**25.11.2024**

**Основные понятия компьютерной безопасности**

**Компьютерная безопасность** является одним из направлений в сфере компьютерной безопасности, которая отвечает в основном за обеспечение безопасности информации в компьютерах.

К **защищаемой** относится информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственниками информации.

**Защитой информации** называют деятельность по предотвращению утечки защищаемой информации, путём её разглашения, несанкционированного доступа к ней, несанкционированных и непреднамеренных воздействий на нём.

Под **утечкой** понимают неконтролируемое распространение защищаемой информации с использованием различных способов его распространения и получения несанкционированного доступа.

**Разглашение** – доведение защищаемой информации до неконтролируемого количества получателей информации.

**Несанкционированный доступ** – получение защищаемой информации заинтересованным субъектом с нарушением правил доступа к ней.

**Несанкционированное воздействие на защищаемую информацию** – воздействие с нарушением правил её изменения.

Целью защиты информации является предотвращение ущерба собственнику, владельцу или пользователю информации.

Под **эффективностью защиты информации** понимают степень соответствия результатов защиты информации поставленной цели.

**Объектом защиты** может быть информация, её носитель или информационный процесс, в отношении которых необходимо обеспечивать защиту в соответствии с поставленной целью.

Под **качеством информации** понимают совокупность свойств, обусловливающих пригодность информации удовлетворять определённые потребности её пользователей в соответствии с назначением информации. Одним из показателей качества информации является её защищённость, т.е. поддержание на заданном уровне тех параметров информации, которые характеризуют установленный статус её хранения, обработки и использования.

Основными характеристиками защищаемой информации являются: ***конфиденциальность***, ***целостность*** и ***доступность***.

**Конфиденциальность информации** – известность её содержания только имеющим соответствующие полномочия субъектам. Конфиденциальность является субъективной характеристикой информации, связанной с объективной необходимостью защиты законных интересов одних субъектов от других.

**Целостностью информации** называют неизменность информации в условиях её случайного и/или преднамеренного искажения или разрушения. Целостность является частью более широкой характеристики информации (это её достоверности), включающей помимо целостности ещё полноту и точность отображения предметной области.

Под **доступностью информации** понимают способность обеспечения беспрепятственного доступа субъектов к интересующей их информации.

**Отказом в обслуживании** называют состояние информационной системы, при котором блокируется доступ к некоторому её ресурсу. Совокупность информационных ресурсов и системы формирования распространения и использования информации называют **информационной средой общества**.

Под **информационной безопасностью** понимают состояние защищённости информационной среды, обеспечивающее её формирование и развитие.

**Политика безопасности** – набор документированных норм, правил и практических приёмов, регулирующих управление, защиту и распределение информации ограниченного доступа.

**Защита информации** – комплекс мероприятий, направленных на обеспечение информационной безопасности. Основными предметными направлениями защиты информации являются: охрана государственной, коммерческой, служебной, банковской тайн, персональных данных и интеллектуальной собственности.

Под **системой безопасности** понимают организованную совокупность специальных органов, служб, средств, методов и мероприятий, обеспечивающих защиту жизненно важных интересов личности, предприятия и государства от внутренних и внешних угроз.

С позиции системного подхода к защите информации предъявляются определённые требования:

1. Обеспечение безопасности информации не может быть одноразовым актом;
2. Безопасность информации может быть обеспечена лишь при комплексном использовании всего арсенала имеющихся средств защиты во всех структурных элементах системы и на всех этапах технологического цикла обработки информации;
3. Планирование безопасности информации осуществляется путём разработки каждой службой детальных планов защиты информации в сфере её компетенции;

**26.11.2024**

1. Защите подлежат конкретные данные, объективно подлежащие охране, утрата которых может причинить организации определённый ущерб;
2. Методы и средства защиты должны надёжно перекрывать возможные пути неправомерного доступа к охраняемым секретам;
3. Эффективность защиты информации означает, что на её осуществление затраты не должны быть больше возможных потерь от реализации информационных угроз;
4. Чёткость определения полномочий и прав пользователей на доступ к определённым видам информации;
5. Предоставление пользователю минимальных полномочий необходимых ему для выполнения порученной работы;
6. Сведение к минимуму числа общих для нескольких пользователей средств защиты;
7. Учёт случаев и попыток несанкционированного доступа к конфиденциальной информации;
8. Обеспечение степени конфиденциальной информации;
9. Обеспечение контроля целостности средств защиты и немедленное реагирование на их выход из строя.

Система защиты информации, как любая система, должна иметь определённые виды системного обеспечения, опираясь на которые она будет выполнять свою целевую функцию. С учётом этого система защиты информации может иметь различные виды или типы обеспечений, к которым относятся:

1. **Правовое обеспечение**. Сюда входят: нормативные документы, положения, инструкции, руководства, требования которых являются обязательными в рамках сферы действия;
2. **Организационное обеспечение**. Под этим пунктом имеется ввиду, что реализация защиты информации осуществляется определёнными структурными единицами, такими как: служба безопасности, служба режима, служба защиты информации техническими средствами, служба защиты информации программными средствами и т.д.;
3. **Аппаратное обеспечение**. Предполагается широкое использование технических средств, как для защиты информации, так и для обеспечения деятельности собственно системы защиты информации;
4. **Информационное обеспечение**. Оно включает в себя документированные сведения, лежащие в основе решения задач, обеспечивающих функционирование системы. Сюда могут входить, как показатели доступа, учёта, хранения, так и системы информационного обеспечения расчётных задач различного характера, связанных с деятельностью службы обеспечения безопасности;
5. **Программное обеспечение**. К нему относятся антивирусные программы, а также программы, реализующие контрольные функции при решении учётных, статистических, финансовых, кредитных и других задач;
6. **Математическое обеспечение**. Предполагает использование математических методов для различных расчётов, связанных с оценкой опасности технических средств злоумышленников, зон и норм необходимой защиты;
7. **Лингвистическое обеспечение**. Это совокупность специальных языковых средств общения специалистов в области защиты информации;
8. **Нормативно методическое обеспечение**. Сюда входят нормы и регламенты деятельности органов, служб, средств, реализующих функции защиты информации различного рода методики, обеспечивающие деятельность пользователей при выполнении своей работы в условиях жёстких требований защиты информации;
9. **Эргономическое обеспечение**. Это совокупность средств, обеспечивающих удобство работы пользователей аппаратных средств защиты информации.

**27.11.2024**

**Угрозы информации**

**Угроза** – потенциальная возмужать нарушения режима информационной безопасности. Под **угрозой информационной безопасности** понимается действие или событие, которое может привести к разрушению, искажению или несанкционированному использованию информационных ресурсов, включая хранимую, передаваемую и обрабатываемую информацию, а также программные и аппаратные средства.

Если ценность информации теряется при её хранении и/или распространении, то реализуется **угроза нарушения конфиденциальности информации**.

Если информация изменяется или уничтожается с потерей её ценности, то реализуется **угроза целостности информации**.

Если информация вовремя не поступает легальному пользователю, то её ценность уменьшается и со временем полностью обесценивается, тем самым реализуется **угроза оперативности использования** или **доступности информации**.

