Template attacks on implementations of cryptographic algorithms

SOFA 2020

Brice Colombier

Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, Grenoble, France

`https://bcolombier.fr`

2020-11-27

Introduction

Brice Colombier

Associate professor at Grenoble INP, Grenoble, France

Research topics:

- Electronics design IP protection,
- Hardware security,
- Physical attacks:
  - Active: fault attacks.
  - Passive: side-channel analysis/attacks

Introduction

Brice Colombier

Associate professor at Grenoble INP, Grenoble, France

Research topics:

- Electronics design IP protection,
- Hardware security,
- Physical attacks:
    - Active: fault attacks.
    - Passive: side-channel analysis/attacks: our topic today!

Symmetric cryptography

Confidentiality

Cryptography aims at delivering several properties, such as:

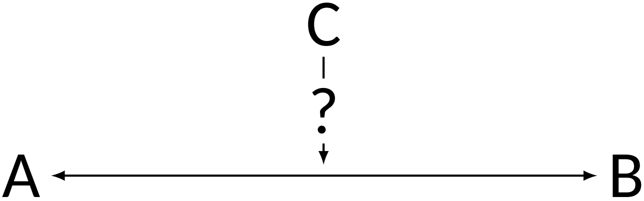
- integrity,
- authenticity,
- confidentiality

Confidentiality

Cryptography aims at delivering several properties, such as:

- integrity,
- authenticity,
- confidentiality

Confidentiality:
Parties A and B can communicate without party C understanding.

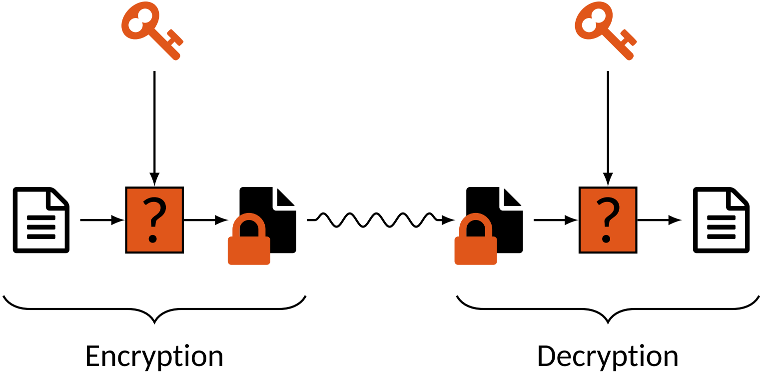


Symmetric cryptography

The message is encrypted by A and decrypted by B.

The same key is used for encryption and decryption.

By obtaining the key, we break the confidentiality.



AES: Advanced Encryption Standard

The Rjindael block cipher [1] was standardized by NIST in 2001.
It is now referred to as AES (Advanced Encryption Standard).

A block cipher operates on blocks of data.

AES-128 [2] operates with:

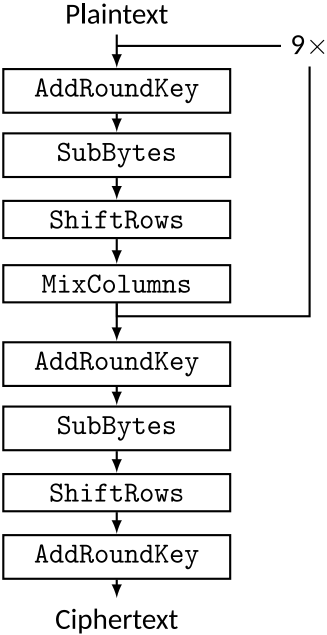
- a 128-bit key,
- on 128-bit blocks.

---

[1] J. Daemen and V. Rijmen. “Rijndael for AES”. *The Third Advanced Encryption Standard Candidate Conference*. New York, USA: National Institute of Standards and Technology, Apr. 2000, pp. 343–348.

