Практика №9. Генерация гаммы шифра с большим периодом. Шифрование и дешифровка методом гаммирования

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

Гаммирование является широко применяемым криптографическим преобразованием.

Под *гаммированием* понимают процесс наложения по определенному закону гаммы шифра на открытые данные. *Гамма шифра* - это псевдослучайная последовательность, выработанная по заданному алгоритму для шифровки открытых данных и дешифровки зашифрованных данных.

Процесс *шифровки* заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы на исходный открытый текст обратимым образом, например с использованием операции сложения по модулю 2.

Следует отметить, что перед шифровкой открытые данные разбивают на блоки $T_o^{(i)}$ одинаковой длины, обычно по 64 бита. Гамма шифра вырабатывается в виде последовательности блоков $Г_ш^{(i)}$ аналогичной длины.

Уравнение шифровки можно записать в виде

$$T_ш^{(i)} = Г_ш^{(i)} \oplus T_o^{(i)},\ i = 1 \ldots M,$$

где $T_ш^{(i)}$ *i*-й блок шифртекста;
$Г_ш^{(i)}$ *i*-й блок гаммы шифра;
$T_o^{(i)}$ *i*-й блок открытого текста;
M количество блоков открытого текста.

Процесс *дешифровки* сводится к повторной генерации гаммы шифра и наложению этой гаммы на зашифрованные данные. Уравнение дешифровки имеет вид

$$T_o^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)},\ i = 1 \ldots M.$$

Получаемый этим методом шифртекст достаточно труден для раскрытия, поскольку ключ здесь является переменным. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока. Если период гаммы превышает длину всего шифруемого текста и злоумышленнику неизвестна никакая часть исходного текста, то такой шифр можно раскрыть только прямым перебором всех вариантов ключа. В этом случае криптостойкость шифра определяется длиной ключа.

Однако, метод гаммирования становится бессильным, если злоумышленник узнает фрагмент исходного текста и соответствующую ему шифрограмму. Простым вычитанием по модулю получается отрезок псевдослучайной последовательности и по нему восстанавливается вся последовательность.

Методы генерации псевдослучайных последовательностей чисел

При шифровании методом гаммирования в качестве ключа используется случайная строка битов, которая объединяется с открытым текстом, также представленным в двоичном виде, с помощью побитового сложения по модулю 2, и в результате получается шифрованный текст. Генерирование непредсказуемых двоичных последовательностей большой длины является одной из важных проблем классической криптографии. Для решения этой проблемы широко используются генераторы двоичных псевдослучайных последовательностей.

Генерируемые псевдослучайные ряды чисел часто называют гаммой шифра или просто гаммой (по названию буквы © греческого алфавита, часто используемой в математических формулах для обозначения случайных величин).

Обычно для генерации последовательности псевдослучайных чисел применяют компьютерные программы, которые, хотя и называются генераторами случайных чисел, на самом деле вырабатывают детерминированные числовые последовательности, которые по своим свойствам очень похожи на случайные.

К криптографически стойкому генератору ПСП чисел (гаммы шифра) предъявляются три основных требования:

период гаммы должен быть достаточно большим достаточно большим для шифрования сообщений различной длины;

гамма должна быть практически непредсказуемой, что означает невозможность предсказать следующий бит гаммы, даже если известны тип генератора и предшествующий кусок гаммы;

генерирование гаммы не должно вызывать больших технических сложностей;

Длина периода гаммы является самой важной характеристикой генератора ПСЧ. По окончании периода числа начнут повторяться и их можно будет предсказать.

Один из первых способов генерации ПСЧ на ЭВМ предложил в 1946 г. Джон фон Нейман. Суть этого способа состоит в том, что каждое последующее случайное число образуется возведением в квадрат предыдущего числа с отбрасыванием цифр младших и старших разрядов. Однако этот способ оказался ненадежным и от него вскоре отказались.

Из известных процедур генерации последовательности ПСЧ наиболее часто применяется так называемый *линейный конгруэнтный генератор*. Этот генератор вырабатывает последовательность ПСЧ Y_1, Y_2, ..., Y_{i-1}, Y_i, ..., используя соотношение

$$Y_i = (a \bullet Y_{i-1} + b) \bmod m,$$

где Y_i i-е (текущее) число последовательности;
Y_{i-1} предыдущее число последовательности;
m – модуль;
a – множитель;
b – приращение;
a Y_0 - порождающее число (исходное значение).

Текущее псевдослучайное число Y_i получают из предыдущего числа Y_{i-1} умножением его на коэффициент a, сложением с приращением b и вычислением остатка от деления на m. Данное уравнение генерирует ПСЧ с периодом повторения, зависящим от выбранных значений a и b и может достигать значения m. Значение m обычно устанавливается равным 2^n, где n - длина машинного слова в битах, либо равным простому числу, например $m=2^{31}-1$. Как показано Д. Кнутом, линейный конгруэнтный датчик ПСЧ имеет максимальный период тогда и только тогда, когда b - нечетное, и a mod 4 = 1.

Также для получения последовательности ПСЧ применяются аддитивные и мультипликативные генераторы.

Мультипликативный генератор вырабатывает последовательности чисел с помощью рекуррентного соотношения:

$$Y_i = (a \bullet Y_{i-1}) \bmod m.$$

Требования к значениям констант a и m такие же, как и для линейного конгруэнтного генератора.

Текущее случайное число Y_i *аддитивного датчика* получается из суммы чисел Y_{i-1} и Y_{i-2} вычислением модуля от деления этой суммы на m:

$$Y_i = (Y_{i-1} + Y_{i-2}) \bmod m.$$

Описание алгоритмов.

Алгоритм шифровки.

Проинициализировать датчик случайных чисел.

Выделить блок открытого текста.

Сгенерировать гамму шифра.

Получить блок зашифрованного текста, сложив по модулю 2 блок открытого текста с гаммой шифра.

Если текст не закончился, перейти к пункту 2, иначе к пункту 6.

Конец алгоритма шифровки.

Алгоритм дешифровки.

Проинициализировать датчик случайных чисел.

Выделить блок зашифрованного текста.

Сгенерировать гамму шифра.

Получить блок открытого текста, сложив по модулю 2 блок зашифрованного текста с гаммой шифра.

Если зашифрованный текст не закончился, перейти к пункту 2, иначе к пункту 6.

Конец алгоритма дешифровки.

ЗАДАНИЕ

Выбрать фрагмент открытого текста.

Сгенерировать гамму шифра требуемой длины одним из генераторов псевдослучайных последовательностей.

Зашифровать фрагмент открытого текста.

Дешифровать зашифрованный текст.