Attaques par canaux auxiliaires quantiques

Imran Meghazi, Eric Bourreau, Florent Brugnier

LIRMM, Univ. Montpellier, CNRS, Montpellier, France
prenom.nom@lirmm.fr

Mots-clés *: Cryptanalyse, Optimisation Quantique.*

1 Contexte

Les ordinateurs quantiques ont été imaginés à la fin du XXe siècle. Ce nouveau paradigme bouleverse l'informatique théorique depuis 20 ans avec notamment les algorithmes de Shor ou de Grover. Depuis quelques années, ces ordinateurs sont une réalité. Les implémentations physiques permettent maintenant d'expérimenter les différents algorithmes jusque-là irréalisables. Avec quelques blocs de code quantique (définis à partir d'un jeu d'instruction restreint de portes quantiques), il est possible de construire des opérateurs, les composer dans des programmes et d'exécuter ceux-ci sur des QPU (machines à processeur quantique).

Nous allons nous intéresser à un problème de cryptographie (recherche de clé cachée), et plus particulièrement de cryptanalyse (recherche par analyse de messages). À l'heure actuelle, les algorithmes de chiffrement utilisés ont fait leurs preuves mathématiquement. Toutefois, si les failles au niveau de l'implémentation logicielle sont peu nombreuses, il est possible d'en trouver au niveau de l'implémentation physique. En effet, le chiffrement est souvent délégué à une puce spécialisée. Ce microprocesseur, étant le seul à connaître la clé de chiffrement, aura pour seule tâche de chiffrer les données. Lors de ces opérations, le microprocesseur en question va consommer une quantité précise de courant en fonction des instructions et donc des transistors utilisés. Il existe plusieurs méthodes de déduction de clé pour ces « attaques par canaux auxiliaires ». Elles sont introduites par Kocher en 1998 et plus formellement décrit par Thomas Messerges en 1999. Nous nous intéresserons à deux d'entre elles pour l'algorithme AES.

Le fonctionnement du chiffrement Advanced Encryption Standard ou plus communément appelé AES nécessite une clé K et effectue un ensemble d'opérations sur une chaîne binaire de données en entrée. Le standard actuel de la taille de la clé pour ce chiffrement est de 128 bits, cela fait 2^{128} possibilités, un nombre beaucoup trop grand pour tester toutes les combinaisons de manière exhaustive. Cependant les microprocesseurs spécialisés dans l'application de ce chiffrement ne chiffrent que par octet, ce qui diminue considérablement le nombre de possibilités. Il reste tout de même à déterminer la clé octet par octet. Pour cela, comme illustré sur la figure 1, on mesure un canal auxiliaire que l'on compare à la consommation théorique en fonction des clés possibles.

FIG. 1 – AES-128 exécuté sur un microcontrôleur

2 Algorithme Quantique

Traditionnellement on utilise comme mesure le poids de Hamming (HW). Mathématiquement, pour un mot m de taille t où m_i le i^e bit de m : HW(m) = $\sum_{i=0}^{t-1} m_i$. En effet les pics de consommation mesurés correspondent au flip à 1 des bits du registre contenant le texte chiffré. Ainsi, le nombre de pics observés correspond directement au poids de Hamming. Trouver la clé de chiffrement revient alors à effectuer les étapes du chiffrement précédant le point d'attaque, à évaluer le poids de Hamming du texte à moitié chiffré, à le comparer aux données mesurées, et ceci pour chacune des clés possibles. C'est ici que les propriétés apportées par le quantique vont nous être utiles, nous pouvons en effet effectuer le chiffrement pour toutes les clés possibles simultanément en les superposant. Il suffira ensuite d'extraire la clé voulue à l'aide de l'algorithme de Grover. C'est un problème SAT.

