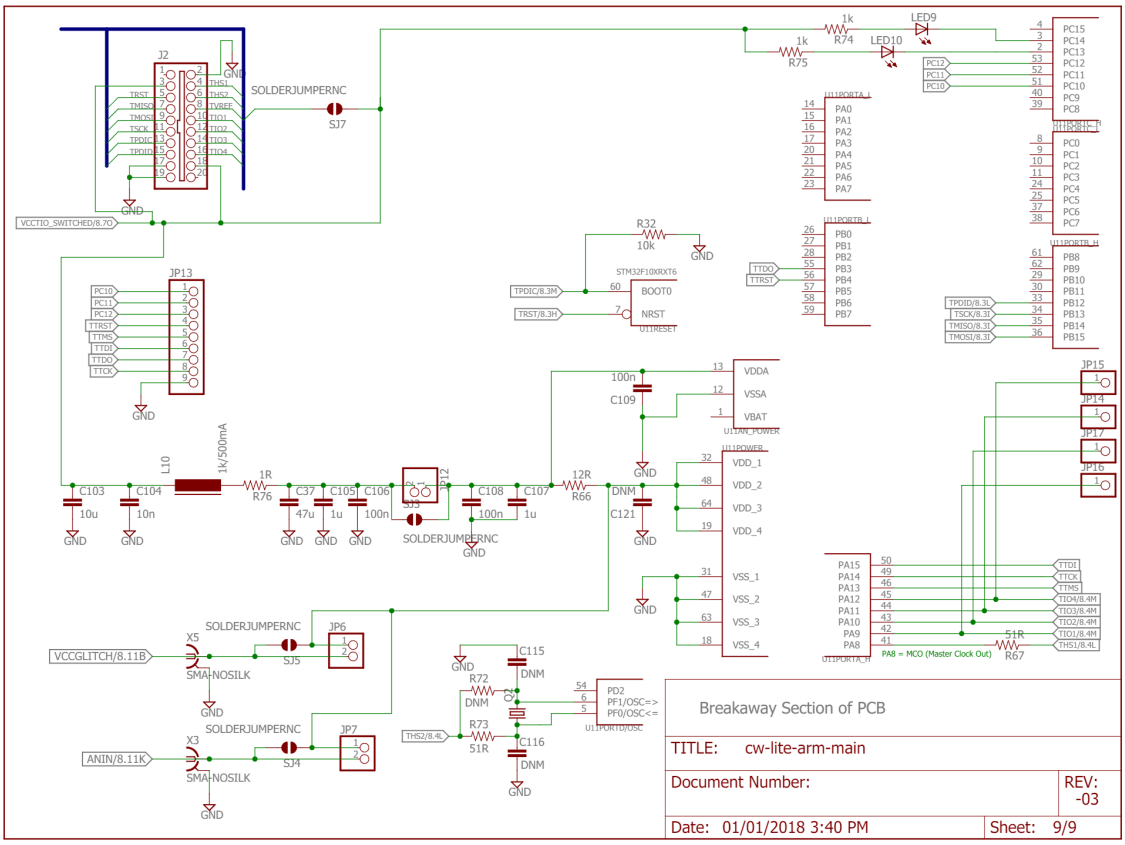
I. Configuration de la mesure

Dans le processus de mesure, tous est déjà mis en place dans la Chipwhisperer, donc il nous suffit de brancher le câble USB et de lancer le code qui contient les instructions pour la capture et le chiffrement des textes.

En examinant de plus près le processus de mesure dans la Chipwhisperer, nous pouvons le diviser en deux parties distinctes.

La première partie concerne la *capture*, qui va permettre de collecter les textes chiffrés et les traces, puis de les transmettre via USB. C'est aussi elle qui va contenir les éléments de mesure tels qu'un "mini oscilloscope" pour mesurer les différences de tension.

