СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

А.Б. Левина

Криптография и криптографические протоколы

Асимметричное шифрование

СПбГЭТУ «ЛЭТИ», 2022 г.

1 ВВЕДЕНИЕ

Мы закончили знакомство с симметричным шифрованием и теперь переходим к **асимметричному шифрованию**, или, как его еще называют, **шифрованию с открытым ключом.**

Сегодня на нашей лекции будет рассмотрено:

1. Шифрование с открытым ключом – асимметричное шифрование;
2. Математические задачи, на которых основывается асимметричное шифрование;
3. Алгоритм RSA;
4. Алгоритм рюкзак.

2 ШИФРОВАНИЕ С ОТКРЫТЫМ КЛЮЧОМ

Мы уже вводили определение асимметричной криптосистемы на первой лекции, давайте его вспомним:

Определение 2.0.1 ***Криптосистема называется асимметричной или криптосистемой с открытым ключом****, если для шифрования и дешифрования/расшифровывания используют два различных ключа.*

Запишем это математически:

Процесс шифрования выглядит следующим образом $C = E_{k1}(m)$.

Процесс дешифрования: $m = D_{k2}(C)$, где, также как и в симметричном шифровании:

E – $Encrypt$- шифрующая функция;

D – $Decrypt$ - дешифрующая функция;

C – $Cipher$ - шифротекст, зашифрованный текст;

m – $message$ - открытый текст, текст который нам необходимо зашифровать.

Но! ключ, который мы будем использовать для шифрования и дешифрования $k = (k_1, k_2)$, где открытый ключ k_1 знают все $k_1 = e$, а закрытый ключ k_2 знает только Алиса $k_2 = d$.

Схематически это представлено на рисунке 1:

Рис. 1: Схема асимметричного шифрования

Почему возникла идея асимметричного шифрования? Мы помним, что симметричное шифрование работает быстро, оно легкое, потребляет мало ресурсов, но у него есть огромный недостаток – проблема передачи ключа. Как раз для решения этой проблемы и было придумано асимметричное шифрование.

Сама идея криптографии с открытым ключом - асимметричного шифрования появилась в 1976 г., в работе Диффи и Хеллмана «Новые направления в криптографии», через год, в 1977 г., была опубликована самая известная и широко применяемая асимметричная криптосистема, криптосистема RSA, ее создатели – Рональд Ривест, Ади Шамир и Леонард Адельман.

После появления криптосистемы RSA, английская разведка сообщила, что у них на вооружении с 1969 г. уже была криптография с открытым ключом, предложенная Джеймсом Эллисом, а с 1973 г. использовалась криптосистема, созданная Клиффордом Коксом «типа» RSA.

В основе асимметричного шифрования лежат математические преобразования, которые тяжело обратить не зная секретной информации – односторонние функции, или, как еще их называют, функции ловушки. В математике таких задач довольно мало, поэтому действительно Диффи и Хэллман, Джеймс Эллис, Ривест-Шамир-Адельман и Клиффорд

Кокс, не сговариваясь, могли использовать одни и те же идеи. Давайте познакомимся с этими идеями.

3 ФУНКЦИИ ЛОВУШКИ

Для знакомства с функциями ловушками, введем определение.

Определение 3.0.1 ***Односторонней называется функция*** F *обладающая двумя свойствами:*

а) существует полиномиальный алгоритм вычисления значений F(x);

б) не существует полиномиального алгоритма инвертирования функции F, т. е. не существует полиномиального алгоритма решения уравнения F(x) = y относительно x.

Введем понятие криптографической функции ловушки.

Определение 3.0.2 ***Криптографической функцией ловушкой называется функция*** F : .

а) существует полиномиальный алгоритм, который создает пару (k_1, k_2), *где* k_1 — *случайный элемент множества* K. *Значение* k_2 *называется секретом;*

б) существует полиномиальный алгоритм, который по данным k_1 *вычисляет* $F(k_1, x)$;

в) не существует полиномиального алгоритма инвертирования функции $F(k_1, \cdot)$ *при известном* k_1 *(и неизвестном* k_2*);*

г) существует полиномиальный алгоритм, который, при известном секрете k_2, *инвертирует функцию* $F(k_1, \cdot)$.

