Лекция 7. Требования к современным криптографическим системам. Системы шифрования

Цель лекции: изучить требования к современным криптографическим системам, шифры на основе различных сетей, а также рассмотреть хэш-функции и схемы ЭЦП.

План лекции:

- Требования к современным криптографическим системам
- Шифры на основе сети Фейстеля
- Шифры на основе SP-сети
- Асимметричные системы шифрования
- Схемы электронной цифровой подписи
- Хэш-функции

Первый вопрос, которому мы сегодня посвятим свое внимание, это **требования к современным криптографическим системам**. Прежде всего стоит сформулировать такое понятие, как блочный шифр. Блочный шифр — это шифр, обрабатывающий за каждое применение операции зашифрования группу символов открытого текста фиксированной длины. Например, это шифр Плейфера или шифр два квадрата. В обоих из них мы зашифровывали по паре символов открытого текста за одно применение операции зашифрования. В двоичном представлении блочный шифр отображает блок открытого текста некой фиксированной длины и ключ длины, которая может отличаться от длины блока в блок шифр-текста длиной, как правило, совпадающей с длиной блока открытого текста.

Альтернативой блочному шифру является шифр поточный. Это шифр, обрабатывающий за каждое применение операции зашифрования один символ открытого текста. В современном понимании, как правило, блочный шифр работает с двоичным представлением информации, а поточный — с символами открытого текста, то есть непосредственно с буквами. Поточные шифры, как правило, строятся в настоящее время на основе идеи Шеннона, то есть их основная идея заключается в том, чтобы, подобно шифру Виженера или шифру Вернама, накладывать некую гамму на открытый текст. А для того чтобы шифр при этом получился стойкий, гамма должна обладать некоторыми свойствами, сходными со свойствами действительно случайной двоичной последовательности. Поэтому основа поточных шифров — это в основном некий алгоритм выработки этой самой двоичной гаммы. У поточного шифра, как правило, есть то неудобство, такая уязвимость, которая заключается в том, что поскольку зашифрование происходит посимвольно, то, зная пару открытый текст и шифр-текст, противник может без особого труда определить верный

ключ зашифрования и далее использовать его для навязывания ложного текста, ложного сообщения.

Поэтому в современном мире особое распространение, особым вниманием пользуются именно блочные шифры, в которых можно реализовать операцию перестановки и, таким образом, разрушить связи, которые наличествуют между символами открытого текста, разрушить соседство символов в открытом тексте и добиться так называемого «лавинного эффекта», то есть перемешивания символов шифр-текста на основе небольшого преобразования открытого текста. Лавинный эффект заключается в том, что если мы вносим незначительные изменения в открытый текст, шифр-текст должен измениться значительно. Существуют математические оценки того, в какой мере он должен измениться, чтобы шифр оставался надежным и стойким к различным атакам, но пока остановимся просто на мысли, что именно блочные шифры заслуживают наиболее пристального внимания в нашей лекции.

Какие требования существуют в современном мире к шифрам вообще, не только к блочным?

- Прежде всего информация должна обрабатываться в цифровом, как правило, в двоичном представлении. То есть шифр должен быть предназначен для работы с информацией, хранящейся в вычислительных устройствах, а не для ручного использования с помощью ручки и бумаги, например.
- Следующее требование: шифр должен быть стойким к дешифрованию при помощи вычислительной техники. История с шифровальными машинами, в частности с шифровальной машиной «Энигма» в годы Второй мировой войны, продемонстрировала, что современные криптоаналитики для атак на различные шифры используют сложные вычислительные устройства, которые могут проводить огромное количество операций ежесекундно и, таким образом, перебирать огромное количество возможных вариантов, которые являются секретными в шифре, и поэтому в связи с развитием вычислительной техники, с построением суперкомпьютеров, с появлением различных правительственных организаций в некоторых государствах, которые занимаются поиском уязвимости в различных шифрах, для современного шифра необходимым является требование устойчивости к различным атакам, осуществляемых при помощи вычислительной техники.
- Следующим требованием является требование отсутствия необходимости использовать длинный двоичный ключ. Это требование следует как раз из того неудобства, которое следует из схемы Шеннона. Для идеального шифра требуется длинный двоичный ключ, то есть не уступающий по длине сообщению, вообще говоря, бесконечный в том смысле, что этот ключ может использоваться для зашифрования сколь угодно длинного текста, при этом в нем нет циклов, повторов, он не имеет каких-то явно выраженных структур, он

строится случайно и равновероятно или максимально приближается по своим статистическим характеристикам к таковой последовательности. Так вот, в действительно используемых на практике современных криптосистемах, в большинстве из них, в тех, которые используются в коммерческих задачах, тех, которые используются в обеспечении некой оперативной связи, для удобства управления ключами вводится требование отказа от этих длинных двоичных ключей. Должно быть возможно обходиться достаточно коротким, по сравнению с длиной шифруемого сообщения, ключом.

