Projet Nigma : Deuxième soutenance

CrypTeam : LAPÔTRE Guillaume (lapotr_g)
GANIVET Justin (ganive_j)
LADEVIE Stéphane (ladevi_s)
GISLAIS Sébastien (gislai_s)

Table des matières

1 Introduction **3**

2 Cryptographie **4**
2.1 Partie de Guillaume Lapôtre 5
2.2 Partie de Sébastien Gislais 6

3 Stéganographie **7**
3.1 Partie de Justin Ganivet 7
3.2 Partie de Stéphane Ladevie 8

4 Conclusion **9**

5 Annexe **10**
5.1 Screenshot Cryptographie 11

1 Introduction

2 Cryptographie

Cette fois-ci encore, nous avons respecté ce que nous avions prévu de présenter pour cette seconde soutenance. En effet nous avons implémenté l'algorithme DES (Data Encryption Standard). Contrairement au RSA présenté en première soutenance, le DES est un algorithme de chiffrement symétrique. Cela signifie que pour chiffrer ou déchiffrer un fichier on utilise la même clé. Ce type de chiffrement a ses avantages et inconvénients par rapport au chiffrement asymétrique. Son principal avantage est sa rapidité : il est 1000 fois plus rapide que le RSA ! Son principal inconvénient est le fait de ne possèder qu'une seule clé. Ainsi le problème principal est l'échange de cette clé entre les différents protagonistes qui veulent échanger des données chiffrées. En effet pour le RSA il suffisait de donner sa clé publique aux personnes désirant chiffrer des messages, ensuite ils renvoient les messages chiffrés et on peut les déchiffrer à l'aide de notre clé privé jamais échangé avec personne. Si l'on envoie notre clé DES au n'importe qui, il y a un risque qu'un pirate puisse intercepté la clé et donc s'en servir ensuite pour déchiffrer les messages confidentiels. Cepandant il y a une astuce permettant de contourner le problème : On chiffre la clé DES avec un chiffrement RSA !

Un petit exemple : Alice veut envoyer un message confidentiel à Bob. Bob crée donc un jeu de clé RSA puis envoie la clé publique à Alice. Alice crée sa clé DES, puis chiffre son message à envoyer avec cette clé. Finalement, elle envoie à Bob sa clé DES chiffrée avec la clé publique RSA ainsi que le message chiffré avec la clés DES. Bob reçoit donc les deux fichiers, avec sa clé privé RSA il déchiffrer le fichier de clé envoyé apr Alice, puis avec la clé DES qu'il vient de déchiffrer il peut déchiffrer le message d'Alice !

Le DES a été achevé en 1977, c'est donc un algorithme de chiffrement ancien. Sa sécurité n'est plus optimal, en effet un état peut casser une clé DES en quelque minutes maintenant. Cepandant nons pensons qu'il est intéressant de se pencher sur cet algorithme qui fût un des premiers algorithmes de chiffrement symétrique défini rigoureusement.

Du fait de son ancienneté les spécifications du DES étaient prévus pour une réalisation matérielle. C'est à dire pour que ce soit des puces qui réalisent le chiffrement. Ainsi certaines partie du protocole sont faciles à réaliser sur une puce mais sont difficile à réaliser en Ocaml. Nous développerons dans nos parties respectives.

2.1 Partie de Guillaume Lapôtre

Pour implémenter le DES en Ocaml je me suis tout d'abord documenté sur cet algorithme de chiffrement complexe bien qu'utilisant que des opérations basiques (en effet le DES est optimisé pour une réalisation matérielle ...). Je me suis donc basé sur un livre qui m'a été bien utile pour la l'implémentation de RSA qui se nomme "Cryptographie Appliquée" par Bruce Schenier. Ce livre est très pratique car il explique très bien comment fonctionne un bon nombre d'algorithme de chiffrement et il y a aussi l'histoire de chaque algorithme : Comment il a été créé, par qui, suite à quels besoins etc...

J'ai donc tout d'abord commencé par la création d'un clé DES. Sa création fût bien plus simple que la création d'une paire de clé RSA. En effet, la génération des clefs RSA nécessitent de possèder au préalable de grands nombres premiers qui furent quelque peu difficile à obtenir. Une clé DES est un nombre de 64 bits. Puis le DES ne se sert que de 56 bits des 64 bits présents initialement. Les 8 bits inutilisés sont des bits de parités. Ils servent à vérifier lorsque l'on envoie la clé que la clé reçue n'est pas corompue. C'est à dire qu'aucun bit n'a changé au court du transfert. Ainsi, une clé DES à la propriété d'avoir un nombre pair de bits à 1 pour chaque octet. Ce sont les 8 bits de parité qui permettent d'assurer cette propriété.

Je me suis ensuite occupé des différentes fonctions de manipulation de la clef. Tout d'abord il y a une permutation de clef qui réarrange les bits dans un ordre prédéfini et qui ignore les bits de parité. On se retrouve donc avec une clé de 56 bits. Ensuite à chaque ronde du DES (le protocole exacte d'une ronde du DES sera expliquée par Sébastien), on sépare la clé en deux sous-clés de 28 bits chacune et on effectue une ou deux rotations gauche suivant la ronde que l'on est en train d'effectuer. Ensuite on réassemble les deux sous clé puis on effectue une permutation compressive qui compresse la clé qui était de 56 bits en une clé de 48 bits. L'orde de lequel les bits sont réarrangés est spécifié dans le DES. Ainsi j'ai pu générer les 16 sous-clés requises pour le chiffrement d'un bloc.

Je me suis ensuite occupé d'une toute autre partie qui est la gestion des S-box ou tables de substitution.

2.2 Partie de Sébastien Gislais

3 Stéganographie

3.1 Partie de Justin Ganivet

3.2 Partie de Stéphane Ladevie

4 Conclusion

5 Annexe

5.1 Screenshot Cryptographie