ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Б1.Б.1.26 Криптографические методы защиты информации

Направление подготовки (специальность) 10.05.03 Информационная безопасность автоматизированных систем

Профиль образовательной программы Информационная **б**езопасность автоматизированных систем критически важных объектов

Форма обучения: очная

СОДЕРЖАНИЕ

1. **Конспект лекций**

Лекция № 1 «Введение».

Лекция № 2-3 «Законодательные и правовые основы защиты компьютерной информации и информационных технологий».

Лекция № 4 «Стойкость криптографических систем и алгоритмов».

Лекция № 5-6 «Вычислительные алгоритмы».

Лекция № 7 «Блочные и поточные шифры».

Лекция № 8-9 «Шифры DES, режимы работы DES, AES, ГОСТ 28147-89».

Лекция № 10 «Поточные шифры: РСЛОС, RC4, шифр Рона».

Лекция № 11 «Распределение ключей.».

Лекция № 12 «Общая схема функционирования систем с открытыми ключами».

Лекция № 13 «Криптосистема RSA и ее модификации. Криптосистема Эль Гамаля. Криптосистема Рабина».

Лекция № 14 «Целостность данных и аутентификация сообщений».

Лекция № 15 «Хэш-функции».

Лекция №16 «Реализация схем электронной цифровой подписи (на основе алгоритмов RSA, El Gamal, Шнорра)».

Лекция № 17 «Реализация схем электронной цифровой подписи (на основе алгоритмов RSA, El Gamal, Шнорра)».

Лекция № 18 «Криптографические протоколы».

Лекция № 19 «Тесты на простоту и факторизация».

2. **Методические указания по выполнению лабораторных работ**

Лабораторная работа № ЛР-1 «Поточные системы шифрования (РСЛОС, RC4, Рона)»

Лабораторная работа № ЛР-2-3 «Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)»

Лабораторная работа № ЛР-4 «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)».

Лабораторная работа № ЛР-5-6 «Асимметричные криптосистемы (RSA, El Gamal, Рабина)».

Лабораторная работа № ЛР-7 «Программная реализация асимметричных криптосистем (RSA, El Gamal, Рабина)».

Лабораторная работа № ЛР-8-9 Исследование тестов на простоту и алгоритмы факторизации.

3. Методические указания по проведению практических занятий ……

Практическое занятие № ПЗ-1 «Поточные системы шифрования (РСЛОС, RC4, Рона)».

Практическое занятие № ПЗ-2 «Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)».

Практическое занятие № ПЗ-3 «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)».

Практическое занятие № ПЗ-4 «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)».

Практическое занятие № ПЗ-5 «Асимметричные криптосистемы (RSA, El Gamal, Рабина) Формирование ассиметричных криптосистем».

Практическое занятие № ПЗ-6 «Асимметричные криптосистемы (RSA, El Gamal, Рабина) Формирование ассиметричных криптосистем RSA».

Практическое занятие № ПЗ-7 «Асимметричные криптосистемы (RSA, El Gamal, Рабина) Формирование ассиметричных криптосистем Рабина».

Практическое занятие № ПЗ-8 «Асимметричные криптосистемы (RSA, El Gamal, Рабина) Формирование ассиметричных криптосистем El Gamal».

Практическое занятие № ПЗ-9 «Программная реализация асимметричных криптосистем (RSA, El Gamal, Рабина)».

1. КОНСПЕКТ ЛЕКЦИЙ

1. 1 Лекция № 1 (2 часа).
Тема: «Введение»

1.1.1 Вопросы лекции:
1. Основные понятия и определения.
2. История развития криптографии. Классификация криптографических систем.
1.1.2 Краткое содержание вопросов:

1. Основные понятия и определения.

Одно из основных понятий криптографии - шифр (араб. ХцЭъС (sifr) - «ноль», «ничто», «пустота», откуда фр. chiffre - «цифра»; арабы первыми стали заменять буквы на цифры с целью защиты исходного текста).

Под шифром понимается совокупность методов и способов обратимого преобразования информации с целью ее защиты от несанкционированного доступа (обеспечения конфиденциальности информации).

Составными элементами шифра являются:

- алфавиты для записи исходных сообщений (защищаемой информации, открытого текста) и шифрованных сообщений (шифртекстов, шифрограмм, криптограмм);
- алгоритмы криптографического преобразования (зашифрования и дешифрования);
- множество ключей.

Азбука или алфавит - форма письменности, основанная на стандартном наборе знаков, один или набор которых соответствуют фонемам1 языка. В общем случае алфавит для записи исходных сообщений и алфавит для записи шифрованных сообщений могут отличаться.

Алгоритм криптографического преобразования — набор правил (инструкций), определяющих содержание и порядок операций по шифрованию и дешифрованию информации.

Шифрование (зашифрование) — процесс применения шифра к защищаемой информации, т.е. преобразование исходного сообщения в зашифрованное.

Дешифрование — процесс, обратный шифрованию, т. е. преобразование шифрованного сообщения в исходное.

Ключ – переменный параметр шифра, обеспечивающий выбор одного преобразования из совокупности всевозможных для данного алгоритма и сообщения. В общем случае, ключ – это минимально необходимая информация (за исключением сообщения, алфавитов и алгоритма), необходимая для шифрования и дешифрования сообщений.

Криптография (греч. ксхрфьт — скрытый и гсЬцщ — пишу, рисую) – наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства, а также невозможности отказа от авторства) информации.

Криптоанализ (греч. ксхрфьт — скрытый и бнЬлхуз — разложение, расчленение) – наука, занимающаяся вопросами оценки сильных и слабых сторон методов шифрования, а также разработкой методов, позволяющих взламывать криптосистемы.

2. История развития криптографии. Классификация криптографических систем.

История криптографии насчитывает около 4 тысяч лет. В качестве основного критерия периодизации криптографии возможно использовать технологические характеристики используемых методов шифрования.

Первый период (приблизительно с 3-го тысячелетия до н. э.) характеризуется господством моноалфавитных шифров (основной принцип — замена алфавита исходного текста другим алфавитом через замену букв другими буквами или символами). Второй период (хронологические рамки — с IX века на Ближнем Востоке и с XV века в Европе (Леон Баттиста Альберти) — до начала XX века) ознаменовался введением в обиход полиалфавитных шифров. Третий период (с начала и до середины XX века) характеризуется внедрением электромеханических устройств в работу шифровальщиков. При этом продолжалось использование полиалфавитных шифров.

Четвёртый период — с середины до 70-х годов XX века — период перехода к математической криптографии. В работе Шеннона появляются строгие математические

определения количества информации, передачи данных, энтропии, функций шифрования. Обязательным этапом создания шифра считается изучение его уязвимости к различным известным атакам — линейному и дифференциальному криптоанализам. Однако, до 1975 года криптография оставалась «классической», или же, более корректно, криптографией с секретным ключом.

Современный период развития криптографии (с конца 1970-х годов по настоящее время) отличается зарождением и развитием нового направления — криптография с открытым ключом. Её появление знаменуется не только новыми техническими возможностями, но и сравнительно широким распространением криптографии для использования частными лицами (в предыдущие эпохи использование криптографии было исключительной прерогативой государства). Правовое регулирование использования криптографии частными лицами в разных странах сильно различается — от разрешения до полного запрета.

Современная криптография образует отдельное научное направление на стыке математики и информатики — работы в этой области публикуются в научных журналах, организуются регулярные конференции. Практическое применение криптографии стало неотъемлемой частью жизни современного общества — её используют в таких отраслях как электронная коммерция, электронный документооборот (включая цифровые подписи), телекоммуникации и других.

1. 2 Лекция № 2-3 (4 часа).

Тема: «Законодательные и правовые основы защиты компьютерной информации и информационных технологий»

1.2.1 Вопросы лекции:

1. Законодательные основы криптографии.

2.Правовые основы криптографии.

1.2.2 Краткое содержание вопросов:

1. Законодательные основы криптографии.

Распространенность криптографических технологий обусловливает потребность в их осмыслении с позиций цивилистической доктрины. В литературе существует ряд работ, посвященных правовому регулированию электронного документооборота. Однако в целом проблема соотношения важнейших институтов гражданского права и криптографических технологий исследована недостаточно.

Представляется, что гражданско-правовое значение криптографии может быть охарактеризовано следующим образом.

Во-первых, шифрование и/или ЭЦП могут использоваться как средство обособления невещественных объектов гражданских прав, если они физически представлены в виде записей, электронных документов (файлов).

Как известно, обособление (индивидуализация) является ключевым вопросом для любых объектов абсолютных гражданских прав, будь то вещи или невещные объекты гражданских прав (результаты интеллектуальной деятельности и т.д.). Кратко, но емко, сформулировал эту важнейшую мысль В.А. Белов: "Объектом всякого абсолютного права может быть только уникальная или индивидуализированная субстанция".

Рассматривая объекты исключительных прав, В.А. Дозорцев называл основные три способа их обособления (индивидуализации):

обособление в силу уникальной формы объекта, что характерно для объектов авторского права;

обособление формализацией признаков объекта, в рамках регистрационной системы. Это характерно для объектов патентного права;

обособление посредством сохранения конфиденциальности объекта. Такой способ используется применительно к ноу-хау.

Для дополнительного обособления объектов авторского права, представленных в электронно-цифровой форме, удобно использовать средства криптографии. Проблема заключается в том, что экземпляры произведений в традиционной форме, будучи вещами (книги, картины и т.п.) легко могут быть индивидуализированы. Тогда как произведения, выраженные в цифровой форме, представляют собой легко копируемую информацию (файл), копии которой даже теоретически не имеют своей уникальности. Невозможно различить оригинал и копию, в т.ч. и ту копию, которая сделана незаконно. Данное обстоятельство затрудняет реализацию и защиту авторских прав их обладателями.

Снабдив же файл ЭЦП, можно придать внешнюю уникальность самой записи (файлу, информации), в которой представлено охраняемое произведение. Это не умалит уникальности содержания произведения, предоставив дополнительные технические возможности для реализации авторских прав и их самозащиты.

2. Правовые основы криптографии.

Широко практикуется "подписание" программного обеспечения (отдельных драйверов, обновлений программ и др.), здесь ЭЦП подтверждает, что данной файл действительно является обновлением программы, исходящим от правообладателя первоначальной программы, а не подброшенным программой-вирусом.

Возможность обособления с помощью ЭЦП произведений в электронной форме также ценна тогда, когда на одной вещи (компьютер-сервер) физически размещается несколько различных охраняемых произведений, принадлежащих разным лицам (т.н. "хостинг").

Добавим также, что концепция управления исключительными авторскими правами на произведения в электронной (цифровой) форме - Digital Right Management (DRM) во многом предполагает использование криптографических методов. Достаточно привести пример "электронных книг" формата *.pdf компании Adobe.

Обособление объекта гражданских прав может быть осуществлено и путем фактического ограничения доступа к нему. Для объектов, физически представленных в виде записей (файлов, электронных документов), ограничение доступа чаще всего организуется путем:

изменения формы представления нематериального объекта, перевод его в нечитаемую форму. Речь идет о шифровании информации, т.е. криптографическом методе;

собственно ограничением доступа средствами технической защиты информации (парольная защита, межсетевые экраны, экранированные ЭВМ и др.).

Например, для такого объекта гражданских прав как информация, составляющая коммерческую тайну, ограничение доступа является одним из условий охраноспособности (ст. 139 ГК РФ; ч. 1, ч. 2 и ч. 4 ст. 10 ФЗ "О коммерческой тайне"). И поскольку ограничение доступа может быть достигнуто путем шифрования, то шифрование такой информации следует квалифицировать как факт, свидетельствующий о принятии мер к

охране конфиденциальности коммерческой тайны. Таким образом, шифрование может выступать как часть юридического состава, порождающего возникновение субъективного права на информацию, составляющую коммерческую тайну.

Во-вторых, шифрование может использоваться в качестве превентивной меры самозащиты гражданских прав. Речь идет, в основном, о самозащите абсолютных гражданских прав на конфиденциальную информацию (коммерческая тайна, личная тайна, семейная тайна). Очевидно, что труднее нарушить право собственности на вещь, помещенную в прочный сейф. Точно так же, злоумышленнику сложнее нарушить право на информацию, составляющую коммерческую тайну, если она зашифрована надежным СКЗИ. Суть самозащиты в этом случае заключается в том, что фактическими мерами затруднен неправомерный доступ к охраняемой информации, ее прочтение и понимание.

При этом речь может идти не только о статичном хранении зашифрованной информации. На практике используются средства защищенного внутрикорпоративного обмена конфиденциальной информацией, т.н. VPN (Virtual Personal Network), осуществляющие шифрование трафика.

1. 3 Лекция № 4 (2 часа).

Тема: «Стойкость криптографических систем и алгоритмов»

1.3.1 Вопросы лекции:

1. Энтропия, теоретическая и практическая стойкость, вычислительная стойкость.

2. Теоретико-информационная стойкость.

1.3.2 Краткое содержание вопросов:

1. Энтропия, теоретическая и практическая стойкость, вычислительная стойкость.

Два подхода к определению стойкости криптографической системы Рассмотрим условия, которым должна удовлетворять криптосистема для надежной защиты информации. Стойкость зашифрованной информации (криптографическая стойкость, или просто стойкость) зависит от возможности несанкционированного чтения данных.

Существует два типа стойкости: теортическая (математическая) и практическая. Эти

концепции были предложены в классической работе Шеннона (Shannon,

1949). Термин "практическая стойкость" не означает, что определение не является математически строгим. Стойкость обоих типов стойкости в следующем. Теоретическая стойкость основана на факте, что криптосистема моделируется некоторым формальным

объектом, и для этой модели формулируются определенные условия невозможности раскола криптосистемы посторонним лицом. Обычно полагается, что доступная злоумышленнику информация должна быть недостаточной для определения открытого текста, даже если информация о криптосистеме несекретна. В качестве меры практической стойкости мы принимаем работу, т.е. число операций или временную сложность определения открытой информации посторонним лицом, либо средние значения этих характеристик над множеством всех открытых текстов. В этом случае цель состоит в получении максимальной сложности задачи несанкционированного

дешифрования. Следует отметить, что существует два подхода к построению практически стойких шифров. В первом случае строится криптосистема, и затем показывается, что ее раскол является сложной задачей. Во втором случае выбирается некоторая сложная

математическая задача, и затем строится соответствующая криптосистема, чей раскол эквивалентен решению этой задачи.

2. Теоретико-информационная стойкость

Вид стойкости теоретической, определяемый с точки зрения математической теории информации. С. т.-и. криптосистемы обычно характеризуется количеством информации (всмысле К. Шеннона) относительно неизвестного противнику и/или нарушителю элемента криптосистемы, содержащимся в перехваченном тексте шифрованном или других доступных данных и вычисленным в рамках той или иной вероятностной модели. Говорят также, что с. т.-и. криптосистемы характеризует ее способность противостоять атакам со стороны противника и/или нарушителя, располагающего неограниченными вычислительными ресурсами. Криптография занимается исследованием методов защиты информации и анализом их эффективности. Криптография представляет собой совокупность методов преобразования данных, направленных на то, чтобы сделать эти данные бесполезными для противника. Такие преобразования позволяют решить две главные проблемы защиты данных: проблему конфиденциальности (лишение противника возможности извлечь информацию из канала связи) и проблему целостности (лишение противника возможности изменить сообщение так, чтобы изменился его смысл, или ввести ложную информацию в канал связи).

Проблемы конфиденциальности и целостности информации тесно связаны между собой, поэтому методы решения одной из них часто применимы для решения другой.

Криптография дает возможность преобразовать информацию таким образом, что ее прочтение (восстановление) возможно только при знании ключа. Под криптографическим ключом подразумевается некоторая секретная информация, известная законному собственнику (пользователю) информации и неизвестная нарушителю.

1. 4 Лекция № 5-6 (4 часа).

Тема: « Вычислительные алгоритмы»

1.4.1 Вопросы лекции:

1. Модульная арифметика.

2. Примеры вычислений по модульной арифметики.

