Опишіть основні проблеми використання симетричних алгоритмів шифрування.

1. Проблема розподілу ключів

- Для роботи симетричних алгоритмів потрібен спільний секретний ключ між відправником та отримувачем.
- Передача цього ключа через незахищений канал створює ризик перехоплення ключа зловмисником.
- Для кожної пари користувачів необхідно зберігати окремий ключ, що ускладнює масштабованість.

2. Відсутність аутентифікації

- Якщо більше двох сторін обмініюються повідомленнями за допомогою спільного ключа, неможливо аунтетифікувати відправника.

3. **Відсутність цифрового підпису**

- не діє неможливість відмови

Чому в режимі RC5-CBC-Pad алгоритму RC5 відкритий текст, довжина якого кратна кількості блоків, все одно доповнюється байтами заповнювача?

В режимі **RC5-CBC-Pad** (Cipher Block Chaining with Padding) алгоритму RC5, навіть якщо довжина відкритого тексту кратна довжині блоків, до нього додається спеціальний байт або блок заповнювача. Основною причиною цього є необхідність гарантувати однозначне розшифрування і правильне видалення заповнювача.

Доповнення допомагає чітко зрозуміти, де закінчується оригінальне повідомлення після дешифрування. Без доповнення для повідомлень, довжина яких кратна блоку, важко зрозуміти, чи додавалися байти доповнення.

Опишіть основні підходи до побудови системи розподілу ключів за допомогою симетричних алгоритмів шифрування.

Розподіл ключів є найважливішою проблемою симетричного шифрування, тому є кілька підходів до розподілу ключів симетричних алгоритмів шифрування:

- а) ключ вибирається стороною А та фізично доставляється напряму до В;
- б) створити ключ може третя сторона С та доставити до А і В;
- в) якщо обидві сторони (А і В) мають встановлений захищений зв’язок із третьою стороною С, то сторону С можна використати як проміжну. В такому випадку ключ передається від А до С, а сторона С відправляє його стороні В;
- г) якщо сторони А та В мають старий спільний ключ, то для передачі нового ключа сторона А може зашифрувати новий ключ старим і передати стороні В.

При виконанні проекту необхідно створити програмний модуль передачі даних, який буде забезпечувати конфіденційність і цілісність інформації. Напишіть відповідну схему використання криптографічних примітивів. 👍

EK[M||H(M)]

1. Конфіденційність: Повне повідомлення M і його хеш H(M) зашифровані ключем K, тому їх може прочитати лише отримувач, який знає ключ K.

2. Цілісність: Отримувач може розшифрувати M, обчислити H(M) самостійно та порівняти з отриманим хешем для перевірки, чи не було змінено повідомлення.

При виконанні проекту необхідно створити програмний модуль передачі даних, який буде забезпечувати конфіденційність і цифровий підпис інформації. Напишіть відповідну схему використання криптографічних примітивів. 👍

EKUb[EKRa[M]]

1. Конфіденційність: оскільки повідомлення шифрується публічним ключем отримувача (EKUb), і тільки він може його розшифрувати.
2. Цифровий підпис: оскільки повідомлення підписується приватним ключем відправника (EKRa), що підтверджує його автентичність.

При виконанні проекту необхідно створити програмний модуль передачі даних, який буде забезпечувати конфіденційність і аутентифікацію інформації. Напишіть відповідну схему використання криптографічних примітивів. 👍

При традиційному шифруванні - A відправляє повідомлення B: Ek[M].

1. Конфіденційність: оскільки повідомлення шифрується секретним ключем k
2. Аутентифікація забезпечується через секретний ключ k бо його знають тільки відправник та отримувач.

При асиметричному шифруванні - A відправляє повідомлення B: Ekub[Ekra[M]] (ця схема також забезпечує неможливість відмови).

3. Конфіденційність: оскільки повідомлення шифрується публічним ключем отримувача (EKUb), і тільки він може його розшифрувати.
4. Аутентифікація забезпечується через підпис відправника його приватним ключем (EKRa)

При виконанні проекту необхідно створити програмний модуль передачі даних, який буде забезпечувати аутентифікацію і цілісність інформації. Напишіть відповідну схему використання криптографічних примітивів. 👍

A -> B M || Ek[H(M)]

1. Забезпечує цілісність повідомлення, оскільки хеш-функція H(M) перевіряє, чи не було змінено повідомлення.
2. Аутентифікація забезпечується за допомогою секретного ключа k бо його знають тільки відправник та отримувач.

