ЛЕКЦИЯ 16. ЭЛЕМЕНТЫ КРИПТОГРАФИИ

КРИПТОГРАФИЯ И КРИПТОАНАЛИЗ

Криптование – шифрование (не путать с кодированием).

Криптоанализ – расшифровка.

Наиболее известны:

- шифры замены (частный случай – шифр Цезаря);
- шифры перестановки.

ОДНОСТОРОННЯЯ ФУНКЦИЯ

В математике можно вспомнить много примеров, когда функция $y = F(x)$ вычисляется легко, а обратная функция $x = F^{-1}(y)$, хоть и существует, но вычисляется гораздо сложнее (пример – квадрат и корень квадратный, показательная функция и логарифм).

Определение. Пусть y=F(x) – функция на сообщениях и N – размер сообщения. Будем называть эту функцию *односторонней*, если существует полиномиальный алгоритм вычисления $y = F(x)$ (шифрование) и не существует полиномиального алгоритма вычисления $x = F^{-1}(y)$ (расшифровка).

Пример использования. Пароли пользователй хранятся на диске в виде $y = F(x)$. Обратная функция просто не нужна!

ФУНКЦИЯ С СЕКРЕТОМ

Определение. Назовем y=F(x) *функцией с секретом* (ключом) K, если существуют эффективные алгоритмы вычисления $F(x)$ и $F^{-1}(x)$ при условии, что ключ K известен, и не существует таких алгоритмов, если ключ K неизвестен.

В шифрах замены и перестановки ключом является перестановка.

Удобнее всего разделить ключ на две «половинки» - ключ для шифрования P (публичный) и ключ для расшифровки (секретный) S:

- $F(x)$ эффективно вычисляется при известном P;
- $F^{-1}(x)$ эффективно вычисляется при известном S;
- $F(x)$ и $F^{-1}(x)$ не могут быть вычислены без знания таковых.

Пример.

P – телефонный справочник с сортировкой по фамилиям

S - телефонный справочник с сортировкой по номерам

Шифрование: берём букву и заменяем её на случайный номер из первого справочника для человека с фамилией на эту букву.

Расшифровка: берём номер, ищем фамилию, берём первую букву.

Проблема: как передать **секретный** ключ по **открытому** каналу связи?

КРИПТОСИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

ОТКРЫТЫЙ И СЕКРЕТНЫЙ КЛЮЧ

В 70-е годы появились *криптографические системы с открытым ключом*, где эта задача была решена (Диффи и Хеллман, 1976).

Участники переговоров – A и B. Каждый имеет открытый и секретный ключ:

$$P_A, S_A \text{ и } P_B, S_B.$$

АЛГОРИТМ ПЕРЕСЫЛКИ СООБЩЕНИЯ

1. A и B публикуют публичные ключи (обмениваются по открытому каналу связи);
2. A создает сообщение M, шифрует его с помощью P_B и посылает по открытому каналу связи $M' = P_B(M)$;
3. B получает его и дешифрует $S_B(M') = S_B(P_B(M)) = M$.

Расшифровать сообщение может только B!

Но где гарантия, что это сообщение действительно послал A???

ЦИФРОВАЯ ПОДПИСЬ

1. A и B публикуют публичные ключи (обмениваются по открытому каналу связи);
2. A создает сообщение M;
3. A генерирует цифровую подпись $Z = S_A(M)$;
4. A шифрует «подписанное» сообщение MZ и посылает $M' = P_B(MZ)$ по открытому каналу связи;
5. B получает сообщение и дешифрует его $S_B(M') = S_B(P_B(MZ)) = MZ$;
6. B проверяет цифровую подпись: $M \equiv P_A(Z)$.

З а д а ч а (давалась на собеседовании в Google). Мистер A решил перед свадьбой сделать своей невесте мисс B подарок: послать шкатулку с брильянтами. Но на почте все шкатулки, если для этого не нужно ломать замок, вскрываются:

1. Если шкатулка открыта, то из неё вытряхивается и присваивается всё содержимое – в том числе и ключи.
2. Если шкатулка закрыта, то к ней пробуются все ключи, которые есть в наличии, и если один из них подходит, то шкатулка открывается и см. п.1

Как переслать брильянты???

