МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВПО "Сыктывкарский государственный университет"

КУРСОВАЯ РАБОТА

Дисциплина: Безопасность функционирования информационных систем

Тема: Шифрование данных асимметричным методом.

Исполнитель – студент 35 гр.
Горбунцов Максим Петрович
Руководитель – Цыбина Татьяна Сергеевна

Дата сдачи: «__»_____2013г

Дата защиты: «__»_____2013г

Сыктывкар 2013г

Оглавление

Введение

Передача электронных документов возможен, если обеспечить их конфиденциальность, защиту от подделки или изменения, гарантированна доставка и есть возможность избежать фальсификацию документа и невозможность отказа от авторства.

Шифрование данных было всегда востребовано, начиная с древних времен и до наших дней.

С появлением вычислительных средств, темп развития криптозащиты увеличился, так же сложность самих процессов шифрования.

Шифрование делится на симметричные алгоритмы, асимметричные алгоритмы и смешанные.

В данной работе будет рассматриваться асимметричный метод шифрования, а так же будет представлена программная реализация одного из метода шифрования.

Алгоритмы с открытым ключом

Начало асимметричным шифрам было положено в работе «Новые направления в современной криптографии» Уитфилда Диффи и Мартина Хеллмана, опубликованной в 1976 году. Находясь под влиянием работы Ральфа Меркле (англ. Ralph Merkle) о распространении открытого ключа, они предложили метод получения секретных ключей, используя открытый канал. Этот метод экспоненциального обмена ключей, который стал известен как обмен ключами Диффи — Хеллмана, был первым опубликованным практичным методом для установления разделения секретного ключа между заверенными пользователями канала. В 2002 году Хеллман предложил называть данный алгоритм «Диффи — Хеллмана — Меркле», признавая вклад Меркле в изобретение криптографии с открытым ключом. Эта же схема была разработана Малькольмом Вильямсоном в 1970-х, но держалась в секрете до 1997 года. Метод Меркле по распространению открытого ключа был изобретён в 1974 и опубликован в 1978 году, его также называют загадкой Меркле.

Вообще, в основу известных асимметричных криптосистем кладётся одна из сложных математических проблем, которая позволяет строить односторонние функции и функции-лазейки.[5]

Асимметричные алгоритмы шифрования основаны на применении однонаправленных функций. Согласно определению функция y = f(x) является однонаправленной, если ее можно легко вычислить для всех возможных вариантов x, а для большинства возможных значений y достаточно сложно вычислить такое значение x, при котором y = f(x) .

Примером однонаправленной функции может служить умножение двух больших чисел: N = S x G. Само по себе, с точки зрения математики, такое умножение представляет собой простую операцию. Однако обратная операция (разложение N на два больших множителя), называемая также факторизацией, по современным временным оценкам представляет собой достаточно сложную математическую задачу.[6]

В связи с этим в настоящее время наиболее широко используются открытые алгоритмы. Стойкость современных криптосистем основывается не на секретности алгоритма, а на секретности ключа.

Естественно, при прочих равных условиях секретность алгоритма шифрования существенно затрудняет проведение успешной криптоаналитической атаки. Поэтому были предложены современные криптосистемы, в которых непосредственно алгоритм шифрования является секретным, но в то же время имеется возможность открытого обсуждения стойкости криптосистемы. Это реализуется в гибких криптосистемах, в которых алгоритм шифрования формируется по специальному алгоритму предвычислений (инициализации) под управлением секретного ключа пользователя. Алгоритм инициализации является открытым, а сам алгоритм шифрования является секретным, так же, как и ключ шифрования.[1]

Правило Керкхоффа

Голландский криптограф Керкхофф (1835-1903) впервые сформулировал фундаментальное правило стойкости криптосистемы стойкость криптосистемы должна определяться только секретностью ключа. Иными словами, правило Керкхоффа состоит в том, что весь алгоритм шифрования, кроме значения секретного ключа, известен криптоаналитику. Это обусловлено тем, что криптосистема обычно рассматривается как открытая система. Такой подход отражает очень важный принцип технологии защиты информации: защищенность системы не должна зависеть от секретности чего-либо такого, что невозможно быстро изменить в случае утечки секретной информации. Предполагается, что все долговременные элементы системы известны криптоаналитику.

