Лекция 9. Симметричные криптосистемы

1. Элементы теории чисел.
2. Способы симметричного шифрования.
3. Современные симметричные криптосистемы. Абсолютно стойкий шифр.

Понятие вычета по модулю

- Целые числа a и b *сравнимы по модулю* n (целому числу, неравному нулю), если выполняется условие

a=b+k•n

для некоторого целого числа k. В этом случае обычно используется следующая запись:

a=b {mod n}

- Сравнимость a и b по модулю n означает, что n делит a-b нацело:
- **n | (a-b)**

Если b≥0, a=b {mod n} и |b|<n, то b называют *вычетом* числа a по модулю n. Вычет равен остатку от целочисленного деления числа a на число n. Операцию нахождения вычета числа a по модулю n называют *приведением* числа a по модулю n.

Свойства вычетов

- -a {mod n} = -a+n {mod n}
- n=0 {mod n}

Примеры:

- 3+10 {mod 12} = 1 {mod 12} («арифметика» часов);
- -5 {mod 7} = 2 {mod 7}.
- *Полным набором вычетов* по модулю n называется множество целых чисел от нуля до n-1:

{0, 1, 2, … , n-1}

- Вычеты по модулю n с применением операций сложения и умножения образуют *коммутативное кольцо*, в котором справедливы законы ассоциативности, коммутативности и дистрибутивности.

Свойства операций над вычетами

- аддитивности

$(a+b) \{mod\ n\} = (a \{mod\ n\} + b \{mod\ n\}) \{mod\ n\}$

- мультипликативности

$(a \cdot b) \{mod\ n\} = (a \{mod\ n\} \cdot b \{mod\ n\}) \{mod\ n\}$

- сохранения степени

$a^b \{mod\ n\} = (a \{mod\ n\})^b \{mod\ n\}$

- Данные свойства операций над вычетами позволяют либо сначала вычислять вычеты, а затем выполнять операцию, либо сначала выполнять операцию, а затем вычислять вычеты. Операция вычисления вычета является *гомоморфным отображением* кольца целых чисел в кольцо вычетов по модулю n.

НОД и простые числа

- *Наибольшим общим делителем* (НОД) целых чисел a и b называется наибольшее целое число, на которое делятся без остатка a и b.
- *Простым числом* называется целое число, которое делится без остатка только на единицу и на себя.
- Целые числа a и b называются *взаимно простыми*, если выполняется условие НОД(a, b)=1.
- Целое число y называется *мультипликативно обратным* целому числу x по модулю n, если выполняется условие $x \cdot y \{\mathrm{mod}\ n\} = 1$. Мультипликативно обратное целое число существует только тогда, когда x и n – взаимно простые числа. Если целые числа a и n не являются взаимно простыми, то сравнение $a^{-1}=x \{\mathrm{mod}\ n\}$ не имеет решения.

Функция Эйлера

- Если из полного набора вычетов по модулю n выделить подмножество вычетов, взаимно простых с n, то получим *приведенный набор вычетов*. Например:

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10} – полный набор вычетов по модулю 11. Приведенным набором вычетов будет то же подмножество целых чисел за исключением нуля.

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9} – полный набор вычетов по модулю 10. Приведенным набором вычетов будет подмножество целых чисел {1, 3, 7, 9}.

Функция Эйлера

- Очевидно, что если n является простым числом, то приведенный набор вычетов по модулю n всегда содержит n-1 элемент (все целые числа от единицы до n-1).
- Значением *функции Эйлера* φ(n) будет количество элементов в приведенном наборе вычетов по модулю n.
- Если n – простое число, то φ(n)=n-1 и φ(n^2)=n•(n-1). Если n=p•q (p и q – простые числа и p≠q), то φ(n)=(p-1)•(q-1).

Китайская теорема об остатках (1-й век н.э.)

Если

- $m_1, m_2, \ldots, m_k$ – попарно взаимно простые числа, большие 1 (модули);
- $M=m_1 \cdot m_2 \cdot \ldots \cdot m_k$ (произведение модулей);
- $a_1, a_2, \ldots, a_k$ – вычеты по модулям $m_1, m_2, \ldots, m_k$ неотрицательного числа x, меньшего M, то

$$x = \sum_{i=1}^{k} a_i * N_i * M_i \{\mathrm{mod}\, M\}$$

где $M_i=M/m_i$ и N_i – мультипликативно обратное к M_i по модулю m_i ($M_i \cdot N_i=1 \{\mathrm{mod}\ m_i\}$).

