Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

Лабораторна робота №2

Реалізація алгоритмів генерації ключів гібридних криптосистем

Виконали: студенти гр. ФІ-32мн,
Костюк К.М., Панасюк Є.С, Пелешенко Л.І.

Перевірив:
Кудін А.М

Київ – 2023

Мета лабораторної роботи:
Дослідження алгоритмів генерації псевдовипадкових послідовностей, тестування простоти чисел та генерації простих чисел з точки зору їх ефективності за часом та можливості використання для гененерації ключів асиметричних криптосистем.

Постановка задачі:
Розробити бібліотеку генерації псевдовипадкової послідовності, тестування простоти чисел та генерації простих чисел для Intel-сумісних ПЕОМ. Розмірність чисел – 768, 1024 біт.

Підгрупа 1А. Запропонувати схему генератора ПВЧ для інтелектуальної картки, токена та смартфону. Розглянути особливості побудови генератора простих чисел в умовах обмеження пам'яті та часу генерації.

Генерація псевдовипадкових послідовностей:

Для генерація псевдовипадкових послідовнистей нами було вибрано алгоритм Блюм-Блюма-Шуба(BBS),а саме його байтова модифікація, сам BBS генерує послідовність таким чином:

9) Генератор BBS (Блюм-Блюма-Шуба) побудовано на ідеях Блюма та Мікалі, однак для генерування псевдовипадкових бітів він використовує апарат теорії чисел. Доведено, що можливість вгадувати біти вихідної послідовності цього генератору еквівалентна можливості розв'язувати задачу факторизації.

Нехай p та q – різні великі прості числа виду $4k+3$ і $n=pq$. Початкове значення $r_0 \geq 2$ обирається довільним чином. Вихідна послідовність $x_1, x_2, \ldots$ обчислюється за таким правилом:

$$r_i = r_{i-1}^2 \bmod n,$$
$$x_i = r_i \bmod 2,$$

тобто x_i є останнім бітом числа r_i (зверніть увагу, що вихідні біти нумеруються з одиниці – стан r_0 не використовується для безпосереднього генерування вихідної послідовності).

Байтова модифікація генератору BBS обчислює вихідну послідовність як $x_i = r_i \bmod 256$, тобто повертаються вісім молодших біт числа r_i.

Також цей генератор проходить тести на рівномірність знаків, незалежність та однорідність, а це означає, що генерується дійсно псевдовипадкова послідовність.

Тестування простих чисел:

Було програмно реалізовано усі 3 імовірнісні тест перевірки на простоту. А саме: Соловея-Штрассена, Ферма і Міллера-Рабіна.

Тест Ферма:

У результаті число p буде перевірено за тестом Ферма при k різних основах . Якщо p виявиться не псевдопростим хоча б за однією основою, то p складене. Якщо p псевдопросте за всіма цими основами, то вважаємо p простим числом.

Існує клас складених чисел, що проходять тест Ферма із імовірністю 1 – так звані числа Кармайкла. Не дивлячись на те, що імовірність натрапити на число Кармайкла при випадковій генерації майже дорівнює нулю, така теоретична загроза вважається достатньою для того, щоб не використовувати тест Ферма у практичних застосуваннях.

Тест Соловея-Штрассена:

Тест Соловея-Штрассена є одностороннім, тобто в ньому відсутні помилки першого роду: якщо p – просте число, то тест завжди поверне, що воно просте. Однак якщо p – число складене, то для кожної можливої основи тест із імовірністю 1⁄2 поверне помилкову відповідь (повідомлення, що p – просте). Використання k різних випадкових основ зменшує помилку тесту до 2^{-k}. Також слід зауважити, що цей тест є доволі простим та невибагливим з обчислювальної точки зору: кожен запуск вимагає лише одного піднесення у степінь за модулем та обчислення символу Якобі, складність якого невелика.

Тест Міллера-Рабіна:

Тест Міллера-Рабіна є одностороннім, тобто в ньому відсутні помилки першого роду: якщо p – просте число, то тест завжди поверне, що воно просте. Однак якщо p – число складене, то для кожної можливої основи тест із імовірністю 1⁄4 поверне помилкову відповідь (повідомлення, що p – просте). Використання k різних випадкових основ зменшує помилку тесту до 4^{-k} .Оскільки у тесті Соловея-Штрассена відповідна помилка складає 2^{-k}, він в ході жорсткої еволюційної боротьби майже всюди витіснений тестом Міллера-Рабіна.

Тест Міллера-Рабіна дозволяє будувати детерміновані тести для перевірки простоти натуральних чисел, якщо вони не перевищують деякої межі. Наприклад, наступний тест перевіряє простоту числа $p < 3 \cdot 10^9$:

Крок 1. Перевіряємо, чи є p сильним псевдопростим числом за основою 2. Якщо ні, то p складене й алгоритм зупиняється.

Крок 2. Те ж за основою 3.

Крок 3. Те ж за основою 5.

Крок 4. Те ж за основою 7.

Крок 5. Якщо не було виявлено, що p складене, то p є простим числом.

Генерація простих чисел:

Програмно було реалізовано алгоритм генерації простих чисел виду 4k+3, яких існує нескінченно багато, в цілому можна генерувати будь-які прості числа

Висновки:

В наш час існує неймовірна кількість девайсів, таких як інтелектуальні картки, токени та смартфони. Для генерації ж ПВЧ для них ми пропонуємо використовувати ARX-примітиви, наприклад генератор Джиффі.

регістр ***L*11**: $x_{11} = x_0 \oplus x_2$;

регістр ***L*9**: $y_9 = y_0 \oplus y_1 \oplus y_3 \oplus y_4$;

регістр ***L*10**: $s_{10} = s_0 \oplus s_3$.

Вихідна послідовність бітів (z_i) обчислюється за правилом $z_i = s_i x_i \oplus (1 \oplus s_i) y_i$.