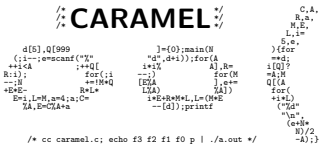
Introduction à la Cryptographie

éléments pour le projet

E. Thomé

Projet CARAMEL, INRIA Nancy



ESIAL, 2A TRS — janvier-février 2012

http://www.loria.fr/~thome/esial-crypto/

Plan

Introduction

Système de hachage de mots de passe

Attaques

Difficultés et recommandations

Objectif du projet

Plusieurs objectifs :

- Imaginer un système de hachage de mots de passe.
- Mettre en place différentes attaques.

Projet à réaliser :

- Dans le langage de votre choix (Java, C, C++, Caml, python, perl, . . .)
- En favorisant l'emploi de bibliothèques existantes le cas échéant.

Plan

Introduction

Système de hachage de mots de passe

Attaques

Difficultés et recommandations

Système de hachage de mots de passe

On demande de mettre en place une fonction de hachage de mots de passe utilisant un chiffrement par blocs.

Choix à faire :

- Un système $\mathcal{S}$ de chiffrement par blocs, parmi ceux qui sont disponibles dans le langage de votre choix (utiliser des bibliothèques !).
- Un mécanisme de transcription mot de passe $\rightarrow$ clé pour $\mathcal{S}$.
- Un mécanisme de transcription cryptogramme $\rightarrow$ version lisible.

Principe de calcul demandé : haché $= E_{\mathrm{mdp}}(IV)$, où IV est une constante de votre choix.

Système de hachage de mots de passe

On demande :

- Une implantation (fonction renvoyant le haché sous forme de chaîne à partir d'un mdp).
- Différentes versions appelées h_4, h_6, et h_8, restreintes aux mots de passe de 4,6,8 caractères.
- Un document spécifiant notamment certains aspects comme :
  - organisation du code, fonctionnalités employées ;
  - gestion (ou pas) des caractères non alphabétiques ;
  - gestion mots de passe plus courts.

Plan

Introduction

Système de hachage de mots de passe

Attaques

Difficultés et recommandations

Attaque 1 : force brute

On demande d'écrire une fonction de recherche exhaustive de mots de passe parmi l'ensemble des mots de passe admissible.

On demande :

- Une implantation.
- Une description de l'organisation du code.
- Indications de performance (ou projection du temps pris) pour les fonctions h_4, h_6, et h_8, sur une ou plusieurs machines dont les caractéristiques doivent être indiquées.

Bonus éventuels :

- Utilisation du parallélisme (plusieurs machines, ou plusieurs coeurs d'une même machine).

Attaque 2 : dictionnaire

On demande une version spécifique adaptée à la recherche par dictionnaire.

On demande :

- De constituer une liste de mots ou de prénoms (à trouver sur le web).
- Une implantation de la recherche.
- Une description de l'organisation du code.
- Indications de performance (ou projection du temps pris).

Variations possibles :

- Parallélisme.
- Autoriser ou pas la capitalisation des initiales.
- Autoriser ou pas les mots avec accents ou ponctuation.
- Autoriser plusieurs mots.
- Organiser la recherche par fréquence.

Attaque 2bis : dictionnaire + chiffres

Étendre la recherche par dictionnaire à l'ajout de chiffres avant ou après le mot choisi.

Éléments de différenciation :

- Trouver plus de mots de passe, sans prendre plus de temps que l'implantation précédente sur les mots de passe qui ne contiennent pas de chiffre.

Attaque 3 : Meet-in-the-middle

Attaque 3 : Meet-in-the-middle

Vous êtes devenu célèbre ! Votre système h_4 est désormais présent dans de nombreux circuits intégrés.

Implanter un système h_{4x} prenant des clés de 8 caractères, et tel que :

$$h_{4x}(\text{abcdefgh}) = E_{\text{abcd}}(E_{\text{efgh}}(IV)).$$

Écrire une procédure de meet-in-the-middle (cf cours 3).

- Précalcul des $E_k(IV)$ pour chaque demi-clé k.
- Tri de $\mathcal{L} = \{E_k(IV)\}$.
- Pour chaque 2ème moitié de clé k', rechercher $D_{k'}(\text{cible})$ dans $\mathcal{L}$.
- Afficher le mot de passe trouvé en cas de succès.

Attaque 4 : tables arc-en-ciel

Une attaque originale, mais avec des précalculs : les tables arc-en-ciel.

Attaque 4 : tables arc-en-ciel

Une attaque originale, mais avec des précalculs : les tables arc-en-ciel. (ce que j'écris au tableau se retrouve facilement sur le net).

Plan

Introduction

Système de hachage de mots de passe

Attaques

Difficultés et recommandations

Difficultés

Normalement il est possible de faire quelque chose de toute façon.

Quelques points pas évidents :

- Les bibliothèques ont des APIs parfois curieuses, ou exagérément complexes.
- Il y a moyen de se faire mordre par les points subtils de la gestion des chiffrés (caractères nuls, conversion sous forme lisible).

Certains développements plus difficiles sont valorisés, mais nullement exigés :

- parallélisme.
- idées d'optimisation de code.

Recommandations

Ne pas réinventer (mal) la roue :

- Utiliser les bibliothèques.
- Structurer votre code.

Ne pas passer trop de temps.

- On ne demande pas de tout faire,
- Mais ce qui est fait doit être bien fait.

On demande de fournir pour le 30 mars.

- une version source du code.
- un document court de présentation de comment il marche (2p à 5p max).

Je suis joignable par mail, `Emmanuel.Thome@gmail.com`.