Содержание

Введение. Краткая историческая справка

Мы окружены многочисленными физическими полями, которые имеют способность к изменению. Изменение – это и есть процесс передачи информации. Если процесс ее считывания является неизбежным, то структура физического поля должна быть организована таким образом, чтобы максимально усложнить считывание информации.

Криптография – наука о шифрах. Ее история восходит к древнейшим временам. Первые сведения о применении шифров в военном деле связаны с именем спартанского полководца Лисандра, использовавшего для передачи сообщений шифр «Сцитала».

Шифр «Сцитала»

Этот способ шифрования основывается на использовании двух эталонных жезлов. На один из них наматывалась пергаментная лента — таким образом, чтобы не было ни просветов, ни нахлестов. Затем на ней записывался текст, дешифровка которого производилась с использованием такого же жезла, только у получателя шифрограммы.

Квадрат Полибия

Шифр, именуемый «квадратом Полибия», был изобретен в Древней Греции. Его суть состоит в замене символов исходного текста на их номера в специальной таблице:

	1	**2**	**3**	**4**
5	A	B	C	D
6	E	F	I	G
7	K	L	M	N
8	O	P	Q	R

Таким образом, слово «МАМА» будет закодировано как «37153715».

Код Альберти

Этот способ шифрования несколько более сложен, чем рассмотренные ранее. Основывался он на использовании «Диска Альберти». Состоял он из двух дисков – внешнего неподвижного, на котором были нанесены латинские буквы в алфавитном порядке и цифры от 1 до 9, и подвижного внутреннего диска, на котором буквы были переставлены. Диски закреплялись на одной оси таким образом, чтобы внутренний мог вращаться. Процесс шифрования заключается в нахождении буквы открытого текста на внешнем диске и замене ее на букву с внутреннего диска, стоящую под ней. После этого внутренний диск сдвигался на одну позицию и шифрование второй буквы производилось уже по новому шифралфавиту.

Решетка Кардано

Решетка Кардано – инструмент шифрования и дешифрования, представляющий собой специальную прямоугольную (иногда квадратную) таблицу, часть ячеек которой вырезана. В эту таблицу записывалось сообщение, в каждой ячейке оказывалась либо буква, либо их последовательность. Для дешифрации использовался следующий принцип: если указанный адрес был открыт (не вырезан), то узнать содержимое ячейки по нему не составляло труда, в противном случае таблицу следовало повернуть на 90 градусов и так далее, пока не попадалась открытая ячейка.

Таблица Виженера

Данный способ шифрования представляет собой универсальную процедуру с ключевым шифрованием. Таблица имеет следующий вид:

	А	Б	В	Г	Д	Е	Ё	Ж	**З**	…	Я
	Я	А	Б	В	Г	…	…	…	…	…	…
	Ю	Я	А	Б	В	…	…	…	…	…	…
	…	…	…	…	…	…	…	…	…	…	…
→	**М**	…	…	…	**…**	…	…	…	**Ф**	…	…
	…	…	…	…	…	…	…	…	…	…	…
	Я	А	Б	…	…	…	…	…	…	…	Ю

Пример:

ЗАЧЕТ – ключевое слово.

МАМА_МЫЛА_РАМУ – сообщение, которое надо закодировать

Ключевое слово записывается столько раз подряд без пробелов, сколько букв в кодируемом сообщении. После этого, по таблице определяется пересечение столбцов, в основаниях которых стоят соответствующие буквы ключевого слова и исходного сообщения. Таким образом, первая буква шифротекста будет Ф.

Платформы шифрования

Основные понятия

Шифрование – преобразование текста с целью сокрытия его содержания. ***Криптология*** – наука, состоящая из двух ветвей: криптографии и криптоанализа. ***Криптоанализ*** – наука о методах и способах вскрытия шифров. ***Криптосистема*** – система шифрования со сменным элементом. Они бывают двух типов – с *открытым ключом,* если ключ шифрования известен всем и с *закрытым ключом,* в случае, если ключи шифрования и дешифрования надо держать в секрете. Второе их название – асимметричные и симметричные системы соответственно.

Шифруемый текст мы будем называть ***исходным текстом*** (plaintext), а результат шифрования – ***шифротекстом*** (ciphertext).

Основное правило шифрования: исходный текст должен однозначно восстанавливаться по шифротексту.

Криптостойкость– способность криптосистемы формировать шифровки, трудные для взлома.

Алфавит

Алфавит в криптографии носит более широкий смысл, чем в языках, он представляет собой знаковую систему, включающую в себя буквы, цифры и спецсимволы. *Текст* – фрагмент, написанный при помощи алфавита. *Единица текста* – последовательность k знаков (k = 3 – триграф, k = n – n-граф). Тексты часто разбивают на

блоки. Блок – конечная последовательность единиц текста. Обычно длина блока фиксирована.

Оцифровка

Одним из самых простых способов шифрования является простая замена. Оцифровка заключается в замещении каждой буквы алфавита на цифру, также допускается замена буквы на букву. Основное требование – разные буквы не должны заменяться одинаковыми. Это необходимо для того, чтобы можно было однозначно декодировать шифротекст.

Пример:

b&α^589 _ - исходный алфавит

0 1 2 3 4 5 6 7 – результат оцифровки

Платформы шифрования и хэш-функция

Платформа шифрования — это система, используемая при шифровании для записи шифровки. Платформы у исходного и конечного текстов могут быть различными.

Возможности:

- совокупность бинарных последовательностей
- кольцо вычетов
- конечное поле
- эллиптическая кривая

В последнее время получило интенсивное развитие новое направление криптографии – *алгебраическая криптография* – криптография, основанная на теории групп.

Хэш-функция – функция, переводящая произвольную конечную бинарную последовательность в последовательность определенной длины. Она должна быть односторонней.

Теория оптимального кодирования

Алфавитное кодирование

A = {a_1; a_2 ; … ; a_n} – исходный алфавит

B = {b_1; b_2 ; … ; b_n} – кодирующий алфавит

A* - множество всех конечных слов в исходном алфавите

B* - множество всех конечных слов в кодирующем алфавите

ϕ: A → B* - частный случай

$\bar{\phi}$: A* → B* - общий случай

! i $\phi(a_i) = B_i$

Слова кодируются побуквенно. $\phi(a_{i1}; a_{i2} ; \dots ; a_{in}) = B_{i1}, B_{i2}, \dots B_{in}$ Множество $\{B_{i1}, B_{i2}, \dots B_{in}\}$ называется множеством кодовых слов. Все кодовые слова в коде различны и существует однозначность восстановления по кодовому слову буквы исходного алфавита.

Пример:

01 ↔ A

11 ↔ B

*1 – однозначности нет

Определение: R: A* → B*. Отношение R будет взаимно-однозначным, если для любых слов A_1 и A_2: $\forall \bar{a}_1 \in A^*, \bar{a}_2 \in A^* \Rightarrow \phi(\bar{a}_1) \neq \phi(\bar{a}_2)$

Определение: алфавитный код называется равномерным, если $\forall i,j\ \phi(\bar{a}_{ij}) \| = \| \phi(\bar{a}_{ji})$.

Теорема: Любой равномерный код является взаимно-однозначным.

Доказательство: если все кодовые слова имеют одинаковую длину m и слово $\bar{b} = \phi(\bar{a})$ - результат преобразования слова $\bar{a}$, следовательно, слово b может быть единственным образом разбито на кодовые слова, которые имеют длину m, по которым устанавливаются исходные слова.

Определение: код называется префиксным, если никакое кодовое слово не являются началом другого кодового слова.

Теорема: любой префиксный код является взаимно-однозначным.

Доказательство: Если $\bar{b} = \phi(\bar{a})$ и первое кодовое слово соответствует первой букве слова a, которое может быть выделено двумя различными способами, то при этом бы получилось кодовое слово, которое бы являлось началом другого. (Первое кодовое слово в $\bar{b}$ выделяется однозначно, после этого процедура применяется по индукции к другим кодовым словам.)

Определение: код называется суффиксным (постфиксным), если никакое слово не является концом другого.

Теорема: Любой суффиксный код является взаимно-однозначным.

Алгоритм распознавания взаимной однозначности алфавитного кодирования. Теорема Маркова. Неравенство Макмиллана.

Будем называть непустое начало слова собственным префиксом, если это начало не тождественно самому слову (собственному суффиксу). Если $\{\beta_1, \beta_2, \dots, \beta_p\}$ – множество всех слов, каждое из которых является собственным префиксом некоторого кодового слова и является одновременно собственным суффиксом некоторого кодового слова. $\{B_1, B_2, \dots, B_n\}$ – множество кодовых слов. Тогда может быть построен орграф с p+1 вершиной. Вершинами этого орграфа являются все слова $\beta_1, \beta_2, \dots, \beta_n$, но добавляется пустое слово λ ($\{\beta_1, \beta_2, \dots, \beta_p\} + \lambda$). Провести ребро из β_i в β_j можно только тогда, когда существует кодовое слово B_k, где $B_k = \beta_i D \beta_j$; (D =λ, либо набору слов из β). Если i = 0 или j = 0 потребуем, чтобы D не совпадало с λ, а если i = 0 и j = 0, то требуется, чтобы D могло быть разбито на 2 или более последовательно записанных кодовых слов.

$\phi: a_i \rightarrow b_i \quad i = \overline{1;r}$

$G(\phi)$ – орграф

Кодирование ϕ будет взаимно-однозначным, если в G нет ориентированных циклов, проходящих через пустую вершину, в том числе, если отсутствуют петли в этой вершине.