M||CK(M)|| H(M)

3. забезпечує цілісність повідомлення, оскільки хеш-функція H(M) перевіряє, чи не було змінено повідомлення.
4. Аутентифікація забезпечується за допомогою функції MAC CK(M), яка підтверджує, що повідомлення надійшло від авторизованого відправника, оскільки він має доступ до секретного ключа для створення MAC.

Перелічіть і опишіть основні властивості функцій хешування.

1. **Застосовність до даних довільної довжини**:
   - Хеш-функція повинна бути здатною обробляти блоки даних будь-якої довжини.
2. **Фіксований розмір вихідного значення**:
   - Незалежно від розміру вхідних даних, результат (хеш-код) завжди має фіксовану довжину.
3. **Ефективність обчислення**:
   - Значення H(M) повинно бути відносно легко обчислюваним для будь-якого повідомлення M, забезпечуючи швидкість та зручність реалізації.
4. **Односторонність**:
   - Для будь-якого хеш-коду hhh практично неможливо знайти таке M, що H(M)=h. Це означає, що функція хешування є необоротною.
5. **Слабка стійкість до колізій**:
   - Для будь-якого блоку xxx практично неможливо знайти інший блок y≠x, для якого H(x)=H(y)
6. **Сильна стійкість до колізій**:
   - Практично неможливо знайти будь-яку пару x та y (x≠y), для яких H(x)=H(y).
7. **Практичні для програмної та апаратної реалізації**

Опишіть найпростішу схему розподілу таємних ключів за допомогою системи з відкритим ключем.

A→B: EKUb[N1 || IDA]
B→A: EKUa[N1 || N2]
A→B: EKUb[N2]
A→B: EKUb[EKRa[KS]]

Версія 1 - складніший опис

Крок 1: A -> B: EKUb[N1||IDA]

- Сторона A генерує випадкове число N1 (nonce) і об'єднує його зі своїм ідентифікатором IDA.
- Це об'єднання шифрується публічним ключем B (KUb), що гарантує, що тільки B зможе його розшифрувати (завдяки приватному ключу KRb).

Крок 2: B -> A: EKUa[N1||N2]

- Сторона B розшифровує повідомлення, отримане від A, щоб отримати N1, і генерує своє N2.
- Потім B об'єднує N1 і N2 та шифрує це публічним ключем A (KUa), щоб A могла перевірити відповідність N1.

Крок 3: A -> B: EKUb[N2]

- A розшифровує попереднє повідомлення, отримує N2, і надсилає його назад B, зашифрувавши публічним ключем B.
- Це дозволяє B впевнитися, що A знає N2, забезпечуючи автентифікацію A.

Крок 4: A -> B: EKUb[EKRa[KS]]

- A генерує сесійний ключ KS і підписує його за допомогою свого приватного ключа KRA (створюючи цифровий підпис).
- Потім A шифрує підписаний ключ публічним ключем B (KUb) і надсилає його B.
- B може розшифрувати це повідомлення, перевірити підпис A, і отримати сесійний ключ KS.

Версія 2 - простіший опис

- Повідомлення надсилається від A до B. Повідомлення складається з оказії N1 та ID A і шифрується відкритим ключем B.
- Повідомлення надсилається від B до A. Містить оказії N1 і N2, шифрується публічним ключем A.
- Повідомлення від A до B. Передається оказія N2, зашифрована публічним ключем B.
- Від A до B. Передається таємний ключ Ks, зашифрований приватним ключем A, а потім і публічним ключем B.

Опишіть найпростішу схему цифрового підпису 👍.

M||EKRa[H(M)] $M||E_{KRa}[H(M)]$

Відправник:

1. Створює хеш повідомлення **H(M)** для перевірки цілісності.
2. Шифрує хеш своїм приватним ключем, створюючи цифровий підпис.
3. Надсилає підписане повідомлення **M** та підпис **EKRa[H(M)]**.

Отримувач:

1. Розшифровує підпис публічним ключем відправника.
2. Порівнює хеш повідомлення з розшифрованим підписом. Якщо співпадає, підтверджується автентичність і цілісність.

Які класи функції можуть служити для створення аутентифікатора повідомлень?