La seconde partie est la *target*, sur laquelle l'attaque est dirigée. Cette partie permet le chiffrement AES et contient également la résistance de shunt utilisée par la partie de *capture* pour mesurer les variations de tension. Dans le schéma électrique de la partie *target* qui se trouve ci-dessous, la résistance de shunt utilisée est la R66 et elle a une valeur de 12 Ohms. Sa position près de la fin du circuit, avant la connexion à la masse (GND), est stratégique. Cette disposition garantit que les mesures de tension soit autour de 0 volts, facilitant ainsi leur interprétation. Placer la résustance au début du circuit, près de l'alimentation, aurait entraîné un décalage de 5 volts dans les mesusres, rendant les graphiques moins lisibles.



II. Nombre de traces acquises

Nous avons commencé en chiffrant 1000 textes et en collectant les 1000 traces associées. Pour les traces, on gardait uniquement les samples 6200 à 6900, ce qui correspond au dernier round AES. Par la suite, nous avons cherché le nombre minimal de traces à acquérir. Pour ce faire, on a procédé par recherche dichotomique et nous sommes arrivés à la conclusion qu'il fallait entre 100 et 200 traces au minimum afin que l'attaque fonctionne correctement.

III. La méthodologie d'attaque

Premièrement, il faut "armer" la ChipWhisperer avec la fonction `scope.arm()`. Ensuite, on génère aléatoirement des textes clairs de 16 octets et les données sont envoyées à la cible avec `target.simpleserial_write("p", ptext)`. La ChipWhisperer chiffre ces N textes clairs puis elle commencera à capturer une trace de la consommation de puissance pendant le chiffrement avec `scope.capture()`. Cette fonction est bloquante jusqu'à ce que la ChipWhisperer ait fini d'enregistrer. On peut lire la capture avec la fonction `scope.get_last_trace()`

À partir du graphe de la dernière trace de consommation de puissance, on constate, comme mentionné plus haut, que le 10ème round se trouve dans les samples 6200 à 6900. Le fait de récupérer le dernier round va nous permettre de trouver la clef en ayant à disposition les textes chiffrés générés précédemment. Ensuite, nous créons 3 matrices; une contenant les textes clairs, une contenant les traces correspondantes et la dernière contenant les textes chiffrés correspondants.

Pour ce qui concerne l'attaque en elle-même, le but est de réaliser le même procédé utilisé durant les exercices afin de récupérer la clef utilisée pour chiffrer les données: On itère sur chacun des 16 octets de la clef. Puis, pour chaque octet de la clef, on boucle sur les 256 possibilités (0 à 255) pour deviner la valeur de cet octet. Ainsi, pour chacune de ces possibilités, on effectue les étapes suivantes:

- L'opération XOR est effectuée entre la valeur devinéee de l'octet de la clé et chaque élément de l'une des 16 colonnes de la matrice représentant les textes chiffrés. Cela produit une liste de résultats.
- On applique la S-Box inverse d'AES sur la liste précédemment obtenue pour obtenir l'état intermédiaire S10.
- Le poids de Hamming (nombre de bits à 1) est calculé pour chaque élément de la liste résultante de l'étape précédente.
- Ensuite, le coefficient de Pearson est calculé entre la liste contenant les poids de Hamming et chacune des colonnes de la matrice représentant les traces de consommation de puissance éléctrique. Cela donne une liste de coefficients de corrélation.
- Finalement, on met à jour la matrice résultante avec les coefficients de corrélation calculés pour chaque valeur de k et chaque colonne de la matrice de trace.

IV. La clef récupérée

Le tableau correspondant à la clef est le suivant `res = [64, 103, 71, 136, 35, 202, 186, 54, 149, 215, 201, 235, 130, 212, 153, 255]`. En le convertissant en caractères ASCII, on obtient le flag suivant: `SCA{RealAES-128}`.