[2] AES-192 and AES-256 exist too but are not covered here

AES internals



Plaintext

9×

`AddRoundKey`

`SubBytes`

`ShiftRows`

`MixColumns`

`AddRoundKey`

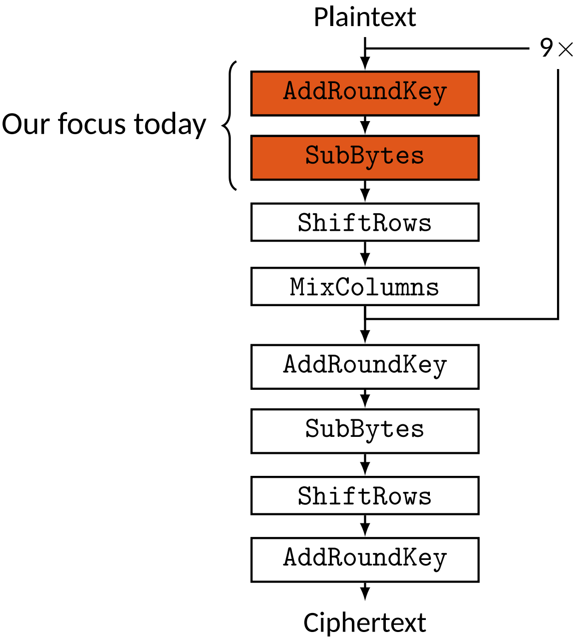
`SubBytes`

`ShiftRows`

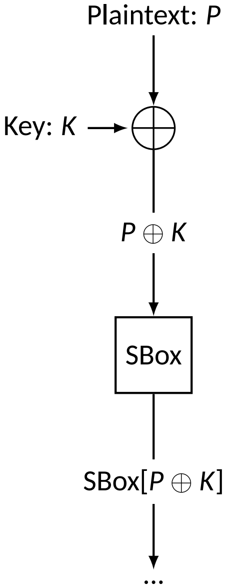
`AddRoundKey`

Ciphertext

AES internals



AES first transformations



Plaintext: P

Key: K

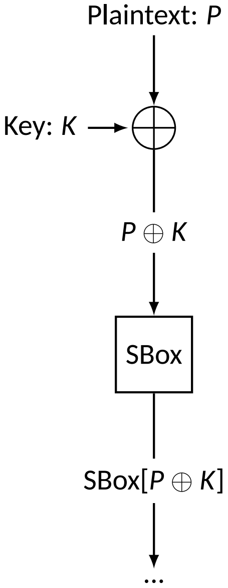
$P \oplus K$

SBox

SBox$[P \oplus K]$

...

AES first transformations



Plaintext: P

Key: K

$P \oplus K$

SBox

$\text{SBox}[P \oplus K]$

...

SBox is an $\{0,1\}^8 \to \{0,1\}^8$ substitution table.

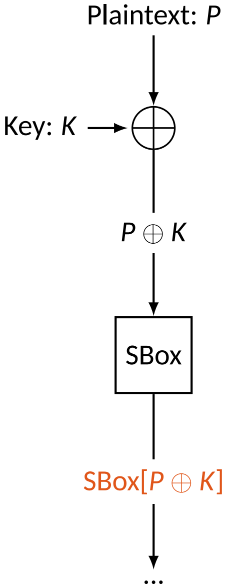
AES S-box

	00	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**0a**	**0b**	**0c**	**0d**	**0e**	**0f**
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

The column is determined by the least significant nibble, and the row by the most significant nibble. For example, the value $9a_{16}$ is converted into $b8_{16}$.

`https://en.wikipedia.org/wiki/Rijndael_S-box`

Target



SBox mapping is **known** and **reversible**.
We assume the **plaintext is known too**.

We want the key!

Intermediate value computation

AES is byte-oriented: the state is a 4 × 4 matrix of bytes.

$$\boxed{\begin{array}{cccc} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{array}}$$

Our target intermediate value is in fact split into 16 bytes

$$\boxed{\begin{array}{cccc} SBox[p_0 \oplus k_0] & SBox[p_4 \oplus k_4] & SBox[p_8 \oplus k_8] & SBox[p_{12} \oplus k_{12}] \\ SBox[p_1 \oplus k_1] & SBox[p_5 \oplus k_5] & SBox[p_9 \oplus k_9] & SBox[p_{13} \oplus k_{13}] \\ SBox[p_2 \oplus k_2] & SBox[p_6 \oplus k_6] & SBox[p_{10} \oplus k_{10}] & SBox[p_{14} \oplus k_{14}] \\ SBox[p_3 \oplus k_3] & SBox[p_7 \oplus k_7] & SBox[p_{11} \oplus k_{11}] & SBox[p_{15} \oplus k_{15}] \end{array}}$$

We will divide and conquer and recover the 128-bit key byte by byte.

Side-channel attacks

Side-channel attacks

Side-channel attacks principle

Physical quantities measured on the device
depend on the data the device handles.

Examples of physical quantities:

- power consumption,
- electromagnetic radiations,
- sound,
- photonic emissions.

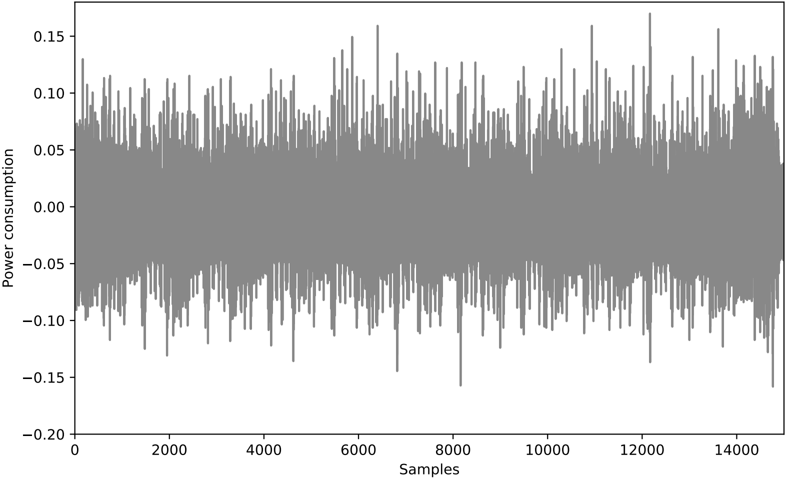
Actual setup

A microcontroller runs multiple AES encryptions.

We put an electromagnetic probe above it and record the electromagnetic field.

