Lecture 3: Wrapping up and Q&A

Q&A

- Can you explain the difference between Hamming weight and Hamming distance models?

$r0_{i-1}$	$r0_i$
0	0
0	1
0	1
0	0
0	1
0	1
0	0
0	0

$r0_{i-1}$	$r0_i$
1	1
0	0
1	1
1	1
0	0
1	1
0	0
1	1

Q&A

- Can you explain the difference between Hamming weight and Hamming distance models?
- What shall we do if we don't know previous register value 'X'?

$r0_{i-1}$	$r0_i$	$r0_{i-1}$	$r0_i$	$r0_{i-1}$	$r0_i$
0	0	1	1	X	1
0	1	0	0	X	0
0	1	1	1	X	1
0	0	1	1	X	1
0	1	0	0	X	0
0	1	1	1	X	1
0	0	0	0	X	0
0	0	1	1	X	1

Q&A

- Taken into account that $i(0 \rightarrow 1) = \alpha$, $i(1 \rightarrow 0) = \beta$ and $\alpha \neq \beta$ can we construct better models that Hamming Distance?

$r0_{i-1}$	$r0_i$	i
1	0	β
0	1	α
1	1	0
1	0	β
0	1	α
1	0	β
0	1	α
1	1	0

Q&A

- Consider a big register of 128 bits (16 bytes) commonly used in AES-128 hardware implementation.
- Will side-channel on one byte will work?
- Does this change anything to you?
- Can you write the model for this example?
- The image below is for bytes (not bits as before).

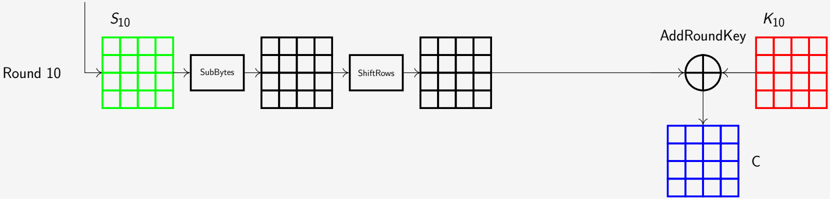
r_0	r_1	
0x00	$Sbox[p_{i,0} \oplus k_{0,0}]$	Attack this byte only
0x00	$Sbox[p_{i,1} \oplus k_{0,1}]$	X
0x00	$Sbox[p_{i,2} \oplus k_{0,2}]$	X
0x00	$Sbox[p_{i,3} \oplus k_{0,3}]$	X
...	...	X
0x00	$Sbox[p_{i,13} \oplus k_{0,13}]$	X
0x00	$Sbox[p_{i,14} \oplus k_{0,14}]$	X
0x00	$Sbox[p_{i,15} \oplus k_{0,15}]$	X

Q&A

- Side-channels can work with 1 bit models: Differential Power Analysis
  - Probably faster than PCC
  - Does not rely on any special dependency between Hamming weight and real power consumption.
  - Relies only on the fact that $\alpha \neq \beta$,
- DPA works for White-Box Crypto (sometimes) while other attacks don't

r_0	r_1	
0x00	$Sbox[p_{i,0} \oplus k_{0,0}]$	$Sbox[p_{i,0} \oplus k_j]\&0x01$
0x00	$Sbox[p_{i,1} \oplus k_{0,1}]$	X
0x00	$Sbox[p_{i,2} \oplus k_{0,2}]$	X
0x00	$Sbox[p_{i,3} \oplus k_{0,3}]$	X
...	...	X
0x00	$Sbox[p_{i,13} \oplus k_{0,13}]$	X
0x00	$Sbox[p_{i,14} \oplus k_{0,14}]$	X
0x00	$Sbox[p_{i,15} \oplus k_{0,15}]$	X

