Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра компьютерных технологий

СОВРЕМЕННЫЕ КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Конспект лекций
для магистрантов по специальности
6М070400 - Вычислительная техника и программное обеспечение

Алматы 2014

СОСТАВИТЕЛИ: Шайхин Б.М., Мусатаева Г.Т., Байжанова Д.О. Современные криптографические методы защиты информации. Конспекты лекций для магистрантов специальности 6М070400- Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2014 . – 35 с.

В конспектах лекций излагаются вопросы применения современных криптографических методов для обеспечения безопасности информации. Доказывается, что только на основе их использования удается успешно решать задачи защищенного взаимодействия удаленных абонентов, в том числе обеспечения секретности информации, аутентичности субъектов и объектов информационного взаимодействия.

Рекомендуются магистрантам по специальности 6М070400 – Вычислительная техника и программное обеспечение.

Таблица – 2, библиография – 7 названий.

Рецензент: канд. техн. наук, доцент Башкиров М.В.

Печатается по дополнительному плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2014 г.

Введение

Преподавание дисциплины «Современные криптографические методы защиты информации» магистрантам специальности 6М070400 – Вычислительная техника и программное обеспечение позволяет выработать адекватное представление о применениях современных криптографических методов для обеспечения безопасности информации.

Цель курса:

– дать основные сведения о современных криптографических методах защиты информации;

- ознакомить с основами криптологии и с криптосистемами с секретным ключом;

- дать основные сведения криптографического преобразования – хеширования;

- познакомить с методами аутентификации и криптосистемами с открытым ключом;

- дать основные сведения о криптографических протоколах и об управлении ключами;

- ознакомить с основами эллиптической криптографии;

Задачи курса:

- изучение современных методов обработки, преобразования и защиты информации в современных компьютерных системах;

- овладение обучающимися криптологией, методами и алгоритмами эксплуатации программных систем сбора, закрытия, восстановления и аутентификации информации;

- изучение современных способов борьбы с несанкционированным блокированием, доступом, копированием, изменением и сбором информации;

- научить и сформировать навыки практического применения знаний для защиты информации;

- изучение принципов построения генераторов псевдослучайных чисел;

- обучить эффективному применению математических и компьютерных моделей для решения задачи защищенного взаимодействия удаленных абонентов;

- обучить методам обеспечения секретности информации, аутентичности субъектов и объектов информационного взаимодействия;

- развить творческий потенциал будущих магистров по криптологии для дальнейшего самостоятельного освоения новых методов и технологий в условиях непрерывного развития и совершенствования современных криптографических методов защиты информации.

Компетентность магистранта:

должен знать:

- основы математики криптографии (модульную арифметику, сравнения и матрицы; алгебраические структуры) и математическое представление секретных систем;

- подходы для создания криптосистемы с открытым ключом и требования к качественной хэш-функции;

- особенности криптографических протоколов и управление ключами;

- практические способы построения систем защиты информации;

уметь:

- анализировать тексты и определять их избыточность;

- строить системы трансформации информационно-статистических характеристик текстов.

Достигнуть указанных целей, можно только изучив совместную работу программных и аппаратных средств, определив необходимую степень защиты.

1 лекция. Криптографические системы

Цель лекции: ознакомление с историей криптографии. Дать основные понятия и определения и требований к криптографическим системам.

Создание защищённой системы невозможно без применения криптографических методов. Они представляют в распоряжение разработчика средства, обеспечивающие определенные гарантии степени защиты. Одним словом, криптография является очень мощным инструментом, без которого построение большинства систем защиты просто невозможно.

С момента, как криптография оказалась рассекречена, появились открытые публикации. Информационное наполнение всех этих публикаций можно разделить на четыре основных категории.

К первой категории относятся история криптографии и описание древних методов шифрования.

Ко второй категории относятся высоконаучная литература по криптографии. Для большинства людей, не имеющих глубокого математического образования, понять то, что относится к научной криптографии, нереально: криптография нашего времени – это чистая математика. Эта литература нужна разработчикам криптографических алгоритмов и криптоаналитикам и бесполезна для тех, кто использует готовые алгоритмы и протоколы.

В третью категорию попадают описания общих понятий криптографии и спецификации различных криптографических алгоритмов и протоколов. Используя подобную информацию, любой программист сможет реализовать симметричное и асимметричное шифрование, генерацию и проверку цифровой подписи, а также решить и другие задачи, требующие применения криптографии.

К четвертой категории относятся рекомендации по правильному применению криптографии. Здесь можно порекомендовать: «Прикладную криптографию» Брюса Шнайера.

Современная криптология, корнями уходящая во многие фундаментальные математические науки, впитала в себя огромное количество теорем, фактов, криптографических схем, алгоритмов и пр.

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании.

Задачи криптографии. Криптографические отображения применяются для решения следующих основных задач защиты информации:

1) Обеспечения секретности (privacy [‘praivəsi], confidentiality [,konfi’denʃəliti], secrecy [‘si:krisi]) информации, то есть защиты от несанкционированного ознакомления с содержанием.

2) Обеспечения целостности (data integrity [in’tegriti]) информации, то есть защиты от несанкционированного изменения, к которому относится вставка, удаление и замена фрагментов исходного сообщения.

3) Обеспечения аутентификации (data origin, authentication) информации, то есть подтверждения подлинности сторон, самой информации, времени создания информации и т.д.

4) Обеспечения неотказуемости (невозможности отказа) (non-repudiation [ri,pju:di’eiʃən]) от авторства.

Криптографические отображения информации стали использоваться в электронном документообороте для защиты электронных платежей, коммерческих сделок и др.

Основные понятия и определения. Шифром называется семейство E обратимых криптографических отображений информации. С каждым отображением шифра связано значение k некоторого параметра, называемого ключом, то есть

$$E = \{E_k\}, \quad k \in K,$$

где K – конечное множество допустимых значений ключа, называемое ключевым множеством. Выбранный ключ k однозначно определяет отображение E_k из семейства шифра E.

Ключ имеет так называемый жизненного цикла, то есть выполнение таких действий с ключом, как генерация, распределение (рассылка), хранение, применение при реализации отображений шифра, смена, уничтожение.

Информацию, подвергающуюся отображениям шифра, называют открытым текстом. Применение отображения шифра к открытому тексту называется шифрованием. Результат шифрования открытого текста называется криптограммой.

Использование шифра для решения той или иной криптографической задачи подразумевает наличие соответствующих действующих лиц (абонентов, использующих секретную связь) и определённого порядка их взаимодействия, называемого криптографическим протоколом.

Семейство отображений шифра в совокупности с используемыми протоколами образуют криптосистему. Ключевое множество и ключевые протоколы (т.е. протоколы, управляющие жизненным циклом ключей) образуют ключевую систему шифра.

Раскрытие криптоаналитиком информации, защищаемой шифром, называют дешифрованием (ключ для расшифрования неизвестен, т.е. неизвестно, какое именно отображение следует использовать для расшифрования). Способ раскрытия шифра или информации, защищаемой шифром, называют криптоаналитической атакой.

Способность шифрсистемы противостоять атакам криптоаналитика получила название криптографической стойкости шифрсистемы.

Секретная связь с использованием симметричной криптосистемы.

Рассмотрим организацию секретной связи с использованием симметричной криптосистемы. Действующими лицами протокола являются отправитель, адресат, пассивный противник, активный противник и др. Задача протокола – передать секретное сообщение X от отправителя адресату. Последовательность действий выглядит следующим образом:

1) отправитель и адресат договариваются об используемой симметричной криптосистеме, то есть о семействе отображений

$$E = \{E_k\}, \quad k \in K;$$

2) отправитель и адресат договариваются о секретном ключе связи k , то есть об используемом отображении $E_k \in E$;

3) отправитель шифрует открытый текст X с помощью отображения Ek, то есть создает криптограмму $Y = E_k(X)$;

4) криптограмма Y передается по линии связи адресату;

5) адресат расшифровывает криптограмму Y, используя тот же ключ k и отображение Ek-1, обратное к отображению Ek, и читает сообщение X:

$$X = E_k\text{-}1(Y).$$

Разрабатывая криптосистему, криптограф исходит из следующих предположений о возможностях криптоаналитика:

1) криптоаналитик контролирует линию связи;

2) криптоаналитику известно устройство семейства E отображений шифра;

3) криптоаналитику неизвестен ключ k, то есть неизвестно отображение E_k, использованное для получения криптограмм Y.

В этих условиях криптоаналитик должен решить следующие задачи дешифрования:

1) определить открытый текст X и использованный ключ k по перехваченной криптограмме Y, то есть построить алгоритм дешифрования Ψ такой, что

$$\Psi(Y) = (X,k).$$

Данная постановка задачи предполагает использование криптоаналитиком статистических свойств открытого текста;

2) определить использованный ключ k по известным открытому и шифрованному текстам, то есть построить алгоритм дешифрования φ такой, что

$$\varphi(X, Y) = k.$$

Такая постановка задачи имеет смысл, когда криптоаналитик перехватил несколько криптограмм, полученных с использованием ключа k, и располагает открытыми текстами не для всех перехваченных криптограмм. В этом случае, решив задачу дешифрования второго типа, он «прочтёт» все открытые тексты, зашифрованные с использованием ключа k;

3) определить используемый ключ k по специально подобранному открытому тексту X и по соответствующему шифрованному тексту Y, то есть построить алгоритм дешифрования φ_x, такой, что

$$\varphi_x(Y) = k.$$

Такая задача возникает тогда, когда криптоаналитик может тестировать криптосистему, т. е. генерировать криптограммы для специально подобранного открытого текста. Чаще такая возможность возникает при анализе асимметричных систем.