Если проговорить эти шаги, сразу ассоциируя их с криптографическим алгоритмом, то:

1. Создание ключа $k = (k_1, k_2)$ - *полиномиальный алгоритм.*
2. Шифрование открытого текста m с помощью открытого ключа k_1 - *полиномиальный алгоритм.*

3. *НЕ СУЩЕСТВУЕТ* полиномиального алгоритма дешифрования полученного шифротекста C ключом k_1.

4. *СУЩЕСТВУЕТ* полиномиальный алгоритма дешифрования шифротекста C ключом k_2.

В криптографии, самыми часто используемыми односторонними функциями/задачами ловушками, являются задача факторизация и задача дискретного логарифмирования.

Поговорим про них более подробно.

Задача факторизации - задача разложения числа на произведение его простых сомножителей, то есть это задача обратная задаче умножения.

Пример 3.0.1 *Задача факторизации: разложить число 126 на сомножители, в нашем случае ответ будет* $2 \cdot 3^2 \cdot 7$, *то есть* $N = 126 = 2^1 \cdot 3^2 \cdot 7^1$.

В криптографии, задача факторизации сводится к разложению числа на два простых сомножителя, то есть дано N, надо найти простые числа p и q: N=p·q.

Из задачи факторизации выведен ряд криптографических/математических задач:

1. Задача RSA:

Дано: $C, E, N = p \cdot q,$

E: НОД $(E, (p-1) \cdot (q-1)) = 1$

Найти: m: $m^E = C(mod\ N)$.

Для решения задачи RSA нам необходимо вспомнить функцию Эйлера.

Определение 3.0.3 *Функция Эйлера* ϕ *числа* N *вычисляется по формуле:*

$\phi(N) = \Pi^n\ \ p^{e^{i-1}} \cdot (p_i - 1)$, *где* p_i *простые сомножители числа* N, *а* e_i *степени этих сомножителей.*

Так как в криптографии N равно произведению двух простых сомножителей $p, q,$ то $\phi(N) = (p-1) \cdot (q-1)$.

2. Тест на квадратичный вычет:

Является ли данное число A полным квадратом по модулю N?

3. Извлечение квадратичных корней:

Дано A: $A = x^2 (mod\ N)$

Нужно вычислить x.

На задаче извлечения квадратных корней построена криптосистема Рабина.

Вторая задача ловушка, используемая в криптографии, это задача дискретного логарифмирования.

Задача дискретного логарифмирования - задача нахождения целого неотрицательного числа x : $B = A^x$ в конечной Абелевой группе G.

Давайте вспомним определение Абелевой группы:

Определение 3.0.4 *Абелева (коммутативная) группа — группа, в которой групповая операция является коммутативной; то есть группа* (G, ·) *Абелева, если* $A \cdot B = B \cdot A$ *для любых двух элементов A, B* $\in$ G.

В криптографии задача дискретного логарифмирования рассматривается в кольце вычетов по модулю простого числа.

Рассмотрим задачи, выходящие из задачи дискретного логарифмирования.

1. ПДЛ - проблема дискретного логарифмирования:

Дано: A, B $\in$ G, G конечная Абелева группа (G;).

Найти: x : $B = A^x$.

2. ЗДХ - задача Диффи-Хеллмана:

Дано: $B = A^x$, $C = A^y$; Найти: $D = A^{x \cdot y}$.

3. ЗДХ - задача Диффи-Хеллмана:

Дано: $B = A^x$, $C = A^y$; Найти: $D = A^{x \cdot y}$.

4 АСИММТЕРИЧНОЕ ШИФРОВАНИЕ, АЛГОРИТМ RSA

4.1 АЛГОРИТМ RSA

Познакомимся с одним из самых известных алгоритмов асимметричного шифрования - алгоритмом RSA.

Данный алгоритм основывается на задаче RSA, которую мы разобрали чуть раньше.

Алиса генерирует два больших простых числа p, q и вычисляет $N = p \cdot q$, N называется модулем алгоритма.

Далее Алиса выбирает простое число E : НОД (E, (p - 1) (q - 1)) = 1.

Доступная пара/открытый ключ – (N, E).

После этого Алиса находит d - расшифровывающую/дешифрующую экспоненту, такую что $E \cdot d = 1 (mod\ (p - 1)\ (q - 1))$, находится d с помощью расширенного алгоритма Евклида.

Секретным ключом является тройка (d, p, q).

