**РЕФЕРАТ**

Отчет стр., \_\_ рис., \_\_ ист.

Пояснительная записка к диссертационной работе на степень магистра по специальности «Информационные системы и технологии» на тему: «Анализ уязвимостей и обеспечение безопасности системы лицензирования пакета программ Логос»

Магистерская диссертация посвящена анализу пакета программ Логос, а также разработке повышающая защищенность данных пакета программ.

Первая глава пояснительной записки содержит теоретические разделы, в которых изучается алгоритмы шифрования их документация и используемые стандарты. Также в этой главе анализируется код и выявляться возможные уязвимости.

Во второй главе представлена программная реализация по повышению безопасности пакета программ Логос на языке C++.

Результатом работы стала разработанная программная часть, повышающая общий уровень защиты пакета программ Логос на языке C++.

Введение

Глава 1. Теоретическая часть

1.1 Исследование алгоритмов шифрованию по обеспечению безопасности пакета программ Логос

1.1.1 Асимметричное шифрование

1.1.2 Симметричное шифрование

1.1.3 Основные используемые стандарты

1.2 Анализ безопасности пакета программ Логос

1.2.1 Исследование возможных уязвимых мест пакета программ Логос

1.2.2 Выбор библиотеки для повышения защиты пакета программ Логос

1.2.3 Выбор оптимального алгоритма для обеспечения защиты пакета программ Логос

Глава 2. Практическая часть

**Введение**

В РФЯЦ-ВНИИЭФ разрабатывается многофункциональный пакет программ инженерного анализа и суперкомпьютерного моделирования ЛОГОС, который позволяет решать задачи математического 3D моделирования в разных сферах науки и техники. В ЛОГОС реализованы передовые технологии математического моделирования; пакет предоставляет широкий круг численных методов и физико-математических моделей для расчета процессов.

В отличие от многих пакетов программ, разрабатываемых на предприятиях Росатома, ЛОГОС предназначен не только для решения внутренних задач Госкорпорации, но и готовится к выходу на коммерческий рынок Российской Федерации, а в будущем, и всего мира. Поэтому разработка пакета ЛОГОС ведется с учетом нужд различных пользователей, работающих в государственных и частных организациях, использующих в процессе моделирования разные операционные системы и языки программирования.

В связи с использование пакета программ Логос в государственных и частных организациях возникает задача в повышение уровня безопасности работы программ Логос.

**Глава 1. Теоретическая часть**

**1.1 Исследование алгоритмов шифрованию по обеспечению безопасности пакета программ Логос**

**1.1.1 Асимметричное шифрование**

Асимметричное шифрование — это метод шифрования данных, предполагающий использование двух ключей — открытого и закрытого. Открытый (публичный) ключ применяется для шифрования информации и может передаваться по незащищенным каналам. Закрытый (приватный) ключ применяется для расшифровки данных, зашифрованных открытым ключом. Открытый и закрытый ключи — это очень большие числа, связанные друг с другом определенной функцией, но так, что, зная одно, крайне сложно вычислить второе.

Асимметричное шифрование используется для защиты информации при ее передаче, также на его принципах построена работа электронных подписей.

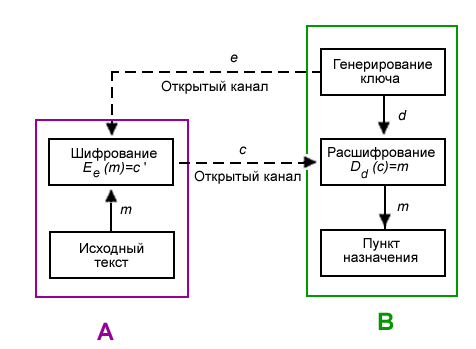


Рисунок 1 - Схема шифрования с открытым ключом

1. Боб выбирает пару (*e*,*d*) и шлёт ключ шифрования *e* (открытый ключ) Алисе по открытому каналу, а ключ расшифрования *d* (закрытый ключ) защищён и секретен (он не должен передаваться по открытому каналу).
2. Чтобы послать сообщение *m* Бобу, Алиса применяет функцию шифрования, определённую открытым ключом *e*: *Ee*(*m*) = *c*, *c* — полученный шифротекст.
3. Боб расшифровывает шифротекст *c*, применяя обратное преобразование *Dd*, однозначно определённое значением *d*

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 (*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 году и опубликован в 1978, его также называют загадкой Меркле.

