Министерство образования Российской Федерации

Пермский государственный университет

## Механико-математический факультет

## Кафедра математического обеспечения вычислительных систем

УДК 519.6+681.83

##### ОТПРАВКА СООБЩЕНИЯ В БУДУЩЕЕ

***Курсовая работа.***

выполнила студентка

3-го курса 1 группы

Научный руководитель

#### профессор

Миков Александр Иванович.

Пермь 2000 г.

###### Аннотация

Вознамерившись погрузиться в летаргический сон или стать клиентом криогенного дипозитария, Вы наверняка пожелаете послать секретное сообщение в будущее в надежде на то, что его расшифруют только в нужный срок. Именно рассмотрению решений данной проблемы и посвящена эта работа. Сейчас существует два основных метода решающие проблему раскрытия сообщения в указанный срок:

* «шарады» с временным замком на базе вычислительных проблем с существенно последовательными алгоритмами решения;
* использования доверенных агентов, принимающих на себя обязательство не раскрывать информацию в течение заданного интервала времени.

Специфика первого метода заключается в том, что в отличие от традиционных криптографических методов, предполагающих наличие у получателя сообщения секретного ключа отправителя (в симметричных криптосистемах) или у отправителя сообщения аутентичного ( подлинного ) открытого ключа получателя ( в асимметричных криптосистемах ), секретный ключ уничтожается сразу после шифрования и неизвестен как отправителю, так и получателю сообщения. А при использовании второго метода с доверенными доверенных агентов возникает проблемма надёжности, которая частично может быть решена за счёт применения криптографической техники разделения секрета. В данной работе будут рассмотрены оба метода.

###### Оглавление

1. Ведение ……………………………………………………………4
2. Анализ литературы………………………………………………..5
3. Используемые обозначения……………………………………...6
4. Глава 1. Решаемые проблемы…………………………………....7
5. Глава 2. Методы построения криптосистем с временным раскрытием…………………………………………………………8
   1. «Шарады» с временным замком (time – lock puzzles) ……9
   2. Используемые понятия…………………………………….12
   3. Схема с использованием доверенных агентов…………14
6. Заключение…………………………………………………………19
7. Список литературы………………………………………………..20

**Ведение**

Испокон веков не было ценности большей, чем информация. ХХ век - век информатики и информатизации. Технология дает возможность передавать и хранить все большие объемы информации. Это благо имеет и оборотную сторону. Информация становится все более уязвимой по разным причинам:

* возрастающие объемы хранимых и передаваемых данных;

1. расширение круга пользователей, имеющих доступ к ресурсам ЭВМ, программам и данным;

* усложнение режимов эксплуатации вычислительных систем.

# Поэтому все большую важность приобретает проблема защиты информации от несанкционированного доступа при передаче и хранении.

# Сейчас услуги криптографии необходимы почти во всех областях деятельности человека. С развитием прогресса, появляется необходимость решать задачи, которые, совсем недавно, писатели научно-фантастического жанра описывали в своих произведениях.

# Ещё совсем недавно проблема отправки секретных сообщений в будущее волновала только любителей фантастической литературы. Однако сегодня она становится всё более актуальной, в связи с научно-техническим прогрессом.

# Например, известно, что современные технологии позволяют

«замораживать» тело человека до уровня поддержания минимальных функций жизнедеятельности организма и надёжно хранить определённое время. Услуги подобного рода, в ряде случаев, являются единственной надеждой людей, страдающих тяжелыми заболеваниями. Следует учесть, что с ростом продолжительности жизни человека, ухудшения экологической обстановки, количество таких болезней сильно увеличилось в наши дни. Например, если человек мучается болезнью, которая сегодня неизлечима, но имеет достаточно средств, то он может «отправить себя в будущее», воспользовавшись услугами криогенных депозитариев. Именно здесь и появляется проблема надёжности шифрования сообщений. Очевидно, что «замороженный» человек не является дееспособным и не может отвечать за какую-либо секретную информацию. Вместе с тем, он должен иметь возможность оставить некоторые распоряжения ( установить номер счёта, точное содержание завещания, если произойдёт несчастный случай, историю болезни и т. п.), которые гарантированно были бы выполнены только по истечении заданного срока – не раньше и не позже.

Эту проблему позволяют решить криптографические методы обеспечения конфиденциальности и целостности при заданном времени дешифрования сообщения на неизвестном ключе.