Доказательство: Пусть в G(ϕ) существует такой цикл. Если это петля $\beta_0 D_0 \beta_0$ и при этом является единственным кодовым словом, то D_0 распадается на две или более последовательностей, то есть декодируется неоднозначно. Пусть в G(ϕ) нет петли на нулевом элементе, но существует цикл $\beta_0\beta_1\beta_2 \dots \beta_k\beta_0$ ($k \geq 1$, $\beta_1 \neq \beta_0$), то есть по построению G(ϕ) существуют слова $D_0, D_1, \dots, D_k$, или пустые, или распространяющиеся на несколько кодовых слов, таких, что $!i : \beta_i D_i \beta_{i+1}$ – является кодовым словом. То же самое справедливо для $\beta_k D_k \beta_0$. Покажем, что соотношение $\beta_0 D_0 \beta_1 D_1 \beta_2 \dots \beta_n D_n \beta_0$ может быть декодировано двумя различными способами:

- n – нечетно: $\beta_0 D_0 \beta_1 D_1 \beta_2 D_2 \dots D_{n-1}\beta_n D_n D_0 \beta_1 D_1 \beta_2 D_2 \beta_3 D_3 \beta_4 \dots D_{n-1}\beta_n D_n \beta_0$ в обоих случаях каждая часть может быть пуста или является кодовым словом/распадается на несколько кодовых слов. Если части последних двух типов разбить на слова, то получим два различных разбиения, так как во втором первое кодовое слово является частью D_0. В первом разбиении первое кодовое слово будет равно $\beta_0 D_0 \beta_1$ ($\beta_1 \neq \lambda$).
- n – четное (докажите самостоятельно)

Пусть теперь кодирование не взаимно-однозначное, тогда покажем, что в G(ϕ) существует цикл, проходящий через пустую вершину. Так как кодирование не взаимно-однозначное, то существует слово, которое может быть декодировано двояко. Рассмотрим все точки, в которых разбиение разделяет два слова, то есть существуют слова, участвующие и в первом и во втором разбиении. Рассмотрим при этом более короткое слово, оно опять допускает различное декодирование и так далее пока не получим слово $C \in B^*$ которое может быть разбито на кодовые слова так, чтобы разбиения не имели общих точек. Возможно, что в одном разбиении слово C рассматривается как единое кодовое слово, а в другом на два или более. Это означает, что в G(ϕ) есть петля пустой вершины, следовательно, существует цикл. Пусть теперь оба разбиения разделяют слово C на два или более кодовых слова. Рассмотрим все точки, в которых эти разбиения разделяют слово C. Каждая точка принадлежит только одному из этих разбиений. Слово располагающееся между двумя последовательными точками разбиения является кодовым словом. Рассмотрим все части слова C, которые заключены между двумя последними точками, принадлежащими разным разбиениям. Такая часть является собственным префиксом собственного разбиения и суффиксом другого кодового слова, то есть является вершиной графа G(ϕ). Обозначим такие части слова $\beta_1\beta_2 \dots \beta_l$, $l \geq 1$, $C = D_0 \beta_1 D_1 \beta_2 \dots D_l\beta_l$, где каждое D либо пустое, либо является ключевым словом. Кроме того, для любого

ключевого слова $\beta_i D_i \beta_{i+1}$ ограничено двумя последовательными точками одного разбиения. Тогда по построению графа G(ϕ) в нем существует ориентированный путь, проходящий последовательно по вершинам $\beta_0\beta_1 \ldots \beta_l\beta_0$. Если встречаются одинаковые вершины, то их можно исключить, но в итоге опять получим путь, проходящий через пустую вершину. Проделав это несколько раз, получим путь, проходящий через пустую вершину, но с отсутствием повторяющихся вершин, то есть ориентированный цикл.

Замечание: так как петли в вершинах, отличающихся от нулевой, не влияют на наличие исходного ориентированного цикла, то их можно и не строить в G(ϕ).

Теорема Маркова:

$\phi : a_i \rightarrow B_i \quad i = \overline{1;r}$

$l_i = \mu(B_i)$ – длина $L = \sum_{n=1}^{r}(l_i)$

Рассмотрим все возможные произведения вида $B_j = C_1 B_{j1} B_{j2} \ldots B_{jk} C_n$, где B_j – кодовые слова. $C_1 \ldots C_n$ – могут быть пустыми.

Пусть W = max(k), то есть максимальное число кодовых слов, которое помещается внутри кодовой посылки. Тогда, если ϕ - не взаимно однозначно, то существуют a’ и a’’ ∈A*, такие что ϕ(a’) = ϕ(a’’), результаты их кодирования совпадают и

$\mu(a') \leq [\frac{(W+1)*(L-r+2)}{2}]$

$\mu(a'') \leq [\frac{(W+1)*(L-r+2)}{2}]$

Доказательство: пусть ϕ не является взаимно однозначным, тогда существует орцикл, проходящий через пустую вершину, а также цикл в орграфе, который описывает отображение ϕ и проходит через ту же вершину, и среди них нет одинаковых или равных β_0.

Используя β–цикл построим слово $C = \beta_0 D_0 \beta_1 D_1 \beta_2 \ldots \beta_n \ldots D_n \beta_0$. Оценим длины μ(a’) = ?, μ(a’’) = ? Построим $\beta_1 D_1 \beta_2 D_2 \beta_3$ два разбиения:

1) $\beta_0 D_0 \ldots$
2) $\lambda_0 \beta_1 D_1 \beta_2 D_2 \beta_3 \ldots$

По условию теоремы $\forall D_i \quad \beta_{i+1}$ разбивается не более чем на W кодовых слов, таким образом, две последних части обоих разбиений дают не более чем W +1 кодовых слов.

$\forall \beta_i$ кроме β_0 является по построению собственным префиксом некоторого кодового слова. β_i имеет длину l_i, число собственных префиксов $l_i - 1$, а у всех кодовых слов по ϕ число различных собственных префиксов не превосходит $\sum_{i=1}^{r}(l_i - 1) = L - 2$. Так как среди $\beta_1\beta_2 \ldots \beta_n$ нет одинаковых или равных β_0, то $n \leq L - r$, то есть в каждом из рассмотренных разбиений имеется не более $L - r + 1$ частей, которые распадаются не более чем на $\frac{(L-r+2)}{2}$ пар. Последняя пара при четном разбиении состоит из одной части. Так как каждая пара частей дает не более $W + 1$ кодовых слов, то $(W + 1)* \frac{(L-r+2)}{2}$ - верхнее значение числа кодовых слов, что и требовалось доказать (В этом заключается суть теоремы Маркова).

Неравенство Макмиллана

Теорема:

$\phi : a_i \rightarrow B_i i = \overline{1; r}$

$\mu(B_i) = l_i$, тогда, если алфавитное кодирование взаимно однозначно, то $capB = q$.

$$\sum_{i=1}^{r} \frac{1}{q^{li}} \leq 1$$

Доказательство: пусть $\lambda = \sum_{i=1}^{r} \frac{1}{q^{li}}$

$$\lambda^n = \sum_{l1=1}^{n} \sum_{l2=1}^{n} \ldots \ldots \sum_{ln=1}^{n} \frac{1}{q^{li1+li2+\cdots+lin}}$$

$l_{max} = \max(l_j)$

$\sum_{k=1}^{lmax} \frac{Ck}{q^k}$ там где это возможно равно 0

Лемма: $\forall k : C_k \leq q^k$

Доказательство: C_k – число наборов, $C_k \leq q^k$

$C_k \leftrightarrow (i_{j1} \ldots i_{jn})$

$l_{l1} + l_{l2} + \ldots + l_{lk} = k$

В силу взаимной однозначности кодирования различным образом таких наборов получается q^k.

Следствие:

$$X^n = \sum_{k=1}^{Nlmax} \frac{Ck}{q^k} \leq \sum_{k=1}^{Nlmax} n * l_{max}$$

$X \leq (n * l_{max})^{1/n}$

Лемма: $\phi : b_i \rightarrow A^*$

$P_1 \geq P_2 \geq \ldots \geq P_{k-1} \geq P_k P_i = \mu(B_i)$, тогда можно так представить слово в коде ϕ, что получится оптимальное кодирование ϕ', такое, что кодовые слова B'_{k-1} и B_k' будут различаться только в последнем разряде.

Лемма: ϕ - оптимальный код, ϕ: $P_1, P_2 \ldots P_k$. Исходный алфавит не важен, указаны лишь частоты и соответствующие кодовые слова.

ϕ': $p_1, p_2 .. p_{k-1} p' p''$

$B_1, B_2, \ldots, B_{k-1}, B_k, 0, B_{k+1}, \ldots 1$

$p' + p'' = p_k$

Одна из частей разбивается на два компонента. Если один из этих кодов префиксный, то второй при этом тоже префиксный и $C(\phi') = C(\phi) + P_k$

Теорема редукции: пусть $p' + p'' = p_k$ (совпадение частот), тогда если ϕ' оптимальный префиксный код, то и ϕ - оптимальный префиксный код. Если ϕ - оптимальный префиксный код и $p_1 \geq p_2 \geq \ldots \geq p' \geq p''$, то ϕ' тоже оптимальный префиксный код.

Коды с исправлением ошибок. Оценка функций исправления ошибок

Будем рассматривать равномерное кодирование в алфавите B_0 (булевом), в котором длины всех кодовых слов равны n. Будем считать, что в кодовом слове может произойти ошибка (замена 0 на 1), при этом длина кодового слова не изменяется, то есть остается равной n и закодированное сообщение можно однозначно разбить на кодовые слова. После этого возникает задача однозначного декодирования.

Определение: Код называется исправляющим R ошибок, если при наличии в кодовом слове ≤ R ошибок существует возможность установить исходное слово.

Определение: Расстоянием Хэмминга ρ(α, β) между двумя наборами длины n (μ(α) = μ(β) = n) называется число разрядов, в которых эти разряды изменяются.

Определение: шаром разбиения с центром в наборе $\bar{\alpha}$ называется множество всех наборов длины n, расстояние от которых до $\bar{\alpha}$ не превосходит R.

Определение: код с кодовым расстоянием ρ_{min} определяется как расстояние по Хэммингу по требованию: $\rho_{min} = \min(\rho(\alpha, \beta))$.

Теорема: Код К = { $\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n$} исправляет R ошибок, если $\rho_{min}(K) \geq 2*R + 1$.

Доказательство: заметим, что если в кодовом слове может произойти не более чем R ошибок, то из него может получиться любое слово из шара с радиусом R с центром в этом кодовом слове, поэтому код исправляет R ошибок тогда и только тогда, когда шар радиуса R с центром в кодовом слове не пересекается с другим шаром.

Определение: Код обнаруживает R ошибок, если при наличии в нем R ошибок путем замещения можно сказать о факте этих ошибок.

Теорема: Код К = { $\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_m$} обнаруживает R ошибок тогда, когда $\rho_{min}(K) \geq R + 1$.

Оптимальные коды и их свойства

Пусть в алфавите А существует К букв и для каждой из них известны частоты их появления $p_1, \dots, p_n$, $\sum p_i = 1 \ \forall p_j > 0$. Рассмотрим алфавитное кодирование ϕ: $A \rightarrow \{0, 1\} \equiv B^0 = A^*$.

$p_1 - a_1 \rightarrow B_1 - l_1$

$p_2 - a_2 \rightarrow B_2 - l_2$

……..

Исходной букве a_i с частотой p_i соответствует кодовое слово B_i. Если в тексте большой длины N примерно выдержаны частоты, то можно сказать, что каждый символ a_i встречается Np_i раз. Длина кодового текста $\sum_{i=0}^{k}(Np_i) * l_i = N * \sum_{i=0}^{k} p_i * l_i$.

Определение: Пусть для всех букв $a_1, \dots, a_n$ исходного алфавита A заданы частоты. Пусть кодирование осуществляется на B^0, тогда ценой кода ϕ называется функция $C(\phi) = \sum_{i=0}^{k} p_i * l_i$.

Определение: пусть для любого a_i исходного алфавита фиксируется частоты их появления и задано взаимно-однозначное алфавитное кодирование на B^0, тогда оно называется оптимальным если на этом кодировании достигается инфинум избыточности (infC(ϕ)).

Теорема: Любому набору частот алфавитного кодирования ($K \geq 2$) соответствует оптимальный код с ценой не более $[log_2(K)]$.

Теорема: Для любого набора частот алфавитного кодирования $p_1, \dots, p_i$ существует оптимальный префиксный код, возможно не единственный.

