НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Комп'ютерний практикум №2 з курсу
Методи Реалізації Криптографічних Механізмів

Реалізація алгоритмів генерації ключів гібридних криптосистем

Виконав студент
групи ФІ-22мн
Кушнір О.С.

Перевірив:
асистент
Селюх П.В.

Київ — 2022

1 МЕТА КОМП'ЮТЕРНОГО ПРАКТИКУМУ

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.

2 ПОСТАНОВКА ЗАДАЧІ

Дослідити функції генерації пвп, тестування чисел на простоту та генерації простих чисел стандартної бібліотеки PGP v.6.5.2 і розробити приклад роботи з нею. А саме оцінити ймовірність помилки під час гененерації простого числа.

3 ХІД РОБОТИ

Було проінстальовано PGP CLI версії 6.5.2, бібліотека була проаналізована за допомогою документації. Згідно до документації бібліотека реалізує стандарт OpenPGP, що не накладає жодних обмежень для генерації псевдовипадкових чисел для генерації ключів. Статистичні методи для виявленя ймовірності помилки будуть неефективними, тому що перевіряти псевдовипадкові числа ми маємо ймовірносними алгоритмами, а враховуючи те, що число після генерації вже перевіряється якимось набором з цих алгоритмів то не знаючи їх чіткого набору ми не зможемо з'ясувати що є помилкою другого роду, ймовірність якої нам потрібно розрахувати. Тому враховуючи те що нам в будьякому випадку потрібно знати якими критеріями число перевірялось на простоту, то найкраще для підрахунку буде аналітичний метод. Тобто знаючи ймовірність помилки кожного ймовірносного алгоритму потрібно знайти ймовірність того, що всі вони помилилися. Нехай $p_1, ..., p_n$ - ймовірність помилки кожного з n алгоритмів, якими перевіряється простота, тоді ймовірність того що жоден з них не помиляється $\prod_{i=1}^{n}(1 - p_i)$, а ймовірність того, що хоча б один помилився $1 - \prod_{i=1}^{n}(1 - p_i)$. Це актуально лише в тому випадку коли для того щоб отримати псевдопросте алгоритм має отримати позитивний результат на всіх тестах, в інших випадках розрахунки будуть іншими. Оскільки ми не маємо доступу до вихідного коду, а документація нічого нам не каже про алгоритмами отримати точне число ми не зможемо.

ВИСНОВКИ

Під час виконання лабораторної работи було досліджено стандартну бібліотеку PGP і стандарт OpenPGP. Як видно з данних отриманих зі стандарту і документації, алгоритми генерації псевдовипадкових простих чисел є вадою всіх алгоритмів, що їх потребують, бо ми ніколи не можемо гарантувати, що число буде саме просте, за вийнятком чисел, що менше ніж 2^{64}, бо ми знаємо всі прості що менше і сучасні алгоритми це враховують, але такі числа замалі в якості ключа для багатьох алгоритмів, зокрема для RSA, і роблять його вразливим до багатьох атак.