В 1977 году учёными Рональдом Ривестом (Ronald Linn Rivest), Ади Шамиром (Adi Shamir) и Леонардом Адлеманом (Leonard Adleman) из Массачусетского Технологического Института (MIT) был разработан алгоритм шифрования, основанный на проблеме о разложении на множители. Система была названа по первым буквам их фамилий (RSA — **R**ivest, **S**hamir, **A**dleman). Эта же система была изобретена Клиффордом Коксом (Clifford Cocks) в 1973 году, работавшим в центре правительственной связи (GCHQ). Но эта работа хранилась лишь во внутренних документах центра, поэтому о её существовании было не известно до 1977 года. RSA стал первым алгоритмом, пригодным и для шифрования, и для цифровой подписи.

Вообще, в основу известных асимметричных криптосистем кладётся одна из сложных математических проблем, которая позволяет строить односторонние функции и функции-лазейки. Например, криптосистемы Меркля — Хеллмана и Хора — Ривеста опираются на так называемую задачу об укладке рюкзака.

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. Рисунок иллюстрирует, как Ева, выполняющая роль активного перехватчика, может захватить систему (расшифровать сообщение, предназначенное Бобу) без взламывания системы шифрования.

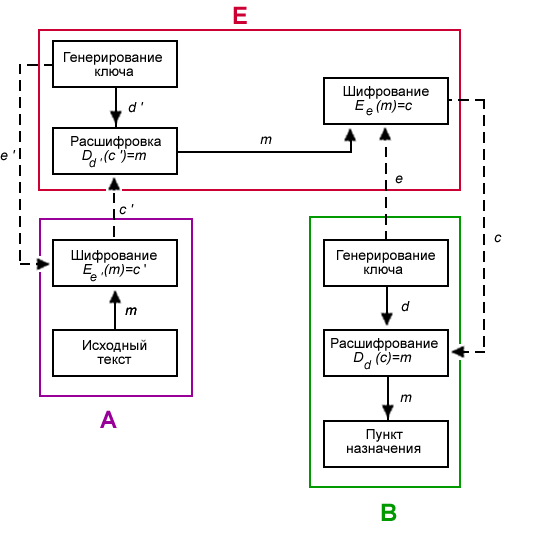


Рисунок 2 - Криптоанализ алгоритмов с открытым ключом

В этой модели Ева перехватывает открытый ключ *e*, посланный Бобом Алисе. Затем создает пару ключей *e*' и *d*', «маскируется» под Боба, посылая Алисе открытый ключ *e*', который, как думает Алиса, открытый ключ, посланный ей Бобом. Ева перехватывает зашифрованные сообщения от Алисы к Бобу, расшифровывает их с помощью секретного ключа *d*', заново зашифровывает открытым ключом *e* Боба и отправляет сообщение Бобу. 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 решает возникшую проблему с помощью поручителей.

Еще одна форма атаки — вычисление закрытого ключа, зная открытый (рисунок 3). Криптоаналитик знает алгоритм шифрования *Ee*, анализируя его, пытается найти *Dd*. Этот процесс упрощается, если криптоаналитик перехватил несколько криптотекстов с, посланных лицом A лицу B.

