Лабораторна робота № 15

Коди аутентифікації повідомлень на основі хеш-функцій

Мета: навчитися формувати коди аутентифікації повідомлень з використанням традиційних криптографічних хеш-функцій типу MD, SHA1-2.

Обладнання: персональний комп'ютер.

Програмне забезпечення: будь-яка мова програмування, криптофреймворк або криптобібліотека, де реалізовано популярні криптографічні хеш-функції типу MD5, SHA1-2, RIPEMD та HMAC на їхній основі. Бажано також мати реалізацію HMAC на основі цих функцій.

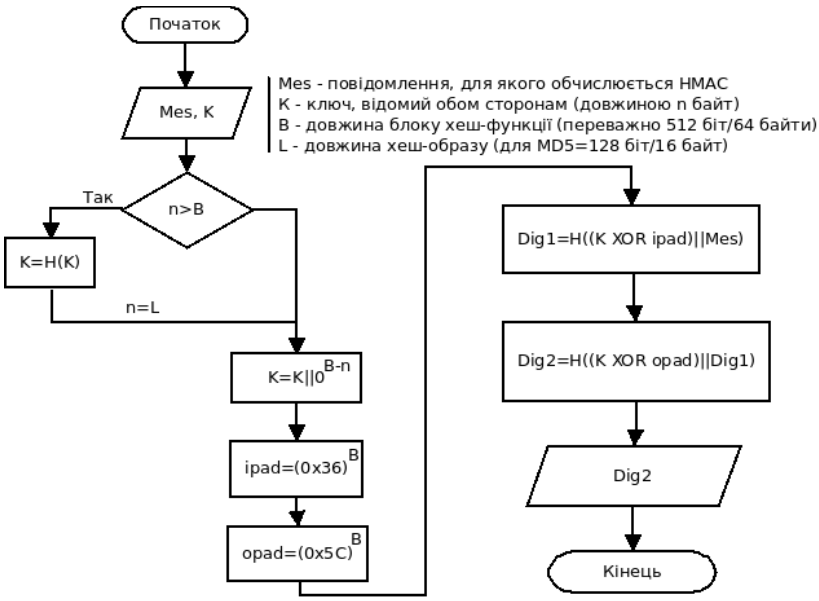
Література:

1. Вікіпедія: https://en.wikipedia.org/wiki/Hash-based_message_authentication_code
2. RFC 2104: HMAC: Keyed-Hashing for Message Authentication;
3. FIPS PUB 198-1: The Keyed-Hash Message Authentication Code (HMAC)

Теоретична частина

Якщо учасники інформаційного обміну мають спільний ключ симетричної системи шифрування, то його можна використати для аутентифікації повідомлень без застосування електронного цифрового підпису. Це так званий HMAC (Hash-based Message Authentication Code).

Схему такого криптографічного перетворення подано на рисунку.



На вхід подається повідомлення *Mes*, для якого треба обчислити HMAC, та ключ *K*, відомий обом учасникам інформаційного обміну. Довжина ключа *n* може бути довільною. Однак важливим є те, наскільки ключ більший або менший за довжину вхідного блоку *B* обраної для перетворення хеш-функції. Як правило, довжина вхідного блоку більшості популярних хеш-функцій складає 512 бітів (64 байти). Отже, якщо ключ *K* більший за 64 байти, його вкорочують, утворивши хеш-образ. Довжина хеш-образу *L* завжди менша за *B* (наприклад, для MD5 *B*=64 байти, а *L*=128 бітів або 16 байтів), тому отриманий хеш-образ ключа доповнюють до розміру *B* нулями. Для MD5 це виглядає так: новий ключ $K=K||0^{B-n}=K||0^{64-16} = K||0^{48}$, тобто необхідно буде додати до хеш-образу ключа 48 нулів.

Якщо ж оригінальний ключ менший за *B*, його просто доповнюють нулями до 64 байтів.