Пример использования теоремы об остатках

Если $x=1 \{mod\ 2\}$, $x=2 \{mod\ 5\}$ и $x=3 \{mod\ 9\}$, то $x=57$.

Малая теорема Ферма

Если a – целое число, n – простое число и НОД(a, n)=1, то

$a^{n-1}=1 \{mod\ n\}$.

Теорема Эйлера

Является обобщением малой теоремы Ферма: если целые числа a и n являются взаимно простыми (НОД(a, n)=1), то

$a^{\varphi(n)}=1 \{\mod n\}$.

Причины использования вычетов в криптографии

- Выполнение обратных операций (логарифмирование, извлечение корня, разложение на простые сомножители – факторизация) гораздо более трудоемко, чем выполнение прямых операций (возведения в степень или произведения).
- При вычислениях с вычетами ограничивается диапазон возможных промежуточных значений и результата (например, $a^{25}\{\mathrm{mod}\ n\}=((((a^2\cdot a)^2)^2)^2)\cdot a\{\mathrm{mod}\ n\}$).

Способы симметричного шифрования

- Перестановки.
- Подстановки (замены).
- Гаммирование.

Шифры перестановок

Биты (или символы) открытого текста переставляются в соответствии с задаваемым ключом шифрования правилом:

$\forall$ i, $1 \le i \le n$ $C_i = P_{k[i]}$, где

- $P = \langle P_1, P_2, \ldots, P_i, \ldots, P_n \rangle$ – открытый текст;
- n – длина открытого текста;
- $C = \langle C_1, C_2, \ldots, C_i, \ldots, C_n \rangle$ – шифротекст;
- $k = \langle k_1, k_2, \ldots, k_i, \ldots, k_n \rangle$ – ключ шифрования.

Шифры перестановок

При расшифровании применяется обратная перестановка:

$\forall$ i, $1 \le i \le n$ $P_{k[i]} = C_i$.

Очевидно, что при шифровании перестановкой ключ должен удовлетворять условию:

$\forall k_i \in k$ $1 \le k_i \le n \wedge \forall k_i, k_j \in k$ $(i \ne j)$ $k_i \ne k_j$.

Шифры перестановок

- Пример. Пусть надо зашифровать слово «связной» (n=7) с помощью ключа k={4, 2, 1, 7, 6, 3, 5}. В результате шифрования мы получаем шифротекст «звсйоян».
- Если длина ключа меньше длины открытого текста, то можно разбить открытый текст на блоки, длина которых равна длине ключа, и последовательно применить ключ перестановки к каждому блоку открытого текста. Если длина открытого текста не кратна длине ключа, то последний блок может быть дополнен пробелами или нулями.

Шифры перестановок

- Можно использовать и другой прием. После разбиения открытого текста длиной *n* на блоки, длина которых равна длине ключа *m*, открытый текст записывается в таблицу с числом столбцов, равным длине ключа (каждый блок открытого текста записывается в столбец таблицы). Количество строк таблицы в этом случае будет равно наименьшему целому числу, не меньшему *n*/*m*. Затем столбцы полученной таблицы переставляются в соответствии с ключом перестановки, а шифротекст считывается по строкам таблицы.

Шифры перестановок

- При расшифровании шифротекст записывается в таблицу того же размера по строкам, затем происходит обратная перестановка столбцов в соответствии с ключом, после чего расшифрованный текст считывается из таблицы по столбцам.
- Достоинством шифрования перестановкой является высокая скорость получения шифротекста.
- К недостаткам шифрования перестановкой относятся сохранение частотных характеристик открытого текста после его шифрования (символы открытого текста лишь меняют свои позиции в шифротексте) и малое число возможных ключей шифрования.

Шифры подстановок

- При шифровании с помощью *подстановки* (замены) каждый символ открытого текста заменяется другим символом одного и того же (*одноалфавитная подстановка*) или разных (*многоалфавитная подстановка*) алфавитов в соответствии с определяемым ключом шифрования правилом.

