ТЕОРИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

И ЗАЩИТЫ ИНФОРМАЦИИ

**АЛГОРИТМ ШИФРОВАНИЯ «RSA»**

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ Н.А.Соловьев

подпись дата

Москва 2011

**Содержание**

Введение…………………………………………………………………………………………………...3

1 Описание алгоритма шифрования RSA……….…….………………………………………………....4

2 Виды атак на алгоритм RSA…….………………………………………………………………….…..7

3 Расчет контрольно примера по алгоритму RSA……………………………………………..............13

Заключение……………………………………………………………………………….………………15

Список использованных источников…………………………………………………………...………16

**Введение**

На современном этапе развития общества информация играет важнейшую роль, поэтому проблемы защиты информации являются исключительно актуальными. Если конфиденциальная информация попадет к злоумышленникам, то предприятие может столкнуться с большими убытками, а если секретная информация государственного значения будет перехвачена противником – это и вовсе может предрешить исход конфликта. Одним из самых эффективных способов защиты информации при передаче и хранении является шифрование. Оно используется повсеместно, как при хранении информации в облачных системах, так и при передаче конфиденциальных данных между филиалами предприятий через Интернет.

Алгоритмы шифрования имеют широкую классификацию, но наиболее важным является их деление на симметричные и асимметричные. Симметричные алгоритмы шифрования – алгоритмы шифрования, которые используют один ключ для шифрования и расшифровки данных, который должен держаться в секрете. Симметричные алгоритмы шифрования более изучены по сравнению с асимметричными в силу их большего возраста. Асимметричные алгоритмы шифрования – алгоритмы шифрования, использующие два ключа: один для шифрования, называемый открытым или публичным, который может быть известен злоумышленнику, т.к. передается по открытому каналу; второй для расшифровки, называется секретным или закрытым, должен храниться в секрете. На практике в сложных системах, как правило, алгоритмы разного типа используются совместно, к примеру, при передаче конфиденциальной информации симметричные используются для шифрования данных, а асимметричные для шифрования ключа симметричного алгоритма.

Алгоритм шифрования RSA был одним из первых ассиметричных алгоритмов, но и сегодня находит широкое применение в различных системах. Алгоритм RSA может использоваться как для формирования электронной цифровой подписи, так и для шифрования с открытым ключом, но в данной работе ограничимся рассмотрение RSA применительно к шифрованию.

**1 Описание алгоритма шифрования RSA**

Наиболее известным и распространенным алгоритмом шифрования с открытым ключом является алгоритм RSA, названный по первым буквам фамилий своих изобретателей – Ривеста (Rivest), Шамира (Samir) и Адлемана (Adleman). RSA был создан в 1978 году в МТИ. Криптосистема RSA - первая практическая реализация криптографии с открытым ключом на основе понятия однонаправленной функции с секретом, предложенного Диффи и Хеллманом. Криптографические системы с открытым ключом используют однонаправленные функции, которые обладают следующими свойствами:

-если известно x\,, то f(x)\,вычислить относительно просто;

-если известно y=f(x)\,, то для вычисления x\,нет простого (эффективного) пути.

Под однонаправленностью понимается не теоретическая однонаправленость, а практическая невозможность вычислить обратное значение, используя современные вычислительные средства, за обозримый интервал времени. В основу криптографической системы с открытым ключом RSA положена задача умножения и разложения составных чисел на простые сомножители, которая является вычислительно однонаправленной задачей.

Алгоритм RSA:

1. Выбрать два случайных простых числа p и q.

2. Вычислить , n называют модулем.

3. Вычислить функцию Эйлера от n: .

4. Выбрать случайное целое число e, называемое открытой экспонентой, удовлетворяющее неравенству и являющееся взаимно простым с .

5. Вычислить число d, называемое секретной экспонентой, мультипликативно обратное к числу e по модулю , т.е. удовлетворяющее: или , где k – некоторое целое число.

6. Пара (e,n) – открытый ключ, передается отправителю сообщения. Пара (d,n) – закрытый ключ, держится в секрете.

7. Преобразовать сообщение m в целое число, такое что . Получить зашифрованное сообщение c, по формуле: .