Таким образом, по организации секретной связи с использованием симметричной криптосистемы можно сделать следующие выводы:

1) протокол должен защищать открытый текст и ключ от НСД постороннего лица на всех этапах передачи информации от источника к получателю сообщений. Секретность ключа более важна, чем секретность нескольких сообщений, шифруемых на этом ключе. Если ключ скомпрометирован, тогда противник, имеющий ключ, может расшифровать все сообщения, зашифрованные на этом ключе. Кроме того, противник сможет имитировать одну из переговаривающихся сторон и генерировать фальшивые сообщения. При частой смене ключей эта проблема сводится к минимуму;

2) протокол не должен допускать выхода в линию связи «лишней» информации, представляющей криптоаналитику противника дополнительные возможности дешифрования криптограмм. Протокол должен защищать информацию не только от посторонних лиц, но и от взаимного обмана действующих лиц протокола;

3) если допустить, что каждая пара пользователей сети связи применяет отдельный ключ, то число необходимых ключей равно n(n - 1)/2 для n пользователей. Это означает, что при большом n генерация, хранение и распределение ключей становится трудоёмкой проблемой.

2 лекция. Симметричные криптосистемы

Цель лекции: ознакомление с основными классами симметричных криптосистем. Дать общие сведения о блочных шифрах, о генерировании блочных шифров, об алгоритмах блочного шифрования: DES; AES; RC6;ГОСТ 28147-89; SAFER+, SAFER++. Режимы применения блочных шифров.

Симметричные криптосистемы (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.

Схемы и принципы симметричного шифрования.

Математические задачи криптографии впервые были рассмотрены К. Шенноном [1]. Он с помощью предложенного им теоретико-информационного подхода решил некоторые проблемы теоретико-информационной криптографии. В частности, им показано, что абсолютной надежностью могут обладать только те шифры, у которых объем ключа не меньше объема шифруемой информации. Там же [1] были предложены и принципы построения криптографически надежных преобразований с помощью композиции некоторых разнородных отображений. Шенноном [1] были сформулированы и доказаны математическими средствами необходимые и достаточные условия не дешифруемой системы шифра. Им установлено, что единственным не дешифруемым шифром является так называемая, лента одноразового использования (One-time Pad), когда открытый текст шифруется с помощью случайного ключа такой же длины. Это обстоятельство делает шифр абсолютно стойким.

Алгоритмы симметричного шифрования различаются способом, которым обрабатывается исходный текст. Возможно шифрование блоками или шифрование потоком. Блок текста рассматривается как неотрицательное целое число, либо как несколько независимых неотрицательных целых чисел. Длина блока всегда выбирается равной степени двойки. В большинстве блочных алгоритмов симметричного шифрования используются следующие типы операций:

1) табличная подстановка, при которой группа бит отображается в другую группу бит. Это так называемые S-box;

2) перемещение, с помощью которого биты сообщения переупорядочиваются;

3) операция сложения по модулю 2, обозначаемая XOR – ИЛИ;

4) операция сложения по модулю 232 или по модулю 216;

5) циклический сдвиг на некоторое число бит.

Эти операции циклически повторяются в алгоритме, образуя так называемые раунды. Входом каждого раунда является выход предыдущего раунда и ключ, полученный по определенному алгоритму из ключа шифрования K. Ключ раунда называется подключом.

Область применения. Стандартный алгоритм шифрования должен быть применим во многих приложениях:

Шифрование данных. Алгоритм должен быть эффективен при шифровании файлов данных или большого потока данных.

Создание случайных чисел. Алгоритм должен быть эффективен при создании определенного количества случайных бит.

Хэширование. Алгоритм должен эффективно преобразовываться в одностороннюю хэш-функцию.

Недостатки симметричного шифрования. Основной недостаток симметричного шифрования заключается в необходимости публичной передачи ключей – «из рук в руки». При такой системе становится практически невозможным использование симметричного шифрования с

неограниченным количеством участников. В остальном же алгоритм симметричного шифрования можно считать достаточно проработанным и эффективным, с минимальным количеством недостатков.

Заключение. Алгоритмы с симметричными ключами имеют очень высокую производительность. Криптография с симметричными ключами очень стойкая, что делает практически невозможным процесс дешифрования без знания ключа. При прочих равных условиях стойкость определяется длиной ключа. Так как для шифрования и дешифрования используется один и тот же ключ, при использовании таких алгоритмов требуются очень надежные механизмы для распределения ключей. Алгоритмы симметричного шифрования используют ключи не очень большой длины и могут быстро шифровать большие объемы данных.

Принципы шифрования, используемыми в алгоритме симметричного шифрования AES; RIJNDAEL.

Advanced Encryption Standard (AES), также известный, как Rijndael (произносится [rɛinda:l] (Рейндол)), — симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES. По состоянию на 2009 год AES является одним из самых распространённых алгоритмов симметричного шифрования [1, 2].

Алгоритм Rijndael.

Алгоритм Rijndael (читается «Рейндал») разработан бельгийскими специалистами Joan Daemen (Proton World International) и Vincent Rijmen (Katholieke Universiteit Leuven). Шифр Rijndael/AES (то есть рекомендуемый стандартом) характеризуется размером блока 128 бит, длиной ключа 128, 192 или 256 бит и количеством раундов 10, 12 или 14 в зависимости от длины ключа. В принципе структуру Rijndael можно приспособить к любым размерам блока и ключа, кратным 32, а также изменить число раундов.

Блок данных, обрабатываемый с использованием Rijndael, делится на массивы байтов, и каждая операция шифрования является байт-ориентированной. Каждый раунд состоит из трех различных обратимых преобразований, называемых слоями. Шифр начинается и заканчивается сложением с ключом. Это позволяет закрыть вход первого раунда при атаке по известному тексту и сделать криптографически значимым результат последнего раунда. В алгоритме широко используются табличные вычисления, причем все необходимые таблицы задаются константно, т.е. не зависят ни от ключа, ни от данных. Необходимо отметить, что в отличие от шифров, построенных по сети Фейштеля, в Rijndael функции шифрования и расшифрования различны.

DES (Data Encryption Standard) – Алгоритм шифрования, действовавший в качестве государственного стандарта в области шифрования данных в США с 1977 по 2001 годы. Основные параметры DES: размер блока 64 бита, длина ключа 56 бит, количество раундов – 16. DES является классической сетью Фейстеля с двумя ветвями. Алгоритм преобразует за несколько раундов 64-битный входной блок данных в 64-

битный выходной блок. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP-1) преобразований.

Алгоритм шифрования RC6.

Алгоритм RC6 был разработан в 1998 г. рядом специалистов научного подразделения фирмы RSA. Фактически, алгоритм претерпел два принципиальных изменения:

а) в отличие от RC5, в алгоритме используется умножение по модулю 232;

б) для сохранения 32-битных вычислений вместо разбиения шифруемого блока данных (128 бит согласно принципиальному требованию конкурса AES) на два 64-битных субблока выполняется его разбиение на 4 32-битных субблока и их обработка по несколько измененной схеме. RC6 имеет гибкую структуру. Помимо секретного ключа, параметрами алгоритма являются следующие:

1) размер слова w;

2) RC6 шифрует блоками по 4 слова;

3) количество раундов алгоритма R;

4) размер секретного ключа в байтахb.

Стандарт шифрования данных ГОСТ 28147-89[1].

Алгоритм реализует шифрование 64-битовых блоков данных с помощью 256-битового ключа. ГОСТ рекомендован к использованию для защиты любых данных, представленных в виде двоичного кода. Данный стандарт формировался с учетом мирового опыта. Были приняты во внимание недостатки и нереализованные возможности алгоритма DES. Алгоритм сложен. ГОСТ предусматривает 3 основных режима шифрования: простую замену, гаммирование и гаммирование с обратной связью, а также режим выработки имитовставки, используемый для аутентификаци. Процесс выработки имитовставки в ГОСТе единообразен для всех режимов шифрования.

Комбинирование алгоритмов блочного шифрования.

Недостаточная стойкость алгоритма шифрования DES привела к идее многократного шифрования. Алгоритм блочного шифрования использовался несколько раз с разными ключами для шифрования одного и того же блока открытого текста.

Двукратное шифрование блока открытого текста с помощью двух разных ключей. В этом случае сначала шифруют блок M ключом k1, а затем получившийся шифртекст $E_{k_1}(M)$ шифруют ключом k2. В результате получают криптограмму

$$C = E_{k_2}(E_{k_1}(M)).$$

При трёхкратном шифровании можно применить три различных ключа. Уравнение шифрования в этом случае принимает вид: C =

[1] ГОСТ 28147-89 закрыт грифом ДСП поэтому дальнейшее изложение сделано по изданию Спесивцев А.В. и др. «Защита информации в персональных ЭВМ», М., Радио и связь, 1992.

$E_{k_3}(D_{k_2}(E_{k_1}(M)))$. При этом возрастает общая длина результирующего ключа и соответственно возрастает стойкость шифра. Возрастание многократного шифрования не может быть безграничным, оно происходит до тех пор, пока суммарное число ключей (256· s) не превзойдёт общее число преобразований, реализуемых схемой, то есть общее число простых замен, «из которых состоит» данный шифр замены.