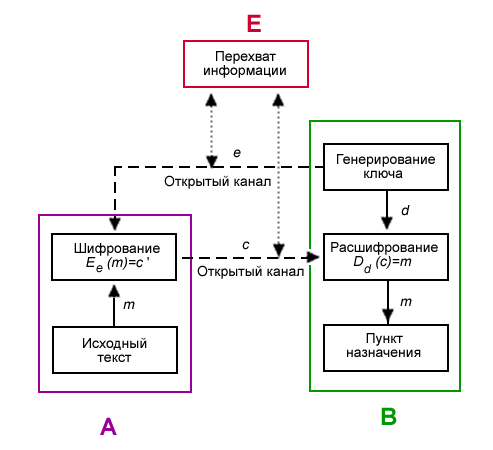


Рисунок 3 - Асимметричная криптосистема с пассивным перехватчиком

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. Ниже приведены данные разложения на множители с помощью алгоритма «Квадратичное решето».

|  |  |  |
| --- | --- | --- |
| **Год** | **Число десятичных разрядов** **в разложенном числе** | **Во сколько раз сложнее разложить** **на множители 512-битовое число** |
| 1983 | 71 | > 20 000 000 |
| 1985 | 80 | > 2 000 000 |
| 1988 | 90 | 250 000 |
| 1989 | 100 | 30 000 |
| 1993 | 120 | 500 |
| 1994 | 129 | 100 |

Также задачу разложения потенциально можно решить с помощью Алгоритма Шора при использовании достаточно мощного квантового компьютера.

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, ФСТЭК.

Алгоритмы криптосистемы с открытым ключом можно использовать:

* Как самостоятельные средства для защиты передаваемой и хранимой информации
* Как средства 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.
* Как средства аутентификации пользователей.

К преимуществу асимметричного шифрования можно отнести:

* Преимущество асимметричных шифров перед симметричными шифрами состоит в отсутствии необходимости предварительной передачи секретного ключа по надёжному каналу.
* В симметричной криптографии ключ держится в секрете для обеих сторон, а в асимметричной криптосистеме только один секретный.
* При симметричном шифровании необходимо обновлять ключ после каждого факта передачи, тогда как в асимметричных криптосистемах пару (*E, D*) можно не менять значительное время.
* В больших сетях число ключей в асимметричной криптосистеме значительно меньше, чем в симметричной.

К недостаткам асимметричного шифрования можно отнести:

* Преимущество алгоритма симметричного шифрования над несимметричным заключается в том, что в первый относительно легко внести изменения.
* Хотя сообщения надежно шифруются, но «засвечиваются» получатель и отправитель самим фактом пересылки шифрованного сообщения.[8]
* Несимметричные алгоритмы используют более длинные ключи, чем симметричные. Ниже приведена таблица, сопоставляющая длину ключа симметричного алгоритма с длиной ключа несимметричного алгоритма (RSA) с аналогичной криптостойкостью:

|  |  |
| --- | --- |
| **Длина симметричного ключа, бит** | **Длина несимметричного ключа, бит** |
| 56 | 384 |
| 64 | 512 |
| 80 | 768 |
| 112 | 1792 |
| 128 | 2304 |

* Процесс шифрования-расшифрования с использованием пары ключей проходит на два-три порядка медленнее, чем шифрование-расшифрование того же текста симметричным алгоритмом.
* В чистом виде асимметричные криптосистемы требуют существенно больших вычислительных ресурсов, потому на практике используются в сочетании с другими алгоритмами.
  1. Для ЭЦП сообщение предварительно подвергается хешированию, а с помощью асимметричного ключа подписывается лишь относительно небольшой результат хеш-функции.
  2. 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.

Виды асимметричных шифров

* RSA (Rivest-Shamir-Adleman, Ривест — Шамир — Адлеман)
* DSA (Digital Signature Algorithm)
* Elgamal (Шифросистема Эль-Гамаля)
* Diffie-Hellman (Обмен ключами Диффи — Хелмана)
* ECDSA (Elliptic Curve Digital Signature Algorithm) — алгоритм с открытым ключом для создания цифровой подписи.
* ГОСТ Р 34.10-2001
* Rabin
* Luc
* McEliece
* Williams System (Криптосистема Уильямса)

**1.1.2 Симметричное шифрование**

Симметричные криптосистемы (также симметричное шифрование, симметричные шифры) — способ шифрования, в котором для шифрования и расшифровывания применяется один и тот же криптографический ключ. До изобретения схемы асимметричного шифрования единственным существовавшим способом являлось симметричное шифрование. Ключ алгоритма должен сохраняться в секрете обеими сторонами. Алгоритм шифрования выбирается сторонами до начала обмена сообщениями.

