ЛАБОРАТОРНА РОБОТА № 4

Алгоритм шифрування RSA

Мета роботи: ознайомитися з програмною реалізацією алгоритму асимметричного шифрування RSA. Розробити програму його використання.

Використовуване програмне забезпечення: середа розробки GNU Octave, Python.

1. Теоретичні відомості

Ілюстрація роботи RSA на прикладі

Навколо алгоритмів шифрування з відкритим та закритим ключем існує безліч непорозумінь та містифікацій. Тут я хотів би максимально коротко і наочно, з конкретними числами та мінімумом формул, показати, як це працює.

Я не вдаюся в теорію (не дуже зрозуміло, на який рівень підготовки читача слід розраховувати), але я впевнений, що прочитавши цю коротку ілюстрацію, будь-якій людині буде простіше розібратися у формулах та суворих доказах.

Отже. Допустимо, я хочу отримати від вас деякі дані. Ми з вами не хочемо, щоб ці дані дізнався хтось, крім нас. І у нас немає жодної впевненості у надійності каналу передачі даних. Приступимо.

Крок перший. Підготовка ключів

Я повинен зробити попередні дії: згенерувати публічний та приватний ключ.

• Вибираю два простих числа. Нехай це буде p=3 та q=7.

• Обчислюємо модуль - добуток наших p і q: n=p×q=3×7=21.

• Обчислюємо функцію Ейлера: φ=(p-1)×(q-1)=2×6=12.

• Вибираємо число e, що відповідає наступним критеріям: воно має бути просте, воно має бути меншим за φ — залишаються варіанти: 3, 5, 7, 11, воно має бути взаємно просте з φ; залишаються варіанти 5, 7, 11. Виберемо e = 5. Це так звана відкрита експонента.

Тепер пара чисел {e, n} – це мій відкритий ключ. Я надсилаю його вам, щоб ви зашифрували своє повідомлення. Але для мене це ще не все. Я мушу отримати закритий ключ.

Мені потрібно обчислити число d, обернене по модулю φ. Тобто залишок від розподілу за модулем φ твору d×e має дорівнювати 1. Запишемо це в позначеннях, прийнятих у багатьох мовах програмування: (d×e)%φ=1. Або (d×5)%12=1.d може бути рівним 5 ((5×5)%12=25%12=1), але щоб воно не плуталося з e в подальшому оповіданні, давайте візьмемо його рівним 17. Можете перевірити самі, що (17×5)%12 дійсно одно 1 (17×5-12×7=1). Отже, d=17. Пара - це секретний ключ, його я залишаю у себе. Його не можна повідомляти нікому. Тільки власник секретного ключа може розшифрувати те, що було зашифровано відкритим ключем.

Крок другий. Шифрування

Тепер настала ваша черга шифрувати ваше повідомлення. Припустимо, повідомлення це число 19. Позначимо його P=19. Крім нього, у вас вже є мій відкритий ключ: {e, n} = {5, 21}. Шифрування виконується за наступним алгоритмом:

• Підключайте ваше повідомлення до ступеня e за модулем n. Тобто, обчислюєте 19 ступенем 5 (2476099) і берете залишок від розподілу на 21. Виходить 10 - це ваші закодовані дані.

Строго кажучи, вам зовсім нема чого обчислювати величезне число «19 ступенем 5». При кожному множенні досить обчислювати не повне твір, лише залишок від розподілу на 21. Але це вже деталі реалізації обчислень, давайте у них заглиблюватися.

Отримані дані E=10 ви надсилаєте мені.

Тут слід зазначити, що повідомлення P=19 має бути більше n=21. інакше нічого не вийде.

Крок третій. Розшифровка

Я отримав ваші дані (E=10), і я маю закритий ключ {d, n} = {17, 21}.

Зверніть увагу, що відкритий ключ не може розшифрувати повідомлення. А закритий ключ я нікому не говорив. У цьому вся краса асиметричного шифрування.

Починаємо розкодувати:

• Я роблю операцію дуже схожу на вашу, але замість e використовую d.Зводжу E в ступінь d: отримую 10 в ступінь 17 (дозвольте, я не писатиму одиниця з сімнадцятьма нулями). Обчислюю залишок від розподілу на 21 і отримую 19 — ваше повідомлення.

Зверніть увагу, ніхто, крім мене (навіть ви!) не може розшифрувати ваше повідомлення (E=10), тому що ні в кого немає закритого ключа.

У чому гарантія надійності шифрування

Надійність шифрування забезпечується тим, що третій особі (намагається зламати шифр) дуже важко обчислити закритий ключ по відкритому. Обидва ключі обчислюються з однієї пари простих чисел (p та q). Тобто ключі пов'язані між собою. Але встановити цей зв'язок дуже складно. Основною проблемою є декомпозиція модуля n на прості співмножники p і q. Якщо число є твором двох дуже великих простих чисел, його дуже важко розкласти на множники.

Намагаюся це показати на прикладі. Давайте розкладемо на множники число 360:

• одразу ясно. що воно ділиться на два (отримали 2)

• 180, що залишилося теж, очевидно парне (ще 2)

• 90 - теж парне (ще двійка)

• 45 не ділиться на 2, але наступна спроба виявляється успішною — воно ділиться на три (отримали 3)

• 15 теж поділяється на 3

• 5 – просте.

Ми на кожному кроці, практично без перебору, отримували нові й нові множники, легко отримавши повне розкладання 360=2×2×2×3×3×5

Давайте тепер візьмемо число 361.

2. Завдання на лабораторну роботу.
    1. Створити програмний продукт для реалізації даного алгоритму шифрування.
    2. Створити відкритий і закритий ключі.

3. Перевірити правильність роботи програми на прикладі передачі числа n^3, де n – порядковий номер у журналі.