**Анализ литературы.**

Публикаций на тему данной работы очень мало. Это обусловлено, в первую очередь тем, что ранее не существовало сильной необходимости в шифровке сообщений на такой длинный период времени. Первым, кто обратился к сообществу Intеrnet с предложением рассмотреть подобную задачу в связи с потребностями людей, пользующихся услугами криогенных депозитариев, был Тимоти Мэй. Именно он и предложил использовать в криптографической схеме довереннях агентов. Как ответ на этот запрос и возникли рассматриваемые схемы, которые были разработаны известными криптографами Рональдом Л. Ривестом , Ади Шамиром и Девидом А. Вагнером. Насколько мне известно, других криптографических схем, направленых именно на решение этой проблеммы нет. Никакой литературы, кроме ответной статьи Ривеста, Шамира, Вагнера и дополнения этой же статьи в журнале “Конфидент”5’96 мне найти не удалось. Следует учесть, что предложенные схемы основываются на базовых понятиях криптографии, с которыми можно ознакомиться в любой специализированной литературе.

**Используемые обозначения.**

* **М**-секретное сообщение.
* **К**-секретный ключ, на котором шифруется **М**.
* **t** - время, на которое шифруется **М**.
* **t˚**- текущее время.
* **S -** производительность компьютера.
* **Е**- выбранный алгоритм шифрования.
* **p** ,**q** – простые числа.
* **n** –формируется, как произведение чисел **p** и **q**.
* **f(n)** –формируется, как произведение чисел **(p-1)** и **(q-1)**.
* **d**– количество, доверенных агнтов.
* **i**–номер агента.
* **θ**– порог схемы разделения секрета.
* **Si**,**t**- секретный ключ агента.
* **Di**,**t** – открытый ключ агента.
* **y**j –“тень” ключа **К** , j-огоагента
* **r**j – криптограмма “тени” ключа **К** , j-огоагента
* **F** – односторонняя хеш-функция.

**Глава 1. Решаемые проблемы.**

Кроме сохранения информации о «замороженном», на заданный срок также, необходимо упомянуть и некоторые другие практические приложения криптографии с временным раскрытием:

* участник торгов может пожелать «запечатать» предложение цены с тем, чтобы оно было «распечатано» по завершении торговой сессии;
* домовладелец хочет предоставить держателю закладной возможность осуществлять платежи с использованием зашифрованного цифрового кэша (digital cash) с различными датами дешифрования так, чтобы оплата выполнялась в начале каждого следующего месяца;
* частное лицо может пожелать зашифровать свой дневник так, чтобы он мог быть дешифрован по истечении определённого срока;
* схема шифрования с депонированием ключей (key-escrow scheme) может быть реализована на базе “шарад” с временным замком с тем, чтобы правительственные организации могли получить ключи для расшифровки сообщения только только по истечении определённого периода времени.

В некоторых из вышеперечисленных ситуациях время на которое сообщение остаётся секретным колеблется в рамках 1 года, когда в случае с «замороженным» человеком или шифровкой дневника этот срок ограничивается снизу, как минимум, 2-3 годами , верхняя оценка является вообще неопределённой. Учитывая данное обстоятельство, естественно было бы предложить два метода шифрования сообщения, на не очень длительный период времени 2-3 года и на достаточно длительное время , порядка нескольких десятилетий, что и было сделано. Каждый из методов должен использоваться в соответствующей ситуации.

В основном, криптографическая задача шифрования сообщения рассматривается таким образом, чтобы полученная криптограмма могла быть дешифрована ( в том числе и самим отправителем сообщения) по истечении заданного интервала времени. Атака на подобную криптосистему считается успешной , если удаётся дешифровать сообщение существенно ранее установленного срока. Такой способ защиты, с возможностью раскрытия секретной информации по истечении определённого времени, и называется криптографией с временным раскрытием ( timed - release crypto ).

**Глава 2. Методы построения криптосистем с временным раскрытием.**

Криптосистема с временным раскрытием, предложенная Р.Л.Райвестом, А.Шамиром и Д.А.Вагнером получила название «шарады» с временным замком (time – lock puzzles ). Данный подход к защите информации связан именно с проблемами, возникающими при отправке секретного сообщения в будущее. Его специфика заключается в том , что в отличие от традиционных криптографических методов, предполагающих наличие у получателя сообщения секретного ключа отправителя ( в симметричных криптосистемах) или у отправителя сообщения аутентичного ( подлинного ) открытого ключа получателя ( в асимметричных криптосистемах ) , секретный ключ уничтожается сразу после шифрования и неизвестен как отправителю, так и получателю сообщения.

В настоящее время существует два основных метода построения криптосистем с временным раскрытием :

* «шарады» с временным замком на базе вычислительных проблем с существенно последовательными алгоритмами решения;
* использования доверенных агентов , принимающих на себя обязательство не раскрывать информацию в течение заданного интервала времени.