Несмотря на то, что, согласно современным требованиям криптосистемы должны выдерживать криптоанализ на основе известного алгоритма, большого объема известного открытого текста и соответствующего ему шифртекста, шифры, используемые специальными службами, сохраняются в секрете, что обусловлено необходимостью иметь дополнительный запас прочности.[1]

Схема шифрования с открытым ключом

Пусть K — пространство ключей, а e и d — ключи шифрования и расшифрования соответственно. E_e — функция шифрования для произвольного ключа eϵK, такая что:

$E_e(m)=c$

Здесь cϵC, где C — пространство шифротекстов, а mϵM, где M — пространство сообщений.

D_d — функция расшифрования, с помощью которой можно найти исходное сообщение m, зная шифротекст c :

$D_d(c)=m$

$\{E_e: e\epsilon K\}$ — набор шифрования, а $\{D_d: d\epsilon K\}$ — соответствующий набор для расшифрования. Каждая пара (E, D) имеет свойство: зная E_e, невозможно решить уравнение $E_e(m)=c$, то есть для данного произвольного шифротекста cϵC, невозможно найти сообщение mϵM. Это значит, что по данному e невозможно определить соответствующий ключ расшифрования d. E_e является односторонней функцией, а d — лазейкой.

На (рис.3) схема передачи информации лицом A лицу B. Они могут быть как физическими лицами, так и организациями и так далее.

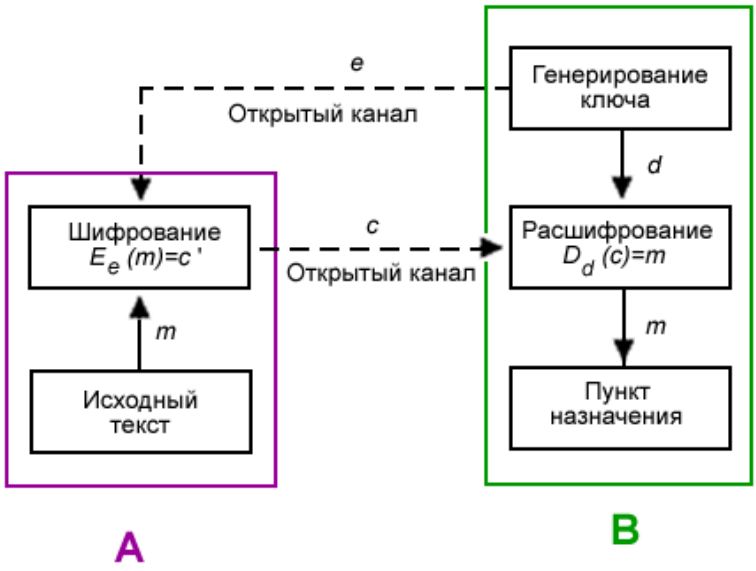


Рис.3 Асимметричная криптосистема

Пользователь B выбирает пару (e, d) и шлёт ключ шифрования e (открытый ключ) пользователю A по открытому каналу, а ключ расшифрования d (закрытый ключ) защищён и секретен (он не должен передаваться по открытому каналу).

Чтобы послать сообщение m пользомателю B, пользователь A применяет функцию шифрования, определённую открытым ключом e: $E_e(m)=c$, c — полученный шифротекст.

Пользователь B расшифровывает шифротекст c, применяя обратное преобразование D_d, однозначно определённое значением d.[5]

Основные принципы построения криптосистем с открытым ключом

Начинаем с трудной задачи P. Она должна решаться сложно в смысле теории: не должно быть алгоритма, с помощью которого можно было бы перебрать все варианты решения задачи P за полиномиальное время относительно размера задачи. Более правильно сказать: не должно быть известного полиномиального алгоритма, решающего данную задачу — так как ни для одной задачи ещё пока не доказано, что для неё подходящего алгоритма нет в принципе.