3 лекция. Потоковые шифры

Цель лекции: дать общие сведения о потоковых шифрах, о самосинхронизирующихся шифрах. Синхронные шифры. Примеры потоковых шифров. Алгоритмы: RC4; SEAL; WAKE.

Поточный шифр (ПШ) — это симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста. ПШ представляет собой автономный автомат, который вырабатывает псевдослучайную двоичную последовательность. Поточные шифры представляют собой разновидность гаммирования и преобразуют открытый текст в шифрованный последовательно по 1 биту.

Генератор ключевой последовательность называется генератором бегущего ключа. Он выдаёт последовательность бит k1, k2 , …, ki, …. Эта ключевая последовательность складывается по модулю 2 с последовательностью бит исходного текста p1, p2, …, pi , … для получения шифрованного текста:

$$ci = pi \oplus ki .$$

На приёмной стороне шифрованный текст складывается по модулю 2 с идентичной ключевой последовательностью для получения исходного текста:

$$ci \oplus ki = pi \oplus ki \oplus ki = pi.$$

Стойкость системы зависит от внутренней структуры генератора ключевой последовательности. Если генератор выдаёт последовательность с небольшим периодом, то стойкость системы будет невелика. Напротив, если генератор будет выдавать бесконечную последовательность истинно случайных (не псевдослучайных!) бит, то мы получим одноразовый блокнот с идеальной стойкостью.

Реальная стойкость ПШ лежит где-то посредине между стойкостью простой моноалфавитной подстановки и одноразового блокнота. Генератор ключевой последовательности выдаёт поток битов. Этот поток выглядит случайным, но в действительности он является детерминированным и может быть в точности воспроизведён на приёмной стороне. Чем больше генерируемый поток похож на случайный, тем больше усилий потребуется от криптоаналитика для взлома шифра.

Если каждый раз при включении генератор будет выдавать одну и ту же последовательность, то взлом криптосистемы будет тривиальной задачей. Перехватив два шифрованных текста, хакер может сложить их по модулю 2 и получить два исходных текста, сложенных также по модулю 2. Такую систему раскрыть очень просто. Если же в руках противника окажется пара: исходный текст - шифрованный текст, задача вообще становится тривиальной. По этой причине все ПШ используют ключ. Выход генератора ключевой последовательности зависит от этого ключа. В этом случае простой криптоанализ невозможен. ПШ используется для шифрования непрерывных потоков данных (в сетях передачи данных).

Структура генератора ключевой последовательности - это конечный автомат с памятью, состоящий из трёх блоков:

1) блока памяти, хранящего информацию о состоянии генератора;

2) выходной функции, генерирующей бит ключевой последовательности в зависимости от состояния;

3) функции переходов, задающей новое состояние, в которое перейдёт генератор на следующем шаге.

Синхронизация поточных шифрсистем.

При использовании ПШ простой замены потеря (или искажение) отдельных знаков шифрованного текста при передаче по каналу связи приводит лишь к локальным потерям: все знаки шифртекста, принятые без искажений, будут расшифрованы правильно. Это объясняется тем, что алгоритм шифрования не зависит ни от расположения знаков в тексте, ни от их конкретного вида.

Многоалфавитные ПШ также не распространяют ошибок при искажении отдельных знаков шифрованного текста, но оказываются неустойчивыми к пропускам знаков шифрованного текста. Пропуск знаков приводит к неправильному расшифрованию всего текста, следующего за пропущенным знаком. Поскольку во всех каналах передачи данных имеются помехи, в системах криптографической защиты, использующих ПШ, заботятся о согласованном порядке применения преобразований при зашифровании и расшифровании. Другими словами, решают проблему синхронизации процедур зашифрования и расшифрования.

По способу решения этой проблемы поточные шифры (ПШ) делят на синхронные (СПШ) и асинхронные (или самосинхронизирующиеся – ССПШ).

Криптосхема СПШ состоит из управляющего и шифрующего блоков. Управляющий блок генерирует управляющую последовательность {γt}, которая используется для формирования шифрующих отображений φ_{γ_t}, t = 1, 2, … Управляющая последовательность называется управляющей гаммой, а управляющий блок – генератором гаммы (члены управляющей последовательности обозначались греческой буквой γ). Шифрующий блок зашифровывает символ открытого текста xt в символ шифрованного текста yt с использованием отображения φ_{γ_t}, t = 1, 2, …

Управляющий блок ПШ моделируется криптографическим генератором (к.г.) - G = (S,Г,К,z,g1,h1,f1), шифрующий блок – автоматом Мили с постоянной памятью A = (X,Г,Y, f2) (множество Г возможных состояний памяти автомата А совпадает с выходным алфавитом к.г.(криптографического генератора) АГ), а СПШ в целом моделируется шифрующим автоматом АСПШ = (X, S, Y, K, z, g, h, f). Шифрующий автомат является последовательным соединением 2-го типа криптографического генератора (к.г.) - G и автомата Мили А:

$$АСПШ = G \Rightarrow A.$$

В t-м такте, t = 1, 2, …, к.г. G генерирует знак гаммы γ_t,

$$\gamma_t = f_1 (s_t, k_t),$$

который записывается в память автомата А. Под воздействием знака γ_t а.п.п. А зашифровывает знак x_t открытого текста в знак yt шифрованного текста с помощью отображения φ_{γ_t}:

$$\varphi_{\gamma_t} (xt) = f2 (yt,xt) = y_t.$$

Множество $\Phi = \{\varphi_\gamma : \gamma \in Г\}$ шифрующих отображений ПШ есть множество подфункций функции f2 (y_t,x_t), соответствующих всевозможным фиксациям переменной γ.

По способу построения моноключевые генераторы гаммы, используемые в СПШ, делятся на два вида:

1) первый вид генераторов образуют генераторы с внутренней ОС (обратной связью) (internal feedback), у которых функция выходов f_1 не зависит от ключа k. Имеется разновидность этого класса генераторов в которых от ключа зависит только начальное состояние;

2) второй вид – генераторы типа счётчика (counter mode), они отличаются тем, что от ключа зависит только функция выходов f_1.

Генераторы второго класса в отличие от генераторов первого класса позволяют вычислить i-тый бит гаммы, не вычисляя всех предыдущих битов. Для этого генератор устанавливается в i-тое внутреннее состояние, после чего вычисляется соответствующий ему i-тый бит гаммы. Это свойство полезно для обеспечения случайного доступа к файлам данных; оно позволяет расшифровать отдельный фрагмент данных, не расшифровывая файл полностью.

Безопасность системы полностью зависит от свойств генератора потока ключей. Если генератор потока ключей выдает бесконечную строку нулей, шифротекст будет совпадать с открытым текстом, и все операции будут бессмысленны. Если генератор потока ключей выплевывает повторяющийся 16-битовый шаблон, алгоритм будет являться простым XOR с пренебрежимо малой безопасностью. Если генератор потока ключей выплевывает бесконечный поток случайных битов, вы получаете одноразовый блокнот и идеальную безопасность.

Генератор потока ключей создает битовый поток, который похож на случайный, но в действительности детерминирован и может быть безошибочно воспроизведен при дешифрировании. Чем ближе выход

генератора потока ключей к случайному, тем больше времени потребуется криптоаналитику, чтобы взломать шифр.

Генератор потока ключей состоит из трех основных частей. Внутреннее состояние описывает текущее состояние генератора потока ключей. Два генератора потока ключей, с одинаковым ключом и одинаковым внутренним состоянием, выдают одинаковые потоки ключей. Функция выхода по внутреннему состоянию генерирует бит потока ключей. Функция следующего состояния по внутреннему состоянию генерирует новое внутреннее состояние.

В самосинхронизирующихся потоковых шифрах (ССПШ) каждый бит потока ключей является функцией фиксированного числа предыдущих битов шифротекста.

Так как внутреннее состояние полностью зависит от предыдущих п шифротекста, дешифрирующий генератор потока ключей автоматически синхронизируется с шифрующим генератором потока ключей, приняв п битов шифротекста.

В интеллектуальных реализациях этого режима каждое сообщение начинается случайным заголовком длиной п битов. Этот заголовок шифруется, передается и затем расшифровывается. Расшифровка будет неправильной, но после этих п битов оба генератора потока ключей будут синхронизированы.

Алгоритм RC4 - потоковый шифр с переменной длиной ключа (разработан в 1987 г. Роном Ривестом для компании RSA Data Security, Inc).

Работает в режиме внешней обратной связи OFB (output feedback). Ключевая последовательность не зависит от исходного текста. Структура алгоритма включает блок замены размерностью 8×8: S0, ..., S255. Блок замены представляет собой зависимую от ключа переменной длины перестановку чисел 0, ..., 255. Имеется два счетчика i и j, первоначально равные 0. Для генерирования псевдослучайного байта выполняются следующие действия:

$$i = (i + 1) \bmod 256;$$

$$j = (j + S) \bmod 256.$$

Переставляются Si и S,: $t = (S_i + S_j) \bmod 256$, $k = S_¡$. Байт k складывается по модулю 2 с байтом исходного текста для получения шифрованного. Шифрование по этому алгоритму быстрее в ~10 раз, чем шифрование DES при программной реализации.

