Implémentation d'une fonction de hachage dans la librairie Crypto++

5 décembre 2011

1 Introduction

Le but du projet est d'intégrer la nouvelle fonction de hachage Blake à la librairie cryptographique Crypto++.

Blake est un des algorithmes finalistes du concours SHA-3 (http ://csrc.nist.gov/groups/ST/hash/sha-3/index.html). Ce concours, lancé par le NIST, a pour but de déterminer le nouveau standard dans les fonctions de hachage, remplaçant ainsi les précédentes fonctions de la famille des SHA.

2 Etude de la librairie Crypto++

Nous allons dans un premier temps étudier la librairie Crypto++ et l'implémentation de la fonction de hachage SHA1.

2.1 Téléchargement et installation

Télécharger la version 5.6.1 de la librairie à partir du site http ://www.cryptopp.com

Installation de la librairie sous Linux/ MacOs :

Après avoir décompressé l'archive tapez les commandes suivantes :

```
cd cryptopp561
make
sudo make install
make clean
```

Des erreurs peuvent apparaitre après le passage de la commande *sudo make install* mais celles-ci n'empêchent pas le bon fonctionnement de la librairie. Pour plus de détail sur l'installation, vous pouvez consulter la page web http ://www.cryptopp.com/wiki/Compiling.

2.2 Analyse du code source

Nous allons analyser le code source de la librairie contenu dans le dossier cryptopp561. Les réponses devront être détaillées.

– Dans quel fichier est déclarée la classe SHA1 ?

- Dans ce fichier, à quoi correspondent CRYPTO_DLL et CRYPTO_API ? Peuvent-ils provoquer des erreurs de compilation ?
- Quelles sont les méthodes déclarées dans la classe SHA1 ?
- Effectuez un diagramme d'héritage de la classe SHA1.
- Dans la classe HashTransformation, quel est le role des méthodes Update et Final ? Les fonctions étant virtuelles, dans quelles classes et quels fichiers sont réellement définies ces méthodes lorsqu'elles sont appellées depuis la classe SHA1 ?
- Quel est le rôle des méthodes DataBuf et StateBuf de la classe IteratedHashBase ? Ces méthodes étant virtuelles, dans quelles classes parents de SHA1 et dans quels fichiers sont elles définies ?
- Quel est le rôle de HashBlock ? Quel est lien entre cette méthode et la méthode SHA1 : :Transform ?
- Quelle est la méthode de la class IteratedHashBase qui initialise l'état courant (state) avant le hachage ?
- Quel est la méthode appelée lorsque le dernier bloc de données est traité ? Quelle méthode effectue le padding ?
- Vérifiez que les fonctions SHA1 : :InitState et SHA1 : :Transform définies dans sha.cpp implémentent bien l'algorithme SHA-1 (cf http ://en.wikipedia.org/wiki/SHA-1 pour le pseudo code de l'algorithme).
- En utilisant les méthodes Update et Final de la classe SHA1, écrivez un programme qui calcule et affiche le hash de "hello world".

Pour vous aider, la commande sous Linux

```
find . | xargs grep "votre recherche"
```

permet de trouver "votre recherche" dans l'ensemble des fichiers présents dans le dossier courrant et les sous dossiers.

Pour compiler un fichier test.cpp utilisant la librairie cryptopp, on pourra utiliser utiliser la commande

```
g++ -o test test.cpp -lcryptopp
```

3 Implémentation de BLAKE-256 dans la librairie Crypto++

Implémentez Blake-256 dans la librairie Crypto++. La spécificationde l'algorithme est disponible sur la page web http ://www.131002.net/blake/, rubrique download, supporting documentation.

Dans un premier temps on pourra créer une classe Blake256 qui héritera de la classe IteratedHashWithStaticTransform. On définira alors les méthodes de classe InitState, Transform et StaticAlgorithmName. Inspirez-vous pour celà de l'implémentation de SHA-1. Dans cette première version le compteur de données ne sera pas pris en compte dans le calcul du hash (on considèrera alors qu'il reste nul). Pourquoi imposer une telle limitation ? De même le padding utilisé sera celui de SHA-1.

Dans un second temp vous modifierez votre code pour prendre en compte le compteur de donnée. Il n'est pas interdit de toucher à la librairie. Il faudra aussi modifier la fonction de padding.

N'oubliez pas de copier vos fichier dans le dossier cryptopp561, de modifier la variable *DLLSRCS* du Makefile et de recompiler/réinstaller la librairie avec les commandes *make/sudo make install*.

Enfin vous écrirez un programme qui calcule le hash de "hello world" avec l'algorithme BLAKE-256 de votre nouvelle librairie.