- **Криптографічні хеш-функції.** Хеш-функції використовуються разом з секретним ключем для створення MAC. Однією з популярних реалізацій є HMAC (Hash-based Message Authentication Code).
- **Симетричні шифри.** Функція MAC може бути побудована на основі симетричних алгоритмів шифрування, таких як DES або AES.
- **Асиметричні шифри.**
- **Коди перевірки на основі теорії чисел:** використовуються математичні операції, такі як модульні перетворення.
- **дописат**
-
- **Лінійні та нелінійні контрольні функції**

Опишіть наступну схему обміну даними

1. A→B: $E_{KUb}[N_1 \,||\, ID_A]$
2. B→A: $E_{KUa}[N_1 \,||\, N_2]$
3. A→B: $E_{KUb}[N_2]$
4. A→B: $E_{KUb}[E_{KRa}[K_s]]$

Це схема розподілу таємних ключів за допомогою системи з відкритим ключем

Версія 1 - складніший опис

Крок 1: A -> B: EKUb[N1||IDA]

- Сторона A генерує випадкове число N1 (nonce) і об'єднує його зі своїм ідентифікатором IDA.
- Це об'єднання шифрується публічним ключем B (KUb), що гарантує, що тільки B зможе його розшифрувати (завдяки приватному ключу KRb).

Крок 2: B -> A: EKUa[N1||N2]

- Сторона B розшифровує повідомлення, отримане від A, щоб отримати N1, і генерує своє N2.
- Потім B об'єднує N1 і N2 та шифрує це публічним ключем A (KUa), щоб A могла перевірити відповідність N1.

Крок 3: A -> B: EKUb[N2]

- А розшифровує попереднє повідомлення, отримує N2, і надсилає його назад В, зашифрувавши публічним ключем В.
- Це дозволяє В впевнитися, що А знає N2, забезпечуючи автентифікацію А.

Крок 4: A -> B: EKUb[EKRa[KS]]

- А генерує сесійний ключ KS і підписує його за допомогою свого приватного ключа KRA (створюючи цифровий підпис).
- Потім А шифрує підписаний ключ публічним ключем В (KUb) і надсилає його В.
- В може розшифрувати це повідомлення, перевірити підпис А, і отримати сесійний ключ KS.

Версія 2 - простіший опис
це схема розподілу таємних ключів за допомогою системи з відкритим ключем

Повідомлення надсилається від А до В. Воно містить випадкове число N1 та ідентифікатор IDA і шифрується відкритим ключем В.
Повідомлення надсилається від В до А. Воно містить випадкові числа N1 і N2 та шифрується публічним ключем А.
Повідомлення надсилається від А до В, передаючи випадкове число N2, зашифроване публічним ключем В.
Від А до В передається сеансовий ключ KS, зашифрований спочатку приватним ключем А, а потім публічним ключем В.

Постачальник локальних мереж забезпечує засоби розподілу ключів за наступною схемою. (опишіть деталі цієї схеми)

1. **A→ЦРК:** $Запит||N_1$
2. **ЦРК→A:** $E_{Ka}[K_S||Запит||N_1||E_{Kb}(K_S||ID_A)]$
3. **A→B:** $E_{Kb}[K_S||ID_A]$
4. **B→A:** $E_{KS}[N_2]$
5. **A→B:** $E_{KS}[f(N_2)]$

крок 1. A -> ЦРК: Запит N1

- Що відбувається:
  - Сторона А надсилає Центру розподілу ключів (ЦРК) запит на отримання сесійного ключа Ks для спілкування з В.
  - N1 - випадково згенероване число (nonce), яке забезпечує унікальність запиту та захист від повторних атак.
- Мета:
  - Ідентифікувати запит і уникнути повторення (Replay Attack).

Що відбувається, якщо зловмисник перехоплює повідомлення?

Якщо зловмисник перехоплює Ek(N1||N2) і повторно надсилає його:

Сторона А помітить, що N1 не є новим, і відкине запит.

Або сторона B виявить невідповідність f(N2).

Крок 2

Що відбувається:

- **ЦРК генерує новий сесійний ключ KS**, який буде використовуватися між A і B.
- **ЦРК шифрує дані двома способами:**
- **Для A (шифрування ключем KA):**
    - Включає KS, копію запиту N1 для валідації, і EKB(Ks||IDa), який буде передано B.
- **Для B (шифрування ключем KB):** KS|IDA.

Мета:

- Передати A необхідні дані для встановлення зв'язку з B, зокрема зашифроване повідомлення для B.

Крок 3

Що відбувається:

- **A надсилає B повідомлення**, яке було зашифроване ЦРК для B.
- **B отримує KS,** розшифровуючи його своїм ключем KB.

Мета:

- **Дати стороні B сесійний ключ KS**, за допомогою якого відбуватиметься подальший обмін даними.

Крок 4

Що відбувається:

- **B генерує своє випадкове число N2 (nonce)** і надсилає його A, зашифрувавши ключем KS.

Мета:

- **Перевірити, чи A дійсно володіє сесійним ключем KS.**
- **Забезпечити автентифікацію сторони A.**

Крок 5

Що відбувається:

- **A обчислює функцію f(N2)** (наприклад, хеш, операцію XOR або будь-яке інше узгоджене перетворення) та надсилає результат B, зашифрувавши його KS.