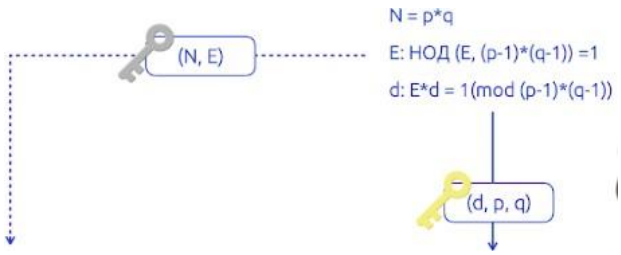

Рис. 2: Создание открытого и закрытого ключа в алгоритме RSA

Процесс шифрования идет следующим образом:

Боб берет открытый текст $m < N$ и вычисляет $C = m^E (mod\ N)$.

Для дешифрования Алиса возводит шифротекст C в степень d и получает обратно открытый текст $m = C^d (mod\ N)$.

Рис. 3: Шифрование и дешифрование в алгоритме RSA

Пример 4.1.1 *Алиса выбрала простые числа p=7, q=11 соответственно N=77,* $\phi(N) = (7-1)\cdot(11-1) = 60$.

Пусть E=37, НОД (37, 60)=1.

Получился открытый ключ (77, 37).

С помощью расширенного алгоритма Евклида Алиса решает уравнение $E \cdot d = 1 (mod\ \phi(N))$ *и находит* $d=13$.

Закрытым ключом является тройка (7, 11, 13).

Боб шифрует сообщение $m = 2$, *тогда шифротекст* $C = 2^{37} (mod\ 77) = 51$.

Алиса дешифрует C *с помощью своего закрытого ключа, получает:* $m = C^d (mod\ N) = 51^{13} (mod\ 77) = 2$.

Докажем, почему же криптосистема RSA работает.

4.1 МАТЕМАТИЧЕСКОЕ ДОКАЗАТЕЛЬСТВО РАБОТЫ КРИПТОСИСТЕМЫ RSA

Для начала вспомним, что мы работаем в группе (Z/NZ), соответственно ее порядок равен $\phi(N) = (p-1)\cdot(q-1)$.

Теорема 4.2.1 ***Теорема Лагранжа:*** *Если* $(G,)$ *группа порядка (с числом элементов)* $n = \#G$, *то каждый ее элемент* $a \quad G$ *удовлетворяет соотношению* $a^n = 1$

Тогда, по теореме Лагранжа, для x из множества Z/NZ выполнено равенство

$$x^{\phi}(N) = 1(mod\ N).$$

Давайте теперь попробуем проверить работу алгоритма RSA.

$E \cdot d = 1(mod\ \phi(N))$, соответственно мы можем это записать как

$$E \cdot d - s(p-1)(q-1) = 1.$$

Возведем C в степень d. Получаем $C^d = (m^E(mod\ N))^d = m^{E\cdot d}(mod\ N) = m^{1+s(p-1)\cdot(q-1)}(mod\ N) = m \cdot m^{s(p-1)(q-1)}(mod\ N)$, а по Теореме Лагранжа $m^{s(p-1)(q-1)} = 1$.

Получаем, что $C^d = m(mod\ N)$, и мы доказали корректность работы алгоритма RSA.

Рассмотрим несколько атак на RSA.

4.2 АТАКИ НА RSA

Рассмотрим несколько атак на RSA, первая атака, с который мы познакомимся, называется атака «раздельный модуль».

1.1 Атака «раздельный модуль».

Криптография с открытым ключом очень «дорогое удовольствие» и требует больших мощностей и времени, поэтому, очень часто, у объектов, обменивающихся информацией есть желание сэкономить.

Один из способов сэкономить на RSA - это использовать общий модуль N для всех пользователей, то есть Алиса не генерирует свой модуль N самостоятельно, а получает его и он у всех пользователей одинаковый.

Тогда у нас может быть две атаки на систему - атака внутреннего пользователя и внешнего.

Рассмотрим атаку **внутреннего пользователя**.

Лемма 4.3.1 *Если известна дешифрующая экспонента d алгоритма RSA, соответствующая открытому ключу (N, E), то число N можно эффективно разложить на множители.*

Так как Алиса знает свое d_1, соответственно согласно Лемме, она сможет разложить число N на сомножители и найти $\phi(N)$. Тогда она может посчитать d_2 Боба по формуле $d_2 = 1/E_2(mod\ \phi(N))$.

И у Алисы есть секретный ключ Боба.

Посмотрим как **внешний атакующий** может произвести атаку, если используется общий модуль N.