- Следующее требование, которое иногда упоминается, это возможность стандартизации алгоритмов. То есть желательно чтобы весь алгоритм был максимально описан, детализирован, раскрыт, его мог бы реализовать аппаратно либо программно, вообще говоря, любой желающий, и все абоненты, которые узнают о том, что их получатель сообщений или отправитель используют такой-то алгоритм, были бы способны адекватно принять или отправить ему электронные сообщения. Кроме того, стандартизация алгоритмов позволяет, только зная какой алгоритм использует другой абонент, проводить оценки надежности. Если мы говорим о том, что там реализован некий шифр и он реализован корректно, есть некое заключение, некая лицензия, которая подтверждает, что шифр реализован корректно, мы можем быть уверены в некоем уровне надежности, которая обеспечивает как раз этот стандарт.
- Еще одно требование — устойчивость к навязыванию сообщения. Как раз то требование, которое следует, в общем-то, из уязвимости поточных шифров. Желательно, чтобы злоумышленник мог бы потратить максимальное количество усилий и времени на то, чтобы определить вот этот самый короткий ключ, который используется в блочном алгоритме шифрования, прежде чем он смог бы использовать его вторично для подделки какого-то сообщения. То есть даже если он при помощи какой-то разведки, шпионской деятельности, получил открытый текст и перехватил по открытому каналу связи шифр-текст, имеет эту пару x и y, вспоминая те обозначения, которые мы ввели для криптографических систем в предыдущей лекции, он бы затратил максимальные усилия на то, чтобы, зная эту пару, получить значения ключа. Вот такое требование тоже устанавливается для современных криптографических систем.

Как же строятся блочные шифры, которые удовлетворяют большинству или всем перечисленным выше требованиям? На настоящее время рассматриваются два действующих и один перспективный подходов к построению блочных шифров. Два распространенных действительно и вошедших в большинство стандартов различных государств подходов, разработаны в ходе так называемого проекта «Люцифер», реализованного в 1970-х годах компанией IBM. Эти подходы называются «Сеть Фейстеля» и «SP-сеть». Перспективным направлением является использование так называемой

«функции губки», которая имеет переменную длину входного блока и переменную длину выходного блока, а также достаточно сложный алгоритм, позволяющий все перечисленные требования реализовывать.

Сеть Фейстеля названа в честь Хорста Фейстеля, американского учёного, который принял участие в проекте Lucifer компании IBM. Он автор схемы построения шифров, названной сетью Фейстеля в его честь. Кроме того, он внёс значительный вклад в разработку первого американского стандарта шифрования DES, Digital Encryption Standard, цифровой стандарт шифрования.

Сеть Фейстеля — это один из методов построения стойких блочных шифров, и этот метод основан на том, что при образовании шифруемого блока некоторые функции повторяются многократно, то есть реализуются так называемые раунды, каждый из которых зависит от ключа, раундового ключа каждой итерации. То есть из вот этого не очень длинного ключа, который пользователь вводит в алгоритм шифрования, развёртывается необходимое количество раундовых ключей, каждоый из которых может быть даже длиннее входного ключа, это не исключается данной моделью. То есть из вот этого не очень длинного ключа с помощью функции развёртки развёртывается ключ сколь угодно требуемой длины.