Алгоритм SEAL (Software Encryption Algorithm) приспособлен для программной реализации ПШ (потоковый шифр), разработан компанией IBM и оптимизирован для 32-разрядных процессоров. Для эффективной работы ему требуются восемь 32-разрядных регистров и кэш объемом несколько килобайт. SEAL не является ПШ в традиционном смысле. Он представляет собой семейство псевдослучайных функций. Ключ k (160-битовый) и 32-битовое значение шифра п (индекс) преобразует в L-битовую строку k(n). L может принимать любое значение, меньшее 64 килобайт. Такой шифр обозначают SEAL (k, n, L). Предполагается, что если k выбирается случайно, то k(n) будет вычислительно неотличима от случайной L-битовой функции

от п. SEAL и предусматривает использование трех зависящих от ключа таблиц: R, S и T (они заполняются предварительно при помощи алгоритма безопасного хэширования SHA (Secure Hash Algorithm), и зависят только от ключа). Идея алгоритма следующая:

1) использование большого секретного, зависящего от ключа блока замены T;

2) перемежение операций, не коммутирующих друг с другом (сложение и исключающее ИЛИ- XOR);

3) использование внутреннего состояния, явно не проявляющегося в потоке данных (значения ni, используемые в конце итерации для модификации регистров A и C);

4) изменение шаговой функции в зависимости от номера шага и изменение итерационной функции в зависимости от номера итерации;

5) использование известных и отработанных алгоритмов для заполнения таблиц.

Алгоритм WAKE (Word Auto Key Encryption - шифрование слов с автоключом). На выходе получается последовательность 32-битовых слов, которые могут служить в качестве гаммы шифра. Работает в режиме кодированной обратной связи CFB (cipher feedback). В режиме CFB - предыдущее слово шифрованного текста используется для генерации следующего слова ключевой последовательности. В алгоритме используется специальный блок замены S из 256 32битных слов. Старшие байты этих слов являются перестановкой чисел от 0 до 255, а остальные три младших байта выбираются случайными. Вначале инициализируется блок замены на основе ключа. Затем инициализируются четыре регистра A, B, C и D начальными значениями, также зависящими от ключа (возможно другого): a0, b0, c0, d0. Очередное слово ключевой последовательности получается по формуле. Данный шифр является достаточно быстрым, но нестойким к атакам по выбранному исходному тексту.

4 лекция. Асимметричные криптосистемы

Цель лекции: рассматриваются общие положения: односторонние функции и функции-ловушки. Асимметричные системы шифрования. Криптосистема Эль-Гамаля. Криптосистема, основанная на проблеме Диффи-Хеллмана. Криптосистема RSA. Криптосистемы Меркля-Хеллмана и Хора-Ривеста. Криптосистемы, основанные на эллиптических кривых.

Системы шифрования с открытыми ключами (асимметричные системы). Чтобы отправитель и получатель узнали свой ключ, его нужно им доставить, причём так, чтобы сохранить его в тайне от всех других. Можно вырабатывать ключ для каждого сообщения, однако, детерменированные алгоритмы для этого не годятся: противник может прогнозировать ключи. Можно ключи вырабатывать каждый раз случайно, но тогда следует оповещать об этом своего корреспондента. Решение этой проблемы – в

односторонних (однонаправленных) функциях. Функция y = f (x) одностороняя, если вычисление y по x имеет малую трудоёмкость, а вычисление x по y – высокую. Криптосистемы открытого ключа основаны на задаче, которую трудно решить. Примеры таких задач – поиск сомножителей (факторинг), доказательство теорем и проблема коммивояжера.

Существуют два главных класс а задач, интересных для криптографии: полиномиальные (Polynomial, «P») – если задача решается за количество времени (или действий), которое можно выразить полиномом, и неполиномиальные (NonPolynomial, «NP») – если за время или за количество действий, описываемое полиномом, можно лишь оценить правильность предложенного решения.

Например, задача умножения двух чисел является полиномиальной, поскольку количество двоичных действий, необходимых для умножения двух чисел длины k (в двоичном виде), составляет максимум k2, то есть может быть выражено полиномом. В отличие от этого задача поиска фактора (сомножителя) некоторого числа является неполиномиальной, потому что предложенное решение может быть оценено в течение периода времени, описываемого полиномом.

Неизвестно, является ли задача полиномиальной, поскольку эффективных способов разложения на множители пока не найдено. Вопрос соотношения множеств полиномиальных и неполиномиальных задач (т.е. равно ли P = NP) является одной из наиболее важных нерешённых задач математики и информатики.

Математическое понятие трудоёмкость алгоритма.

Трудоёмкость данного алгоритма есть O (n2) (читается «о большое от n квадрат »), поскольку наибольшая степень п о л и н о м а - в т о р а я и, следовательно, при увеличении n в два раза время работы алгоритма увеличится в 4 раза.

К односторонним функциям относятся:

1) возведение в степень, (обратная функция – логарифмирование) в конечном (чаще простом) п о л е либо в другой алгебраической структуре (например, в группе точек эллиптической кривой);

2) разложение чисел на множители (прямая задача - умножение);

3) задачи кодирования-декодирования линейных кодов (типа кодов Рида – Малера или Рида - Соломона).

Криптосистемы открытого ключа основаны на односторонних функциях, имеющих «лазейку», которой является частный ключ. При известной «лазейке» легко вычисляются функции в обоих направлениях, но, не зная «лазейки», можно рассчитать функцию только в прямом направлении. Прямое направление используется для проверки подписи и для шифрования информации, обратное же – для создания подписи и расшифрования. В асимметричных криптосистемах размер ключа соответствует размеру входа односторонней функции. Чем больше ключ, тем больше различие между сложностью расчёта функции в прямом и обратном направлениях. Чтобы цифровая подпись была защищена в течение,

например, нескольких лет, необходимо использовать одностороннюю функцию, имеющую «лазейку» с достаточно большим входным значением. Злоумышленник, не знающий «лазейки», затратит годы на вычисление функции в обратном направлении.

Асимметричные системы шифрования.

Важная часть такой системы – «защищенный канал», по которому секретный ключ Z = [Z1, Z2,…,ZK], порожденный в источнике ключа и защищенный от криптоаналитика, передается предполагаемому получателю. Если используется один и тот же ключа в шифраторе источника и дешифраторе получателя сообщений, криптосистемы с секретными ключами называются одноключевыми, или симметричными, системами.

K знаков ключа – это символы некоторого конечного алфавита (часто используется двоичный алфавит {0, 1}). Источник сообщений порождает открытый текст X = [X1, X2,…, XM], а рандомизатор порождает рандомизирующую последовательность R = [R1, R2, …, RJ]. Шифратор образует криптограмму Y = [Y1, Y2, …, YN], как функцию X, R и Z. Это преобразование записывается в виде

$$Y = EZR(X).$$

Для того чтобы подчеркнуть, что криптограмма Y является функцией одного лишь открытого текста X, конкретный вид которой определяется секретным ключом Z и рандомизирующей последовательности R. Дешифратор способен также выполнить обратное преобразование без знания рандомизирующей последовательности. Это преобразование записывается в виде

$$X = DZ(Y)$$

и выражает тот факт, что открытый текст X является функцией криптограммы Y, конкретный вид которой определяется одним лишь секретным ключом Z.

Криптоаналитик противника видит только криптограмму Y и образует оценку $\hat{X}$ открытого текста X и (или) оценку $\hat{Z}$ секретного ключа Z. В соответствии с правилом Керкхоффа криптоаналитику известны все детали процессов зашифрования и расшифрования, кроме X, R, и особенно Z.

Алгоритм D-H*) работает следующим образом.

Предположим, что двум абонентам (P1 и P2) требуется установить между собой безопасное соединение, для которого необходимо согласовать ключ шифрования.

1) Абоненты P1 и P2 принимают к использованию два больших целых числа a и b, причем $1 < a < b$.

2) Абонент P1 выбирает случайное число i и вычисляет $I = a^i \bmod b$.

3) Абонент P1 передает I абоненту P2.

4) Абонент P2 выбирает случайное число j и вычисляет $J = a^j \bmod b$.

5) Абонент P2 передает J абоненту P1.

6) Абонент P1 вычисляет $k1 = J^i \bmod b$.

7) Абонент P2 вычисляет $k2 = I^j \bmod b$.

Имеем $k_1 = k_2 = ai*j \bmod b$, следовательно, k_1 и k_2 являются секретными ключами, предназначенными для использования при передаче других данных.

Уровень безопасности системы зависит от сложности нахождения i при известном $I = a_i \bmod b$. Эта задача дискретного логарифмирования и с помощью вычислительного оборудования ее решить невозможно, если числа очень велики.

Алгоритм RSA состоит из следующих пунктов:

1) выбрать простые числа p и q;

2) вычислить $n = p \cdot q$;

3) вычислить $m = (p - 1) \cdot (q - 1)$;

4) выбрать число d взаимно простое с m.

8) выбрать число e так, чтобы $e \cdot d = 1 \pmod m$.

6) числа e и d являются ключами. Шифруемые данные необходимо разбить на блоки - числа от 0 до n - 1.

Шифрование и дешифровка данных производятся следующим образом. Шифрование: $b = ae \pmod n$; Дешифровка: $a = bd \pmod n$. Ключи e и d равноправны, т.е. сообщение можно шифровать как ключом e, так и ключом d, при этом расшифровка должна быть произведена с помощью другого ключа. На практике при использование RSA длина p и q составляет 100 и более десятичных знаков (обеспечивается криптостойкость шифротекста).

Схема Эль-Гамаля (Elgamal) — криптосистема с открытым ключом, основана на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи (ЦП).

Основная идея ElGamal состоит в том, что не существует эффективных методов решения сравнения $ax == b \pmod p$.