1.4.2 Краткое содержание вопросов:

1. Модульная арифметика.

Если два целых числа и при делении на дают одинаковые остатки, то они называются сравнимыми (или равноостаточными) по модулю числа .

Для фиксированного натурального числа отношение сравнимости по модулю обладает следующими свойствами:

1. Рефлексивность

2.Транзитивность

3.Симметричность

Таким образом, отношение сравнимости по модулю является отношением эквивалентности на множестве целых чисел.

В криптографии и криптоанализе часто бывает необходимо сложить две последовательности чисел или же вычесть одну из другой. Такое сложение и вычитание производится, как правило, не с помощью обычных арифметических действий, а с помощью операций, называемых модульной арифметикой. В модульной арифметике сложение, вычитание выполняется относительно некоторого фиксированного числа, которое называется модуль. Типичными значениями модулей, используемые в криптографии, являются 2, 10 и 26. Какой бы модуль мы ни взяли, все встречающиеся числа заменяются на остатки от деления этих чисел. Если в остатке получается отрицательное число, то к нему прибавляют значение модуля, чтобы остаток стал неотрицательным. Например, если используется модуль 26, то единственно возможные числа лежат в диапазоне от 0 до 25. Так, если прибавить 17 к 19, то результат равен 10, поскольку 17+19 = 36, а 36 при делении на 26 дает остаток 10. Чтобы указать, что используется модуль 26, принята форма записи:

17+19=10(mod26).

Если вычесть 19 из 17, то результат (-2) получается отрицательным, поэтому к нему прибавляется 26, и в итоге получается 24.

При сложении по модулю 26 двух числовых последовательностей сформулированные правила сложения применяются в каждой паре чисел по отдельности, без «переноса» на следующую пару. Аналогично, при вычитании по модулю 26 одной числовой последовательности из другой правила вычитания применяются к каждой паре чисел по отдельности, без «заимствования» из следующей пары.

Пример 1.1

Сложить по модулю 26 последовательности 15 11 23 06 11 и 17 04 14 19 23

Решение

15 11 23 06 11

17 04 14 19 23

32 15 37 25 34

06 15 11 25 08

и в результате 06 15 11 25 08

Если модуль равен 10, то используются числа от 0 до 9; при модуле 2 – только 0 и 1

Арифметика по модулю 2, или, как ее обычно называют, двоичная (бинарная) арифметика, имеет особое значение, поскольку в этом случае сложение и вычитание являются идентичными операциями, т.е всегда дают одинаковый результат, а именно:

0 0 1 1 0 0 1 1

+ 0 1 0 1- 0 1 0 1

0 1 1 2 0 -1 1 0

0 1 1 0 0 1 1 0 (mod 2) в обоих случаях.

2. Примеры вычислений по модульной арифметики.

В теории чисел криптографии и других областях науки часто возникает задача отыскания решений сравнения первой степени вида:

Решение такого сравнения начинается с вычисления НОД(a, m)=d. При этом возможны 2 случая:

Если b не кратно d, то у сравнения нет решений.

Если b кратно d, то у сравнения существует единственное решение по модулю m / d или,

что то же самое, dрешений по модулю m. В этом случае в результате сокращения исходного сравнения на d получается сравнение:

где a1 = a / d, b1 = b / d и m1 = m / d являются целыми числами, причем a1 и m1 взаимно просты. Поэтомучисло a1 можно обратить по модулю m1, то есть найти такое число c, что другими словами, Теперь решение находится умножением полученного сравнения на c:

Практическое вычисление значения c можно осуществить разными способами: с помощью теоремы Эйлера,алгоритма Евклида, теории цепных дробей и др.

В частности, теорема Эйлера позволяет записать значение c в виде:

Пример: решим уравнение Здесь d = 2, поэтому по модулю 22 сравнение имеетдва решения. Заменим 26 на 4,

сравнимое с ним по модулю 22, и затем сократим все 3 числа на 2:

Поскольку 2 взаимно просто с модулем 11, можно сократить левую и правую части на 2. В итоге получаемодно решение по модулю 11: , эквивалентное двум решениям по модулю 22:

.

Сравнения второй степени

Решение сравнений второй степени сводится к выяснению, является ли данное число квадратичным вычетом и последующему вычислению квадратного корня поданному

модулю.

Множество всех чисел, сравнимых с a по модулю n называется классом вычетов a по

модулю n, и обычнообозначается [a]n или Таким образом, сравнение равносильно равенству классов вычетов [a]n = [b]n.

Поскольку сравнение по модулю n является отношением эквивалентности на множестве

целых чисел ,то классы вычетов по модулю n представляют собой классы эквивалентности; их количество равно n. Множество всех классов вычетов по модулю n обозначается или .

Операции сложения и умножения на индуцируют соответствующие операции на множестве :

[a]n + [b]n = [a + b]n

Относительно этих операций множество является конечным кольцом, а если n простое — конечным полем.

1. 5 Лекция № 7 (2 часа).

Тема: «Блочные и поточные шифры»

1.5.1 Вопросы лекции:

1. Блочные шифры.

2. Поточные шифры.

1.5.2 Краткое содержание вопросов:

1. Блочные шифры.

Блочный шифр — разновидность симметричного шифра. Особенностью блочного шифра является обработка блока нескольких байт за одну итерацию (как правило 8 или 16). Блочные криптосистемы разбивают текст сообщения на отдельные блоки и затем осуществляют преобразование этих блоков с использованием ключа.

Преобразование должно использовать следующие принципы:

- Рассеивание (diffusion) — то есть изменение любого знака открытого текста или ключа влияет на большое число знаков шифротекста, что скрывает статистические свойства открытого текста;
- Перемешивание (confusion) — использование преобразований, затрудняющих получение статистических зависимостей между шифротектстом и открытым текстом.

К достоинствам блочных шифров относят похожесть процедур шифрования и расшифрования, которые, как правило, отличаются лишь порядком действий. Это упрощает создание устройств шифрования, так как позволяет использовать одни и те же блоки в цепях шифрования и дешифрования.

Основная идея

Блочный шифр состоит из двух взаимосвязанных алгоритмов: алгоритм шифрования E и алгоритм расшифрования E-1. Входными данными служат блок размером n бит и k-битный ключ. На выходе получается n-битный зашифрованный блок. Для любого фиксированного ключа функция расшифрования является обратной к функции шифрования для любого блока M и ключа K.

Для любого ключа K, EK является биективной функцией (перестановкой) на множестве n-битных блоков.

Размер блока n — это фиксированный параметр блочного шифра, обычно равный 64 или 128 битам, хотя некоторые шифры допускают несколько различных значений. Длина 64 бита была приемлема до середины 90-х годов, затем использовалась длина 128 бит, что примерно соответствует размеру машинного слова и позволяет эффективную реализацию на большинстве распространённых вычислительных платформах. Различные схемы шифрования позволяют зашифровывать открытый текст произвольной длины. Каждая имеет определенные характеристики: вероятность ошибки, простота доступа, уязвимость к атакам. Типичными размера ключа являются 40, 56, 64, 80, 128, 192 и 256 бит. В 2006 г. 80-битный ключ способен был предотвратить атаку грубой силой.

2. Поточные шифры.

Поточный шифр — 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. Поточный шифр реализует другой подход к симметричному шифрованию, нежели блочные шифры.

Классификация поточных шифров

Допустим, например, что в режиме гаммирования для поточных шифров при передаче по каналу связи произошло искажение одного знака шифротекста. Очевидно, что в этом случае все знаки, принятые без искажения, будут расшифрованы правильно. Произойдёт потеря лишь одного знака текста. А теперь представим, что один из знаков шифротекста при передаче по каналу связи был потерян. Это приведёт к неправильному расшифрованию всего текста, следующего за потерянным знаком. Практически во всех каналах передачи данных для поточных систем шифрования присутствуют помехи. Поэтому для предотвращения потери информации решают проблему синхронизации шифрования и расшифрования текста. По способу решения этой проблемы шифрсистемы подразделяются на синхронные и системы с самосинхронизацией.

1. Синхронные поточные шифры

Определение:

Синхронные поточные шифры (СПШ) — шифры, в которых поток ключей генерируется независимо от открытого текста и шифротекста.

При шифровании генератор потока ключей выдаёт биты потока ключей, которые идентичны битам потока ключей при дешифровании. Потеря знака шифротекста приведёт к нарушению синхронизации между этими двумя генераторами и невозможности расшифрования оставшейся части сообщения. Очевидно, что в этой ситуации отправитель и получатель должны повторно синхронизоваться для продолжения работы.

Обычно синхронизация производится вставкой в передаваемое сообщение специальных маркеров. В результате этого пропущенный при передаче знак приводит к неверному расшифрованию лишь до тех пор, пока не будет принят один из маркеров.

Заметим, что выполняться синхронизация должна так, чтобы ни одна часть потока ключей не была повторена. Поэтому переводить генератор в более раннее состояние не имеет смысла.

Плюсы СПШ:

- отсутствие эффекта распространения ошибок (только искажённый бит будет расшифрован неверно);
- предохраняют от любых вставок и удалений шифротекста, так как они приведут к потере синхронизации и будут обнаружены.

Минусы СПШ:

- уязвимы к изменению отдельных бит шифрованного текста. Если злоумышленнику известен открытый текст, он может изменить эти биты так, чтобы они расшифровывались, как ему надо.

2. Самосинхронизирующиеся поточные шифры

Определение:

Самосинхронизирующиеся поточные шифры (асинхронные поточные шифры (АПШ)) – шифры, в которых поток ключей создаётся функцией ключа и фиксированного числа знаков шифротекста.

Итак, внутреннее состояние генератора потока ключей является функцией предыдущих N битов шифротекста. Поэтому расшифрующий генератор потока ключей, приняв N битов, автоматически синхронизируется с шифрующим генератором.

Реализация этого режима происходит следующим образом: каждое сообщение начинается случайным заголовком длиной N битов; заголовок шифруется, передаётся и расшифровывается; расшифровка является неправильной, зато после этих N бит оба генератора будут синхронизированы.

Плюсы АПШ:

- Размешивание статистики открытого текста. Так как каждый знак открытого текста влияет на следующий шифротекст, статистические свойства открытого текста распространяются на весь шифротекст. Следовательно, АПШ может быть более устойчивым к атакам на основе избыточности открытого текста, чем СПШ.

Минусы АПШ:

- распространение ошибки (каждому неправильному биту шифротекста соответствуют N ошибок в открытом тексте);
- чувствительны к вскрытию повторной передачей.

Основные отличия поточных шифров от блочных

Большинство существующих шифров с секретным ключом однозначно могут быть отнесены либо к поточным, либо к блочным шифрам. Но теоретическая граница между ними является довольно размытой. Например, используются алгоритмы блочного шифрования в режиме поточного шифрования (пример: для алгоритма DES режимы CFB и OFB). Рассмотрим основные различия между поточными и блочными шифрами не только в аспектах их безопасности и удобства, но и с точки зрения их изучения в мире:

- важнейшим достоинством поточных шифров перед блочными является высокая скорость шифрования, соизмеримая со скоростью поступления входной информации; поэтому, обеспечивается шифрование практически в реальном масштабе времени вне зависимости от объема и разрядности потока преобразуемых данных.
- в синхронных поточных шифрах (в отличие от блочных) отсутствует эффект размножения ошибок, то есть число искаженных элементов в расшифрованной последовательности равно числу искаженных элементов зашифрованной последовательности, пришедшей из канала связи.
- структура поточного ключа может иметь уязвимые места, которые дают возможность криптоаналитику получить дополнительную информацию о ключе (например, при малом периоде ключа криптоаналитик может использовать найденные части поточного ключа для дешифрования последующего закрытого текста).
- ПШ в отличие от БШ часто могут быть атакованы при помощи линейной алгебры (так как выходы отдельных регистров сдвига с обратной линейной связью могут иметь корреляцию с гаммой). Также для взлома поточных шифров весьма успешно применяется линейный и дифференциальный анализ.

Теперь о положении в мире:

- в большинстве работ по анализу и взлому блочных шифров рассматриваются алгоритмы шифрования, основанные на стандарте DES; для поточных же шифров нет выделенного направления изучения; методы взлома ПШ весьма разнообразны.
- для поточных шифров установлен набор требований, являющихся критериями надёжности (большие периоды выходных последовательностей, постулаты Голомба, нелинейность); для БШ таких чётких критериев нет.
- исследованием и разработкой поточных шифров в основном занимаются европейские криптографические центры, блочных – американские.
- исследование поточных шифров происходит более динамично, чем блочных; в последнее время не было сделано никаких заметных открытий в сфере DES-алгоритмов, в то время как в области поточных шифров случилось множество успехов и

неудач (некоторые схемы, казавшиеся стойкими, при дальнейшем исследовании не оправдали надежд изобретателей).

1. 6 Лекция № 8-9 (4 часа).

Тема: «Шифры DES, режимы работы DES, AES, ГОСТ 28147-89»

1.6.1 Вопросы лекции:

1. Шифры DES.

2. Режимы работы шифров.

1.6.2 Краткое содержание вопросов:

1. Шифры DES

Общая структура DES представлена на рис. 6.1. Процесс шифрования каждого 64-битового блока исходных данных можно разделить на три этапа:

1. начальная подготовка блока данных;
2. 16 раундов "основного цикла";
3. конечная обработка блока данных.

На первом этапе выполняется начальная перестановка 64-битного исходного блока текста, во время которой биты определенным образом переупорядочиваются.

На следующем (основном) этапе блок делится на две части (ветви) по 32 бита каждая. Правая ветвь преобразуется с использованием некоторой функции F и соответствующего частичного ключа, получаемого из основного ключа шифрования по специальному алгоритму преобразования ключей. Затем производится обмен данными между левой и правой ветвями блока. Это повторяется в цикле 16 раз.

Наконец, на третьем этапе выполняется перестановка результата, полученного после шестнадцати шагов основного цикла. Этаперестановка обратна начальной перестановке.

рис. 6.1

2. Режимы работы шифров.

Режим шифрования — метод применения блочного шифра (алгоритма), позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных. При этом для шифрования одного блока могут использоваться данные другого блока.

Обычно режимы шифрования используются для изменения процесса шифрования так, чтобы результат шифрования каждого блока был уникальным вне зависимости от шифруемых данных и не позволял сделать какие-либо выводы об их структуре. Это обусловлено, прежде всего, тем, что блочные шифры шифруют данные блоками фиксированного размера, и поэтому существует потенциальная возможность утечки информации о повторяющихся частях данных, шифруемых на одном и том же ключе.

Применение блочного шифра для нужд практического кодирования информации возможно в одном из четырех основных режимов шифрования (рис. 6.2):

1. режим электронной кодовой книги (Electronic Code Book, ECB);
2. режим сцепления блоков шифра (Cipher Block Changing, CBC);
3. режим обратной связи по шифротексту (Electronic Feedback, CFB);
4. режим обратной связи по выходу (Output Feedback, OFB).

рис.6.2

1. 7 Лекция №10 (2 часа).

Тема: «Поточные шифры: РСЛОС, RC4, шифр Рона.»

1.7.1 Вопросы лекции:

1. Характеристика шифров.

2. Режимы работы шифров.

1.7.2 Краткое содержание вопросов:

1. Характеристика шифров. Режимы работы шифров.

Большинство реальных поточных шифров основано на регистрах сдвига с обратной связью. Регистр сдвига применяют для генерации ключевой последовательности. Регистр сдвига с обратной связью состоит из двух частей: регистра сдвига и функции обратной связи.

Регистр сдвига представляет собой последовательность битов. (Количество битов определяется длиной сдвигового регистра. Если длина равна n битам, то регистр называется n-битовым регистром сдвига.) Всякий раз, когда нужно извлечь бит, все биты регистра сдвига сдвигаются вправо на 1 позицию. Новый крайний левый бит является функцией всех остальных битов регистра.

На выходе регистра оказывается один, обычно младший значащий бит. Периодом регистра называется длина получаемой последовательности до начала ее повторения.