Очевидно, что при использовании доверенных агентов возникает проблемма надёжности, которая частично может быть решена за счёт применения криптографической техники разделения секрета. Процессорное время необходимое для решения «шарады» зависит от количества и вида применяемого аппаратного обеспечения, а также достижений в области распаралеливания вычислений. Далее будут рассмотрены оба метода.

# **«Шарады» с временным замком (time – lock puzzles)**

Идея заключается в том, что решение «шарады» позволяет получить секретный ключ для дешифрования ранее зашифрованного сообщения. Это значит, что применяя «силовую атаку» (исчерпывающий перебор в ключевом пространстве), злоумышленник сможет раскрыть сообщение, только тогда, когда содержание, раскрытого им сообщения уже не будет актуальным. Как было отмечено выше, сложность ( время ) решения «шарады» существенно зависит от количества вычислительных ресурсов . Таким образом , основная задача при построении любой «шарады» сводится к выбору алгоритма доказуемо – последовательной природы, то есть алгоритма, который не может быть распараллелен в принципе и эффективность ( сложность ) которого существенно не зависит от вложенных в аппаратуру и програмное обеспечение средств. При этом использование нескольких работающих параллельно компьютеров не позволяет ускорить решение «шарады».

Однако такой подход к построению «шарад» не позволяет точно определить время решения, так как использование различных технологических элементов приводит к разбросу производительности конечных аппаратных реализациий. Метод, основаный на использовании доверенных агентов, является более продподчтительным в случае, когда решение должно быть предьявлено точно в указанный срок.

Необходимо подчеркнуть, что предлагаемый метод построения «шарад» не позволяет автоматически получать решение через определённое время, а требует непрерывной работы компьютера в течение заданного времени. Например, решение, рассчитанное на 10 лет, требует непрерывных вычислений в течение всего этого времени. Очевидно, что решение не будет получено через 10 лет, если вычислительный процесс был запущен через 5 лет ( на машине, производительность которой соответствует пятилетней давности ) после того , как сообщение было зашифровано. Таким образом по сравнению с использованием доверенных агентов, метод последовательных ввычислений требует большего количества ресурсов ( для выполнения непрерывных вычислений ) и может эффективно применяться для решения простых «шарад» ( например с временем раскрытия в один месяц ).

Для пояснения задачи рассмотрим следующий пример. Обозначим через **М** сообщение, которое должно быть зашифровано, а через **S** производительность ( дешифрований в секунду ) компьютера. Для шифрования сообщения **М** так, чтобы оно могло быть дешифровано по истечении **Т**  секунд, выберем симметричную криптосистему ( например **RC5** ).И выполним шифрование сообщения на ключе длинной

**K = lg ( 2ST )**

бит. Сохраним криптограмму и уничтожим ключ. После этого применение «силовой атаки» ( исчерпывающего перебора в ключевом пространстве ) позволит найти ключ в среднем за **Т** секунд.

В связи с таким построением возникают две проблемы :

* «силовая атака» допускает тривиальное распараллеливание, в результате чего применение **N** компьютеров позволяет получить результат в **N** раз быстрее
* время **T**  является ожидаемым временем дешифрования; на практике это время может быть существенно больше или меньше, в зависимости от того , в каком порядке проверяются ключи.

То есть, необходимо чтобы, схема была построена таким образом, чтобы распараллелить вычисления не представлялось возможным. Справиться со второй проблеммой при построении схемы не удастся, так как порядок перебора ключей, во всём их множестве, может регулировать только сам злоумышленник, желающий узнать конфиденциальную информацию. Первую проблему можно решить выбрав алгоритм шифрования доказуемо – последовательной природы, что и было сделано.

Рассмотрим метод построения “шарад”, предложенный Р.Л.Райвестом, А.Шамиром, Д.А.Вагнером и основанный на последовательном применении операции возведения в квадрат.

Предположим, Алиса желает зашифровать сообщение **М**, так, чтобы его можно было расшифровать через **Т** секунд.

Для этого Алиса :

* генерирует сооставной модуль **n = pq** как произведение двух простых случайно выбранных чисел **p** и **q**  и вычисляет **f(n) = (p-1) (q-1)**;
* далее вычисляет **t = T S** , где **S** – производительность (число возведений в квадрат по модулю **n**  в секунду ) компьютера, предназначенного для решения шарады;
* генерирует случайный ключ **К** для симметричной криптосистемы, например **RC5** . Ключ должен быть достаточно длинным;
* шифрует **М** на **К** с помощью **RC5** . **C(M) = RC5( K, M )** ;
* случайным образом выбирает **а** по модулю **n** (**1<a<n**), и шифрует секретный ключ **К** таким бразом: **C(К) = K + b** , для чего сначала вычисляет **e = 2t(mod f(n)) (1)**, и затем  **b = ae (mod n) (2)**;
* обьявляет “шараду” в виде набора параметров **(n, a, t, C(K), C(M) )** и стирает переменные ( такие ,как : **p, q, e, b, n** ), созданные в процессе вычислений.