Наступний етап — створення констант *ipad* (*inner padding* — внутрішнє доповнення) та *opad* (*outer padding* — зовнішнє доповнення): $ipad=0x36^B$ (*B*-разове повторення 16-кових 36),

o*pad*=0x5C^B (*B*-разове повторення 16-кових 5C). Ці константи використовують для додавання до модифікованого ключа за модулем два: (*K* XOR *ipad*); (*K* XOR *opad*).
Використовуючи ці конструкції, обчислюють HMAC: *Digest*1=H((*K* XOR *ipad*)||*Mes*) та *Digest*2=H((*K* XOR *opad*)||*Digest*1).
Це означає, що до (*K* XOR *ipad*) додається (конкатенується) оригінальне (не доповнене) повідомлення *Mes*, результат першого (внутрішнього) хешування, *Digest*1, так само конкатенується з константою (*K* XOR *opad*), і знову хешується. Виходом, тобто кодом аутентифікації повідомлення *Mes*, буде величина *Digest*2.
В канал зв'язку відправляється саме повідомлення (відкрите або зашифроване) разом з *Digest*2 — кодом аутентифікації повідомлень, обчисленим за наведеним алгоритмом.
На приймальному боці, отримавши повідомлення, обчислюють HMAC та порівнюють з отриманим з мережі. Якщо вони співпадають, то:
- повідомлення не зазнало змін під час передавання мережами зв'язку;
- автором повідомлення є власник ключа *K* і ніхто інший (якщо ключ не скомпрометовано, звичайно).
Вказаний метод захищає як цілісність, так і аутентичність повідомлення.
Перевагою такого методу є те, що швидкість роботи традиційних хеш-функцій значно більша як за швидкість блокового шифру (особливо DES), так і за обчислення та перевірку електронно-цифрового підпису.
Таким чином, обчислення HMAC може бути, деякою мірою, швидкою альтернативою ЕЦП.

Практична частина

1. Використовуючи обрану криптобібліотеку, розробіть систему обчислення та перевірки HMAC для довільного повідомлення. В якості функцій хешування оберіть будь-які, присутні у вашій криптобібліотеці (за винятком SHA-3, яка сама може генерувати HMAC).
2. Розроблена система повинна задовольняти такі функціональні вимоги:
- генерувати ключ для використання в HMAC та розповсюджувати його на усі сторони інформаційного обміну за допомогою будь-якого захищеного методу;
- дозволяти обчислювати та перевіряти HMAC для усіх уасників інформаційного обміну;
- повідомляти користувача про успішність/неуспішність перевірки HMAC.
3. Для перевірки правильності реалізації HMAC необхідно використати контрольні приклади з FIPS PUB 198-1. Якщо у Вашій криптобібліотеці є реалізація HMAC, можна виконати перевірку з її допомогою.
4. При виконанні лабораторної роботи необхідно дотримуватися усіх правил написання коду, прийнятих у програмній інженерії. Змінні типу Form1, Procedure1, Function1, a1,b2,c3 — не допускаються.
5. Код необхідно супроводжувати детальними коментарями, з яких було би зрозуміло, що виконує кожна процедура/функція/клас/об'єкт тощо.
6. Звіт з лабораторної роботи повинен містити:
- Протокол дій з обчислення та перевірки HMAC та генерування ключів;
- Приклади HMACдля конкретних повідомлень та ключів4;
- Код системи;
- Відповіді на контрольні запитання.

Контрольні запитання

1. Архітектура та основні складові частини HMAC. Чому, на Вашу думку, обрано таку архітектуру?
2. В чому Ви вбачаєте переваги та недоліки HMAC?
3. Які властивості інформації захищає HMAC і чому Ви так вважаєте?
4. В чому відмінність HMAC від електронного цифрового підпису? Обгрунтуйте Вашу відповідь.

5. В чому відмінність HMAC від звичайного блокового шифрування? Обгрунтуйте Вашу відповідь.
6. Чи можна використати HMAC для аутентифікації користувачів, припустимо, в комп’ютерній мережі чи для локальної аутентифікації? Обгрунтуйте Вашу відповідь.