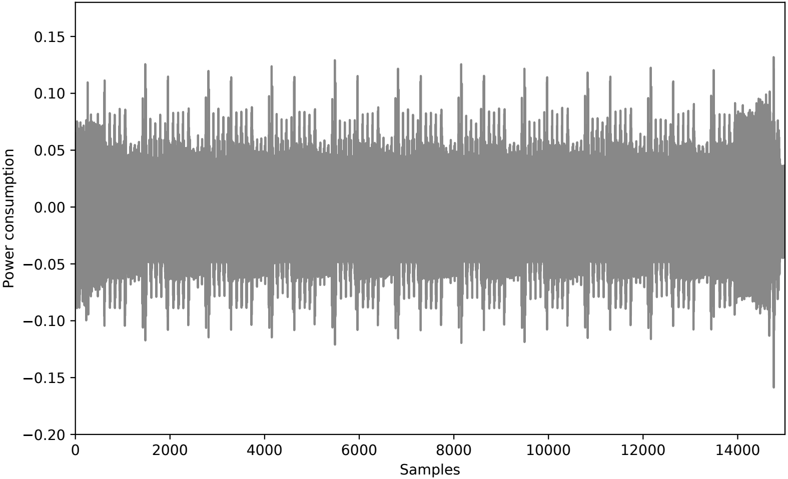
Examples of measurements

First, one measurement



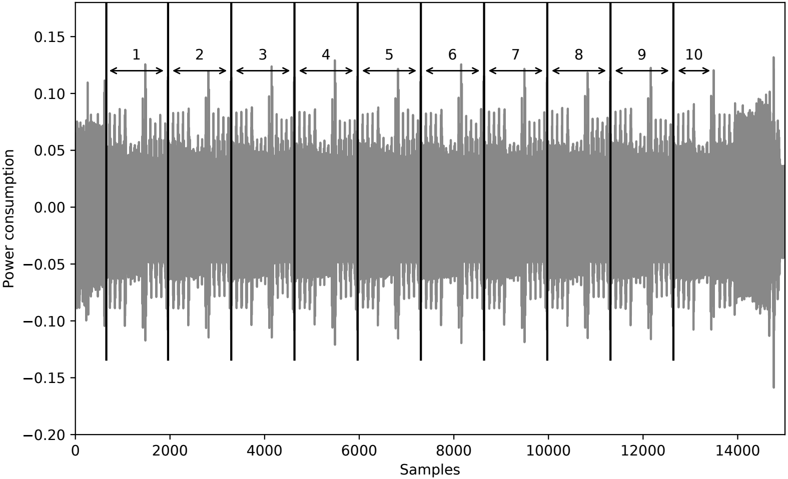
Examples of measurements

Averaging 50 identical measurements (denoising)



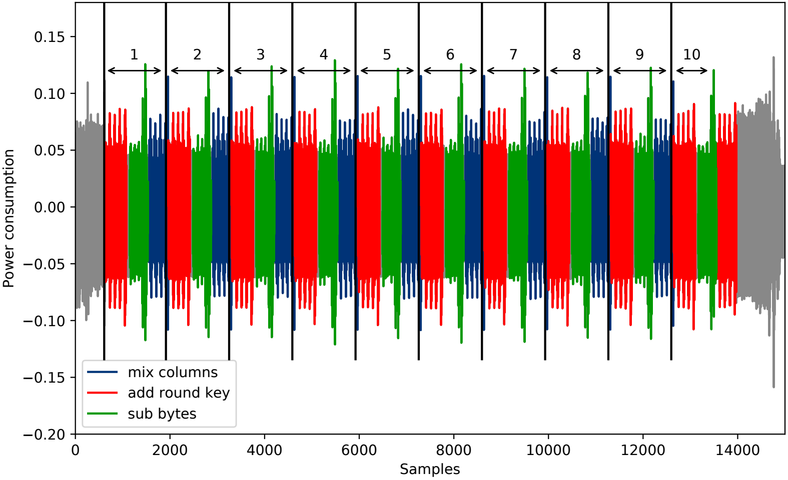
Examples of measurements

AES rounds are visible



Examples of measurements

AES transformations are visible within rounds



Theory of template attacks

History

Template attacks were introduced in 2002 [3].

The information leakage can be modeled as a Gaussian distribution. This is **fully described** by the following parameters:

- the mean: μ
- the variance: σ^2

A template is the (μ, σ^2) pair.

A template attack follows a two-step process:

- profiling phase,
- matching phase.

---

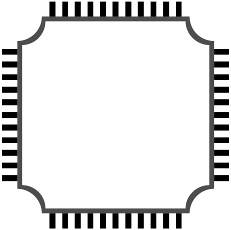
[3] S. Chari, J. R. Rao, and P. Rohatgi. "Template Attacks". *CHES*. 2002, pp. 13–28.

Profiling phase

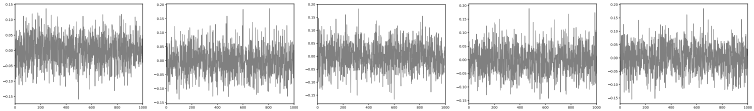
Aim:
build a template (μ, σ^2) for every intermediate value $\in \{0, \ldots, 255\}$.

We do this on an open device:

- we control the inputs: **key** K and **plaintext** P.
- we know the intermediate value of interest: $\text{SBox}[p_i \oplus k_i]$
- we can perform side-channel measurements on it.

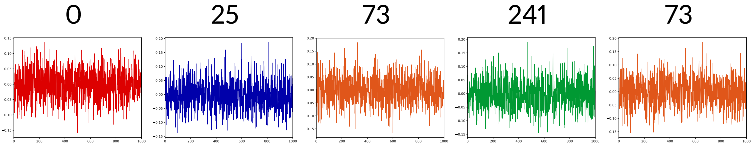


Profiling phase



Profiling phase

Intermediate value $\mathrm{SBox}[p_i \oplus k_i] =$



We build 256 sets of traces, according to the intermediate value.

$\mathcal{T}_i$ is the set for which the intermediate value is equal to i.