Простейшим видом регистра сдвига с обратной связью является регистр сдвига с линейной обратной связью (РСЛОС). Обратная связь представляет собой XOR некоторых битов регистра; эти биты называютсяотводной последовательностью.

РСЛОС (n-битовый) может находиться в одном из 2n−1 внутренних состояний. Это означает, что теоретически такой регистр может генерировать псевдослучайную последовательность с периодом 2n−1 битов. (Число внутренних состояний и период равны 2n−1, потому что заполнение РСЛОС нулями приведет к тому, что сдвиговый регистр будет выдавать бесконечную последовательность нулей, что абсолютно бесполезно.) Только при определенных отводных последовательностях РСЛОС циклически пройдет через все 2n−1 внутренних состояний. Такие РСЛОС имеют максимальный период. Получившийся результат называется M-последовательностью. Для того чтобы конкретный РСЛОС имел максимальный период, многочлен, ассоциированный с отводной последовательностью, должен быть примитивным по модулю 2 — то есть не раскладываться на произведение двоичных многочленов меньшей степени.

Например, многочлен x32 + x7 + x5 + x3 + x2 + x + 1 примитивен по модулю 2. Рассмотрим этот многочлен в терминах РСЛОС с максимальным периодом. Степень многочлена задает длину РСЛОС. Свободный член многочлена всегда равен 1, и его можно опустить. Степени формальной переменной многочлена, за исключением 0-й, задают отводную последовательность, отсчитываемую от левого края сдвигового регистра. То есть члены многочлена с меньшей степенью соответствуют позициям, расположенным ближе к правому краю регистра. Тогда для взятого 32-битового сдвигового регистра новый бит генерируется с помощью XOR тридцать второго, седьмого, пятого, третьего, второго и первого битов; получающийся РСЛОС будет иметь

максимальную длину, циклически проходя до повторения через 232−1 различных значений.

Сами по себе РСЛОС являются хорошими генераторами псевдослучайных последовательностей, но они обладают некоторыми нежелательными неслучайными свойствами. Для РСЛОС длины n внутреннее состояние представляет собой предыдущие n выходных битов генератора. Даже если схема обратной связи хранится в секрете, она может быть определена по 2n выходным битам генератора с помощью алгоритма Берлекэмпа-Мэсси.

Кроме того, большие случайные числа, генерируемые с использованием идущих подряд бит этой последовательности, сильно коррелированы и для некоторых типов приложений вовсе не являются случайными. Несмотря на это, РСЛОС часто используются при разработке алгоритмов шифрования.

RC4 — это поточный шифр с переменным размером ключа, разработанный в 1987 г. Ривестом (R. Rivest) для RSA Data Security, Inc. Алгоритм работает в режиме OFB: поток ключей не зависит от открытого текста. Используется S-блок размером 8Ч8: S0, S1, S2, …, S255. Элементы представляют собой перестановку чисел от 0 до 255, а перестановка является функцией ключа переменной длины. В алгоритме применяются два счетчика, i и j, с нулевыми начальными значениями. Для генерации случайного байта выполняются следующие вычисления:

- i = (i + 1) mod 256;
- j = (j + Si) mod 256.
- Поменять местами Si и Sj.
- t = (Si + Sj) mod 256;
- K = St.

К используется в операции XOR с открытым текстом для получения шифротекста или в операции XOR с шифротекстом для получения открытого текста. Шифрование выполняется примерно в 10 раз быстрее, чем в DES. Также несложна и инициализация S-блока. Сначала S-блок заполняется по правилу: S0 = 0, S1 = 1, …, S255 = 255. После этого ключ записывается в массив: K0,K1, …, K255. Затем при начальном значении j = 0 в цикле выполняются следующие вычисления:

- for i = 0 to 255 do j = (j + Si + Ki) mod 256
- Поменять местами Si и Sj.

Компания RSA Data Security, Inc. утверждает, что алгоритм устойчив к дифференциальному и линейному криптоанализу и что он в высокой степени нелинеен. S-блок медленно изменяется при использовании: i и j обеспечивают случайное изменение каждого элемента. RC4 входит в десятки коммерческих продуктов, включая Lotus Notes, AOCE компании Apple Computer и Oracle Secure SQL. Этот алгоритм также является частью спецификации стандарта Сотовой цифровой пакетной передачи данных CDPD (Cellular Digital Packet Data).

1. 8 Лекция № 11 (2 часа).

Тема: «Распределение ключей.»

1.8.1 Вопросы лекции:

1. Схема обмена ключами.

2. Основные схемы обмена ключами.

1.8.2 Краткое содержание вопросов:

1. Схема обмена ключами.

Распределение ключей - самый ответственный процесс в управлении ключами. К нему предъявляются следующие требования:

- оперативность и точность распределения;
- скрытность распределяемых ключей. Распределение ключей между пользователями компьютерной сети реализуется двумя способами:

1. Использованием одного или нескольких центров распределения ключей;
2. Прямым обменом сеансовыми ключами между пользователями сети.

Недостаток первого подхода состоит в том, что центру распределения ключей известно, кому и какие ключи распределены, и это позволяет читать все сообщения, передаваемые по сети. Возможные злоупотребления существенно влияют на защиту. При втором подходе проблема состоит в том, чтобы надежно удостоверить подлинность субъектов сети. В обоих случаях должна быть обеспечена подлинность сеанса связи. Это можно осуществить, используя механизм запроса-ответа или механизм отметки времени.

Механизм запроса-ответа заключается в следующем. Пользователь А включает в посылаемое сообщение (запрос) для пользователя В непредсказуемый элемент (например, случайное число). При ответе пользователь В должен выполнить некоторую операцию с этим элементом (например, добавить единицу), что невозможно осуществить заранее, поскольку неизвестно, какое случайное число придет в запросе. После получения результата действий пользователя В (ответ) пользователь А может быть уверен, что сеанс является подлинным.

Механизм отметки времени предполагает фиксацию времени для каждого сообщения. Это позволяет каждому субъекту сети определить, насколько старо пришедшее сообщение, и отвергнуть его, если появится сомнение в его подлинности. При использовании отметок времени необходимо установить допустимый временной интервал задержки. В обоих случаях для защиты элемента контроля используют шифрование, чтобы быть уверенным, что ответ отправлен не злоумышленником и не изменен штемпель отметки времени.

Задача распределения ключей сводится к построению протокола распределения ключей, обеспечивающего:

- взаимное подтверждение подлинности участников сеанса;
- подтверждение достоверности сеанса механизмом запроса ответа или отметки времени;
- использование минимального числа сообщений при обмене ключами;
- возможность исключения злоупотреблений со стороны центра распределения ключей (вплоть до отказа от него).

В основу решения задачи распределения ключей целесообразно положить принцип отделения процедуры подтверждения подлинности партнеров от процедуры собственно распределения ключей. Цель такого подхода состоит в создании метода, при котором после установления подлинности участники сами формируют сеансовый ключ без участия центра распределения ключей с тем, чтобы распределитель ключей не имел возможности выявить содержание сообщений.

2. Основные схемы обмена ключами.

Передача ключа по открытым каналам была большой проблемой криптографии 20 века. Но эту проблему удалось решить после появления алгоритма Диффи-Хеллмана. Данный алгоритм позволил дать ответ на главный вопрос:

«Как при обмене зашифрованными посланиями уйти от необходимости передачи секретного кода расшифровки, который, как правило, не меньше самого послания?» Открытое распространение ключей Диффи-Хеллмана позволяет паре пользователей системы выработать общий секретный ключ, не обмениваясь секретными данными.

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

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

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

Основы криптографии с открытыми ключами были выдвинуты Уитфилдом Диффи (Whitfield Diffie) и Мартином Хеллманом (Martin Hellman), а также независимо от них Ральфом Мерклом (Ralph Merkle). Их вкладом в криптографию было убеждение, что ключи можно использовать парами — ключ зашифрования и ключ расшифрования— при условии, что исключается возможность определения содержимого ключа для расшифрования исходя из содержимого открыто передаваемого ключа для зашифрования. Диффи и Хеллман впервые представили эту идею на Национальной компьютерной конференции 1976 года, а через несколько месяцев была опубликована их основополагающая работа «New Directions in Cryptography» («Новые направления в криптографии»)

1. 9 Лекция № 12 (2 часа).

Тема: «Общая схема функционирования систем с открытыми ключами.»

1.9.1 Вопросы лекции:

1. Понятие открытого ключа.

2. Схема функционирования систем с открытым ключом.

1.9.2 Краткое содержание вопросов:

1. Понятие открытого ключа.

Ключ это секретная информация, используемая криптографическим алгоритмом при шифровании/расшифровке сообщений, постановке и проверке цифровой подписи, вычислении кодов аутентичности (MAC). При использовании одного и того же алгоритма результат шифрования зависит от ключа. Для современных алгоритмов сильной криптографии утрата ключа приводит к практической невозможности расшифровать информацию.

Согласно принципу Керхгоффса, надёжность криптографической системы должна определяться сокрытием секретных ключей, но не сокрытием используемых алгоритмов или их особенностей.

Открытый ключ — ключ, который может быть опубликован и используется для проверки подлинности подписанного документа, а также для предупреждения мошенничества со стороны заверяющего лица в виде отказа его от подписи документа. Открытый ключ подписи вычисляется, как значение некоторой функции от закрытого ключа, но знание открытого ключа не дает возможности определить закрытый ключ.

Идея криптографии с открытым ключом очень тесно связана с идеей односторонних функций, то есть таких функций , что по известному довольно просто найти значение , тогда как определение из невозможно за разумный срок.

Но сама односторонняя функция бесполезна в применении: ею можно зашифровать сообщение, но расшифровать нельзя. Поэтому криптография с открытым ключом использует односторонние функции с лазейкой. Лазейка — это некий секрет, который помогает расшифровать. То есть существует такой , что зная и , можно вычислить . К примеру, если разобрать часы на множество составных частей, то очень сложно собрать вновь работающие часы. Но если есть инструкция по сборке (лазейка), то можно легко решить эту проблему.

Понять идеи и методы криптографии с открытым ключом помогает следующий пример — хранение паролей в компьютере. Каждый пользователь в сети имеет свой пароль. При входе он указывает имя и вводит секретный пароль. Но если хранить пароль на диске компьютера, то кто-нибудь его может считать (особенно легко это сделать администратору этого компьютера) и получить доступ к секретной информации. Для решения задачи используется односторонняя функция. При создании секретного пароля в компьютере сохраняется не сам пароль, а результат вычисления функции от этого пароля и имени пользователя. Например, пользователь Алиса придумала пароль «Гладиолус». При сохранении этих данных вычисляется результат функции (АЛИСА_ГЛАДИОЛУС), пусть результатом будет строка РОМАШКА, которая и будет сохранена в системе.

Когда Алиса вводит «секретный» пароль, компьютер проверяет, даёт или нет функция, применяемая к АЛИСА_ГЛАДИОЛУС, правильный результат РОМАШКА, хранящийся на диске компьютера. Стоит изменить хотя бы одну букву в имени или в пароле, и результат функции будет совершенно другим. «Секретный» пароль не хранится в компьютере ни в каком виде. Файл паролей может быть теперь просмотрен другими пользователями без потери секретности, так как функция практически необратимая.

2. Схема функционирования систем с открытым ключом.

Пусть — пространство ключей, а и — ключи шифрования и расшифрования соответственно. — функция шифрования для произвольного ключа , такая что:

Здесь , где — пространство шифротекстов, а , где — пространство сообщений.

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

{ : } — набор шифрования, а { : } — соответствующий набор для расшифрования. Каждая пара имеет свойство: зная , невозможно решить уравнение , то

есть для данного произвольного шифротекста , невозможно найти сообщение . Это значит, что по данному невозможно определить соответствующий ключ расшифрования . является односторонней функцией, а — лазейкой.

Ниже показана схема передачи информации лицом А лицу В. Они могут быть как физическими лицами, так и организациями и так далее. Но для более лёгкого восприятия принято участников передачи отождествлять с людьми, чаще всего именуемыми Алиса и Боб. Участника, который стремится перехватить и расшифровать сообщения Алисы и Боба, чаще всего называют Евой.

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

1. 10 Лекция № 13 (2 часа).

Тема: «Криптосистема RSA и ее модификации. Криптосистема Эль Гамаля. Криптосистема Рабина»

1.10.1 Вопросы лекции:

1. Криптосистема RSA и ее модификации.
2. Криптосистема Эль Гамаля.
3. Криптосистема Рабина.

1.10.2 Краткое содержание вопросов:

1. Криптосистема RSA и ее модификации.

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

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

Опубликованная в ноябре 1976 года статья Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии» (англ. New Directions in Cryptography) перевернула представление о криптографических системах, заложив основы криптографии с открытым ключом. Разработанный впоследствии алгоритм Диффи — Хеллмана позволял двум сторонам получить общий секретный ключ, используя незащищенный канал связи. Однако этот алгоритм не решал проблему аутентификации. Без дополнительных средств пользователи не могли быть уверены, с кем именно они сгенерировали общий секретный ключ.

Изучив эту статью, трое учёных Рональд Ривест, Ади Шамир и Леонард Адлеман из Массачусетского технологического института (MIT) приступили к поискам математической функции, которая бы позволяла реализовать сформулированную Уитфилдом Диффи и Мартином Хеллманом модель криптографической системы с открытым ключом. После работы над более чем 40 возможными вариантами им удалось найти алгоритм, основанный на различии в том, насколько легко находить большие

простые числа и насколько сложно раскладывать на множители произведение двух больших простых чисел, получивший впоследствии название RSA. Система была названа по первым буквам фамилий её создателей.

В 1982 году Ривест, Шамир и Адлеман организовали компанию RSA Data Security (англ.) (в настоящий момент — подразделение EMC). В 1989 году RSA, вместе с симметричным шифром DES, упоминается в RFC 1115, тем самым начиная использование алгоритма в зарождающейся сети Internet, а в 1990 году использовать алгоритм начинает министерство обороны США.

2. Криптосистема Эль Гамаля.

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

Схема была предложена Тахером Эль-Гамалем в 1985 году.[1] Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.

Генерация ключей:

1. Генерируется случайное простое число .
2. Выбирается целое число — первообразный корень .
3. Выбирается случайное целое число такое, что .
4. Вычисляется .
5. Открытым ключом является тройка , закрытым ключом — число .

Шифросистема Эль-Гамаля является фактически одним из способов выработки открытых ключей Диффи — Хеллмана. Шифрование по схеме Эль-Гамаля не следует путать с алгоритмом цифровой подписи по схеме Эль-Гамаля.

Шифрование

Сообщение должно быть меньше числа . Сообщение шифруется следующим образом:

1. Выбирается сессионный ключ — случайное целое число такое, что
2. Вычисляются числа и .
3. Пара чисел является шифротекстом.

Нетрудно видеть, что длина шифротекста в схеме Эль-Гамаля длиннее исходного сообщения вдвое.

Расшифрование

Зная закрытый ключ , исходное сообщение можно вычислить из шифротекста по формуле:

При этом нетрудно проверить, что

и поэтому

.

Для практических вычислений больше подходит следующая формула:

3. Криптосистема Рабина.

Криптосистема Рабина — криптографическая система с открытым ключом, безопасность которой обеспечивается сложностью поиска квадратных корней составного числа. Безопасность системы, как и безопасность метода RSA, обусловлена сложностью разложения на множители больших чисел. Зашифрованное сообщение можно расшифровать 4 способами. Недостатком системы является необходимость выбора истинного сообщения из 4-х возможных.

В январе 1979 года Майкл О. Рабин опубликовал описание своей системы. Было доказано, что восстановление исходного текста из зашифрованного столь же трудно, как факторизация больших чисел. Система Рабина стала первой асимметричной криптосистемой, для которой было выполнено такое доказательство. Сложность восстановления связана с трудностью извлечения квадратного корня по модулю составного числа $N = p \cdot q$. Задача факторизации и задача по извлечению квадратного корня эквивалентны, то есть:

1. зная простые делители числа N можно извлекать квадратные корни по модулю N;

2. умея извлекать квадратные корни по модулю N, можно разложить N на простые множители.