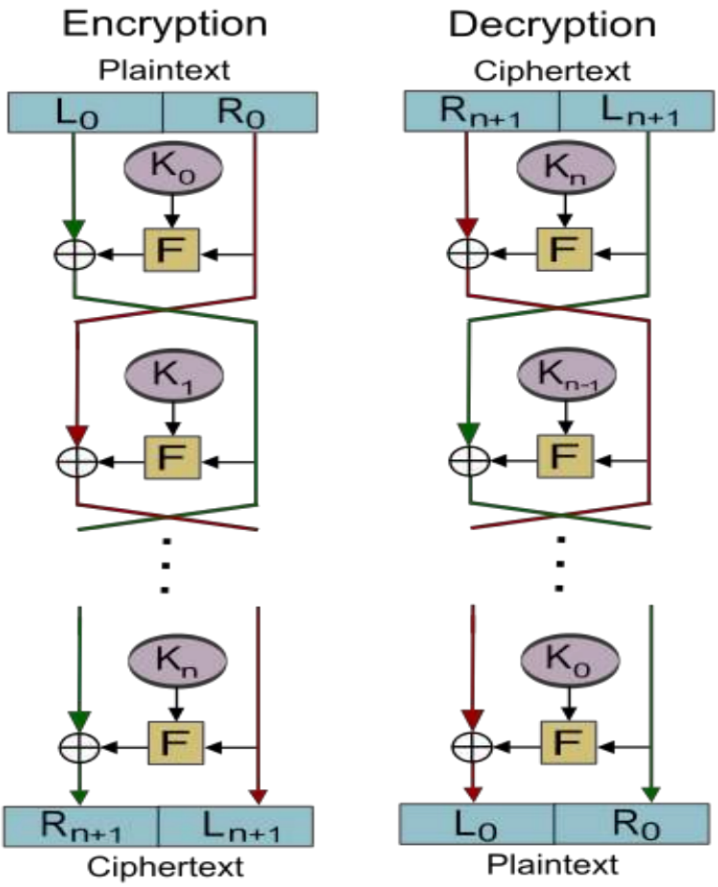

Рисунок 1 Конструкция блочного шифра на основе сетей Фейстеля

Принципиальная модель схемы Фейстеля заключается в следующем: блок открытого текста, поступающий на вход функции зашифрования, делится на два подблока — левый и правый. Далее левый подблок поступает на функцию усложнения, так называемую функцию Фейстеля, которая как раз-таки зависит от раундового ключа. После чего результат применения этой функции складывается по модулю два с правым подблоком. После чего результат этого сложения становится левым подблоком следующего раунда, а левый подблок данного раунда в неизменном виде занимает место правого подблока следующего раунда. Происходит нужное количество итераций, то есть реализуется нужное количество раундов, после чего левый и правый блоки, полученные в результате применения последнего раунда, в результате реализации последнего раунда, предъявляются как зашифрованный текст.

Рассмотрим некоторое количество шифров, которые в различное время являлись действующими государственными стандартами Соединённых Штатов Америки и Российской Федерации, построенные на сети Фейстеля. Первый пример — это американский шифр DES, тот самый, к созданию которого был причастен Хорст Фейстель, который принял участие в проекте Lucifer. Данный алгоритм шифрования принят в качестве стандарта в Соединённых Штатах в 1977 году. Это реализация сети Фейстеля с размером входного блока 64 бита, с размером ключа 56 бит, то есть первого исходного ключа, и с реализацией 16 раундов. Здесь исходный блок делится на два подблока. Его правая часть попадает на функцию фейстеля, после чего складывается по модулю два с левым подблоком, после чего подблоки меняются местами. Реализуется 16 раундов, и на этом последний раунд завершается. Помимо собственно операции обработки в раундах в данном шифре реализуется начальная и конечная перестановки, из которых первая предваряет первый раунд, а вторая применяется после завершения шестнадцатого раунда, после чего результат конечной перестановки предъявляется в качестве зашифрованного текста. Функция Фейстеля в данном шифре выглядит следующим образом. Поступивший на вход один из подблоков длинной 32 бита изначально расширяется с помощью так называемой функции расширения до двоичной строки длиной 48 бит. После чего эта строка складывается по модулю два с раундовым ключом, а затем делится на восемь подстрок, каждая длиной в восемь бит. Каждая из таких строк попадает на блоки замены, S-блоки так называемые. В результате обработки каждым их таких блоков на выходе получается строка снова четыре бита, после чего эти строки сцепляются в одну итоговую строку снова длинной 32 бита. Далее над этой строкой осуществляется перестановка битов или так называемый P-блок применяется. И результат этого применения выходит из функции Фейстеля как результат обработки.

Ввиду простоты операций сеть Фейстеля легко реализовать как программно, так и аппаратно. Большинство современных блочных шифров (DES, RC2, RC5, RC6, Blowfish, FEAL, CAST-128, TEA, XTEA, XXTEA и др.) использует сеть Фейстеля в качестве основы. Альтернативой сети Фейстеля является подстановочно-перестановочная сеть (AES и др.).

Шифры на основе SP-сети

Альтернативой сети Фейстеля как подхода к построению стойких блочных шифров является так называемая SP-сеть. SP означает Substitution Permutation, то есть перестановочно-подстановочная сеть в дословном переводе, или сеть, которая реализует замену и перестановку.