Обозначения: Z(n) - вычеты по модулю n; Z(n) - мультипликативная группа обратимых элементов в Z(n); ab (mod n) - возведение a в степень b в кольце Z(n). Если p - простое число, то группа Z(p) изоморфна Z(p-1).

Эллиптическая криптография кривой. Криптография с открытым ключом применяется как для шифрования сообщений, так и для аутентификации (т. н. цифровая подпись (ЦП)). Каждая из этих систем полагается на трудную математическую проблему для ее защиты. Ведущие математики и компьютерщики не сумели создать эффективные алгоритмы для их решения.

Эллиптические кривые - математические конструкции, которые изучались математиками с XVII столетия. В 1985 Нейл Коблиц и Виктор Миллер независимо предложили криптосистемы с ключом общего пользования, использующие группу точек на эллиптической кривой. Так была рождена эллиптическая криптография кривой (код с исправлением ошибок). Разработчики исследовали силу кода с исправлением ошибок и улучшили методы для его выполнения. Код с исправлением ошибок дает самую высокую силу в криптосистеме с ключом общего пользования из-за трудности жесткой проблемы. Эта т.н. жесткая проблема эллиптической

кривой, дискретной проблемы логарифма (ECDLP). Она означает, что меньший размер ключа выдает эквивалентные уровни защиты.

5 лекция. Электронные цифровые подписи (ЭЦП)

Цель лекции: постановка задачи об алгоритмах ЭЦП. Цифровые подписи, основанные на асимметричных криптосистемах. Стандарты ЦП: DSS; ГОСТ Р 34.10-94; ГОСТ Р 34.10-2001. Цифровые подписи, основанные на симметричных криптосистемах. ФУНКЦИИ ХЭШИРОВАНИЯ. Функции хэширования: SHA; SHA-256; SHA-512; SHA-384; ГОСТ Р 34.11-94; MD5.

Подписание (аутентификация). Процесс подписания состоит главным образом из выбора секретного случайного числа, создания третьей точки на кривой, вычисления двух подписей и передачи сообщения и подписей.

Алиса выбирает секретное случайное число, r, между 1 и q - 1.

Алиса выбирает третью точку на кривой, P (U, v) = r × e1 (.....).

Алиса использует первые координаты P (u, v), чтобы вычислить первую подпись S1. Это означает S1 = u mod q.

Алиса использует дайджест сообщения, свой секретный ключ и секретное случайное число r и S1, чтобы вычислить вторую подпись S2 = (h (M) + d × S1) r-1 mod q.

Алиса передает M, S1 и S2.

Проверка (верификация). Проверка состоит из восстановления третьей точки и подтверждения, что первая координата эквивалентна S1 по модулю q. Третья точка создаётся подписывающим лицом, использующим секретное случайное число r. Верификатор не имеет этого значения. Он должен создать третью точку из дайджеста сообщения, S1, и S2.

Боб применяет M, S1 и S2 для создания двух промежуточных результатов A и B: A = h(M)S2-1 mod q и B = S2-1S1 mod q. Затем Боб восстанавливает третью точку T(x,y) = A × e1(… , …) + B × e2(… , …) 2. Боб использует первую координату из T(x,y), чтобы проверить сообщение. Если x = S1 mod q, подпись принимается, иначе - отклоняется.

Передача сообщения отправителем (пользователь А) получателю (пользователь В) предполагает передачу данных, побуждающую пользователей к определённым действиям. Передача данных может представлять собой: передачу фондов между банками; продажу акций или облигаций на автоматизированном рынке; передачу приказов (сигналов) по каналам электросвязи. Участники коммуникации защищаются от злонамеренных действий (отказ от переданного сообщения; фальсификация сообщения; изменения сообщения; маскировка пользователя под другого). Для верификации (подтверждения) сообщения М (пользователя А – пользователю В) необходимо следующее: отправитель (А) должен внести в М подпись, содержащую дополнительную информацию, зависящую от М и от получателя сообщения (В) (известной только отправителю закрытой информации kA).

Подпись М: SIG {kA, M, идентификатор B} в сообщении для пользователя B нельзя составить без kA. Процедура составления ЦП зависит от времени (т.е. устаревшие сообщения не могут быть использованы.). Пользователь B имеет возможность удостовериться в том, что SIG {kA, M, идентификатор B} – есть правильная п о д п и с ь M пользователем A.

1) подпись сообщения - способ шифрования M путём криптографического преобразования, закрываемым элементом kA в преобразовании. Идентификатор B, M → SIG {kA, M, идентификатор B} является ключом криптопреобразования. В криптосистемах kA принадлежит Конечному множеству ключей “K”. Исчерпывающая проверка всех ключей, задаваемых соответствующими парами < M, идентификатор B > ↔ SIG {kA, M, идентификатор B}, в общем должна привести к определению ключа kA злоумышленником. Если множество “K” велико и ключ k определён методом случайного выбора, то полная проверка ключей невозможна. Определение SIG {kA, M, идентификатор B} без kA с вычислительной точки зрения эквивалентно поиску ключа;

2) доступ к аппаратуре, программам и файлам системы обработки информации контролируется паролями. П о д п и с ь – это пароль, зависящий от отправителя, получателя информации и содержания передаваемого сообщения;

3) подпись меняется от сообщения к сообщению для предупреждения её повторного использования с целью проверки нового сообщения. ЦП отличается от рукописной. Рукописная не зависит от времени составления и данных. Цифровая и рукописная подписи идентичны - они характерны только для данного владельца;

4) получатель информации должен уметь удостоверять подлинность ЦП. (В коммерческих сделках это - нотариус). Установление подлинности ЦП – это процесс, посредством которого каждая сторона устанавливает подлинность другой.

В основе криптографического контроля целостности лежат два понятия: хэш-функция; электронная цифровая подпись (ЭЦП).

Хэш-функция – это труднообратимое преобразование данных (односторонняя функция), реализуемое, как правило, средствами симметричного шифрования со связыванием блоков. Результат шифрования последнего блока (зависящий от всех предыдущих) и служит результатом хэш-функции. Хэш-функцию обозначают через h, исходные данные – через T, проверяемые данные – через T'. Контроль целостности данных сводится к проверке равенства h(T') = h(T). Если оно выполнено, считается, что T' = T. Для проверки целостности данных используется дайджест (MD – Message Digest). Совпадение дайджестов для различных данных называется коллизией. Коллизии возможны, поскольку мощность множества дайджестов меньше, чем мощность множества хэшируемых данных, однако то, что h - функция односторонняя, означает, что за приемлемое время специально организовать коллизию невозможно. Все криптографические хэш-функции (h) должны создавать дайджест (MD – Message Digest) фиксированного

размера из сообщения переменного размера. Применяют такую функцию, используя итерацию. Вместо h с вводом переменного размера создана и используется необходимое количество раз h функция с вводом фиксированного размера, называемая функцией сжатия. Она сжимает n-битовую строку и создает m-битовую строку, где n>m. Эта схема Меркеля-Дамгарда (Merkle-Damgard) - итеративная хэш-функция. Если функция сжатия в схеме Меркеля-Дамгарда устойчива к коллизии, хэш-функция также устойчива к коллизии. Схема Меркеля-Дамгарда стала основанием для многих функций криптографического хэширования.

SHA-512 - версия SHA (Secure Hash Algorithm) - алгоритм безопасного хэширования с 512-битовым дайджестом сообщения. Она похожа на другие алгоритмы этого семейства, которые основаны на схеме Меркеля-Дамгарда.

Применение асимметричного шифрования.

Пусть E(T) - результат зашифрования текста T с помощью открытого ключа, а D(T) – результат расшифрования текста T (шифрованного) с помощью секретного ключа. Для реализации ЭЦП, необходимо выполнение тождества E(D(T)) = D(E(T)) = T.

При использовании асимметричных методов шифрования (в ЭЦП) необходимо иметь гарантию подлинности пары (имя пользователя, открытый ключ пользователя). Для решения этой задачи в спецификациях X.509 вводятся понятия цифрового сертификата и удостоверяющего центра.

Удостоверяющий центр (УЦ) – компонент глобальной службы каталогов, (отвечает за управление криптоключами пользователей). Открытые ключи и информация о USER-ах хранится в УЦ в виде цифровых сертификатов, имеющих следующую структуру:

а) порядковый № сертификата;

б) идентификатор алгоритма ЭП;

в) имя УЦ;

г) срок годности;

д) имя владельца сертификата;

е) открытые ключи владельца сертификата;

ж) идентификаторы алгоритмов, ассоциированных с открытыми ключами владельца сертификата;

и) ЭП, сгенерированная с использованием секретного ключа УЦ (подписывается результат хэширования всей информации, хранящейся в сертификате).

Цифровые сертификаты обладают следующими свойствами:

1) любой пользователь, знающий открытый ключ УЦ, может узнать открытые ключи других клиентов центра и проверить целостность сертификата;

2) никто, кроме УЦ, не может модифицировать информацию о USER без нарушения целостности сертификата.

В спецификациях X.509 пары ключей могут порождаться следующими способами:

- ключи может генерировать сам пользователь. (Секретный ключ не попадает в руки третьих лиц, однако, нужно решать задачу безопасной связи с удостоверяющим центром);

- ключи генерирует доверенное лицо (необходимо решать задачи безопасной доставки секретного ключа владельцу и предоставления доверенных данных для создания сертификата);

- ключи генерируются удостоверяющим центром.