Система Рабина, как и любая асимметричная криптосистема, использует открытый и закрытый ключи. Открытый ключ используется для шифрования сообщений и может быть опубликован для всеобщего обозрения. Закрытый ключ необходим для расшифровки и должен быть известен только получателям зашифрованных сообщений.

Процесс генерации ключей следующий:

- выбираются два случайных числа p и q с учётом следующих требований:
- числа должны быть большими (см. разрядность);
- числа должны быть простыми;
- должно выполняться условие: $p \equiv q \equiv 3 \bmod 4$.

Выполнение этих требований сильно ускоряет процедуру извлечения корней по модулю p и q;

- вычисляется число $n = p \cdot q$;
- число n — открытый ключ; числа p и q — закрытый.

Пример. Пусть $p = 7$ и $q = 11$. Тогда $n = p \cdot q = 7 \cdot 11 = 77$. Число $n = 77$ — открытый ключ, а числа $p = 7$ и $q = 11$ — закрытый. Получатель сообщает отправителям число 77. Отправители шифруют сообщение, используя число 77, и отправляют получателю. Получатель расшифровывает сообщение с помощью чисел 7 и 11. Приведённые ключи плохи для практического использования, так как число 77 легко раскладывается на простые множители (7 и 11).

Исходное сообщение m (текст) шифруется с помощью открытого ключа — числа n по следующей формуле:

$c = mI \bmod n$.

Благодаря использованию умножения по модулю скорость шифрования системы Рабина больше, чем скорость шифрования по методу RSA, даже если в последнем случае выбрать небольшое значение экспоненты.

Пример (продолжение). Пусть исходным текстом является m = 20. Тогда зашифрованным текстом будет: c = mI mod n = 20I mod 77 = 400 mod 77 = 15

Для расшифровки сообщения необходим закрытый ключ — числа p и q. Процесс расшифровки выглядит следующим образом:

- сначала, используя алгоритм Эвклида, из уравнения находят числа и ;
- далее, используя китайскую теорему об остатках, вычисляют четыре числа:

Одно из этих чисел является истинным открытым текстом m.

Пример (окончание). В результате расшифровки получаем: . Видим, что один из корней является исходным текстом m.

1. 11 Лекция № 14 (2 часа).

Тема: «Целостность данных и аутентификация сообщений»

1.1.1 Вопросы лекции:

1. Целостность данных.

2. Аутентификация сообщений.

1.1.2 Краткое содержание вопросов:

1. Целостность данных.

Целостность информации (также целостность данных) — термин в информатике (криптографии, теории телекоммуникаций, теории информационной безопасности), означающий, что данные не были изменены при выполнении какой-либо операции над ними, будь то передача, хранение или отображение.

В телекоммуникации целостность данных часто проверяют, используя хеш-сумму сообщения, вычисленную алгоритмом MAC (англ. message authentication code).

В криптографии и информационной безопасности целостность данных (в широком смысле) — это сохранение данных в том виде, в каком они были созданы. Примеры нарушений целостности данных:

- попытка злоумышленника изменить номер аккаунта в банковской транзакции, или попытка подделки документа;
- случайное изменение информации при передаче или при неисправной работе жёсткого диска;
- искажение фактов средствами массовой информации с целью манипуляции общественным мнением.

В теории баз данных целостность данных означает корректность данных и их непротиворечивость. Обычно она также включает целостность связей, которая исключает ошибки связей между первичным и вторичным ключом. Примеры нарушений целостности данных:

- существование записей-сирот (дочерних записей, не имеющих связи с родительскими записями);
- существование одинаковых первичных ключей.

Для проверки целостности данных в криптографии используются хеш-функции, например, MD5. Хэш-функция преобразует последовательность байт произвольного размера в последовательность байт фиксированного размера (число). Если данные изменятся, то и число, генерируемое хеш-функцией, тоже изменится.

Целостность данных — свойство, при выполнении которого данные сохраняют заранее определённый вид и качество.

2. Аутентификация сообщений.

Аутентификация (установление подлинности) – проверка принадлежности субъекту доступа предъявленного им идентификатора и подтверждение его подлинности. Другими словами, аутентификация заключается в проверке: является ли подключающийся субъект тем, за кого он себя выдает.

Парольные методы аутентификации по степени изменяемости паролей делятся на:

- методы, использующие постоянные (многократно используемые) пароли;
- методы, использующие одноразовые (динамично изменяющиеся) пароли.

Общая процедура идентификации и аутентификации пользователя при его доступе в защищенную информационную систему заключается в следующем.

Пользователь предоставляет системе свой личный идентификатор (например, вводит пароль или предоставляет палец для сканирования отпечатка). Далее система сравнивает полученный идентификатор со всеми хранящимися в ее базе идентификаторами. Если результат сравнения успешный, то пользователь получает доступ к системе в рамках установленных полномочий. В случае отрицательного результата система сообщает об ошибке и предлагает повторно ввести идентификатор. В тех случаях, когда пользователь превышает лимит возможных повторов ввода информации (ограничение на количество повторов является обязательным условием для защищенных систем) система временно блокируется и выдается сообщение о несанкционированных действиях (причем, может быть, и незаметно для пользователя).

Если в процессе аутентификации подлинность субъекта установлена, то система защиты информации должна определить его полномочия (совокупность прав). Это необходимо для последующего контроля и разграничения доступа к ресурсам.

В целом аутентификация по уровню информационной безопасности делится на три категории:

- Статическая аутентификация.
- Устойчивая аутентификация.
- Постоянная аутентификация.

В аутентификации сообщений могут быть выделены два основных уровня:

1. на низшем уровне должна выполняться некоторая функция, порождающая аутентификатор (удостоверение, используемое для подтверждения подлинности субъекта).
2. аутентификатор затем используется как примитив в протоколе аутентификации высшего уровня, дающем получателю сообщения возможность проверить достоверность сообщения.

1. 12 Лекция № 15 (2 часа).

Тема: «Хэш-функции.»

1.1.1 Вопросы лекции:

1. Хэш-функции (MD4, SHA).

2. Алгоритмы ЭЦП: RSA, Эль Гамаля, Шнорра, Нибберга-Руппеля.

1.1.2 Краткое содержание вопросов:

1. Хэш-функции (MD4, SHA).

Хэш-функцией называется алгоритм, конвертирующий строку произвольной длины (сообщение) в битовую строку фиксированной длины, называемой хэш-кодом, проверочной суммой или цифровым отпечатком.

CRC32 (Cyclic redundancy check - Циклический избыточный код) простая хэш функция разработанная для защиты данных от случайных изменений в компьютерных устройствах, таких как сетевые карты и жёсткие диски. CRC32 определяется международным стандартом CRC32-IEEE 802.3 Алгоритм очень быстр и, несмотря на полную криптографическую незащищённость, широко используется благодаря простоте реализации и скорости. 32-битный хэш-код обычно представляется шестнадцатеричным числом из 8 символов.

MD4 алгоритм хэширования, разработанный Рональдом Л. Ривестом из RSA Data Security, Inc. В настоящее время считается ненадёжным. Это быстрый алгоритм (на 32-битных процессорах) и его используют при вычислении хэшей в peer-to-peer сети EDonkey 2000. Алгоритм описан в RFC 1320. Хэш-код представляет шестнадцатеричное число из 32 символов.

MD5 ещё один алгоритм хэширования, разработанный Рональдом Л. Ривестом из RSA Data Security, Inc. Представляет улучшенную версию MD4. Алгоритм описан в RFC 1321. В течении многих лет MD5 был стандартом интернет, но сейчас считается сломанным. Хэш-код представляет шестнадцатеричное число из 32 символов.

SHA1 (Secure Hash Algorithm 1) алгоритм хэширования, разработанный NSA в 1993. Описан в RFC 3174. Он примерно в 2-3 раза медленнее алгоритма MD5. Хэш-код представляет шестнадцатеричное число длины 40.

Tiger современная хэш-функция изобретённая Россом Андерсом и Эли Бихамом. Была специально придумана такой, чтобы быстро вычисляться на 64-битных процессорах. См. описание. Хэш-код представляет шестнадцатеричное число длины 48.

TTH (Tiger Tree Hash) хэш, вычисляющийся в древовидной форме с использованием алгоритма Tiger. См.описание. TTH используется в нескольких peer-to-peer сетях: Direct Connect, Gnutella, Gnutella2, а также в таких программах как DC++, Phex и Shareaza. Хэш-код представляет base32-закодированную строку длины 39.

BTIH (BitTorrent InfoHash) используется в p2p сети BitTorrent. Хэш-сумма зависит не только от данных, но и от имени файла и даже от программы вычисляющей хэш. RHash использует тот же метод, что и uTorrent. Хэш является строкой из 40 шестнадцатеричных чисел.

2. Алгоритмы ЭЦП: RSA, Эль Гамаля, Шнорра, Нибберга-Руппеля.

Электромнная помдпись (ЭП), Электромнная цифровая помдпись (ЭЦП) — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата

ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

Существует несколько схем построения цифровой подписи:

- На основе алгоритмов симметричного шифрования. Данная схема предусматривает наличие в системе третьего лица — арбитра, пользующегося доверием обеих сторон. Авторизацией документа является сам факт зашифрования его секретным ключом и передача его арбитру.
- На основе алгоритмов асимметричного шифрования. На данный момент такие схемы ЭП наиболее распространены и находят широкое применение.

Кроме этого, существуют другие разновидности цифровых подписей (групповая подпись, неоспоримая подпись, доверенная подпись), которые являются модификациями описанных выше схем. Их появление обусловлено разнообразием задач, решаемых с помощью ЭП.

Поскольку подписываемые документы — переменного (и как правило достаточно большого) объёма, в схемах ЭП зачастую подпись ставится не на сам документ, а на его хэш. Для вычисления хэша используются криптографические хэш-функции, что гарантирует выявление изменений документа при проверке подписи. Хэш-функции не являются частью алгоритма ЭП, поэтому в схеме может быть использована любая надёжная хэш-функция.

Использование хэш-функций даёт следующие преимущества:

- Вычислительная сложность. Обычно хэш цифрового документа делается во много раз меньшего объёма, чем объём исходного документа, и алгоритмы вычисления хэша являются более быстрыми, чем алгоритмы ЭП. Поэтому формировать хэш документа и подписывать его получается намного быстрее, чем подписывать сам документ.
- Совместимость. Большинство алгоритмов оперирует со строками бит данных, но некоторые используют другие представления. Хэш-функцию можно использовать для преобразования произвольного входного текста в подходящий формат.
- Целостность. Без использования хэш-функции большой электронный документ в некоторых схемах нужно разделять на достаточно малые блоки для применения ЭП. При верификации невозможно определить, все ли блоки получены и в правильном ли они порядке.

Использование хэш-функции не обязательно при электронной подписи, а сама функция не является частью алгоритма ЭП, поэтому хэш-функция может использоваться любая или не использоваться вообще.

1. 13 Лекция №16 (2 часа).

Тема: «Реализация схем электронной цифровой подписи (на основе алгоритмов RSA, El Gamal, Шнорра)»

1.13.1 Вопросы лекции:

1. Электронная подпись на основе алгоритмов RSA.
2. Электронная подпись El Gamal, Шнорра.

1.13.2 Краткое содержание вопросов:

1. Электронная подпись на основе алгоритмов RSA.

Математическая схема электронной цифровой подписи по алгоритму RSA была предложена в 1977 году сотрудниками Массачусетского технологического института

США. Данная система цифровой подписи стала первым практическим решением задачи подписи электронных документов при помощи криптосистем с открытым ключом. Процедура вычисления цифровой подписи в данной системе использует криптографическое преобразование по алгоритму RSA.

В соответствии с данной системой цифровой подписи, субъект, желающий пересылать подписанные им документы, должен сформировать два ключа алгоритма RSA: открытый и закрытый.

Пару значений (KO, r), которая является открытым ключом подписи, отправитель передаёт всем возможным получателям его сообщений. Именно эти значения будут использоваться для проверки подлинности и принадлежности отправителю полученных от него сообщений.

Значение KC сохраняется отправителем в секрете. Данное значение вместе с модулем r является секретным ключом, который будет использоваться отправителем для постановки подписей под своими сообщениями.

Схема использования алгоритма цифровой подписи на базе RSA для обмена двух абонентов подписанными сообщениями показана на рис. 13.1

рис.13.1

2. Электронная подпись El Gamal, Шнорра.

Название EGSA происходит от слов EI GamaI Signature Algorithm (алгоритм цифровой подписи Эль Гамаля). Идея EGSA основана на том, что для обоснования практической невозможности фальсификации цифровой подписи может быть использована более сложная вычислительная задача, чем разложение на множители большого целого числа,- задача дискретного логарифмирования. Кроме того, Эль Гамалю удалось избежать явной слабости алгоритма цифровой подписи RSA, связанной с возможностью подделки цифровой подписи под некоторыми сообщениями без определения секретного ключа.

Рассмотрим подробнее алгоритм цифровой подписи Эль Гамаля. Для того чтобы генерировать пару ключей (открытый ключ - секретный ключ), сначала выбирают некоторое большое простое целое число P и большое целое число G, причем G < P. Отправитель и получатель подписанного документа используют при вычислениях одинаковые большие целые числа P (~10308 или ~21024) и G (~10154 или ~2512), которые не являются секретными.

Отправитель выбирает случайное целое число X, 1 < X J (P-1), и вычисляет

Y =GX mod P.

Число Y является открытым ключом, используемым для проверки подписи отправителя. Число Y открыто передается всем потенциальным получателям документов.

Число X является секретным ключом отправителя для подписывания документов и должно храниться в секрете.

Для того чтобы подписать сообщение M, сначала отправитель хэширует его с помощью хэш-функции h() в целое число m:

m = h(M), 1<m<(P-1),

и генерирует случайное целое число K, 1 < K < (P -1), такое, что K и (P-1) являются взаимно простыми. Затем отправитель вычисляет целое число a:

a = GK mod P

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа X целое число b из уравнения

m = X * a + K * b (mod (P-1)).

Пара чисел (a,b) образует цифровую подпись S:

S = (a, b),

проставляемую под документом M.

Тройка чисел (M, a, b) передается получателю, в то время как пара чисел (X, K).держится в секрете.

После приема подписанного сообщения (M, a, b) получатель должен проверить, соответствует ли подпись S = (a, b)

сообщению M. Для этого получатель сначала вычисляет по принятому сообщению M число m = h(M),

т.е. хэширует принятое сообщение M.

Затем получатель вычисляет значение A = Ya ab (mod P)

и признает сообщение M подлинным, если, и только если

A = Gm (mod P).

Иначе говоря, получатель проверяет справедливость соотношения

Ya ab (mod P) = Gm (mod P).

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись S=(a, b) под документом M получена с помощью именно того секретного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправителем сообщения M был обладатель именно данного секретного ключа X, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ M.

Следует отметить, что схема Эль Гамаля является характерным примером подхода, который допускает пересылку сообщения M в открытой форме вместе с присоединенным аутентификатором (a, b). В таких случаях процедура установления подлинности принятого сообщения состоит в проверке соответствия аутентификатора сообщению.

Схема цифровой подписи Эль Гамаля имеет ряд преимуществ по сравнению со схемой цифровой подписи RSA:

1. При заданном уровне стойкости алгоритма цифровой подписи целые числа, участвующие в вычислениях, имеют запись на 25% короче, что уменьшает сложность вычислений почти в два раза и позволяет заметно сократить объем используемой памяти.

2. При выборе модуля P достаточно проверить, что это число является простым и что у числа (P-1) имеется большой простой множитель (т.е. всего два достаточно просто проверяемых условия).

3. Процедура формирования подписи по схеме Эль Гамаля не позволяет вычислять цифровые подписи под новыми сообщениями без знания секретного ключа (как в RSA).

Однако алгоритм цифровой подписи Эль Гамаля имеет и некоторые. недостатки по сравнению со схемой подписи RSA. В частности, длина цифровой подписи получается в 1,5 раза больше, что, в свою очередь, увеличивает время ее вычисления.

1. 14 Лекция № 17 (2часа).