Данный метод основан на комбинировании нескольких слоев, преобразовании шифруемого блока открытого текста, каждый из которых реализует либо перестановки либо замены, некоторые из которых могут зависеть от ключа, могут все, могут некоторая часть. Это остается на усмотрение автора конкретной реализации. Принципиальный момент — в перестановочно-подстановочной сети, или в SP-сети, ограничений на этот параметр не накладывает. Кроме того, возможно многократное использование пары слоев на основе раундовых ключей. Возможно построение одного слоя перестановки и одного слоя замены, а далее многократное повторное использование этих слоев каждый раз с новым раундовым ключом, то есть по сути, применение одних и тех же операций, только зависящих от все время нового параметра.

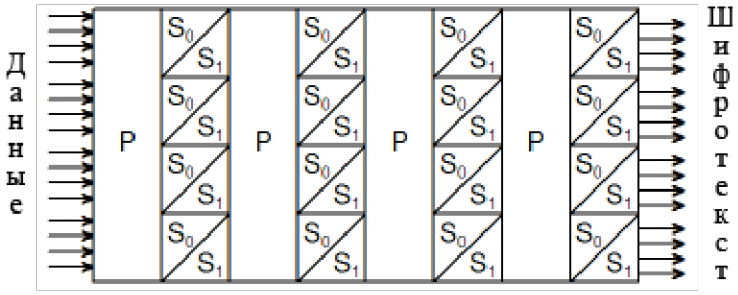

Рисунок 2 Упрощённая схема S- и P-слоёв в алгоритме «Люцифер»

SP-сеть — разновидность блочного шифра, предложенная в 1971 году Хорстом Фейстелем. В простейшем варианте представляет собой «сэндвич» из слоёв двух типов, используемых многократно по очереди. Первый тип слоя — P-слой, состоящий из P-блока большой разрядности, за ним идёт второй тип слоя — S-слой, представляющий собой большое количество S-блоков малой разрядности, потом опять P-слой и т. д. Первым криптографическим алгоритмом на основе SP-сети был «Люцифер» (1971). В настоящее время из алгоритмов на основе SP-сетей широко используется AES (Rijndael). Альтернативой SP-сетям являются сети Фейстеля.

В современных алгоритмах вместо S- и P-блоков используются различные математические или логические функции. Любая двоичная функция может быть сведена к S-блоку, некоторые функции — к P-блоку. Например, к P-блоку сводится циклический сдвиг, сам P-блок является частным случаем S-блока. Такие функции, как правило, легко реализуются в аппаратуре, обеспечивая при этом хорошую криптостойкость.

Одним из ярких представителей шифров, основанных на такой сети, является действующий стандарт Соединенных Штатов, стандарт AES, Advanced Encryption Standard. Для выбора такого шифра, который бы впоследствии стал таким стандартом, был проведен конкурс, на который было представлено много разных шифров, ну и в результате отбора экспертами был выбран шифра под названием Rijndael, который и, собственно, был включен в данный стандарт. Данный шифр основан на SP-сети, является алгоритм блочного шифрования, его размер блока равен 128 битам, размер ключа выбирается пользователем, либо 128 либо 192 либо 256 бит, а в зависимости от длины ключа выбирается 10, 12 либо 14 раундов. Чем длиннее ключ, тем больше раундов применяется, поскольку считается, что с более длинным ключом проще проводить криптоанализ, то есть проще предпринимать попытки по восстановлению исходного ключа на основе полученного шифра текста. Данный шифр устроен следующим образом: открытый текст в виде матрицы байт подается на операцию двоичного сложения с раундовым ключом. Открытый текст в виде матрицы байт размера 4 на 4, то есть 16 байт, 16 * 8 — размер входного блока — подается на операцию сложения по модулю 2 с раундовым ключом. Затем результат операции сложения по модулю 2 подается уже на первый раундовый слой, это слой S, слой замены. Результат затем подается на слой P, который реализует сдвиг, затем на еще один слой P, который переставляет столбцы в этой матрице, после чего прохождение шифруемого блока через данные три слоя повторяется количество раз, задаваемого раундами, и после чего при наступлении последнего раунда результат прохождения шифруемого блока через слой P, то есть слой перестановки, выдается в качестве шифр-текста, и на этом работа шифра завершается.