Попытка реализации угрозы называется **атакой**, а того, кто предпринимает такую попытку называют **злоумышленником**. Потенциальные злоумышленники называются **источниками угрозы**. Чаще всего угроза является следствием наличия уязвимых мест в защите информационных систем.

Реализация угрозы информационной безопасности заключается в нарушении конфиденциальности, целостности и доступности информации. Злоумышленник может ознакомиться с конфиденциальной информацией, модифицировать её или даже уничтожить, а также ограничить или блокировать доступ легального пользователя к информации. При этом злоумышленник может являться либо сотрудником организации, либо посторонним лицом. Кроме того, ценность информации может уменьшиться ввиду случайных неумышленных ошибок персонала, а также различных природных явлений.

Промежуток времени, от момента, когда появляется возможность использовать слабое место до момента, когда пробел ликвидируется называется – **окном опасности**, который ассоциируется с данным уязвимым местом. Пока существует окно опасности, возможны успешные атаки на информационную систему.

В основном существуют два вида угроз, это **умышленные угрозы** и **неумышленные угрозы** (случайные). Умышленные угрозы преследуют цель нанесения ущерба пользователям информационной системы, и в свою очередь подразделяются на активные и пассивные.

**Пассивные угрозы** как правило направлены на несанкционированное использование информационных ресурсов, не оказывая при этом влияния на их функционирование.

**Активные угрозы** имеют целью нарушение нормального процесса функционирования системы посредством целенаправленного воздействия на аппаратные, программные и информационные ресурсы.

**Умышленные угрозы** также подразделяются на внутренние и внешние.

**Внутренние угрозы** чаще всего определяются социальной напряжённостью и тяжёлым моральным климатом.

**Внешние угрозы** могут определяться злонамеренными действиями конкурентов, экономическими условиями и другими причинами.

К основным угрозам безопасности относятся:

1. Раскрытие конфиденциальной информации;
2. Компрометация информации;
3. Несанкционированное использование информационных ресурсов;
4. Ошибочное использование ресурсов;
5. Несанкционированный обмен информацией;
6. Отказ от информации;
7. Отказ от обслуживания.

Угрозы информационной безопасности классифицируются по нескольким критериям:

1. По аспекту информационной безопасности (доступность, целостность, конфиденциальность), против которого угрозы направлены в первую очередь;
2. По компонентам информационных систем, на которые угрозы нацелены (данные, программы, аппаратура, поддерживающая инфраструктура);
3. По способу осуществления (случайные или преднамеренные действия природного или техногенного характера);
4. По расположению источника угрозы (внутри или вне рассматриваемой информационной системы).

**Виды угроз**

Бывают по основным нарушениям:

1. Физическая целостность;
2. Логическая целостность;
3. Содержание (изменение блоков информации и внешнее навязывание ложной информации);
4. Конфиденциальность (разрушение защиты и уменьшение степени защищённости информации);
5. Права собственности на информацию.

Виды по характеру происхождения:

1. **Умышленные факторы**:
2. Хищение носителей информации;
3. Подключение к каналам связи;
4. Перехват электромагнитных излучений;
5. Несанкционированный доступ;
6. Разглашение информации;
7. Копирование данных.
8. **Естественные факторы**:
9. Несчастные случаи;
10. Стихийные бедствия;
11. Ошибки в процессе обработки информации.

**Существует 3 выраженных угроз:**

1. Подверженность физическому искажению или уничтожению.
2. Возможность несанкционированной модификации.
3. Опасность несанкционированного получения информации лицами, для которых она не предназначена.

Предпосылками появления угроз являются:

1. **Объективные причины** – не связанные непосредственно с деятельностью людей и вызывающие случайные по характеру происхождения угрозы
2. **Субъективные причины** – непосредственно связанные с деятельностью человека и вызывающие как преднамеренные, так и непреднамеренные угрозы информации

**29.11.2024**

**Вредоносное программное обеспечение**

Одним из опаснейших способов проведения атак является внедрение в атакуемые системы **вредоносного ПО**. Существуют в основном 3 основных грани вредоносного ПО:

1. Вредоносная функция;
2. Способ распространения;
3. Внешнее представление.

Вредоносное программное обеспечение по механизму распространения разделяется на 2 группы:

1. **Вирусы** – код, обладающий способностью к распространению путём внедрения в другие программы;
2. **Черви** – код, способный самостоятельно, т.е. без внедрения в другие программы вызывать распространение своих копий по информационной системе и их выполнение (для активации вируса требуется запуск заражённой программы).

**Компьютерный вирус** – специально написанная программа, как правило небольшая по размерам, способная самопроизвольно присоединяться к другим программам, создавать свои копии и внедрять их в файлы, системные области диска и в вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создание всевозможных помех в работе на компьютере. Процесс внедрения вирусом своих копий в другие программы называется **заражением**, а объект содержащий вирус является **заражённым**.

Основными путями проникновения вирусами в компьютер являются съёмные носители информации, а также компьютерные сети.

Заражение жёсткого диска вирусом может произойти при загрузке компьютера с диска, содержащего вирус. Такое заражение может быть и случайным, например, не отключив внешний носитель, выполнить перезагрузку в случае, когда на компьютере установлены параметры загрузки с внешних носителей.

**Заражённый диск** – диск, в загрузочном секторе которого находится вирус.

**Заражённый файл** – файл, содержащий внедрённый в него вирус.

При заражении компьютера вирусом очень важно своевременно его обнаружить, так как действия вирусов могут наносить большой вред владельцам компьютеров. Существуют следующие основные признаки появления вирусов:

1. Невозможность загрузки ОС;
2. Прекращение работы или неправильная работа ранее успешно функционировавших программ;
3. Блокировка ввода с клавиатуры;
4. Замедление работы компьютера;
5. Изменение размеров, даты и времени создания файлов;
6. Значительное увеличение количества файлов на диске;
7. Исчезновение файлов и каталогов или их искажение;
8. Существенное уменьшение размера свободной оперативной памяти;
9. Блокировка записи на жёсткий диск;
10. Непредусмотренное требование снятия защиты с дисков или других внешних носителей;
11. Вывод на экран непредусмотренных сообщений или изображений;
12. Подача непредусмотренных звуковых сигналов;
13. Частые зависания и сбои в работе компьютера.

Перечисленные признаки необязательно вызываются присутствием вируса, а могут быть следствием других причин. Поэтому иногда затруднена правильная диагностика. Для правильно диагностики необходимо использование нескольких различных видов антивирусных ПО. На сегодняшний день известно огромное количество вирусов, например, в базах антивирусных ПО содержатся более 28 миллионов записей. Как и обычным вирусам для размножения компьютерным вирусам нужен носитель, т.е. здоровая программа или документ, в котором они прячут в участке своего программного кода.

Способы распространения вирусов и червей отличаются:

**Вирусы** обычно распространяются локально в пределах узла сети, для передачи по сети им требуется внешняя помощь, такая как пересылка заражённого файла.

**Черви**, напротив, ориентированы в первую очередь на путешествие по сети. Сам вирус не велик, редко его размер измеряется килобайтами. В тот момент, когда пользователь запускает на своём компьютере программу или открывает документ содержащий вирус, вирус активизируется и заставляет компьютер следовать его инструкциям. Это приводит к изменению или удалению какой-либо информации, причём чаще всего без права исправления или восстановления.