8. Расшифровать зашифрованное сообщение и получить исходное сообщение m по формуле: . Выполнить обратное преобразование m из целого числа в сообщение.

Блок-схема описанного алгоритма представлена на рисунке 1.



Рисунок 1 – Блок-схема алгоритма RSA

На стороне Б происходит генерация ключей, затем открытый ключ передается по незащищенному каналу стороне А, от которой нужно получить сообщение. Сторона А шифрует сообщение с использованием открытого ключа, и зашифрованное сообщение передается по открытому каналу стороне Б. Сторона Б дешифрует зашифрованное сообщение с использованием секретного ключа и получает исходное сообщение. Но есть еще и сторона В, которая может перехватывать все что передается по открытому каналу, о ней речь пойдет в следующем разделе.

Для безопасности криптосистемы RSA самую важную роль играет выбор чисел p и q, следующие рекомендации основаны на теоретических и экспериментальных результатах:

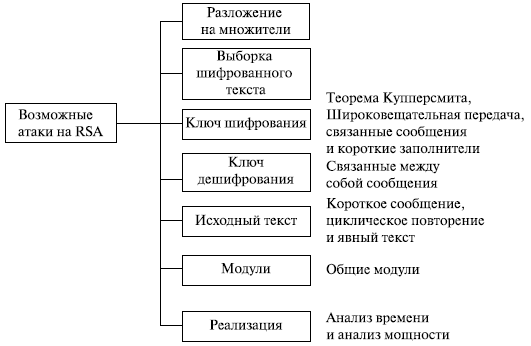
1. Число бит для n должно быть, по крайней мере, 1024, т.е. n должно быть приблизительно 21024, или 309 десятичных цифр.
2. Два простых числа p и q должны каждый быть по крайней мере 512 битов, т.е. p и q должны быть приблизительно 2512 или 154 десятичными цифрами.
3. Значения p и q не должен быть очень близки друг к другу, иначе их можно найти методом факторизации Ферма.
4. Отношение p/q не должно быть близко к рациональному числу с маленьким числителем или знаменателем.
5. Модуль n не должен использоваться совместно.
6. Число e, как правило, стараются выбрать таким, чтобы число единиц в двоичной записи было минимально, это влияет на скорость шифрования. Так же e не должно быть слишком малым, так как это может снизить безопасность шифрования.
7. Число d, как правило, вычисляют, используя расширенный алгоритм Евклида. Если произошла утечка секретного ключа d, нужно немедленно изменить n так же, как e и d. Было доказано, что знание n и одной пары (e, d) может привести к открытию других пар того же самого модуля.
8. Сообщения должны быть дополнены, для большей рандомизации шифрованных сообщений. На практике используются различные схемы заполнения, например перед сообщением добавляется определенное количество «мусора» - сгенерированных символов не несущих информацию, и уже такое сообщение шифруется, - это обеспечивает более высокую стойкость.

RSA считается достаточно медленным алгоритмом – скорость шифрования около 30 кбит/с при 512 битном ключе на процессоре 2 ГГц, поэтому в чистом виде используется редко, не считая генерации цифровой подписи. При необходимости передачи большого количества информации используется гибридная схема шифрования алгоритмом RSA c каким-нибудь симметричным алгоритмом, например AES. Сами данные шифруются алгоритмом AES, а ключ шифрования для AES шифруется алгоритмом RSA, далее оба зашифрованных сообщения могут быть переданы по открытому каналу и на приемной стороне осуществлена дешифрация.

**2 Виды атак на алгоритм RSA**

Для шифрования и дешифрования применяют возведение в степень по модулю. При использовании быстрого алгоритма, возведение в степень по модулю выполнимо в полиномиальное время. Однако нахождение модульного логарифма так же сложно, как и разложение числа по модулю. Для него нет алгоритма с полиномиальным временем. Это означает, что сторона А может зашифровать сообщение общедоступным ключом (e) в полиномиальное время. Сторона Б также может расшифровать его в полиномиальное время (потому что известно d). Но сторона В не может расшифровать это сообщение, потому что она должна была бы вычислить корень e-той степени из c, используя модульную арифметику. Другими словами, сторона А использует одностороннюю функцию (возведение в степень по модулю) с секретом, известным только стороне Б. Сторона В не знает секрет, поэтому не может расшифровать сообщение. Для того чтобы получить исходное сообщение сторона В должна была бы вычислить . Если когда-нибудь найдут полиномиальный алгоритм для вычисления корня e-той степени по модулю n, то возведение в степень по модулю n не будет больше односторонней функцией.