Мета:

- Завершити взаємну автентифікацію.

- Підтвердити, що А також володіє сесійним ключем KS.

Описати схему обміну даними

$$A \rightarrow B: \quad E_{K2}[M] \| C_{K1}[E_{K2}[M]]$$

EK2[M]:

- **Конфіденційність: Повідомлення M шифрується за допомогою публічного ключа K2 користувача B, що гарантує, що тільки B, який має відповідний приватний ключ K2, може розшифрувати повідомлення.**

CK1[EK2[M]]:

- **Аутентифікація та цілісність: Створюється MAC (Message Authentication Code) за допомогою секретного ключа K1 відправника A для вже зашифрованого повідомлення EK2[M]. Це гарантує, що повідомлення не було змінене під час передачі та підтверджує, що воно надійшло від A.**

Описати схему обміну даними 👍

$$A \rightarrow B: \quad E_K[M \| E_{KRa}[H(M)]]$$

У даній схемі відбувається таке: обчислюється хеш H(M) для даного повідомлення M. Даний хеш шифрується приватним ключем відправника A (Ekra). Повідомлення M разом із зашифрованим хешем повідомлення (Ekra [H(M)]) шифрується за допомогою методу традиційного шифрування (симетричного), за допомогою спільного для A та B ключа K.

Дана схема обміну даними забезпечує такі сервіси:

- Аутентифікація і цифровий підпис: Відправник A своїм приватним ключем підписує хеш повідомлення, що засвідчує, що це справді він.
- Цілісність: До повідомлення прикріпляється хеш повідомлення, що дозволить отримувачу B перевірити, чи не було внесено змін, або наявних помилок у переданому повідомленні.
- Конфіденційність: Дане повідомлення разом із його зашифрованим хешем шифрується за допомогою алгоритму симетричного шифрування, з використанням спільного ключа K. Інформація захищена, при умові, що лише A та B знають ключ K.

Описати схему обміну даними 👍

$$A \rightarrow B: \quad M \| E_{KRa}[H(M)]$$

У даній схемі передається відкритий текст повідомлення, а також зашифрований за допомогою приватного ключа K відправника A хеш даного повідомлення. Така схема буде забезпечувати

- цілісність, адже передається хеш, за яким можна перевірити, чи передане повідомлення не було модифіковане в процесі передачі. Хеш повідомлення у цій схемі шифрується за допомогою приватного ключа відправника A, що реалізує цифровий підпис.

- При цьому цифровий підпис забезпечує неможливість відмови, бо приватний ключ відправника A відомий лише йому, та аутентифікацію, адже за допомогою шифрування своїм приватним ключем відправник A підтверджує, що це справді він.

Описати схему обміну даними 👍

$$A \rightarrow B: \quad M \parallel E_K[H(M)]$$

У даній схемі передається відкритий текст повідомлення, а також зашифрований за допомогою ключа K хеш даного повідомлення. Така схема буде забезпечувати:

- **цілісність**, адже передається хеш, за яким можна перевірити, чи передане повідомлення не було модифіковане в процесі передачі;
- **аутентифікацію**, адже хеш повідомлення шифрується за допомогою секретного ключа K, який є відомий лише відправнику A та отримувачу B.
-

Описати схему обміну даними 👍

$$A \rightarrow B: \quad E_K[M \parallel H(M)]$$

- Конфіденційність: Повне повідомлення M і його хеш H(M) зашифровані ключем K, тому їх може прочитати лише отримувач, який знає ключ K.
- Цілісність: Отримувач може розшифрувати M, обчислити H(M) самостійно та порівняти з отриманим хешем для перевірки, чи не було змінено повідомлення.
- Аутентифікація: Якщо ключ K є спільним лише між A і B, отримання коректного хешу підтверджує, що повідомлення надійшло від A.

Описати схему обміну даними 👍

$$A->B: E_{KUb}[M]$$

конфіденційність - оскільки тільки отримувач (користувач B), який володіє приватним ключем, може розшифрувати повідомлення і отримати його оригінальний зміст.

Описати схему обміну даними 👍

$$A->B: E_{KUb}[E_{KRa}[M]]$$

цифровий підпис і конфіденційність

- Конфіденційність: оскільки повідомлення шифрується публічним ключем отримувача (EKUb), і тільки він може його розшифрувати.

- Цифровий підпис: оскільки повідомлення підписується приватним ключем відправника (EKRa), що підтверджує його автентичність.