Алгоритмы шифрования и дешифрования данных широко применяются в компьютерной технике в системах сокрытия конфиденциальной и коммерческой информации от злонамеренного использования сторонними лицами. Главным принципом в них является условие, что *передатчик и приемник заранее знают алгоритм шифрования*, а также ключ к сообщению, без которых информация представляет собой всего лишь набор символов, не имеющих смысла.

Классическим примером таких алгоритмов являются **симметричные криптографические алгоритмы**, перечисленные ниже:

* Простая перестановка
* Одиночная перестановка по ключу
* Двойная перестановка
* Перестановка "Магический квадрат"

**Простая перестановка без ключа** — один из самых простых методов шифрования. Сообщение записывается в таблицу по столбцам. После того, как открытый текст записан колонками, для образования шифровки он считывается по строкам. Для использования этого шифра отправителю и получателю нужно договориться об общем ключе в виде размера таблицы. Объединение букв в группы не входит в ключ шифра и используется лишь для удобства записи не смыслового текста.

**Одиночная перестановка по ключу** очень похожа на предыдущее шифрование. Он отличается лишь тем, что колонки таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

**Двойная перестановка** - для дополнительной скрытности можно повторно шифровать сообщение, которое уже было зашифровано. Этот способ известен под названием двойная перестановка. Для этого размер второй таблицы подбирают так, чтобы длины ее строк и столбцов были другие, чем в первой таблице. Лучше всего, если они будут взаимно простыми. Кроме того, в первой таблице можно переставлять столбцы, а во второй строки. Наконец, можно заполнять таблицу зигзагом, змейкой, по спирали или каким-то другим способом. Такие способы заполнения таблицы если и не усиливают стойкость шифра, то делают процесс шифрования гораздо более занимательным.

Перестановка магическими квадратами - называются квадратные таблицы со вписанными в их клетки последовательными натуральными числами от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число. Подобные квадраты широко применялись для вписывания шифруемого текста по приведенной в них нумерации. Если потом выписать содержимое таблицы по строкам, то получалась шифровка перестановкой букв. На первый взгляд кажется, будто магических квадратов очень мало. Тем не менее, их число очень быстро возрастает с увеличением размера квадрата. Так, существует лишь один магический квадрат размером 3 х 3, если не принимать во внимание его повороты. Магических квадратов 4 х 4 насчитывается уже 880, а число магических квадратов размером 5 х 5 около 250000. Поэтому магические квадраты больших размеров могли быть хорошей основой для надежной системы шифрования того времени, потому что ручной перебор всех вариантов ключа для этого шифра был немыслим.

В квадрат размером 4 на 4 вписывались числа от 1 до 16. Его магия состояла в том, что сумма чисел по строкам, столбцам и полным диагоналям равнялась одному и тому же числу — 34. Впервые эти квадраты появились в Китае, где им и была приписана некоторая «магическая сила».

Шифрование по магическому квадрату производилось следующим образом. Например, требуется зашифровать фразу: «ПриезжаюCегодня.». Буквы этой фразы вписываются последовательно в квадрат согласно записанным в них числам: позиция буквы в предложении соответствует порядковому числу. В пустые клетки ставится точка.

После этого шифрованный текст записывается в строку (считывание производится слева направо, построчно):

*.ирдзегюСжаоеянП*

При расшифровывании текст вписывается в квадрат, и открытый текст читается в последовательности чисел «магического квадрата», Программа должна генерировать «магические квадраты» и по ключу выбирать необходимый. Размер квадрата больше, чем 3х3.

В настоящее время симметричные шифры — это:

* блочные шифры. Обрабатывают информацию блоками определённой длины (обычно 64, 128 бит), применяя к блоку ключ в установленном порядке, как правило, несколькими циклами перемешивания и подстановки, называемыми раундами. Результатом повторения раундов является лавинный эффект — нарастающая потеря соответствия битов между блоками открытых и зашифрованных данных.
* поточные шифры, в которых шифрование проводится над каждым битом либо байтом исходного (открытого) текста с использованием гаммирования. Поточный шифр может быть легко создан на основе блочного (например, ГОСТ 28147-89 в режиме гаммирования), запущенного в специальном режиме.