$\mathcal{T}_{0}$	$\mathcal{T}_{1}$	$\mathcal{T}_{2}$	$\mathcal{T}_{3}$	$\mathcal{T}_{4}$	$\mathcal{T}_{5}$	$\mathcal{T}_{6}$	$\mathcal{T}_{7}$	$\mathcal{T}_{8}$	$\mathcal{T}_{9}$	$\mathcal{T}_{10}$	$\mathcal{T}_{11}$	$\mathcal{T}_{12}$	$\mathcal{T}_{13}$	$\mathcal{T}_{14}$	$\mathcal{T}_{15}$
$\mathcal{T}_{16}$	$\mathcal{T}_{17}$	$\mathcal{T}_{18}$	$\mathcal{T}_{19}$	$\mathcal{T}_{20}$	$\mathcal{T}_{21}$	$\mathcal{T}_{22}$	$\mathcal{T}_{23}$	$\mathcal{T}_{24}$	$\mathcal{T}_{25}$	$\mathcal{T}_{26}$	$\mathcal{T}_{27}$	$\mathcal{T}_{28}$	$\mathcal{T}_{29}$	$\mathcal{T}_{30}$	$\mathcal{T}_{31}$
$\mathcal{T}_{32}$	$\mathcal{T}_{33}$	$\mathcal{T}_{34}$	$\mathcal{T}_{35}$	$\mathcal{T}_{36}$	$\mathcal{T}_{37}$	$\mathcal{T}_{38}$	$\mathcal{T}_{39}$	$\mathcal{T}_{40}$	$\mathcal{T}_{41}$	$\mathcal{T}_{42}$	$\mathcal{T}_{43}$	$\mathcal{T}_{44}$	$\mathcal{T}_{45}$	$\mathcal{T}_{46}$	$\mathcal{T}_{47}$
$\mathcal{T}_{48}$	$\mathcal{T}_{49}$	$\mathcal{T}_{50}$	$\mathcal{T}_{51}$	$\mathcal{T}_{52}$	$\mathcal{T}_{53}$	$\mathcal{T}_{54}$	$\mathcal{T}_{55}$	$\mathcal{T}_{56}$	$\mathcal{T}_{57}$	$\mathcal{T}_{58}$	$\mathcal{T}_{59}$	$\mathcal{T}_{60}$	$\mathcal{T}_{61}$	$\mathcal{T}_{62}$	$\mathcal{T}_{63}$
$\mathcal{T}_{64}$	$\mathcal{T}_{65}$	$\mathcal{T}_{66}$	$\mathcal{T}_{67}$	$\mathcal{T}_{68}$	$\mathcal{T}_{69}$	$\mathcal{T}_{70}$	$\mathcal{T}_{71}$	$\mathcal{T}_{72}$	$\mathcal{T}_{73}$	$\mathcal{T}_{74}$	$\mathcal{T}_{75}$	$\mathcal{T}_{76}$	$\mathcal{T}_{77}$	$\mathcal{T}_{78}$	$\mathcal{T}_{79}$
$\mathcal{T}_{80}$	$\mathcal{T}_{81}$	$\mathcal{T}_{82}$	$\mathcal{T}_{83}$	$\mathcal{T}_{84}$	$\mathcal{T}_{85}$	$\mathcal{T}_{86}$	$\mathcal{T}_{87}$	$\mathcal{T}_{88}$	$\mathcal{T}_{89}$	$\mathcal{T}_{90}$	$\mathcal{T}_{91}$	$\mathcal{T}_{92}$	$\mathcal{T}_{93}$	$\mathcal{T}_{94}$	$\mathcal{T}_{95}$
$\mathcal{T}_{96}$	$\mathcal{T}_{97}$	$\mathcal{T}_{98}$	$\mathcal{T}_{99}$	$\mathcal{T}_{100}$	$\mathcal{T}_{101}$	$\mathcal{T}_{102}$	$\mathcal{T}_{103}$	$\mathcal{T}_{104}$	$\mathcal{T}_{105}$	$\mathcal{T}_{106}$	$\mathcal{T}_{107}$	$\mathcal{T}_{108}$	$\mathcal{T}_{109}$	$\mathcal{T}_{110}$	$\mathcal{T}_{111}$
$\mathcal{T}_{112}$	$\mathcal{T}_{113}$	$\mathcal{T}_{114}$	$\mathcal{T}_{115}$	$\mathcal{T}_{116}$	$\mathcal{T}_{117}$	$\mathcal{T}_{118}$	$\mathcal{T}_{119}$	$\mathcal{T}_{120}$	$\mathcal{T}_{121}$	$\mathcal{T}_{122}$	$\mathcal{T}_{123}$	$\mathcal{T}_{124}$	$\mathcal{T}_{125}$	$\mathcal{T}_{126}$	$\mathcal{T}_{127}$
$\mathcal{T}_{128}$	$\mathcal{T}_{129}$	$\mathcal{T}_{130}$	$\mathcal{T}_{131}$	$\mathcal{T}_{132}$	$\mathcal{T}_{133}$	$\mathcal{T}_{134}$	$\mathcal{T}_{135}$	$\mathcal{T}_{136}$	$\mathcal{T}_{137}$	$\mathcal{T}_{138}$	$\mathcal{T}_{139}$	$\mathcal{T}_{140}$	$\mathcal{T}_{141}$	$\mathcal{T}_{142}$	$\mathcal{T}_{143}$
$\mathcal{T}_{144}$	$\mathcal{T}_{145}$	$\mathcal{T}_{146}$	$\mathcal{T}_{147}$	$\mathcal{T}_{148}$	$\mathcal{T}_{149}$	$\mathcal{T}_{150}$	$\mathcal{T}_{151}$	$\mathcal{T}_{152}$	$\mathcal{T}_{153}$	$\mathcal{T}_{154}$	$\mathcal{T}_{155}$	$\mathcal{T}_{156}$	$\mathcal{T}_{157}$	$\mathcal{T}_{158}$	$\mathcal{T}_{159}$
$\mathcal{T}_{160}$	$\mathcal{T}_{161}$	$\mathcal{T}_{162}$	$\mathcal{T}_{163}$	$\mathcal{T}_{164}$	$\mathcal{T}_{165}$	$\mathcal{T}_{166}$	$\mathcal{T}_{167}$	$\mathcal{T}_{168}$	$\mathcal{T}_{169}$	$\mathcal{T}_{170}$	$\mathcal{T}_{171}$	$\mathcal{T}_{172}$	$\mathcal{T}_{173}$	$\mathcal{T}_{174}$	$\mathcal{T}_{175}$
$\mathcal{T}_{176}$	$\mathcal{T}_{177}$	$\mathcal{T}_{178}$	$\mathcal{T}_{179}$	$\mathcal{T}_{180}$	$\mathcal{T}_{181}$	$\mathcal{T}_{182}$	$\mathcal{T}_{183}$	$\mathcal{T}_{184}$	$\mathcal{T}_{185}$	$\mathcal{T}_{186}$	$\mathcal{T}_{187}$	$\mathcal{T}_{188}$	$\mathcal{T}_{189}$	$\mathcal{T}_{190}$	$\mathcal{T}_{191}$
$\mathcal{T}_{192}$	$\mathcal{T}_{193}$	$\mathcal{T}_{194}$	$\mathcal{T}_{195}$	$\mathcal{T}_{196}$	$\mathcal{T}_{197}$	$\mathcal{T}_{198}$	$\mathcal{T}_{199}$	$\mathcal{T}_{200}$	$\mathcal{T}_{201}$	$\mathcal{T}_{202}$	$\mathcal{T}_{203}$	$\mathcal{T}_{204}$	$\mathcal{T}_{205}$	$\mathcal{T}_{206}$	$\mathcal{T}_{207}$
$\mathcal{T}_{208}$	$\mathcal{T}_{209}$	$\mathcal{T}_{210}$	$\mathcal{T}_{211}$	$\mathcal{T}_{212}$	$\mathcal{T}_{213}$	$\mathcal{T}_{214}$	$\mathcal{T}_{215}$	$\mathcal{T}_{216}$	$\mathcal{T}_{217}$	$\mathcal{T}_{218}$	$\mathcal{T}_{219}$	$\mathcal{T}_{220}$	$\mathcal{T}_{221}$	$\mathcal{T}_{222}$	$\mathcal{T}_{223}$
$\mathcal{T}_{224}$	$\mathcal{T}_{225}$	$\mathcal{T}_{226}$	$\mathcal{T}_{227}$	$\mathcal{T}_{228}$	$\mathcal{T}_{229}$	$\mathcal{T}_{230}$	$\mathcal{T}_{231}$	$\mathcal{T}_{232}$	$\mathcal{T}_{233}$	$\mathcal{T}_{234}$	$\mathcal{T}_{235}$	$\mathcal{T}_{236}$	$\mathcal{T}_{237}$	$\mathcal{T}_{238}$	$\mathcal{T}_{239}$
$\mathcal{T}_{240}$	$\mathcal{T}_{241}$	$\mathcal{T}_{242}$	$\mathcal{T}_{243}$	$\mathcal{T}_{244}$	$\mathcal{T}_{245}$	$\mathcal{T}_{246}$	$\mathcal{T}_{247}$	$\mathcal{T}_{248}$	$\mathcal{T}_{249}$	$\mathcal{T}_{250}$	$\mathcal{T}_{251}$	$\mathcal{T}_{252}$	$\mathcal{T}_{253}$	$\mathcal{T}_{254}$	$\mathcal{T}_{255}$

