Лабораторна робота №14

Розробка та дослідження генератора випадкових чисел за допомогою пристроїв персонального комп’ютера.

Мета: навчитися розробляти генератори випадкових чисел за допомогою фізичних пристроїв персонального комп’ютера.
Обладнання: персональний комп’ютер.
Програмне забезпечення: будь-яка операційна система, будь-яка мова програмування, криптобібліотека, де релізовано основні криптографічні функції хешування та генератори псевдовипадкових чисел.

Література

1. Кнут Д.Э. Искусство программирования. Том 2. Получисленные алгоритмы. — 2011. — С. 12—14. — 832 с. — ISBN 978-5-8459-0081-4.
2. *Henk C. A. va Tilborg* Encyclopedia of Cryptography and Security. — USA : Springer Science+Business Media, 2005. — С. 509-514. — ISBN 978-0-387-23473-1.
3. Апаратний генератор випадкових чисел. Вікіпедія. - Електронний ресурс. - Режим доступу: https://uk.wikipedia.org/wiki/%D0%90%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%B8%D0%B9_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%B2%D0%B8%D0%BF%D0%B0%D0%B4%D0%BA%D0%BE%D0%B2%D0%B8%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB

Теоретична частина

Однією з найважливіших задач криптографії є генерування істинно випадкових послідовностей. Це важливо з багатьох точок зору, хоча основною є генерування криптографічно стійких ключів та векторів ініціалізації для різних алгоритмів шифрування. Не менш важливим вважається також утворення випадкового ключового потоку для потокового шифру, який в такому разі міг би вважатися ідеальним (при правильному використанні, звісно).
Основною відмінністю випадкової послідовності від псевдовипадкової є те, що випадкову послідовність неможливо повторити навіть при однакових вхідних умовах, оскільки фізичний датчик завжди непередбачуваний. У таких датчиках можуть використовуватися різні фізичні процеси, наприклад, радіоактивний розпад, білий шум, дробовий шум в електричних колах тощо.
Не зважаючи на отримання від таких датчиків істинно випадкових послідовностей, вони не позбавлені деяких недоліків. Це, зокрема, низька швидкість генерування, необхідність передавання послідовності на приймальну сторону, оскільки повторити таку послідовність практично неможливо, зміна характеристик з часом внаслідок “старіння” фізичного датчика, зміщення математичного сподівання послідовності бітів на виході, висока вартість. Саме тому широке розповсюдження отримали криптографічно стійкі генератори псевдовипадкових послідовностей, застосування яких розглядаються у ЛР № 13.
Тим не менше існують можливості розробки досить якісного апаратного генератора випадкових послідовностей з використанням пристроїв персонального комп’ютера. Цьому й присвячена дана лабораторна робота.
Розглянемо, які пристрої комп’ютера можна використати для генерування випадкових послідовностей.
Позиціонування мишки. Для генерування випадкової послідовності використовується хаотичний рух мишки у виділеному для цього спеціальному вікні. Координати курсора зчитуються програмним забезпеченням, виділяються найменш значущі біти від двійкового представлення координат, виконується їх постобробка, та результат подається для шифрування. Можна ускладнити завдання,вимагаючи користувача не тільки хаотично рухати

мишкою, а й так само хаотично натискати кнопки. Зчитуючи і координати, й час між натисканнями, можна отримати випадкову послідовність (звичайно, після постобробки).

Використання клавіатури комп'ютера. Іншим пристроєм, який можна використати для генерування, є клавіатура. Програмне забезпечення просить користувача хаотично натискати на клавіші клавіатури. Вимірюючи проміжки часу між послідовними натисканнями на клавіші, перетворюючи їх у двійкове представлення, програмне забезпечення виділяє найменш значущі біти, виконує постобробку й подає результат для шифрування інформації.

Перелічені методи мають один спільний недолік: вони залучають до генерування користувача, а значить, процес буде досить повільним. Крім того, без користувача цей процес взагалі неможливий, що також незручно.