Таким образом, по построению ключ **К** не может быть найден при помощи “силовой атаки”. Поэтому самый быстрый способ решения “шарады” – это вычисление

**b = ae (mod n)**.

При известном **f(n)**  можно быстро вычислить **e**, по формуле **(1)** и затем **b** по формуле **(2)**. Однако известно, что вычисление **f(n)**  по **n**  столь же трудоёмкая задача, что и разложение **n**  на множители. Таким образом, единственный известный в настоящее время способ вычисления **b** ( при правильно выбранных параметрах **p, q, а** ) сводится к последовательному возведению **а**  в кврдрат (**t** раз), причём каждый раз в квадрат возводится предыдущий результат (таким образом исключается распараллеливание вычислений при “силовой атаке” ).

Хотя попытка разложения **n** на множители представляет собой альтернативный метод решения, но при достаточно больших **p**  и **q** такой подход менее эффективен, чем последовательное возведение в квадрат.

Число возведений в квадрат **t** может контролироваться с точностью до операции, следовательно имеется возможность построения “шарад” с различными уровнями сложности решения.

Более важное обстоятельство заключается в том, что алгоритм вычисления **b** по формуле **(2)** является доказуемо – последовательным. Иными словами, алгоритм параллельного вычисления **b** по формуле **(2)**  в настоящее время неизвестен. Возможность распараллеливания существует только для отдельной операции возведения в квадрат, таким образом, в данной ситуации число компьютеров, применяемых для решения, значения не имеет.

Чтобы, сама Алиса могла расшифровать криптограмму ей не нужно хранить в тайне ключ **К** , но необходимо знать секрет **f(n)**, чтобы в заданный момент времени вычислить **e** по формуле **(1)** и **b**  по формуле **(2)**, расшифровать секретный ключ **К** и дешифровать своё сообщение.

Следует учесть что такую схему стоит применять только в случае, когда **Т**  не превышает 5 лет , при выполнении всех условий построения схемы. Такой вывод можно сделать по результатам представленым в статье Ю. Е. Пудовченко «Когда наступит время подбирать ключи», а именно , что через каждые 5 лет производительность комтьютеров возрастает в 10 раз. То есть, если зашифровать сообщение , используя такую схему , на десять лет, то через пять лет «силовая атака» ( на более мощных, соответствующих своему времени машинах ) займёт времени в 10 раз меньше , в нашем случае это составит 1 год. Таким образом всё время секретности данного сообщения составит 6 лет, что намного меньше требуемого срока.

Например, для преодоления этого барьера, можно за **S** (производительность машины ) принять величину, которая , по каким – то соображениям, будет соответствовать времени раскрытия сообщения или использовать ключи такой длины, чтобы энергия, требуемая для вскрытия ( считая, что на один шаг затрачивается минимальный квантовомеханический квант энергии ) превзошла массу солнца или вселенной. Но, тогда длина ключа может так сильно возрасти, что превысит длину самого сообщения. К тому же оценка может оказаться неправильной. Поэтому , наиболее надёжной схемой, для решения задачи отправки сообщения в далёкое будущее , будет являться схема с использованием доверенных агентов.

Используемые понятия

Для того, чтобы более подробно понять схему с использованием доверенных агентов дадим некоторые базовые понятия.

