НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

МЕТОДИ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕХАНІЗМІВ

Комп'ютерний практикум № 3

Реалізація основних асиметричних криптосистем.

Підгрупа 1С.

Виконали:
Волинець Сергій ФІ-42мн
Сковрон Роман ФІ-42мн
Радомир Беш ФІ-42мн

Київ — 2025

1 Мета

Дослідження можливостей побудови загальних та спеціальних криптографічних протоколів за допомогою асиметричних криптосистем.

2 Завдання на лабораторну роботу

Дослідити можливість реалізації сліпого цифрового підпису за допомогою різних асиметричних алгоритмів (не менше як двох) та порівняти їх ефективність за обраним критерієм.

3 Сліпий цифровий підпис

Цифровий підпис – це криптографічний механізм, який використовується для підтвердження автентичності та цілісності електронних документів або повідомлень. Він захищає дані, які передаються або зберігаються, від зміни без дозволу, і дозволяє перевірити особу або організацію, що підписала цей документ.

Сліпий цифровий підпис – це форма цифрового підпису, яка дозволяє підписувати повідомлення або документ таким чином, що сам підписувач не може бачити вміст підписаного документа після того, як він його підписав. Тобто, підписувач не знає зміст інформації, яку він підтверджує, що забезпечує додатковий рівень конфіденційності та безпеки.

Основна ідея сліпих підписів полягає в наступному:

1. Аліса шифрує документ і надсилає його Бобу.
2. Боб, не бачачи вміст документа, підписує його і повертає назад Алісі.
3. Аліса знімає свій шифр, залишаючи на документі тільки підпис Боба.

Математично, це можна зобразити наступним чином:

1. Аліса зашифровує повідомлення m функцією f, отримуючи шифротекст $c = f(m)$.
2. Аліса надсилає шифротекст c Бобу.
3. Боб наосліп (так як не знає, що знаходиться всередині) підписує повідомлення c функцією g, отримуючи $c' = g(c) = g(f(m))$.
4. Боб надсилає c' назад Алісі.
5. Аліса отримує c' і прибирає шифрування, отримуючи: $c'' = g(f(m)) * f^{-1} = g(m)$.

По завершенні цього протоколу Боб нічого не знає ні про повідомлення m, ні про підпис під цим повідомленням. Це і є матою сліпого підпису, тобто перешкодити Бобу пов'язати підпис повідомлення з Алісою.

Схема безпечного сліпого підпису повинна задовольняти наступним властивостям:

- Нульове розголошування. Ця властивість допомагає користувачеві отримати підпис на даному повідомленні, не розкриваючи самого повідомлення підписуючій стороні.
- Невідстежуваність. Підписуюча сторона не може відстежити пару підпис-повідомлення після того, як користувач оприлюднив підпис на повідомленні.

- Непідкладність. Тільки підписуча сторона може сгенерувати дійсний підпис. Ця властивість найважливіша і повинна задовольнятися для всіх схем підписів.

3.1 Цифровий підпис RSA

RSA (Rivest–Shamir–Adleman) – це один з найпоширеніших алгоритмів асиметричного шифрування, який використовується для забезпечення конфіденційності та автентичності в цифрових підписах та електронних повідомленнях. Сама система може бути застосовною і для простого шифрування даних. Основа криптосистеми RSA це функція $\mathrm{RSA}_{n,e} : \mathbb{Z}_n \to \mathbb{Z}_n$,

$$\mathrm{RSA}_{n,e}(x) \equiv x^e \bmod n.$$

Алгоритм був розроблений в 1977 році Рівестом, Шаміром та Адлеманом і ґрунтується на складності задачі факторизації, тобто на складності розкладу великих чисел на прості множники. У системі RSA існують два ключі: публічний, який можна передавати іншим особам для шифрування даних або перевірки підписів, і приватний, який використовується для дешифрування або створення цифрових підписів.

Генерування ключів відбувається наступним чином:

1. Аліса обирає два великих простих числа p, q $(p \neq q)$ та обчислює значення

$$n = pq, \quad \varphi(n) = (p-1)(q-1).$$

2. Також Аліса обирає випадкове число e, яке є взаємнопростим із $\varphi(n)$, та обчислює обернене до e число d:

$$e \in_R \mathbb{Z}^*_{\varphi(n)}, \quad d = e^{-1} \bmod \varphi(n),$$

   іншими словами, повинно виконуватись співвідношення

$$ed \equiv 1 (\mathrm{mod} \varphi(n)).$$

3. Пара (n, e) є відкритим ключем Аліси, а трійка (d, p, q) — її секретним ключем.

Підпис відбувається за наступним алгоритмом:

1. Аліса хоче підписати повідомлення $m \in \mathbb{Z}_n$. Для цього Аліса за допомогою свого секретного ключа d обчислює значення підпису

$$s = m^d \bmod n.$$

2. Число s є підписом під повідомленням m і часто об'єднується із ним у підписане повідомлення (m, s).

Перевірка підпису (m, s)**, виконується перевіркою рівності:**

$$s^e \equiv m (\mathrm{mod} n).$$

3.2 Цифровий підпис Шнорра

Підпис Шнорра – це криптографічна схема підпису, яка базується на проблемі дискретного логарифмування. Вона є ефективною та безпечною, часто використовується для створення цифрових підписів в криптографічних системах. Для підпису, генерується випадкове значення, яке комбінується з хешем повідомлення для створення підпису. Під час перевірки підпису використовуються публічний ключ підписувача та інформація, яку він надав. Підпис Шнорра забезпечує високу стійкість до атак, таких як підробка підпису, і є ефективним у системах, де важлива швидкість і безпека підписування.

При генерації ключів, утворюється секретний ключ sk та публічний ключ $pk = g^{sk}$, де g – це генератор.