Лемма: ϕ - оптимальный код, $\phi \in$ ОПТ(B*), $p_i > p_j$, тогда имеет место быть $l_i \leq l_j$.

Определение: $S_r(n)$ – число точек наборов длины n в гипершаре радиуса R.

Теорема: $S_r(n) = 1 + \binom{n}{1} + \dots + \binom{n}{r}$

Доказательство: центр гипершара радиуса R представляет собой пространство, вокруг которого находятся кодовые слова, отличающиеся от координат центра на одном из булевых компонент и слова, отличающиеся от координат центра гипершара на две координаты, а также слова, отличающиеся на r булевых координат, что и требовалось доказать.

*Определение:*Mr(n) – максимальное число слов длины n, которое может образовывать код, исправляющий R ошибок.

Теорема:

$$\frac{2^n}{S_{2r}(n)} \leq Mr(n) \leq \frac{2^n}{S_r(n)}$$

Доказательство: рассмотрим произвольное кодовое слово $K = \{\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n\}$. Будем считать, что для данного кодового слова исправляется r ошибок (n гипершаров с центрами в $\bar{\alpha}$, которые не пересекаются), следовательно, число всех точек всех шаров не превосходит 2^n.

$$m * S_r(n) < 2^n \Leftrightarrow m \leq \frac{2^n}{S_r(n)} \Rightarrow Mr(n) \leq \frac{2^n}{S_r(n)}$$

Строим код, который исправляет 2r ошибок. Кодовое слово имеет максимальную длину 2r + 1, и для него

$m^* S_{2r}(n) \geq 2^n \Rightarrow Mr(n) \geq S_{2r}(n)$

Код Хэмминга. Численная оценка Mr(n)

Рассмотрим код, исправляющий одну ошибку – замещение в словах длины n. Выберем $K \in N$

$$2^{K-1} \leq n \leq 2^K - 1 \Leftrightarrow \begin{cases} K \leq \lg_2(n+1) \\ K \geq \lg_2(n+1) \end{cases} \Leftrightarrow K = [\lg_2 n+1] = [\lg_2(n+1)]$$

Тогда каждое число от 1 до n может быть представлено в булевом кубе, используя не более чем K разрядов.

Разобьем все числа от 1 до n на K классов.

$D_i = \{m/m = (m_k, m_{k-2}, \dots, m_0);\ m_i = 1\ i = \overline{1,n}\}$

$D_0 = \{1, 3, 5, 7, \dots\}$

$D_1 = \{2, 3, 6, 7, 10, 11, \dots\}$

Кодом Хэмминга порядка n называется множество наборов $\bar{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$, удовлетворяющих системе

$$\begin{cases} \sum \alpha_i = 0 \quad i \in D_0 \\ \sum \alpha_j = 0 \quad j \in D_1 \\ \dots \\ \sum \alpha_t = 0 \quad t \in D_t \end{cases}$$

Теорема: код Хэмминга порядка n содержит 2^{n-k} наборов, где $K = \lfloor \log_2 n \rfloor + 1$

Теорема: справедлива оценка

$$\frac{2^n}{n} \leq Mr(n) \leq \frac{2^n}{n+1}$$

Компоненты, образующие криптосистемы

Шифровальный блокнот

Одноразовый шифровальный блокнот – единственный в теоретическом смысле стойкий метод шифрования. В его основе лежит та же идея, что и в шифре Цезаря.

Пусть открытое сообщение записано с помощью символов расширенного алфавита, состоящего из 33 букв алфавита, 10 знаков препинания {..:;?!()-“} и знака пробела между словами. Число символов расширенного алфавита в русском варианте равно 44. Занумеруем их числами от 0 до 43. Тогда любой передаваемый текст можно рассматривать как последовательность $\{a_n\}$ чисел множества {0,1,2,…,43}.

■ В качестве секретного ключа берётся абсолютно случайная последовательность $\{c_n\}$ из того же множества чисел. Эта последовательность должна иметь ту же длину, что и передаваемый текст. Складывая по модулю 44 число a_n с соответствующим числом c_n ключа (т. е, складывая числа a_n и c_n и беря остаток от целочисленного деления их суммы на 44), получаем новое число b_n, лежащее в промежутке от 0 до 43:

$a_n + c_n = b_n$, $0 \le b_n \le 43$. Вычисления производятся по модулю 44.

Заменяя цифры последовательности $\{b_n\}$ символами расширенного алфавита, получим зашифрованный текст.

Чтобы восстановить открытый текст, надо воспользоваться тем же ключом:

$a_n = b_n - c_n$, $0 \le b_n \le 43$. Вычисления производятся по модулю 44.

После однократного шифрования использованный ключ уничтожается и в дальнейшем больше никогда не применяется. Для того чтобы зашифровать новое сообщение, отправитель должен воспользоваться новым одноразовым блокнотом.

В случае использования одноразового шифровального блокнота потенциально получается ключ бесконечной длины с неограниченным количеством возможных комбинаций.

Методика шифрования с использованием одноразового шифровального блокнота обладает исключительной надёжностью, поскольку существует бесконечно большое число ключей, с помощью которых из зашифрованного сообщения можно получить

осмысленный текст (или правдоподобную информацию). При этом до тех пор, пока взломщик не получит в своё распоряжение копию шифровального блокнота, он не сможет узнать, является ли полученный результат расшифровки в действительности оригинальным сообщением.

При всей своей привлекательности для этого метода существуют определённые ограничения, которые не позволяют применять их в практике передачи информации по компьютерным сетям, и, в частности, в сети Интернет.

Прежде всего, последовательность значений в бесконечном ключе должна быть действительно случайной, а не псевдослучайной. Сформировать действительно случайную последовательность чисел с помощью компьютера невозможно.

Другая проблема – в передаче копии блокнота получателю. Размер ключа (блокнота) должен быть не меньше длины сообщения, которое шифруется с помощью этого ключа. Далее, после использования, ключ должен быть уничтожен. Необходимость уничтожить ключ исключает возможность применения компакт-дисков или цифровых видеодисков.

Еще одна проблема – синхронизация последовательности ключей у отправителя и получателя сообщения.

Алгоритмы генерации псевдослучайных последовательностей. Равномерно распределенные случайные последовательности.

В случае с криптосистемами мы в основном имеем дело с равномерно распределенными случайными последовательностями (РРСП). Случайная последовательность вычетов $x_i \in \mathbb{Z}_n$, i= 1, …, t, … называется равномерно распределенной в случае выполнения следующих условий:

Для любого натурального k и произвольных значений индексов $1 \leq t_1 \leq t_k$ соответствующие случайные величины $x_{t_1}, \ldots, x_{t_k} \in \mathbb{Z}_n$ независимы в совокупности.

Для любого натурального *t* случайная величина $x_t \in \mathbb{Z}_n$ распределена равномерно:

$$P(x_t = m) = \frac{1}{n}$$

Для любого $m \in \mathbb{Z}_n$.

Псевдослучайной называется последовательность $x_i \in \mathbb{Z}_n$, *i*= 1, …, t, …, вычисляемая по известному рекуррентному соотношению и по своим свойствам схожая с РРСП. Последовательности формируются на компьютерах. Так как это подразумевает использование определенных алгоритмов, мы не можем говорить о создании по-настоящему случайных последовательностей, а только о псевдослучайных. Для этого используется несколько видов генераторов и соответствующих им алгоритмов.

Линейный конгруэнтный генератор

$\vec{X}$ конгруэнтен $\vec{Y}$, если $\vec{X} = k * \vec{Y}$

Линейный конгруэнтный генератор подразумевает следующую последовательность:

$x_{t+1} = ax_t + b\{n_r\}$
$x_n \in \mathbb{Z}$
$0 \neq a \in \mathbb{Z}$
a — мультипликатор.

BBS

- Выбирается большое случайное простое число p.

  Элемент q сравним с $3_{\{4\}}$

  n = p*q

  1) Выбирается случайное S, взаимно простое с n, такое, что $1 \leq S \leq n - 1$;
  2) Вычисляется $x_0 = S^2{}_{\{n\}}$

     $x_i = x_{i-1\{n\}}$

     Берется z_i — минимальный бит x_i

     Извлекаем корни $z_i \ldots z_e$ по модулю n

Ленточные криптосистемы

$\delta_{i+1} = f(\delta_i; k)$

$k_i = g(\delta_i; k)$

$c_i = h(k_i; m_i)$

Ленточный ключ k_iгенерируется на основе входной последовательности и ключа. Важна синхронизированность передающих и принимающих частей.

Самосинхронизация

$\delta_i = (\mathrm{C}_{i-t};\ \mathrm{C}_{i-t+1}; \dots; \mathrm{C}_i)$

$K_i = g(\delta_i;\ \ K)$

$C_i = h(K_i;\ \ m_i)$, где

δ_i – щелевая функция

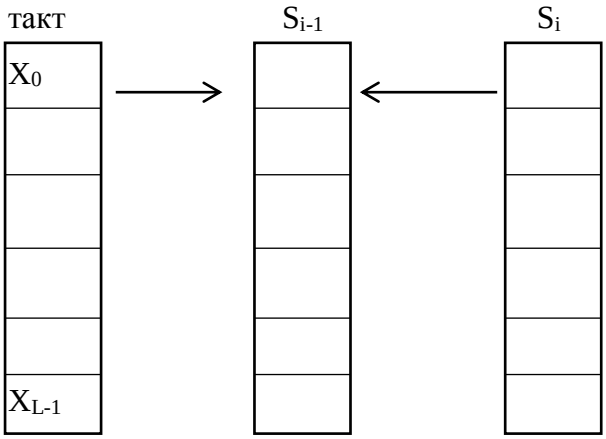
K_i – функция генерации ключа

C_i – функция шифрования

В данной системе дешифровка зависит от ширины щели (забрала), то есть добавление (исключение) некоторой части информации в ходе передачи/приема не будет играть большую роль, через промежуток времени, значительно больший ширины щели.

Линейный регистр сдвига с обратной связью (LFSR)

Пусть число элементов задержки равно *l,* число тактов обработки тоже равно *l.* Каждый из элементов тракта может принимать информацию, то есть хранить 1 ее бит фиксированное время.



Одновременно с тактом осуществляется передача из состояния S_i в S_{i-1}.

$C(D) = 1 + C_1D + \dots + C_2D \in Z_2[D]$ – Шифрование при помощи связующего многочлена на Z_2. Вектор начальных состояний – $[S_{i-1}, S_{i-2}, \dots, S_1, S_0]$. Выполняющаяся последовательность – $S_i = C_1*S_{j-1} + C_2*S_{j-2} + \dots + C_1*S_{j-1\{2\}}$

Пример:

$L = 4$

$[S_3, S_2, S_1, S_0] = [0101]$

$C(D) = 1 + D + D^3 + D^4 \quad C = [1; 0; 1; 1]$

$S_4 = S_3 + S_1 + S_0 = 0 + 0 + 1 = 1$

$S_5 = S_4 + S_2 + S_1 = 1 + 1 + 0 + 0 = 2$

(1, 0, 1, 0, 1, 0, 1, 0, …) – выполняющаяся последовательность имеет период 2.