Можно выделить легкую подзадачу $P^{/}$ из P. Она должна решаться за полиномиальное время и лучше, если за линейное.

“Перетасовываем и взбалтываем” $P^{/}$, чтобы получить задачу $P^{//}$, совершенно не похожую на первоначальную. Задача $P^{//}$ должна по крайней мере выглядеть как оригинальная труднорешаемая задача P.

$P^{//}$ открывается с описанием, как она может быть использована в роли ключа зашифрования. Как из $P^{//}$ получить $P^{/}$, держится в секрете как секретная лазейка.

Криптосистема организована так, что алгоритмы расшифрования для легального пользователя и криптоаналитика существенно различны. В то время как второй решает $P^{//}$ -задачу, первый использует секретную лазейку и решает $P^{//}$-задачу. [5]

Криптоанализ алгоритмов с открытым ключом

Казалось бы, что криптосистема с открытым ключом — идеальная система, не требующая безопасного канала для передачи ключа шифрования. Это подразумевало бы, что два легальных пользователя могли бы общаться по открытому каналу, не встречаясь, чтобы обменяться ключами. К сожалению, это не так.

Нарушитель, выполняющий роль активного перехватчика, может захватить систему (расшифровать сообщение, предназначенное второму пользователю) без взламывания системы шифрования. (рис. 1).

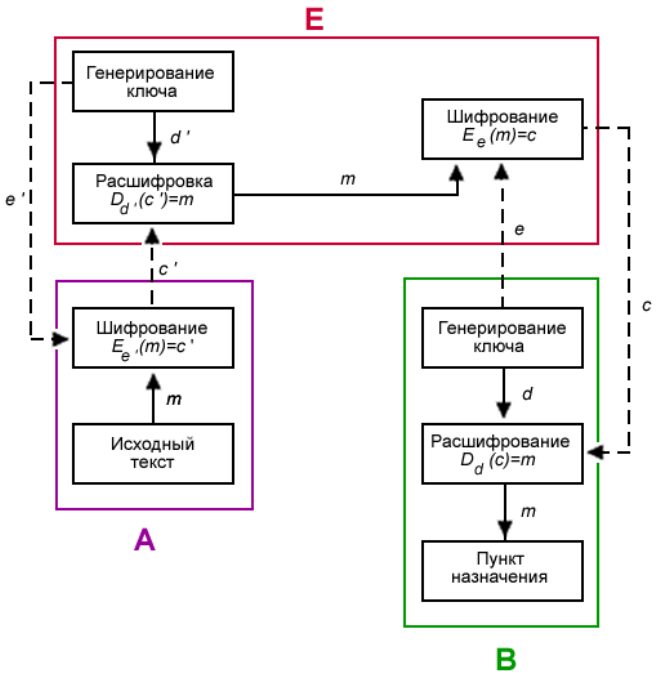


Рис.1 криптосистема с открытым ключом и активным перехватчиком

В этой модели нарушитель перехватывает открытый ключ e, посланный вторым пользователем первому. Затем создает пару ключей e' и d', «маскируется» под второго пользователя, посылая первому открытый ключ e', который, как думает первый, открытый ключ, посланный ему вторым. Нарушитель перехватывает зашифрованные сообщения от первого ко второму, расшифровывает их с помощью секретного ключа d', заново зашифровывает открытым ключом e второго и отправляет сообщение второму. Таким образом, никто из участников не догадывается, что есть третье лицо, которое может как просто перехватить сообщение m, так и подменить его на ложное сообщение m'. Это подчеркивает

необходимость аутентификации открытых ключей. Для этого обычно используют сертификаты. Распределённое управление ключами в PGP ((англ. Pretty Good Privacy) — компьютерная программа, также библиотека функций, позволяющая выполнять операции шифрования и цифровой подписи сообщений, файлов и другой информации, представленной в электронном виде, в том числе прозрачное шифрование данных на запоминающих устройствах, например, на жёстком диске.) решает возникшую проблему с помощью поручителей.