Profiling phase

First, we find a point of interest :

- we compute the average signal for each set,
- we compute pairwise differences betweeen average signals,
- we keep the point where this is maximum.

Then, for each set, at this point of interest, we compute :

- The **average signal** μ_i (we have it already),
- The **noise variance** σ_i^2.

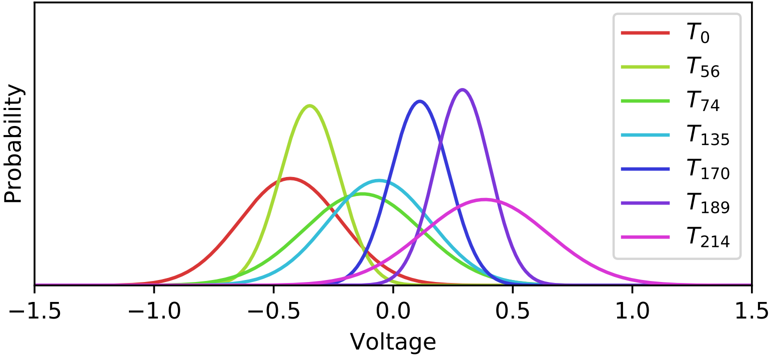
Profiling phase

First, we find a point of interest :

- we compute the average signal for each set,
- we compute pairwise differences betweeen average signals,
- we keep the point where this is maximum.