Другим алгоритмом блочного шифрования, который основан на SP-сети, является другой действующий стандарт Российской Федерации, вступивший в действие с января 2016 года, под названием «Кузнечик». В отличие от ситуации с Соединенными Штатами, где действующим является только шифр на основе SP-сети, в нынешнем стандарте 2015 года, в том самом, который активен с 2016 года, оба шифра, и уже рассмотренный «Магма», и новый шифр «Кузнечик» представлены как два равноправных блочных шифра, с некоторыми ограничениями по тому, в каких ситуациях их можно использовать. Но они оба описаны в стандарте, и оба считаются действующими. Размер блока в алгоритме, основанном на SP-сети, составляет 128 бит, размер ключа 256 бит, и десять раундов в нем реализуются. При зашифровании в данном алгоритме

применяется следующая последовательность действий: вектор открытого текста складывается по модулю 2 с раундовым ключом. После чего в каждом из раундов данный шифруемый блок проходит через два слоя: слой S, реализующий замену, и слой P, реализующий линейное преобразование. После того как действие основных раундов завершено, в конце прохождения через слой P результат подается на выход данного алгоритма в качестве шифр-текста. Преобразование в слое замены реализуется на основе 8-битных подблоков по таблице. А преобразование в слое P — на основе линейного регистра сдвига с обратной связью над полем Галуа по модулю неприводимого многочлена восьмой степени.

Асимметричные системы шифрования

Асимметричное шифрование, или криптографическая система с открытым ключом, представляет собой криптографическую систему, использующую открытые (public key) и закрытые (private key) ключи для шифрования и расшифровки данных. Эти ключи образуют так называемую ключевую пару и представляют собой большие числа, которые связаны некоторой зависимостью, но отличаются друг от друга. Открытый ключ передается по незащищенным каналам связи и известен всем. С помощью открытого ключа осуществляется шифрование данных и проверка электронной подписи документов (ЭЦП). Для расшифровки данных используется закрытый ключ, который хранится в тайне.

Таким образом, главным преимуществом асимметричного шифрования по сравнению с симметричным шифрованием является возможность сторон связываться и обмениваться данными друг с другом без использования секретных каналов связи.

Надежность шифрования (криптографическая стойкость) зависит от длины ключа и экспоненциально возрастает при увеличении длины ключа в два раза. Асимметричное шифрование применяется в различных протоколах таких, как SSH, OpenPGP, S/MIME, и SSL/TLS, а также в различных системах, требующих установки безопасного соединения в незащищенной сети или проверки цифровой подписи.

Наиболее распространенным асимметричным алгоритмом шифрования является RSA, используемый в протоколе SSL/TLS и применяемый как для шифрования, так и для цифровой подписи. В основе алгоритма лежит вычислительная сложность задачи факторизации больших целых чисел. Длина ключа RSA обычно составляет 1024 или 2048 бит. Однако эксперты считают, что 1024-битные ключи RSA в скором времени могут быть взломаны, поэтому некоторые организации переходят на 2048-битные ключи.

Схемы электронной цифровой подписи

При передачи электронных документов или сообщений по незащищенным каналам возникает угроза их перехвата. Злоумышленник может заполучить

электронный документ, модифицировать его и отправить первоначальному адресату в целях извлечения выгоды. Чтобы избежать подобных ситуаций была разработана цифровая подпись. Правильно применённая цифровая подпись даёт получателю уверенность в том, что электронный документ достоверный и выслан указанным отправителем. Цифровая подпись для электронных документов является аналогом ручной – для бумажных носителей информации. Цифровая подпись также обеспечивает свойство неотказуемости. Это означает, что подписывающий документ не сможет впоследствии отказаться от своей подписи.

Свойства:

1) Подлинность, убеждающая, что именно это лицо подписало документ.

2) Уникальность, так как и ручная подпись - часть документа, которую нельзя переместить на другие документы. То есть у любого отдельного документа будет своя уникальная цифровая подпись.

3) Целостность подписываемого документа, т. е. невозможность изменения подписанного документа.

4) Неотказуемость от авторства или согласия с содержимым документа, от подписи в дальнейшем нельзя отказаться.

Протокол цифровой подписи с достоверным посредником

Пользователь A хочет подписать цифровое сообщение и отправить его пользователю B. Существует посредник, у которого есть ключ k_1 для секретной связи с A и ключ k_2 для секретной связи с B. Эти ключи установлены до начала протокола и могут быть использованы многократно. Шифрование происходит при помощи симметричного шифра E.

Протокол:

1. Отправитель A зашифровывает на ключе k_1 сообщение a для адресата B и отправляет криптограмму b посреднику.

2. Посредник расшифровывает криптограмму b, используя ключ k_1.

3. Посредник формирует блок информации I = [a, b, pa], где a - расшифрованное сообщение, b - копия криптограммы, pa - идентификатор абонента A. Зашифровывает блок информации I на ключе k_2 и отправляет криптограмму с пользователю B.