Ещё одна форма атаки — вычисление закрытого ключа, зная открытый. Криптоаналитик знает алгоритм шифрования E_e, анализируя его, пытается найти D_d. Этот процесс упрощается, если криптоаналитик перехватил несколько криптотекстов, посланных лицом A лицу B.(рис. 2)

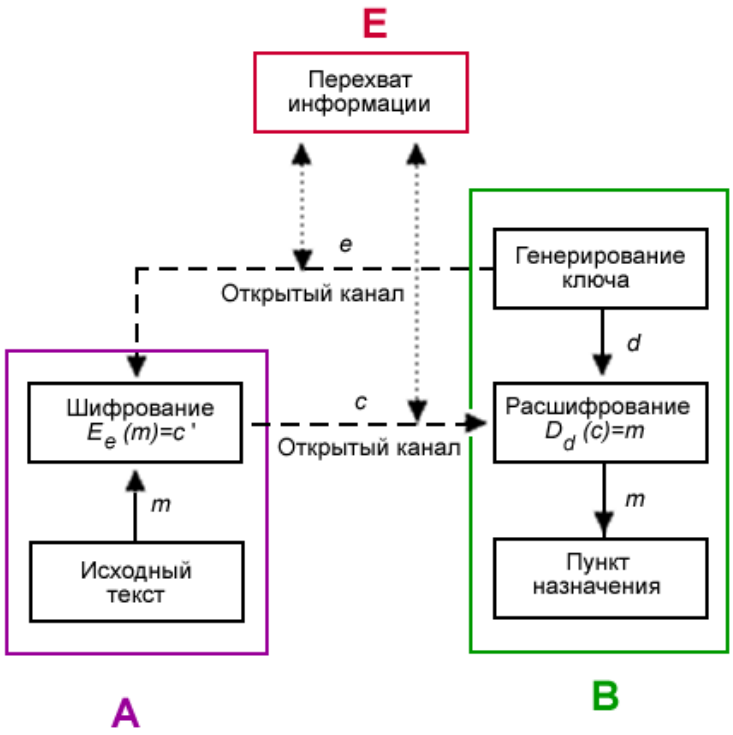


Рис. 2 Асимметричная криптосистема с пассивным перехватчиком

Большинство криптосистем с открытым ключом основаны на проблеме факторизации больших чисел. К примеру, RSA использует в качестве открытого ключа n произведение двух больших чисел. Сложность взлома такого алгоритма состоит в трудности разложения числа n на множители. Но

эту задачу решить реально. И с каждым годом процесс разложения становится все быстрее.

Для многих методов несимметричного шифрования криптостойкость, полученная в результате криптоанализа, существенно отличается от величин, заявляемых разработчиками алгоритмов на основании теоретических оценок. Поэтому во многих странах вопрос применения алгоритмов шифрования данных находится в поле законодательного регулирования. В частности, в России к использованию в государственных и коммерческих организациях разрешены только те программные средства шифрования данных, которые прошли государственную сертификацию в административных органах, в частности, в ФСБ (федеральная служба безопасности), ФСТЭК (Федеральная служба по техническому и экспортному контролю).[5]

Применение

Алгоритмы криптосистемы с открытым ключом можно использовать в:

- самостоятельных средствах для защиты передаваемой и хранимой информации
- средствах распределения ключей (обычно с помощью алгоритмов криптосистем с открытым ключом распределяют ключи, малые по объёму, а саму передачу больших информационных потоков осуществляют с помощью других алгоритмов)
- средствах аутентификации пользователей.[5]
- электронных цифровых подписях
- распределенных проверках подлинности [10]

Преимущества и недостатки

Преимущества асимметричных шифров перед симметричными:

- Не нужно предварительно передавать секретный ключ по надёжному каналу.
- Только одной стороне известен ключ шифрования, который нужно держать в секрете (в симметричной криптографии такой ключ известен обеим сторонам и должен держаться в секрете обеими).
- Пару (E, D) можно не менять значительное время (при симметричном шифровании необходимо обновлять ключ после каждого факта передачи).
- В больших сетях число ключей в асимметричной криптосистеме значительно меньше, чем в симметричной.[5]
- решена сложная проблема распределения ключей между пользователями, так как каждый пользователь может сгенерировать свою пару ключей сам, а открытые ключи пользователей могут свободно публиковаться и распространяться по сетевым коммуникациям
- исчезает квадратичная зависимость числа ключей от числа пользователей; в асимметричной криптосистеме количество используемых ключей связано с количеством абонентов линейной зависимостью (в системе из N пользователей используется 2× N ключей), а не квадратичной, как в симметричных системах
- асимметричные криптосистемы позволяют реализовать протоколы взаимодействия сторон, которые не доверяют друг другу, поскольку при использовании асимметричных криптосистем закрытый ключ должен быть известен только его владельцу.[9]

Недостатки алгоритма несимметричного шифрования в сравнении с симметричным:

- В алгоритм сложнее внести изменения.
- Хотя сообщения надежно шифруются, но получатель и отправитель самим фактом пересылки шифрованного сообщения «засвечиваются».
- Шифрование-расшифрование с использованием пары ключей проходит на два-три порядка медленнее, чем шифрование-расшифрование того же текста симметричным алгоритмом.

- Требуются существенно большие вычислительные ресурсы, поэтому на практике асимметричные криптосистемы используются в сочетании с другими алгоритмами:
  - Для ЭЦП сообщение предварительно подвергается хешированию, а с помощью асимметричного ключа подписывается лишь относительно небольшой результат хеш-функции.
  - Для шифрования они используются в форме гибридных криптосистем, где большие объёмы данных шифруются симметричным шифром на сеансовом ключе, а с помощью асимметричного шифра передаётся только сам сеансовый ключ.[5]
- на настоящий момент нет математического доказательства необратимости используемых в асимметричных алгоритмах функций
- необходимо защищать открытые ключи от подмены.[9]

Виды асимметричных шифров:

RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений, включая PGP, S/MIME, TLS/SSL, IPSEC/IKE и других.[5]

DSA (Digital Signature Algorithm) алгоритм с использованием открытого ключа для создания электронной подписи, но не для шифрования (в отличие от RSA и схемы Эль-Гамаля). Подпись создается секретно, но может быть публично проверена. Это означает, что только один субъект может создать подпись сообщения, но любой может проверить её корректность. Алгоритм основан на вычислительной сложности взятия логарифмов в конечных полях.

Алгоритм был предложен Национальным институтом стандартов и технологий (США) в августе 1991 и является запатентованным U.S. Patent 5 231 668, но НИСТ сделал этот патент доступным для использования без лицензионных отчислений. Алгоритм вместе с криптографической хеш-функцией SHA-1 является частью DSS (Digital Signature Standard), впервые опубликованного в 1994 (документ FIPS-186 (Federal Information Processing Standards)). Позднее были опубликованы 2 обновленные версии стандарта: FIPS 186-2 (27 января 2000 года) и FIPS 186-3 (июнь 2009).[5]

Elgamal (Шифросистема Эль-Гамаля) криптосистема с открытым ключом, основанная на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе стандартов электронной цифровой подписи в США (DSA) и России (ГОСТ Р 34.10-94).

Схема была предложена Тахером Эль-Гамалем в 1984 году. Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые

использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.[5]

Diffie-Hellman (Обмен ключами Диффи — Хелмана) криптографический протокол, позволяющий двум и более сторонам получить общий секретный ключ, используя незащищенный от прослушивания канал связи. Полученный ключ используется для шифрования дальнейшего обмена с помощью алгоритмов симметричного шифрования.

Схема открытого распределения ключей, предложенная Диффи и Хеллманом, произвела настоящую революцию в мире шифрования, так как снимала основную проблему классической криптографии — проблему распределения ключей.