Современные вирусы могут не только внести изменения в работе компьютера, т.е. программы, но и могут вывести из строя железо: могут уничтожать содержимое, BIOS-а, других чипов материнской платы или повреждают жёсткий диск.

Вирусы появились более 50 лет назад. Именно тогда, в конце 60-ых годов, несколько больших электронов ЭВМ, располагавшихся в крупных исследовательских центрах США, про которые можно было прочесть лишь в фантастических романах или об их частях в научных журналах, обнаружились очень необычные программы. Они не выполняли распоряжения человека как другие программы, а действовали сами по себе. Такими своими действиями они очень сильно замедляли работу машин, но при этом ничего не портили и не размножались.

С 70-ых годов были зарегистрированы первые вирусы, способные к размножению и получившие собственные имена, например, большой компьютер Univac 1108 был заражён вирусом Pervading Animal или компьютеры серии IBM-360/370 были заражены Christmas Tree.

В 80-ых годах число активных вирусов определялось уже сотнями, а появление и распространение персональных компьютеров породило настоящую эпидемию – счёт вирусов пошёл на тысячи. Термин **компьютерный вирус** появился в 1984 году и впервые его использовал в своём докладе на конференции по информационной безопасности сотрудник лихайского университета США Фред Коуэн.

**03.12.2024**

Первые компьютерные вирусы были простыми и неприхотливыми и не скрывались от пользователей и скрашивали все свои разрушительные действия, к которым относятся: удаление файлов, разрушение логической структуры дисков, выводимыми на экран картинками и шутками. Выявить такие вирусы было нетрудно так как он присоединялись к исполняемым файлам изменяя их оригинальные размеры.

Позднее вирусы стали прятать свой программный код так что ни один антивирус не мог его обнаружить. Такие вирусы назывались **невидимками**.

В 90-х годах вирусы стали мутировать, т.е. постоянно изменять свой программный код и при этом спрятав его в различных участках жёсткого диска. Такие вирусы-мутанты стали называть **полиморфными**.

В 1995 году, после появления ОС Windows 95, были зарегистрированы вирусы, работающие под управлением Windows. Примерно через полгода были обнаружены вирусы, которые действовали в документах, подготовленных программами пакета Microsoft Office. Долгое время заражение вирусами файлов документов считалось невозможным, так как документы не содержали исполняемых программ. Однако, при встраивании языка программирования VBA в документах текстового языка Word и электронной таблицы Excel, программисты корпорации Microsoft. Язык VBA в данном случае использовался для создания специальных дополнений к процессам под названием **макросы**. Эти макросы сохранялись в теле документов Microsoft Office и легко могли быть заменены вирусами. После открытия заражённого файла вирус активировался и заражал все документы пакета.

Первоначально **макровирусы** наносили вред только текстовым документам. Позднее стали уничтожать информацию. Весомый вклад в распространение вирусов внесла сеть интернет. Впервые внимание общественности к проблеме **интернет-вирусов** было привлечено после появления знаменитого **Червя Морриса,** распространившегося по всей мировой сети.

**Червь Морриса** является одним из первых [сетевых червей](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%87%D0%B5%D1%80%D0%B2%D1%8C), распространявшихся через [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Написан [аспирантом](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D0%BF%D0%B8%D1%80%D0%B0%D0%BD%D1%82%D1%83%D1%80%D0%B0) Корнельского университета [Робертом Таппаном Моррисом](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%80%D1%80%D0%B8%D1%81,_%D0%A0%D0%BE%D0%B1%D0%B5%D1%80%D1%82_%D0%A2%D1%8D%D0%BF%D0%BF%D1%8D%D0%BD) и запущен 2 ноября 1988 года в [Массачусетском технологическом институте](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%81%D0%B0%D1%87%D1%83%D1%81%D0%B5%D1%82%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82). Это был первый вирус, получивший значительное внимание в средствах массовой информации, и парализовал работу 6.000 узлов в США. Он также привёл к первой судимости в США по Computer Fraud and Abuse Act 1986 года.

К 1998 Internet стал главным поставщиком вирусов. Возник даже целый класс Internet-вирусов, названных **троянскими**. Эти программы не причиняли вреда компьютеру и хранящейся в нем информации, зато с легкостью могли украсть пароль и логин для доступа к сети, а также другую секретную информацию.

В течение 1998-1999 гг. мир потрясли несколько разрушительных вирусных атак: в результате деятельности вирусов Melissa, Win95.CIH и Chernobyl были выведены из строя около миллиона компьютеров во всех странах мира. Вирусы портили жёсткий диск и уничтожали BIOS материнской платы.

Для защиты от компьютерных вирусов следует соблюдать следующие правила работы с компьютером:

1. Установка современного антивирусного ПО, а также его обновление;
2. Проверять внешние носители на наличие вирусов перед использованием;
3. Проверять архивы;
4. Периодически проводить проверку на вирусы;
5. Настроить файрволл.

**Классификация компьютерных вирусов**

В основе классификации компьютерных вирусов лежат 4 признака:

1. **По разрушительным возможностям** выделяются 3 вида вирусов:
   1. **Неопасные** – уменьшают объёмы памяти в результате своего распространения и иногда выдают какие-либо текстовые графические или звуковые сообщения, но не осуществляют сознательной порчи информации;
   2. **Опасные** – приводят к различным нарушениям в работе компьютера;
   3. **Очень опасные** – приводят к потере программ и данных, стиранию информации в системных областях памяти и даже к выходу из строя комплектующих частей компьютера.
2. **По способу заражения** выделяются 2 вида вирусов:
   1. **Резидентные** – при заражении компьютера оставляют в оперативной памяти свою резидентную часть, которая потом перехватывает обращение ОС ко всем объектам и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера;
   2. **Нерезидентные** – не заражают память компьютера и являются активными ограниченное время. Такие вирусы активизируются в определённые моменты при выполнении определённых операций.
3. **По среде обитания** существует 4 вида вирусов:
   1. **Файловые** – заражают исполняемые файлы и различные вспомогательные, загружаемые при выполнении других программ;
   2. **Загрузочные** – внедряются в начальный сектор дисков содержащий загрузчик ОС и начинают свою работу при загрузке компьютера. Загрузочные вирусы являются резидентными и заражают все носители вставляемый в компьютер;
   3. **Файлово-загрузочные** – заражают одновременно файлы и загрузочные секторы дисков, часто системные файлы. Как правило такие вирусы имеют довольно сложный алгоритм работы, часто применяют оригинальные методы проникновения в систему и используют технологии «стелс» и «полиморфию»;
   4. **Сетевые** – распространяются по различным компьютерным сетям. Они самостоятельно передают свой код на удалённый сервер или рабочую станцию. Обладают ещё и возможностью запустить на выполнение свой код на удалённом компьютере или по крайней мере подтолкнуть пользователя к запуску заражённого файла.
4. **По особенностям алгоритма** выделяют 7 видов вирусов:
   1. **Компаньоны (спутники)** – не изменяют файлы, а создают для исполняемых программ (\*.exe) одноимённые командные программы (\*.com), которые при выполнении исходной программы запускаются первыми, а затем передают управление исходной программе;