Одноалфавитная подстановка

$\forall$ i, $1 \le i \le n$ $C_i = P_i + k \{\mathrm{mod}\ m\}$, где

- $P = \langle P_1, P_2, \ldots, P_i, \ldots, P_n \rangle$ – открытый текст;
- n – длина открытого текста;
- $A = \{A_1, A_2, \ldots, A_m\}$ – алфавит символов открытого текста ($\forall$ i, $1 \le i \le n$ $P_i \in A$);
- $C = \langle C_1, C_2, \ldots, C_i, \ldots, C_n \rangle$ – шифротекст;
- k – ключ шифрования ($0 \le k < m$);
- $\forall a_i \in A$, $1 \le i \le m$ $a_i + k \{\mathrm{mod}\ m\} = a_{i+k\{\mathrm{mod}\ m\}}$.

Одноалфавитная подстановка

- При расшифровании символ шифротекста заменяется символом, номер которого в используемом алфавите больше номера символа шифротекста на величину m-k (m – мощность используемого алфавита, а k – ключ шифрования; применяется операция сложения в кольце вычетов по модулю m):

$\forall\ i,\ 1 \le i \le n\ C_i = P_i + m - k \{mod\ m\}$

- Пример. При шифровании открытого текста «наступайте» с помощью одноалфавитной подстановки по ключу 3 (так называемой подстановки Цезаря) получаем шифротекст «ргфхцтгмхз».

Одноалфавитная подстановка

К основным недостаткам относится:

- сохранение частоты появления различных символов открытого текста в шифротексте (одинаковые символы открытого текста остаются одинаковыми и в шифротексте);
- малое число возможных ключей.

Многоалфавитная подстановка

$\forall$ i, $1 \le i \le n$ $C_i = P_i + k_i \{\mathrm{mod}\ m\}$, где

- $P = \langle P_1, P_2, \ldots, P_i, \ldots, P_n \rangle$ – открытый текст;
- n – длина открытого текста;
- $A = \{A_1, A_2, \ldots, A_m\}$ – алфавит символов открытого текста ($\forall$ i, $1 \le i \le n$ $P_i \in A$);
- $C = \langle C_1, C_2, \ldots, C_i, \ldots, C_n \rangle$ – шифротекст;
- $k = \langle k_1, k_2, \ldots, k_i, \ldots, k_n \rangle$ – ключ шифрования ($\forall$ i, $1 \le i \le n$ $0 \le k_i < m$);
- $\forall a_i \in A$, $1 \le i \le m$ $a_i + k \{\mathrm{mod}\ m\} = a_{i+k\{\mathrm{mod}\ m\}}$.

Многоалфавитная подстановка

- Расшифрование:

$\forall$ i, $1 \le i \le n$ $C_i = P_i + m - k_i$ {mod m}.

- Если длина ключа меньше длины открытого текста, то необходимо разбить открытый текст на блоки, длина которых равна длине ключа, и последовательно применить ключ подстановки к каждому блоку открытого текста. Если длина открытого текста не кратна длине ключа, то для шифрования последнего блока надо взять только первые l элементов ключа (l – длина последнего блока).

Многоалфавитная подстановка

- К достоинствам относится то, что в шифротексте маскируется частота появления различных символов открытого текста. Поэтому криптоаналитик не может при вскрытии шифра использовать частотный словарь букв естественного языка.

Шифры гаммирования

Шифротекст получается путем наложения на открытый текст *гаммы* шифра с помощью какой-либо обратимой операции (как правило, поразрядного сложения по модулю 2):

$\forall$ i, $1 \le i \le n$ $C_i = P_i \oplus G_i$, где

- $P = \langle P_1, P_2, \ldots, P_i, \ldots, P_n \rangle$ – открытый текст;
- n – длина открытого текста;
- $C = \langle C_1, C_2, \ldots, C_i, \ldots, C_n \rangle$ – шифротекст;
- $G = \langle G_1, G_2, \ldots, G_i, \ldots, G_n \rangle$ – гамма шифра;
- $\oplus$ - операция поразрядного сложения по модулю 2.

Шифры гаммирования

- Расшифрование заключается в повторном наложении той же гаммы шифра на шифротекст:

$\forall$ i, $1 \le i \le n$ $P_i = C_i \oplus G_i$.