В чистом виде, алгоритм Диффи — Хеллмана уязвим для модификации данных в канале связи, в том числе для атаки «Человек посередине», поэтому схемы с его использованием применяют дополнительные методы односторонней или двухсторонней аутентификации.[7]

ECDSA (Elliptic Curve Digital Signature Algorithm) — алгоритм с открытым ключом для создания цифровой подписи.[5]

ГОСТ Р 34.10-2001 Настоящий стандарт определяет схему электронной цифровой подписи, процессы формирования и проверки цифровой подписи под заданным сообщением (документом), передаваемым по незащищенным телекоммуникационным каналам общего пользования в системах обработки информации различного назначения [4]

Rabin - криптографический алгоритм с открытым ключом. Ее безопасность, как и у RSA, связана с трудностью разложения на множители.

Безопасность схемы Рабина опирается на сложность поиска квадратных корней по модулю составного числа. Сложность этого алгоритма аналогична проблеме разложения на множители.

Главным неудобством практического применения криптосистемы Рабина является то, что при расшифровке текста получается четыре различных сообщения. И нужно применить дополнительные усилия для нахождения истинного исходного текста.[5]

McEliece - криптосистема с открытыми ключами на основе теории алгебраического кодирования, разработанная в 1978 году Робертом Мак-Элисом. Это была первая схема, использующая рандомизацию в процессе шифрования. Алгоритм не получил широко признания в криптографии, но в то же время является кандидатом для постквантовой криптографии, так как устойчив к атаке с использованием Алгоритма Шора .

Алгоритм основан на сложности декодирования полных линейных кодов (общая задача декодирования является NP-сложной).[5]

RSA

В данной работе будет рассматриваться алгоритм RSA

RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Опубликованная в ноябре 1976 года статья Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии» перевернула представление о криптографических системах, заложив основы криптографии с открытым ключом. Разработанный впоследствии алгоритм Диффи-Хеллмана-Меркле позволял двум сторонам получить общий секретный ключ, используя незащищенный канал связи. Однако этот алгоритм не решал проблему аутентификации. Без дополнительных средств, один из пользователей не мог быть уверен, что он обменялся ключами именно с тем пользователем, который ему был нужен.

Изучив эту статью, трое ученых Рональд Райвест (Ronald Linn Rivest), Ади Шамир (Adi Shamir) и Леонард Адлеман (Leonard Adleman) из Массачусетского Технологического Института (MIT) приступили к поискам математической функции, которая бы позволяла реализовать сформулированную Уитфилдом Диффи и Мартином Хеллманом модель криптографической системы с открытым ключом. После работы над более чем 40 возможными вариантами, им удалось найти алгоритм, основанный на различии в том, насколько легко находить большие простые числа и насколько сложно раскладывать на множители произведение двух больших простых чисел, получивший впоследствии название RSA. Система была названа по первым буквам фамилий её создателей. [5]

Описание RSA было опубликовано в августе 1977 года в журнале Scientific American. Авторы RSA поддерживали идею её активного распространения. В свою очередь управление национальной безопасности(NSA) США, опасаясь использования этого алгоритма в негосударственных структурах, на протяжении нескольких лет безуспешно требовало прекращения

распространения системы. Ситуация порой доходила до абсурда, например, когда программист Адам Бек(Adam Back) описал алгоритм RSA на языке Perl, состоящий из пяти строк, правительство США запретило распространение этой программы за пределами страны.

В основу криптографической системы с открытым ключом RSA положена задача умножения и разложения простых чисел на множители, которая является вычислительно однонаправленной задачей.

В криптографической системе с открытым ключом каждый участник располагает как открытым ключом (public key), так и секретным ключом (secret key). Каждый ключ — это часть информации. В криптографической системе RSA каждый ключ состоит из пары целых чисел. Каждый участник создаёт свой открытый и секретный ключ самостоятельно. Секретный ключ каждый из них держит в секрете, а открытые ключи можно сообщать кому угодно или даже публиковать их. Открытый и секретный ключи каждого участника обмена сообщениями образуют «согласованную пару» в том смысле, что они являются взаимно обратными.[8]

Алгоритм создания открытого и секретного ключей.