**04.12.2024**

* 1. **Репликаторы (черви)** – распространяются по компьютерным сетям, проникая в память компьютера, вычисляя адреса других сетевых компьютеров и рассылая по ним свои копии;
  2. **Паразиты** – при распространении своих копий изменяют содержимое файлов и секторов диска. К этой группе относятся вирусы, которые не являются спутниками и червями;
  3. **Троянские вирусы (квазивирусы)** – маскируются под какие-нибудь полезные программы и активизируются при наступлении некоторого события или условии срабатывания. Такие вирусы содержат некоторые деструктивные действия связанные с нарушением безопасности в компьютерной системе, например передаёт конфиденциальную информацию или модифицируют программы систем защиты;
  4. **Невидимки (стелс)** – перехватывают обращения ОС к поражённым файлам (секторам диска) и подставляют вместо себя незаражённые файлы (участки диска). Поэтому их очень сложно обнаружить и обезвредить;
  5. **Мутанты (призраки)** – также маскируются постоянно модифицируя себя таким образом, что не содержат одинаковых фрагментов. Такие вирусы содержат алгоритмы шифровки/расшифровки и хранят своё тело в закодированном виде, постоянно меняя параметры кодировки. Поэтому такие вирусы самые сложные в обнаружении;
  6. **Макровирусы** – заражают документы, в которых предусмотрено выполнение макрокоманд (макросов). При открытии такого документа в начале исполняются содержащиеся в нём макросы, в том числе и макровирусы. Таким образом вирус получает управление и совершает все вредные действия.

Ещё есть **студенческие вирусы** – вирусы, составленные недотёпами, что только начали изучать программирование и по приколу решили написать какой-то не очень вредный вирус, которым сами и заражались.

**Антивирусное ПО**

Специальные программы для обнаружения, уничтожения и защиты от компьютерных вирусов называются **антивирусными программами**. Современные антивирусные программы представляют собой многофункциональные продукты, содержащие в себе как профилактические возможности, так и средства лечения от вирусов и восстановления данных. Из-за того, что существует большое количество разнообразных вирусов для их быстрого и эффективного обнаружения антивирусная программа должна отвечать определённым требованиям:

* + - 1. Стабильность и надёжность работы;
      2. Объём вирусной базы;
      3. Скорость работы программы;
      4. Наличие дополнительных возможностей;
      5. Многоплатформенность.

Существует 5 групп, по которым классифицируются антивирусные программы:

**Детекторы** – обеспечивают обнаружение вирусов в оперативной памяти и на внешних носителях выдавая соответствующие сообщения, они выполняют поиск известных вирусов по сигнатуре (повторяющийся участок кода) и обнаруживать их;

**Доктора (фаги)** – не только находят заражённые вирусами файлы, но и лечат их, то есть удаляют из файлов тело вируса возвращая файлы в исходное состояние. В начале своей работы доктора ищут и уничтожают вирусы в оперативной памяти и только после этого переходят к лечению файла;

**Фильтры (сторожа)** – представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий в работе компьютера, характерных для вируса: запись в загрузочные сектора диска, прямая запись на диск по абсолютному адресу, изменение атрибутов файлов, попытка коррекции исполняемых файлов, загрузка резидентной программы;

**Ревизоры (инспекторы)** – запоминают исходное состояние программ каталогов и системных областей диска, тогда когда компьютер не заражён вирусом а затем периодически сравнивают текущее состояние с исходным, обнаруженные изменения выводятся на экран при сравнении который производятся сразу после загрузки ОС проверяются состояние загруженного сектора и таблицы размещения файлов, длина, дата и время модификации файлов, контрольная сумма файла и другие параметры. Ревизоры имеют достаточно развитые алгоритмы обнаруживают стел вирусы и могут отличить изменения версии проверяемой программы от изменений внесённых вирусов;

**Вакцинаторы (иммунизаторы)** – предотвращают заражение файлов известными вирусами, вакцина модифицирует программу или диск аким образом что бы это не отражалось на их работе, а вирус будет воспринимать их заражёнными и не внедряются в них, в настоящее время вакцины редко применяются так как имеют ограниченные возможности по предотвращению заражения от большого числа разнообразных вирусов

Kaspersky Anti-Virus

Kaspersky Internet Security

Dr. Web

AVIRA

AVG

ESET Nod32

Avast

360 Total Security

**09.12.2024**

**Методы защиты информации**

Одним из способов защиты информации является криптографическое преобразование.

**Криптографическое преобразование** – специальные математические и алгоритмические средства, с помощью которых выполняется преобразование информации в непонятный вид.

Наука, которая занимается криптографическим преобразованием информации называется **криптографией**. Криптография взята от слова «cryptos» в переводе с латинского означает «тайна». История криптографии ровесница истории человеческого языка. Она делится на 4 этапа:

1. Наивная криптография;
2. Формальная криптография;
3. Научная криптография;
4. Компьютерная криптография.

Для **наивной криптографии** (до начала XVI в.) характерно использование любых, обычно примитивных, способов запутывания противника относительно содержания шифруемых текстов. На начальном этапе для защиты информации использовались методы кодирования и стеганографии, которые родственны, но не тождественны криптографии.

**Формальная криптография** (конец 15 начало 20 вв.) связана с появлением формализованных и относительно стойких к ручному криптоанализу шифров. В европейских странах это произошло в эпоху Возрождения, когда развитие науки и торговли вызвало спрос на надежные способы защиты информации. Важная роль на этом этапе принадлежит Леону Батисте Альберти, итальянскому архитектору, который одним из первых предложил **многоалфавитную подстановку.** Данный шифр, получивший имя дипломата XVI в. **Блеза Вижинера,** состоял в последовательном "сложении" букв исходного текста с ключом (процедуру можно облегчить с помощью специальной таблицы). Его работа "Трактат о шифре" (1466 г.) считается первой научной работой по криптологии.

В **научной криптографии** (1930 – 60-е гг.) появились криптосистемы со строгим математическим обоснованием криптостойкости. К началу 30-х годов окончательно сформировались разделы математики, являющиеся научной основой криптологии:

* теория вероятностей и математическая статистика,
* общая алгебра,
* теория чисел,
* теория алгоритмов,
* теория информации,
* кибернетика.

Своеобразным водоразделом стала работа Клода Шеннона "Теория связи в секретных системах" (1949), которая подвела научную базу под криптографию и криптоанализ. Этап развития криптографии и криптоанализа до 1949 г. стали называть донаучной криптологией.

Шеннон ввел понятия "**рассеивание"** и "**перемешивание**", обосновал возможность создания сколь угодно стойких криптосистем.

В 1960-х гг. ведущие криптографические школы подошли к созданию блочных шифров, еще более стойких по сравнению с роторными криптосистемами, однако допускающих практическую реализацию только в виде цифровых электронных устройств.

**Компьютерная криптография** (с 1970-х гг.) обязана своим появлением вычислительным средствам с производительностью, достаточной для реализации криптосистем, обеспечивающих при большой скорости шифрования на несколько порядков более высокую криптостойкость, чем "**ручные**" и "**механические**" шифры.