Тема: «Криптографические протоколы.»

1.14.1 Вопросы лекции:

1.Характеристика протоколов идентификации и аутентификации, идентификация на основе пароля.

2.Взаимная проверка подлинности пользователей.

3.Идентификация с нулевой передачей знаний.

4.Схемы обязательств.

5.Системы электронного голосования.

6.Цифровые сертификаты: системы перераспределения доверия, неявные сертификаты.

1.14.2 Краткое содержание вопросов:

1. Характеристика протоколов идентификации и аутентификации, идентификация на основе пароля.

Идентификация - присвоение пользователям идентификаторов (уникальных имен или меток) под которыми система "знает" пользователя. Кроме идентификации пользователей, может проводиться идентификация групп пользователей, ресурсов ИС и т.д.Идентификация нужна и для других системных задач, например, для ведения журналов событий. В большинстве случаев идентификация сопровождается аутентификацией. Аутентификация - установление подлинности - проверка принадлежности пользователю предъявленного им идентификатора. Например, в начале сеанса работы в ИС пользователь вводит имя и пароль. На основании этих данных система проводит идентификацию (по имени пользователя) и аутентификацию (сопоставляя имя пользователя и введенный пароль).

Система идентификации и аутентификации является одним из ключевых элементов инфраструктуры защиты от несанкционированного доступа (НСД) любой информационной системы. В соответствии с рассмотренной ранее моделью многоуровневой защиты, аутентификация пользователя компьютера относится к уровню защиты узлов.

Обычно выделяют 3 группы методов аутентификации.

1. Аутентификация по наличию у пользователя уникального объекта заданного типа. Иногда этот класс методов аутентификации называют по-английски "I have" ("у меня есть"). В качестве примера можно привести аутентификацию с помощью смарт-карт или электронных USB-ключей.
2. Аутентификация, основанная на том, что пользователю известна некоторая конфиденциальная информация - "I know" ("я знаю"). Например, аутентификация по паролю. Более подробно парольные системы рассматриваются далее в этом разделе.
3. Аутентификация пользователя по его собственным уникальным характеристикам - "I am" ("я есть"). Эти методы также называются биометрическими.

Нередко используются комбинированные схемы аутентификации, объединяющие методы разных классов. Например, двухфакторная аутентификация - пользователь предъявляет системе смарт-карту и вводит пин-код для ее активации.

Наиболее распространенными на данный момент являются парольные системы аутентификации. У пользователя есть идентификатор и пароль, т.е. секретная информация, известная только пользователю (и возможно - системе), которая используется для прохождения аутентификации.

В зависимости от реализации системы, пароль может быть одноразовым или многоразовым. Операционные системы, как правило, проводят аутентификацию с использованием многоразовых паролей. Совокупность идентификатора, пароля и,

возможно, дополнительной информации, служащей для описания пользователя составляют учетную запись пользователя.

Если нарушитель узнал пароль легального пользователя, то он может, например, войти в систему под его учетной записью и получить доступ к конфиденциальным данным. Поэтому безопасности паролей следует уделять особой внимание.

Как отмечалось при рассмотрении стандарта ISO 17799, рекомендуется, чтобы пользователи системы подписывали документ о сохранении конфиденциальности пароля. Но нарушитель также может попытаться подобрать пароль, угадать его, перехватить и т.д. Рассмотрим некоторые рекомендации по администрированию парольной системы, позволяющие снизить вероятность реализации подобных угроз.

1. Задание минимальной длины используемых в системе паролей. Это усложняет атаку путем подбора паролей. Как правило, рекомендуют устанавливать минимальную длину в 6-8 символов.
2. Установка требования использовать в пароле разные группы символов - большие и маленькие буквы, цифры, специальные символы. Это также усложняет подбор.
3. Периодическая проверка администраторами безопасности качества используемых паролей путем имитации атак, таких как подбор паролей "по словарю" (т.е. проверка на использование в качестве пароля слов естественного языка и простых комбинаций символов, таких как "1234").
4. Установка максимального и минимального сроков жизни пароля, использование механизма принудительной смены старых паролей.
5. Ограничение числа неудачных попыток ввода пароля (блокирование учетной записи после заданного числа неудачных попыток войти в систему).
6. Ведение журнала истории паролей, чтобы пользователи, после принудительной смены пароля, не могли вновь выбрать себе старый, возможно скомпрометированный пароль.

2.Взаимная проверка подлинности пользователей.

Обычно стороны, вступающие в информационный обмен, нуждаются во взаимной проверке подлинности (аутентификации) друг друга. Этот процесс взаимной аутентификации выполняют в начале сеанса связи.

Для проверки подлинности применяют следующие способы:

- механизм запроса-ответа;
- механизм отметки времени ("временной штемпель").

Механизм запроса-ответа состоит в следующем. Если пользователь. А хочет быть уверенным, что сообщения, получаемые им от пользователя В, не являются ложными, он включает в посылаемое для В сообщение непредсказуемый элемент - запрос X (например, некоторое случайное число). При ответе пользователь В допжен выполнить над этим элементом некоторую операцию (например, вычислить некоторую функцию f(X)). Это невозможно осуществить заранее, так как пользователю В неизвестно, какое случайное число X придет в запросе. Получив ответ с результатом действий В, пользователь А может быть уверен, что В - подпинный. Недостаток этого метода - возможность установления закономерности между запросом и ответом.

Механизм отметки времени подразумевает регистрацию времени для каждого сообщения. В этом случае каждый попьзователь сети может определить, насколько

"устарело" пришедшее сообщение, и решить не принимать его, поскольку оно может быть ложным.

В обоих случаях для защиты механизма контроля следует применять шифрование, чтобы быть уверенным, что ответ послан не злоумышленником.

При использовании отметок времени возникает проблема допустимого временного интервала задержки для подтверждения подлинности сеанса. Ведь сообщение с "временным штемпелем" в принципе не может быть передано мгновенно. Кроме того, компьютерные часы получателя и отправителя не могут быть абсолютно синхронизированы. Какое запаздывание "штемпеля" является подозрительным?

Для взаимной проверки подлинности обычно используют процедуру рукопожатия. Эта процедура базируется на указанных выше механизмах контроля и заключается во взаимной проверке ключей, используемых сторонами. Иначе говоря, стороны признают друг друга законными партнерами, если докажут друг другу, что обладают правильными ключами. Процедуру рукопожатия обычно применяют в компьютерных сетях при организации сеанса связи между пользователями, пользоватепем и хост-компьютером, между хост-компьютерами и т.д. Рассмотрим в качестве примера процедуру рукопожатия для двух пользователей А и В. (Это допущение не влияет на общность рассмотрения. Такая же процедура используется, когда вступающие в связь стороны не являются пользователями). Пусть применяется симметричная криптосистема. Пользователи А и В разделяют один и тот же секретный ключ КАВ.

3.Идентификация с нулевой передачей знаний.

Широкое распространение смарт-карт (интеллектуальных карт) для разнообразных коммерческих, гражданских и военных применений (кредитные карты, карты социального страхования, карты доступа в охраняемые помещения, компьютерные пароли и ключи и т.д.) потребовало обеспечение безопасности идентификации таких карт и их владельцев. Во многих приложениях главная проблема заключается в том, чтобы при предъявлении смарт-карты оперативно обнаружить обман и отказать обманщику в допуске, ответе и обслуживании.

Для безопасного использования смарт-карт разработаны протоколы идентификации с нулевой передачей знаний. Секретный ключ владельца карты становится неотъемлемым признаком его личности. Доказательство знания этого секретного ключа с нулевой передачей этого знания служит доказательством подлинности личности владельца карты.

Схему идентификации с нулевой передачей знаний предложили в 1986 г. У.Фейге, А.Фиат и А.Шамир. Она является наиболее известным доказательством идентичности с нулевой передачей конфиденциальной информации.

Рассмотрим сначала упрощенный вариант схемы идентификации с нулевой передачей знаний для более четкого выявления ее основной концепции.

Но прежде всего определимся с терминологией.

Пусть a, b, d, g O Z (множество целых чисел), n O N (множество натуральных чисел, т.е. положительных целых чисел).

Определение 1. Число a сравнимо с b по модулю n, если a и b при делении на n дают одинаковые остатки, т.е. $a \bmod n = b \bmod n$.

Принятое обозначение: $a \equiv b \pmod n$.

Определение 2. Число d называют обратным к a по модулю n, если произведение a*d при делении на n дает в остатке 1, т.е. $a*d \bmod n = 1$.

Принятое обозначение: $b = a^{-1} \pmod n$.

Примите к сведению, что целое число $a^{-1} \pmod n$ существует тогда и только тогда, когда a является взаимно простым с n, т.е. имеет с модулем n наибольший общий делитель, равный единице.

Тех, кому интересно, почему это действительно так, отсылаю к расширенному алгоритму Евклида. В Интернете вы найдете не один сайт, посвященный этой теме.

Определение 3. Число g называют квадратным корнем из a по модулю n, если произведение g*g при делении на n дает в остатке a, т.е. g*g mod n = a .

Принятое обозначение: $g = \sqrt{a} \pmod n$.

4. Схемы обязательств.

В криптографии, схема обязательств или битовая схема обязательств (англ. Commitment scheme) — это метод, позволяющий пользователю подтверждать какое-либо значение, которое не разглашается, то есть в случае разглашения этого значения благодаря этой схеме будет известно, что пользователь знал его на момент выдачи обязательства и что оно не изменилось. Работу данной схемы можно представить как две стадии:

- «Commit» — посылку закрытой на ключ коробки (обязательство),
- «Reveal» — более поздняя отправка ключа от коробки (значение).

Одна из популярных реализаций схемы: Pedersen Commitment Scheme (основана на задаче дискретного логарифмирования)

5.Системы электронного голосования.

Электронное голосование — термин, определяющий различные виды голосования, охватывающий как электронные средства голосования (электронная демократия), так и технические электронные средства подсчёта голосов. Разновидностью электронного голосования являются Интернет-выборы.

Технологии электронного голосования могут включать в себя перфокарты, системы оптического сканирования и специализированные терминалы для голосования. Они также могут включать передачу избирательных бюллетеней и голосов по телефону, частным компьютерным сетям или через Интернет.

Технология электронного голосования позволяет ускорить процесс подсчёта голосов, а также упростить голосование людям с ограниченными возможностями. Но в настоящее время ведутся споры о том, что электронное голосование может быть подвержено нарушениям (большим, чем при традиционных системах голосования).

Устройства подсчёта голосов

Системы электронного подсчёта голосов применяются на выборах с 1960-х годов, с тех пор, как появились перфокарты.

Более новая система оптического сканирования может считывать с бюллетеня отметку, поставленную избирателем.

Системы прямой записи голосов, накапливающие голоса на одном устройстве, используются повсеместно в Бразилии, также достаточно широко распространены вИндии, Нидерландах, Венесуэле и США.

Системы Интернет-голосования завоевали популярность и используются в правительственных выборах и референдумах в Великобритании, Эстонии и Швейцарии, а также муниципальных выборах в Канаде и партийных выборах в США и Франции.

Устройства заполнения

Существуют системы, включающие в себя и устройство заполнения электронного бюллетеня (сенсорный экран, либо сканер штрих-кода). Также они, зачастую, оснащены дополнительным вспомогательным устройством для распечатки бумажной копии бюллетеня либо квитанции о голосовании. Хранение и подсчёт голосов при этом происходит на отдельном устройстве.

6.Цифровые сертификаты: системы перераспределения доверия, неявные сертификаты.

Цифровой сертификат — выпущенный удостоверяющим центром электронный или печатный документ, подтверждающий принадлежность владельцу открытого ключа или каких-либо атрибутов.

Сертификат открытого ключа

Сертификат открытого ключа удостоверяет принадлежность открытого ключа некоторому субъекту, например, пользователю. Сертификат открытого ключа содержит имя субъекта, открытый ключ, имя удостоверяющего центра, политику использования соответствующего удостоверяемому открытому ключу закрытого ключа и другие параметры, заверенные подписью удостоверяющего центра.

Сертификат открытого ключа используется для идентификации субъекта и уточнения операций, которые субъекту разрешается совершать с использованием закрытого ключа, соответствующего открытому ключу, удостоверяемому данным сертификатом.

Формат сертификата открытого ключа X.509 v3 описан в RFC 5280

Сертификат атрибутов

Структура сертификата атрибутов аналогична структуре сертификата открытого ключа. Отличие же заключается в том, что сертификат атрибутов удостоверяет не открытый ключ субъекта, а какие-либо его атрибуты — принадлежность к какой-либо группе, роль, полномочия и т. п. Сертификат атрибутов применяется дляавторизации субъекта. Формат сертификата атрибутов описан в RFC 5755.

Классификация сертификатов

VeriSign предложила следующую концепцию классификации цифровых сертификатов :

- Class 1 — индивидуальные, для идентификации электронной почты;
- Class 2 — для организаций;
- Class 3 — для серверов и программного обеспечения;
- Class 4 — для онлайн-бизнеса и транзакций между компаниями;
- Class 5 — для частных компаний или правительственной безопасности.

1. 15 Лекция № 18 (2 часа).

Тема: «Тесты на простоту и факторизация»

1.15.1 Вопросы лекции:

1.Тесты на простоту: пробное деление, тест Ферма, тест Миллера-Рабина.

2.Алгоритмы факторизации: пробное деление, гладкие числа, (P-1)-метод Полларда, разность квадратов, современные методы факторизации.

1.15.2 Краткое содержание вопросов:

1. Тесты на простоту: пробное деление, тест Ферма, тест Миллера-Рабина.

Вопрос определения того, является ли натуральное число простым, известен как проблема простоты.

Тестом простоты (или проверкой простоты) называется алгоритм, который, приняв на входе число , позволяет либо не подтвердить предположение о составностичисла, либо точно утверждать его простоту. Во втором случае он называется истинным тестом простоты. Таким образом, тест простоты представляет собой толькогипотезу о том, что если алгоритм не подтвердил предположение о составности числа , то это число может являться простым с определенной вероятностью. Это определение подразумевает меньшую уверенность в соответствии результата проверки истинному положению вещей, нежели истинное испытание на простоту, которое дает математически подтвержденный результат.

Тест простоты Ферма в теории чисел — это тест простоты натурального числа n, основанный на малой теореме Ферма.

Если n — простое число, то оно удовлетворяет сравнению для любого a, которое не делится на n.

Выполнение сравнения является необходимым, но не достаточным признаком простоты числа. То есть, если найдётся хотя бы одно a, для которого , то число n — составное; в противном случае ничего сказать нельзя, хотя шансы на то, что число является простым, увеличиваются. Если для составного числа n выполняется сравнение , то число n называют псевдопростым по основанию a . При проверке числа на простоту тестом Ферма выбирают несколько чисел a. Чем больше количество a, для которых , тем больше шансы, что число n простое. Однако существуют составные числа, для которых сравнение выполняется для всех a, взаимно простых с n — это числа Кармайкла. Чисел Кармайкла — бесконечное множество, наименьшее число Кармайкла — 561. Тем не менее, тест Ферма довольно эффективен для обнаружения составных чисел.

Тест Миллера — Рабина — вероятностный полиномиальный тест простоты. Тест Миллера — Рабина, наряду с тестом Ферма и тестом Соловея — Штрассена, позволяет эффективно определить, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера — Рабина часто используется в криптографии для получения больших случайных простых чисел.

2. Алгоритмы факторизации: пробное деление, гладкие числа, (P-1)-метод Полларда, разность квадратов, современные методы факторизации.

Факторизамцией натурального числа называется его разложение в произведение простых множителей. Существование и единственность (с точностью до порядка следования множителей) такого разложения следует из основной теоремы арифметики.

В отличие от задачи распознавания простоты числа, факторизация предположительно является вычислительно сложной задачей. В настоящее время неизвестно, существует ли эффективный не квантовый алгоритмфакторизации целых чисел. Однако доказательства того, что не существует решения этой задачи за полиномиальное время также нет.