Определение: Неприводимый многочлен называется примитивным, если D порождает группу $f(D) \in (F_{p^*}, D)$.

Если $C(D) \in Z_2[D]$ и многочлен C(D) неприводим, то для каждого из возможных вариантов начальных состояний порождается выполняющаяся последовательность максимально возможного периода, равного $2^*l - 1$.

Если выполняющаяся последовательность получена с использованием примитивного многочлена, то есть имеют место быть максимальные периоды и при этом длина подпоследовательности k удовлетворяет такому условию:

$1 \le k \le L - 1$,

то в любой, не тождественно равной нулю последовательности длины k, появляется в точности 2^{L-k} раз подпоследовательность $[S_{i-1}, S_{i-2}, \dots, S_1, S_0]$.

Режимы использования блочных шифров: ECB, CBC, OFB, CFB

Стандарты:

- ECB (Electronic Code Book)
  m – исходный текст, он делится на qблоков, то есть $m = m_1, \dots, m_q$.
  $C_i = E_k(m_i)$ – шифротекст. Шифротекст в данном случае не устойчив к атаке (удаление блока).
- CBC (Cipher Block Chaining)
  Процедура обработки предполагает наличие фиктивного блока $\tilde{m}$, такого что
  $C_1 = E_k(m_1 \oplus \tilde{m})$
  $C_2 = E_k(m_2 \oplus C_1)$
  …
  $C_q = E_k(m_q \oplus C_{q-1})$
  Ошибка в j-м блоке распространяется на все остальные блоки
- OFB (Output Feed Back)
  Полностью адаптирован к поточному шифрованию.
  n – размерность блока.
  $1 \le j \le n$
  Создается поток ключей мощностью j бит за 1 такт. Исходное деление блока перебивается на блок размером в j бит. $x_1 = \tilde{m}_1$
  $y_1 = E_k(x_1)$
  $y_2 = E_k(x_2)$
  …
  $C_i = m_i \oplus l_i$ l_i – j крайних слева бит для y_j, при чем$x_{i+1} = y_i$
- CFB (Cipher Feed Back)
  $y_0 = \tilde{m}_1$
  $z_i = E_k(y_{i-1})$
  c_i = j - j крайних слева бит блока z_i
  $y_i = m_1 \oplus l_i$

■ Вероятностная модель системы шифрования. Теорема Шеннона

M – множество исходных текстов

C – множество всех шифротекстов

Л – множество всех ключей

E_k: M → C

D_k: C → M

$E_k * D_k = I_m$

$D_k * E_k = I_c$ (биекция)

E(m, k): MxK → C
D(c, m): C x K → M
D(E(m, k), k) = m
E(D(c, k), k) = c

Совершенная система шифрования – система, где знание шифрованного текста не меняет вероятностного пространства исходного текста.

p(m, c) = p(c) * p(m, k) = p(m)*p(c/m)

По теореме Байеса:

$M = z_n{}^t n \geq 2*t \geq 1 \quad \Rightarrow \forall m \in M p(m) = \frac{1}{n^t}$, тогда

$E_k(m) = m \oplus k_{\{n\}}$

$D_k(c) = C + k_{\{n\}}$ порождает равнораспределенную систему

$p(m/c) = p(m) = \frac{1}{n^t}$

Теорема Шеннона:

CapM = CapC = CapK = n

Регулярная система совершенна тогда и только тогда, когда:

- $\forall m \in M; c \in C \exists 1\ k \in KE(m, k) = c$
- РРСВ (распределение вероятностей на пространстве ключей равномерно)

Правило Керкгоффса. Мнемоническое определение стойкости системы

Принцип Керкгоффса— правило разработки криптографических систем, согласно которому в засекреченном виде держится только определённый набор параметров алгоритма, называемый ключом, а сам алгоритм шифрования должен быть открытым. Другими словами, криптосистема должна оставаться безопасной даже в том случае, когда злоумышленнику известно всё, кроме применяемых ключей.

Стойкость криптосистемы определяется степенью безопасности использования в ней ключевых материалов, в то время как все долговременные процедуры и элементы системы неизбежно определяются.

Основные понятия об однонаправленных функциях

Однонаправленной называется такая функция f, для которой легко определить значение функции y = f(x), но практически невозможно отыскать для заданногоу такое x, что y = f(x).

Для построения криптографических систем защиты информации чаще используются однонаправленные функции, для которых обратное преобразование существует и однозначно, но вычислительно нереализуемо. Они называются вычислительно необратимыми функциями.

■ *Пример*: в качестве примера однонаправленной функции y = f(x) рассмотрим широко известную функцию дискретного возведения в степень – $y = a^x_{\{p\}}$, где x – целое число в диапазоне от 1 до p – 1 включительно, а вычисление производится по модулю p, где p – очень большое простое число, a – целое число ($1<a<p$). Функция $y = a^x_{\{p\}}$ вычисляется относительно просто, а обратная к ней функция $x = \log_y p$ является вычислительно сложной. Задачу нахождения такого x называют задачей дискретного логарифмирования.

$f(x) = x^3 + ax + b$

a) charF $\neq 2$

$f'(x) = 3x^2 + a$

$D = 4a^3 + 27b^2 \neq 0$. Исследуется эллиптическая кривая E(x, y)

$y^2 = x^3 + ax + b = f(x)$

b) charF = 2

$y^2 + xy = x^3 + ax + b$

c) charF = 3

$y^2 = x^3 + ax^2 + bx + c$ $\{a, b, c\} \in F$

Однонаправленные функции с потайным ходом

Однонаправленная функция с потайным ходом (trap door function, сокращенно — TDF) — это однонаправленная функция f из множества X в множество Y, обладающая свойством (потайным ходом, лазейкой), благодаря которому функцию можно обратить — то есть найти для любого $y \in Imf, x \in X$ такое, что $f(x) = y$.

Рассмотрим применение однонаправленной функции с потайным ходом на примере криптосистемы RSA. Возьмем число n = pq, где pи q — простые числа. Считаем, что для данного nнахождение pи qявляется математически трудной задачей. Функция шифрования RSA — $E(m) = m^e{}_{\{n\}}$, где e — взаимно простое с (p — 1)(q — 1). Числа pи q считаются потайным ходом, с помощью которого можно найти функцию, обратную E. На сегодняшний день лучшей атакой на функцию RSA считается факторизация n.
Однонаправленные функции применяются в различных криптосистемах — например, в протоколе Эль-Гамаля.

Основные методы шифрования

Шифрование методом перестановки

Перестановка осуществляется записью и чтением шифра по разным путям геометрических фигур, например, запись исходного текста по строкам, а чтение – по столбцам матрицы. Ключ – это блок из определенного числа символов.

1	И	Е	-	П
2	Е	Р	Е	С
3	О	В	А	Н
4	Т	А	Н	О
5	Ш	И	Ф	Р
6	В	К	О	И
К1/К2	1	2	3	4

Запись по строкам осуществляется по ключу К1, а чтение по столбцам по ключу К2. Для перестановки может быть использован граф. Если длина текста не кратна числу элементов графа, то пишется произвольный символ. Выборка из таблицы дешифрования может быть как последовательная, так и в порядке, задаваемом ключом.

Пример:

ПСНОРЙЕРВАИК_ЕАНФОИЕОТШВ

Ответ: ШИФРОВАНИЕ ПЕРЕСТАНОВКОЙ

Шифр Цезаря

Шифр Цезаря, также известный как шифр сдвига или код Цезаря — один из самых простых и наиболее известных методов шифрования. Этот шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите.

Пример:

Исходный алфавит —А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я

Шифрованный —Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я А Б В

Исходный текст —Съешь же ещё этих мягких французских булок, да выпей чаю

Шифротекст —Фэзыя йз зьи ахлш пвёнлш чугрщцкфнлш дцосн, жг еютзм ъгб

Аффинная система шифрования

■ Пусть шифруется оцифрованный текст, единицами которого являются вычеты по модулю n, то есть платформой шифрования является кольцо вычетов Z_n. Единицами шифрованного текста также служат элементы кольца Z_n.

Пусть $m \in Z_n$ – произвольная единица исходного текста, соответствующая ей единица шифрованного текста – $c \in Z_n$ вычисляется по правилу:

$c = am + b_{\{n\}}$, где $a,b \in Z_n$ – некоторый параметры. Пару $e = (a, b)$ можно считать ключом дешифрования. Параметры ключа не могут быть произвольными. Для обращения функции шифрования

$E_e(m) = am + b_{\{n\}}$

необходима обратная запись $a_{\{n\}}$, тогда

$m = a^{-1}c – a^{-1}b_{\{n\}}$

$\exists a^{-1}_{\{n\}} \Leftrightarrow$ НОД(a; n) = 1 (существование $a^{-1}_{\{n\}}$ равносильно взаимной простоте a и n)

На параметр b ограничений не накладывается, он может быть любым.

Дешифрование осуществляется по той же схеме, что и шифрование, при этом ключ дешифрования равен

$d = (a^{-1}; -a^{-1}b)$

Криптостойкость данного шифра невелика, так как знание двух соответствующих друг другу пар единиц (m_1, c_1) и (m_2, c_2), позволяет решить систему из двух уравнений

$$\begin{cases} C_1 = am_1 + b_{\{n\}} \\ C_2 = am_2 + b_{\{n\}} \end{cases}$$

Решение приводит к такой закономерности:

$C_1 - C_2 = a(m_1 - m_2)_{\{n\}}$

$НОД(m_1 - m_2; n) = 1 \quad \Rightarrow \quad a = (c_1 - c_2)*(m_1 - m_2)^{-1}_{\{n\}}$

Единственность решения не гарантируется, чтобы ее достичь, необходимо брать переопределенную систему:

$$\begin{cases} C_3 = am_3 + b_{\{n\}} \\ C_2 = am_2 + b_{\{n\}} \end{cases}$$

Решение будет лежать на пересечении множеств решений исходной и переопределенной систем.

Гаммирование

Гаммирование — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных. Суммирование, обычно, выполняется в каком-либо конечном поле.

Шифр Вернама

Шифр Вернама является разновидностью криптосистемы одноразовых блокнотов. В нём используется булева функция «Исключающее ИЛИ». Шифр Вернама является примером системы с абсолютной криптографической стойкостью. При этом он считается одной из простейших криптосистем.

Криптосистема была предложена для шифрования телеграфных сообщений, которые представляли собой бинарные тексты, в которых открытый текст представляется в коде Бодо (в виде пятизначных «импульсных комбинаций»). В этом коде, например, буква «А» имела вид (11000). На бумажной ленте цифре «1» соответствовало отверстие, а цифре «0» — его отсутствие. Секретный ключ должен был представлять собой хаотичный набор букв того же самого алфавита.