6 лекция. Управление криптографическими ключами

Цель лекции: рассматриваются обычная система управления ключами. Управление ключами, основанное на системах с открытым ключом. Протокол обмена секретным ключом. Использование сертификатов. Протоколы аутентификации. Анонимное распределение ключей.

Распределение с симметричными ключами. Для шифрования больших сообщений криптография с симметричными ключами более эффективна, чем криптография с асимметричными ключами. Криптография с симметричными ключами нуждается в ключе засекречивания, который используется двумя сторонами. Если Алиса должна обмениваться конфиденциальными сообщениями с N людьми, она нуждается в N различных ключах. Если же N людей должны общаться друг с другом, тогда необходимое общее количество ключей равно N (N - l). Поскольку Алиса и Боб используют два одинаковых ключа для двунаправленной связи для обоих направлений, тогда необходимое общее количество ключей нужно только N (N - 1)/2. Это называется N-проблемой, т.к. число требуемых ключей для N объектов - N2. Число ключей - не единственная проблема; распределение ключей - другая проблема, поскольку Алисе и Бобу необходим эффективный способ поддерживания и распределения ключей засекречивания. Практическое решение - привлечение третьего лица, которому доверяют. Оно называется центром распределения ключей (KDC - Key-Distribution Center). Чтобы уменьшать число ключей, каждый человек устанавливает открытый ключ засекречивания с KDC. Ключ засекречивания установлен между KDC и каждым членом сообщества. Алиса имеет ключ засекречивания с KDC - KAlice. Боб имеет ключ засекречивания с KDC - KBob. Процесс передачи конфиденциального сообщения Бобу следующий:

1) Алиса передает запрос KDC - заявление, что она нуждается в сеансе (временно) и ключе засекречивания между собой и Бобом;

2) KDC сообщает Бобу о запросе Алисы;

3) если Боб соглашается, между ними создается ключ сеанса.

Ключ засекречивания между Алисой и Бобом, который установлен с KDC, используется, чтобы подтвердить подлинность Алисы и Боба к KDC и препятствовать Еве исполнять роль любого из них. KDC создает ключ засекречивания для каждого абонента. Этот ключ засекречивания может использоваться только между абонентом и KDC, а не между двумя членами

сообщества. Если Алиса должна связаться тайно с Бобом, она нуждается в ключе засекречивания между собой и Бобом. KDC может создать ключ сеанса между Алисой и Бобом, используя их ключи с центром. Ключи Алисы и Боба используются, чтобы подтвердить доступность и полномочность Алисы и Боба к центру и друг к другу перед тем, как будет установлен ключ сеанса. После того как связь закончена, ключ сеанса больше не нужен. Симметричный ключ сеанса между двумя сторонами используется только однажды.

Цербер - протокол установления подлинности и в то же самое время – KDC. Несколько систем используют протокол Цербер. Первоначально разработанный в MIT, он прошел несколько версий. Протокол Цербер включает в себя работу с тремя серверами: опознавательный сервер (AS - Authentication Server), сервер, предоставляющий билет (TGS - Ticket-Granting Server), и реальный сервер (сервер обработки данных), который обеспечивает услуги. Опознавательный сервер (AS- Authentication Server) - в протоколе Цербер - KDC. Каждому пользователю, зарегистрированному в AS, предоставляют пользовательский идентификационный код и пароль. AS имеет базу данных с этими идентификационными кодами и соответствующими паролями. AS верифицирует пользователя, выдает ключ сеанса, который используется между Алисой и TGS- Ticket-Granting Server, и передает билет для TGS. Предоставляющий билет сервер (TGS) вырабатывает билет для реального сервера (Боба). Он обеспечивает ключ сеанса (KAB) между Алисой и Бобом. Протокол Цербер отделяет верификацию пользователя от выдачи билета. Этим способом Алиса проверяет свой ID с AS только один раз. В контакт с TGS она может войти много раз, чтобы получить билеты для различных реальных серверов. Реальный сервер (Боб) обеспечивает услуги для пользователя (Алиса). Цербер разработан для взаимодействия с программой «клиент-сервер», такой как, например, протокол передачи файлов FTP (File Transfer Protocol), в котором пользователь использует процесс клиента, чтобы обратиться к процессу сервера. Цербер не используется для установления подлинности «человек-человек».

Работа цербера:

1) Алиса передает свой запрос AS в открытом тексте, используя свой зарегистрированный код идентификации;

2) AS передает сообщение, зашифрованное постоянным симметричным ключом Алисы, KA. AS-сообщение содержит два объекта: ключ сеанса, KA-TGS, который используется Алисой, чтобы войти в контакт с TGS, и билет для TGS, который зашифрован TGS-симметричным ключом (KAS-TGS). Алиса не знает KA-AS, но когда сообщение прибывает, она печатает (сообщает) свой симметричный пароль. Пароль и соответствующий алгоритм вместе создают KA-AS, если пароль правильный. Пароль затем немедленно уничтожают; его не передают по сети, и он не остается в терминале. Он используется только на мгновение, чтобы создать KA-AS. Процесс теперь

использует KA-AS для того, чтобы расшифровывать передаваемое сообщение KA-TGS и извлечь билет;

3) Алиса теперь передает три объекта TGS. Первый - билет, полученный от AS. Второй - имя реального сервера (Боб), третий - метку времени, которая зашифрована ключом KA-TGS. Метка времени предотвращает ложный ответ Евы;

4) теперь TGS передает два билета: каждый содержит ключ сеанса между Алисой и Бобом, KA-B. Билет для Алисы – зашифрованный KA-TGS, билет для Боба - зашифрованный с ключом Боба KTGS-B. Обратите внимание, что Ева не может извлечь KAB, потому что Ева не знает KA-TGS или KTGS-B;

5) она не может ответить на шаг 3, потому что она не может заменить метку времени новой меткой. Она не знает KA-TGS, и даже если она будет действовать очень быстро и передаст на шаге 3 сообщение прежде, чем истечет метка времени, она все равно получит те же самые два билета, которые она не может расшифровать;

6) Алиса передает билет Боба с меткой времени, зашифрованной ключом KA-B;

7) Боб подтверждает, что получил эту информацию, прибавляя 1 к метке времени. Сообщение шифруется ключом KA-B и передается Алисе.

Алгоритмы смены ключей. Для обмена подлинными и конфиденциальными сообщениями клиенту и серверу нужны шесть криптографических объектов секретности (4 ключа и 2 вектора инициализации). Чтобы создать их, между клиентом и сервером должен быть установлен один предварительный главный секретный код (pre-master secret). SSL (Secure Sockets Layer –уровень безопасных розеток) определяет шесть методов обмена ключами, чтобы установить этот предварительный объект секретности: NULL, RSA, анонимный Диффи-Хеллман (Diffie-Hellman), кратковременный Диффи-Хеллман, фиксированный Диффи-Хеллман и Fortezza .

NULL (пустой указатель). В этом методе нет никакой смены ключей. Между клиентом и сервером не установлен предварительный главный секретный код. И клиент и сервер должны знать значение предварительного главного секретного кода. RSA. В этом методе предварительный главный секретный код - 48-байтовое случайное число, созданное клиентом, зашифрованное открытым ключом RSA-сервера и передаваемое серверу. Сервер должен передать свой сертификат шифрования/дешифрования RSA. Анонимный протокол Диффи-Хеллмана. Это самый простой и наиболее ненадежный метод. Предварительный главный секретный код устанавливают между клиентом и сервером, используя протокол Диффи-Хеллмана. При этом передают половину ключа в исходном тексте - это называется анонимным протоколом Диффи-Хеллмана. (недостаток этого метода - возможность атаки «посредника»). Кратковременный метод Диффи-Хеллмана. Чтобы сорвать атаку «посредника», может быть использована кратковременная смена ключей методом Диффи-Хеллмана. Каждая сторона

передает ключ Диффи-Хеллмана, подписанный своим секретным ключом. На приемной стороне должны проверить подпись, используя открытый ключ передатчика. Обмен открытыми ключами для проверки использует либо RSA-, либо DSS-сертификат цифровой подписи. Фиксированный метод Диффи-Хеллмана. Все объекты в группе могут подготовить фиксированные параметры (g и p). Затем каждый объект может создать фиксированную половину ключа (gx). Для дополнительной безопасности каждая отдельная половина ключа Диффи-Хеллмана вставляется в сертификат, проверенный центром сертификации (CA). Стороны отдельно не обмениваются полуключами; CA передает полуключи в специальном сертификате RSA или DSS. Когда клиент должен вычислить предварительный главный секретный код, он использует свой собственный фиксированный полуключ и полуключ сервера, полученный в сертификате. Сервер делает то же самое, но в обратном порядке. В этом методе не передаются сообщения смены ключей, а происходит только обмен сертификатами. Fortezza (образован из итальянского слова «крепость») - торговая марка американского NSA - National Security Agency (не обсуждаем из-за его сложности).

Аутентификация с нулевым разглашением.

Общая схема процедуры аутентификации с нулевым разглашением состоит из последовательности информационных обменов (итераций) между двумя участниками процедуры, по завершению которой проверяющий с заданной вероятностью делает правильный вывод об истинности проверяемого утверждения. С увеличением числа итераций возрастает вероятность правильного распознавания истинности (или ложности) утверждения.

Классическим примером неформального описания системы аутентификации с нулевым разглашением служит так называемая пещера Али-Бабы.