Первым классом криптосистем, практическое применение которых стало возможно с появлением мощных и компактных вычислительных средств, стали **блочные шифры**.

В 70-е гг. был разработан американский **стандарт шифрования DES** (принят в 1978 г.). Один из его авторов, Хорст Фейстель (сотрудник IBM), описал модель блочных шифров, на основе которой были построены другие, более стойкие симметричные криптосистемы, в том числе отечественный стандарт шифрования ГОСТ 28147–89.

С появлением DES обогатился и криптоанализ, для атак на американский алгоритм был создано несколько новых видов криптоанализа (линейный, дифференциальный и т.д.), практическая реализация которых опять же была возможна только с появлением мощных вычислительных систем.

В середине 70-х гг. ХХ столетия произошел настоящий прорыв в современной криптографии – **появление асимметричных криптосистем**, которые не требовали передачи секретного ключа между сторонами.

Здесь отправной точкой принято считать работу, опубликованную **Уитфилдом Диффи** и **Мартином Хеллманом** в 1976 г. под названием "**Новые направления в современной криптографии**". В ней впервые сформулированы принципы обмена шифрованной информацией без обмена секретным ключом. Независимо к идее асимметричных криптосистем подошел Ральф Меркли. Несколькими годами позже Рон Ривест, Ади Шамир и Леонард Адлеман открыли систему RSA, первую практическую асимметричную криптосистему, стойкость которой была основана на проблеме факторизации больших простых чисел. Асимметричная криптография открыла сразу несколько новых прикладных направлений, в частности системы электронной цифровой подписи (ЭЦП) и электронных денег.

В 1980–90-е гг. появились совершенно новые направления криптографии:

* вероятностное шифрование,
* квантовая криптография и другие.

**Криптография** – наука, изучающая математические методы защиты информации, методы преобразования, обеспечивающие ее конфиденциальность и аутентичность.

**Конфиденциальность** – невозможность получения информации из преобразованного массива, без знания дополнительной информации (ключа).

**Аутентичность** – проверка подлинности.

**Криптоанализ** – объединенные математические методы нарушения конфиденциальности и аутентичности информации без знания ключей.

**Стеганография** – обеспечение скрытности передаваемой информации.

**Криптографическая стойкость** – способность системы быть устойчивой к анализу аналитических методов перебора.

**Алфавит** – конечное множество, используемых для кодирования информации знаков.

**Требования к криптосистемам**

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании. Для современных криптографических систем защиты информации сформулированы следующие общепринятые требования.

1. Знание алгоритма шифрования не должно снижать криптостойкости шифра;
2. Зашифрованное сообщение должно поддаваться чтению только при наличии ключа;
3. Шифр должен быть стойким даже в случае, если нарушителю известно достаточно большое количество исходных данных и соответствующих им зашифрованных данных;
4. Число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей, должно иметь строгую нижнюю оценку и должно либо выходить за пределы возможностей современных компьютеров, либо требовать создания и использования дорогих вычислительных систем;
5. Незначительное изменение ключа или исходного текста должно приводить к существенному изменению вида зашифрованного текста. Этому требованию не соответствуют практически все шифры донаучной криптографии;
6. Структурные элементы алгоритма шифрования должны быть неизменными;
7. Длина шифрованного текста должна быть равной длине исходного текста;
8. Дополнительные биты, вводимые в сообщение в процессе шифрования, должны быть полностью и надежно скрыты в шифрованном тексте;
9. Не должно быть простых и легко устанавливаемых зависимостей между ключами, последовательно используемыми в процессе шифрования;
10. Любой ключ из множества возможных должен обеспечивать надежную защиту информации.

-------------------------------------------------------------------------------------------------------------

Для шифрования текста методом Плейфера необходимо подготовить

Для расшифровки методом Плейфера должны выполняться следующие 4 правила:

1. Символы открытого текста разделяются на биграммы (пара символов). Если в биграмме попадаются 2 одинаковых символа, то вместо второго должен появляться символ заполнитель, а остальной … Если в конце последняя биграмма остаётся неполной, то она заполняется сомволом-заполнителем.
2. Если символы биграммы находятся на одной и той же строке, тогда вместо них берутся символы, которые находятся справа (слева) от них. Если один из символов биграммы находится в конце той же строки, что и другой символ, то вместо него берётся символ с начала этой же строки
3. Если символы биграммы находятся в одной и той же строке, то вместо них берутся символы снизу (сверху) каждого из них, если

-------------------------------------------------------------------------------------------------------------

**10.12.2024**

**Шифр Хилла**

Шифр Хилла был изобретён в **1929 году** американским математиком **Лестером Хиллом**, который основан на линейной алгебре и модульной арифметике. Для шифрования текста методом Хилла используется матрица и выполнение некоторых операций с матрицами. Для этого задаётся некоторая матрица, которая является квадратной, в ней записаны числовые обозначения символов ключа. Далее необходимо разбивать текст на группу символов, где размер каждой группы должен быть равным размерам матрицы. После этого необходимо выполнять умножение матрицы ключа с числовыми обозначениями каждой группы по модулю равному количеству символов в используемом алфавите.

Для расшифрования необходимо для заданного ключа составить матрицу и определить обратную матрицу и выполнять аналогичные операции с обратной матрицей и шифро-текстом. Надо отметить, что для нахождения обратной матрицы по шифру Хилла используется не простой способ нахождения обратной матрицы, а нахождение обратной матрицы в кольце. Это означает, что для матрицы ключа находятся алгебраические дополнения, с их помощью находятся транспонированная матрица и эта матрица умножается на обратную матрицу ключа. Полученный результат умножается с группой числовых обозначений шифро-текста и в результате получается расшифрованный текст.