4. B расшифровывает блок информации I, используя ключ k_2. B получил сообщение a и сертификат посредника pa, удостоверяющий, что сообщение пришло именно от A.

Достоинства протокола:

1. Только посредник и отправитель A имеют секретный ключ k_1, поэтому только A может отправить посреднику сообщение, зашифрованное на ключе k_1.(защита ЦП от подделки)

2. Подпись не тиражируется и подписанный документ неизменяем.

3. От этой подписи нельзя отречься (неотказуемость).

Определение схемы цифровой подписи

Схема цифровой подписи - это совокупность вероятностных полиномиальных алгоритмов (**Gen; Sign; Vrfy**), удовлетворяющих следующему:

1) Алгоритм генерации ключа **Gen** принимает на вход секретный параметр 1^n и на выходе выдает (pk; sk; s0) - открытый ключ, секретный ключ и начальное состояние соответственно. Предположим, что $|pk| \geq n$ и что n может быть вычислено из pk.

2) Алгоритм Подписания **Sign** принимает на вход секретный ключ sk, величину s_{i-1} и сообщение m. На выходе получается подпись σ наряду с величиной s_i, и записано, как $(\sigma, s_i) \leftarrow$ Sign $_{sk,s_{i-1}}(m)$.

3) Детерминированный алгоритм проверки подписи **Vrfy** принимает в качестве входных данных открытый ключ pk, сообщение m и подпись σ. В качестве выходных данных выступает бит b, и записывается, как $b :=$Vrfy $_{pk}(m, \sigma)$

RSA

Первой и наиболее известной во всем мире конкретной системой ЭЦП стала система RSA, математическая схема которой была разработана в 1977 г. в Массачуссетском технологическом институте США.

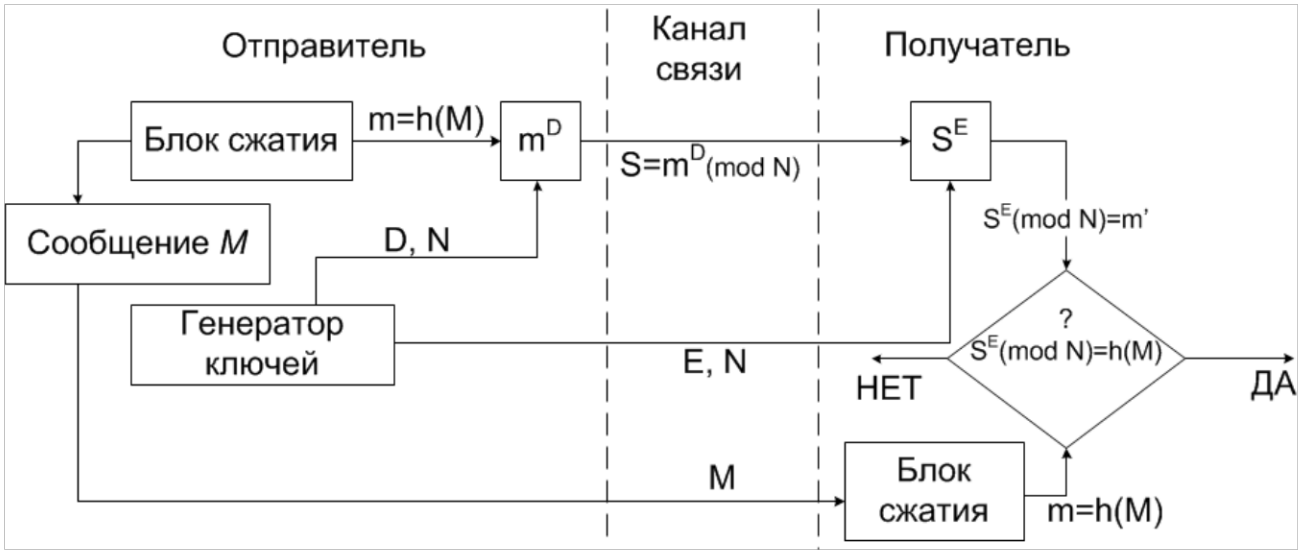

Рисунок 3 Обобщенная схема формирования и проверки цифровой подписи RSA

Сначала необходимо вычислить пару ключей (секретный ключ и открытый ключ). Для этого отправитель (автор) электронных документов вычисляет два больших простых числа p и q, затем находит их произведение

N = p * q

и значение функции f(N) = (p - 1)(q - 1).

Далее отправитель вычисляет число E из условий:

E <= f(N), НОД(E, f(N)) = 1

и число D из условий: D < N, E*D сравнимо с единицей по модулю f(N).