До настоящего момента не было обнаружено никаких разрушительных атак RSА. Несколько атак были предсказаны. Они основаны на слабом исходном тексте, слабом выборе параметра или несоответствующей реализации. На рисунке 2 представлены категории потенциальных атак.

  
Рисунок 2 - Возможные атаки на RSA

##### Атака разложения на множители

Безопасность RSА базируется на следующей идее: модуль настолько большой, что разложение на множители в разумное время неосуществимо. Сторона Б выбирает p и q и вычисляет . Число n общедоступно, p и q являются секретными. Если сторона В сможет разложить на множители n и получить p и q, то она может вычислить . Затем сторона В может вычислить d, потому что e общедоступен. Секретный ключ d — секрет, который сторона В может использовать, чтобы расшифровать зашифрованное сообщение. Существует много алгоритмов разложения на множители, но ни один из них не может найти сомножители большого целого числа с полиномиальной сложностью времени. Для того чтобы обеспечить безопасность, RSA требует, чтобы n был больше чем 300 десятичных цифр, т.е. модуль должен быть, по крайней мере, 1024 бита. Даже при использовании мощнейшего и самого быстрого компьютера, доступного на сегодня, разложение на множители целого числа такого размера требует неосуществимо большого времени и больших денежных затрат. Это означает, что RSA безопасен, пока не будет найден эффективный алгоритм разложения на множители.

##### Атака с выборкой зашифрованного текста

Потенциальная атака RSА базируется на мультипликативном свойстве RSA. Предположим, сторона А создает зашифрованный текст с и передает его стороне Б. Также предположим, что сторона Б расшифрует произвольный зашифрованный текст для стороны В – c1, отличный от c. Сторона В перехватывает с и использует следующие шаги, чтобы найти m:

1. Выбирает случайное целое число x<n.

2. Вычисляет .

3. Передает y стороне Б для дешифрования и получает - это шаг атаки выборкой зашифрованного текста.

4. Далее легко может найти m, потому что:

-

- →

Сторона В использует расширенный алгоритм Евклида для того, чтобы найти мультипликативную инверсию x, и в конечном счете значение m.

##### Атаки на показатель степени шифрования

Чтобы уменьшить время шифрования, можно попытаться использовать короткий ключ шифрования — малое значение числа e, например, значение e = 3. Однако есть некоторые потенциальные атаки на показатель степени шифрования при его малом значении. Эти атаки не кончаются вскрытием системы, но они должны быть предотвращены. Для того чтобы сорвать эти виды атак, рекомендуется использовать достаточно большое e, например e = 216 + 1 = 65537.

Атака теоремы Куперcмита может быть главной для атаки малого показателя степени на ключ шифрования. Основное положение этой теоремы: для полинома f(x) степени e по модулю n, чтобы найти корни, если один из корней является меньшим чем n1/e, можно использовать алгоритм сложности log n. Эта теорема может быть применена к RSA-криптосистеме . Если e = 3 и известны хотя бы две трети битов в исходном тексте m, алгоритм может найти все биты в исходном тексте.

Атака широковещательной передачи может быть начата, если один объект передает одно и то же сообщение группе получателей с тем же самым ключом шифрования. Например, сторона А хочет передать одно и то же сообщение трем получателям с тем же самым общедоступным ключом e = 3 и модулями n1, n2 и n3: , , . Применяя китайскую теорему об остатках к этим трем уравнениям, сторона В может найти уравнение формы . Это означает, что , и тогда решается с помощью обычной, не модульной, арифметики: .

Атака связанных между собой сообщений была обнаружена Франклином Рейтером. Она может быть кратко описана следующим образом. Сторона А зашифровала два исходных текста, m1 и m2, с помощью e = 3 и передает c1 и c2 стороне Б. Если c1связан с c2 линейной функцией, то сторона В может восстановить m1 и m2 в выполнимое время вычисления.