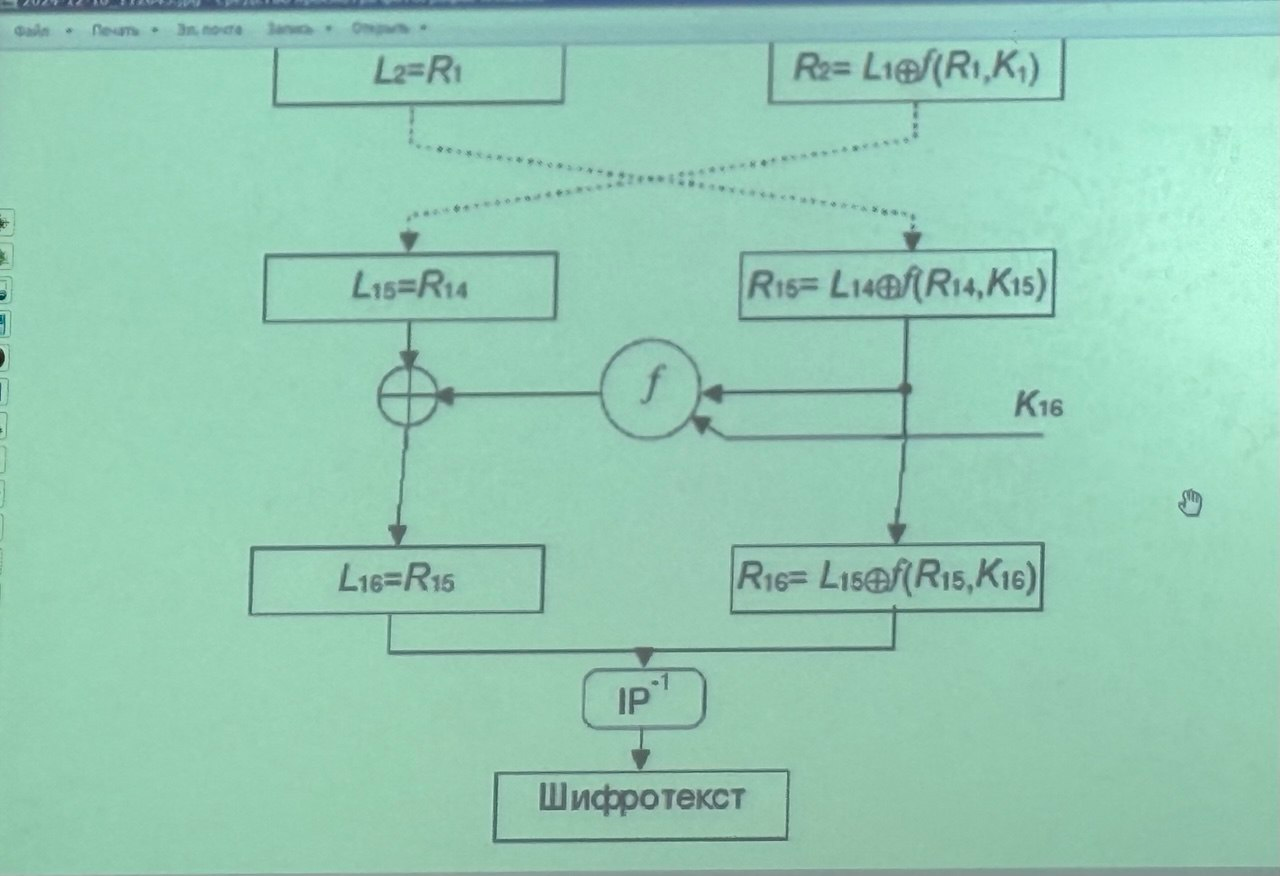
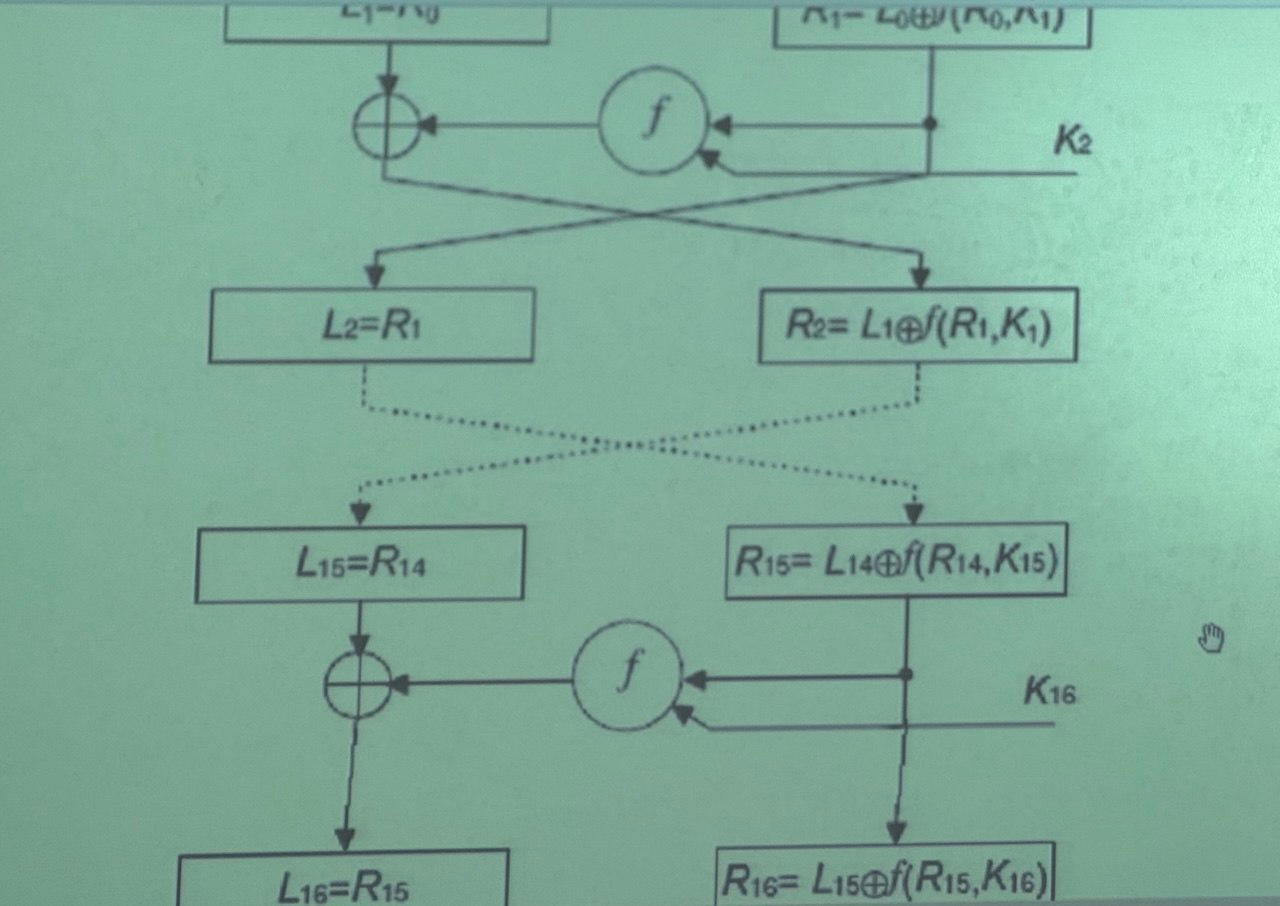
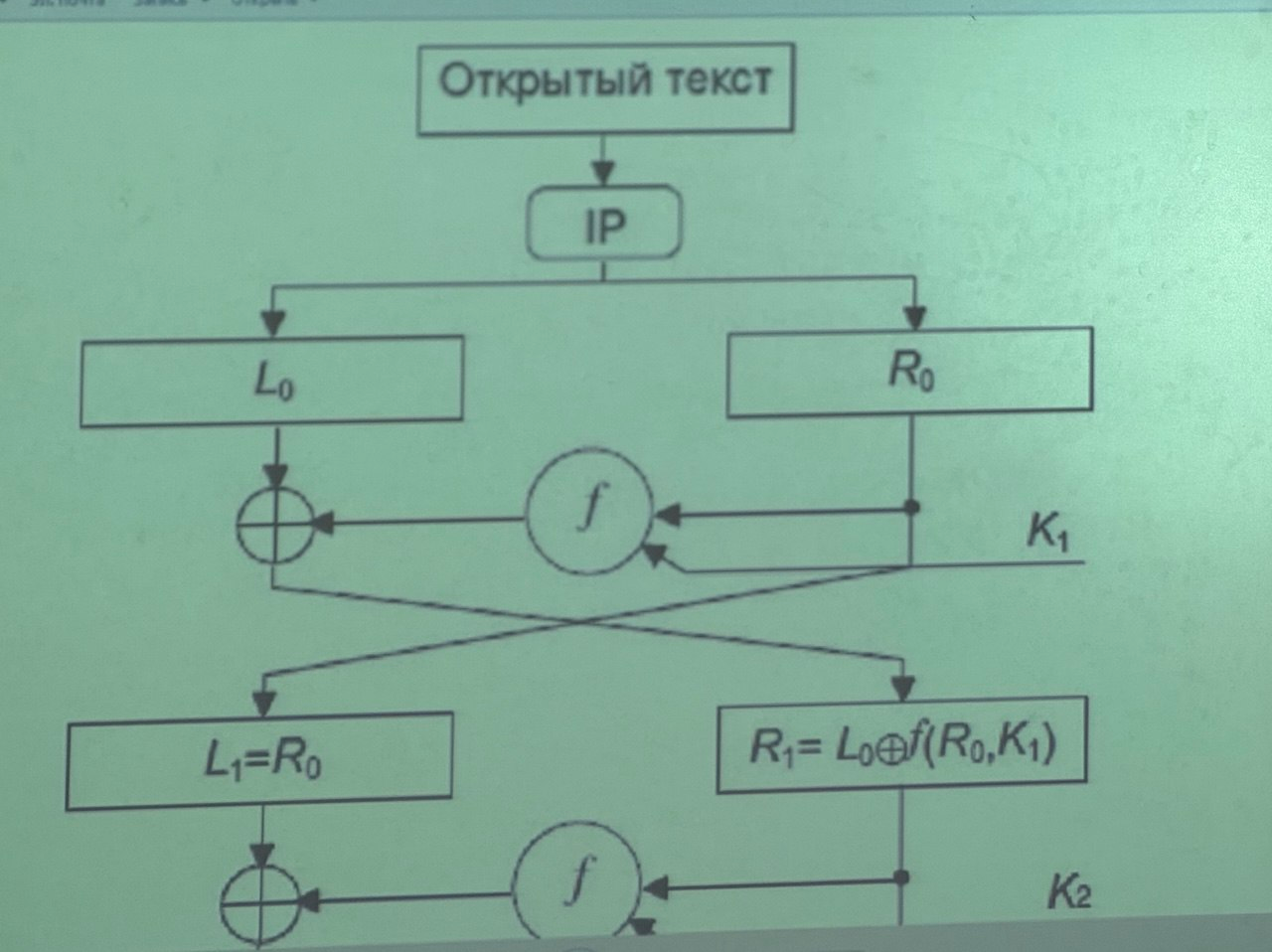
**Шифр алгебры матрицы**