* Криптография с открытым ключом. В схеме с открытым ключом имеется два ключа, открытый [public] и секретный [private, secret], выбранные таким образом, что их последовательное применение к массиву данных оставляет этот массив без изменений. Шифрующая процедура использует открытый ключ, дешифрующая - секретный. Дешифрование кода без знания секретного ключа практически неосуществимо; в частности, практически неразрешима задача вычисления секретного ключа по известному открытому ключу. Основное преимущество криптографии с открытым ключом – упрощенный механизм обмена ключами. При осуществлении коммуникации по каналу связи передается только открытый ключ, что делает возможным использование для этой цели обычного канала и устраняет потребность в специальном защищенном канале для передачи ключа.
* Цифровая подпись. Цифровой подписью называют блок данных, сгенерированный с использованием некоторого секретного ключа. При этом с помощью открытого ключа можно проверить, что данные были действительно сгенерированы с помощью этого секретного ключа. Это делается таким образом : отправитель шифрует своё сообщение на своём секретном ключе и на открытом получателя, после чего отсылает криптограмму получателю; получатель, в свою очередь дешифрует полученую криптограмму на своём секретном ключе и на открытом отправителя; после этих манипуляций у получателя должно получиться искомое сообщение от отправителя. Цифровые подписи используются для того, чтобы подтвердить, что сообщение пришло действительно от данного отправителя (в предположении, что лишь отправитель обладает секретным ключом, соответствующим его открытому ключу). Также подписи используются для проставления штампа времени (timestamp) на документах: сторона, которой мы доверяем, подписывает документ со штампом времени с помошью своего секретного ключа и, таким образом, подтверждает, что документ уже существовал в момент, объявленный в штампе времени.
* Односторонняя хеш функция. Такая функция не поддаётся обращению - нельзя узнать её аргумент, зная результат. Также существует односторонняя функция с потайным ходом ("лазейкой"). Идея состоит в том, чтобы построить функцию, обратить которую можно только зная некоторую "лазейку" - секретный ключ.
* RC5. RC5 это довольно-таки быстрый блочный шифр разработанный Ривестом для RSA Data Security. Этот алгаритм параметричен, т.е. с пременным размером блока, длинной ключа и переменным числом проходов. Размер блока может быть 32, 64, или 128 битов. Количество проходов в промежутке от 0 до 2048 бит. Параметричность такого рода дает гибкость и эффективность шифрования. RC5 состоит из ввода ключа (key expansion), шифрования и дешифрования. При вводе ключа вводятся также количество проходов, размер блока и т.д. Шифрование состоит из 3 примитвных операций : сложения, побитового XOR и чередования (rotation). Исключительная простота RC5 делает его простым в использовании, RC5 текст, также как и RSA, может быть дописан в конец письма в зашифрованном виде. Безопасность RC5 основывается на зависящем от данных чередованием и смешиванием результатов различных операций. RC5 с размером блока 64 бита и 12 или более проходов обеспечивает хорошую стойкость против дифференциального и линейного криптанализов.
* Схема разделения секрета. Математика разделения секрета достаточно сложна, и является темой отдельного разговора, поэтому дадим неформальное определение данного понятия. Схемой разделения секрета называется такая схема, которая позволяет «распределить» секрет между **n** участниками таким образом, чтобы заранее заданные разрешённые множества ( множества “теней секрета” ) участников могли однозначно восстановить секрет, а неразрешённые - не получили никакой дополнительной информации о возможном значении секрета. Пороговая схема разделения секрета, (**n** , **θ** ) – схема, позволяет восстанавливать секрет, если разрешённым множеством является любое множество из **θ** или более “теней секрета”.

# **Схема с использованием доверенных агентов**.

Данная схема является более предпочтительной для сохранения секретности сообщения долгое время, именно эта схема должна использоваться в ситуации с «замороженными» людьми, так как время пребывания в летаргическом сне ограничивается ни несколькими годами, а несколькими десятилетиями.

# Итак, подход данного метода заключается в использовании доверенных агентов для хранения сообщения **М**в течение заданного интервала времени **t** . Для большей надёжности схемы шифрования , ключ **К**, на котором собираемся зашифровать сообщение **М**поделим на **d**“теней”, воспользовавшись техникой разделения секрета предложенной А.Шамиром, и распределим “ тени “ секретного ключа среди нескольких агентов, заручившись с их стороны обязательством, что соответствующие фрагменты будут предъявлены по истечении времени **t** . Заметим, что используемая техника разделения секрета обладает избыточностью и позволяет восстанавливать секретный ключ в случае, когда некоторые агенты не в состоянии выполнять свои функции. Тогда криптограмма ***С =Е (* К,М *)*** может быть помещена в общедоступное место с тем, чтобы можно было получить сообщение **М**(воостановив ключ **К**и дешифровав сообщение ***С*** ) по истечении времени **t** .

Итак Райвист, Шамир и Вагнер предложили альтернативный метод со следующими свойствами:

* Агенты не хранят ключи, применяемые в схеме шифрования.. Каждый агент хранит “тень” ключа, получаемую с помощью техники разделения секрета. Необходимый объём памяти, выделенной под “тень” ключа, фиксирован и не зависит от числа секретных компонент, доверенных агенту.
* Сначала, каждый агент формирует свой секретный ключ, который он раскроет в момент времени **t** . Далее, с помощью этого ключа, агент должен будет подтверждать своё существование и свою личность.
* Основная задача агента : периодически ( например, в начале каждого часа ) раскрывать ранее секретное значение - **Si**,**t** ( получать новое значение хеш-функции) и заверять раскрытый секрет цифровой подписью на своём секретном ключе, который раскрывается в заданный момент времени. Под выражением ранее секретное значение - **Si**,**t** понимается старый результат хеш-функции.
* Также, агент должен отвечать на вопросы вида : “Для заданных значений **у** и **t** возвратите значение функции **E( y** **, Si**,**t )** – результат шифрования **у** на секретном ключе **Si**,**t** , который Вы предполагаете раскрыть в момент времени **t**”. Предполагается, что используемая для шифрования криптосистема устойчива к атаке на открытом тексте, то есть злоумышленник не сможет восстановить секретный ключ **Si**,**t** , располагая результатами шифрования различных **у** –ов на фиксированном ключе **Si**,**t**. Сформировать сообщение, подписать его на открытом ключе пользователя и на закрытом агента. Сформированное сообщение должно содержать номер агента, текущее время, время **t,** раскрытый и подписаный секрет **Si**,**t** и значение функции **E( y** **, Si**,**t )**.
* каждый агент всегда подписывает свои секретный и открытый ключи.

