**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра дифференциальных уравнений и системного анализа

**КАТЛИНСКИЙ Илья Геннадьевич**

# КРИПТОСИСТЕМА RSA

Дипломная работа

Студента V курса специализации 1-31 03 01-06 01 -

на соискание квалификации “Математик.Системный аналитик.”

Руководитель

ЧЕРГИНЕЦ Дмитрий Николаевич

доцент кафедры ДУиСА

Допустить к защите

Заведующий кафедрой,

профессор

Минск, 2013

# Оглавление

# Глава 1

# Введение

## 1.1 Основные определения

Криптография — наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства, а также невозможности отказа от авторства) информации.

Изначально криптография изучала методы шифрования информации — обратимого преобразования открытого (исходного) текста на основе секретного алгоритма и/или ключа в шифрованный текст (*шифротекст*). Традиционная криптография образует раздел симметричных криптосистем, в которых зашифрование и расшифрование проводится с использованием одного и того же *секретного ключа*. Помимо этого раздела современная криптография включает в себя асимметричные криптосистемы, системы электронной цифровой подписи (ЭЦП), хеш-функции, управление ключами, получение скрытой информации, квантовую криптографию.

В традиционном шифровании с *секретным ключом* (secret key) (симметричное шифрование) зашифровывающий и расшифровывающий ключи, совпадают. Стороны, обменивающиеся зашифрованными данными, должны знать общий секретный ключ. Процесс обмена информацией о секретном ключе представляет собой брешь в безопасности вычислительной системы.

Фундаментальное отличие шифрования с *открытым ключом* (асимметричное шифрование) заключается в том, что зашифровывающий и расшифровывающий ключи не совпадают. Шифрование информации является односторонним процессом: открытые данные шифруются с помощью зашифровывающего ключа, однако с помощью того же ключа нельзя осуществить обратное преобразование и получить открытые данные. Для этого необходим расшифровывающий ключ, который связан с зашифровывающим ключом, но не совпадает с ним. Подобная технология шифрования предполагает, что каждый пользователь имеет в своем распоряжении пару ключей — *открытый ключ* (public key) и личный или *закрытый ключ* (private key). Свободно распространяя *открытый ключ*, вы даете возможность другим пользователям посылать вам зашифрованные данные, которые могут быть расшифрованы с помощью известного только вам личного ключа. Аналогично, с помощью личного ключа вы можете преобразовать данные так, чтобы другая сторона убедилась в том, что информация пришла именно от вас. Эта возможность применяется при работе с цифровыми или электронными подписями. Шифрование с *открытым ключом* имеет все возможности шифрования с *закрытым ключом*, но может проходить медленнее из-за необходимости генерировать два ключа. Однако этот метод безопаснее.

## 1.2 Строгие математические определения

*Криптография* - область знаний, которая занимается разработкой методов преобразования информации с целью обеспечения ее конфиденциальности, целостности и аутентификации.

Пусть $A$ и $B$ - конечные множества, будем называть их алфавитами. Информацию, состоящую из конечного объединения элементов множества $A$, которую будем защищать, будем называть *открытым текстом.* Конечное объединение элементов множества $B$ будем называть *шифротекстом.* Пусть $X$ и $Y$ - множества открытых текстов и шифрованых текстов соответственно.

Функцию $E_k : X \rightarrow Y$, где $k$ - параметр функции,который будем называть ключом, принадледит множеству ключей $K$, будем называть *функцией шифрования.* Функция $D_k : Y \rightarrow X$называется *функцией дешифрования.*

*Шифром* или *криптосистемой* называется набор $(A,\ B,\ X,\ Y,\ K,\ E_k,\ D_k)$, удовлетворяющий требованию $D_k(E_k(x)) = x$ для каждого $x \subseteq X$ и $k \subseteq K$

*Шифрование* - процесс применения шифра к защищаемой информации, преобраование информации *(открытого текста)* в шифрованное сообщение *(шифротекст)* с помощью определенных правил, содержщихся в шифре.

*Дешифрование* - процесс, обратный *шифрованию*, преоразрвание шифрованного сообщения в защищаемую информацию с помощью определенных правил, содержащихся в шифре.

Криптосистемы ($X,\ Y,\ K,\ E_k,\ D_k$), в которых в функции шифрования $E_k$ и в функции дешифрования $D_k$ используется один и тот же ключ $k \subseteq K$, называется симметричным. Шифры,в которых для штфрования используется один ключ, а для расщифрования - другой, называются ассиметричными или криптосистемами с открытым ключом. Таким образом, криптосистемой с открытым ключом называется система ($X,\ Y,\ (k_e,\ k_d) \subseteq K,\ E_{k_e},\ D_{k_e, k_d}$), где алгоритмы шифрования и дешифрования являются открытыми, шифрованный текст $C$ и открытый ключ $k_e$ могут передаваться по незащищенному каналу, секретный ключ $k_d$ является секретным.