Атака короткого списка, обнаруженная Куперсмитом, может быть кратко описана следующим образом. Сторона А имеет сообщение m для передачи стороне Б. Она записывает сообщение и зашифровывает его как сообщение m1, а результат записывает как c1 и передает его стороне Б. Сторона В перехватывает с1 и удаляет его. Сторона Б сообщает стороне А, что сообщение не получено, так что сторона А заполняет сообщение, снова зашифровывает как сообщение m2 и передает его стороне Б. Сторона В также перехватывает и это сообщение. Сторона В теперь имеет с1 и с2, и она знает, что оба зашифрованных текста принадлежат одному и тому же исходному тексту. Куперсмит доказал, что если m1 и m2 короткие, то сторона В способна восстановить первоначальное сообщение m.

##### Атаки на показатель степени дешифрации

Атака раскрытого показателя степени дешифрации.Очевидно, что если сторона В может найти показатель степени дешифрации d, она сможет расшифровать текущее зашифрованное сообщение, однако на этом атака не останавливается. Если сторона В знает значение d, она может использовать вероятностный алгоритм к числу n и найти значения p и q. Следовательно, если сторона Б изменит только угрожающий безопасности показатель степени дешифрования, но сохранит тот же самый модуль n, сторона В сможет расшифровать будущие сообщения, потому что она сможет разложить на множители n. Поэтому если сторона Б узнает, что показатель степени скомпрометирован, она должна выбрать новые значения p и q, вычислить n и создать полностью новые секретный и открытый ключи доступа.

Атака малого значения показателя степени дешифрации.Сторона Б может решить, что использование малого значения степени секретного ключа d приводит к более быстрой работе алгоритма дешифрации. Винер показал, что в случае d < 1/3n1/4 возможен специальный тип атаки, основанной на непрерывной дроби. Этот тип атаки может подвергнуть риску безопасность RSА. Для того чтобы это произошло, должно выполняться условие, что q < p < 2q. Если эти два условия выполняются, то сторона В может разложить n на сомножители в полиномиальное время. Поэтому рекомендовано, чтобы d имело величину d > 1/3 n1/4, чтобы предотвратить атаку малого значения ключа дешифрации.

##### Атаки исходного текста

Исходный текст и зашифрованный текст в RSA — это перестановки друг друга, потому что это целые числа в том же самом интервале (от 0 до n – 1). Другими словами, сторона В уже знает кое-что об исходном тексте. Эти характеристики могут позволить некоторые атаки исходного текста.

Атака короткого сообщения. В атаке короткого сообщения, если сторона В знает множество возможных исходных текстов, то ей известна еще одна информация и дополнительный факт, что зашифрованный текст — перестановка исходного текста. Сторона В может зашифровать все возможные сообщения, пока результат не будет совпадать с перехваченным зашифрованным текстом. Например, если известно, что сторона А посылает число с четырьмя цифрами стороне Б, то сторона В может легко испытать числа исходного текста от 0000 до 9999, чтобы найти исходный текст. По этой причине короткие сообщения должны быть дополнены случайными битами в начале и/или в конце, чтобы сорвать этот тип атаки. Настоятельно рекомендуется заполнять исходный текст случайными битами прежде чем зашифровывать.

Атака циклического повторения построена на факте, что если переставлять зашифрованный текст, который сам по себе является перестановкой исходного текста, то непрерывное шифрование зашифрованного текста в конечном счете кончится исходным текстом. Иначе говоря, если сторона В непрерывно шифрует перехваченный зашифрованный текст с, она в итоге получит исходный текст. Однако сторона В не знает, каков исходный текст, так что ей неизвестно, когда пора остановиться. Она должна пройти один шаг далее. Когда сторона В получает зашифрованный текст с снова, она возвращается на один шаг, чтобы найти исходный текст. Может ли это быть серьезной атакой на криптосистему RSA? Показано, что сложность такого алгоритма эквивалентна сложности разложения на множители n. Другими словами, нет никакого эффективного алгоритма, который может завершить эту атаку в полиномиальное время, если n является большим.