Then, for each set, at this point of interest, we compute :

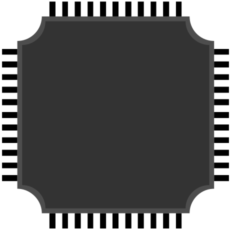
- The **average signal** μ_i (we have it already),
- The **noise variance** σ_i^2.



Matching phase

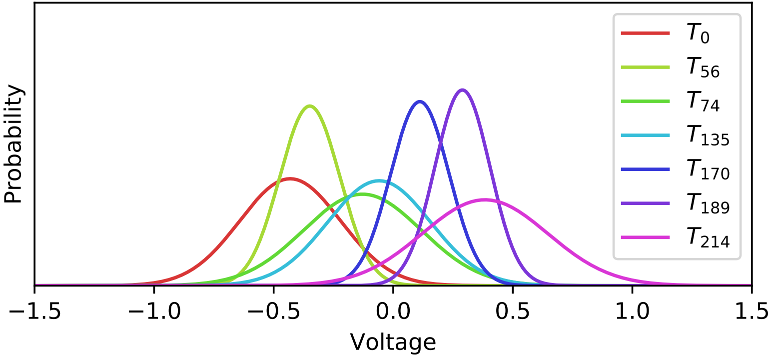
Attack on a closed device:

- we know the **plaintext** input *P* but not the key,
- we look for the intermediate value of interest: $\mathrm{SBox}[p_i \oplus k_i]$,
- we can perform side-channel measurements on it.



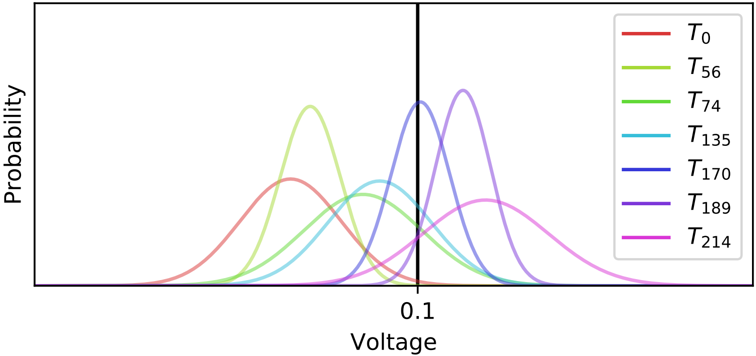
Matching phase

Let us assume we measure a voltage of 0.1.
We now “match” this on our templates.



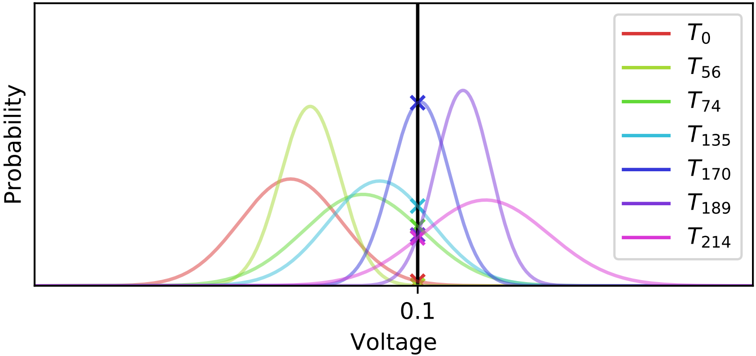
Matching phase

Let us assume we measure a voltage of 0.1.
We now “match” this on our templates.



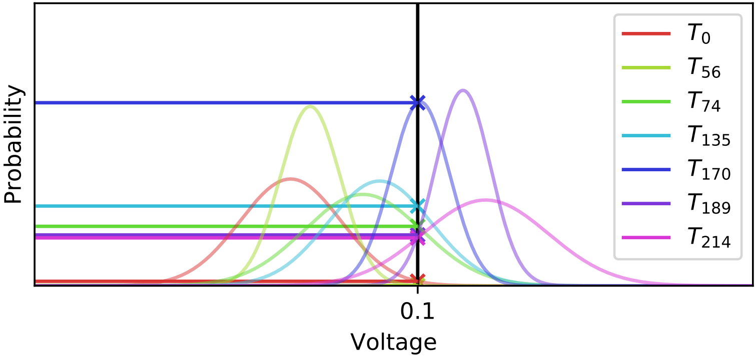
Matching phase

Let us assume we measure a voltage of 0.1.
We now “match” this on our templates.



Matching phase

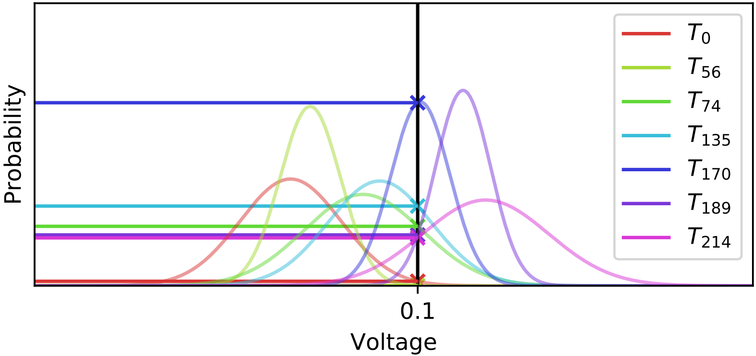
Let us assume we measure a voltage of 0.1.
We now “match” this on our templates.



We can now sort the target byte values by probability.
Values can then be enumerated until we find **the correct key**.

Matching phase

Let us assume we measure a voltage of 0.1.
We now “match” this on our templates.



We can now sort the target byte values by probability.
Values can then be enumerated until we find **the correct key**.

