Projet ZZ1 - Informatique embarquée et sécurité
Implémentation du chiffrement AES sur microcontrôleur

Auteurs :	Tuteur :	Année :
FERRAND Lysandre BLANCHARD Marine	LAFFONT Jacques	2016

Remerciements

Nous tenons à remercier Monsieur Jacques LAFFONT, pour nous avoir suivi et aidé tout au long de notre projet.

Introduction - Problématique

Le chiffrement AES (Advanced Encryption Standard) est actuellement un moyen très utilisé et relativement sûr pour sécuriser les transactions en ligne, les communications sans fil ou encore le stockage de données sensibles. Les algorithmes sont basés sur plusieurs substitutions, permutations et transformations linéaires, chacune réalisée sur des blocs de données de 16 octets. Ces opérations sont répétées plusieurs fois, appelées "rondes". Durant chaque ronde, une clé de ronde unique est calculée à partir de la clé de cryptage, et incorporée dans les calculs. Basé sur cette structure de bloc de l'AES, le changement d'un seul bit soit dans la clé, soit dans le bloc de texte brut donne un bloc de texte de chiffrement complètement différent - un avantage sur les chiffrements de flux traditionnels. Ajouté à cela, les systèmes embarqués sont de plus en plus présents dans notre quotidien: téléphone portable, distributeur de billet, box ADSL, GPS…

Le but de notre projet consiste ainsi à évaluer :

Dans quelles mesures est-il possible d'implémenter ce chiffrement sur un microcontrôleur ?

Sommaire

Introduction……..…...1

Remerciements……...……..…...1

I. Technologies utilisées ……..……...3

MPLAB X……...……....3
MPLAB XC8..……...3
Le microcontrôleur PIC18F45K50 ……..…..……..3

II. Évolution du projet…. ……..4

Compréhension du chiffrement AES...……...……...4
Recherche d'un algorithme...…...……...4
Utilisation de MPLAB ..……4

III. Tests - Résultats…...……....5

Tests sur la clé……...……...5
Tests sur le message à crypter/décrypter...……..5
Mesure du temps ……..……...5

Conclusion ……6

I. Technologies utilisées

- MPLAB X

Nous avons utilisé le logiciel MPLAB X tout au long de notre projet. MPLAB est un environnement de développement d'applications embarquées sur microcontrôleurs PIC et dsPIC , et est développé par Microchip Technology. MPLAB X est la dernière édition de MPLAB, elle nous permet d'éditer nos codes, de les déboguer et de programmer des microcontrôleurs. MPLAB X était très utile pour notre projet car contient un simulateur.

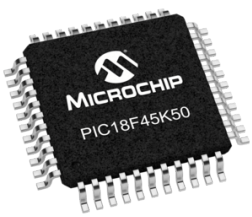
La version que nous avons utilisée est MPLAB X IDE v3.30.
Ce logiciel est entièrement gratuit et peut être téléchargé sur le site de Microchip pour toute plate-forme (Windows, Mac, Linux).

- MPLAB XC8

Le compilateur C que nous avons utilisé est MPLAB XC8 de Microchip dans sa version 1.37. Comme MPLAB X IDE, il est téléchargeable gratuitement sur le site de Microchip, sur toute plate-forme (Windows, Mac, Linux).

- Le microcontrôleur PIC18F45K50

Nous avons développé notre projet pour un microcontrôleur PIC. Un microcontrôleur PIC est une unité de traitement et d'exécution de l'information à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants annexes. Un microcontrôleur PIC peut donc fonctionner de façon autonome après programmation.



Le microcontrôleur utilisé est le *PIC18F45K50*. Il fait donc partie de la famille PIC 18 qui a un jeu d'instruction qui lui permet de faire fonctionner du code C compilé de manière nettement plus efficace que les familles précédentes (PIC10, PIC12, PIC14, PIC16, PIC17). Le F indique qu'il s'agit d'une mémoire flash et donc effaçable électriquement. Il s'agit d'un microcontrôleur 8 bits.