Явная атака сообщения - другая атака, которая базируется на отношениях перестановки между исходным текстом и зашифрованным текстом**.** Явное сообщение — сообщение, которое зашифровано само в себя (не может быть скрыто). Было доказано, что есть всегда некоторые сообщения, которые шифруются сами в себя. Поскольку ключ шифрования обычно нечетен, имеются некоторые исходные тексты, которые зашифрованы сами в себя, такие как m = 0 и m = 1. Но если ключ шифровки выбран тщательно, число их незначительно. Программа шифровки может всегда проверить, является ли вычисленный зашифрованный текст таким же, как исходный текст, и отклонить исходный текст перед передачей зашифрованного текста.

##### Атаки модуля

Главной атакой RSA является атака разложения на множители. Ее можно рассматривать как атаку малого модуля, которая уже рассмотрена выше, поэтому здесь рассмотрим общую атаку модуля.

Общая атака модуля.Она может быть начата, если сообщество использует общий модуль n. Например, люди в сообществе могли бы позволить третьей стороне – стороне С, которой они доверяют, выбирать p и q, вычислять n и , а так же создавать пару образцов (ei, di) для каждого объекта. Предположим, что сторона А должна передать сообщение стороне Б. Зашифрованный текст для стороны Б: , сторона Б использует свой секретный ключ, d1, чтобы расшифровывать сообщение: . Проблема в том, что сторона В может также расшифровать сообщение, если она член сообщества и ей была назначена пара образцов e2 и d2. Используя свои собственные ключи e2 и d2, сторона В может начать вероятностную атаку на сомножители n и найти d1 стороны Б. Чтобы сорвать этот тип атаки, модуль не должен быть в совместном пользовании, каждый объект должен иметь свой собственный модуль.

##### Атаки реализации

Предыдущие атаки базировались на основной структуре RSА. Как показал Дэн Бонех, есть несколько атак реализации RSА. Ниже приведены две из них: атака анализом времени и атака анализом мощности.

Атака анализом времени. Пауль Кочер демонстрировал атаку только зашифрованного текста, называемую атака анализом времени. Атака основана на быстром алгоритме с показательным временем. Алгоритм использует только возведение во вторую степень, если соответствующий бит в секретном показателе степени d есть 0; он используется и при возведении во вторую степень и умножении, если соответствующий бит - 1. Другими словами, синхронизация требует сделать каждую итерацию более длинной, если соответствующий бит - 1. Эта разность синхронизации позволяет стороне В находить значение битов в d, один за другим.

Есть два метода сорвать атаку анализом времени: добавить случайные задержки к возведению в степень, чтобы каждое возведение в степень занимало одно и то же время; Ривест рекомендовал "ослепление" - по этой идее, зашифрованный текст умножается на случайное число перед дешифрованием.

Атака анализом мощности подобна атаке анализом времени. Было показано, что если сторона В может точно измерить мощность, использованную в течение дешифрования, она может начать атаку анализа мощности на основании принципов, рассмотренных для атаки анализом времени. Итеративное умножение и возведение в квадрат потребляют больше мощности, чем только итеративное возведение в квадрат. Та же самая группа методов, которая предотвращает атаки анализом времени, может сорвать атаки анализа мощности.

**3 Расчет контрольно примера по алгоритму RSA**

Воспользуемся алгоритмом, описанным в разделе 1, блок-схема которого представлена на рисунке 1, для шифрования и дешифрования сообщения “RSA”:

1. Выберем два случайных простых числа p = 23 и q = 31.

2. Вычислим модуль .

3. Вычислим функцию Эйлера от n = 713: .

4. Выберем открытую экспоненту e = 257, удовлетворяющую неравенству .

5. Для вычисления секретной экспоненты d, мультипликативно обратной к e = 257 по модулю , т.е. удовлетворяющей , воспользуемся расширенным алгоритмом Евклида.

Примем исходные данные: r1=660; r2=257; s1=1; s2=0; t1=0; t2=1.

Воспользуемся следующими формулами для заполнения таблицы 1:

;

; ; ;

; ; ;

; ; .