✓

Improvements and options

Combining information

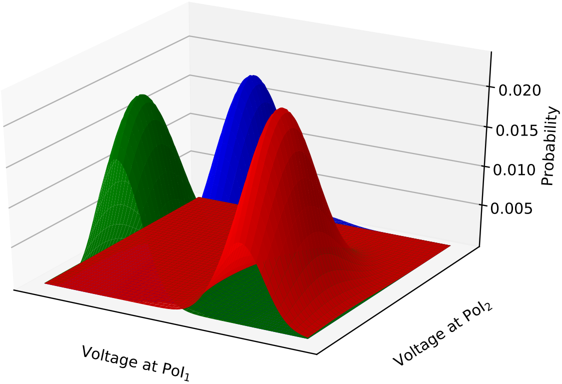
One measurement is (usually) not enough for the matching phase.
We combine information obtained from multiple measurements.

Intermediate values	Measurements				Overall Probability
	1	2	…	N	
0	0.12	**0.15**		0.13	$\prod_{i=0}^{N} p_i$
1	0.01	0.02		0.01	
2	**0.13**	0.14		**0.16**	
3	0.02	0.03		0.04	
…	…	…		…	
255	0.04	0.05		0.03	

We can stop when the confidence is large enough.

More dimensions

With only **one** point of interest, we may miss valuable information.
We can take into account more points of interest.
Templates are then multivariate Gaussian distributions.

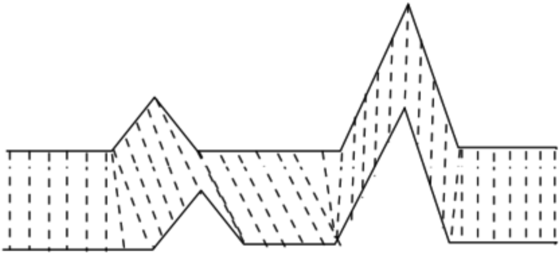


These are specified by a mean vector and a covariance matrix.

Preprocessing

For the template attack to work, samples must be perfectly aligned. Pre-processing them might be necessary:

- Variable shift based on correlation value (linear),
- Dynamic time warping (non-linear).

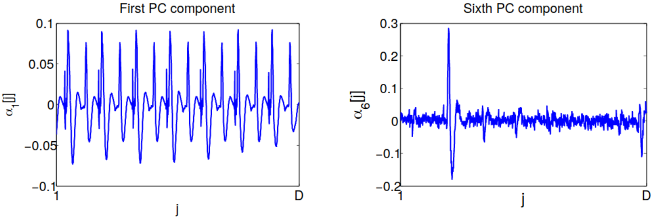


🅭 https://en.wikipedia.org/wiki/Dynamic_time_warping

Selecting points of interest

Selecting points of interest is **not easy**…
Information can spread over multiple samples.
Principal Component Analysis can help reduce the data dimension.

Get principal components of the signal, but which one to keep? [4]



[5]

Still an open question, relies on attacker's knowledge.

---

[4] L. Batina, J. Hogenboom, and J. G. J. van Woudenberg. "Getting More from PCA: First Results of Using Principal Component Analysis for Extensive Power Analysis". *CT-RSA*. 2012, pp. 383–397.

[5] E. Cagli, C. Dumas, and E. Prouff. "Enhancing Dimensionality Reduction Methods for Side-Channel Attacks". *CARDIS*. 2015, pp. 15–33.

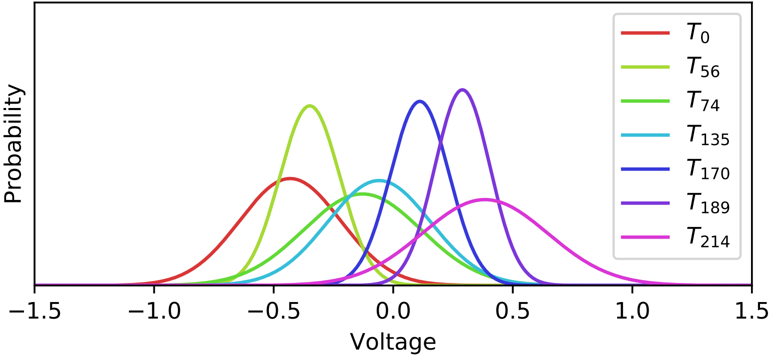
Computational aspects

As highlighted in [6], computational problems may arise in practice:

- The covariance matrix might not be invertible,
- Multiplying the probabilites can lead to floating-point errors.

They propose the following solutions:

- Use the logarithm of the multivariate normal distribution,
- Use a pooled covariance matrix,



---

[6] O. Choudary and M. G. Kuhn. “Efficient Template Attacks”. *CARDIS*. 2013, pp. 253–270.

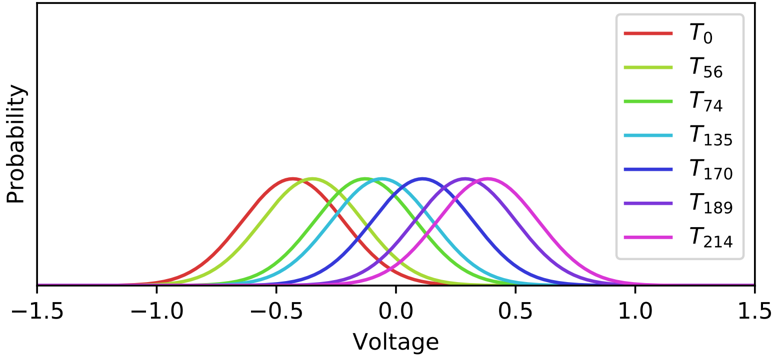
Computational aspects

As highlighted in [6], computational problems may arise in practice:

- The covariance matrix might not be invertible,
- Multiplying the probabilites can lead to floating-point errors.

