МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»
КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

«АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ»

ЛАБОРАТОРНАЯ РАБОТА №9 (дополнительная)

Асимметричный алгоритм шифрования RSA

Автор: Гуков Сергей Юрьевич

Версия: 1.0 (28.10.2024)

Данная работа не является обязательной. При ее успешном выполнении и защите, а также посещении минимум 75% лекций появляется возможность получить автоматом за экзамен оценку «отлично». Данная работа сдается только после сдачи всех остальных лабораторных работ. Срок сдачи – до конца зачетной недели.

Цель и задание

Программно реализовать на любом языке программирования ассиметричный алгоритм шифрования RSA. Продемонстрировать на примерах его работу по зашифровыванию и расшифровыванию текста. Разрешается реализовать как версию с пользовательским интерфейсом, так и консольную версию с дружелюбным командно-текстовым интерфейсом.

Описание

Источник и более подробная информация в статье на Хабре.

В отличии от симметричных алгоритмов шифрования, имеющих всего один ключ для шифрования и расшифровки информации, в алгоритме RSA используется 2 ключа – открытый (публичный) и закрытый (приватный).

Публичный ключ шифрования передаётся по открытым каналам связи, а приватный всегда держится в секрете. Но зачем нужно целых два ключа и как они работают?

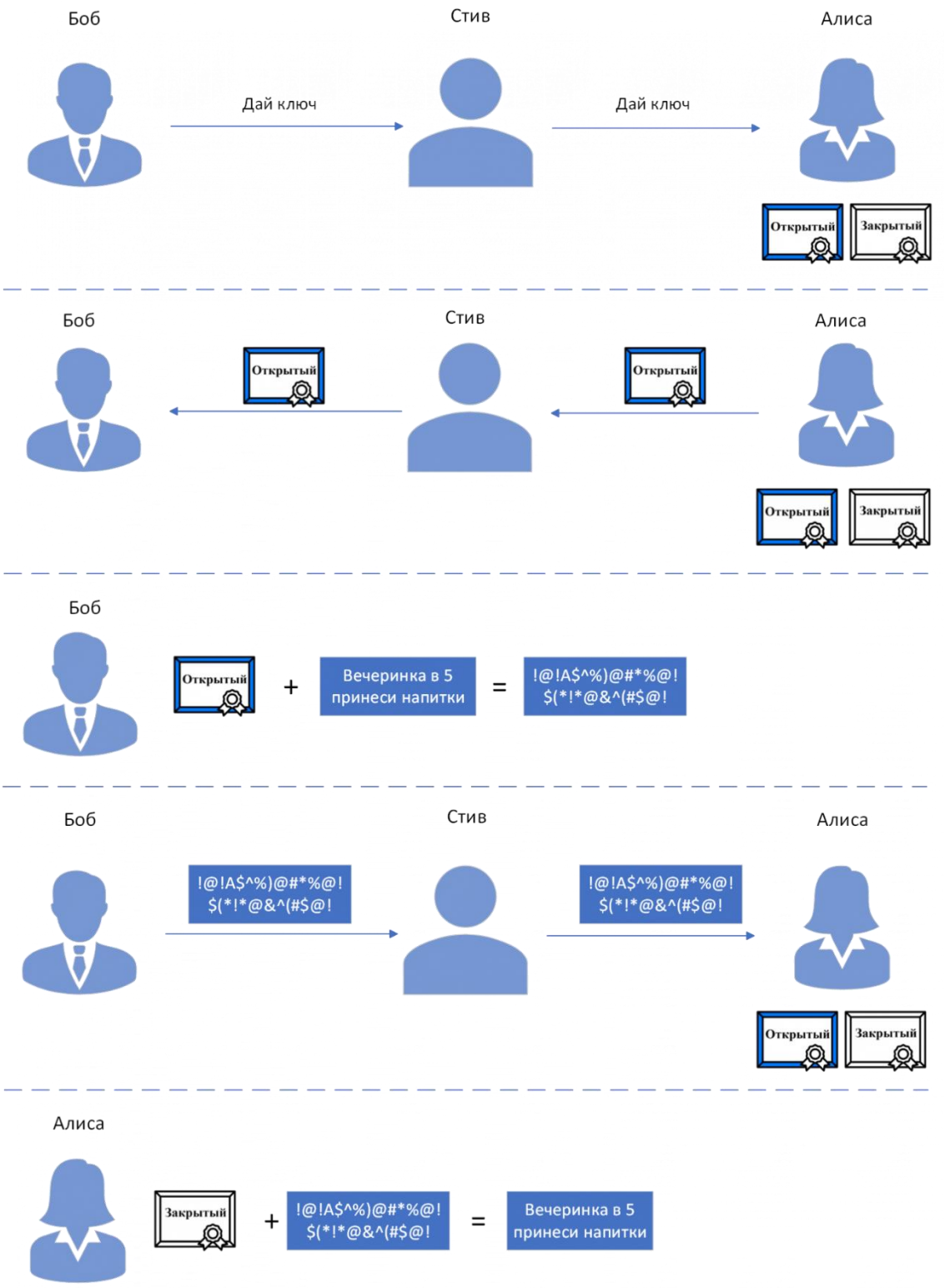
В ассиметричной криптографии и алгоритме RSA, в частности, публичный и приватный ключи являются двумя частями одного целого и неразрывны друг с другом. Для шифрования информации используется открытый ключ, а для её расшифровки приватный.