Таблица 1 – Расчеты по расширенному алгоритму Евклида

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **q** | **r1** | **r2** | **r** | **s1** | **s2** | **s** | **t1** | **t2** | **t** |
| 2 | 660 | 257 | 146 | 1 | 0 | 1 | 0 | 1 | -2 |
| 1 | 257 | 146 | 111 | 0 | 1 | -1 | 1 | -2 | 3 |
| 1 | 146 | 111 | 35 | 1 | -1 | 2 | -2 | 3 | -5 |
| 3 | 111 | 35 | 6 | -1 | 2 | -7 | 3 | -5 | 18 |
| 5 | 35 | 6 | 5 | 2 | -7 | 37 | -5 | 18 | -95 |
| 1 | 6 | 5 | 1 | -7 | 37 | -44 | 18 | -95 | 113 |
| 5 | 5 | 1 | 0 | 37 | -44 | 257 | -95 | 113 | -660 |
|  | 1 | 0 |  | -44 | 257 |  | 113 | -660 |  |

Получили НОД(660,257)=1, d=113, действительно , следовательно, секретная экспонента рассчитана верно.

6. Пара (257,713) – открытый ключ, передается отправителю сообщения.

Пара (113,713) – закрытый ключ, держится в секрете.

7. Преобразуем сообщение m=”RSA” в целые числа, так что , для этого воспользуемся таблицей ASCII кодов представленной на рисунке 3: “R”=82, ”S”=83, ”A”=65.



Рисунок 3 – Таблица ASCII кодов

Так как при группировании по 2 и больше символа, при выбранных ключах не будет выполняться условие , то разделим сообщение по одному символу: m1=82; m2=83; m3=65.

Получим блоки зашифрованного сообщения:

=524

=86

=549

8. Расшифруем блоки зашифрованного сообщения:

=82

=83

=65

Получим исходное сообщение, выполнив обратное преобразование m из целых чисел в буквы по таблице 1: 82=“R”; 83=”S”; 65=”A”. Получили исходное сообщение m=”RSA”.

**Заключение**

Алгоритм RSA является мощным средством шифрования, и находит широкое применение повсеместно. Криптосистема RSA используется в самых различных продуктах, на различных платформах и во многих отраслях. В настоящее время криптосистема RSA встраивается во многие коммерческие продукты, число которых постоянно увеличивается. Также ее используют операционные системы Microsoft, Apple, Sun и Novell. В аппаратном исполнении RSA алгоритм применяется в защищенных телефонах, на сетевых платах Ethernet, на смарт-картах, широко используется в криптографическом оборудовании. Кроме того, алгоритм входит в состав всех основных протоколов для защищенных коммуникаций Internet, в том числе S/MIME, SSL и S/WAN, а также используется во многих учреждениях, например, в правительственных службах, в большинстве корпораций, в государственных лабораториях и университетах. К его достоинствам можно отнести относительную простоту реализации, высокую криптостойкость, на современном этапе развития вычислительной техники, и возможность гибкого ее варьирования изменением длины ключа. К его недостаткам можно отнести его медленную скорость работы, которая тем не менее позволяет его эффективно использовать совместно с другими алгоритмами, но зависит от размера ключа, а так же возможные проблемы с криптостойкостью в будущем относительно уже зашифрованных данных с ключами, которые будут считаться короткими при дальнейшем развитии вычислительной техники.

**Список использованных источников**

1. Вебно М., Современная криптография: теория и практика. – М.: Вильямс, 2005г.

2. Коутинхо С., Введение в теорию чисел. Алгоритм RSA. – M.: Постмаркет, 2001г.

3. http://www.cyberguru.ru/algorithms/cryptography/rsa.html - Статья «Алгоритм RSA»

4. http://www.intuit.ru/department/security/mathcryptet/14/3.html - Курс «Математика в криптографии и теории шифрования»

5. http://en.wikipedia.org/wiki/Euclidean\_algorithm - Статья «Алгоритм Евклида»

6. http://en.wikipedia.org/wiki/RSA - Статья «RSA»

7. http://en.wikipedia.org/wiki/Sieve\_of\_Eratosthenes - Статья «Решето Эратосфена»

8. http://ru.wikipedia.org/wiki/RSA - Статья «RSA»