Лабораторная работа № 3

Цель работы: Научиться применять операцию XOR в шифровании, научиться шифровать методом «одноразового блокнота».

Теоретическая часть:

XOR

XOR представляет собой операцию «исключающее или»: '^' в языке C.

Это обычная операция над битами:

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

Также заметим, что:

$$a \oplus a = 0$$

$$a \oplus b \oplus b = a$$

Открытый текст подвергается операции «исключающее или» вместе с ключевым текстом для получения шифротекста. Так как повторное применение

операции XOR восстанавливает оригинал для шифрования и дешифрирования используется одна и та же программа.

Ниже приведена программная реализация XOR-шифрования и расшифровки:

```
import java.util.Arrays;
public class Xor
{
	public static void main(String[] args)
	{
		byte openText[] = {'h', 'e', 'l', 'l', 'o', 1 , 2, 3, 4, 5};
		byte key[] = {5, 9, 12};
		byte encryptMas[] = new byte[openText.length];
		byte decryptMas[] = new byte[openText.length];
		for(int i = 0; i < openText.length; i++)
			encryptMas[i]=(byte)(openText[i] ^ key[i % key.length]);
		for(int i = 0; i < openText.length; i++)
			decryptMas[i]=(byte)(encryptMas[i] ^ key[i % key.length]);
		boolean bOK = Arrays.equals(openText, decryptMas);
		if(bOK)
			System.out.print("Сообщение восстановлено");
	}
}
```

Одноразовый блокнот

Метод шифрования, называемый **одноразовым блокнотом** был изобретен в 1917 году Мэйджором Джозефом Моборном (Major Joseph Mauborgne) и Гилбертом Вернамом (Gilbert Vernam) из AT&T. В классическом понимании одноразовый блокнот является большой неповторяющейся последовательностью

символов ключа, распределенных случайным образом, написанных на кусочках бумаги и приклеенных к листу блокнота. Первоначально это была одноразовая лента для телетайпов. Отправитель использовал каждый символ ключа блокнота для шифрования только одного символа открытого текста. Шифрование представляет собой сложение по модулю 26 символа открытого текста и символа ключа из одноразового блокнота.

Каждый символ ключа используется только единожды и для единственного сообщения. Отправитель шифрует сообщения и уничтожает использованные страницы блокнота или использованную часть ленты. Получатель, в свою очередь, используя точно такой же блокнот, дешифрирует каждый символ шифротекста. Расшифровав сообщение, получатель уничтожает соответствующие страницы блокнота или часть ленты. Новое сообщение - новые символы ключа.

В предположении, что злоумышленник не сможет получить доступ к одноразовому блокноту, использованному для шифрования сообщения, эта схема совершенно безопасна. Данное шифрованное сообщение

на вид соответствует любому открытому сообщению того же размера.

Так как все ключевые последовательности совершенно одинаковы (помните, символы ключа генерируются случайным образом), у противника отсутствует информация, позволяющая подвергнуть шифротекст криптоанализу.

Так как все открытые тексты равновероятны, у криптоаналитика нет возможности определить, какой из открытых текстов является правильным. Случайная ключевая последовательность, сложенная с неслучайным открытым текстом, дает совершенно случайный шифротекст, и никакие вычислительные мощности не смогут это изменить. Символы ключа должны генерироваться случайным образом.

Ключевую последовательность никогда нельзя использовать второй раз. Даже если вы используете блокнот размером в несколько гигабайт, то если криптоаналитик получит несколько текстов с перекрывающимися ключами, он сможет восстановить открытый текст. Он сдвинет каждую пару шифротекстов

относительно друг друга и подсчитает число совпадений в каждой позиции. Если шифротексты смещены правильно, соотношение совпадений резко возрастет - точное значение зависит от языка открытого текста. С этой точки зрения криптоанализ не представляет труда. Это похоже на показатель совпадений, но сравниваются два различных «периода». Не используйте ключевую последовательность повторно.

Ниже приведена программная реализация, демонстрирующая шифрование и дешифрование сообщения с помощью одноразового блокнота.

```
import java.util.Random;
public class Bloknot
{
	public static void main(String[] args)
	{
		if(args.length == 0)
			return;
		byte bloknotMas[] = new byte[1000];
		//// вместо инициализации блокнота
		Random rand = new Random();
		rand.nextBytes(bloknotMas);
		////
		byte ret[][] = new byte[args.length][];
		int iCurIndex = 0;
		for(int j = 0; j < args.length; j++)
		{
			String str = args[j];
			byte mas[] = str.getBytes();
			ret[j] = new byte[mas.length];
			int k = 0;
			for( int i = 0; i < mas.length; i++, k++)
```

```
                {
                        if(iCurIndex + i >= bloknotMas.length)
                        {
                // вместо смены страницы блокнота и уничтожения старой
                                rand.nextBytes(bloknotMas);
                                iCurIndex = 0;
                                k = 0;
                        }
                        ret[j][i]=(byte)(mas[i]^bloknotMas[iCurIndex+k]);
                }
                iCurIndex += k;
        }
    }
}
```

Идея одноразового блокнота легко расширяется на двоичные данные. Вместо одноразового блокнота, состоящего из букв, используется одноразовый блокнот из битов. Вместо сложения открытого текста с ключом одноразового блокнота используйте XOR. Для дешифрирования примените XOR к шифротексту с тем же одноразовым блокнотом. Все остальное не меняется, и безопасность остается такой же совершенной.

Задание 1: Реализуйте на любом языке программирования XOR-шифрование, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его.

Задание 2: Реализуйте на любом языке программирования шифрование методом одноразового блокнота, зашифруйте с помощью вашего шифра сообщение на русском языке, после чего дешифруйте его.

Задание 3: Реализуйте на любом языке программирования шифрование методом одноразового блокнота, зашифруйте с помощью вашего шифра сообщение, состоящее из двоичных данных (используя XOR), после чего дешифруйте его.

Контрольные вопросы:

1. В чем заключается смысл шифрования одноразовым блокнотом?
2. Что такое XOR-шифрование? Где его применяют?
3. Почему шифрование методом одноразового блокнота считается идеальным шифрованием.
4. Какие достоинства и недостатки шифрования методом одноразового блокнота вы можете выделить?
5. Каким образом можно получить данные, зашифрованные с помощью одноразового блокнота?

6. Каким образом можно получить данные, зашифрованные XOR?

Содержание отчета:

1. Титульный лист.
2. Задание.
3. Экранные снимки, подтверждающие выполнение проделанных шагов.
4. Ответы на контрольные вопросы.

Литература:

1. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер – М.: ТРИУМФ, 2002. –816 с.