ЛАБОРАТОРНА РОБОТА № 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ХЕШ-ФУНКЦІЙ

3.1 Мета роботи

Ознайомитись з можливостями криптографічних хеш-функцій при організації контроля цілісності цифрових об’єктів та отримати навички їх використання

3.2 Теоретичні відомості

3.2.1 Поняття хеш-функції

Хеш-функція – це обчислювально ефективна функція, що відображає двійковий рядок довільної довжини l в двійковий рядок деякої фіксованої довжини m (зазвичай $m << l$), звану згорткою (хеш-значенням, дайджестом).

Особливий інтерес представляють *односпрямовані хеш-функції*, для яких практично неможливо відновити вхідну послідовність за її згорткою. Велику важливість мають *хеш-функції, вільні від колізій*, тобто хеш-функції, для яких обчислювально складно знайти два повідомлення з однаковими хеш-значеннями. Значущою властивістю хеш-функцій є *властивість повного перемішування* – зміна хоча б одного біта вхідної послідовності призводить до зміни в середньому половини бітів хеш-значення. Хеш-функції, що володіють властивостями свободи від колізій, односпрямованості і повного перемішування, називаються *криптографічними хеш-функціями*.

Криптографічна хеш-функція називається *ключовою*, якщо значення згортки залежить не тільки від повідомлення, що хешується, але і від секретного ключа. Значення ключової хеш-функції зазвичай називається кодом аутентифікації (MAC - Message Autentification Code) або імітовставкою.

3.2.2 Криптографічні хеш-функції на основі симетричних блокових алгоритмів

Криптографічний хеш-функцію можна побудувати, використовуючи

симетричний блоковий алгоритм. Найбільш очевидний підхід полягає в тому, щоб шифрувати повідомлення M за допомогою блочного алгоритму в режимі CBC або CFB за допомогою фіксованого ключа і деякого вектору ініціалізації V. Останній блок шифртексту можна розглядати в якості хеш-значення повідомлення M. При такому підході не завжди можливо побудувати безпечну односпрямовану хеш-функцію, але завжди можна отримати код аутентифікації повідомлення MAC.

Більш безпечний варіант хеш-функції можна отримати, використовуючи блок повідомлення в якості ключа, попереднє хеш-значення – в якості входу, а поточне хеш-значення – як вихід. Реальні хеш-функції проектуються ще більш складними. Довжина блоку зазвичай визначається довжиною ключа, а довжина хеш-значення збігається з довжиною блоку. Оскільки більшість блокових алгоритмів є 64-бітовими, деякі схеми хешування проектують так, щоб хеш-значення мало довжину, рівну подвійній довжині блоку.

Якщо прийняти, що хеш-функція, що отримується, коректна, безпека схеми хешування базується на безпеці блочного алгоритму, що лежить в її основі. Схема хешування, у якій довжина хеш-значення дорівнює довжині блоку, показана на рисунку 3.1. Її робота описується виразами:

$$H_0 = I_H,$$

$$H_i = E_A(B) \oplus C,$$

де I_H – деяке випадкове початкове значення;

A, B і C можуть приймати значення M_i, H_{i-1}, $(M_i \oplus H_{i-1})$ або бути константами.

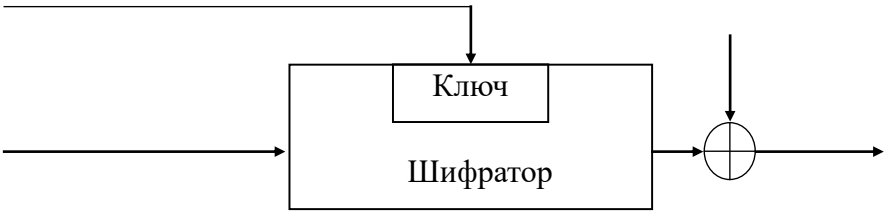

Рисунок 3.1 – Узагальнена схема формування хеш-функції

Повідомлення M розбивається на блоки M_i прийнятної довжини, які обробляються по черзі.

Три різні змінні A, B і C можуть приймати одне з чотирьох можливих значень, тому в принципі можна отримати 64 варіанти загальної схеми цього типу.

3.2.3 Проблема контроля цілісності

Цілісність – це властивість інформації, яка полягає в її існуванні в неспотвореному вигляді (незмінному по відношенню до деякого фіксованого її стану).

Яким способом можна забезпечити цілісність інформації? Можна, звичайно ж, вчинити дуже просто, записавши цю інформацію на деякий фізично незмінний носій (наприклад, CD-R). Але чи вирішить це проблему? Носій може бути підмінений, і, значить, ми просто підмінимо проблему забезпечення цілісності інформації проблемою забезпечення цілісності носія. Більш того, в більшості випадків інформація, для якої має бути дотримана цілісність, зберігається на мережевих дисках, повинна бути загальнодоступною і відкритою для зміни користувачем, що має відповідні права.

Криптографія надає універсальний засіб контролю цілісності повідомлень – *хеш-функції*.

Хеш-функції можуть бути використані для забезпечення цілісності даних в такий спосіб. У певний момент часу обчислюється згортка h(M), відповідна конкретним вхідним даними M. Будемо називати h(M) еталонним хеш-значенням. За допомогою організаційних і/або інженерно-технічних заходів забезпечується цілісність h(M). Надалі для контролю цілісності M обчислюється згортка інформації, що перевіряється, M' – h(M') і порівнюється з еталонним хеш-значенням h(M). Якщо h(M) = h(M'), інформація є цілісною, інакше цілісність порушена. Конкретним застосуванням такої технології є контроль цілісності програмного забезпечення.

Якщо необхідно, щоб перевірка цілісності була доступна тільки певній особі (групі осіб), а також в якості додаткового рівня захисту від підміни еталонного хеш-значення, може бути використана ключова криптографічна хеш-функція.

Слід зазначити, що контроль цілісності за допомогою криптографічних хеш-функцій за визначенням не є абсолютно достовірним, однак може бути виконаний з необхідною точністю.

3.3 Завдання для лабораторної роботи

1. Розробити свою хеш-функцію, яка в якості результату отримує значення 2, 4, 8 біт, яка при зміні будь-якого байту в тексті, призводить до зміни не менше 30% результату.
2. Створити три документа (документ word, вихідний текст на будь-якій мові програмування source, зображення)
3. Обчислити дайджест повідомлення для кожного з файлів.
4. Внести необхідні зміни в документи, так, щоб дайджест повідомлення нового документа співпали зі старим.
5. Реалізувати програму що дозволяє автоматично робити колізію хеш-функції.
6. *Реалізувати веб-інтерфейс, який дозволить завантажувати файли та перевіряти дайджест повідомлення.
7. *На основі RSA реалізувати цифровий підпис дайджест повідомлення.

(*) додатково

3.4 Зміст звіту

Звіт повинен включати в себе:

- титульний лист;
- мету лабораторної роботи;
- завдання і тексти програм;
- результати виконання програм;
- висновки.

3.5 Контрольні питання

1. Поняття хеш-функції.
2. Поняття криптографічної хеш-функції.
3. Поняття колізії.
4. Поняття ключової хеш-функції.
5. Властивості односпрямованості і повного перемішування
6. Можливості побудови хеш-функцій на основі симетричних блокових алгоритмів.
7. Використання хеш-функцій при вирішенні завдання контролю цілісності.