V. Le coefficient de corrélation des trois meilleurs candidats

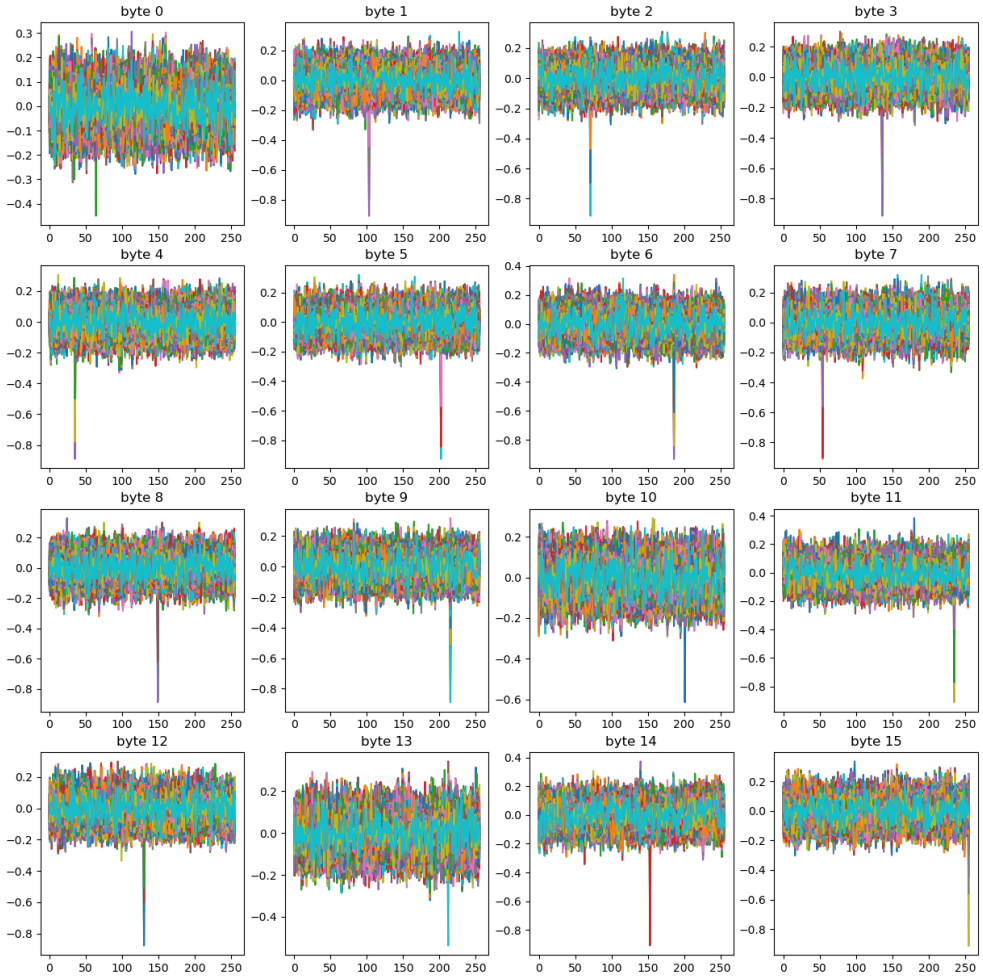
Voici la liste des trois meilleurs coefficients de corrélation obtenus pour chaque octets de la clef. Le format de la liste correspond à `(candidat, n° de la trace) coefficient de corrélation`.

```
byte 0
(64, 142) -0.4815748053213781
(55, 143) -0.3324176671914239
(193, 143) 0.33229962501685856
byte 1
(103, 22) -0.9391358302056646
(5, 199) -0.3435452955269484
(4, 199) -0.33468551859050566
byte 2
(71, 69) -0.9193734062351578
(40, 67) -0.36161848748144254
(114, 404) -0.3583719630373861
byte 3
(136, 112) -0.9165211792929316
(109, 412) -0.31121323045225907
(89, 113) -0.3098406998229267
byte 4
(35, 14) -0.9130850342627416
(132, 13) -0.35804881160779184
(112, 610) -0.3135297863119017
byte 5
(202, 59) -0.9235002841664007
(103, 433) -0.3734675949365823
(185, 434) 0.3271041888007638
byte 6
(186, 104) -0.9248242853937181
(28, 258) -0.3292720083413411
(91, 103) -0.3259896768714068
byte 7
(54, 1) -0.9234820671658761
(205, 8) 0.32930321267847207
(29, 450) 0.32478208202857106
byte 8
(149, 54) -0.8563050391015067
(11, 164) 0.3634633290764056
(135, 163) 0.34624543536437447
byte 9
(215, 99) -0.9161591336967374
(64, 222) 0.3144662821891159
(113, 349) 0.3039923776603872
byte 10
(201, 230) -0.5523636819536228
(86, 340) 0.33640098533387663
```

```
(220, 480) 0.3157572781578233
byte 11
(235, 38) -0.9060702134937032
(26, 37) 0.3981167531681077
(10, 289) -0.3330188441281084
byte 12
(130, 88) -0.8827714631825138
(156, 497) 0.3340170865340675
(158, 497) 0.3238702093125326
byte 13
(212, 189) -0.6205947814216272
(117, 666) -0.32524178308917334
(7, 496) 0.3042728234702088
byte 14
(153, 33) -0.9028901312691092
(41, 225) -0.346904008247814
(205, 256) 0.3295028888400653
byte 15
(255, 78) -0.8920626425624191
(46, 74) -0.327001381664428
(99, 681) -0.3217731923798533
```

