Lab 1 - AES CPA

April 7th, 2024

Rayane ANNEN **Hugo DUCOMMUN** **Alexis MARTINS**

Introduction

Dans ce premier laboratoire, nous avions à disposition une carte physique ChipWhisperer qui réalisait des chiffrements AES-128 et qui n'avait aucune protection contre les attaques par canaux auxiliaires. Le but va donc être d'analyser les traces de la puissance consommée durant le chiffrement et d'effectuer une CPA (Correlation Power Analysis) afin de retrouver la clé utilisée pour le chiffrement.

Setup

Hardware:

- DTU: STM32F303 avec AES-128 non protégé contre les side-channels.
- Shunt resistor = 12 Ω

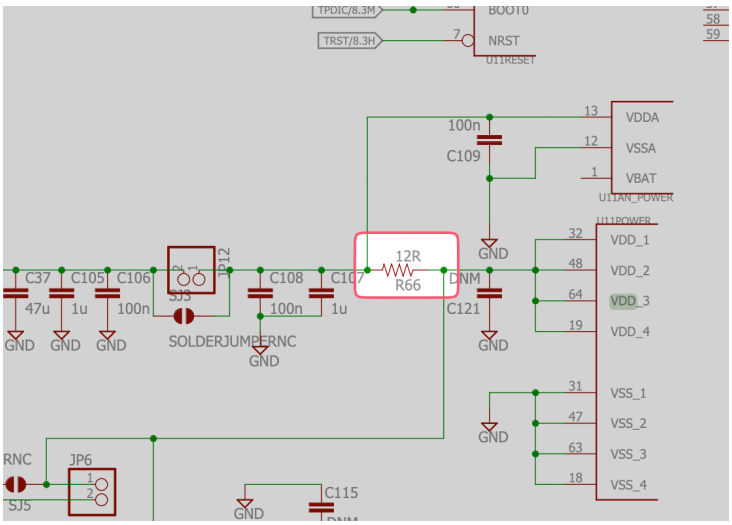

Figure 1: Shunt Resistor (R66) 12 Ω

Nous avons enregistré 1000 traces de 8000 points avec le ChipWhisperer. La taille des textes clairs est de 16 bytes.

Méthodologie d'attaque

Dans un premier temps, nous avons décidé de dessiner les traces que nous avions récoltées sur le ChipWhisperer. Le but était d'identifier l'intervalle dans lequel se situait chaque round.

Pour cela, nous avons tracer le coefficient de corrélation (de Pearson) entre les hamming weights du plaintext (et du ciphertext) avec les traces récoltées :

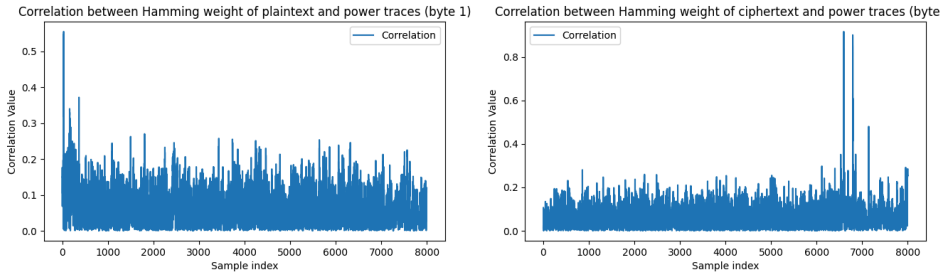

Figure 2: Identification de l'intervalle des rondes AES

Nous avons décidés d'attaquer le dernier round, étant donné que c'est ce que nous faisions jusqu'à là dans les autres laboratoires. Nous avons aussi compris, après coup, que cela aurait été possible sur le premier round (en plus la clé obtenue est directement la master key).

Dans les 8000 traces, nous avons donc décidé d'isoler l'intervalle des traces de 6300 à 6900 pour représenter le dernier round AES selon le graphe précédent.

Ensuite, l'attaque est assez similaire aux précédents laboratoires. Nous avons fait une matrice de tous nos textes chiffrés, que nous avons ensuite parcouru byte à byte. En faisant cela, on a un vecteur de tous les bytes x (x compris entre 0 et 15) pour nos 1000 textes chiffrés.