Итак, схема с использованием доверенных агентов выглядит так : сначала на случайно выбраном ключе ***К***  при помощи симметричной криптосистемы пользователь шифрует сообщение **М**, и получает

***С=(* К, М *)***.

Далее, выбрав **d**агентов **i**1, **i2,**… **,i**d пользователь публикует :

***( С,* i**1***,* i2*,***… ***,*i**d. ***,* r**1***,* r**rrRRgКкпрпл**2*,***… ***,*r**d ***)*** ,

где **r**1***,* r**rrRRgКкпрпл**2*,***… ***,*r**d -**d** криптограмм “теней” ключа ***К*** . “Тени” получены по схеме разделения секрета, позволяющие восстанавливать ключ **К**  в момент времени **t** ,после того, как агенты раскроют свои секреты. Размер “тени” ключа фиксирован и не зависит от числа секретных компонент, доверенных агенту. Также пользователь может установить порог **θ** (**0** < **θ** ≤**d**) такой, что восстановление ключа **К**будет возможно только при **θ**  или большем числе “теней” ключа **К** . Для этого пользователь просто разбивает ключ **К** на **d**“теней” по любой схеме разделения секрета с порогом **θ**.

После шифрования ключ **К** удаляется.

Далее пользователь просит агента возвратить ему криптограмму, соответствующей агенту, «тени» ключа на секрете агента . Тогда

**r**j = **E( y**j **, Si**,**t )** ,

где **(y**1, **y**2 , …. ,**y**d**)**   **d** “теней” ключа ***К*** .

После, пользователь генерирует составной модуль

**n = pq** ,

как произведение двух простых случайно выбранных чисел **p** и **q**  . После чего вычисляет

**f(n) = (p-1) (q-1)** и

**e = 2t(mod f(n))**.

Пара чисел **(e, n)** и будет являться открытым ключём пользователя.

Итак, данная схема представляет собой асимметричную систему с открытым ключом. Каждый из участников данной схемы имеет свой открытый и закрытый ключ, закрытые ключи агентов - их секреты, у пользователя количество закрытых ключей может равняться количеству агентов, так, чтобы понимать их сообщения или же пользователь должен иметь один универсальный закрытый ключ, который он смог бы применять для дешифрования всех получаемых им сообщений.

Таким образом обязанности агента заключаются в следующем:

* переодически раскрывать ранее секретное значение –получать новое значение хеш-функции. Обозначим за **Si**j,**t**секретное значение раскрываемое **i**j агентом в момент времени **t** . Последовательность секретов, раскрытых одним агентом, не зависит от последовательности секретов раскрытых другими агентами. Такая последовательность обладает следующим свойством : из каждого **Si**j,**t΄**можно просто вычислить **Si**j,**t**для всех **t΄** ≤ **t** .Секрет, раскрываемый агентом может быть использован для вычисления всех ранее раскрываемых секретов в силу следующего рекуррентного уравнения : **Si,(t-1) = F ( Si,t ) (3),** где **f** - некоторая односторонняя хеш-функция (**Si,(t-1)** –новый секрет, **Si**, **t** – старый, индекс **(t-1)** - означаетвремя которое осталось до раскрытия секрета, **1** применяемая в записи, условна и означает время, которое прошло с прошлого раскрытия секрета до текущего момента времени) Поскольку функция **F** является односторонней, раскрытие **Si,t** не позволяет раскрыть прошлые секреты **Si,t**. (В противном случае злоумышленник мог бы вычислить последовательность секретов по формуле **(3),** начиная с некоторой точки в будущем, или выбрать функцию **F** с “лазейкой”, так чтобы только он смог вычислить **Si,t** - секретный ключ агента из **Si,(t-1)** ). Каждый раскрытый секрет агент должен подписать на своём секретном ключе. Новый, полученый секрет объявляется открытым и используется для общения агента с пользователем то есть, чтобы пользователь мог удостовериться в личности агента .
* дешифровать, на своём секретном ключе, сообщение пользователя вида **(y, t, (e, n))**  зашифрованные на открытом ключе агента, где **y** - любое сообщение пользователя не содержащее никакой секретной информации.
* шифровать сообщение **y** на секрете **Si,t** ,раскрываемого агентом в момент времени **t** . Таким образом имеем криптограмму **E(Si,t** ,**y)**.
* сформировать сообщение вида :