Differential Power Analysis



C^1	18	···	72
C^2	2f	···	89
C^i	···	···	···
C^n	5d	···	b7

	Random data				Real leakage			Random data	
L_t^1	7.8	4.4	8	4.6	5.5	···	8.4	7	3.6
L_t^2	6.1	2.7	6.3	2.9	3.8	···	7.7	5.3	1.9
L_t^i	···	···	···	···	···	···	···	···	···
L_t^n	2.7	6.3	2.9	8.5	4.4	···	6.3	1.9	5.5

Create power consumption model for one Sbox operation

$K_0 = 0$	$K_j = j$	$K_{FF} = 255$
$\mathrm{Sb}^{-1}[K_0 \oplus 18]$ & 0x1 $= 0$	···	$\mathrm{Sb}^{-1}[K_{FF} \oplus 18]$ & 0x1 $= 1$
$\mathrm{Sb}^{-1}[K_0 \oplus 2f]$ & 0x1 $= 1$	···	$\mathrm{Sb}^{-1}[K_{FF} \oplus 2f]$ & 0x1 $= 0$
···	···	···
$\mathrm{Sb}^{-1}[K_0 \oplus 5d]$ & 0x1 $= 1$	···	$\mathrm{Sb}^{-1}[K_{FF} \oplus 5d]$ & 0x1 $= 0$

Differential Power Analysis

C^1	18	···	72
C^2	2f	···	89
C^i	···	···	···
C^n	5d	···	b7

	Random data				Real leakage			Random data	
L_t^1	7.8	4.4	8	4.6	5.5	···	8.4	7	3.6
L_t^2	6.1	2.7	6.3	2.9	3.8	···	7.7	5.3	1.9
L_t^i	···	···	···	···	···	···	···	···	···
L_t^n	2.7	6.3	2.9	8.5	4.4	···	6.3	1.9	5.5

Create power consumption model for one Sbox operation

$K_0 = 0$	$K_j = j$	$K_{FF} = 255$
$\mathrm{Sb}^{-1}[K_0 \oplus 18]$ & 0x1 $= 0$	···	$\mathrm{Sb}^{-1}[K_{FF} \oplus 18]$ & 0x1 $= 1$
$\mathrm{Sb}^{-1}[K_0 \oplus 2f]$ & 0x1 $= 1$	···	$\mathrm{Sb}^{-1}[K_{FF} \oplus 2f]$ & 0x1 $= 0$
···	···	···
$\mathrm{Sb}^{-1}[K_0 \oplus 5d]$ & 0x1 $= 1$	···	$\mathrm{Sb}^{-1}[K_{FF} \oplus 5d]$ & 0x1 $= 0$

Group 0

7.8	4.4	8	4.6	5.5	···	8.4	7	3.6
6.2	3.8	1.4	8	4.8	···	7.7	0.4	7
···	···	···	···	···	···	···	···	···
9.4	7	4.6	2.2	3.4	···	6.3	3.6	1.2

Group 1

6.1	2.7	6.3	2.9	5.5	···	8.4	5.3	1.9
6.6	3.7	0.8	6.9	3.8	···	7.7	4.3	1.4
···	···	···	···	···	···	···	···	···
2.7	6.3	2.9	8.5	4.4	···	6.3	1.9	5.5

$abs(\mu(Group_1) - \mu(Group_0))$

Q&A

- Side-channels can work with 1 bit models: Differential Power Analysis
- Fix the bit you want to attack: assume 0x01
- Take a key candidate k_i
- Compute a target operation $Sbox^{-1}[k_i \oplus p_{:,0}]$
- Take one bit from this operation $Sbox^{-1}[k_i \oplus p_{:,0}]\&0x01$
- Split traces in two groups:
  - Group 0: traces for which your model bit is 0
  - Group 1: traces for which your model bit is 1
- Compute difference of means $abs(\mu(Group_1) - \mu(Group_0))$ (mean over columns)
- For a concrete key k_i take the maximum value from this group $max(abs(\mu(Group_1) - \mu(Group_0)))$
- Repeat the process for all k_i and then select the key with the maximum difference of means

