Лабораторна робота №13

Порівняльні дослідження властивостей потокового шифру власної розробки

Мета: навчитися розробляти прості потокові шифри на основі клітинних автоматів та досліджувати їх властивості.

Обладнання: персональний комп'ютер.

Програмне забезпечення: операційна система Windows (не старша за Windows 7); пакет статистичного тестування NIST STS (NIST Statistical Suite) v. 2.1.2; будь-яка мова програмування; криптофреймворк (криптобібліотека), де реалізовано основні криптоалгоритми.

Література

1. S. Worfram. WolframAlfa. - Електронний ресурс. - Режим доступу: https://www.wolframalpha.com/input/?i=rule+30&lk=1&rawformassumption=%22ClashPrefs%22+-%3E+%22ClashPrefs%22

2. Валь О.Д., Жихаревич, В.В., Овчар Р.І., Остапов С.Е. Розробка та дослідження генераторів бінарного ключового потоку на основі клітинних автоматів. Радіоелектроніка, інформатика, управління. - 2015. - № 3. - СС. 58-63.

Теоретична частина

В цій лабораторній роботі студентам пропонується розробити власний потоковий шифр та дослідити його статистичні характеристики в порівнянні з класичними шифрами.

В якості класичного шифру можна вибрати буд-який, який входить в обрану криптобібліотеку. Як платформу для власного потокового шифру пропонується обрати одновимірні клітинні автомати.

Одновимірні клітинні автомати. Під одновимірним клітинним автоматом будемо розуміти одновимірний масив з логічних елементів (boolean), які взаємодіють за заданими правилами з найближчими сусідами (або з будь-якими елементами масиву). Такі елементи називаються клітинами, а весь масив — одновимірним клітинним автоматом (КА). Елементарні правила взаємодії класифіковано С.Вольфрамом [1]. Всього їх нараховується 256. Усі розроблені та класифіковані правила взаємодії клітин можна подивитися та вивчити на сайті С.Вольфрама WolframAlfa. Це не означає, однак, що не має сенсу розробляти й використовувати в потокових шифрах власні правила (див., наприклад, [2]).

Деякі елементарні правила міжклітинної взаємодії для полегшення подано у таблиці 1.

Таблиця 1

Деякі правила міжклітинної взаємодії КА

№	Правило	Логічна форма	Арифметична форма
1	"22"	$b' = a \oplus a \wedge b \wedge c \oplus b \oplus c$	$b' = ((a+b+c+abc) \bmod 2)$
2	"30"	$b' = a \oplus (b \vee c)$	$b' = ((a+b+c+bc) \bmod 2)$
3	"54"	$b' = (a \vee c) \oplus b$	$b' = ((a+b+c+bc) \bmod 2)$
4	"86"	$b' = (a \vee b) \oplus c$	$b' = ((a+b+ab+c) \bmod 2)$
5	"135"	$b' = 1 \vee a \oplus b \vee c$	$b' = ((1+a+bc) \bmod 2)$
6	"149"	$b' = a \vee b \oplus c \vee 1$	$b' = ((1+ab+c) \bmod 2)$
7	"150"	$b' = a \oplus b \oplus c$	$b' = ((a+b+c) \bmod 2)$

8	“158”	$b' = a \oplus b \oplus c \vee b \wedge c$	$b' = ((a + b + c + bc + abc) \bmod 2)$

Для тих, хто буде використовувати КА з логічних елементів, подано логічну форму правил, тим, хто буде використовувати арифметичну форму — арифметичний еквівалент.

Взаємодія клітин за правилами відбувається наступним чином:

- масив згортається в кільце, тобто наступною для останньої клітини буде перша, а попередня для першої — остання;

- позначення в таблиці 1 такі: *b* — значення (0 або 1) поточної клітини; *a* — значення найближчого сусіда зліва; *c* — найближчого сусіда праворуч; результат взаємодії записується в поточну клітину і позначається *b’*. Іншими словами, $a_i = b$; $a_{i-1} = a$; $a_{i+1} = c$. Позначення логічних операцій стандартні.

- взаємодія починається з першої клітини, до якої застосовується одне або декілька правил, а результат після взаємодії записується до першої клітини;

- лічильник збільшується на одиницю, і взаємодіє вже друга клітина, яка використовує нове значення першої, і так до закінчення масиву.

Після того, як провзаємодіяв увесь масив, генерується черговий біт/біти для шифрування (тобто ключовий потік).