Основные требования, которые предъявляются к криптосистемам с открытым ключом:

1 Вычисление пары ($k_e,\ k_d$) получателем должно быть простым (полиномиальный алгоритм).

2 Отправитель, знаю открытый ключ $k_e$ и сообщение $m$, может легко вычислить криптограмму $c = E_{k_e}(m)$.

3 Получатель, используя секретный ключ $k_d$ и криптограмму $c$, может легко восстановить исходное сообщение $m = D_{k_d}(c)$.

4 Противник, зная открытый ключ $k_e$, при попытке вычислить секретный ключ $k_d$ не может его высилить

5 Противник, зная пару ($k_e,\ c$), при попытке вычилить исходное сообщение $m$ не может его вычислить

## 1.3 Постановка задачи

# Глава 2

# Простые числа

## 2.1 Построение больших простых чисел

### 2.1.1 Алгоритм

### 2.1.2 Примеры

## 2.2 Проблема факторизации

### 2.2.1 Теория

### 2.2.2 LLL-алгоритм

### 2.2.3 Теорема Копперсмита

# Глава 3

# RSA

## 3.1 Основы RSA

### 3.1.1 Теория

### 3.1.2 Алгоритм создания ключей

### 3.1.3 Алгоритм шифрования и расшифрования

### 3.1.4 Примеры

## 3.2 Криптоанализ RSA

## 3.3 Атаки на RSA

### 3.3.1 Relaxed RSA problem

### 3.3.2 Атака Франклина-Райтера

### 3.3.3 Расширенная атака Хастаадта

### 3.3.4 Factoring with High Bits Known

### 3.3.5 Атака Винера

### 3.3.6 Циклическая атака

### 3.3.7 Метод Полларда

### 3.4.1 Пример 1

Современная асимметричная криптосистема может считаться стойкой, если злоумышленник, имея два открытых текста $M_1$ и $M_2$, а также один шифротекст $C_b$ не может с вероятностью большей, чем 0.5 определить какому из двух открытых текстов соответствует шифротекст $C_b$.

Проверим, удовлетворяет ли $RSA$ данному требованию. Пусть злоумышленник прослушивает переписку $A$ и $B$. Злоумышленник видит, что $B$ в открытом виде задал $A$ вопрос. $A$ односложно отвечает $B$ на этот вопрос. $A$ шифрует свой ответ открытым ключом $B$ и отправляет шифротекст. Далее злоумышленник перехватывает шифротекст и подозревает, что в нем зашифровано либо Да, либо Нет. Всё, что ему теперь нужно сделать для того чтобы узнать ответ $A$ это зашифровать открытым ключом $B$ слово Да и если полученный криптотекст совпадет с перехваченным, то это означает, что $A$ ответила Да, в противном же случает злоумышленник поймет, что ответом было Нет.

Как видно из примера, $RSA$ не столь надежна как это принято считать. Чтобы избежать подобных ситуаций, достаточно чтобы алгоритм добавлял к тексту некоторую случайную информацию, которую бы невозможно было предугадать.

### 3.4.2 Пример 2

Рассмотрим следующий пример: пусть злоумышленник имеет доступ к расшифровывающему «черному ящику». Таким образом любой криптотекст по просьбе злоумышленника может быть расшифрован. Далее злоумышленник создает два открытых текста $M_1$ и $M_2$. Один из этих текстов шифруется и полученный в результате криптотекст $C_b$ возвращается злоумышленнику. Задача злоумышленника угадать с вероятностью большей чем 0.5 какому из сообщений $M_1$ и $M_2$ соответсвует криптотекст $C_b$. При этом он может попросить расшифровать любое сообщение, кроме $C_b$. Говорят что криптосистема стойкая, если злоумышленник, даже в таких прекрасных для себя условиях, не сможет указать какому исходному тексту соответствует $C_b$ с вероятностью большей 0.5.

Рассмотрим насколько криптостойкой окажется $RSA$ в данном случае. Итак, злоумышленник имеет два сообщения $M_1$ и $M_2$. А также криптотекст $C_b = M_1^e (mod\ n)$. Ему необходимо указать какому конкретно из двух текстов соответствует $C_b$. Для этого он может предпринять следующее. Зная открытый ключ $e$, он может создать сообщение $C'=2^e\ C_b(mod\ n)$. Далее он просит расшифровывающий «черный ящик» расшифровать сообщение $C'$. А затем несложная арифметика ему в помощь. Имеем:

$$M'=C'^d(mod\ n)=2^{e\ d}M_1{}^{e\ d}(mod\ n)=2M_1(mod\ n).$$

Таким образом вычислив $M'/2$ злоумышленник увидит $M_1$. А это означает, что он поймет что в нашем примере было зашифровано сообщение $M_1$, а следовательно мы еще раз убедились в неприемлемости использования *RSA* в его изначальном виде на практике.