Для получения шифротекста открытый текст объединяется операцией «исключающее ИЛИ» с секретным ключом. Так, например, при применении ключа (11101) на букву «А» (11000) получаем зашифрованное сообщение (00101): $(11000) \oplus (11101) = (00101)$. Зная, что для принимаемого сообщения имеем ключ (11101), легко получить исходное сообщение той же операцией: $(00101) \oplus (11101) = (11000)$.

Для абсолютной криптографической стойкости ключ должен обладать тремя критически важными свойствами:

1 Иметь случайное равномерное распределение: $P_k(\mathrm{k}) = 1/2^N$, где k — ключ, а N — количество бинарных символов в ключе;
2 Совпадать по размеру с заданным открытым текстом;
3 Применяться только один раз.

Также хорошо известен так называемый **шифр Вернама по модулю m**, в котором знаки открытого текста, шифрованного текста и ключа принимают значения из кольца вычетов Z_m. Шифр является обобщением оригинального шифра Вернама, где m = 2.

Шифр Виженера. Тест Казисского. Алгоритм вычисления длины ключа

Шифр Виженера базируется на принципе гаммирования. Для шифровки производится сложение исходного текста с определенным модулем и ключа, в роли которого выступает исходный текст. Шифр получается как повторяющаяся комбинация сдвигов.

Тест Казисского — эффективный способ определения длины ключа и сдвигов в ключе. Пусть *ci* — индекс косовпадения данного текста. Рассматривая текст *m*, соответствующий алфавиту из *n* букв. Пусть l=|m| — длина текста.l_i — число вхождений буквы с номером *i* в текст *m*. Тогда индекс косовпадения:

$$ci = \left(\frac{l_1}{l}\right)^2 + \left(\frac{l_2}{l}\right)^2 + \cdots + \left(\frac{l_n}{l}\right)^2$$

Чем «осмысленнее» текст, тем выше его индекс косовпадения. Это помогает вычислить длину ключа в тексте Виженера.

Рассмотрим алгоритм вычисления длины ключа шифра Виженера. Для примера возьмем роман «Моби Дик». Известно, что его индекс косовпадения равен приблизительно 0,065 (в тексте присутствуют только 26 букв английского алфавита).

Пусть m = m1m2m3… — исходный текст с i-тыми буквами, а c = c1c2c3… — шифровка по Виженеру.

Если используем обычный сдвиг, то есть длина ключа kравна 1, то ci(m) = ci(c), так как изменяются только номера букв, но не числа их вхождений. Предполагаем, что m — осмысленный текст, поэтому сiдолжен соответствовать стандартному значению для данного языка. В нашем примере — английский, поэтому ci(c) = 0,065. Вряд ли в общем случае мы столкнемся с ключом длины 1, поэтому последовательно вычисляем индексы косовпадений.

$ci = (c1c2c3 \ldots) = d1$

$ci = (c1c3c5 \ldots) = d2$

$ci = (c1c4c7 \ldots) = d3$

…

$ci = (c_1 c_{1+t} c_{1+2t} \ldots) = dt$

Покаdt не станет приблизительно равно 0,065. Тогда длина ключа может быть равна t.

Роторный шифр

В роторном шифре в качестве повторяющегося ключа рассматривается комбинация произвольных замен b_1, b_2, b_3, … , b_t с соответствующими изменениями в преобразовании элементов текста. При взломе такого шифра нужно также сначала определить длину ключа. Для этого можно использовать тест Казисского. Далее для определения замен можно использовать частотный анализ.

Криптосистема Хилла

Шифр Хилла — обобщение аффинного шифрования. Единицами исходного и шифрованного текста являются элементы кольца вычетов $\mathbb{Z}_n$. Текст разбивается на блоки одинаковой ширины *l*. Шифрование осуществляется по правилу:

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_l \end{pmatrix} = A \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_l \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{pmatrix}$$

Где *m* — блок исходного текста, *c* — блок шифрованного текста, A — матрица порядка *l*x*l* с элементами из кольца вычетов, а *b*— вектор с этими элементами.

Для однозначности восстановления шифротекста достаточно, чтобы Абыла обратима над кольцом вычетов. Существование обратной матрицы A^{-1} равносильно тому, что ее определитель обратимmod(n), то есть *НОД (delA, n) = 1*. Тогда дешифрование проходит по формуле *m* = $A^{-1}(c)$— A^{-1} (*b*).

RSA

Криптосистема RSAназвана в честь ее создателей — Ривеста, Шамира и Эйдлмена. Алгоритм шифрования RSA — первый алгоритм шифрования с открытым ключом. Так называют алгоритмы, в которых из двух ключей — шифрования и дешифрования — секретным является только ключ дешифрования.

■ Установка криптосистемы RSA проходит в несколько этапов. Выбираются два достаточно больших простых числа p и q, они секретны. Вычисляется модуль n = pq, вычисляется функция Эйлера f(n) = (p – 1)(q – 1). Выбирается такое число e, что НОД

(f(n), e) = 1. Это число — открытый ключ шифрования. Значение функции держится в тайне. Взаимная простота позволяет вычислить d такое, что e*d = 1(modf(n)). Значение d — секретный ключ дешифрования.

Единицами исходного и шифрованного текста являются элементы кольца вычетов $\mathbb{Z}_n$, это платформа шифрования. Единица шифротекста c получается из исходного текста mпо правилу: $c = m^e{}_{\{n\}}$.

Для дешифрования с достаточно знать значение d. Дешифрование происходит следующим образом: $m = m^{ed}{}_{\{n\}}$.

Шифр замены

Наиболее известными и часто используемыми шифрами являются шифры замены. Они характеризуются тем, что отдельные части сообщения заменяются на какие-либо другие буквы, числа, символы и т.д. При этом замена осуществляется так, чтобы потом по шифрованному сообщению можно было однозначно восстановить передаваемое сообщение.

Пусть, например, зашифровывается сообщение на русском языке и при этом замене подлежит каждая буква сообщения. Формально в этом случае шифр замены можно описать следующим образом. Для каждой буквы α исходного алфавита строится некоторое множество символов M_α так, что множества M_α и M_β попарно не пересекаются при$\alpha \neq \beta$, то есть любые два различные множества не содержат одинаковых элементов. Множество M_αназывается множеством шифробозначений для буквы α.

Электронная подпись

Электронные (цифровые) подписи удостоверяют подлинность электронных документов. Главным отличием электронной подписи от обычной является ее зависимость от подписываемого документа.

Основные требования, предъявляемые к электронным подписям:

- *Проверяемость*
- *Неотклоняемость* (должна быть предусмотрена возможность доказательства того, что подпись поставлена именно лицом, выпустившим этот документ)

- *Уникальность* (подпись должна быть присуща одному лицу, ее подделка должна быть трудной задачей)
- *Неотъемлемая часть документа* (это означает невозможность перенесения подписи на другой документ)
- *Защита целостность документа* (его содержание не может быть изменено)

Электронные подписи бывают двух видов:

- added (Добавленные)
- recover (Восстанавливающие)

В системе, предусматривающей электронную подпись, должны быть выделены следующие элементы:

- M – пространство документов
- M_s – пространство дайджестов документов
- S – пространство подписей
- R: M → M_s – инъективная функция, сопоставляющая документу его дайджест
- R^{-1}: ImR → M – обратная к R функция
- h: M → M_s – односторонняя функция

Чаще всего подпись ставят не в пространстве документов, а в пространстве дайджестов.

Электронная подпись на базе RSA

Каждый пользователь A некоторой информационной системы устанавливает систему шифрования RSA, выбирая открытые данные (n_A, e_A), где n_A – модуль системы, e_A – открытый ключ шифрования, и вычисляет секретный ключ дешифрования d_A, известный только пользователю A: $e_A d_A = 1_{\{\phi(nA)\}}$. Для пользователя A определяются соответствующие пространства $M = M_s = S = Z_n$ и функция R: $Z_n \rightarrow Z_n$.

Алгоритм подписи:

1) Имеется документ $m \in Z_n$

2) Вычисляется дайджест $\bar{m} = R(m) \in Z_n$

3) Вычисляется подпись $s = \bar{m}^{dA}{}_{\{nA\}}$

Алгоритм проверки подписи:

1) Берутся открытые данные (n_A, e_A)

2) Вычисляются $s^{eA} = \bar{m}^{dAeA} = \bar{m}_{\{nA\}}$

3) Восстанавливается $m = R^{-1}(\bar{m})$

Поставить подпись может только А, так как никто другой не может вычислить секретный ключ d_A. Процедура постановки подписи противоположна процедуре шифрования.

■ *Пример:*

Пусть p = 11, q = 7, тогда n = 77, ϕ(n) = 60.

Пусть e = 13, тогда d = 37. Предположим, что m = 2, $\bar{m}$ = R(m) = m + 1 = 3. Подписью для$\bar{m}$ будет $\bar{m}^d = 3^{37} = 31_{\{77\}}$.

Проверка: $\bar{m} = 31^{13} = 3_{\{77\}}$, m = 3 – 1 = 2

Подпись на зашифрованном документе:

Допустим, что пользователь А хочет не только снабдить документ m подписью s, но и зашифровать сообщение с использованием открытых данных (n_B, e_B) другого пользователя В. Имеются две возможности – сначала зашифровать, затем подписать или сделать все то же самое но в обратном порядке.

Подпись в зашифрованном документе $s = (R(m^{eB})_{\{nB\}})^{dA}_{\{nA\}}$ имеет слабость, которую обычно стараются не допускать. Третье лицо, обозначим его С, может вначале «снять» подпись абонента А, используя открытый ключ e_A:

$s^{eA} = (R(m^{eB})_{\{nB\}})^{dAeA}_{\{nA\}} = R(m^{eB})_{\{nB\}\{nA\}}$, а затем заменить ее своей подписью

$s_1 = (R(m^{eB})_{\{nB\}})^{dC}_{\{nC\}}$

Таким образом можно поставить свою подпись на чужой документ.

Недостатки второй схемы, когда документ вначале подписывается, а затем шифруется, не столь очевидны. Продемонстрируем их на примере:

Пример:

Пусть $n_A = p_A q_A = 8387*7449 = 62894113$, $e_A = 5$, $d_A = 37726937$, $n_B = p_B q_B = 55465219$, $e_B = 5$, $d_B = 44360237$.

Пусть R≡ 1, m = 1368797. Тогда

1) $s = m^{dA} = 59847900_{\{nA\}}$

2) $c = s^{eB} = 38842235_{\{nB\}}$

Пользователь B проверяет правильность подписи:

1) $\bar{s} = c^{dB} = 4382681_{\{nB\}}$

2) $\bar{m} = \bar{s}^{eA} = 543835568_{\{nA\}}$

Обнаруживается, что m ≠ m, не смотря на то, что все делалось по протоколу. Причиной этого может быть то, что $n_A > s > n_B$. Шифровать можно только те сообщения, которые не превосходят собственного модуля.

Правило: для передачи подписи и для передачи текста должны быть использованы два различных модуля криптосистемы.

Криптосистемы, использующие дискретное логарифмирование

Дискретный логарифм

Перед тем, как мы сформулируем задачу дискретного логарифмирования, напомним об однонаправленных функциях, которые играют особую роль в криптографии.