Выбираются два случайных простых числа p и q и заданного размера (например, 512 битов каждое).

Вычисляется их произведение n=pq

Вычисляется значение функции Эйлера от числа n:

$\phi = (p - 1) (q - 1)$

Выбирается целое число e, взаимно простое со значением функции $\phi (n)$. Обычно в качестве e берут простые числа, содержащие небольшое количество единичных битов в двоичной записи, например, простые числа Ферма 17, 257, или 65537.

Вычисляется число d мультипликативное обратное к числу по модулю , т.е число удовлетворяющее сравнению:

$de = 1 \bmod \phi (n)$

Пара (e,n) публикуется в качестве открытого ключа RSA (RSA public key).

Пара (d,n) играет роль секретного ключа RSA (RSA secret key) и держится в секрете. [5]

Вводится сообщение для шифрования = M

Для шифрования сообщения необходимо вычислить $C= M^e \bmod n$

Для дешифрирования вычисляется $M = C^d \bmod n$

Таким образом, чтобы зашифровать сообщение, необходимо знать пару чисел (e,n), а чтобы расшифровать - пару чисел (d n). Первая пара - это открытый ключ, а вторая – закрытый.

Зная открытый ключ (E, n), можно вычислить значение закрытого ключа d. Необходимым промежуточным действием в этом преобразовании является нахождение чисел p и q для чего нужно разложить на простые множители очень большое число n для этого требовалось очень много времени. Именно с огромной вычислительной сложностью разложения большого числа на простые множители связана высокая криптостойкость алгоритма RSA.[2]

Программная реализация

Данная программа реализует несимметричный алгоритм RSA. Простые числа p, q и текст шифруемого сообщения (M) вводятся вручную, остальные значения высчитывается автоматически. Так же программа может обрабатывать сообщения большой длинны до 255 знаков, за счет ввода символов в строку и работы с ними как с элементами массива. У данной программы есть недостатки, это невозможность работы с большими числами, т.е. при расшифровке требуется возводить в большие степени, программа ограничена 2^63 числом

