Лабораторна робота 2.1

АЛГОРИТМ RSA

Мета: отримати навички у створенні програмної реалізації алгоритму асиметричного алгоритму RSA.

Основні завдання:

1. Розробити об’єктно-орієнтованою мовою програмування консольний або віконний додаток, що реалізує шифрування та дешифрування вмісту текстового або виконуваного файлу з використанням асиметричного алгоритму RSA. Програма повинна запитувати ім'я вхідного і вихідного файлів, тип шифру та вхідні параметри для проведення криптографічних перетворень. В якості вхідних параметрів виступають: e – відкритий ключ шифрування та n – добуток двох взаємнопростих чисел p та q. З використанням розширеного алгоритму Евкліда розрахувати значення d – закритий ключ дешифрування.
2. Представити блок структурну схему своєї програмної реалізації.
3. Зробити висновки на підставі проведених теоретичних та практичних досліджень. У висновках слід зазначити, які навички та знання отримано під час виконання завдань.

Основні теоретичні відомості

RSA - перший повноцінний алгоритм з відкритим ключем, який можна використовувати і для шифрування, і для створення цифрових підписів, алгоритм RSA. Стійкість RSA базується на складності факторизації великих цілих чисел. Відкритий і закритий ключі є функціями двох великих простих чисел розрядністю 100…200 десяткових цифр і навіть більше. Відновлення відкритого тексту за шифртекстом та відкритим ключем є рівнозначне до розкладання числа на два великі прості множники. 1993 року алгоритм RSA було ухвалено за стандарт PKCS # 1: RSA Encryption Standard.

Для шифрування використовується проста операція піднесення до степеня за модулем n. Для дешифрування ж необхідно обчислити функцію Ейлера від числа n, для цього необхідно знати розкладання числа n на прості множники (в цьому і полягає завдання факторизації).

Розглянемо основні етапи алгоритму RSA:

1. У довільний спосіб обираються два великі взаємно прості

числа p p та q q. Обчислюється добуток $n = p \cdot q$ $n = p \cdot q$.

2. Обчислюється функція Ейлера: $\varphi(n) = (p-1)\cdot(q-1)$ $\varphi(n) = (p-1)\cdot(q-1)$.

3. Довільно обирається просте число e e – ключ зашифровування, яке задовольняє умовам $e < \varphi(n)$ $e < \varphi(n)$; $НСД[e, \varphi(n)] = 1$ $НСД\left(e, \varphi(n)\right) = 1$.

4. Обчислюється число d d – ключ розшифровування, яке є оберненим до числа e e, тобто $e * d \equiv 1 (mod\, \phi\,(n))$ $e * d \equiv 1 (mod\, \phi\,(n))$ $ed \equiv 1 (mod\, \varphi(n))$.

5. Пара чисел (e, n) (e, n) виступає в якості відкритого ключа і передається по відкритих незахищених каналах зв'язку, а числа (p, q) (p, q) – знищують для того щою унеможливити компрометацію закритого ключа, d – закритий ключ дешифрування, який зберігається в секреті.

При шифруванні повідомлення M M спочатку розкладаємо на цифрові блоки, чиї розміри є менше за n n, тобто якщо p p та q q є 100-розрядними простими числами, то n n міститиме близько 200 розрядів і кожен блок повідомлення m_i m_i повинен мати близько 200 розрядів у довжину. Зашифроване повідомлення C C складатиметься з блоків C_i C_i такої самої довжини. Формула шифрування є досить простою:

$$C \equiv M^e \pmod{n}$$.

Дешифрування забезпечується операцією піднесення до степеня d d за модулем n d одержаного шифротексту C C:

$$M \equiv C^e \pmod{n}$$. $M \equiv C^d (mod\, n)$

Система RSA використовується для захисту програмного забезпечення та в схемах цифрового підпису. Крім того, алгоритм використовується у великій кількості криптографічних додатків і прикладних протоколів: SSL; IPSec; IKE; SILC; SSH.

В апаратному виконанні RSA алгоритм застосовується в захищених телефонах, на мережевих платах Ethernet, на смарт-картах, широко використовується в криптографическом обладнанні Zaxus Datacryptor.

Також вона використовується у відкритій системі шифрування PGP та інших системах шифрування в поєднанні з симетричними алгоритмами. Через низьку швидкість шифрування (близько 30 кбіт/с при 512 бітному ключі на процесорі 2 ГГц), повідомлення зазвичай шифрують за допомогою більш продуктивних симетричних алгоритмів з випадковим сеансовим ключем (наприклад, AES, IDEA, Serpent, Twofish), а за допомогою RSA шифрують лише цей ключ, таким чином реалізується гібридна криптосистема. Такий механізм має потенційні уразливості зважаючи на необхідність використовувати криптографічно стійкий генератор псевдовипадкових чисел для формування випадкового сеансового ключа симетричного шифрування.

Запитання для самоперевірки

1. Що таке асиметричне шифрування?
2. Поясність основні етапи знаходження закритого ключа шифрування?
3. Поясніть основні умови вибору відкритого ключа.
4. Охарактеризуйте основні напрями використання алгоритму RSA.
5. Назвіть основні переваги та недоліки алгоритму RSA.