Пещера имеет один вход, путь от которого разветвляется в глубине пещеры на два коридора, сходящихся затем в одной точке, где установлена дверь с замком. Каждый, кто имеет ключ от замка, может переходить из одного коридора в другой в любом направлении. Одна итерация алгоритма состоит из последовательности шагов:

1) проверяющий становится в точку A;

2) доказывающий проходит в пещеру и добирается до двери (оказывается в точке C или D). Проверяющий не видит, в какой из двух коридоров тот свернул;

3) проверяющий приходит в точку B и в соответствии со своим выбором просит доказывающего выйти из определенного коридора;

4) доказывающий, если нужно, открывает дверь ключом и выходит из названного проверяющим коридора.

7 лекция. Имитозащита информации в АСУ

Цель лекции: рассматриваются защита от навязывания ложной информации. Построение имитозащищенного канала связи.

Имитозащита - защита от навязывания ложной информации. Текст остаётся открытым, но появляется возможность проверить, что его не изменяли ни случайно, ни намеренно. Имитозащита достигается за счет включения в пакет передаваемых данных имитовставки. Реализуется с помощью добавления к сообщению дополнительного кода, т.е. имитовставки, MAC (message authentication code), зависящей от содержания сообщения и секретного элемента, известного только отправителю и получателю (ключа). Закладка избыточности позволяет обнаружить внесённые в сообщение несанкционированные изменения. $T'' = (T, Y)$, где $Y = f(T, K)$. Получатель проверяет выполнение условия $Y = f(T, K)$, где K - криптографический ключ, известный только отправителю и получателю. Сообщение подлинно, если условие справедливо. В противном случае message отвергается. Пример имитовставки - контрольная сумма блоков сообщения по модулю некоторого числа (ключа). Имитозащита (в её классическом, «симметричном» виде) применяется там, где важна оперативность передачи, но не требуется полная секретность.

Защита от навязывания ложной информации. Защита от навязывания ложных сообщений в каналы связи вычислительной сети, в том числе и от ранее переданных сообщений источника, требует решения в АСУ. Защита от навязывания ложных сообщений становится возможной только тогда, когда в криптосистеме дополнительно реализована функция размножения (распространения) ошибки и ее обнаружения за счет введения в структуру зашифрованного сообщения некоторой избыточности, обеспечивающей контроль ошибок и обнаружение факта навязывания ложной информации (сообщений). Такого рода защита иначе называется защитой целостности сообщений. Рассмотрим схемы основных криптосистем с точки зрения обеспечения целостности сообщений.

Побитное шифрование потока данных.

Если исходный текст частично известен нарушителю, модификация битов текста реализуется путем инвертирования битов шифрованного текста в тех местах, где требуется инверсия битов исходного текста. Если сообщение содержит несколько контрольных знаков, они могут быть также изменены, поскольку все биты, участвующие в вычислении этих знаков, известны. Это оказывается возможным независимо от вида функции проверки избыточности (линейная или нелинейная). Нарушитель может знать эту функцию, поскольку проверочная функция не является секретной. Это означает, что шифрованные контрольные знаки, используемые в обычных протоколах передач по линиям связи, - символ контроля блока (BCC) и контроль избыточным циклическим кодом (CRC), не могут помочь

получателю выявить манипуляции с сообщениями, поскольку нарушитель может легко вычислить их значения.

Прежде чем обсудить системы засекречивания и соответствующие математические проблемы, рассмотрим трудности, которые возникают при реализации криптографических систем. Алгоритм – это процесс, описывающий проблему, которую нужно решить. При поиске математической проблемы, чтобы обосновать криптографическую систему, шифровальщики ищут такую проблему, для которой самый быстрый алгоритм берет показательное время. Чем больше времени требуется, чтобы вычислить лучший алгоритм для этой проблемы, тем более безопасной будет общее - ключевая система шифрования, основанная на той проблеме.

Существуют три типа безопасных и эффективных систем:

1) целочисленная проблема факторизации (IFP - Integer-valued Factorization Problem): RSA и Rabin-Уильям;

2) дискретная проблема логарифма - Discrete Problem Logarithm (ПРОЦЕССОР ПЕРЕДАЧИ ДАННЫХ);

3) эллиптическая кривая дискретная проблема логарифма (ECDLP-Elliptic Curves Digital Logarithm Problem).

Целочисленная проблема факторизации (IFP): RSA и Рабин-Уильям.

Описание задачи. Целочисленная проблема факторизации (IFP) находят p и q, учитывая составное число n, которое является произведением двух больших простых чисел p и q.

Обнаружение больших простых чисел - относительно простая задача. Проблема же разложения на множители, произведение двух таких чисел рассматривается в вычислительном отношении как трудная.

IFP занимала внимание математиков подобно Ферма и Гауссу более чем сто лет. В 20 годах был сделан прогресс в разрешении этой проблемы. Изобретение RSA-системы шифрования в 1978 стимулировало математиков к изучению этой проблемы. Быстродействующие ЭВМ стали доступными для выполнения и испытания сложных алгоритмов.

Разложения на множители. Имеются в основном два типа специализированных и универсальных алгоритмов разложения на множители. Специализированные алгоритмы разложения на множители пытаются эксплуатировать специальные особенности числа n, разлагаемого на множители. Текущие времена универсальных алгоритмов разложения на множители зависят только от размера n.

Один из наиболее мощных специализированных алгоритмов разложения на множители - эллиптический метод разложения на множители кривой, т.н. режим исправления ошибок (изобретен в 1985 Х. Ленстром младшим). Текущее время этого метода зависит от размера главных множителей n. Алгоритм находит сначала маленькие множители. В 1995 году А. Мюллер (студент в Universitaet des Saarlandes, Германия) нашел 44-десятичную цифру с 147-разрядным множителем и 99-десятичную цифру с 329-разрядным составным целым числом, используя режим исправления ошибок. До развития RSA системы шифрования, лучший универсальный

алгоритм разложения на множители был алгоритм цепной дроби, который имел числа множителя до 40 десятичных цифр (133 бита). Лучшие универсальные алгоритмы, используемые сегодня: квадратичное решето (QS) и решето поля цифр (NFS). Оба эти алгоритмы могут быть легко параллелизованы, чтобы разрешить разложение на множители на распределительных сетях АРМ. Квадратичное решето было разработано в 1984году. Первоначально, это применялось к числам множителя в 70-десятичной цифре 233-разрядный диапазон. В 1994г. это использовалось к множителю 129-десятичной цифры 429-разрядного номера проблемы RSA. Факторизация была выполнена через 8 месяцев примерно на 1600 компьютерах во всем мире. Долговечность для факторизации была оценена как 5000 MIPS годы.

Эксперименты доказали, что решето поля цифр (NFS) является превосходным алгоритмом для целых чисел разложения на множители, имеющих, по крайней мере, 120 десятичных цифр (400 битов). В 1996, группа с А. Ленстром использовала NFS к множителю 130-десятичной цифры 432-разрядного номера. Это - самый большой номер, разложенный на множители до настоящего времени.

Дискретная проблема логарифма.

Описание задачи. Если p - простое число, то Zp обозначает набор целых чисел 0, 1, 2,..., p - 1, где сложение и амплитудное искажение - выполняются с модулем. Существует ненулевой элемент O Zp такой, что каждый ненулевой элемент в Zp может быть написан как мощность a, такой элемент называется генератором Zp.

Дискретная проблема логарифма (процессор передачи данных) заключается в следующем: учитывая штрих p, генератор Zp и ненулевой элемент O Zp, находят уникальное целое число 0,1,2,..., p - 2, такое, что b принадлежит a l (mod p). Целое число l называется дискретным логарифмом b к основе a.

Основываясь на трудности этой проблемы, Диффи и Хеллман в 1976 году предложили ключевую схему соглашения. С тех пор были предложены многочисленные другие криптогафические протоколы, чья защита зависит от процессора передачи данных. С должным интересом этот процессор экстенсивно изучался математиками в течение прошлых 20 лет.

Программные разложения функции на множители. Криптографический алгоритм RSA использует только один тип вычислений – возведение в степень. Показатель степени определяет длительность выполнения процедуры вычислений. Чтобы обеспечить требуемый уровень надежности , показатель степени, являющийся секретным ключом , должен быть достаточно большим , поэтому для вычислений требуется много времени. Производительность вычислительных устройств с недавнего времени принято оценивать в MIPS (Million Instruction Per Second): 1MIPS=10^6 опер./с. MIPS года – такая сложность алгоритма, которая требует годовой работы компьютера чтобы его вскрыть. По отношению к эллиптическим кривым производительность 1 MIPS соответствует примерно 4·10^4

операций сложения кривой в секунду, поскольку длина ключа существенно превышает длину единицы данных. Устойчивость алгоритмов криптографии принято оценивать в MIPS годах. Иначе говоря, устойчивость – это число лет непрерывной работы, необходимое вычислителю с производительностью 1 MIPS ,чтобы взломать данный шифр.

Программное выполнение на SPARC IPC исполняют 2,000 эллиптических сложений кривой в секунду. Тогда число эллиптических сложений кривой, которые могут быть выполнены 1 механизмом MIPS в одном году: (4 x 104) · (60 x 60 x 24 x 365) ^ 240.

Например, если 10,000 компьютеров каждый в 1,000 MIPS году доступны, то эллиптическая кривая дискретного логарифма может быть вычислена через 96,000 лет.