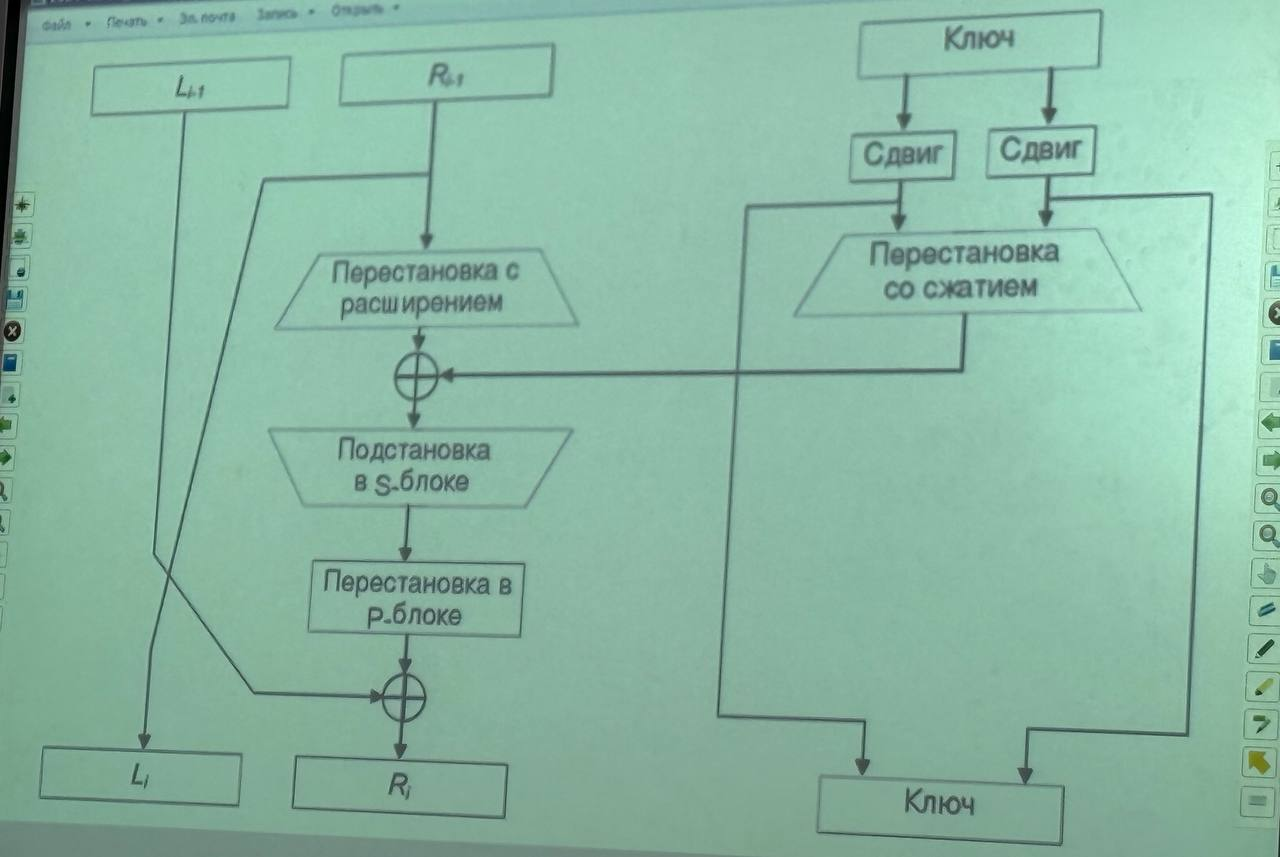
Шифр алгебры матрицы похож на шифр Хилла, только там обратная матрица ключа находится обычным образом и при выполнении операций с матрицами не используются операции деления по модулю. Имеется только ограничение в выборе ключа. Ключ должен быть подобран таким образом, что при выполнении операции не получились бы числа, выходящие за пределы числовых обозначений символов в компьютере.

**Шифр Гронсфельда**

Шифр Гронсфельда является полиалфавитным подстановочным шифром, созданным графом Гронсфельдом, который являлся руководителем первой дешифровальной службы Германии в 17 веке. Шифр Гронсфельда является усовершенствованным вариантом шифра Цезаря и Вижинера.

Идея метода заключается в следующем: для заданного текста используется операция сдвига каждого символа, равная значениям ключа, при этом длина ключа должна соответствовать количеству символов в тексте.