Ева видит два шифротекста C_1 и C_2 и знает, что модуль у них один, тогда она вычисляет $T_1 = {E_1}^{-1}(mod\ E_2)$, и $T_2 = (T_1E_1\text{-}1)/E_2$.

После этого она возводит в степень T_1 и $-T_2$ соответственно C_1 и C_2 и получает:

$$C^{T1}\,C^{-T2} = m^{(E_1)\cdot(T_1)}m^{(-E_2)\cdot(T_2)} = m^{1+E_2T_2}\,m^{-E_2T_2} = m.$$

Пример 4.3.1 *Рассмотрим пример с внешним атакующим: Пусть* $N = N_1 = N_2 = 18923$, $E_1 = 11$, $E_2 = 5$ *Перехвачено* $C_1 = 1514$, $C_2 = 8189$.

Необходимо найти m. *Ева вычисляет* $T_1 = 1$, $T_2 = 2$, *и получает* $m = C^{T1} * C^{-T2} = 100(mod\ N)$.

Рассмотрим еще одну атаку, атаку "малых экспонент".

1.2 Атака "малые экспоненты"

Пусть у нас есть три пользователя с тремя различными модулями N_1, N_2, N_3 и они все используют $E = 3$.

Предположим, что им всем было послано одно и тоже сообщение m, тогда мы получаем три разных шифротекста C_1, C_2, C_3 такие, что:

$C_1 = m^3(mod\ N_1)$, $C_2 = m^3(mod\ N_2)$, $C_3 = m^3(mod\ N_3)$.

Ева видит все эти три шифротекста и, с помощью Китайской Теоремы об остатках,

решает систему уравнений $X = C_i(mod\ N_i)\ i = 1, 2, 3$.

$X = m^3(mod\ N_1N_2N_3)$, так как $m^3 < N_1N_2N_3$ следовательно целые числа X и m^3 должны совпадать. Извлекая кубический корень из X мы находим m.

Мы познакомились с криптосистемой RSA, разобрали ее работу и некоторые наиболее известные атаки.

Предположим, что им всем было послано одно и тоже сообщение m, тогда мы получаем три разных шифротекста C_1, C_2, C_3 такие, что:

$C_1 = m^3(mod\ N_1)$, $C_2 = m^3(mod\ N_2)$, $C_3 = m^3(mod\ N_3)$.

Ева видит все эти три шифротекста и, с помощью Китайской Теоремы об остатках, решает систему уравнений $X = C_i(mod\ N_i)\ i = 1, 2, 3$.

$X = m^3(mod\ N_1N_2N_3)$, так как $m^3 < N_1N_2N_3$ следовательно целые числа X и m^3 должны совпадать. Извлекая кубический корень из X мы находим m.

Мы познакомились с криптосистемой RSA, разобрали ее работу и некоторые наиболее известные атаки.

4 АСИММЕТРИЧНОЕ ШИФРОВАНИЕ, КРИПТОСИСТЕМА «РЮКЗАК»

Мы говорили ранее, что основные задачи ловушки, используемые в асимметричном шифровании, это задача факторизации и задача дискретного логарифмирования. На самом деле, найти задачу ловушку очень сложно, так как необходимо найти задачу, которая при одних условиях будет решаться за полиномиальное время, а при других нет. Еще одна известная криптосистема асимметричного шифрования основывается на задаче о рюкзаке.

Вспомним эту задачу:

Задача о ранце (рюкзаке) (англ. Knapsack problem) — одна из NP-полных задач комбинаторной оптимизации. Название своё получила от максимизационной задачи укладки как можно большего числа нужных вещей в рюкзак при условии, что общий объём - вес всех предметов, способных поместиться в рюкзак, ограничен.

Математически ее можно записать так: $A = a_1, \ldots, a_n$ последовательность из n различных положительных целых чисел. Пусть есть число k тоже целое и положительное.

Задачей является нахождение такого набора a_i чтобы в сумме они давали k.

Существует другая задача, связанная с рюкзаком - **задача упаковки сверхвозрастающего рюкзака,** она является полиномиальной. Вспомним, что такое сверхвозрастающая последовательность.

Определение 5.0.1 *Последовательность называется* ***свехвозрастающей****, если каждый ее член меньше суммы всех предыдущих.*

Рассмотрим пример упаковки сверхвозрастающего рюкзака.