Предположим, Боб хочет передать Алисе какое-то сообщение, но лично он это сделать не может, поэтому ему необходимо использовать посредника, например Стива. Однако Боб передаёт Алисе информацию про сюрприз для Стива на его день рождения, так что не может допустить, чтобы Стив это сообщение увидел. И тут ему пригодится протокол RSA.

1. Перед обменом сообщением, Боб просит у Алисы её открытый ключ
2. После получения ключа, переданного через Стива, Боб шифрует своё сообщение ключом Алисы
3. Далее Боб, через Стива, передаёт Алисе зашифрованное сообщение
4. Алиса расшифровывает сообщение своим закрытым ключом

Таким образом, Стив видел открытый ключ Алисы и зашифрованное сообщение от Боба, но без закрытого ключа Алисы это сообщение не расшифровать. То есть, пусть Стив и держал в руках все передаваемые данные, но он не может узнать, что Боб передал Алисе!

Наглядная схема:



Основные шаги алгоритма RSA

** mod – операция взятия остатка от деления.*

** взаимно простыми называются такие числа, которые не имеют между собой ни одного общего делителя, кроме единицы.*

Теперь опишем последовательность шагов алгоритма RSA:

- выбрать два больших простых числа p и q;
- вычислить: $n = p*q$ и $\varphi(n) = (p-1)*(q-1)$;
- выбрать случайное число e, взаимно простое с $\varphi(n)$;
- определить такое число d, для которого является истинным выражение: $(e*d) \bmod(\varphi(n)) = 1$;
- числа e и n – это открытый ключ, а числа d и n – это закрытый ключ;

На практике это означает следующее: открытым ключом зашифровывают сообщение, а закрытым – расшифровывают. Пара чисел закрытого ключа держится в секрете.

- разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа $M(i)$;

Обычно блок берут равным одному символу и представляют этот символ в виду числа – его номера в алфавите или кода в таблице символов (например ASCII или Unicode).

- шифрование алгоритмом RSA производится по формуле: $C(i) = (M(i)^e) \bmod n$;
- расшифровка сообщения производится с помощью формулы: $M(i) = (C(i)^d) \bmod n$.

