Auteurs : *Gautier MAURICE*
Date : 15/01/2020

Compte-rendu du TP A : Authentification de courriels

Auto-évaluation

- Même si le code marche, il pourrait être plus performant.
+ Deux utilisateurs utilisant ce code pourraient vérifier l'authenticité de leur mails de manière sécurisé (le code fonctionne).
+ Découpage partie mail / partie hachage → modularité.
+ Passage d'une fonction de hachage à une autre facilement → modularité.
+ Réutilisation de fonctions existantes pour chaque fonction de hachage.
+ Code documenté.

Pour la note cela dépend du barème. Sinon pour répondre au 'degré de réussite', les exécutables peuvent être directement utilisés dans un contexte de production.

Pédagogie

Ce TP m'as appris que selon les éditeurs de textes et les OS, le retour à la ligne pouvait être encoder différemment. Pour contourner cette difficulté, il suffisait d'ignorer les retours lignes en utilisant d'autre repères ainsi que s'assurer d'avoir la main mise sur l'écriture dans un fichier. Différencier le code hexadécimal d'un caractère avec du code hexadécimal stocké dans un caractère nécessitait des conversions plus ou moins difficiles. Pour résoudre cette difficulté il fallait être capable de toujours travailler avec la même unité, c'est à dire dès qu'un résultat change de type, il fallait tout de suite le convertir.

Test

La valeur de l'appendice HMAC obtenu à l'exercice A.4 pour l'email 'email1.txt' fourni ce TP est :

bc4ccdb7e941ee72ed343fe60a9925cc

Codage (modularité)

Les fonctions principale qui ont été implémentées sont :
Mailer::readCorpse() : string lie le corps d'un email.
Mailer::getHeaderValue(header_name : string) : string lie la valeur du champs passé en paramètre.
Mailer::addHeader(header_name : string, header_value : string) : void ajoute une ligne aux headers de l'email.
HMAC::get_soft_hash(secret : string) : string retourne le résumé spécifié dans l'énoncé du TP.
HMAC::get_RFC_2104_hash(secret : string) : string retourne le résumé spécifié dans la question A.4 du TP.

Pour plus d'information une documentation doxygen est accessible depuis le fichier : 'doc/index.html'. Les méthodes privées sont documentées dans les fichiers du dossier 'include/'.

Compilation et exécution

Pour compiler les sources : '$ cmake .' puis '$ make'.
Vous devez donc disposé de Cmake (et des bibliothèques standard C).
Deux exécutables seront créés : 'bin/cert' et 'bin/check' ; les deux nécessitent en argument un chemin vers l'email qui doit être traité.

Un fichier tmp.tmp.tmp sera créé lors de l’exécution de ‘bin/cert’ pour la fonctionnalité qui permet de rajouter une ligne aux headers de l’email.