Виведення бітів для шифрування відкритого повідомлення може виконуватися у різний спосіб:

- після кожного циклу взаємодії виводиться один, строго визначений біт масиву, наприклад, тринадцятий; для прискорення роботи генератора можна вибирати кілька бітів, наприклад, 13-й, 157-й, 213-й тощо, тобто одразу шифруються кілька бітів відкритого повідомлення;

- вибираються кілька бітів, над ними виконуються якісь додаткові логічні операції, а результат виводиться для шифрування;

- біти вибираються по додатковому ключу, який узгоджується перед шифруванням.

Можливі й інші варіанти організації ключового потоку, тут немає якихось строгих рекомендацій за винятком якості ключового потоку, про яку буде сказано пізніше.

Потокові шифри. Нагадаємо, що потоковим шифром ми називаємо такий, коли до бінарного представлення відкритого повідомлення додається за правилом додавання за модулем два (XOR) випадковий (або псевдовипадковий) ключовий потік. В результаті утворюється зашифрований текст. Криптостійкість таких шифрів грунтується на відомому факті з теорії ймовірностей: якщо до детермінованої величини (відкритий текст) додається випадкова величина (ключ шифрування), то в результаті отримується випадкова величина.

Більше того, для розшифрування повідомлення використовується той самий ключовий потік, який знову накладається за правилом XOR на зашифроване повідомлення. Оскільки операція XOR обернена сама до себе, в результаті такої операції ми отримаємо відкритий текст.

Таким чином, усі властивості (в тому числі, й криптостійкість) потокового шифру визначаються якістю ключового потоку. Він повинен задовольняти вимоги, визначені ще Клодом Шенноном:

- бути випадковим;

- довжина його повинна дорівнювати довжині вхідного тексту;

- використовуватися лише один раз.

У цьому випадку цей потоковий шифр вважається ідеальним. Однак, основна проблема такого шифру полягає в тому, що для розшифрування необхідно на приймальну сторону передати ключовий потік. Це надзвичайно незручно, тому такий шифр дуже рідко використовується на практиці.

Простіше використати не випадковий ключовй потік, отриманий від випадкового джерела, а псевдовипадковий, який можна відтворити на приймальному боці без передавання його мережею. Однак, в такому разі псевдовипадковий ключовий потік за своїми властивостями повинен наближатися якомога ближче до випадкового. Як в такому разі оцінити його криптостійкість? Для цього треба оцінити його статистичні властивості за допомогою пакета NIST STS, роботу з яким описано у попередній лабораторній роботі. Якщо ключовий потік проходить усі тести статистичного пакету, вважають, що такий генератор придатний для використання у криптографічних застосуваннях.

Практична частина

Для виконання цієї лабораторної роботи необхідно зробити таке:

1. Реалізувати потоковий шифр, де в якості генератора ключової послідовності використати один з генераторів, доступних у криптобібліотеці (BBS, Blum-Micali, SHA тощо).

2. Зашифрувати вхідний файл за допомогою цього шифру. Розмір файлу повинен бути не менше 12,5 МБ для того, щоби можно було виконати статистичне тестування у пакеті NIST STS.

3. Виконати статистичне тестування за допомогою пакета NIST STS 2.1.2.

4. Розробити власний генератор ключової послідовності на КА, використавши будь-які правила міжклітинної взаємодії з таблиці 1. Можна також використати інші елементарні правила взаємодії, скориставшись сайтом WolframAlfa, або придумати власні правила. Можно також використати комбінації правил, застосовуючи їх по черзі або для послідовних клітин.

5. На основі розробленого генератора побудувати потоковий шифр.

6. Зашифрувати файл, використаний у п. 2, за допомогою розробленого шифру.

7. Результат шифрування протестувати за допомогою пакету NIST STS.

8. Порівняти статистичні характеристики Вашого шифру з характеристиками, отриманими у п.3. Для цього побудувати необхідні порівняльні таблиці та діаграми.

9. Зробити висновок стосовно якості розробленого Вами потокового шифру.

10. Експериментуючи з правилами взаємодії та їх комбінаціями, досягти максимально можливих статистичних характеристик Вашого шифру. Зробити відповідні висновки.

11. Підготувати звіт з лабораторної роботи, який повинен містити:

- Протокол Ваших дій;

- Правила міжклітинної взаємодії, обрані та модифіковані Вами;

- Результати статистичного тестування зашифрованих файлів;

- Висновки, зроблені Вами з лабораторної роботи;

- Відповіді на контрольні запитання.

Контрольні запитання

1. Охарактеризуйте поняття ідеального шифру.

2. Чи є шифр, розроблений Вами, ідеальним? Чому?

3. Особливості, переваги та недоліки потокових шифрів.

4. Яка математична проблема лежить в основі криптостійкості потокового шифру?

5. Охарактеризуйте особливості використаного у Вашій розробці генератора ключової послідовності.

6. Охарактеризуйте особливості розробленого Вами генератора ключової послідовності.