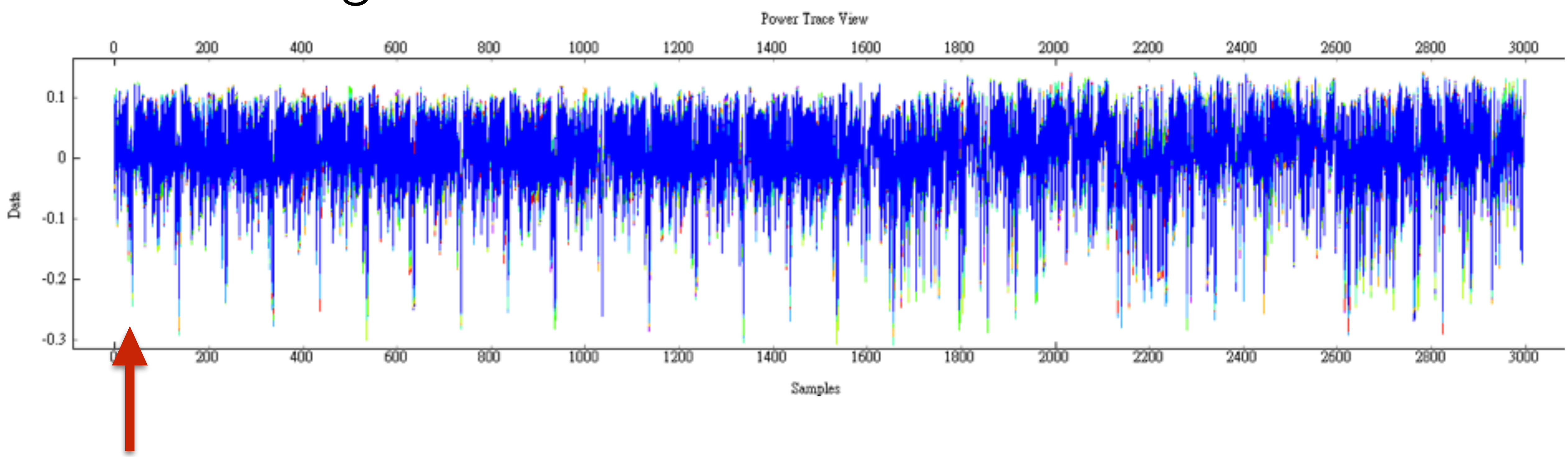
Challenges

Name	Added	Trace File	Firmware	Source Code	Attacks	Owner	Note
Stagegate #1	1 month, 2 weeks ago	Download	Download	Download	1st [10 pts]: 팀당근 2nd [10 pts]: rboix 3rd [10 pts]: OverTime	robot	A very straight-forward AES-128 implementation written in C. Standard CPA attack should work.
Whitebox Crypto #1	1 month, 2 weeks ago	Not Found	Download	Download	1st [100 pts]: esanfelix 2nd [100 pts]: 팀당근 3rd [100 pts]: OverTime	robot	A whitebox AES-128, with provided C source and Linux binary.
Stagegate #2	1 month, 2 weeks ago	Download	Download	Download	1st [10 pts]: esanfelix 2nd [10 pts]: OverTime 3rd [10 pts]: rboix	robot	AES-128 in C with a tiny bit of random jitter before the encryption happens.
Stagegate #3B (Hard)	1 month, 2 weeks ago	Download	Download	Download	1st [10 pts]: OverTime 2nd [10 pts]: 팀당근 3rd [10 pts]: esanfelix	robot	AES-128 in C with additional pseudo-random S-Box lookups. Hard because trace does not cover entire S-Box operation.
AES-RSI	1 month, 1 week ago	Download	Download	Download	1st [50 pts]: Rauf 2nd [30 pts]: SICADA 3rd [10 pts]: Nabil Hamzi	dpalab	AES with Random Starting Index shuffling countermeasure on SubBytes