Предположение о том, что для больших чисел задача факторизации является вычислительно сложной лежит в основе широко используемых алгоритмов (например, RSA). Множество областей математики и информатики находят применение в решении этой задачи. Среди них: эллиптические кривые, алгебраическая теория чисел иквантовые вычисления.

Одним из ключевых моментов в развитии факторизации целых чисел было создание алгоритма RSA, что возобновило интерес ученых в данном направлении, так как имело практическое применение в области шифрования. Этот алгоритм был предложен в 1977 году тремя учеными Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом из Массачусетского Технологического Института и назван по первым буквам фамилий авторов методом RSA. Он основан на идее криптографии с открытым ключом и для взлома системы необходимо выполнить разложение числа на простые сомножители. На момент публикации алгоритма RSA были известны методы, которые позволяли факторизовать числа, состоящие не более чем из 25—30 цифр, а наиболее известным и применяемым все ещё оставался метод Ферма. Метод RSA позволяет факторизовывать числа из 100 и более десятичных знаков. Создатели, в свою очередь, пообещали за факторизацию числа из 129 десятичных знаков символические сто долларов США[2].

На популярность задачи факторизации также повлияла публикация в 1977 году в журнале Scientific American Мартина Гарднер «Новый алгоритм шифрования, для взлома которого потребуется миллионы лет». Столь громкое название было воспринято в качестве вызова всему математическому сообществу. В результате этой гонки было предложено несколько новых и нестандартных идей факторизации[2].

Эпопея с разложением 129-значного числа завершилась в 1994 году, когда коллектив под руководством А. Ленстры, используя 1600 компьютеров, подготовил за 220 дней систему линейных уравнений, содержавшую более полумиллиона неизвестных. Решение этой системы суперкомпьютером заняло два дня. Несмотря на то, что в то время уже были известны методы решета числового поля, данный результат был получен с помощью алгоритма квадратичного решета.

1. 16 Лекция № 19 (2 часа).

Тема: «Надежность криптосистем. Элементы криптоанализа.»

1.16.1 Вопросы лекции:

1.Виды атак: Атака Винера на RSA, атаки на RSA основанные на решетках, атака Хостада, атака Франклина-Рейтера, частичное раскрытие ключа.

2.Стойкость актуальных алгоритмов шифрования.

3.Доказуемая стойкость со случайным оракулом.

4.Доказуемая стойкость без случайного оракула

1.16.2 Краткое содержание вопросов:

1. Виды атак: Атака Винера на RSA, атаки на RSA основанные на решетках, атака Хостада, атака Франклина-Рейтера, частичное раскрытие ключа.

До настоящего момента не было обнаружено никаких разрушительных атак RSA. Несколько атак были предсказаны. Они основаны на слабом исходном тексте, слабом выборе параметра или несоответствующей реализации. Рисунок 16.1 показывает категории потенциальных атак.

Атака Винера на RSA

В некоторых приложениях требуется ускорить процесс расшифровывания в алгоритме RSA. Поэтому выбирается небольшая расшифровывающая экспонента. В случае когда расшифровывающая экспонента можно определить за полиномиальное время с помощью атаки Винера, опирающейся на непрерывные дроби.

Поскольку НОД то подходящая дробь в разложении дроби в непрерывную. Таким образом, можно узнать расшифровывающую экспоненту, поочерёдно подставляя знаменатели подходящих дробей в выражение:

для некоторого случайного числа Получив равенство, найдём Общее число подходящих дробей, которое придётся проверить оценивается как

2.Стойкость актуальных алгоритмов шифрования.

Стойкость алгоритма шифрования.

Алгоритм шифрования считается стойким до тех пор, пока не будет доказано обратное. Таким образом, если алгоритм шифрования опубликован, существует более 5 лет, и для него не найдено серьезных уязвимостей, можно считать, что его стойкость подходит для задач защиты секретной информации.

Теоретическая и практическая стойкость.

В 1949 г. К.Э. Шеннон опубликовал статью "Теория связи в секретных системах". Шеннон рассматривал стойкость криптографических систем как Практическую и Теоритическую. Вывод по теоритической стойкости до сих пор остается пессимистическим: длина ключа должна быть равна длине открытого текста.

Поэтому Шеннон также рассмотрел вопрос и по практической стойкости криптографических систем. Надежна ли система, если злоумышленник обладает ограниченным временем и вычислительными ресурсами для анализа перехваченных сообщений ?

Обычно уязвимости находят в программах, которые шифруют данные по какому-либо алгоритму. В этом случае, программисты допускают ошибку в логике программы или в криптографическом протоколе, благодаря чему, изучив, как работает программа (на низком уровне), можно в итоге получить доступ к секретной информации.

Считается, что криптосистема раскрыта, если злоумышленник сможет вычислить секретный ключ, а также выполнить алгоритм преобразования, эквивалентный исходному криптоалгоритму. И чтобы этот алгоритм был выполним за реальное время.

В криптологии есть подраздел - криптоанализ, который изучает вопросы взлома или подделывания зашифрованных сообщений. Существует много способов и методов криптоанализа. Самый популярный - это метод прямого перебора всех возможных значений ключа шифрования (так называемым методом "грубой силы" или brute force). Суть данного метода состоит в переборе всех возможных значений ключа шифрования до тех пор, пока не будет подобран нужный ключ.

3.Доказуемая стойкость со случайным оракулом.

Расширенная математическая модель криптографического протокола, в которой все участники имеют доступ к оракулу, вычисляющему случайную функцию. При каждом новом запросе значение функции на данном аргументе выбирается случайным образом. При этом оракул запоминает пару (аргумент, значение) и при повторном запросе для этого аргумента, независимо от того, кто из участников его выдал, будет возвращено все то же запомненное значение.

Модель со случайным оракулом была введена для протоколов электронной подписи, где случайная функция заменила хэш-функцию. Впоследствии модель была обобщена на криптографические протоколы различных типов и в настоящее время в криптологии значительная часть результатов остойкости доказывается именно в этой модели.

4.Доказуемая стойкость без случайного оракула.

В модели случайного оракула (МСО).участникам предоставляется доступ при помощи оракула к некоторой случайной функции, называемой случайным оракулом (СО). Случайный оракул часто рассматривается как идеализированная хэш функция. Доказательства безопасности в модели случайного оракула часто помогают обосновать безопасность множества важных схем, в которых используются настоящие хэш функции. Редукции безопасности, применяемые в доказательствах безопасности , могут использовать многие свойства СО.

Одним из таких свойств является программируемость. Не вдаваясь в подробности, можно сказать, что случайный оракул может быть наполнен динамически выбираемыми возвращаемыми значениями, тогда при условии, что значения выбираются правильным образом (сбалансировано на определенной области значений), любой метод выбора этих значений считается правильным. Техника программирования выходных значений СО в редукции безопасности имеет ключевое значение в бессчетном множестве правильных результатов . Ни одна из известных функций стандартной модели не может предоставить возможность абсолютно случайной и адаптивной программируемости , которой обладает СО. Возникает вопрос : какой из данных результатов (если таковой вообще есть) мог бы быть получен без использования полной программируемости МСО?

В данной статье формально рассмотрены модели, в которых программируемость случайного оракула в редукциях безопасности ограничена, то есть предложена модель, являющаяся промежуточной между полностью программируемой и не программируемой. В статье предложены два разных, но при этом эквивалентных описания данной ограниченной формы программируемости. Они используются чтобы показать, что :

1. каждый может доказать (используя новый вариант техники Hsiao-Reyzin разделения на 2 оракула) невозможность предоставить ограниченно-программируемую редукцию безопасности в виде черного ящика для алгоритма цифровой подписи ХПОО [1]
2. схема Shoup’a TDP-KEM [15] доказуемо безопасна от атак на основе выбранного шифр текста при условии частичной программируемости, но ни одна редукция безопасности в виде чёрного ящика не работает, если программируемость запрещена.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

2.1 Лабораторная работа №1 (2 часа).

Тема: «Поточные системы шифрования (РСЛОС, RC4, Рона)»

2.1.1 Цель работы: Исследование систем РСЛОС, Рона, RC4

2.1.2 Задачи работы:

1. Изучить потоковые алгоритмы шифрования.

2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Вычислительная машина.

2.1.4 Описание (ход) работы:

1.Ознакомьтесь с теоретическими основами шифрования данных в учебном пособии «Криптография», а также в учебно-методическом пособии к выполнению лабораторного практикума по дисциплине «Криптография».

2. Получите вариант задания у преподавателя.

3. Напишите программу согласно варианту задания.

4. Отладьте разработанную программу и покажите результаты работы программы преподавателю.

5. Составьте отчет по лабораторной работе

2.2 Лабораторная работа № 2-3 (4 часа).

Тема: «Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)»

2.1.1 Цель работы: Исследование поточных систем шифрования.

2.1.2 Задачи работы:

1. Изучить потоковые алгоритмы шифрования.

2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:

1.Электронно-вычислительная машина.

2.1.4 Описание (ход) работы:

1.Ознакомьтесь с теоретическими основами шифрования данных в учебном пособии «Криптография», а также в учебно-методическом пособии к выполнению лабораторного практикума по дисциплине «Криптография».

2. Получите вариант задания у преподавателя.

3. Напишите программу согласно варианту задания.

4. Отладьте разработанную программу и покажите результаты работы программы преподавателю.

5. Составьте отчет по лабораторной работе

2.3 Лабораторная работа № 4 (2 часа).

Тема: «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)»

2.3.1 Цель работы: Исследование схем распределения ключей по схеме Диффи-Хеллмана.

2.3.2 Задачи работы:

1. Освоить алгоритм обмена ключами

2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Электронно-вычислительная машина.

2.3.4 Описание (ход) работы:

Задание .

Открытый элемент P задан в табл. 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы i, вычислить ключ обмена для участника с номером 35 – i по алгоритму Диффи – Хэллмана

Пример выполнения Задания3. Алгоритм Диффи-Хеллмана

Открытый элемент P задан в таблице 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы i, вычислить ключ обмена для участника с номером 35 – i по алгоритму Диффи-Хэллмана.

Вариант № 15, группа 2091 (№ 1)

Номер в списке группы i = 15

Номер группы k = 1

P = 37

Открытый элемент P = 37. Найти примитивный элемент поля. Секретный ключ каждого участника i = 15, вычислить ключ обмена для участника с номером 35-15 = 20.

GF(37) = <0, 1, 2, 3, …, 35, 36, 37>

Найдем примитивный элемент поля GF(37).

Требуется найти такое число, принадлежащее интервалу [2,37], которое при возведении в 37-ю степень по модулю 37 будет давать в результате единицу. Если же единица будет получена раньше, чем при возведении в 36-ю степень, результаты возведения в степень начнут повторяться, и через выбранный элемент не удастся представить все элементы поля. Исходя из таких соображений, получаем несложный алгоритм нахождения примитивных элементов поля GF(37).

Алгоритм нахождения примитивных элементов поля GF(37).

```
int _tmain(int argc, _TCHAR* argv[])
{
    long int mem, i, j, num, deg, modul, res;
    unsigned char mas[38];
```

```
	deg = 0; modul = 37; mem = 2;
	while (mem < 36)
	{
		num = mem; deg = 0;
		while (deg != 36)
		{
			res = 1; deg = 0;
			for (i = 0; i < 36; i++) mas[i] = 0;
			do
			{
				res = res*num; res = res % modul;
				deg++; mas[deg] = res;
			}
			while(res != 1);
			num++;
		}
		std::cout << num-1 << std::endl;
		for (i = 0; i < 36; i++)
		{
			for (j = 0; j < 36; j++)
			{
				if ((i != j) && (mas[j] == mas[i]))
					std::cout << "Wrong" << std::endl;
			}
			mem = num;
		}
	}
}
```

В результате выполнения программы получим следующие значения:

2, 5, 13, 15, 17, 18, 19, 20, 22, 24, 32, 35

Выберем значение $\alpha = 2$.

Секретный ключ участника A: $X_A = 15$.

Секретный ключ участника B: YB = 35-15 = 20.

Открытый ключ участника A:

Открытый ключ участника B:

Обменный ключ участника A:

Обменный ключ участника B:

Значения обменного ключа для A и B совпадают.

Обменный ключ: K = 26

2.4 Лабораторная работа №5-6 (4 часа).

Тема: «Асимметричные криптосистемы (RSA, El Gamal, Рабина)»

2.4.1 Цель работы: Исследование формирование асимметричных криптосистем.

2.4.2 Задачи работы:

1. Исследование формирование асимметричных криптосистем.
2. Исследование формирование асимметричных криптосистем RSA.
3. Формирование асимметричных криптосистем Рабина.
4. Формирование асимметричных криптосистем El Gamal.

2.4.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Электронно-вычислительная машина.

2.4.4 Описание (ход) работы:

1. Основные принципы асимметричной криптографии Долгое время шифрование как способ преобразования сообщения в форму, недоступную для восприятия неавторизованным пользователям, существовало только в форме симметричной криптографии, когда и отправитель, и получатель должны знать секретный ключ, используемый для шифрации/дешифрации сооб- щений. Симметричное шифрование имеет недостатки, которые ограничивают воз- можности его применения в ряде конкретных случаев. В частности, зачастую невозможно организовать секретный канал для обмена ключами шифрования между участниками взаимодействия. Еще одним недостатком симметричных шифров является необходимость хранения большого количества ключей. Для того чтобы в вычислительной сети могли конфиденциально попарно взаимодействовать N участников, необходимо наличие в системе N*(N-1)/2 ключей. Эти недостатки можно устранить, используя алгоритмы асимметричного шифрования. Например, для асимметричной системы достаточно иметь 2*N пар открытый/закрытый ключ, чтобы можно было организовать секретный канал между

каждой парой участников. Асимметричная система шифрования работает по схеме, представленной на рис. 1. Отличительной особенностью асимметричных алгоритмов является наличие пары ключей шифрования: открытого (публичного) k от, который передается второй стороне по незащищенному каналу связи и поэтому может быть известен криптоаналитику, а также закрытого (частного) kзак, который известен лишь од- ному человеку (получателю сообщения) и держится в секете [1]. Пара ключей обладает тем свойством, что сообщение, зашифрованное на одном из ключей, может быть расшифровано только на другом ключе. Фактически это означает, что секретным каналом передачи информации на схеме рис. 1 является направление “A-B”, поскольку сообщение, зашифрованное на открытом ключе отправителем, может дешифровать своим закрытым ключом только получатель. Если необходимо организовать двунаправленный безопасный обмен сообщениями, необходимо использовать две пары ключей и пользователь B должен шифровать информацию с использованием открытого ключа пользователя A. Асимметричная система решает указанные выше проблемы симметричного шифрования – здесь нет необходимости передавать секретный ключ на противоположную сторону, публичный ключ можно передавать по открытому каналу связи. Единственное требование к каналу распространению открытого ключа – он должен быть аутентичным, т.е. всякий, получивший этот ключ, должен иметь возможность убедиться в его принадлежности лицу, заявленному как владелец соответствующей ключевой пары. На практике это требование реализуется путем сертификации открытого ключа – третья сторон, которой доверяют обе стороны взаимодействия, заверяет открытый ключ и выдает на него сертификат, подписанный электронной цифровой подписью. Открытый ключ в этом случае распространяется вместе с сертификатом, что дает возможность всегда удостоверится в его принадлежности. К асимметричным относятся такие алгоритмы шифрования как RSA, El Gemal, Рабина, Месси-Омуры.

2. Исследование формирование асимметричных криптосистем RSA.