They propose the following solutions:

- Use the logarithm of the multivariate normal distribution,
- Use a pooled covariance matrix,



---

[6] O. Choudary and M. G. Kuhn. "Efficient Template Attacks". *CARDIS*. 2013, pp. 253–270.

Known plaintext VS known ciphertext

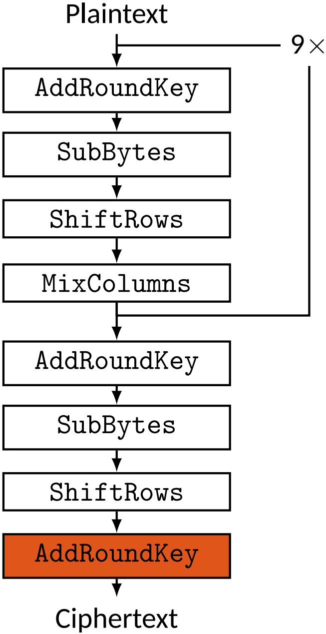
The presented attack requires to know the plaintext.

Same principles apply if we know the ciphertext instead.

This time we attack the last round.

We recover $C \oplus K_{10}$ and we know C.

From the **round-key** K_{10} we recover the key K by reverting the key-schedule.



Experimental aspects

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. “An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks”. *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

A: As many as possible! Typically hundreds of thousands.

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. "An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks". *SAC.* vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. “An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks”. *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?

A: Target dependent. Less than 5 for unprotected ones.

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. “An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks”. *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?
A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?
A: Target dependent. Less than 5 for unprotected ones.

Q: Can we profile **one** device to attack another?

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. "An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks". *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?
A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?
A: Target dependent. Less than 5 for unprotected ones.

Q: Can we profile **one** device to attack another?
A: Theoretically yes (very powerful).

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. "An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks". *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?

A: Target dependent. Less than 5 for unprotected ones.

Q: Can we profile **one** device to attack another?

A: Theoretically yes (very powerful).

Q: How long does the attack take?

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. “An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks”. *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?

A: Target dependent. Less than 5 for unprotected ones.

Q: Can we profile **one** device to attack another?

A: Theoretically yes (very powerful).

Q: How long does the attack take?

A: Typically a few seconds, measurements take time.

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. “An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks”. *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q:	How many traces are needed for the **profiling** phase?
A:	As many as possible! Typically hundreds of thousands.
Q:	How many traces are needed for the **matching** phase?
A:	Target dependent. Less than 5 for unprotected ones.
Q:	Can we profile **one** device to attack another?
A:	Theoretically yes (very powerful).
Q:	How long does the attack take?
A:	Typically a few seconds, measurements take time.
Q:	What if the **correct** key does not rank first?

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. "An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks". *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?

A: Target dependent. Less than 5 for unprotected ones.

Q: Can we profile **one** device to attack another?

A: Theoretically yes (very powerful).

Q: How long does the attack take?

A: Typically a few seconds, measurements take time.

Q: What if the **correct** key does not rank first?

A: Key enumeration methods [7] exploit the probabilities.

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. "An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks". *SAC*. vol. 7707. 2012, pp. 390–406.

Experimental aspects

Q: How many traces are needed for the **profiling** phase?

A: As many as possible! Typically hundreds of thousands.

Q: How many traces are needed for the **matching** phase?

A: Target dependent. Less than 5 for unprotected ones.

Q: Can we profile **one** device to attack another?

A: Theoretically yes (very powerful).

Q: How long does the attack take?

A: Typically a few seconds, measurements take time.

Q: What if the **correct** key does not rank first?

A: Key enumeration methods [7] exploit the probabilities.

Q: Other questions?

---

[7] N. Veyrat-Charvillon, B. Gérard, M. Renauld, and F. Standaert. “An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks”. *SAC*. vol. 7707. 2012, pp. 390–406.

Conclusion

Conclusion

Template attacks are a very powerful tool.

Even **protected** implementations can be targeted.

They can be used to attack other algorithms (asymmetric, etc.)

Conclusion

Template attacks are a very powerful tool.

Even **protected** implementations can be targeted.

They can be used to attack other algorithms (asymmetric, etc.)

Slightly less fashionable now, because of...

Conclusion

Template attacks are a very powerful tool.

Even **protected** implementations can be targeted.

They can be used to attack other algorithms (asymmetric, etc.)

Slightly less fashionable now, because of... deep learning.

Conclusion

Template attacks are a very powerful tool.

Even **protected** implementations can be targeted.

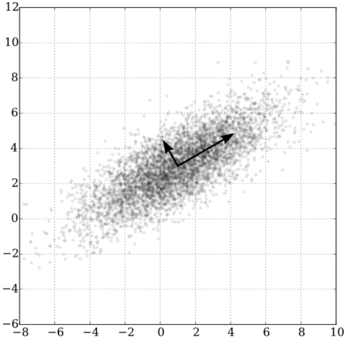
They can be used to attack other algorithms (asymmetric, etc.)

Slightly less fashionable now, because of... deep learning.

— Questions? —

Backup slides

PCA



Identify the components where data varies the most.
Orthogonal vectors.