

Лабораторная работа #7

Элементы криптографии. Однократное гаммирование

Критский Сергей Димитриевич

Содержание

Цель работы	3
Выполнение лабораторной работы	4
Написание программы	4
Результат выполнения	6
Выводы	7

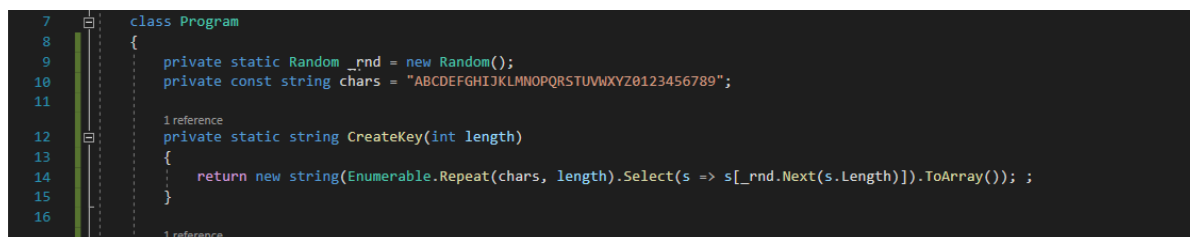
Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение лабораторной работы

Написание программы

Написал функцию генерации случайного ключа одинаковой с сообщением длины из указанных символов.



```
7 class Program
8 {
9     private static Random _rnd = new Random();
10    private const string chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
11
12    private static string CreateKey(int length)
13    {
14        return new string(Enumerable.Repeat(chars, length).Select(s => s[_rnd.Next(s.Length)]).ToArray()); ;
15    }
16 }
```

Рис. 1: Функция GenerateKey

Написал функции шифрования и дешифрования сообщений, путем выполнения операции XOR к каждому символу ключа и исходного сообщения (зашифрованного сообщения, если необходимо расшифровать).

```

private static string Encrypt(string message, string key)
{
    string result = "";

    for(int i = 0; i < message.Length; i++)
    {
        result += Convert.ToChar(message[i] ^ key[i]);
    }

    return result;
}

1 reference
private static string Decrypt(string encrypted, string key)
{
    string result = "";

    for (int i = 0; i < encrypted.Length; i++)
    {
        result += Convert.ToChar(encrypted[i] ^ key[i]);
    }

    return result;
}

```

Рис. 2: De- Encrypt функции

Проинициализировал необходимые переменные в основном теле программы, перевел ключ и зашифрованное сообщение в 16-ную систему для понятности.

```

    }

    0 references
    static void Main(string[] args)
    {
        string str = "С Новым годом, друзья!";
        int length = str.Length;
        string key = CreateKey(length);
        string encrypted = Encrypt(str, key);

        byte[] encrypted_code = Encoding.Unicode.GetBytes(encrypted);
        byte[] key_code = Encoding.Unicode.GetBytes(key);

        string trans_encrypted = BitConverter.ToString(encrypted_code).Replace("-", " ");
        string trans_key = BitConverter.ToString(key_code).Replace("-", " ");

        Console.WriteLine($"Сгенерированный ключ: {key}");
        Console.WriteLine($"Сгенерированный ключ в 16-ом виде: {trans_key}");
        Console.WriteLine($"Зашифрованные данные в 16-ом виде: {trans_encrypted}");
        Console.WriteLine("Результат дешифровки:" + Decrypt(encrypted, key));
    }
}

```

Рис. 3: Main программы

Результат выполнения

Запустил программу.

```

Microsoft Visual Studio Debug Console
Сгенерированный ключ: Q5ZY089QD2Q1CNBH9IMEL4
Сгенерированный ключ в 16-ом виде: 51 00 35 00 5A 00 59 00 4F 00 38 00 39 00 51 00 44 00 32 00 51 00 31 00 43 00 4E 00 42 00 48 00 39 00 49 00 57 00 45 00 4C 00 34 00
Зашифрованные данные в 16-ом виде: 70 04 15 00 47 04 67 04 7D 04 73 04 05 04 71 00 77 04 0C 04 65 04 0F 04 7F 04 62 00 62 00 7C 04 79 04 0A 04 60 04 09 04 03 04 15 00
Результат дешифровки:С Новым годом, друзья!
D:\Info security\Lab_7\Prog\Lab_7\Lab_7\bin\Debug\netcoreapp3.1\Lab_7.exe (process 35044) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

Рис. 4: Результат выполнения написанной программы

Выводы

Я освоил на практике применение режима однократного гаммирования.