Определение открытого «e» и секретного «d» ключей

Выбор двух взаимно простых больших чисел p и q

Определение их произведения: $n = p * q$

Определение функции Эйлера: $\varphi(n) = (p-1)(q-1)$

Выбор открытого ключа e с учётом условий:

$1 < e \leq \varphi(n)$, НОД $(e, \varphi(n)) = 1$

Определение секретного ключа d, удовлетворяющего условию

$e * d \equiv 1 \pmod{\varphi(n)}$, где $d < n$

3.2. Алгоритм шифрования сообщения M (действия отправителя)

Разбивает исходный текст сообщения на блоки M1, M2, ..., Mп

(Mi = 0, l, 2, ..., n)

Шифрует текст сообщения в виде последовательности блоков:

Ci = M (mod n)

Отправляет получателю криптограмму: C1, C2, ..., Cп

Получатель расшифровывает криптограмму с помощью секретного ключа d по формуле: Mi = C (mod п)

Процедуру шифрования данных рассмотрим на следующем примере (для простоты и удобства расчётов в данном примере использованы числа малой разрядности):

Выбираем два простых числа p и q, p = 3, q = 11;

Определяем их произведение (модуль) n = p*q = 33;

Вычисляем значение функции Эйлера φ (п)= (p-1)(q-1)

φ (п) = 2*10 = 20

Выбираем случайным образом открытый ключ с учётом выполнения условий $1 < e \leq \varphi$ (п), НОД (e, φ (п)) = 1, e = 7;

Вычисляем значение секретного ключа d, удовлетворяющего условию

e * d ≡ 1 (mod φ (п)), 7 * d ≡ 1 (mod 20); d = 3;

Отправляем получателю пару чисел (п =33, e = 7);

Представляем шифруемое сообщение M как последовательность целых чисел 312.

Разбиваем исходное сообщение на блоки M1 = 3, M2 = 1, M3 = 2;

Шифруем текст сообщения, представленный в виде последователь-ности блоков: Ci = M (mod n)

C1 = 37(mod 33) = 2187 (mod 33) = 9,

C2 = 17(mod 33) = 1 (mod 33) = 1,

C3 = 27(mod 33) = 128 (mod 33) = 29.

Отправляем криптограмму C1 = 9, C2 = 1, C3 = 29.

Получатель расшифровывает криптограмму с помощью секретного ключа d по формуле: Mi = C (mod п)

M1 = 93(mod 33) = 729 (mod 33) = 3,

M2 = 13(mod 33) = 1 (mod 33) = 1,

M3 = 293(mod 33) = 24389 (mod 33) = 2.

Полученная последовательность чисел 312 представляет собой исходное сообщение M.

3. Формирование асимметричных криптосистем Рабина.

4. Формирование асимметричных криптосистем El Gamal.

Методика выполнения работы

Задание на выполнение лабораторной работы выдаётся преподавателем после прохождения студентами собеседования по основам криптографической защиты информации.

Порядок выполнения работы соответствует приведённой ниже криптосистеме шифрования данных по схеме Эль-Гамаля.

Схема алгоритма шифрования данных Эль-Гамаля

Определение открытого "y" и секретного "x" ключей

Выбор двух взаимно простых больших чисел p и q, q <p

Выбор значения секретного ключа x, x <p

. Определение значения открытого ключа y из выражения:

$y = qx \pmod p$

Алгоритм шифрования сообщения M

Выбор случайного числа a, удовлетворяющего условию:

$0 \leq k < p - 1$ и НОД (к ,p -1) = 1

Определение значения a из выражения: $a = qx \pmod p$

Определение значения b из выражения: $b = yk\ M \pmod p$

Криптограмма C, состоящая из a и b, отправляется получателю

Получатель расшифровывает криптограмму с помощью выражения:

$M\ ax = b \pmod p$

Процедуру шифрования данных рассмотрим на следующее примере:

(для удобства расчётов в данном примере использованы числа малой разрядности):

Выбираем два взаимно простых числа p = 11 и q = 2;

Выбираем значение секретного ключа x, (x <p), x = 8;

Вычисляем значение открытого ключа y из выражения

$y = qx \pmod p = 28 \pmod{11} = 256 \pmod{11} = 3$

Выбираем значение открытого сообщения M= 5;

Выбираем случайное число k = 9; НОД (9, 10) = 1;

Определяем значение a из выражения:

$a = qk \pmod p = 29 \pmod{11} = 512 \pmod{11} = 6$;

Определяем значение b из выражения:

$b = yk\ M \pmod p = 39*5 \pmod{11} = 98415 \pmod{11} = 9$.

Таким образом, получаем зашифрованное сообщение как (a, b) = (6, 9) и отправляем

получателю.

Получатель расшифровывает данный шифротекст, используя секретный ключ x и решая следующее сравнение:

M*a ≡b (mod p) = 5*68 ≡ 9 (mod 11) = 8398080 =9 (mod 11)

Вычисленное значение сообщения M = 5 представляет собой заданное исходное сообщение.

2.5 Лабораторная работа №7 (2 часа).

Тема: «Программная реализация асимметричных криптосистем (RSA, El Gamal, Рабина)»

2.5.1 Цель работы: Освоить программную реализацию асимметричных криптосистем (RSA, El Gamal, Рабина)

2.5.2 Задачи работы:

1. Программная реализация асимметричного алгоритма шифрования данных RSA.

2.5.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Электронно-вычислительная машина.

2.5.4 Описание (ход) работы:

На данный момент асимметричное шифрование на основе открытого ключа RSA (расшифровывается, как Rivest, Shamir and Aldeman - создатели алгоритма) использует большинство продуктов на рынке информационной безопасности.

Его криптостойкость основывается на сложности разложения на множители больших чисел, а именно - на исключительной трудности задачи определить секретный ключ на основании открытого, так как для этого потребуется решить задачу о существовании делителей целого числа. Наиболее криптостойкие системы используют 1024-битовые и большие числа.

Рассмотрим алгоритм RSA с практической точки зрения.

Для начала необходимо сгенерировать открытый и секретные ключи:

- Возьмем два больших простых числа p and q.
- Определим n, как результат умножения p on q (n= p*q).
- Выберем случайное число, которое назовем d. Это число должно быть взаимно простым (не иметь ни одного общего делителя, кроме 1) с результатом умножения (p-1)*(q-1).
- Определим такое число e, для которого является истинным следующее соотношение (e*d) mod ((p-1)*(q-1))=1.
- Назовем открытым ключем числа e и n, а секретным - d и n.

- Для того, чтобы зашифровать данные по открытому ключу {e,n}, необходимо следующее:
- разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа M(i)=0,1,2..., n-1(т.е. только до n-1).
- зашифровать текст, рассматриваемый как последовательность чисел M(i) по формуле C(i)=(M(I)^e)mod n.

Чтобы расшифровать эти данные, используя секретный ключ {d,n}, необходимо выполнить следующие вычисления: M(i) = (C(i)^d) mod n. В результате будет получено множество чисел M(i), которые представляют собой исходный текст.

Следующий пример наглядно демонстрирует алгоритм шифрования RSA:

Зашифруем и расшифруем сообщение "CAB" по алгоритму RSA. Для простоты возьмем небольшие числа - это сократит наши расчеты.

- Выберем p=3 and q=11.
- Определим n= 3*11=33.
- Найдем (p-1)*(q-1)=20. Следовательно, d будет равно, например, 3: (d=3).
- Выберем число e по следующей формуле: (e*3) mod 20=1. Значит e будет равно, например, 7: (e=7).
- Представим шифруемое сообщение как последовательность чисел в диапозоне от 0 до 32 (незабывайте, что кончается на n-1). Буква A =1, B=2, C=3.

Теперь зашифруем сообщение, используя открытый ключ {7,33}

C1 = (3^7) mod 33 = 2187 mod 33 = 9;

C2 = (1^7) mod 33 = 1 mod 33 = 1;

C3 = (2^7) mod 33 = 128 mod 33 = 29;

Теперь расшифруем данные, используя закрытый ключ {3,33}.

M1=(9^3)mod33=729mod33=3(C);

M2=(1^3)mod33=1mod33=1(A);

M3=(29^3)mod33=24389mod33=2(B);

Данные расшифрованы!

Задание.

1. Создать программную реализацию алгоритма RSA.
2. Исследовать зависимость времени шифрования и дешифрования файлов от размера файла и длины ключа, результаты представить в графическом или табличном виде.
3. Сформировать и представить преподавателю отчет по результатам выполнения лабораторной работы.

2.6 Лабораторная работа №8-9 (4часа).

Тема: «Исследование тестов на простоту и алгоритмы факторизации»

2.6.1 Цель работы: Провести исследование тестов на простоту и алгоритмы факторизации

2.6.2 Задачи работы: Провести исследование тестов на простоту и алгоритмы факторизации

2.6.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Электронно-вычислительная машина.

2.6.4 Описание (ход) работы:

1. Рабочее задание

1.1. Написать программу на языке "UBASIC", реализующую алгоритм факторизации модуля шифрования n из примера, приведенного ниже.

1.2. Найти разложение n=p*q.

1.3. Восстановить секретный ключ d.

1.4. Разложить на множители числа p-1 и q-1, убедиться, что в их разложении имеются большие делители.

2. Выполнение задания

2.1. Предположим, что Вы имеете открытый ключ схемы RSA (e,n), где n=525169521992627614583344195951527749, а в качестве ключа зашифрования e используете числа по вариантам, определенным преподавателем:

1 вариант - e=131

2 вариант - e=133

3 вариант - e=137

4 вариант - e=139

5 вариант - e=149

6 вариант - e=151

7 вариант - e=157

8 вариант - e=161

2.2. Составьте программу на языке "UBASIC", реализующую алгоритм факторизации модуля шифрования n по формуле (3). При реализации алгоритма используйте встроенные функции языка:

Y=MODINV(X,P) - вычисляет обратное к X по модулю P;

Y=MODPOW(A,X,P) - вычисляет Y=A**X mod P;

Y=GCD(A,B) - вычисляет наибольший общий делитель чисел A и B.

2.3. По разложению n=pq восстановите секретный ключ расшифрования d.

2.4. С помощью программы ECMX.UB разложите на множители числа p-1 и q-1.

2.5. Напишите отчет.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

3.1 Практическое занятие № 1 (2 часа).

Тема: «Поточные системы шифрования (РСЛОС, RC4, Рона)**»**

3.1.1 Задание для работы:

1. Система РСЛОС.
2. Система Рона.
3. Система RC4.

3.1.2 Краткое описание проводимого занятия:

Пусть известно, что исходное сообщение представляло собой двоично-десятичное число, то есть число, каждая тетрада (четыре бита) которого получена при переводе десятичной цифры 0...9 в двоичный вид. Перехвачено 24 бита зашифрованного сообщения Y, то есть шесть тетрад Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, а именно *значение* 1100 1101 1110 1111 0000 0001. Известно, что *ключ шифрования* состоял из четырех *бит*, которые тоже представляют собой однозначное десятичное число, то есть одно и то же *значение* $0 \le K \le 9$ использовалось для шифрования каждых четырех *бит* исходного сообщения. Таким образом, *шифрование* числа X_1, X_2, X_3, X_4, X_5, X_6 ключом K можно представить в виде системы уравнений:

$$X_1 \oplus K = 1100 \;\; X_2 \oplus K = 1101 \;\; X_3 \oplus K = 1110 X_4 \oplus K = 1111 \;\; X_5 \oplus K = 0000 \;\; X_6 \oplus K = 0001$$

Исходя из условия, что X_i принимает десятичные значения от 0 до 9, для поиска неизвестного K определим все возможные значения X_1'; и K, сумма которых по модулю 2 приводит к результату 1100:

K = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001
Y_1 = 1100 1100 1100 1100 1100 1100 1100 1100 1100 1100
X_1' = 1100 1101 1110 1111 1000 1001 1010 1011 0100 0101

Так как исходное *значение* состояло из цифр от 0 до 9, то можно исключить из рассмотрения значения ключа 0000, 0001, 0010, 0011, 0110, 0111, так при сложении с ними получаются значения большие 9 в десятичном эквиваленте. Такие значения не могли присутствовать в открытом тексте. Таким образом, первый этап анализа уже позволил сократить количество возможных ключей с десяти до четырех.

Для дальнейшего поиска неизвестного K определим все возможные значения X_2'; и оставшихся вариантов ключа, сумма которых по модулю 2 приводит к результату Y_2 = 1101:

K = 0100 0101 1000 1001
Y_2 = 1101 1101 1101 1101
$X_2^{,}$ = 1001 1000 0101 0100

Видно, что этот этап не позволил отбросить ни одного из оставшихся вариантов ключа. Попытаемся это сделать, используя Y_3=1110:

K = 0100 0101 1000 1001
Y_3 = 1110 1110 1110 1110
$X_2^{,}$ = 1010 1011 0110 0111

После проведения этого этапа становится ясно, что ключом не могли быть значения 0100 и 0101. Остается два возможных значения ключа:$1000_{(2)}=8_{(10)}$ и $1001_{(2)}=9_{(10)}$.

Дальнейший *анализ* по данной методике в данном случае, к сожалению, не позволит однозначно указать, какой же из двух полученных вариантов ключа использовался при шифровании. Однако можно считать успехом уже то, что *пространство* возможных ключей снизилось с десяти до двух. Остается попробовать каждый из двух найденных ключей для дешифровки сообщений и проанализировать смысл полученных вскрытых текстов.

В реальных случаях, когда исходное сообщение составлено не только из одних цифр, но и из других символов, использование статистического анализа позволяет быстро и точно восстановить *ключ* и исходные сообщения при короткой длине ключа, закрывающего *поток* секретных данных.

3.1.3 Результаты и выводы:

На практике освоены поточные системы шифрования (РСЛОС, RC4, Рона)

3.2 Практическое занятие № 2 (2 часа).

Тема: «Программная реализация поточных систем шифрования (РСЛОС, RC4, Рона)**»**

3.2.1 Задание для работы:

1. Программная реализация поточных систем шифрования.

3.2.2 Краткое описание проводимого занятия:

В блочном шифре из двух одинаковых блоков открытого текста получаются одинаковые блоки шифрованного текста. Избежать этого позволяют поточные шифры, в которых шифрующее преобразование “элемента” открытого текста меняется от одного элемента к другому. Так в DES в режиме сцепления блоков фактически происходит преобразование блочного шифра в поточный, что облегчает обнаружение искажений блоков и затрудняет попытки имитации и подмены. Специалисты, однако, используют термин поточный шифр только в том случае, когда “элементы” открытого текста очень малы (одна буква или один бит). Обычно аппаратные реализации поточных шифров быстрее и проще, чем блоковых. Потоковые шифры пригодны для шифрования непрерывных потоков данных, например, в сетях передачи данных. Самые популярные сейчас поточные шифры можно назвать двоичными аддитивными. В таких шифрах k -

битовый секретный ключ ' Z используется только для управления генератором ключевого потока, порождающего двоичную последовательность 0 1 1 , ,..., N − z z z , называемую ключевым потоком, где N >> k . Шифр текст образуется путем сложения по модулю 2 битов открытого текста и битов ключевого потока: Шифрование i i i Y = X ⊕z , i = 1,0 ,..., N −1 Дешифрование i i i X = Y ⊕ z , i = 1,0 ,..., N −1 Шифрование и дешифрование выполняются одинаковыми устройствами. Аддитивный поточный шифр похож на двоичный шифр с ключом однократного применения. Если ' i Zi z = , i = 1,0 ,..., k , k = N , т. е. секретный ключ используется как ключевая последовательность, то аддитивный поточный шифр есть шифр с ключом однократного применения. В практических поточных шифрах длина N шифртекста много больше длины k секретного ключа, а ключевая последовательность является псевдослучайной и имеет некоторый период. Стойкость системы целиком зависит от внутренней структуры генератора ключевой последовательности. Если генератор выдает последовательность с небольшим периодом, то стойкость системы будет невелика. Если бы генератор выдавал бесконечную последовательность битов, в которой каждый бит порождался независимо и с вероятностью 1/2 принимал значения 0 или 1 , то мы получили бы совершенно стойкий шифр.

3.2.3 Результаты и выводы:

На практике освоить программную реализацию поточных систем шифрования (РСЛОС, RC4, Рона).

3.3 Практическое занятие № 3 (2 часа).

Тема: «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)»

3.3.1 Задание для работы:

1. Схема распределения ключей, основанных на эллиптических кривых.

3.3.2 Краткое описание проводимого занятия:

Открытый элемент P задан в табл. 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы ***i,*** вычислить ключ обмена для участника с номером ***35 – i*** по алгоритму Диффи – Хэллмана.