**m=( i**, **t**, **t˚ , E(Si**,**t** ,**y))**,

где **i** - индекс агента, **t** – время раскрытия в будущем,  **t˚** - текущее время ( по

часам агента). Это сообщение шифруется на открытом ключе пользователя

**(e, n)**  и подписывается на полученном секрете агента - **Si**,**t˚** :

**E( E( m , (e, n) ) , Si**,**t˚ )**.

Выполнение неравенства **t > t˚** не требуется, однако

рассматривается как норма.

* раскрыть подписаный секрет **Si**,**t** в момент времени **t** .

Сразу после шифровки “тени” основного ключа **К** агент должен раскрыть свой секрет – получить значение хеш-функции, проделать все полагающиеся манипуляции и объявить пользователю свой открытый ключ – раскрытый секрет **Si**,**t** , подписаный на своём секретном ключе. Изначально аргументом используемой хеш-функции является секретный ключ агента. Таким образом агент уже выполнил свои обязанности один раз и схема должна сработать. Как видно по построению, сколько раз или когда именно агент будет раскрывать свой секрет – не важно. Также нет никакой зависимости между количеством раскрытия секрета разными агентами. Вообще, агент может учавствовать во всей схеме два раза : первый – зашифровать “тень”, подписать новый секрет и т.д. ; второй – раскрыть свой секретный ключ в заданное время.

Столь простой набор функций допускает простую реализацию в виде устойчивого к вскрытию, секретного, компактного и надежного устройства. Роль пользователя в такой схеме осуществляет сервер, который контролирует все манипуляции агентов при раскрытии секрета. Сообщение **у** может либо заранее задаваться администратором, либо генерироваться самим сервером, так как никакой смысловой нагрузки не несёт, а служит для аутентификации агента. Как было сказано выше, каждый агент имеет свой секретный ключ, который раскрывается в момент времени **t**, этот ключ учавствует в схеме в нескольких случаях – когда шифруется “тень” основного ключа **К**, когда агент дешифрует сообщение пользователя , подписывает сообщение **у**  и раскрываемый секрет. Новый раскрытый и подписанный секрет, учавствует в диалоге пользователя и агента, и является открытым для пользователя, чтобы пользователь мог дешифровать ответное сообщение.

Работоспособность такой схемы достигается за счет применения пороговой схемы разделения секрета, обладающей избыточностью и позволяющей восстанавливать сообщение в случае, когда некоторые агенты не в состоянии выполнять свои функции. Если в системе существует не менее **θ** агентов, сообщение с гарантией будет восстановлено в указанные сроки, в противном случае будет восстановлено в будущем.

Описанная схема использования доверенных агентов не “верифицируема” в том смысле, что по опубликованым данным невозможно заранее принять решение о восстановлении сообщения. Сообщение **М** может быть восстановлено только после раскрытия агентами своих секретов, дешифрования **r**j с целью получения **y**j для дальнейшего восстановления секретного ключа **К** и дешифрования ***С***. Для

решения проблемы необходимо применять “верифицируемые” схемы разделения секрета.

Для уменьшения потока обращений к пользователю рекомендуется поступать таким образом : с самого начала, агенты формируют свои открытые и секретные ключи **Di**,**t** и **Si**,**t** соответственно, где **Di**,**t = f (Si**,**t)**. Для большей надёжности агенты всегда подписывают свои открытые и закрытые ключи. Агент делает доступным ключ **Di**,**t** для пользователя. Тогда пользователь сам может проделать все необходимые манипуляции с сообщением **y** используя вместо секретного ключа агента, подписаный агентом открытый ключ **Di**,**t**. Таким образом в обязанности агента будет входить :

* периодическое раскрытие своего открытого ключа **Di**,**t** – получение нового значения хеш – функции.
* подписывание полученного значения на своём секретном ключе **Si**,**t**.
* раскрыть подписаный секрет **Si**,**t** в момент времени **t** .

Таким образом пользователь будет уверен, в личности агента и в том, что агент существует.

Обе приведённые схемы с использованием доверенных агентов будут выполнять свою задачу – сохранять секретное сообщение одинакого. Разница заключается только в том, что во второй схеме у агента намного меньше обязанностей : агенту не так часто надо раскрывать свой секрет и время выполнения обязательных манипуляций намного меньше. На практике оба метода могут использоваться вместе. Например, агенты, которые находятся далеко от пользователя ( то есть передача сообщения между ними занимает значительное количество времени ) могут использовать вторую схему, тогда как, агенты, находяциеся не так далеко могут использовать первую схему.