Пара чисел (E, N) является открытым ключом. Эту пару чисел автор передает партнерам по переписке для проверки его цифровых подписей. Число D сохраняется автором как секретный ключ для подписывания.

Допустим, что отправитель хочет подписать сообщение M перед его отправкой. Сначала сообщение M (блок информации, файл, таблица) сжимают с помощью хэш-функции (см. хэш-функция) h(M) в целое число m:

$$m = h(M).$$

Затем вычисляют цифровую подпись S под электронным документом M, используя хэш-значение m и секретный ключ D:

S = (m^D)(mod N). Пара (M,S) передается партнеру-получателю как электронный документ M, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа D.

После приема пары (M,S) получатель вычисляет хэш-значение сообидения M двумя разными способами. Прежде всего он восстанавливает хэш-значение m', применяя криптографическое преобразование подписи S с использованием открытого ключа E:

$$m' = (S^E)(\mathrm{mod}\ N).$$

Кроме того, он находит результат хэширования принятого сообщения M с помощью такой же хэш-функции h(M):

$$m = h(M).$$

Если соблюдается равенство вычисленных значений, т.е.

$$(S^E)(\mathrm{mod}\ N) = h(M),$$

то получатель признает пару (M,S) подлинной. Доказано, что только обладатель секретного ключа D может сформировать цифровую подпись S по документу M, а определить секретное число D по открытому числу E не легче, чем разложить

модуль N на множители. Кроме того, можно строго математически доказать, что результат проверки цифровой подписи S будет положительным только в том случае, если при вычислении S был использован секретный ключ D, соответствующий открытому ключу E. Поэтому открытый ключ E иногда называют "идентификатором" подписавшего.

Функция хэширования H m( ) или **хэш-функция** (hash-function) – это детерминированная функция, на вход которой подается строка битов произвольной длины, а выходом всегда является битовая строка фиксированной длины n .

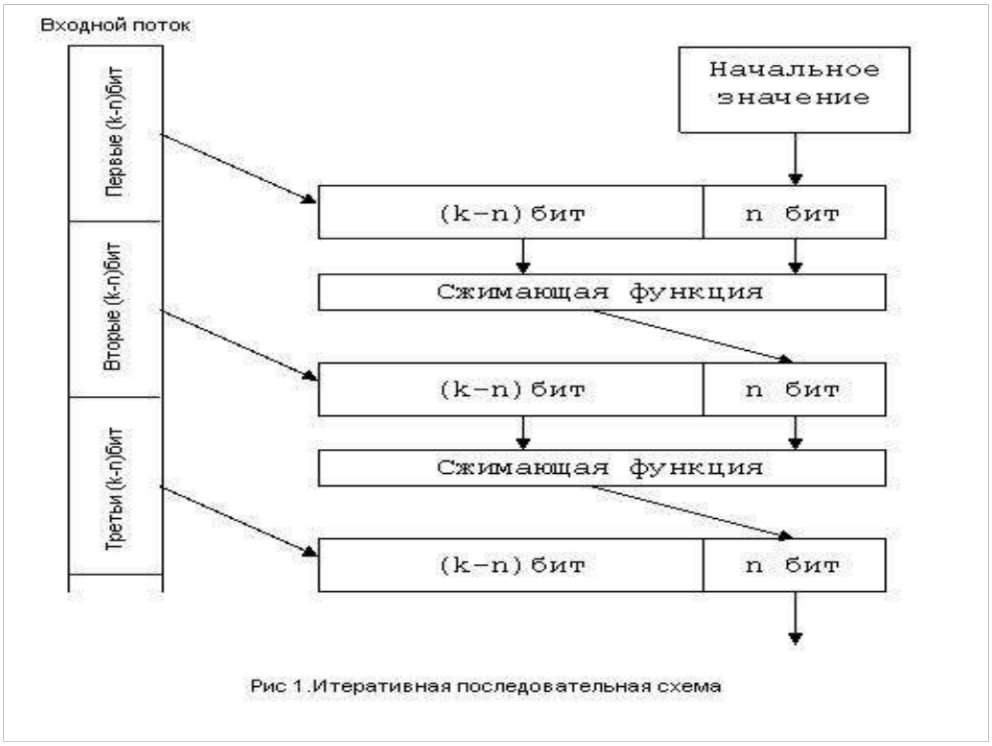

Рис 1.Итеративная последовательная схема

Общем случае в основе построения хеш-функции лежит итеративная последовательная схема. Ядром алгоритма является сжимающая функция — преобразование k входных в n выходных бит, где n — разрядность хеш-функции, а k — произвольное число, большее n. При этом сжимающая функция должна удовлетворять всем условиям криптостойкости.