Таблица 1 - Сравнение размеров ключей, необходимых для обеспечения эквивалентных уровней безопасности

Время на взлом MIPS лет	Размер ключа RSA/DSA	Размер ключа ECC	Отношение длин ключей RSA/DSA
10^4	512	106	5:1
10^8	768	132	6:1
10^11	1.024	160	7:1
10^20	2.048	210	10:1
10^78	21	600	35:1

Выбор основного поля Fq и эллиптической кривой E. При установке режимов эллиптической системы шифрования кривой имеются три основных пункта, которые должны быть сделаны:

1) выбор основного конечного поля Fq.

Два наиболее общего выбора в практических приложениях для основного конечного поля - F2m и Fp (где p - вспомогательный штрих). ECDLP одинаково труден для образцов, которые используют F2m и для образцов, которые используют Fp и где размеры 2m и p полей приблизительно равны. Не имелось никаких математических открытий до настоящего времени, которые показывают, что ECDLP для эллиптических кривых по F2m может быть проще или тяжелее, чем ECDLP для эллиптических кривых по Fp;

2) выбор представления для элементов Fq.

Если поле F2m выбрано как основное конечное поле, то имеются много путей, в которых элементы F2m могут быть представлены. Два наиболее эффективных пути : оптимальное, нормальное представление основания и полиномиальное представление основания. Так как элементы в одном представлении могут быть эффективно преобразованы к элементам в другом представлении, используя соответствующую матрицу изменения основания, на ECDLP не воздействует выбор представления;

3) выбор эллиптической кривой E по Fq.

8 лекция. Вопросы реализации криптографических систем

Цель лекции: рассматриваются эффективные алгоритмы и генерация ключей, генерация простых чисел. Проблемы реализации криптосистем.

MOV алгоритм приведения выдает алгоритм для ECDLP, когда эллиптическая кривая суперсингулярна. В большинстве случаев эллиптические кривые являются не суперсингулярными. Кроме того, можно легко проверить, действительно ли MOV алгоритм приведения выполним для данной эллиптической кривой. Также можно легко обнаружить, является ли данная кривая аномальной. При выборе не суперсингулярной эллиптической кривой можно выбирать кривую наугад, или можно выбирать кривую специальными свойствами, которые могут привести быстрее к эллиптической арифметике кривой. Пример специальной категории кривых, который был предложен, - кривые Koblitz . ECDLP одинаково труден для образцов, которые используют беспорядочно сгенерированные кривые, и для тех, которые используют кривые Koblitz. До настоящего времени пока нет никаких математических открытий, которые показывают, что ECDLP для беспорядочно сгенерированных эллиптических кривых проще или тяжелее, чем ECDLP для кривых Koblitz.

Потоковое шифрование данных. Наиболее очевидным является побитовое сложение входящей последовательности (сообщения) с некоторым бесконечным или периодическим ключом, получаемым от генератора ПСП[2.]. Примером стандарта потокового шифрования является RC4, разработанный Ривестом. Однако, технические подробности этого алгоритма держатся в секрете[3.]

Другим, иногда более эффективным методом потокового шифрования, является шифрование блоками, т.е. накапливается фиксированный объем информации (блок), а затем преобразованный некоторым криптографическим методом передается в канал связи.

Использование «блуждающих ключей». Проблема распределения ключей является наиболее острой в крупных информационных системах. Эта проблема снимается за счет использования открытых ключей. Но наиболее надежные криптосистемы с открытым ключом типа RSA достаточно трудоемки, а для шифрования мультимедийных данных и вовсе не пригодны.

Наиболее доступным является использование полей Галуа. За счет возведения в степень порождающего элемента можно последовательно переходить от одного числа к другому. Эти числа принимаются в качестве ключей.

---

[2] Отчасти это метод похож на гаммирование и информацию о способах генерации ПСП можно почерпнуть из соответствующей главы. Но важным отличием потокового шифрования является то, что шифрованию подвергаются не символы сообщения, а отдельные биты.

[3] Данный алгоритм является собственностью RSA Data Security, и на его экспорт правительством США наложены серьезные ограничения.

Ключевой информацией в данном случае является исходный элемент, который перед началом связи должен быть известен и отправителю и получателю.

Надежность таких методов должна быть обеспечена с учетом известности злоумышленнику используемого правила смены ключей. Перспективной задачей является реализация метода «блуждающих ключей» не для двух абонентов, а для достаточно большой сети, когда сообщения пересылаются между всеми участниками.

Шифрование, кодирование и сжатие информации. Эти три вида преобразования информации представлены в таблице 2.

Таблица 2 - Преобразования информации

Вид преобразования	Цель	Изменение объема информации после преобразования
Шифрование	передача конфиденциальной информации; обеспечение аутентификации и защиты от преднамеренных изменений;	обычно не изменяется, увеличивается лишь в цифровых сигнатурах и подписях
Помехоустойчивое кодирование	защита от искажения помехами в каналах связи	увеличивается
Сжатие (компрессия)	сокращение объема передаваемых или хранимых данных	уменьшается

Эти три вида преобразования информации дополняют друг друга, и их комплексное использование помогает эффективно использовать каналы связи для надежной защиты предаваемой информации.

Возможность объединения методов кодирования и шифрования. Посути кодирование - это элементарное шифрование, а шифрование - это элементарное помехоустойчивое кодирование.

Комбинирование алгоритмов шифрования и сжатия информации. Задача сжатия – это преобразовать сообщение в пределах одного и того же алфавита, чтобы его длина (количество букв алфавита) стала меньше, но при этом сообщение можно было восстановить без использования дополнительной информации. Наиболее популярные алгоритмы сжатия - RLE, коды Хаффмана, алгоритм Лемпеля-Зива. Для сжатия графической и видеоинформации используются алгоритмы JPEG и MPEG.

Главное достоинство алгоритмов сжатия с точки зрения криптографии состоит в том, что они изменяют статистику входного текста в сторону ее

выравнивания[4]. В обычном тексте, сжатом с помощью эффективного алгоритма, все символы имеют одинаковые частотные характеристики, и даже использование простых систем шифрования сделают текст недоступным для криптоанализа. Разработка и реализация таких универсальных методов - это перспектива информационных систем[5].

Реализация криптографических методов. Проблема реализации методов защиты информации имеет два аспекта:

1) разработку средств, реализующих криптографические алгоритмы;

2) методику использования этих средств.

Каждый из рассмотренных криптографических методов может быть реализован либо программным, либо аппаратным способом.

Возможность программной реализации обуславливается тем, что все методы криптографического преобразования формальны и могут быть представлены в виде конечной алгоритмической процедуры.

При аппаратной реализации все процедуры шифрования и дешифрования выполняются специальными электронными схемами. Наибольшее распространение получили модули, реализующие комбинированные методы.

Достоинством программных методов реализации защиты является возможность быстрого изменения алгоритмов шифрования. Недостатком программной реализации является меньшее быстродействие по сравнению с аппаратными средствами (примерно в 10 раз).

Появились комбинированные, так называемые программно-аппаратные средства шифрования. В этом случае в компьютере используется «криптографический сопроцессор - вычислительное устройство, ориентированное на выполнение криптографических операций (сложение по модулю, сдвиг и т.д.). Меняя ПО для такого устройства, можно выбирать метод шифрования. Такой метод объединяет в себе достоинства программных и аппаратных методов.

Таким образом, выбор типа реализации криптозащиты для конкретной ИС зависит от ее особенностей и должен опираться на всесторонний анализ требований, предъявляемых к системе защиты информации (СЗИ).

---

[4] Принципиально важно с точки зрения криптостойкости, чтобы сначала осуществлялось сжатие информации, а потом шифрование, но не наоборот.

[5] Так, в криптографическом пакете PGP перед шифрованием информации происходит ее сжатие по алгоритму, лицензированному у PKWARE.

Список литературы

1. Алферов А.П., Зубков А.Ю., Кузьмин А.С., Черемушкин А.В. - Основы криптографии. - М.: Гелиос АРВ, 2011.

2. Криптографические методы защиты информации / Под ред. Е. М. Сухарева. Кн. 4. - М.: Радиотехника, 2007. - 312 с: ил. (Сер. Защита информации. Редактор Е. М. Сухарев).

3. Герман О.Н. Теоретико-числовые методы в криптографии. - М.: «Академия», 2012.

4. Рябко Б.Я. Основы современной криптографии и стеганографии. - М.: «Горячая линия-телеком», 2010.

5. Фороузан Б.А. Криптография и безопасность сетей. - М.: «Бином», 2010.

6. Шнайер Б. Прикладная криптография. Протоколы,алгоритмы, исходные тексты на языке Си. - М.: «Триумф», 2012.

7. Шайхин Б.М., Байжанова Д.О. Основы информационной безопасности. Конспект лекций. - Алматы: АУЭС, 2012. - 35 с.

Содержание

Дополнительный план 2014 г. поз. 21

Берк Мурзахметович Шайхин
Гульсим Толегеновна Мусатаева
Дина Ондасыновна Байжанова

СОВРЕМЕННЫЕ КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Конспекты лекций для магистрантов по специальности
6М070400 - Вычислительная техника и программное обеспечение

Редактор Л. Т. Сластихина
Специалист по стандартизации Н. К. Молдабекова

Подписано в печать ___.___._____.
Тираж 20 экз.
Объем 2,2 уч.- изд. л.

Формат 60х84 1/16
Бумага типографская №1
Заказ _____ Цена 1100 тн.

Копировально-множительное бюро
Некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, Байтурсынулы, 126