Пример 5.0.1 *Пусть у нас дан рюкзак весом 30 кг. и сверхвозрастающая последовательность весов - 2, 6, 9, 20 кг. Мы начинаем укладывать рюкзак с самого тяжелого предмета и кладем груз весом 20 кг., свободный вес рюкзака остался 10 кг., последовательность оставшихся весов стала 2, 6, 9 кг. Мы идем по последовательности и кладем самый тяжелый из оставшихся предметов - груз весом 9 кг., и свободный вес у нас стал 1 кг., как мы видим больше ничего положить мы не можем, соответственно в наш рюкзак веса 30 кг. мы положили предметы весом 9 и 20 кг.*

Как раз на объединение этих двух задач - простой и сложной, Меркл и Хэллман построили криптосистему «рюкзак».

1.1 Криптосистема «рюкзак».

Для создания ключа - мы берем сверхвозрастающую последовательность $B = \{b_1, \ldots, b_n\}$ и два простых числа N и p, это наш секретный ключ.

$k_1 = (B, N, p)$.

Открытым ключом k_2 будет новая последовательность $A = a_1, \ldots, a_n$, полученная умножением каждого члена сверхвозрастающей последователь- ности на p и взятие по модулю N, то есть $a_i = (b_i \quad p) mod\ N$- полученная последовательность будет уже не возрастающей.

$k_2 = A$.

Для шифрования мы делим полученное сообщение на блоки равные количеству

элементов в нашей последовательности A.

Открытый текст m делим на r блоков по n бит.

Для каждого блока из n бит мы начинаем «собирать» рюкзак. Те позиции открытого текста, где стоит бит=1 означают, что элемент ключа с этим номером мы кладем в наш рюкзак.

Шифротекстом буде последовательность, состоящая из получившихся весов C.

Для дешифрования мы вычисляем $l = p^{-1} mod\ N$, берем нашу сверхвозрастающую последовательность B и получившийся шифротекст C. Каждый вес C_j мы умножаем на l и берем по модулю N. Получившиеся числа «упаковываем» с помощью сверзвозрастающей последовательности, после этого мы получаем снова открытый текст.

Давайте рассмотрим это на примере.

Пример 5.0.2 *Пусть у нас есть последовательность {2, 3, 6, 13, 27, 52} и* $p = 31$, $N = 105$ *- это наш закрытый ключ. Открытый ключ получаем следующим образом:*

*2*31 mod 105= 62,*

*3*31 mod 105= 93,*

*6 * 31 mod 105 = 81,*

*13*31 mod 105 =88,*

*27*31 mod 105 = 102,*

*2*31 mod 105= 37,*

соответственно последовательность {62, 93, 81, 88, 102, 37 } наш открытый ключ.

Открытый текст 011000110101101110.

Мы разбиваем его на блоки, равные количеству элементов в ключе, соответственно на блоки по 6 бит.

Получаем: 011000, 110101, 101110

В первом блоке единицы стоят на второй и третей позиции, соответственно $C_1 = a_2 + a_3 = 93 + 81 = 174$, *во втором блоке мы получаем*

$C_2 = a_1 + a_2 + a_4 + a_6 = 62 + 93 + 88 + 37 = 280$ *и* $C_3 = a_1 + a_3 + a_4 + a_5 = 62 + 81 + 88 + 102 = 333$.

Шифротекстом будет последовательность {174, 280, 333}.

Давайте теперь дешифруемся. Для начала вычислим $l = p^{-1} mod\ N = 31^{-1} mod\ 105 = 61$, *умножим каждый* C_j *на* l, *возьмем по модулю* N *и упакуем сверхвозастающий рюкзак* B, *получаем:*

*174 * 61 mod N= 9 = 6+3* → 011000

*280 * 61 mod N= 70=52+13+3+3* → 110101

*333 * 61 mod N =48=27+13+6+2* → 101110

И мы получили обратно открытый текст 011000110101101110.

Мы рассмотрели работу криптосистемы «рюкзак». Она очень проста и очень легко реализуема, но ее криптостойкость значительно ниже, чем у криптосистемы RSA.

4 ЗАКЛЮЧЕНИЕ

В этой лекции мы начали наше знакомство с очень важным направление криптографии - асимметричным шифрованием. Мы рассмотрели, что такое асимметричное шифрование, какая логика лежит в его основе, что такое задачи ловушки. Разобрали задачи факторизации и дискретного логарифмирования – являющиеся задачами ловушками. Познакомились с самой известной криптосистемой асимметричного шифрования - криптосистемой RSA. Рассмотрели работу криптосистемы рюкзак.