### 3.4.3 RSA-OAEP

Таким образом, уже сейчас можно сказать, что *RSA* во всех своих проявлениях будь то *PGP* или *SSL* не шифрует только отправленные на вход шифрующей функции данные. Алгоритм сперва добавляет к этим данным блоки содержащие случайный набор бит. И только после этого полученный результат шифруется. Это значит, что вместо привычной всем $c=m^e(mod\ n)$ получаем более близкую к действительности $c=(m||r)^e(mod\ n)$, где $r$ - случайное число. Такую методику называют схемами дополнения. В настоящее время использование *RSA* без схем дополнения является не столько плохим тоном, сколько непосредственно нарушением стандартов.

Устранить и эту неприятность помогают схемы дополнения. Только теперь к ним выдвигается требование не только о том, чтобы дополнительная информация была абсолютно случайной и непрогнозируемой. Но так же и том, чтобы дополнительные блоки помогали определить был ли шифротекст получен в результате работы шифрующей функции или он смоделирован злоумышленником. Причем в случае, если будет обнаружено, что шифротекст смоделирован вместо расшифрованных данных атакующему будет выдано сообщение о несоответствие данных реальному криптотексту.

В *RSA* при подписи и при шифровании данных используют две различные схемы дополнений. Схема, реализуемая для подписания документов, называется *RSA-PSS(probabilistic signature scheme)* или вероятностная схема подписи. Схема, используемая при шифровании – *RSA-OAEP(Optimal asymmetric encryption padding)* или оптимизированное асимметричное дополнение шифрования, на примере *OAEP* и рассмотрим как на самом деле происходит шифрование сообщений в *RSA*.

Итак чтобы зашифровать абсолютно любое сообщение в *RSA-OAEP* делается следующее:
Выбираются две хеш-функции *G(x)* и *H(x)* таким образом, чтобы суммарная длина результатов хеш-функций не превышала длины ключа *RSA*. Генерируется случайная строка битов *l*. Длина строки должна быть равна длине результата хеш-функции *H(x)*.

Итак чтобы зашифровать абсолютно любое сообщение в *RSA-OAEP* делается следующее:

1 Выбираются две хеш-функции *G(x)* и *H(x)* таким образом, чтобы суммарная длина результатов хеш-функций не превышала длины ключа *RSA*.

2 Генерируется случайная строка битов *l*. Длина строки должна быть равна длине результата хеш-функции *H(x)*.

3 Сообщение *M* разбивают на блоки по *k-бит*. Затем к каждому полученному блоку *m* дописывают *(n-k)* нулей. Где *n-длина* хеш-функции *G(x)*.

4 Определяют следующий набор бит: $\{\mathrm{m}||0^{(\mathrm{n-k})}\oplus\mathrm{G(l)}\}||\{\mathrm{l}\oplus\mathrm{H(m}||0^{(\mathrm{n-k})}\oplus\mathrm{G(l))}\}$

5 Полученные биты представляют в виде целого числа $M_1$

6 Криптотекст получают по формуле: $C=M_1{}^e(mod\ n)$

Процесс дешифрования выглядит следующим образом:

1 Находят $M_1$ по формуле $M_1=C^d(mod\ n)$

2 В полученном наборе бит отсекают левую часть. В смысле: левой частью служат *n* левых бит числа $M_1$ где *n*-длина хеш-функции *G(x)*. Обозначим эти биты условно *T*. И заметим, что $T=\{m//0^{(n-k)} \oplus G(l)\}$. Все остальные биты являются правой частью.

3 Находим $H(T)=H(m//0^{(n-k)}\ G(l))$

4 Зная H(T) получаем l, поскольку знаем $\mathrm{l} \oplus \mathrm{H(T)}$-это правая часть блока

5 Вычислив l, находим m из $T \oplus G(l)$, поскольку $T=\{m||0^{(n\text{-}k)} \oplus G(l)\}$

6 Если *m* заканчивается *(n-k)*-нулями значит сообщение зашифровано правильно. Если нет то это значит, что шифротекст некорректен, а следовательно он скорее всего подделан злоумышленником.

Таким образом *RSA* это не только возведение в степень по модулю большого числа. Это еще и добавление избыточных данных позволяющих реализовать дополнительную защиту вашей информации. Вы, возможно, спросите: а зачем это все нужно? Неужели в действительности может произойти такая ситуация, когда атакующий получит доступ к расшифровывающему алгоритму? Совсем по другому поводу как-то было сказано: если какая-либо неприятность может произойти, она обязательно произойдет.

# Глава 4

# Заключение

## 4.1 Выводы

## 4.2 Список литературы