Однонаправленной называется такая функция f, для которой легко определить значение функции y = f(x), но практически невозможно отыскать для заданногоутакое x, чтобы y = f(x).

Примером такого рода функций может служить широко известная функция дискретного возведения в степень: $y = \alpha^x{}_{\{p\}}$, где x–целое число в диапазоне от 1 до p– 1 включительно, называемое *дискретным логарифмом*, а p–очень большое простое число; α –целое число, удовлетворяющее условию $1 < \alpha < p$.

Значение функции $y = \alpha^{x}{}_{\{p\}}$вычисляется достаточно просто, а обратная к ней функция $x = \log_y$рявляется вычислительно сложной практически для всех $1<y<p$ при условии, что p и p – 1 имеют большой простой множитель. Такую задачу называют *задачей дискретного логарифмирования*.

Протокол Диффи-Хеллмана

■ F – конечное поле

<q> - порождающий элемент для мультипликативной группы F*

Абоненты A и B осуществляют обмен секретной информацией, их общение начинается с выборки секретным образом случайного большого натурального числа. Переписка между A и B производится по открытому каналу связи без предварительных договоренностей. Выбор конечного поля F и образующего элемента qосуществляется в ходе диалога по открытому каналу, поэтому они являются открытыми данными. Все элементы поля Fперенумерованы.

Процесс разделения ключа осуществляется следующим образом:

1. *Установка*. A и B по открытому каналу связи договариваются о выборе поля F и порождающего элемента q.
2. *Генерация абонентами A и B случайных чисел и передача данных по открытой сети*: абонент A выбирает случайным образом натуральное число x, вычисляет $y = g^x$ и сообщает полученный результат абоненту B. Тот выбирает случайным образом число z, вычисляет $u = g^z$ и сообщает результат абоненту A. Все вычисления производятся в поле F по модулю p. Передаваемые по открытой сети значения u, уобязательно имеют стандартные имена вычетов по модулю p, то есть $0 \le y,u \le p-1$. Если p – составное, то uи увсе равно имеют стандартные имена.
3. *Разделение секретного ключа*. Абонент A, зная x и u, вычисляет элемент $u^x = g^{zx}$ и записывает его стандартным именем. Абонент B, зная y и z, вычисляет $y^z = g^{xz}$, также присваивая ему стандартное имя. В результате, каждый из них узнает элемент g^{xz} и соответствующее ему число q–его номер при стандартной (или какой либо другой) нумерации элементов поля F. Таким образом они «разделили» число q.

Возможные действия несанкционированного пользователя (противника):

- перехват q^x и q^z
- попытка восстановить q^{xz} или q^{zx} приведет к задаче дискретного логарифмирования (нахождение xпо $y = g^x$ или z по $u = g^z$).

Остается открытым вопрос: упрощает ли знание элементов g^xиg^zнахождение элемента g^{xz}.

Существуют и другие протоколы и системы шифрования, использующие понятие дискретного логарифма и основанные на трудности его нахождения. Например, Эль-Гамалю принадлежит идея, которую в общих чертах можно сформулировать так: пусть абоненту Атребуется передать сообщение vабоненту B. Элемент vпринадлежит полю F. Как замечено выше, A и B могут обменяться секретным ключом w, тогда A может передать сообщение в виде v + wили в виде v*w. Абонент Врасшифрует его, либо вычитая w, либо деля на w.

Протокол Эль-Гамаля

1. *Установка.* Абоненты некоторой системы по открытому каналу связи договариваются о выборе большого конечного поля $F = F_{p^r}$ порядка p^r и элемента большого порядка $g \in F^*$. Желательно, чтобы элемент gпорождал мультипликативную группу F^*. Мультипликативная группа F^* вместе с элементами g являются платформой шифрования. Далее, каждый из абонентов, например B, выбирает секретное натуральное число b и вычисляет стандартное имя элемента g^b, которое является открытым для других пользователей системы. Оно позволяет им передавать засекреченные сообщения, прочитать которые может только тот, кто знает значение b, то есть только абонент B. Каких-либо секретных договоренностей между различными абонентами нет.
2. *Генерация случайных чисел и передача данных по сети*. Пусть абонент A хочет передать секретное сообщение $m \in F$ абоненту B. Значение g^b, играющее роль открытого ключа, ему известно. A выбирает случайным образом натуральное число $1 \le k \le p^r - 2$. Послание, передаваемое им по сети имеет вид: (g^k, mg^{bk}).
3. *Расшифровка абонентом B сообщения m*. Абонент B знает секрет b. Сперва он вычисляет $(g^k)^b = g^{bk}$, а затем $m = (mg^{bk})(g^{bk})^{-1}$.

Слабость данного алгоритма заключается в уязвимости перед принципом радиотехнической разведки типа «человек посередине».

Протокол Масси—Омуры

1. *Установка*. Абоненты A и B по открытой сети договариваются о выборе простого числа p и большого конечного поля F = F_{p^r} порядка p^r.
2. *Генерация случайных чисел и передача данных по сети*. У абонента A есть секретное сообщение m, представленное в виде $0 \leq m \leq p^r - 1$. Этому числу соответствует элемент поля с тем же обозначением. Далее он случайным образом выбирает число $0 < x \leq p^r - 2$, взаимно простое с $p^r - 1$ (НОД(x, $p^r - 1$) = 1). Одновременно он вычисляет $X^{-1}_{\{p^r-1\}}$, и вычисляет стандартное имя элемента m^x. Аналогично, абонент B, который должен получить от A секретное сообщение m, выбирает случайное число $0 < z \leq p^r - 2$, взаимно простое с $\leq p^r - 1$ и вычисляет $z^{-1}_{\{p^r-1\}}$. Затем, абонент A передает по сети стандартное имя элемента m^x. Абонент Ввычисляет стандартное имя элемента m^{xz} и посылает его обратно абоненту A, который вычисляет элемент$m^{xzx^{-1}}$ = m^z и посылает его стандартное имя абоненту B.
3. *Расшифровка абонентом B сообщения m*. Абонент B вычисляет элемент$(m^z)^{z^{-1}}$ = m.

Атака на шифрование на основе дискретного логарифма

Пусть F_q – конечное поле порядка $q = p^r$. Пусть $q - 1 = \Pi_p p^{a_p}$ - разложение числа q – 1 в произведение степеней простых чисел.

Если все делители числа q – 1 малы, то примарное число q называется гладким. В этом случае существует достаточно быстрый алгоритм нахождения дискретного логарифма в F_q*, он называется алгоритмом Сильвера-Полига-Хеллмана.

Базовые понятия о квантовых вычислениях

Физический смысл квантовых вычислений. Кубиты

Принципиальное отличие квантовых компьютеров заключается в использовании в их работе принципа суперпозиции. Кубит – наименьший элемент для хранения информации в квантовом компьютере. Как и бит, кубит допускает два собственных состояния – 0 и 1, а также третье состояние – состояние суперпозиции. Принцип суперпозиции для объекта, имеющего фиксированные состояния, заключается в существовании третьего состояния равного сумме вероятностей его нахождения в фиксированных состояниях.

Пример:

Если число фиксированных состояний равно 2, то кубит будет иметь 4 состояния.

3, то кубит будет иметь 8 состояний.

4, то кубит будет иметь 16 состояний.

Ключевым отличием квантового компьютера от обычного является использование квантового принципа запутанности, который позволяет переносить изменение состояния одного из запутанных квантов на другой квант. Благодаря этому, оказывается достижимым сверхплотное кодирование.

Сложность функционирования устройств, использующих память вида кубит, заключается в простоте приведения структуры кубита в некогерентное состояние. Это может быть вызвано изменением температуры. Память на кубитах функционирует при сверхнизких температурах, что значительно сдерживает развитие квантовых компьютеров. Вероятным выходом из этой ситуации может служить использование топологических изоляторов. Топологический изолятор – особый вид материала, который внутри представляет собой диэлектрик, а на поверхности проводит электрический ток. Примерами таких материалов могут служить минерал кавацулит или тонкая пленка ртути. Электромагнитные явления в структурах такого типа изучаются в двухмерной физике. В ходе получения кубита были сделаны масштабные физические открытия, одним из которых является получение частицы, имеющей дробный заряд.

Принцип работы квантового компьютера

Квантовая запутанность. Между двумя квантами может существовать корреляция. Если два кванта коррелированы, то, в соответствии с принципом неопределенности Гейзенберга, внесение изменений на одном из квантов автоматически приводит к разрушению корреляции между этими квантами. Вообще говоря, состояние запутанности может возникнуть между любыми квантами.

Для облегчения понимания темы условимся использовать вместо понятия кванта, понятие некого шара, обладающего неким параметром. В начале раздачи шаров набор параметров у участников А и В является неопределенным. Далее, случайным образом выбирается параметр шара для измерения. Как только участник А определится с параметром, то в случае запутанности шаров, находящихся у участников, такой же параметр появится и у участника В. Проведем аналогию между условными «шарами» и частицами света – фотонами. Свет – это поперечная волна, то есть волна, в которой колебания осуществляются в плоскости, перпендикулярной его распространению. Колебания осуществляются в электромагнитном поле, при этом фотону нельзя задать вопрос: какая у тебя плоскость поляризации? Пусть участники А и В, имеют запутанные фотоны. Если у фотона участника А угол поляризации составляет 45^{o}, то и у фотона участника В будет такой же угол поляризации. Если между А и В появится 3-е лицо, то фотоны у А и В перестанут быть коррелированными.

Для передачи одного бита информации достаточно двух запутанных фотонов. Для передачи на малые расстояния запутанность необязательна, но при передаче на большие расстояния возникает проблема потери фотонов, так на каждые десять метров оптоволоконного кабеля теряется один фотон, следовательно, возникает необходимость установки усилителей. Но возникает проблема, так как для квантового компьютера усилитель неотличим от шпиона. Выход заключается в использовании повторителей, которые синхронизируются с отправителем информации. Устанавливать такие устройства следует не более чем через 50 километров от источника сигнала. Между отдельными квантами повторителями может быть установлена связь сродни телепортации.Состояние квантовой запутанности, транслируемое на внешнюю сторону, называется квантовым катализом.

Двухмодовое сжатое состояние света - состояние света, в котором он не имеет свойств частицы. Оно математически может быть разложено в "фотонное состояние". О

фотоне нельзя сказать, дошел ли он или не дошел до цели, но можно говорить о степени запутанности фотонов - парадигме квантовой теории.

Для квантового компьютера обычным состоянием является коллапс. Любая связь, установившаяся между квантовым процессором и внешним миром, может привести к коллапсу.

Основные достоинства квантового компьютера:

- Квантовый компьютер позволяет работать с гиперименами за счет квантовой запутанности
- Применение принципа неопределенности

Приложение. Необходимые сведения из математического аппарата

Аддитивная абелева группа

Множество A с определенной на нем бинарной операцией называется аддитивной абелевой группой относительно операции сложения.