На мой взгляд, по истечении хотя бы половины заданного срока можно добиться раскрытия сообщения применяя “грубую силу” к самой криптограмме ***С*** ( то есть простой перебор ключей **К** ), к тому времени производительность машин может сильно возрасти и время ожидания при атаке “грубая сила” значительно снизится . Для преодоления этого препядствия, как мне кажется, можно внести некоторые изменения. Например потребовать, чтобы в заданное время **t** часть секретов была раскрыта агентами на определённых терминалах ( например, находящихся в здании криогенного депозитария, рядом с сервером ) или в определённой последовательности.

Для преодоления договорённости между агентами , на мой взгляд, необходимо увеличить число агентов, причём некоторым раздать “тени” искомого ключа, а некоторым “белый шум” , таким образом агентам будет очень сложно сговориться. Но вообще , считается, что агент - человек заслуживающии доверия.

Учитывая всё вышесказанное следует отметить, что такой подход требует большего объёма памяти для хранения списка открытых ключей, которые будут использованы в будущем, и списка раскрытых ранее секретных ключей. Так , необходимый объём памяти на пятьдесят лет (из расчёта один ключ на каждый день ) при размере ключа 200 бит составит 3,5 Мбайт.

**Заключение.**

Эта работа посвящена криптографическим схемам сохранения секретного сообщения на долгое время (от месяца до нескольких десятков лет). Существует две схемы подобного типа : «Шарады» с временным замком (time – lock puzzles) и схема с использованием доверенных агентов. Первая схема заключается в том, что сообщение кодируется ключом, который неизвестен как отправителю, так и получателю сообщения и который не будет раскрыт в течение времени секретности данного сообщения, за счёт своей длины. Недостаток такого метода, заключается в небольшом периоде времни секретности сообщения, из за прогресса вычислительных мощностей компьютеров. Наиболее интересна вторая схема, так как позволяет оставить сообщение секретным на более долгий срок (этот срок ограничивается длиной жизни как минимум **θ** агентов) . Суть метода заключается в том, что “тени” ключа, на котором шифруется секретное сообщения, распределяются между доверенными агентами. Агенты шифруют свои “тени” с помощью выбранного алгоритма на своём секретном ключе, который они раскроют только по истечении заданного времени. Каждый агент должен периодически подтверждать своё существование и личность с помощью определённых действий. В заданный момент времени агенты должны раскрыть свои секреты, дешифровать соответствующие “тени”, “собрать” из полученных “теней” искомый ключ и дешифровать сообщение. Стойкость такой схемы обеспечивает не только длина ключа но и не “верифицируемость” , то есть принять какое либо решение о восстановлении сообщения, по опубликованным данным, невозможно. Все действия агентов, по заверению своей личности, выполняются в строгом порядке и регулируются “пользователем” – сервером.

Насколько мне известно, других криптографических схем, направленых именно на решение этой проблеммы нет. Сейчас существуют криптографические схемы, которые решают проблему длительной секретности сообщения, за счёт длины ключа. На сегодняшний день, рекомендованная длина ключа составляет около 4000 бит.

На мой взгляд, в будущем, большую популярность приобретут именно схемы с доверенными агентами. Со временем, мощности машин будут увеличиваться, а значит многие задачи ( разложение большого целого числа на простые множители, вычисление дискретного логарифма по модулю большого целого числа) будут решаться намного быстрее и срок секретности сообщения будет также уменьшаться. Использование доверенных агентов или иерархий доверия, основаных на цифровой подписи, в такой ситуации будет оптимальным решением проблемы секретности собщений. Поэтому , как мне кажется, развиваться будут именно криптографические схемы “с доверием”.

**Литература.**

1. Ronald L. Rivest, Adi Shamir and David A.Wagner “Time – lock puzzle and time-release Crypto”. (http://theory.lcs.mit.edu/~rivest - 1999г. на сегодняшний день сайт обновлён и данной публикации не содержит)
2. А.Л.Чмора, “Шифровка в будущее” , “Конфидент”5’96
3. Ю. Е. Пудовченко, “Когда наступит время подбирать ключи“
4. Перевод статьи Tatu Ylonen “Introduction to Cryptography“
5. Г.А.Кабатянский, “Математика разделения секрета”.

Особое спасибо web-мастеру страницы http://www.ssl.stu.neva.ru/ Александру Ежову за некоторые идеи для решения этой задачи и статью Ю. Е. Пудовченко, “Когда наступит время подбирать ключи“.