ШИФРОВАНИЕ С ПОМОЩЬЮ «РЮКЗАКА»

Всегда можно считать, что сообщение представляет собой набор чисел (или одно очень длинное число). Одну из систем с открытым ключом можно построить на *задаче о рюкзаке*.

Дано число S и N целых чисел $x_1, \ldots, x_N$. Нужно выбрать из них такое подмножество чисел, которые в сумме дают ровно S (или сообщить, что такого нет).

ШИФРОВАНИЕ И ДЕШИФРОВАНИЕ

Публикуется набор из N чисел (N=10):

104 416 624 1248 913 243 798 1596 1609 676

Шифрование: переводим каждые N бит в соответствующую сумму:

0100011010 → S=416+243+798+1609=3066

Дешифровка: нужно для числа S=3066 решить задачу о рюкзаке.

Т.о. на дешифровку N бит требуется 2^N операций !!!

Но как сделать дешифровку доступной для адресата?

ПРОСТАЯ И СЛОЖНАЯ ЗАДАЧИ

Определение. Назовем задачу о рюкзаке *простой*, если каждое следующее число в ней больше суммы всех предыдущих. Например:

1 4 6 12 25 51 105 210 421 850

Тогда алгоритм решения очень простой (найдите его!). Но ведь им может воспользоваться и противник!

Создадим «для себя» простую задачу, а опубликуем сложную:

1	4	6	12	25	51	105	210	421	850
104	416	624	1248	913	243	798	1596	1609	676

Утверждается, что эти задачи *двойственные*: решение сложной можно заменить на решение простой. Набор чисел простой задачи это и есть секретный ключ. Как построить сложную задачу, двойственную к простой?

Построение сложной задачи по простой:

1. Найдем сумму C всех чисел в простой задаче (у нас C=1685)
2. Возьмем D>C (у нас D=1687)
3. Найдем любые два взаимно обратных числа X и Y по (mod D) (у нас X=104 и Y=146)
4. Домножим все числа простой задачи на X (mod D):

104 416 624 1248 913 243 798 1596 1609 676

Решение сложной задачи через решение простой:

1. Пусть нужно набрать сумму S (например, 3066)
2. Найдем S'=S*Y (mod D) (получится 581)
3. Решим для S' простую задачу: 581=421+105+51+4
4. Возьмем соответствующие числа из сложной задачи: 3066=416+243+798+1609

Итак, наша криптографическая система выглядит полностью так:

Секретный ключ	1 4 6 12 25 51 105 210 421 850 D=1687 x=104 y=146
Публичный ключ	104 416 624 1248 913 243 798 1596 1609 676

АЛГОРИТМ RSA

RSA (Ron Rivest, Adi Shamir, Leonard Adleman - 1978)

В основе шифрования – сложность задачи о разложении длинного числа на простые множители.

ФУНКЦИЯ ЭЙЛЕРА

Определение. *Функцией Эйлера* называется целочисленная функция $\varphi(n)$, равная количеству чисел от 1 до n, взаимно простых с n.

Свойства:

1. Если p – простое, то $\varphi(p) = p - 1$.
2. Если p – простое, то $\varphi(p^\alpha) = p^\alpha - p^{\alpha-1}$.

   Доказательство: НЕ взаимно простые с p^α - это $1\cdot p, 2\cdot p, ..., p^{\alpha-1}\cdot p$. Значит, НЕ взимно простых с p^α будет $p^{\alpha-1}$, а взаимно простых - $(p^\alpha - p^{\alpha-1})$.
3. Если a и b – взаимно простые, то $\varphi(ab) = \varphi(a)\varphi(b)$.
   Доказательство: через *китайскую теорему об остатках*.