$\forall$ (a, b, c) $\in$ A :

- a + b = b + a
- (a + b) + c = a + (b + c)
- $\exists$ 0; a + 0 = a

$\forall$ (a) $\in$ A $\exists$ (-a): = (-a) + a = 0

Примеры абелевых групп:

- Множество Z_0 = mod(N) относительно операций модульного сложения.
- Множество всех функций, определенных на заданном подмножестве числовой оси R относительно сложения обычных функций.

Мультипликативная абелева группа

Множество B с элементами a, b, c для которых справедливы ассоциативность и коммутативность, существование единицы и обратного элемента называется мультипликативной абелевой группой.

Примеры:

- Q* = Q \ {0}
- R* = R \ {0}

Порядок группы – мощность множества A (для конечного числа элементов).

Замыкание множества – множество с его предельными точками.

Подмножество $B \subset A$ – множество, уложенное в A со следующими свойствами:

- $\bar{B}_{\oplus}$ замкнута относительно операции сложения. $\forall$ (a, b) $\in$ B / a + b $\in$ B.
- $\forall$ (a) $\in$ B $\exists \bar{a}$ / a + $\bar{a}$ = 0

Пример: $Z \subset Q \subset R$

По умножению:

$\bar{B}_{\oplus}$

$\forall$ (a) $\in$ B / a*b $\in$ B

$\forall$ (a) $\in$ B $\exists \bar{a}$ / a*$\bar{a}$ = ll $\in$ B

Пример:

$\{\pm 1\} \subset O^* \subset R^*$

Циклическая группа, порожденная элементом A, все элементы которой являются произведениями образующего элемента для мультипликативной абелевой группы или являются суммой некоторого числа итераций для аддитивной абелевой группы

<a> = {ka; k $\in$ Z}

<a> = { a^k; k $\in$ Z }

Группы. Циклические группы и их виды

Множество Gсалгебраической операцией * называется группой, если выполняются следующие условия:

- операция * в Gассоциативна: $a * (b * c) = (a * b) * c \forall a,b \in G$
- в Gсуществует нейтральный элемент θ: $a * \theta = \theta * a = a \forall a \in G$
- для каждого элемента $a \in G$ существует обратный ему элемент $a^{-1} \in G$: $a * a^{-1} = a^{-1} * a = 0$

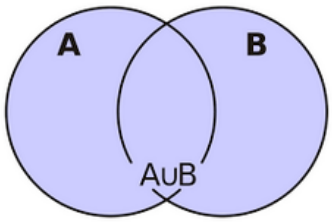
Если операция * коммутативна, то группа называется коммутативной или абелевой. В противном случае группа называется некоммутативной.

Определение: группа Gназывается циклической, если в ней существует порождающий элемент (то есть такой элемент $d \in G$, что G = <d>).

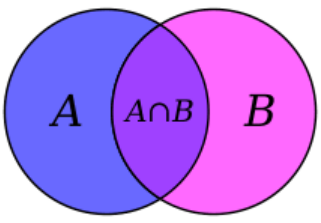
Основные понятия теории множеств

Операции над множествами:

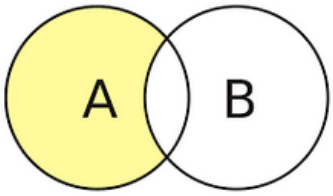
- объединение множеств ($A \cup B$)



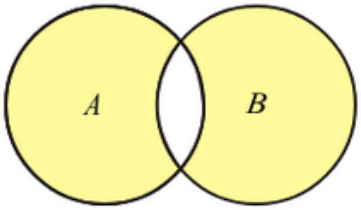
- пересечение множеств ($A \cap B$)



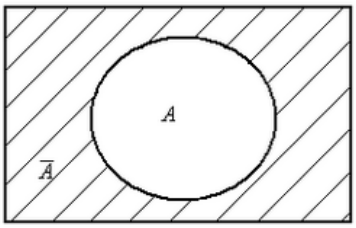
- разность множеств ($A \setminus B$)



- симметрическая разность (A Δ B)



- дополнение множества

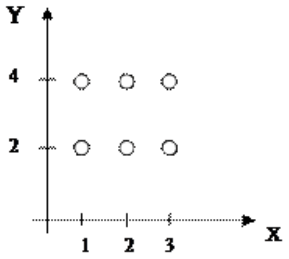


- декартово произведение

  $\forall x \in A; y \in B$

  $AxB = ((x, y)| x \in A, y \in B)$

  Пример: X = {1,2,3} Y = {2, 4}



Инъекция, сюръекция, биекция

Инъекция – отображение f множестваXна множество Y (f:X → Y), при котором разные элементы множества Xпереводятся в разные элементы множества Y.

Сюръекция – отображение множества X на множество Y, при котором каждый элемент множества Yявляется образом хотя бы одного элемента множества X.

Биекция – отображение, которое является одновременно и субъективным и инъективным. При биективном отображении каждому элементу одного множества соответствует ровно один элемент другого множества, при этом определено обратное отображение, которое обладает теме же свойствами.

Мощность множества. Счетные множества

Мощность множества – число элементов этого множества.

Счетное множество – это бесконечное множество, элементы которого можно пронумеровать натуральными числами.

Отношения на множествах

Область определения: $dom(R) := \{x \mid \exists y := xRy\}$

Область значений: $rang(R) := \{y \mid \exists x := xRy\}$

Основные теоремы теории множеств

- любая часть счетного множества либо конечна, либо счетна
- сумма конечного или счетного числа конечных или счетных множеств есть счетное множество
- любое бесконечное множество содержит счетное подмножество
- если множество Мне счетно, а $B \in M$ есть конечное или счетное множество, то множество M и множество M / B имеют одинаковый ординал
- присоединение к некоторому счетному/несчетному бесконечному множеству M счетного или конечного множества A дает множество с таким же ординалом
- множество рациональных чисел счетно
- множество всех пар натуральных чисел счетно
- если некоторое бесконечное множество содержит в себе произвольное множество, то они относятся к одному классу эквивалентности

- множество всех конечных последовательностей состоящих из элементов данного счетного множества есть счетное множество
- множество всех алгебраических чисел счетно

Множества образуют континуум С. Континуум – мощность множества всех вещественных чисел.

Кольца

Кольцо – абелева группа относительно операции сложения. Оно обладает свойствами:

- Дистрибутивности относительно сложения
- Коммутативности
- Ассоциативности
- Существования единицы

Примеры колец:

- Целые числа
- Рациональные числа
- Действительные числа

Характеристика кольцаK – некоторое число, равное минимальному натуральному числу, для которого: $!a \in K / a + a + \ldots + a = 0$, если K – несчетное число.

$CharK = 0 \Leftrightarrow \lambda^{\lambda} \sim K$

Поля

Полем называется коммутативное, ассоциативное кольцо с единицей, в котором любой ненулевой элемент обратим (Q, R).

В Z обратимы только 1 и -1.

Группа точек эллиптических кривых

Задача шифрования - поиск биективного отображения, у которого нахождение обратного является сложным с точки зрения вычислительных затрат мероприятием.

Алгебраическая кривая порядка N называется множеством точек аффинной плоскости, для которых: f(x, y) = 0. При этом сама функция представляет собой полином с коэффициентами на поле k: $degf_k(x, y) = degP_k(x, y) = n$.

Особые точки алгебраической кривой – это набор точек вида:

(x_{0i}, y_{0i}) $i = \overline{1,S}$ |
$$\begin{cases} \frac{\partial f(x,y)}{\partial x}(x_{0i}, y_{0i}) = 0 \\ \frac{\partial f(x,y)}{\partial y}(x_{0i}, y_{0i}) = 0 \end{cases}$$

Двойные особые точки алгебраических кривых и парные особые точки алгебраических кривых – обобщение на дифференциалах более высоких порядков рассмотренной выше системы.

Точки бифуркации – особые точки кривой, в которых все производные равны нулю.

Родом алгебраической кривой P называется главная характеристика алгебраической кривой порядка N, вычисляемой как $P = \frac{1}{2}(n-1)(n-2) - r$, где r – число точек возврата

Эллиптические кривые и групповые задачи на эллиптической кривой

Эллиптическая кривая над полем K, есть кривая первого рода в P^2 (проективное пространство) с набором точек (x, y, z) $\in$K, удовлетворяющие условию Вейерштрасса:

E: $y^2z + a_1xyz + a_2yz = x^3 + a_2x^2z + a_4xz^2 + a_5z^3 a_i \in K_{\oplus}$

Если charK> 3, то уравнение Вейерштрасса может быть записано в аффинных координатах с учетом специальной точки 0(∞;∞).

E:$y^2 = x^3 + ax + b$

Известно, что на эллиптической кривой существует групповой закон, определяющий аддитивную абелеву группу, при этом нейтральный элемент группы 0, есть несобственная точка. Групповая операция двух полиномов приводит к получению результирующего полинома $P_3(x_3, y_3)$ через сложение, задаваемое системой:

$$\lambda = \begin{cases} \frac{y2-y1}{x2-x1} x2 \neq x1 \\ \frac{3(x1)^2}{2y1} x1 = x2 \end{cases}$$

Китайская теорема об остатках

Китайская теорема об остатках применяется для решения систем уравнений вида:

$$\begin{cases} x = b_{1\{n1\}} \\ x = b_{2\{n2\}} \\ \ldots\ldots\ldots \\ x = b_{k\{nk\}} x \in Z \end{cases}$$

К n_i выдвигается требование взаимной простоты, то есть НОД(n_i, n_j) = 1 $i \neq j$

Данная система всегда имеет решение, более того, любое целое $y \in Z$, сравнимое с x, также является решением.

Пример:

Решим систему из уравнений с помощью китайской теоремы об остатках

$$\begin{cases} x = 2_{\{5\}} \\ x = 15_{\{17\}} \\ x = 5_{\{12\}} \end{cases}$$

Вычислим $N = n_1 * n_2 * n_3 = 5 * 17 * 12 = 1020$

$N1 = N/n_1 = 1020/5 = 204$

$N2 = N/n_2 = 1020/17 = 60$

$N3 = N/n_3 = 1020/12 = 85$

Решим вспомогательные уравнения вида $N_i * y_i = a_{i\{ni\}}$

Рассмотрим подробно первое из них:

$204 * y_1 = 2_{\{5\}}$

$200 * y_1 + 4 * y_1 = 2_{\{5\}}$

$0 + 4 * y_1 = 2_{\{5\}}$

Перебирая $y_1 = 1, 2, 3, 4, 5, \ldots$ находим его значение. Оно равно 3.

Проверка: $4 * 3 = 12_{\{5\}} = 2_{\{5\}}$

Остальные два уравнения решаются аналогично:

$60y_2 = 15_{\{17\}} y_2 = 13$

$85y_3 = 5_{\{12\}} y_3 = 5$

Итоговое решение имеет вид: $x = N_1 * y_1 + N_2 * y_2 + N_3 * y_3 = 1817_{\{1020\}} = 797_{\{1020\}}$

Математический аппарат теории чисел

$n \in N$, p – простое число, $Z = \{0, 1, 2, \ldots, N-1\}$

Z_{12}: $9 + 8 = 5_{\{12\}}$

$5 * 7 = 11_{\{12\}}$

$5 - 7 = 10_{\{12\}}$

$\forall x,y \in Z \; \exists a,b \in Z \; \exists$ НОД$(x, y) = ax + by$

Определение: обратным к y^{-1}: $x \in Z$: y будет называться такое число y, что $xy \equiv 1_{\{N\}}$, $N = 2n+1$, $n \in N$.