Входной поток разбивается на блоки по (k − n) бит. Алгоритм использует вре́менную переменную размером в n бит, в качестве начального значения которой берётся некое общеизвестное число. Каждый следующий блок данных объединяется с выходным значением сжимающей функции на предыдущей итерации. Значением хеш-функции являются выходные n бит последней итерации. Каждый бит выходного значения хеш-функции зависит от всего

входного потока данных и начального значения. Таким образом достигается лавинный эффект.

При проектировании хеш-функций на основе итеративной схемы возникает проблема с размером входного потока данных. Размер входного потока данных должен быть кратен (k − n). Как правило, перед началом алгоритма данные расширяются неким, заранее известным, способом.

Рис.2. Обобщенная схема формирования хэш-функции

Помимо однопроходных алгоритмов, существуют многопроходные алгоритмы, в которых ещё больше усиливается лавинный эффект. В этом случае данные сначала повторяются, а потом расширяются до необходимых размеров.

Криптографическая хеш-функция является доказуемо защищённой от коллизий, если задача нахождения коллизий может быть средуцирована за полиномиальное время от задачи P, которая считается неразрешимой за полиномиальное время. Иначе говоря, если алгоритм A позволял бы за полиномиальное время решить задачу нахождения коллизий при существовании редуцирующего алгоритма R, работающего также за полиномиальное время, то последний позволил бы алгоритму A решить задачу P за полиномиальное время, что противоречит её сложности, а значит задача нахождения коллизий не легче задачи P.

Идеальной криптографической хеш-функцияей является такая криптографическая хеш-функция, к которой можно отнести пять основных свойств:

1. Детерминированность. При одинаковых входных данных результат выполнения хеш-функции будет одинаковым (одно и то же сообщение всегда приводит к одному и тому же хешу);
2. Высокая скорость вычисления значения хеш-функции для любого заданного сообщения;
3. Невозможность сгенерировать сообщение из его хеш-значения, за исключением попыток создания всех возможных сообщений;
4. Наличие лавинного эффекта. Небольшое изменение в сообщениях должно изменить хеш-значения, так широко, что новые хеш-значения не совпадают со старыми хеш-значениями;
5. Невозможность найти два разных сообщения с одинаковыми хеш-значениями.

Таким образом, идеальная криптографическая хеш-функция, у которой длина n (то есть на выходе n бит), для вычисления прообраза должна требовать как минимум 2^n операций.

Шифраторы фирмы Cylink имеют клавиатуру для ввода ключей и могут передавать сигналы тревоги и состояния прибора. Дистанционная система управления сетью CNMS (Cylink Network Management System) фирмы Cylink может контролировать до 256 шифраторов CIDECHSi из одного центрального пункта.

Несимметричные криптосистемы используются для формирования цифровой подписи и шифрования (формирования) симметричных ключей при их рассылке по каналам связи. Среди протоколов распределения ключей на практике используется метод Диффи-Хеллмана и метод цифрового конверта. Среди методов циф-ровой подписи наибольшее применение нашли RSA-подобные алгоритмы и алгоритмы на основе метода Эль-Гамаля, стандар-тизованные в ряде стран. Наиболее перспективным представляется использование усовершенствованного метода цифровой подписи Эль-Гамаля, который в последние годы стандартизован в США и России.

Список использованной литературы

1. Adam Shostack. “Threat Modeling: Designing for Security”. Published by John Wiley & Sons, Inc., Canada 2014.- 626 p.

2. Richard Bejtlich. “The Practice of Network Security Monitoring”. Published by No Starch Press, Inc., USA 2013. – 380 p.

3. Scott E. Donaldson, Stanley G. Siegel, Chris K. Williams and Abdul Aslam. “Enterprise Cybersecurity: how to build a successful Cyberdefense program against advanced threats”. Published by Apress, 2015. – 508 p.

4. Хорев А. А. Организация защиты конфиденциальной информации в коммерческой структуре // Защита информации. Инсайд : журнал. — 2015. — № 1. — С. 14—17. — ISSN 2413-3582

5. Фред Б. Риксон. Коды, шифры, сигналы и тайная передача информации. — Астрель, 2011. — ISBN 978-5-17-074391-9.

6. Morris J. Dworkin. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (англ.) // Federal Inf. Process. Stds. (NIST FIPS) - 202. — 2015-08-04.