Теоретическая часть

Предположим, Алиса считает нужным разрешить всем желающим отправлять ей секретные сообщения, расшифровать которые способна только она. Тогда Алиса подбирает два больших простых числа p и q. Держа их в секрете, Алиса публикует их произведение

$$N = p*q$$

которое называют модулем алгоритма. Кроме того, Алиса выбирает шифрующую экспоненту E, удовлетворяющую условию

$$НОД(E,(p-1)(q-1)) = 1$$

Как правило E берут равным 3, 17 или 65 537. Пара, доступная всем желающим, – это (N, E). Для выбора секретного ключа Алиса применяет расширенный алгоритм Евклида к паре чисел E и $(p-1)(q-1)$, получая при этом расшифровывающую экспоненту d. Найденная экспонента удовлетворяет соотношению

$$E * d = 1(\mathrm{mod}(p-1)(q-1))$$

Секретным ключом является тройка (d, p, q). Фактически, можно было бы выбросить простые делители p и q из ключа и помнить лишь о d и всем числе N.

Допустим теперь, что Боб намерен зашифровать сообщение, адресованное Алисе. Он сверяется с открытым ключом и представляет сообщение в виде числа m, строго меньшего модуля N алгоритма. Шифротекст C получается из га по следующему правилу:

$$C = m^E (\mathrm{mod}\, N)$$

Алиса, получив шифрограмму, расшифровывает её, возводя число C в степень d:

$$m = C^d (\mathrm{mod}\, N).$$

Равенство имеет место в связи с тем, что порядок группы $(Z / NZ)^*$ равен $\varphi(N) = (p-1)(q-1)$. Поэтому, по теореме Лагранжа, $x^{(p-1)(q-1)} = 1(\mathrm{mod}\, N)$ для любого числа. Поскольку E и d взаимно обратны по модулю $(p-1)(q-1)$, при некотором целом числе s получается равенство $Ed - s(p-1)(q-1) = 1$.

Следовательно,

$$C^d = (m^E)^d = m^{Ed} = m^{1+s(p-1)(q-1)} = m * m^{s(p-1)(q-1)} = m(\mathrm{mod}\, N)$$

Для прояснения ситуации рассмотрим пример. Пусть $p = 7$ и $q = 11$. Тогда $N = 77$, а $(p-1)(q-1) = 6*10 = 60$. В качестве открытой шифрующей экспоненты возьмём число $E = 37$, поскольку $НОД(37, 60) = 1$. Применяя расширенный алгоритм Евклида, найдём $d = 13$, т. к. $37*13*481 = 1(\mathrm{mod}\, 60)$.

Предположим, нужно зашифровать сообщение, численное представление которого имеет вид: $m = 2$. Тогда мы вычисляем

$$C = m^E (\mathrm{mod}\, N) = 2^{37} (\mathrm{mod}\, 77) = 51.$$

Процесс расшифровывания происходит аналогично:

$$m = C^d \pmod N = 51^{13} \pmod{77} = 2.$$

В RSA открытый и закрытый ключ состоит из пары целых чисел. Закрытый ключ хранится в секрете, а открытый ключ сообщается другому участнику, либо где-то публикуется.

Генерация ключей RSA

Шифрование начинается с генерации ключевой пары (открытый, закрытый ключ). Генерация ключей в RSA осуществляется следующим образом:

1. Выбираются два простых числа p и q (такие что p неравно q).

2. Вычисляется модуль $N = p*q$.

3. Вычисляется значение функции Эйлера от модуля N: $\varphi(N) = (p-1)(q-1)$.

4. Выбирается число e, называемое открытой экспонентой, число e должно лежать в интервале $1 < e < \varphi(N)$, а так же быть взаимно простым со значением функции $\varphi(N)$.

5. Вычисляется число d, называемое секретной экспонентой, такое, что $d*e = 1(\mathrm{mod}\,\varphi(N))$ то есть является мультипликативно обратное к числу e по модулю $\varphi(N)$.

Итак, мы получили пару ключей:

Пара (e, N) – открытый ключ.

Пара (d, N) – закрытый ключ.