Підпис відбувається за наступним алгоритмом:

1. Аліса, яка хоче підписати повідомлення, вибирає випадкове значення r.
2. Аліса обчислює значення $R = g^r$, $e = \mathrm{H}(m||R||PK)$, де H – це криптографічна геш функція.
3. Аліса обчислює значення $s = r + e \cdot sk$.
4. Пдписом повідомлення m буде пара (R, s)

Перевірка підпису (R, s)**, виконується перевіркою рівності:**

$$g^s \equiv R + pk^e.$$

3.3 Сліпий цифровий підпис RSA

Сліпий цифровий підпис RSA схожий до стандартної схеми RSA.

Сліпий підпис відбувається наступним чином:

1. Аліса вибирає випадковий маскуючий множник r, взаємно простий з p, і обчислює $m' \equiv mr^e \bmod p$.
2. Аліса відсилає m' по відкритому каналу Бобу.
3. Боб обчислює $s' \equiv (m')^d \bmod p$, використовуючи свій закритий ключ (p, d).
4. Боб відсилає s' назад Алісі.
5. Аліса прибирає своє початкове маскування і отримує підписане Бобом вихідне повідомлення m наступним чином:

$$s \equiv s'r^{-1} \bmod p \equiv m^d \bmod p.$$

3.4 Сліпий підпис на основі підпису Шнорра

Сліпий підпис на основі підпису Шнорра, на відмінну від підпису RSA, підписує геш значення повідомлення а не саме повідомлення. Це є перевагою, адже такий підхід захищає підпис від деяких атак, та зменшує загальну довжину підпису.

Підписати повідомлення можна за наступним алгоритмом:

1. Боб надсилає Алісі значення $R = a^k \mod p$.
2. Аліса обчислює $R^{'} = Ra^{-w}y^{-t} \mod y$ (де w і t – випадкові числа, що не перевищують y), $E^{'} = H(m||R^{'})$ і $E = E^{'} + t \mod y$, після чого відправляє Бобу значення E.
3. Боб обчислює значення S, таке що $R = a^S y^E \mod p$, і відправляє S Алісі.

4. Аліса обчислює підпис (E', S'), де $E' = E^{-t} \mod y$ і $S' = S - w \mod y$, яка є справжньою по відношенню до повідомлення m.

3.5 Вразливості сліпого цифрового підпису RSA

Алгоритм RSA може бути вразливим до атаки, яка дозволяє розшифрувати попередньо підписане наосліп повідомлення, видавши його за повідомлення, що ще має бути підписане. Оскільки процес підпису є аналогічним розшифруванню повідомлення з використанням секретного ключа підписувача, атакуючий може підмінити повідомлення, яке вже було підписано наосліп, на зашифровану версію цього повідомлення за допомогою відкритого ключа підписувача і подати його для підпису.

$$\begin{aligned} m'' &= m'r^e \pmod n \\ &= (m^e \pmod n \cdot r^e) \pmod n \\ &= (mr)^e \pmod n \end{aligned}$$

де m' – це зашифрована версія повідомлення. Коли повідомлення підписане, відкритий текст m легко отримуємо як:

$$\begin{aligned} s' &= m''^d \pmod n \\ &= ((mr)^e \pmod n)^d \pmod n \\ &= (mr)^{ed} \pmod n \\ &= m \cdot r \pmod n, \text{ since } ed \equiv 1 \pmod{\varphi(n)} \end{aligned}$$

де $\varphi(n)$ – це Функція Ейлера. Тепер повідомлення легко отримати.

$$m = s' \cdot r^{-1} \pmod n$$

Атака здійснюється через те, що в цій схемі підписувач підписує саме повідомлення. У стандартних схемах підпису, підписується зазвичай криптографічний хеш повідомлення. Як висновок, можна сказати, що через мультиплікативну властивість RSA, один і той самий ключ не повинен використовуватися одночасно для шифрування та підписання наосліп.

Варто зауважити, що в дійсності ніхто не використовує систему RSA такм чином. В більшості стандартах та специфікаціях прописано, що система RSA повинна використовуватися для підпису хешів повідомлень, а не самих повідомлень.

4 Реалізація

У екосистемі мови програмування Java існує підтримка багатьох різних криптографічних примітивів таких як генераторів псевдовипадкових чисел, перевірки на простоту, хеш функцій, як симетричних та асиметричних алгоритмів та функцій. В тому числі є майже готові практичні рішення такі як криптографічні примітиви RSA, EDDSA, тощо. Тому в якості демонстрації було імплементовано RSA, EDDSA та підписів Шнора для модульної арифметики та арифметики на еліптичних кривих вбудованими рішеннями.

Однак як виявилось, схеми сліпого цифрового підпису не має серед готових реалізації, тож було прийнято рішення до їх імплементації. Реалізовано схему сліпого цифрового підпису RSA. Основні проблеми в даній демонстрації було у розділення зон відповідальності та доступу між двома абонентами та коректної генерації параметрів та проведення коректних модульних операції.

5 Висновки

Асиметричні криптосистеми, зокрема RSA та система Шнорра, широко застосовуються для створення цифрових підписів, що забезпечують як автентичність, так і конфіденційність повідомлень. У цих системах використовується пара ключів — приватний і відкритий, що дозволяє створювати підписи та перевіряти їх без необхідності передачі приватного ключа. Підпис гарантує, що повідомлення надійшло від конкретного відправника і не було змінене в процесі передачі. Хешування дає можливість зменшити ймовірність маніпуляцій з повідомленням, оскільки будь-яка зміна в самому тексті повідомлення змінює і його хеш, що робить підпис недійсним. В RSA підписується саме повідомлення або його хеш, залежно від специфікації (хоча в реальних системах перший підхід не використовують).