Большинство симметричных шифров используют сложную комбинацию большого количества подстановок и перестановок. Многие такие шифры исполняются в несколько (иногда до 80) проходов, используя на каждом проходе «ключ прохода». Множество «ключей прохода» для всех проходов называется «расписанием ключей» (key schedule). Как правило, оно создается из ключа выполнением над ним неких операций, в том числе перестановок и подстановок.

Типичным способом построения алгоритмов симметричного шифрования является сеть Фейстеля. Алгоритм строит схему шифрования на основе функции F(D, K), где D — порция данных, размером вдвое меньше блока шифрования, а K — «ключ прохода» для данного прохода. От функции не требуется обратимость — обратная ей функция может быть неизвестна. Достоинства сети Фейстеля — почти полное совпадение дешифровки с шифрованием (единственное отличие — обратный порядок «ключей прохода» в расписании), что сильно облегчает аппаратную реализацию.

Операция перестановки перемешивает биты сообщения по некоему закону. В аппаратных реализациях она тривиально реализуется как перепутывание проводников. Именно операции перестановки дают возможность достижения «эффекта лавины». Операция перестановки линейна — f(a) xor f(b) == f(a xor b)

Операции подстановки выполняются как замена значения некоей части сообщения (часто в 4, 6 или 8 бит) на стандартное, жестко встроенное в алгоритм иное число путем обращения к константному массиву. Операция подстановки привносит в алгоритм нелинейность.

Зачастую стойкость алгоритма, особенно к дифференциальному криптоанализу, зависит от выбора значений в таблицах подстановки (S-блоках). Как минимум считается нежелательным наличие неподвижных элементов S(x) = x, а также отсутствие влияния какого-то бита входного байта на какой-то бит результата — то есть случаи, когда бит результата одинаков для всех пар входных слов, отличающихся только в данном бите.

К основным алгоритмам симметричного шифрования можно отнести:

* AES (англ. *Advanced Encryption Standard*) - американский стандарт шифрования
* ГОСТ 28147-89 — отечественный стандарт шифрования данных
* DES (англ. *Data Encryption Standard*) - стандарт шифрования данных в США до AES
* 3DES (Triple-DES, тройной DES)
* RC6 (Шифр Ривеста )
* Twofish
* IDEA (англ. *International Data Encryption Algorithm*)
* SEED - корейский стандарт шифрования данных
* Camellia - сертифицированный для использования в Японии шифр
* CAST (по инициалам разработчиков Carlisle Adams и Stafford Tavares)
* XTEA - наиболее простой в реализации алгоритм

Если сравнивать симметричное и асимметричное шифрования, то можно выделить следующие достоинства и недостатки.

Достоинства

* скорость (по данным Applied Cryptography — на 3 порядка выше)
* простота реализации (за счёт более простых операций)
* меньшая требуемая длина ключа для сопоставимой стойкости
* изученность (за счёт большего возраста)

Недостатки

* сложность управления ключами в большой сети. Означает квадратичное возрастание числа пар ключей, которые надо генерировать, передавать, хранить и уничтожать в сети. Для сети в 10 абонентов требуется 45 ключей, для 100 уже 4950, для 1000 — 499500 и т. д.
* сложность обмена ключами. Для применения необходимо решить проблему надёжной передачи ключей каждому абоненту, так как нужен секретный канал для передачи каждого ключа обеим сторонам.

Для компенсации недостатков симметричного шифрования в настоящее время широко применяется комбинированная (гибридная) криптографическая схема, где с помощью асимметричного шифрования передаётся сеансовый ключ, используемый сторонами для обмена данными с помощью симметричного шифрования.

Важным свойством симметричных шифров является невозможность их использования для подтверждения авторства, так как ключ известен каждой стороне.