Q&A

- Instead of difference of means you can use Welch's T-test

$$T = \frac{\text{abs}(\mu(\text{Group}_1) - \mu(\text{Group}_0))}{\sqrt{\frac{\sigma(\text{Group}_1)}{n_1} - \frac{\sigma(\text{Group}_0)}{n_0}}}$$

- $\sigma(\text{Group}_1)$ is a standard deviation (again for each column)
- n_i is the number of column elements (number of traces in each group)
- T-test can be used instead of simple difference of means in DPA
- Also T-test can be used to characterise if two sets of data have different power distribution (and this is useful)

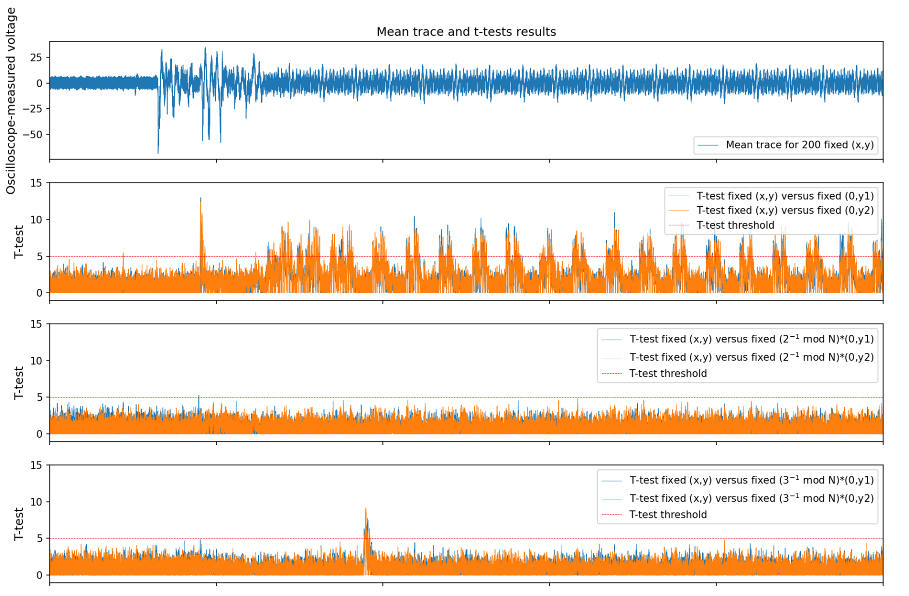
Q&A

- Assume, you have a device where you control everything for AES: a key, a plaintext, and a ciphertext
- Your task is to tell if this device leaks power or not (and also where it leaks power in time)
- What would you do?

Q&A

- T-test can be used for leakage recognition
- One of the main approaches is called: random versus fixed
- You acquire N traces with all fixed values (key, plaintext)
- You acquire N traces where you modify the plaintext
- You compute a Ttest between two groups of traces.
- Ttest > 5 will illustrate you where the two groups have different power distribution
- Since the only difference is plaintext - differences in power distribution will highilight all places where plaintext has impact (plaintext transfer and AES)

Example of Ttest usage for ECDH



Q&A

- T-test is also used to assess EM leakage
- EM leakage is stronger over a block we attack (e.g. AES hardware engine)
- Since we don't know where AES is located inside the chip we need to try various positions of the probe
- Instead of doing an entire attack we can use Ttest to approximate location of the "nice" leakage

Q&A

- Leakage assessment is very common task in side-channel attacks
- You have a device running an algorithm (ECDH, homomorphic encryption, etc)
- You control everything there
- How to check if there is any leakage associated with the algorithm processing?
- Lets discuss a bit this point.

Q&A

- Similar task is an attack against a new algorithm
- Post-quantun algorithms require "new" attacks
- How would you proceed to get a new attack?
- Getting Hamming weight/distance of internal operations?