Пример выполнения Задания3. Алгоритм Диффи-Хеллмана

Открытый элемент P задан в таблице 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы ***i,*** вычислить ключ обмена для участника с номером ***35 – i*** по алгоритму Диффи-Хэллмана.

Вариант № 15, группа 2091 (№ 1)

Номер в списке группы i = **15**

Номер группы k = **1**

P = 37

Открытый элемент P = 37. Найти примитивный элемент поля. Секретный ключ каждого участника i = 15, вычислить ключ обмена для участника с номером 35-15 = 20.

GF(37) = <0, 1, 2, 3, …, 35, 36, 37>

Найдем примитивный элемент поля GF(37).

Требуется найти такое число, принадлежащее интервалу [2,37], которое при возведении в 37-ю степень по модулю 37 будет давать в результате единицу. Если же единица будет получена раньше, чем при возведении в 36-ю степень, результаты возведения в степень начнут повторяться, и через выбранный элемент не удастся

представить все элементы поля. Исходя из таких соображений, получаем несложный алгоритм нахождения примитивных элементов поля GF(37).

Алгоритм нахождения примитивных элементов поля GF(37).

```
int _tmain(int argc, _TCHAR* argv[])
{
  long int mem, i, j, num, deg, modul, res;
  unsigned char mas[38];
  deg = 0; modul = 37; mem = 2;
  while (mem < 36)
  {
        num = mem; deg = 0;
        while (deg != 36)
        {
              res = 1; deg = 0;
              for (i = 0; i < 36; i++) mas[i] = 0;
              do
              {
                    res = res*num; res = res % modul;
                    deg++; mas[deg] = res;
              }
              while(res != 1);
              num++;
        }
        std::cout << num-1 << std::endl;
        for (i = 0; i < 36; i++)
        {
              for (j = 0; j < 36; j++)
              {
                    if ((i != j) && (mas[j] == mas[i]))
                          std::cout << "Wrong" << std::endl;
              }
              mem = num;
        }
  }
}
```

В результате выполнения программы получим следующие значения:

2, 5, 13, 15, 17, 18, 19, 20, 22, 24, 32, 35

Выберем значение б = 2.

Секретный ключ участника А: $X_A = 15$.

Секретный ключ участника В: $Y_B = 35\text{-}15 = 20$.

Открытый ключ участника А: $K_A = \alpha^{X_A} = \alpha^{15} = 2^{15} \bmod 37 = 23$

Открытый ключ участника В: $K_B = \alpha^{Y_B} = \alpha^{20} = 2^{20} \bmod 37 = 33$

$K_A = 23$

$K_B = 33$

Обменный ключ участника А: $K = K_B{}^{X_A} \bmod \alpha = 33^{15} \bmod 37 = 26$

Обменный ключ участника В: $K = K_A{}^{Y_B} \bmod \alpha = 23^{20} \bmod 37 = 26$

Значения обменного ключа для А и В совпадают.

Обменный ключ: K = 26

3.3.3 Результаты и выводы:

На практике решить задачу распределения ключей на основе алгоритма Диффи — Хеллмана.

3.4 Практическое занятие № 4 (2 часа).

Тема: «Схемы распределения ключей (Шамира, Диффи-Хеллмана, протоколов основанных на эллиптических кривых)**»**

3.4.1 Задание для работы:

1. Схема распределения ключей, основанных на эллиптических кривых.

3.4.2 Краткое описание проводимого занятия:

Открытый элемент *P* задан в табл. 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы ***i,*** вычислить ключ обмена для участника с номером ***35 – i*** по алгоритму Диффи – Хэллмана.

Пример выполнения Задания3. Алгоритм Диффи-Хеллмана

Открытый элемент P задан в таблице 3 – графа 2. Найти примитивный элемент поля. Считая, что секретный ключ каждого участника равен номеру студента в списке группы ***i,*** вычислить ключ обмена для участника с номером ***35 – i*** по алгоритму Диффи-Хэллмана.

Вариант № 15, группа 2091 (№ 1)
Номер в списке группы i = **15**
Номер группы k = **1**
P = 37

Открытый элемент P = 37. Найти примитивный элемент поля. Секретный ключ каждого участника i = 15, вычислить ключ обмена для участника с номером 35-15 = 20.

GF(37) = <0, 1, 2, 3, …, 35, 36, 37>

Найдем примитивный элемент поля GF(37).

Требуется найти такое число, принадлежащее интервалу [2,37], которое при возведении в 37-ю степень по модулю 37 будет давать в результате единицу. Если же единица будет получена раньше, чем при возведении в 36-ю степень, результаты возведения в степень начнут повторяться, и через выбранный элемент не удастся представить все элементы поля. Исходя из таких соображений, получаем несложный алгоритм нахождения примитивных элементов поля GF(37).

Алгоритм нахождения примитивных элементов поля GF(37).

```
int _tmain(int argc, _TCHAR* argv[])
{
  long int mem, i, j, num, deg, modul, res;
  unsigned char mas[38];
  deg = 0; modul = 37; mem = 2;
  while (mem < 36)
  {
        num = mem; deg = 0;
        while (deg != 36)
        {
              res = 1; deg = 0;
              for (i = 0; i < 36; i++) mas[i] = 0;
              do
```

```
                {
                        res = res*num; res = res % modul;
                        deg++; mas[deg] = res;
                }
                while(res != 1);
                num++;
        }
        std::cout << num-1 << std::endl;
        for (i = 0; i < 36; i++)
        {
                for (j = 0; j < 36; j++)
                {
                        if ((i != j) && (mas[j] == mas[i]))
                                std::cout << "Wrong" << std::endl;
                }
                mem = num;
        }
  }
}
```

В результате выполнения программы получим следующие значения:
2, 5, 13, 15, 17, 18, 19, 20, 22, 24, 32, 35
Выберем значение б = 2.
Секретный ключ участника А: $X_A = 15$.

Секретный ключ участника В: $Y_B = 35\text{-}15 = 20$.

Открытый ключ участника А: $K_A = \alpha^{X_A} = \alpha^{15} = 2^{15} \bmod 37 = 23$

Открытый ключ участника В: $K_B = \alpha^{Y_B} = \alpha^{20} = 2^{20} \bmod 37 = 33$

$K_A = 23$

$K_B = 33$

Обменный ключ участника А: $K = K_B{}^{X_A} \bmod \alpha = 33^{15} \bmod 37 = 26$

Обменный ключ участника В: $K = K_A{}^{Y_B} \bmod \alpha = 23^{20} \bmod 37 = 26$

Значения обменного ключа для А и В совпадают.

Обменный ключ: K = 26

3.4.3 Результаты и выводы:

На практике решить задачу распределения ключей на основе алгоритма Диффи — Хеллмана.

3.5 Практическое занятие № 5 (2 часа).

Тема: «Асимметричные криптосистемы (RSA, El Gamal, Рабина)»

3.5.1 Задание для работы:

1. Формирование асимметричных криптосистем.
2.Формирование асимметричных криптосистем RSA
3. Формирование асимметричных криптосистем Рабина .

4. Формирование асимметричных криптосистем El Gamal.

3.5.2 Краткое описание проводимого занятия:

1. Выбираем простые числа (небольшие, чтобы упростить вычисления): p=3p=3 и q=11q=11
2. Вычисляем модуль n=p∗q=3∗11=33n=p∗q=3∗11=33
3. Вычисляем функцию Эйлера от

модуля nn : φ(N)=(p−1)∗(q−1)=2∗10=20φ(N)=(p−1)∗(q−1)=2∗10=20.
4. Выбираем открытую экспоненту e=7e=7

5. Определяем закрытую экспоненту dd : d∗e=1(modφ(N))d∗e=1(modφ(N)) =>=> d=3d=3
Будем шифровать сообщение RSA, пусть букве A соответствует цифра 1, B - 2, C - 3 и т.д (Подобное соответствие вносим для простоты), тогда :
R=18;R=18; S=19;S=19; A=1;A=1;Открытый ключ : (e,n)=(7,33)(e,n)=(7,33)
C1=(187)mod33=6C1=(187)mod33=6
C2=(197)mod33=13C2=(197)mod33=13
C3=(17)mod33=1C3=(17)mod33=1
C("RSA")=6131

3.5.3 Результаты и выводы:

На практике рассмотреть пример шифрование с помощью алгоритма RSA

3.6 Практическое занятие № 6 (2 часа).

Тема: «Асимметричные криптосистемы (RSA, El Gamal, Рабина)**»**

3.6.1 Задание для работы:

1. Формирование асимметричных криптосистем.
2.Формирование асимметричных криптосистем RSA
3. Формирование асимметричных криптосистем Рабина .
4. Формирование асимметричных криптосистем El Gamal.

3.6.2 Краткое описание проводимого занятия:

1. Выбираем простые числа (небольшие, чтобы упростить вычисления): p=3p=3 и q=11q=11
2. Вычисляем модуль n=p∗q=3∗11=33n=p∗q=3∗11=33
3. Вычисляем функцию Эйлера от

модуля nn : φ(N)=(p−1)∗(q−1)=2∗10=20φ(N)=(p−1)∗(q−1)=2∗10=20.
4. Выбираем открытую экспоненту e=7e=7

5. Определяем закрытую экспоненту dd : d∗e=1(modφ(N))d∗e=1(modφ(N)) =>=> d=3d=3
Будем шифровать сообщение RSA, пусть букве A соответствует цифра 1, B - 2, C - 3 и т.д (Подобное соответствие вносим для простоты), тогда :
R=18;R=18; S=19;S=19; A=1;A=1;Открытый ключ : (e,n)=(7,33)(e,n)=(7,33)
C1=(187)mod33=6C1=(187)mod33=6
C2=(197)mod33=13C2=(197)mod33=13
C3=(17)mod33=1C3=(17)mod33=1
C("RSA")=6131

3.6.3 Результаты и выводы:

На практике рассмотреть пример шифрование с помощью алгоритма RSA

3.7 Практическое занятие № 7 (2 часа).

Тема: «Асимметричные криптосистемы (RSA, El Gamal, Рабина)»

3.7.1 Задание для работы:

1. Формирование асимметричных криптосистем.
2.Формирование асимметричных криптосистем RSA
3. Формирование асимметричных криптосистем Рабина .
4. Формирование асимметричных криптосистем El Gamal.

3.7.2 Краткое описание проводимого занятия:

1. Выбираем простые числа (небольшие, чтобы упростить вычисления): p=3p=3 и q=11q=11
2. Вычисляем модуль n=p∗q=3∗11=33n=p∗q=3∗11=33
3. Вычисляем функцию Эйлера от модуля nn : φ(N)=(p−1)∗(q−1)=2∗10=20φ(N)=(p−1)∗(q−1)=2∗10=20.
4. Выбираем открытую экспоненту e=7e=7
5. Определяем закрытую экспоненту dd : d∗e=1(modφ(N))d∗e=1(modφ(N)) =>=> d=3d=3

Будем шифровать сообщение RSA, пусть букве А соответствует цифра 1, В - 2, С - 3 и т.д (Подобное соответствие вносим для простоты), тогда :
R=18;R=18; S=19;S=19; A=1;A=1;Открытый ключ : (e,n)=(7,33)(e,n)=(7,33)
C1=(187)mod33=6C1=(187)mod33=6
C2=(197)mod33=13C2=(197)mod33=13
C3=(17)mod33=1C3=(17)mod33=1
C("RSA")=6131

3.7.3 Результаты и выводы:

На практике рассмотреть пример шифрование с помощью алгоритма RSA

3.8 Практическое занятие № 8 (2 часа).

Тема: «Асимметричные криптосистемы (RSA, El Gamal, Рабина)»

3.8.1 Задание для работы:

1. Формирование асимметричных криптосистем.
2.Формирование асимметричных криптосистем RSA
3. Формирование асимметричных криптосистем Рабина .
4. Формирование асимметричных криптосистем El Gamal.

3.8.2 Краткое описание проводимого занятия:

1. Выбираем простые числа (небольшие, чтобы упростить вычисления): p=3p=3 и q=11q=11
2. Вычисляем модуль n=p∗q=3∗11=33n=p∗q=3∗11=33
3. Вычисляем функцию Эйлера от модуля nn : φ(N)=(p−1)∗(q−1)=2∗10=20φ(N)=(p−1)∗(q−1)=2∗10=20.
4. Выбираем открытую экспоненту e=7e=7
5. Определяем закрытую экспоненту dd : d∗e=1(modφ(N))d∗e=1(modφ(N)) =>=> d=3d=3

Будем шифровать сообщение RSA, пусть букве А соответствует цифра 1, В - 2, С - 3 и т.д (Подобное соответствие вносим для простоты), тогда :
R=18;R=18; S=19;S=19; A=1;A=1;Открытый ключ : (e,n)=(7,33)(e,n)=(7,33)

C1=(187)mod33=6C1=(187)mod33=6
C2=(197)mod33=13C2=(197)mod33=13
C3=(17)mod33=1C3=(17)mod33=1
C("RSA")=6131

3.8.3 Результаты и выводы:

На практике рассмотреть пример шифрование с помощью алгоритма RSA

3.9 Практическое занятие №9 (2 часа).

Тема: «Программная реализация асимметричных криптосистем (RSA, El Gamal, Рабина)»

3.9.1 Задание для работы: реализация асимметричных криптосистем

3.9.2 Краткое описание проводимого занятия:

Когда заходит речь о выборе хорошего криптографического алгоритма, у выбирающего, как правило, имеется несколько возможностей:

- Можно воспользоваться известным алгоритмом, сравнительно давно опубликованным в специальном издании по проблемам криптографии. Если никто пока не сообщил о том, что сумел вскрыть этот алгоритм, значит, он стоит того, чтобы обратить на него внимание.
- Можно довериться известной фирме, специализирующейся на продаже средств шифрования. Вряд ли эта фирма будет рисковать своим добрым именем, торгуя нестойкими криптографическими алгоритмами.
- Можно обратится к независимому эксперту. Скорее всего, он сможет объективно оценить достоинства и недостатки различных криптографических алгоритмов.
- Можно обратится за поддержкой в соответствующее правительственное ведомство. Вряд ли правительство будет вводить своих граждан в заблуждение, давая им ложные советы относительно стойкости того или иного криптографического алгоритма.
- Можно попытаться создать собственный криптографический алгоритм. Мало кто заинтересован сам себя обманывать. Чем чёрт не шутит: а вдруг вы обладаете выдающимися способностями в области криптографии?

Во всех перечисленных возможностях имеются свои существенные изъяны. Полагаться только на одну фирму, на одного эксперта или на одно ведомство не совсем разумно. Многие люди, называющие себя независимыми экспертами, мало понимают в криптографии. Большинство фирм, производящих средства шифрования, ничуть не лучше. В АНБ и ФАПСИ работают лучшие криптографы в мире, однако, по понятным соображениям, они не спешат поделиться своими секретами с первым встречным. Впрочем, и со вторым тоже. И даже если вы гений в криптографии, глупо использовать криптографический алгоритм собственного изобретения без того, чтобы его всесторонне проанализировали и протестировали опытные криптологи.
Поэтому наиболее предпочтительной представляется первая из перечисленных возможностей. Данный подход к оценке стойкости криптографических алгоритмов можно было бы признать идеальным, если бы не один его недостаток. К сожалению, ничего неизвестно о результатах криптоаналитических исследований этих алгоритмов, которые, несомненно, активно велись в прошлом и продолжают также активно проводится во всем мире многочисленными сотрудниками различных правительственных ведомств, в компетенцию которых входят криптологические изыскания. Эти ведомства, скорее всего, гораздо лучше финансируются, чем академические институты, ведущие аналогичные

исследования. Да и начали они заниматься криптологией значительно раньше, чем ученые, не имеющие воинских званий, и специалисты из частных фирм. Поэтому можно предположить, что военные нашли гораздо более простые способы вскрытия известных шифров, нежели те, которые изобретены за пределами строго охраняемых зданий сверхсекретных правительственных ведомств.

3.9.3 Результаты и выводы:

На практике освоить программную реализацию асимметричных криптосистем.