Avec ces vecteurs, il faudra parcourir toutes les clés candidats pour le byte x sélectionné et revenir en arrière dans le calcul du dernier round d'AES (AddRound, puis inverser le SubByte) et calculer les Hamming Weights.

```
# Final round key
subkey = np.zeros(16)
# Liste des candidats de clé (0 à 255)
key_candidates = np.arange(256).astype(np.uint8)

# Initialiser une figure pour les graphiques 4x4
plt.figure(figsize=(15, 10))

# Loop pour chaque byte des ciphertexts
for i in range(16):
    # Contient des listes des coefficients de correlation de chaque trace (index = key
guess)
    cpa = [0] * 256
    # Contient les meilleurs coefficients de correlations pour chaque key guess (index
= key guess)
    max_cpa = [0] * 256
    # i-ème byte de tous les ciphertexts (colonne)
    ct_col = ctexts[:,i]

    for key in key_candidates:
        sub_bytes = invSbox[np.bitwise_xor(ct_col, key)]
        model = HW_uint8[sub_bytes]
        mean_model = np.mean(model) #hyp
        # Liste des coefficients de correlation de chaque trace
        pcc = []
        for j in range(power_traces.shape[1]):
                                    pcc.append(np.corrcoef(np.squeeze(np.asarray(model)),
np.squeeze(np.asarray(power_traces[:, j])))[0,1])
        cpa[key] = np.array(pcc)
        max_cpa[key] = max(abs(cpa[key]))
```

Après cela, il suffit de sortir les 3 hypothèses de clé qui ont le meilleur coefficient (en valeur absolue) de corrélation :

```
# Trouver et stocker les top 3 de corrélation pour ce byte
indices_of_three_highest = sorted(range(len(max_cpa)), key=lambda i: max_cpa[i],
reverse=True)[:3]
subkey[i] = indices_of_three_highest[0]
```

En résumé, nous faisons :

- Calcule des corr_coeff pour chaque trace (stocké dans `cpa`, une liste par hypothèse de clé / index)
- Stockage du maximum de `cpa` dans `max_cpa` (1 par hypothèse de clé / index)
- Stockage du key_guess correspondant au meilleur corr_coeff stocké dans `max_cpa`

Clé trouvée

`SCA{RealAES-128}`

3 meilleures clés par byte

Index = Key guess

```
Byte 0:
  Top 1: Index 64 with CPA 0.5770241595444894
  Top 2: Index 60 with CPA 0.36504412871383146
  Top 3: Index 46 with CPA 0.34980048486233656
Byte 1:
  Top 1: Index 103 with CPA 0.42414054655746913
  Top 2: Index 180 with CPA 0.38125919116884205
  Top 3: Index 208 with CPA 0.37569937764209904
Byte 2:
  Top 1: Index 71 with CPA 0.48649837204348
  Top 2: Index 224 with CPA 0.352796706075585
  Top 3: Index 24 with CPA 0.3390238134327905
Byte 3:
  Top 1: Index 136 with CPA 0.9265080188646125
  Top 2: Index 119 with CPA 0.39264439202294416
  Top 3: Index 127 with CPA 0.3721606633443588
Byte 4:
  Top 1: Index 35 with CPA 0.5771433996781732
  Top 2: Index 190 with CPA 0.37371640226604735
  Top 3: Index 96 with CPA 0.37170235388736766
Byte 5:
  Top 1: Index 202 with CPA 0.6200235438921873
  Top 2: Index 115 with CPA 0.46413749846877045
  Top 3: Index 71 with CPA 0.4124273536944024
Byte 6:
  Top 1: Index 186 with CPA 0.9306194921783321
  Top 2: Index 107 with CPA 0.41216319779283767
  Top 3: Index 191 with CPA 0.35671320365284975
Byte 7:
  Top 1: Index 54 with CPA 0.7024611955673524
  Top 2: Index 159 with CPA 0.3894443045139234
  Top 3: Index 68 with CPA 0.372291515264502
Byte 8:
  Top 1: Index 149 with CPA 0.6594642569244467
  Top 2: Index 215 with CPA 0.3577077436581626
  Top 3: Index 226 with CPA 0.3416083453539352
```

```
Byte 9:
  Top 1: Index 215 with CPA 0.50727175354863
  Top 2: Index 84 with CPA 0.37858200025005595
  Top 3: Index 251 with CPA 0.3634924552614519
Byte 10:
  Top 1: Index 201 with CPA 0.5373716659198619
  Top 2: Index 88 with CPA 0.4050972477023634
  Top 3: Index 254 with CPA 0.3906522254289343
Byte 11:
  Top 1: Index 235 with CPA 0.4312246940117489
  Top 2: Index 99 with CPA 0.3498942276435232
  Top 3: Index 97 with CPA 0.34945605091407905
Byte 12:
  Top 1: Index 130 with CPA 0.4748008335431175
  Top 2: Index 245 with CPA 0.3576062362871892
  Top 3: Index 207 with CPA 0.34797257072908055
Byte 13:
  Top 1: Index 212 with CPA 0.5191533915992961
  Top 2: Index 187 with CPA 0.38218189869557867
  Top 3: Index 234 with CPA 0.34363813752229433
Byte 14:
  Top 1: Index 153 with CPA 0.41632773590575295
  Top 2: Index 101 with CPA 0.36504327374722856
  Top 3: Index 9 with CPA 0.3459993227928831
Byte 15:
  Top 1: Index 255 with CPA 0.606115674509139
  Top 2: Index 153 with CPA 0.3449372850543634
  Top 3: Index 201 with CPA 0.3444572465715154
```