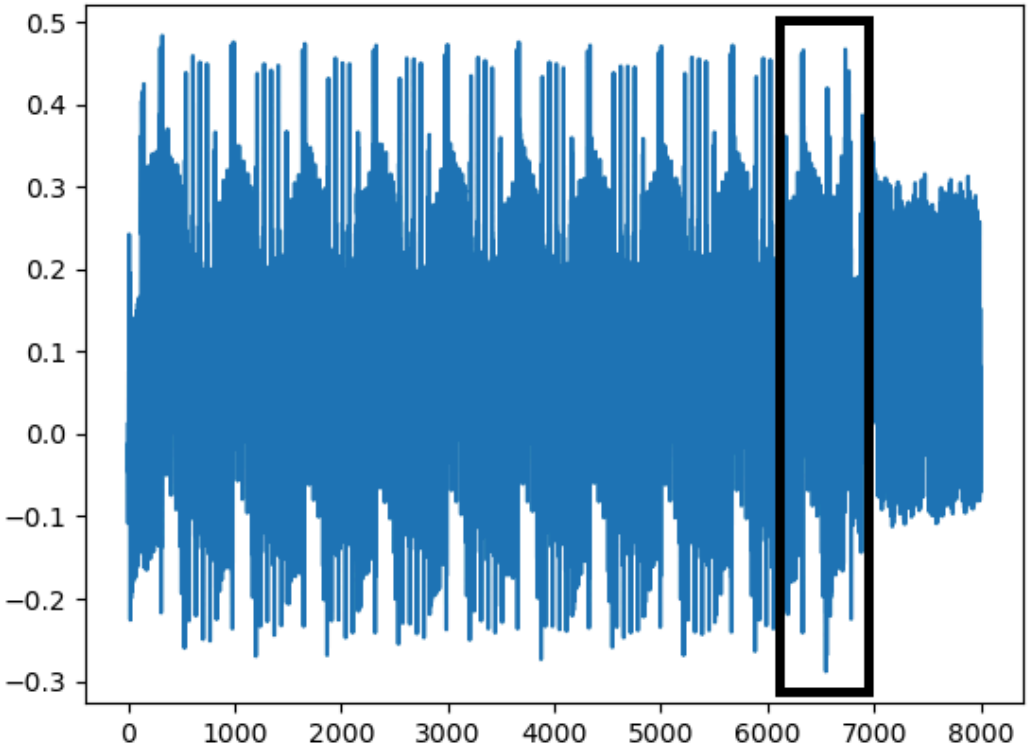
VI. Les graphes de corrélation

Voici les graphes de corrélation pour chaque octet de la clef. Comme on peut le constater sur chaque graphe, il y a une valeur qui a un coefficient de Pearson plus élevé (valeur absolue) que les autres et ces valeurs correspondent au tableau obtenu précédemment.



VII. Trace avec un point culminant de la région attaquée

Voici une des traces utilisées avec un rectangle noir sur la partie que nous avons attaquée lors de ce laboratoire. Cela correspond aux samples entre les valeurs 6200 et 6900, qui correspondent au dernier round AES et qui est le plus facile à attaquer vu que c'est le seul qui interagisse directement avec nos textes chiffrés.


0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
1000
2000
3000
4000
5000
6000
7000
8000