Шифрование и расшифровывание в RSA

Есть следующий сценарий: Боб и Алиса переписываются в интернете, но хотят использовать шифрование, чтобы поддерживать переписку в секрете. Алиса заранее сгенерировала закрытый и открытый ключ, а затем отправила открытый ключ Бобу. Боб хочет послать зашифрованное сообщение Алисе:

Шифрование: Боб шифрует сообщение m, используя открытый ключ Алисы (e, N): $C = E(M) = M^e \bmod(N)$, и отправляет C Алисе.

Расшифровывание: Алиса принимает зашифрованное сообщение C. Используя закрытый ключ (d, N), расшифровывает сообщение $M = D(C) = C^d \bmod(N)$.

Теорема Эйлера для понижения степени:

Теорема Эйлера. Для любого модуля m и целого числа a, взаимно простого с m, справедливо сравнение $a^{\varphi(m)} = 1 \bmod m$.

Следствие 1 (малая теорема Ферма). Для любого простого числа p и натурального числа a, взаимно простого с ним, верна формула Ферма:

$$a^{p-1} = 1 \bmod p$$

Следствие 2 (о вычислении обратного элемента).

$$a^{-1} = a^{\varphi(m)-1} \bmod m$$

для любых двух натуральных простых чисел a и m.

Пример. Вычислите значение выражения $11^{219} \bmod 91$. Решение.

$$91 = 7*13;$$

$$\varphi(91) = 6*12 = 72;$$

$$(11, 91) = 1$$

По теореме Эйлера имеем:

$$11^{219} \bmod 91 = 11^{72*3+3} \bmod 91 = (11^{72})^3 * 11^3 \bmod 91 \equiv 11^3 \bmod 91 \equiv 121*11 \bmod 91 \equiv 330 \bmod 91 \equiv 57 \bmod 91 = 57$$

Пример шифрования и расшифровывания в RSA

Шифрование:

1. Выбираем простые числа $p = 3, q = 11$
2. Вычисляем модуль $N = p*q = 3*11 = 33$
3. Вычисляем функцию Эйлера от модуля n: $\varphi(N) = (p-1)(q-1) = 2*10 = 20$
4. Выбираем открытую экспоненту $e = 7$
5. Определяем закрытую экспоненту d: $d*e = 1(\bmod \varphi(N)) \Rightarrow d = 3$

Будем шифровать сообщение "RSA", пусть букве А соответствует цифра 1, В – 2, С – 3 и т.д., тогда:

$$R = 18; S = 19; A = 1$$

Открытый ключ:

$$(e, N) = (7, 33)$$

$$C_1 = (18^{7}) \bmod 33 = 6$$

$$C_2 = (19^{7}) \bmod 33 = 13$$

$$C_3 = (1^{7}) \bmod 33 = 1$$

$$C(\text{"}RSA\text{"}) = 6, 13, 1$$

Расшифровывание:

Используем закрытый ключ

$$(d, N) = (3, 33)$$

$$M_1 = (6^3) \bmod 33 = 18$$

$$M_2 = (13^3) \bmod 33 = 19$$

$$M_3 = (1^3) \bmod 33 = 1$$

$$18 = R, 19 = S; 1 = A$$

Получаем исходное сообщение "RSA".

Требования к структуре проекта

- ✓ Весь программный код должен быть разбит на отдельные функции/методы и (или) классы, писать весь код внутри функции «void main()» не допускается
- ✓ Имена функций, переменных и классов (при наличии) должны отражать их назначение
- ✓ Основные части кода должны иметь комментарии

Отчет должен содержать

1. Цель работы
2. Задание
3. Краткое описание хода разработки

4. Схема работы алгоритма шифрования RSA
5. Исходный код программы (с комментариями)
6. Результаты работы программы с примерами
7. Выводы

При оценке выполнения работы будут учитываться грамотность оформления исходного кода, работоспособность программы и соответствие отчета правилам оформления текстовых документов по ГОСТ 7.32-2017. Титульные листы лабораторных работ представлены на сайте ГУАП (https://guap.ru/standart/doc).