Пример:

$y^{-1}_{\{2\}} \equiv N \equiv \frac{N+1}{2}$

Лемма: число x имеет обратное по модулю n число тогда и только тогда, если НОД(x,N) = 1

$Z^*_{\{p\}} := \{x \in Z \mid НОД(x, N) = 1\}$

Пример: пусть p – простое число, тогда

$Z^*_p = Z_p \setminus \{0\} = \{1, 2, \dots, p-1\}$

Для любого $x \in Z^*_p$ может быть найден обратный элемент путем нахождения множителей Безу через расширенный алгоритм Евклида $ax + b = O_{\{N\}}$.

$X = -b*a^{-1}_{\{N\}}$

Теорема Ферма

Для любого натурального числа n> 2 уравнение $a^n + b^n = c^n$ не имеет решений в целых ненулевых числах a, b, c.

$\forall x \in Z^*_p : x^{p-1} \equiv 1_{\{p\}}$

Пример: p = 5

$Z_p = \{0, 1, 2, 3\}$ $x \in Z^*_p \mid НОД(x, p) = 1$

$3^4 = 81 \equiv 1_{\{5\}}$

$\forall x \in Z^*_p : x*x^{p-2} \equiv 1_{\{p\}}$ представляет собой альтернативный способ вычисления обратных элементов, однако значительно менее эффективный, чем метод Евклида.

Тест на простоту Ферма

Пусть необходимо сгенерировать большое простое число P длиной 1 килобайт. Для этого используется следующий алгоритм, как следствие из теоремы Ферма:

- случайно выбирается числор$\in [2^{1024}; 2^{1025} - 1]$
- $2^{p-1} \approx 1_{\{p\}}$, если это так, то p – искомое число, если нет, то начинаем снова с 1-го пункта

Теоремы Эйлера

$\exists g \in Z^*_p \mid \{1, g, g^2, \dots, g^{p-2}\} = Z^*_p$, при этом p называется образующим элементом циклический группы <g>.

Пример:

p = 24

$Z^*_{24} = \{1, 5, 7, 11, 13, 17, 19, 23\}$

Не каждый элемент группы является образующим.

Порядком элемента g в группе Z^*_p называется число элементов в ней. Порядок равен наименьшему a> 0, такому, что $g^a \equiv 1_{\{p\}}$

Пример:

$ord_7(3) = 6$; $ord_7(1) = 1$; $ord_7(2) = 3$

Теорема Лагранжа

$\forall g \in Z^*_p ord_p(g) \quad (p-1)$

Для любого образующего порядок образующего делит порядок (p-1). Для такого случая вводят $\varphi(N) = |Z^*_N|$.

$\varphi(12) = |\{1, 5, 7, 11\}| = 4$

$\varphi(p) = p - 1$

$N = pq$; $\varphi(N) = N - p - q + 1 = (p-1)(q-1)$

Обобщенная теорема Эйлера

$$\forall x \in Z^*_N : x^{\varphi(N)} \equiv 1_{\{N\}}$$

Пример: $5^{\varphi(24)} = 5^8 = 390625 = 1_{\{24\}}$

Обобщенная теорема Эйлера является базисным понятием криптографии, носит более широкий характер, чем теорема Ферма и является основой криптосистем стандарта RSA.

Нелинейные сравнения

Задача извлечения корня произвольной степени из некоторого целого числа по модулю n.

Число $x \in Z^*_p | x^t \equiv C_{\{P\}}$ называется корнем C степени t из числа x.

Лемма: $Z^*_p x \rightarrow x^2$ не является биекцией

{10}	{11}
1^2	1
10^2	1
2^2	4
9^2	4
3^2	9
8^2	9
4^2	5
7^2	5
5^2	3
6^2	3

Определение: число $x \in Z_p$ называется квадратичным вычетом (QR), если оно имеет квадратный корень Z_p.

Если P – простое число, то

$QR_{\{p\}}(p) = \frac{P-1}{2} + 1$ – число вычетов

Модальная теорема Эйлера

$x \in Z^*_p$, где p – нечетное простое число, тогда $x^{\frac{P-1}{2}} \equiv 1_{\{P\}}$. Значение $x^{\frac{P-1}{2}}$ называют символом Лежандра.

Лемма: $C \in Z^*_p \Rightarrow sqrt(C) = C_{\{P\}}^{\frac{P-1}{2}}$

Если $p \equiv 1_{\{4\}}$, то корень может быть вычислен за кратчайшее время, во всех остальных случаях время пропорционально $O(\log_2^3 P)$.

Квадратные уравнения по модулю p

$ax^2 + bx + c = O_{\{p\}}$

Особенности вычисления: $x_{1,2} = (\frac{1}{2a})*(-b \pm D^{1/2})$ D – дискриминант. Корень из D на конечном поле может и не существовать.

Функция $Dlog_q(g^x) = x$, $x \in [0; q-2]$ называется функцией *дискретного логарифмирования*.

Z_{11}

$Dlog_2(1) = 0_{\{11\}}$

$Dlog_2(2) = 1_{\{11\}}$

$Dlog_2(3) = 8_{\{11\}}$

$Dlog_2(4) = 2_{\{11\}}$

$Dlog_2(5) = 4_{\{11\}}$

$Dlog_2(6) = 9_{\{11\}}$

$Dlog_2(7) = 7_{\{11\}}$

$Dlog_2(8) = 3_{\{11\}}$

$Dlog_2(9) = 6_{\{11\}}$

$Dlog_2(10) = 5_{\{11\}}$

Функция дискретного логарифма может быть обобщена. Пусть $G = <g>_{q-1} = \{1; g; g^2; \dots ; g^{q-1}\}$. Говорят, что задача дискретного логарифмирования является вычислительно сложной в циклической группе G, если любому алгоритму $\mathscr{A}$, вычисляемому за полиномиальное время, вероятность $P_{g \leftarrow G;\ x \in Zp}\ [\mathscr{A}(G, q; q^x) = x]$ составляет пренебрежимо малую величину.

Примером группы, в которой задача дискретного логарифмирования является вычислительно сложной, является Z^*_p, где p – большое простое число или группа точек конечной эллиптической кривой над конечным полем.

Лучшим из известных алгоритмом дискретного логарифмирования таких типов задач является алгоритм – решетка числового поля. Его трудоемкость – $e^{0(n^{1/3})}$, где n – числа в битах.

Следующая вычислительно сложная процедура – факторизация. Факторизация – разложение составного числа на простые составляющие.

$Z^+_2(n) = \{N/N = pq\}$, где p и q – n-битные простые числа, их нахождение является вычислительно сложной задачей.

На данный момент рекордом является факторизация 768 битного числа.

Необходимые сведения из теории вероятностей

U – финитное множество. Cap|U| - мощность этого множества. Можно сопоставить частоту появления элементов этого множества во входном потоке шифрования/дешифрования, то есть собрать статистику их встречаемости. Эта частота будет стремиться к вероятности появления элемента множества входного/выходного потока.

$P: U \rightarrow B^1$

$\sum_{x \in U} p(x)=1$

Если все символы множества во входном/выходном потоке равновероятны, то распределение носит равномерный характер.

$P(x) = \frac{1}{capU}$

$P(x_0) = 1;\ x_0 \in U$

$P(x_0) = 0;\ \forall x \notin U$

Рассмотрим случай совпадения множества U с булевым кубом B_n и исследуем вопросы распределения вероятностей для таких случаев:

Полная группа элементарных событий имеет вид [p(0; 0; … ; 0); p(0; 0; … ; 1); … ; p(1; 1; … ; 1)]. $\Theta \subseteq U$.

$P_r(\Theta) = [(0, 0, \dots, 0); \dots; (1, 1, \dots, 1)]$

Мощность множества Θ определяется по формуле $|\Theta| = \frac{2}{cap|B|}$

$P_r[\Theta] = \sum_{x \in U} p(x)$ – по всем событиям из покрытия B_n.

$\tau \subseteq B^8$;

$\tau \mid \subseteq B^8$

$lsb_2(x) = 11$ (младший бит)

$\Delta \mid \subseteq B^8$

$msb_2(x) = 11$ (старший бит)

Теорема сложения вероятностей

$$p[A_1 \cup A_2] \leq p[A_1] + p[A_2]$$

$p[A] = 0.5$

$p[(lsb_2(x) = 11) \cup (msb_2(x) = 11)] \leq 0.25 + 0.25 = 0.5$

$X: U \rightarrow V^x$

$B^n \rightarrow B^1$

$p[x = 0] = 0.5 \quad p[x = 1] = 0.5$

$p[x = v]: p[x^{-1}(v)]$

Пусть U – финитное множество. Равновероятность выбора $\forall a \in U p[r = a] = \frac{1}{capU}$.

Вероятностный алгоритм

$\mathscr{A}$: $y \leftarrow \mathscr{A}(m)$. По выходному значению случайной переменной m однозначно вычисляется значение выходной переменной y.

$C \leftarrow \mathscr{A}(m)$

$p[A \cap B] = p[A]p[B]$

Случайные величины x и y независимы, если для $\forall a;b \in V$:

$p[(Y = b)] \cup p[(x = a)] = P[(x = a)]p[(Y = b)]$

$U = B^2$

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

$r \leftarrow U$

$x = lsb(r)$

$y = msb(r)$

$p[(x = 0) \cup (y = 0)] = p[(r = 00)] = 0.5 = p[x = 0]*p[y = 0]$

xB^n

y – независимая от x равномерно распределенная величина на B^n.

Z:=x + y – новая случайная величина, представляет собой равномерно распределенную случайную величину.

Теорема (парадокс дней рождения):

$r_1, \dots, r_n \in U$. n = 1, 2, …, sqrt(capU) . Тогдар[$\exists i \neq j; r_i = r_j] \geq 0.5$

Литература

1. Зегжда П.Д. Теория и практика обеспечения информационной безопасности. - М.: Москва, 1996 - 292 с.

2. Казарин О.В. Методология защиты программного обеспечения. - М.: МЦНМО Москва, 2009 - 464 с.

3. Запечников С.В., Казарин О.В., Тарасов А.А. Криптографические методы защиты информации. - М.: ЮРАЙТ, 2015 - 309 с.

4. Грушо А.А., Применко Э.А., Тимошина Е.Е. Теоретические основы компьютерной безопасности. - М.:Academia, 2009 - 340 с.

5. Романьков В.А. Введение в криптографию. - М.: Форум, 2012 - 240 с.

6. Бабаш А.В., Баранова Е.К. Криптографические методы защиты информации. - М.: КноРус, 2017 - 200 с.