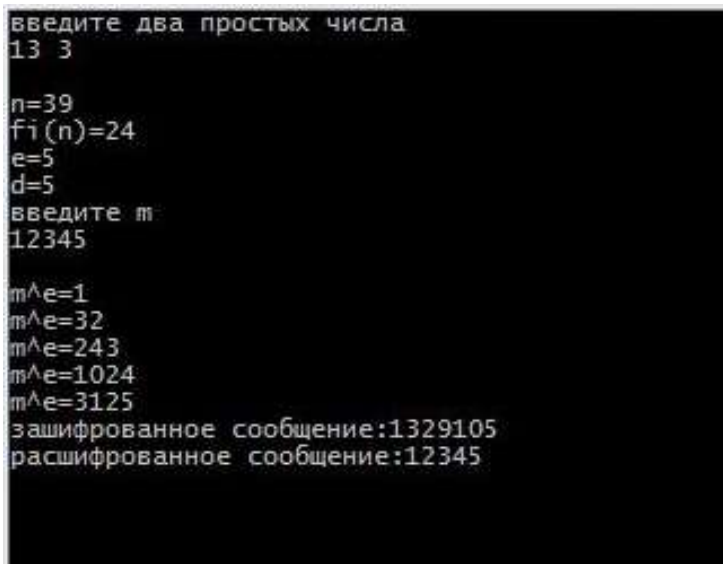
```
#include <vcl.h>
#pragma hdrstop
#include<conio.h> //getch()
#include<iostream.h> //cin/cout
#include <tchar.h>
#include <windows.h>  // вывод русских букв
#pragma argsused
int _tmain(int argc, _TCHAR* argv[])
{
SetConsoleCP(1251);        //вывод русских букв
SetConsoleOutputCP(1251);

int p,q; //простые числа, вводятся с клавиатуры
cout<<"введите два простых числа"<<endl;
cin>>p>>q;
cout<<endl;

int n,fin;
n=p*q;
cout<<"n="<<n<<endl;
fin=(p-1)*(q-1); //функция Эйлера
cout<<"fi(n)="<<fin;
cout<<endl;

                                    //e взаимно простое с f(n) и высчитывается
                                    //автоматически, постоянно перебирая e
                                    //и перебирая делители и сравниваются
                                    //общие делители
int i=4,del,e;
```

```
bool flag=false,flag2=false;
while(i!=1000 && flag!=true){ flag2=false;
                        del=2;
                       while (del!=fin && flag2!=true)  {
                                        if (i%del==0 && fin%del==0)
                                         flag2=true;
                                        del++;
                                        }
                         i++;
                        if (flag2==false) flag=true;
                    }
e=i-1;
cout<<"e="<<e<<endl;

int d;          //высчитывание закрытого ключа автоматически,
bool flag3;     //причем берется самое маленькое значение
i=0;
while (i!=1000 && flag3!=true){
if ((e*i)%fin==1){ d=i;
                        flag3=true;
                              }
                             i++;
                                    }
cout<<"d="<<d<<endl;

char m[255]; //сообщение или ключ для шифрования, ввод как строка
cout<<"введите m"<<endl;
cin>>m;
cout<<endl;
```

```
int dln=strlen(m);   //длинна введенной строки

int mass[255];
for (i = 0; i < 100; i++) //обнуление массива
        mass[i]=0;

int c,j=0; //шифрование сообщения
unsigned long long z=1;  //каждая цифра, введенная, шифруется
while (j!=dln) {        //отдельно и помещается в одномерный
                z=1;            //массив
                for (i=0; i<e; i++){ //выщитывание m^e
                z=z*(m[j]-'0');              }
                cout<<"m^e="<<z<<endl;
                mass[j]=z%n;
                j++;
                }
cout<<"зашифрованное сообщение:";   //вывод зашифрованного
for (j = 0; j < dln; j++) {        //текста
                                        cout<<mass[j];
}
cout<<endl;

cout<<"расшифрованное сообщение:";
unsigned long long mr; //расшифрование сообщения
z=1;                //а тут наоборот беруться элементы массива
j=0;                //и расшифровываются отдельно каждый
while (j!=dln) {
z=1;
for (i=0; i<d; i++){ //выщитывание c^d
z=z*mass[j];              }
```

```
mr=z%n;
j++;
cout<<mr;
                                        }

getch();
return 0;
}
```

Результат выполнения программы на (Рис.3)



Заключение

В основу асимметричных криптосистем кладётся одна из сложных математических проблем, например: факторизация чисел и вычислительной сложности взятия логарифмов в конечных полях, которая позволяет строить односторонние функции.

Асимметричные криптосистемы применяются для защиты информации, передаваемой по каналам связи, а так же при хранении данных, для распространения ключей, аутентификации и электронных цифровых подписей.

У данных алгоритмов кодирования есть свои плюсы и минусы, но лучше применять их в совокупности с симметричными методами шифрования.

Список литературы:

1. Д.А. Беляев, Ю. В. Гольчевский “Введение в криптологию” 2004г.
2. В. Олифер, Н. Олифер “Компьютерные сети Принципы, технологии, протоколы” 4-е издание 2010г.
3. Э. Таненбаум “Компьютерные сети” 4-е издание 2003г.
4. http://standartgost.ru
5. http://ru.wikipedia.org
6. http://www.razlib.ru/kompyutery_i_internet/zashiti_svoi_kompyuter_na_100_ot_virusov_i_hakerov/p4.php
7. http://dic.academic.ru/dic.nsf/ruwiki/988628
8. http://mind-control.wikia.com/wiki/RSA
9. http://infomir.forum2x2.ru/t42-topic
10. http://www.comprice.ru/articles/detail.php?ID=41120