Graphes de correlation

Pour voir la totalité des graphes, voir le jupyter notebook ci-joint.

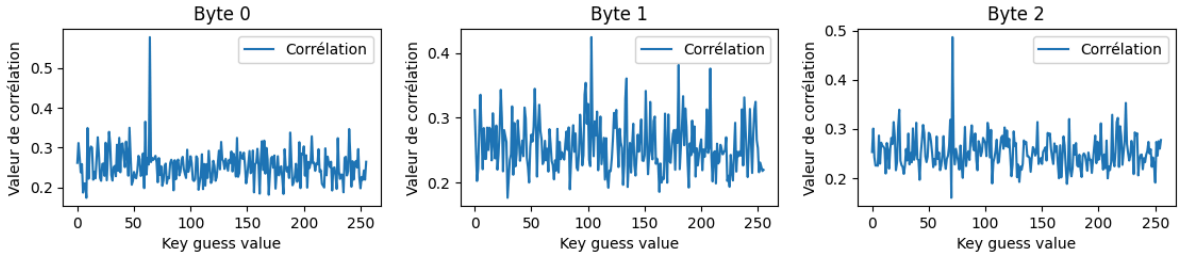

Figure 3: Graphes de correlation pour les 3 premiers bytes de la clé

Partition de la trace attaquée

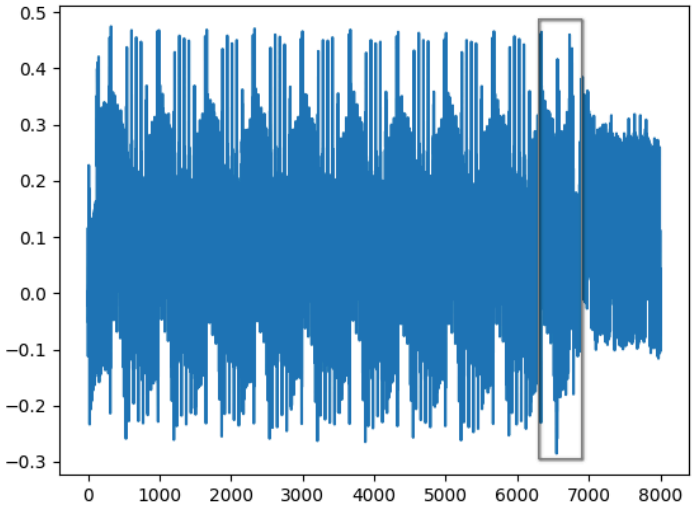

Figure 4: Partition de la trace attaquée

Index de début : 6300

Index de fin : 6900

Bonus : Nombre de traces minimums

Nous avons réussi avec notre méthodologie à descendre à **150 traces** pour retrouver la clé.

En dessous, les coefficients de corrélation sont trop proches entre les key_guess et la clé obtenue est erronée.