- Гамма шифра вычисляется с помощью программного или аппаратного *датчика* (*генератора*) *псевдослучайных чисел*, параметры которого определяются ключом шифрования.

Современные симметричные криптоалгоритмы

- Потоковые (результат шифрования каждого бита открытого текста зависит от ключа шифрования и значения этого бита).
- Блочные (результат шифрования каждого бита открытого текста зависит от ключа шифрования и значений всех битов шифруемого блока и, возможно, предыдущего блока).

Потоковые шифры

- В основе лежит гаммирование. Криптостойкость полностью определяется структурой используемого генератора псевдослучайной последовательности (чем меньше период псевдослучайной последовательности, тем ниже криптостойкость потокового шифра).
- Основным преимуществом является высокая производительность. Эти шифры наиболее пригодны для шифрования непрерывных потоков открытых данных (например, в сетях передачи данных или связи).

Потоковые шифры

- К наиболее известным относятся:
- RC4 (Rivest Cipher 4), разработанный Р. Ривестом (R.Rivest); в шифре RC4 может использоваться ключ переменной длины;
- SEAL (Software Encryption ALgorithm) – приспособленный для программной реализации потоковый шифр, использующий ключ длиной 160 бит;
- WAKE (Word Auto Key Encryption).

Блочные шифры

- В этих криптосистемах открытый текст разбивается на блоки фиксированной, как правило, длины, и к каждому блоку применяется функция шифрования, использующая перестановки битов блока и многократное повторение операций подстановки и гаммирования, после чего над зашифрованными блоками может выполняться дополнительная операция перед включением их в шифротекст.

Блочные шифры

К наиболее распространенным способам построения блочных шифров относится *сеть Фейстела*, при использовании которой каждый блок открытого текста представляется сцеплением двух полублоков одинакового размера $L_0||R_0$. Затем для каждой итерации (раунда) i выполняется следующее:

1. $L_i=R_{i-1}$;
2. $R_i=L_{i-1} \oplus f(R_{i-1}, k_i)$, где

- f – функция шифрования;
- k_i – *внутренний ключ*, используемый на i-м раунде шифрования (k_i определяется исходным ключом шифрования открытого текста и номером раунда).

Название шифра	Длина блока	Раундов	Длина ключа
DES (Data Encryption Standard)	64	16	64 (8 контрольных)
3-DES (Triple-DES)	64	48	168
DESX (DES eXtended)	64	16	184
ГОСТ 28147-89	64	32	256
IDEA (International Data Encryption Algorithm)	64	8	128
AES (Advanced Encryption Standard)	128	14	128, 192, 256
RC2 (Rivest Cipher 2)	64	Переменное	Переменная
RC5 (Rivest Cipher 5)	32, 64, 128	Переменное	Переменная
RC6 (Rivest Cipher 6)	Переменная	Переменное	Переменная
CAST (C.Adams, S.Tavares)	64	16	128
Blowfish	64	16	Переменная
SAFER+	128	8, 12, 16	128, 192, 256
Skipjack	64	32	80

Совершенный шифр

- $\forall$ X, Y p(X|Y)=p(X), где
- p(X) – вероятность выбора для шифрования открытого текста X,
- p(X|Y) – вероятность передачи открытого текста X при условии перехвата шифротекста Y.

Условия построения идеального (абсолютно стойкого) шифра

Определены К.Шенноном:

- ключ шифрования вырабатывается совершенно случайным образом;
- один и тот же ключ должен применяться для шифрования только одного открытого текста;
- длина шифруемого открытого текста не должна превышать длину ключа шифрования.

Условия К.Шеннона

- К сожалению, в большинстве случаев выполнение этих условий обеспечить практически невозможно, хотя короткие и наиболее важные сообщения следует шифровать именно так. Для открытых текстов большой длины главной проблемой симметричной криптографии является генерация, хранение и распространение ключа шифрования достаточной длины.

Блочные шифры

- Очевидно, что за счет увеличения длины ключа шифрования можно уменьшить требования к сложности алгоритма блочного шифрования (например, уменьшить количество раундов) и наоборот – более короткий ключ требует увеличения сложности криптоалгоритма.