Отсюда получаем **формулу для вычисления** $\varphi(n)$ через разложение n на простые множители:

если $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$, то

$$\varphi(n) = (p_1^{\alpha_1} - p_1^{\alpha_1-1}) \dots (p_k^{\alpha_k} - p_k^{\alpha_k-1}) = n\left(1-\frac{1}{p_1}\right)\dots\left(1-\frac{1}{p_k}\right).$$

ТЕОРЕМА ЭЙЛЕРА И ТЕОРЕМА ФЕРМА

Самое известное и важное свойство функции Эйлера выражается в следующей теореме Эйлера**.**

Теорема 1 (Эйлер). Если a и n взаимно просты, то

$$a^{\varphi(n)} \equiv 1 (\mathrm{mod}\, n)$$

В частном случае, когда $n = p$ простое, теорема Эйлера превращается в так называемую малую теорему Ферма.

Теорема 2 (Ферма). Если a не делится на p, то

$$a^{p-1} \equiv 1 (\mathrm{mod}\, p)$$

ШИФРОВАНИЕ И ДЕШИФРОВАНИЕ

Множеством сообщений служит Z_n (n – ОЧЕНЬ длинное число, которое представляет собой произведение двух ОЧЕНЬ длинных простых чисел n=pq)

1. Возьмем p и q – длинные простые числа (больше 1000 знаков)
2. Вычислим n=pq
3. Вычислим $\varphi(n) = (p-1)\cdot(q-1)$
4. Возьмем небольшое число e, взаимно простое с $\varphi(n)$ (обычно берут простые числа, содержащие небольшое количество 1 в двоичной записи, например, числа Ферма: 3, 17, 257,

65537, … $2^{2^k}-1$). Время шифрования пропорционально количеству 1 в двоичной записи. Число e называют *открытой экспонентой*.

5. Найдем $d = e^{-1} (\mathrm{mod}\, \varphi(n))$. Для этого можно использовать расширенный алгоритм Евклида для нахождения линейного представления НОД:

$$d \cdot e - x \cdot \varphi(n) = 1.$$

Число d называют *секретной экспонентой*.

6. **Публичный ключ: P=(n, e)**
7. **Секретный ключ: S=(n, d)**

Шифрование: $M' = M^e (\mathrm{mod}\, n)$

Дешифрование: $M = (M')^d (\mathrm{mod}\, n)$

Вопрос: почему «враг» не может вычислить d, зная e?

Ответ: для этого нужно знать $\varphi(n)$, а для этого – разложение n на простые множители.

Теорема. $M^{ed} = M (\mathrm{mod}\, n)$.

Доказательство. $ed = 1 + x \cdot \varphi(n)$. Отсюда

$$M^{ed} \equiv M^{1+x\cdot\varphi(n)} \equiv M(M^{\varphi(n)})^x \equiv M \cdot 1^x \equiv M (\mathrm{mod}\, n).$$

(случай, когда M не взаимно просто с n вообще говоря возможен, но он тоже разбирается)

Пример.

1. p=557, q=571
2. N=pq= 318 047
3. $\varphi(N)$=(p-1)(q-1)=316 920
4. $e = 17$ – **открытая экспонента** (взаимно просто с $\varphi(N)$)
5. $d \equiv e^{-1} (\mathrm{mod}\, \varphi(N))$=260 993 – **секретная экспонента** (находим по алгоритму Евклида из линейного представления НОД: $1 = e \cdot d + \varphi(N) \cdot x$)
6. Открытый ключ: (318 047, 17)
7. Закрытый ключ: (318 047, 260 993)

Шифрование: M=111

M'= 111^{17} mod 318047 = 286048

Дешифрование: M = 286048^{260993} mod 318047 = 111

При возведении в большую степень используем **алгоритм быстрого возведения в степень**.

$260993 = 111111101110000001_2 = 2^{17} + 2^{16} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{11} + 2^9 + 2^8 + 2^7 + 2^0$

ПРИМЕНЕНИЯ

Из истории. Авторы RSA опубликовали для расшифровки фразу из 6-ти английских слов. Расшифровка длилась 17 лет и завершилась в 1994 году. Работа возглавлялась четырьмя видными учеными и продолжалась (не считая предварительной подготовки) 220 дней. На добровольных началах в ней участвовало около 600 человек и 1600 компьютеров, объединенных в сеть Internet.

На базе RSA разработана система PGP (Pretty Good Privacy) для шифрования сообщений в Интернете.