https://ctf.newae.com/flags/

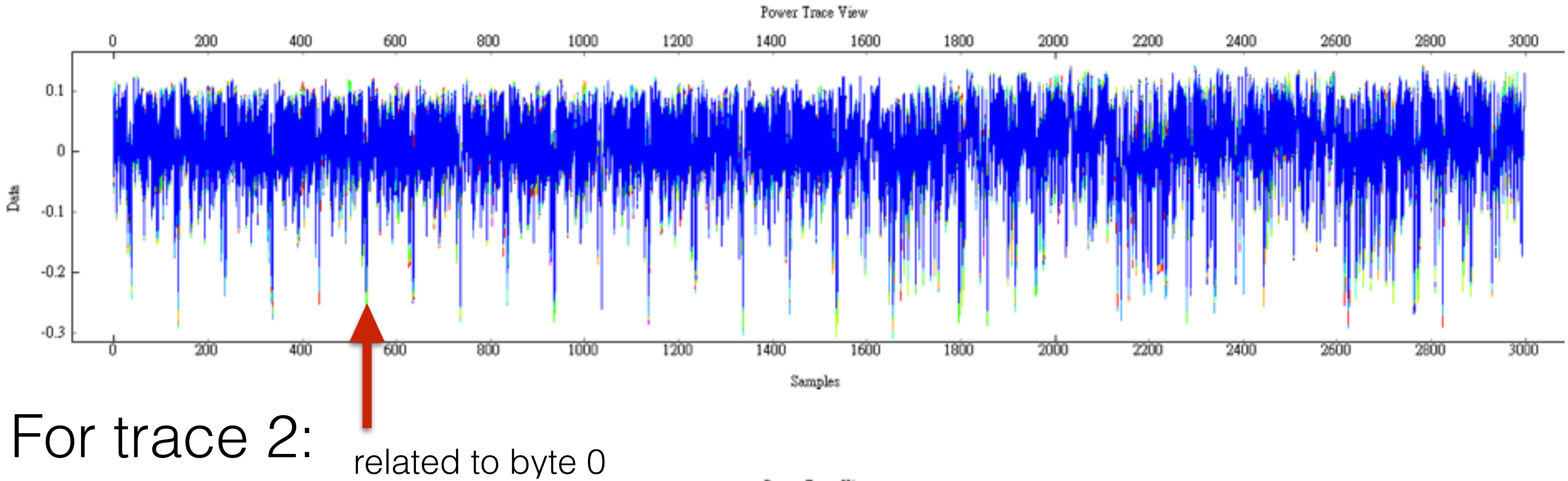
Countermeasure (1)

- Shuffling:

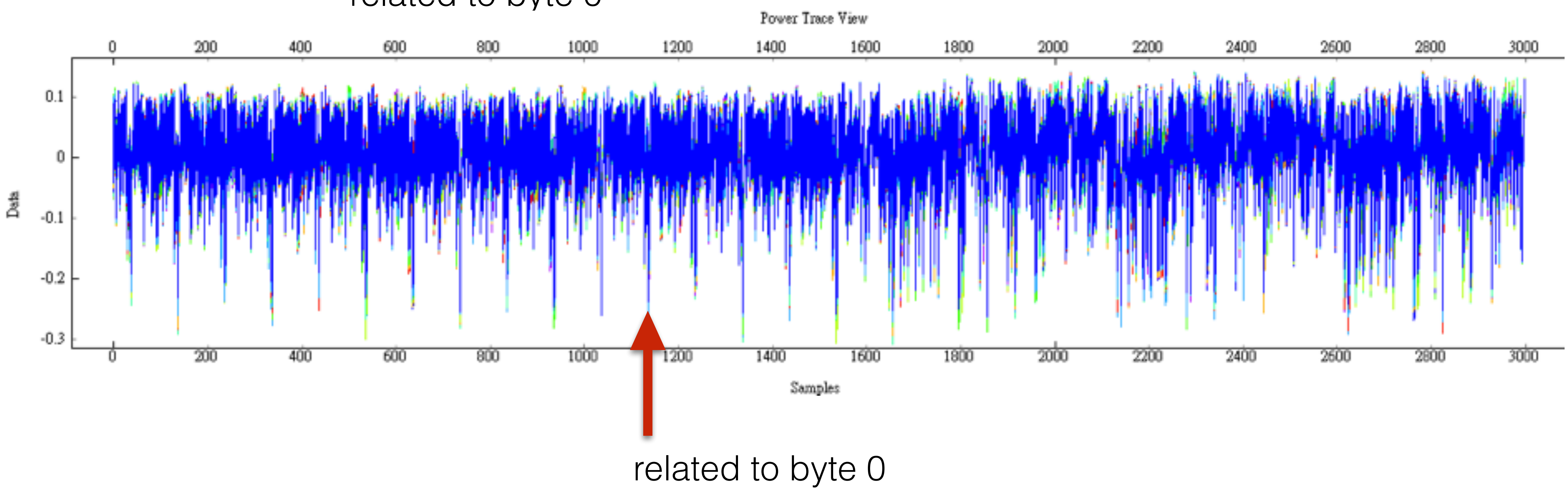


related to byte 0

For trace 1:

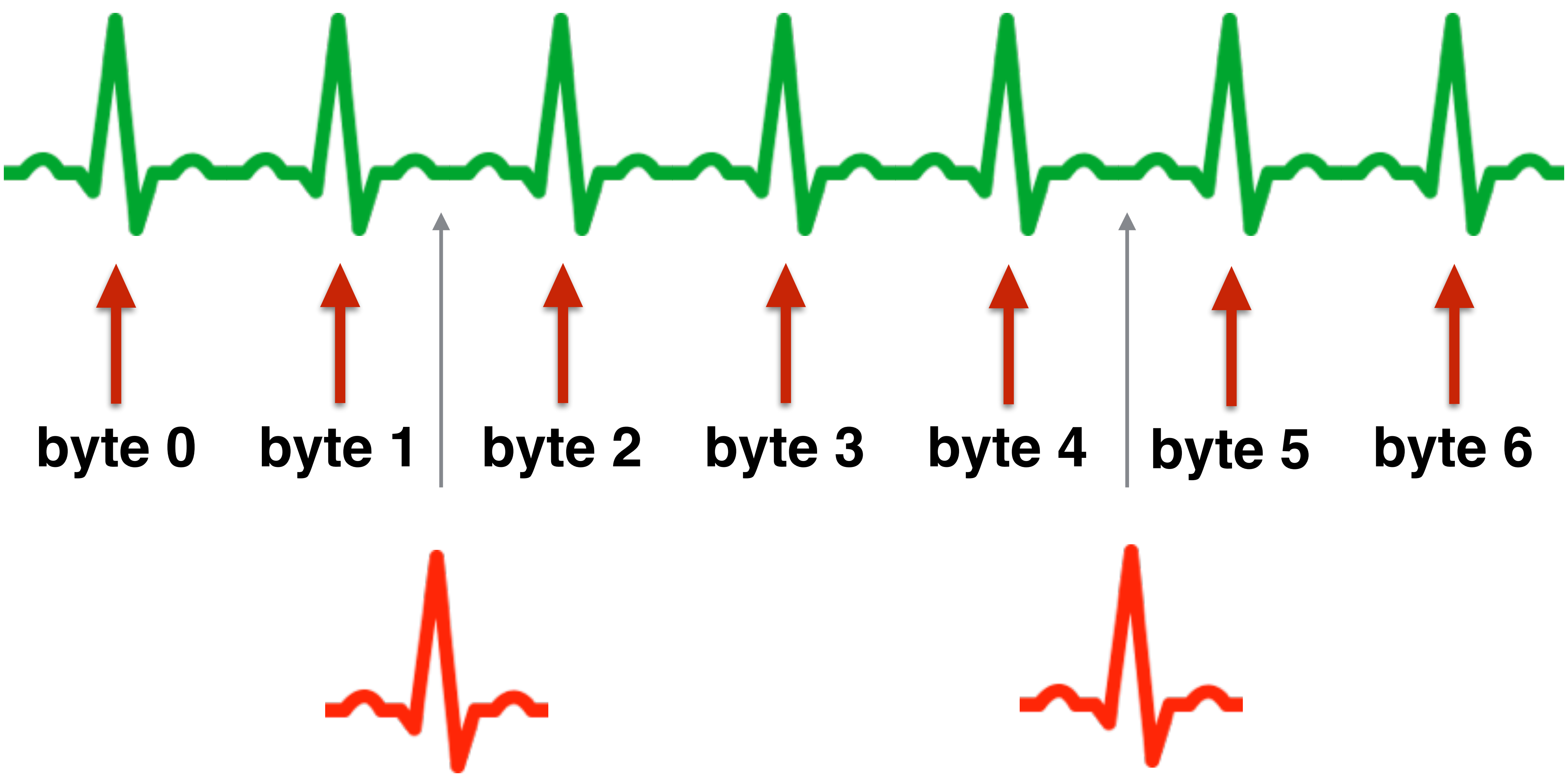


For trace 2:

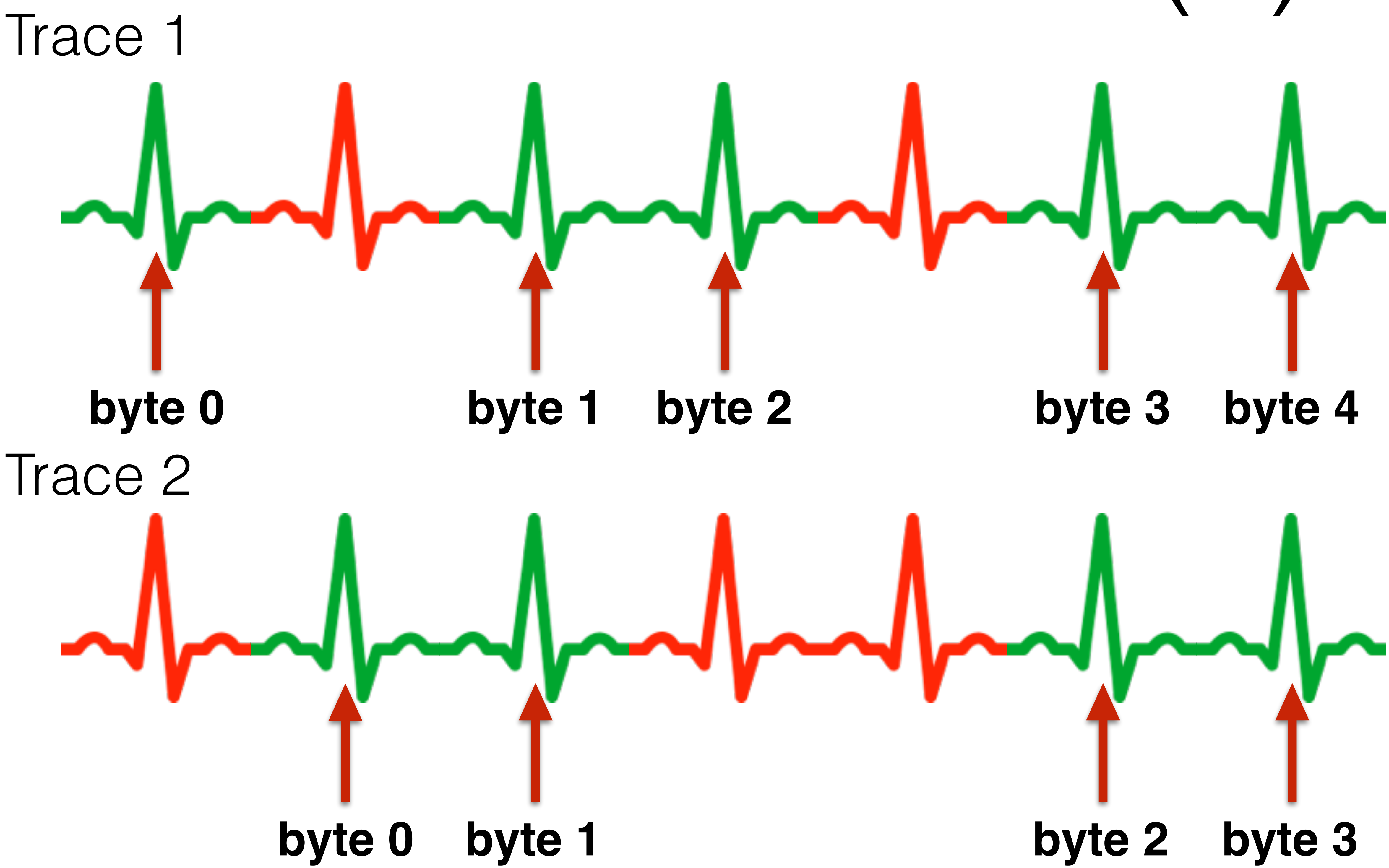


Countermeasure (2)

- Adding Dummy:



Countermeasure (1)



Outlines

- SCA/DPA/CPA
- Hardware Implementation
- Demo Video
- CPA Implementation on AES Rounds
- Countermeasures
- **Conclusion**

Conclusion

- With more statistical techniques applied, SCA is more powerful than ever.
- Encryption systems could be insecure without any protections from SCA.
- SCA protections should be taken into account when using microcontrollers like ATmega328P and their applications in IoT.

Reference

- S.Mangard *et al*. Power Analysis Attacks.
- Colin O'flynn ChipWhisperer.
  http://www.newae.com/sidechannel/cwdocs/
- CHES CTF 2016
  https://ctf.newae.com
- Papers from CHES, Eurocrypt, Crypto and Asiacrypt
- Arduino
  https://www.arduino.cc
- Atmel
  http://www.atmel.com