Algorithme 6 : ORACLE NAÏF POUR MESURE NON BRUITÉE

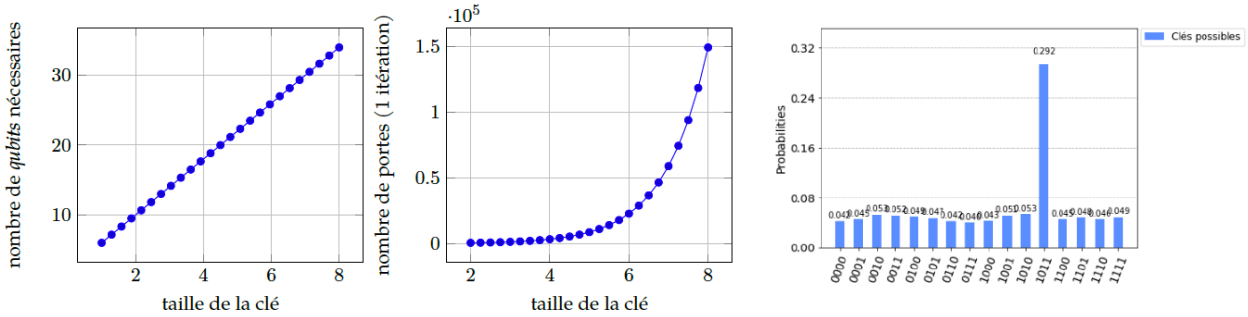
Données : M une liste de messages, $SubBox$ le tableau de substitution, $trace$ une liste de mesures, $|cle\rangle$ de taille n représentant la clé, $|q\rangle$ un *qubit* pour l'algorithme de Grover

```
1  début
2      INITIALISATION :
3      |enc⟩ un registre de taille n                /* le chiffré */ ;
4      |buffer⟩ un registre de taille n;
5      |hamming⟩ un registre de taille n + 1;
6      |p⟩ un registre de taille m ;
7      |ancilla⟩ un registre de taille max(n − 1, m − 1)   /* qubits ancillas
        nécessaires */ ;
8      TRAITEMENT :
9      pour chaque message ∈ M faire
10         /* Chiffrement du message                                    */
11         ENCODE (message, |buffer⟩);
12         XOR (|cle⟩, |buffer⟩) ;
13         SUBBYTES(SubBox, |buffer⟩, |enc⟩) ;
14         nettoyage du registre |buffer⟩ ;
15         HW(|enc⟩, |hamming⟩)          /* calcul du Hamming */ ;
16         ENCODE (mesure[message], |buffer⟩);
17         EGALITÉ(|somme⟩, |buffer⟩, |p[message]⟩)   /* calcul du prédicat
            (mesure = Hamming) */ ;
18         nettoyage des qubits hors |p⟩;
19     NCNOT (|p⟩, |ancilla⟩, |q⟩;
20     nettoyage des qubits ;
```

Evidement la vie est plus compliquée, et souvent les données observées sont bruitées. Il est possible de transformer notre problème SAT en un problème d'optimisation en minimisant la somme de la valeur absolue de la différence entre le poids de Hamming calculé et celui mesuré, sur plusieurs messages consécutifs.

Ces deux algorithmes (recherche de Hamming et recherche différentielle dans le cas bruité) ont été implémentés sous forme d'oracle en créant les opérateurs nécessaires : valeur absolue, somme, égal et XOR. Ce sont les opérateurs classiques que l'on utilise pour coder des oracles de problèmes NP-complets.

3 Expérimentations et perspectives



Des expérimentations ont été réalisées et validées sur des instances de petites tailles, montrant la possibilité de réaliser ces tests à plus grande échelle au fur et à mesure où les machines réelles apparaitront sur le marché. En 2019 IBM proposait des machines à 27 Qbits, en 2020 une machine à 65 QBits. Cette année en 2021 c'est une machine à 127 Qbits qui vient d'être dévoilée. On annonce des machines autour de 400 QBits en 2022 et plus de 1000 Qbits en 2023 ! A suivre ….