II. Évolution du projet

- Compréhension du chiffrement AES

Nous avons dans un premier temps mené des recherches afin d'assimiler les différentes étapes puis le fonctionnement général du chiffrement AES. Les documentations étant nombreuses (mais surtout en anglais), cette étape fut relativement rapide.

- Recherche d'un algorithme

Dans un second temps, nous avons recherché des algorithmes du chiffrement AES en langage C (ce langage nous étant plus familier que le langage assembleur). Nous avons choisi un code de chiffrement AES 256 (256 bits correspondant à la taille de la clé), à la fois simple et pratique.

- Utilisation de MPLAB

Cependant afin de procéder à l'implémentation dans un microcontrôleur il est nécessaire de faire appel au langage assembleur, c'est pourquoi nous avons eu recours au logiciel MPLAB associé au compilateur XC8. En effet l'utilisation d'un compilateur C facilite la production de code pour les applications embarquées. Afin d'alléger le code principal et permettre une réutilisation rapide il faut créer deux types de fichiers: un fichier d'en-tête (*.h) et un fichier source (*.c). Nous avons donc créé un fichier d'en-tête (ou header file) *aes.h* qui inclue l'ensemble des définitions et des macros utilisables sur le microcontrôleur, puis les fichiers sources *aes.c* et *mainaes.c*, le premier contenant les fonctions permettant notamment la substitution par octet, le décalage par ligne, le mélanges des colonnes ainsi que toutes les sous-fonctions nécessaires aux fonctions de cryptage et de décryptage, le second quant à lui constitue le *main,* il permet d'appliquer les fonctions à un texte et une clé choisie.

Durant cette étape nous avons dû apporter des modifications au code:

- Supprimer les *printf,* inutiles sur un microcontrôleur
- Remplacer les *void* par des *static void* car ils étaient déclarés ainsi mais cette déclaration n'avait pas été reprise (problème de cohérence)

La compilation se faisant sans erreur et les résultats retournés par MPLAB étant similaires, d'une part à ceux renvoyés lors de la compilation sous *gcc* et d'autre part à ceux donnés par le simulateur en ligne *aes.online-domain-tools.com,* nous avons pu entamer une série de tests.

III. Tests - Résultats

- Tests sur la clé

La clé est un mot de 0 à 32 octets de type *unsigned char,* elle peut donc être sous forme de texte.

Nous avons testé la clé avec tous ces octets à 0, ainsi que tous ces octets à 1, cela fonctionne toujours. De plus, nous avons testé en entrant des caractères et cela fonctionne aussi, grâce au type de la clé. Il est également possible d'entrer une clé de taille inférieure à 32 octets.

- Tests sur le message à crypter/décrypter

Le message est un mot de 0 à 16 octets de type *unsigned char,* il peut être entré sous forme de caractères, d'entiers, peut être identique à la clé, et de taille inférieure à 16 octets, le cryptage et le décryptage se font correctement.

- Mesure du temps

Pour mesurer le temps d'exécution, nous avons réglé la fréquence d'instruction à 48 millions d'instructions par seconde (*File* > *Project Properties* > *Simulator* > *Oscillator Options*). Le temps ne varie que peu selon la clé et le texte.

En moyenne :

- Pour le cryptage : 963 µs
- Pour le décryptage : 887 µs

Ces temps varient seulement de quelques microsecondes, et doivent être multipliés par 4 pour représenter les performances du microcontrôleur.

Conclusion

Ce projet n'était pas celui que nous avions choisi mais malgré les difficultés techniques comme la prise en main du logiciel, nous avons apprécié travailler sur ce sujet. Il nous a apporté un premier contact avec les systèmes embarqués en implémentant un algorithme de chiffrement sur un microcontrôleur.

Notre travail a permis de mettre en évidence la possibilité d'adapter un code de chiffrement AES pour le microcontrôleur défini en début de projet. De plus, l'algorithme utilisé permet de crypter et décrypter des messages texte de 16 octets en un temps raisonnable.