Використання температурних датчиків. Комп'ютер має велику кількість датчиків, які слідкують за температурою процесора, відеокарти, материнської плати, жорсткого диску та інших комплектуючих. Як і кожен фізичний датчик, ці детектори мають флуктуації вимірювань температури, тобто, кожне виміряне значення відрізняється на якісь долі градуса незалежно від того, наскільки стабільна реальна температура. Використання цих температурних флуктуацій дозволяє розробити генератор випадкових послідовностей, якщо знімати показники температури через короткі проміжки часу, виділяти найменш значущі біти (які найбільше змінюються) та обов'язково піддавати їх постобробці.

Використання сигналу з мікрофону. Можна також використати в якості датчика мікрофон ноутбука, якщо його просто включити "на прослуховування" оточуючого середовища. Випадковий шум, який буде сприймати мікрофон, також можна обробити та з його допомогою згенерувати випадкову послідовність. В цьому випадку, правда, є деякі нюанси, бо мікрофон ноутбука буде сприймати гуркіт вентиляторів та періодичний фон блоку живлення, який має період в 50 Гц, що погіршить якість послідовності. Тому, по-перше, обов'язково треба виділяти найменш значущу частину значень, а по-друге, виконувати постобробку отриманих значень.

Використання номерів блоків жорсткого диску. Одним зі спірних методів генерування є використання молодших бітів номерів секторів/блоків жорсткого диску комп'ютера при хаотичному читанні. Обробляти цю інформацію треба так само, як і у попередніх випадках.

Можна використати інші датчики, які присутні в комп'ютері.

Постобробка. Отримані від фізичного детектора дані необхідно обробляти. Для цього існують багато способів. Розглянемо найпростіші з них.

1. Використання криптографічних хеш-функцій. Потік даних з фізичного датчика подається на вхід функції хешування, яка й виконує постобробку. Результат хешування, хеш-образ, подається вже на вихід для шифрування відкритого повідомлення.
2. Використання програмного генератора псевдовипадкової послідовності. Потік даних з фізичного датчика додається за модулем два (XOR) з послідовністю, яку генерує програмний генератор, наприклад, BBS, Blum-Micali тощо. Результат використовується для шифрування відкритого повідомлення.

Практична частина

Для виконання ЛР необхідно зробити таке.

1. Студент обирає один зі способів отримання випадкової послідовності. Можна також розробити комбінований спосіб, в якому використовуються одночасно кілька фізичних датчиків комп'ютера.
2. Студент розробляє програмне забезпечення, яке отримує потік даних з обраного датчика/датчиків.
3. Студент обирає спосіб постобробки отриманих даних та реалізує його. Для цієї мети можна також використати розроблений в ЛР№13 генератор ключового потоку на клітинних автоматах.

4. За допомогою реалізованого генератора випадкової послідовності розробляється потоковий шифр.
5. На вхід потокового шифру подається файл розміром 12.5 МБ. Зашифрований файл подається на вхід пакету статистичного тестування NIST STS для дослідження статистичних характеристик розробленого шифру.
6. Результати тестування порівнюються з результатами ЛР №12 та №13.
7. Формується звіт з лабораторної роботи, який повинен містити:
- протокол Ваших дій;
- архітектуру розробленого програмного забезпечення та принципи постобробки даних;
- результати тестування статистичних характеристик у порівнянні з ЛР №12 та №13.
8. Висновки з лабораторної роботи.
9. Код програми. Код повинен задовольняти вимоги до написання коду.
10. Відповіді на контрольні запитання.

Контрольні запитання

1. Які відмінності генераторів випадкових та псевдовипадкових послідовностей Ви знаєте?
2. В чому полягають відмінності потокових шифрів від блокових?
3. Які переваги та недоліки потокових шифрів Ви знаєте?
4. Які вимоги висуваються до випадкових та псевдовипадкових генераторів ключового потоку?
5. Яка мета постобробки даних з фізичних датчиків? Чому не відправити їх для шифрування безпосередньо?
6. Порівняйте статистичні характеристики генераторів з цієї ЛР з результатми ЛР №12 та 13. Які генератори виявилися найстійкішими?
7. Чи можна робити висновок про стійкість генераторів лише на основі статистичних характеристик? Чому?