**13.12.2024**

**Шифр гаммирования**

Предположим, что рассматриваемый алфавит отожествлён с элементами кольца

**Ассиметричные криптосистемы**

Симметричные криптосистемы

Под симметричными криптосистемами понимается криптосистема, в которой используется один и тот же ключ для шифрования и расшифрования. Симметричные криптосистемы можно привести в виде следующей схемы:

Отправитель А

Шифратор

Исходное шифрованное

сообщение сообщение

(х) Y=

 Ключ к ------------



 Получатель В



Дешифратор

Открытое сообщение

 Х=



Ключ к

Для симметричного шифрования важно чтобы обе стороны договорились об использовании секретного ключа в процессе шифровании. Все многообразие симметричных криптосистемы основывается на следующих базовых классов: моно и многоалфавитные подстановки.

Моно алфавитные подстановки – это наиболее простой вид подстановок, заключающийся на замене символа исходного текста на другие слова того же алфавита по более или менее сложному правилу.

В случае моноалфавитной подстановки каждый символ исходного текста преобразуется в символ шифрованного текста по одному и тому же закону.

При многоалфавитном подстановки закон преобразования меняется от символа к символу. Один и тот же шифр может рассматриваться и как моно и как много алфавитный в зависимости от определяемого алфавита.

Подстановки.

Перестановки так же не сложный метод криптографического преобразования заключающиеся в перестановке метами символов исходного текста по некоторому правилу. Шифры перестановок в настоящее время не используются в чистом виде так как их криптостойкости недостаточно.

Блочные шифры.

Представляют собой семейство обратимых преобразований, блоков (частей фиксированной длины) исходного текста фактически блочный шрифт — это система подстановки на алфавите блоков.

В настоящее время блочные шифры являются наиболее распространёнными на практике. Американский, российский и стандарты некоторых других государств для шифрования относятся именно к этому классу шифров.

Гаммирование представляет собой преобразование исходного текста, при котором символы исходного текста складываются по модулю равного мощности алфавита с символами псевдослучайной последовательности, вырабатываемой по некоторому правилу.

Гаммирование нельзя целиком выделить в отдельный класс криптографических преобразований так как эта псевдослучайная последовательность может вырабатываться, например с помощью блочного шифра. В случае если последовательность является истинно случайной (то есть его значение могут быть сняты с датчика иначе говори от природы с помощью реального фактора) и каждый её фрагмент используется только один раз, то имеет место в криптосистемах с одноразовым ключом.

Рассматриваем несколько методов криптографического распознавания текста, который относится к первому или второму классу.

Самый первый классический метод защиты информации – это само письмо.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
| а | б | в | г | д | е | ё | ж | з | и |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| й | к | л | м | н | о | п | р | с | т |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| у | ф | х | ц | ч | ш | щ | ъ | ы | ь |

|  |  |  |
| --- | --- | --- |
| 31 | 32 | 33 |
| э | ю | я |

Книга

1215100401

Шифр Акбаша

Защита информации данным шифром заключается в обратно написании исходного текста.

Шифр Атбаша

исходный текст: Мы шифрует информацию.

шифрованный текст: .юицамрофин меурфиш Ым

Шифр Цезарь

3 26

c = (t+k) mod n

3 26

d = (c-k) mod n

n – количество букв в алфавите

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| a | b | c | d | e | f | g | h | i | j |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| k | l | m | n | o | p | q | r | s | t |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 21 | 22 | 23 | 24 | 25 | 26 |
| u | v | w | x | y | z |

K=3

46 % 26 = 20

x-a

table

Шифр Вижинера

computer science

word word word wor

Ключ мы задаём сами

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| a | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| b | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a |
| c | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b |
| d | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c |
| e | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d |
| g | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e |
| h | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f |
| i | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g |
| j | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h |
| k | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i |
| l | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j |
| m | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k |
| n | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l |
| o | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m |
| p | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n |
| q | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| r | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p |
| s | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q |
| t | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r |
| u | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s |
| v | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t |
| w | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u |
| x | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v |
| y | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w |
| z | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| c | o | m | p | u | t | e | r | s | c | i | e | n | c | e | Столбец (исходный текст ) |
| w | o | r | d | w | o | r | s | w | o | r | d | w | o | к | Строка (ключ) |
| y | c | d | s | q | h | v | u | o | g | z | h | j | g | v | Итог |

Шифр Плейфера

Для шифрования текстов с использованием метода Плейфера необходимо подготовить таблицу шифрования и расшифрования таблица составляется на основе ключа следующим образом:

без повторений построчно в таблице записываются символы ключа, а потом оставшиеся символы алфавита.

Например, классический шифр Плейфера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | t | u | d | e |
| n | a | b | c | f |
| g | h | i/j | k | l |
| m | o | p | q | r |
| v | w | x | y | z |

Чтобы шифровать (расшифровывать) должны соблюдаться 4 основные следующие основные правила:

**Первое правило:**

Символы открытого текста разделяются на биграммы (набор из 2-х символов). Если в биграмме попадаются два одинаковых символов, то вместо второго должно добавится символ заполнитель, а второй символ использовался с оставшимся символом исходного текста, если в конце последняя биграмма остаётся неполным то она заполняется символом заполнения

Пример:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| f | o | o | t | b | a | l | l |
| fo | ot | ba | lx | lx |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| c | a | l | l | i | n | g |
| ca | lx | li | ng |  |  |  |

**Второе правило:**

Если символы в биграмме находится на одной и той же строки, тогда вместо них берутся символы, которые находятся справа (слева) от них. Если один символ биграммы находится в конце той же строке, в которой другой символ, то вместо него берётся символ с начала этой же строки.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | t | u | d | e |
| n | a | b | c | f |
| g | h | i/j | k | l |
| m | o | p | q | r |
| v | w | x | y | z |

ac - bf

qm – ro

te – us

**Третье правило:**

Если символы биграммы находятся в одном и том же столбце, то в место каждого из них берутся символы, которые находятся с низу от каждого из них. Если символы биграммы находятся в одном столбце и один из них в конце столбца, то в место него берётся символ, находящийся в начале этого же столбца.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | t | u | d | e |
| n | a | b | c | f |
| g | h | i/j | k | l |
| m | o | p | q | r |
| v | w | x | y | z |

**Четвёртое правило:**

Если символы биграммы находятся в различных строках и столбца, то место каждого из них символы берутся следующим образом: для первого символа биграммы берётся символ, который находится на пересечении строки первого символа и столбца второго символа биграммы, а вместо второго символа берётся символ, который находится на пересечении строки второго и столбца первого символа.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | t | u | d | e |
| n | a | b | c | f |
| g | h | i/j | k | l |
| m | o | p | q | r |
| v | w | x | y | z |

Итог:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s | t | u | d | e |
| n | a | b | c | f |
| g | h | i/j | k | l |
| m | o | p | q | r |
| v | w | x | y | z |

Ответ:

aqoqdufz\_dnluafuz

Шифр в вертикальной перестановке

Для шифра перестановки, как и в других шифров составляется таблица на основе ключа. В таблице символы ключа добавляются без повторения.

Мы будем шифровать текст методом вертикальной перестановки

student

4 5 6 1 2 3

с т у д е н

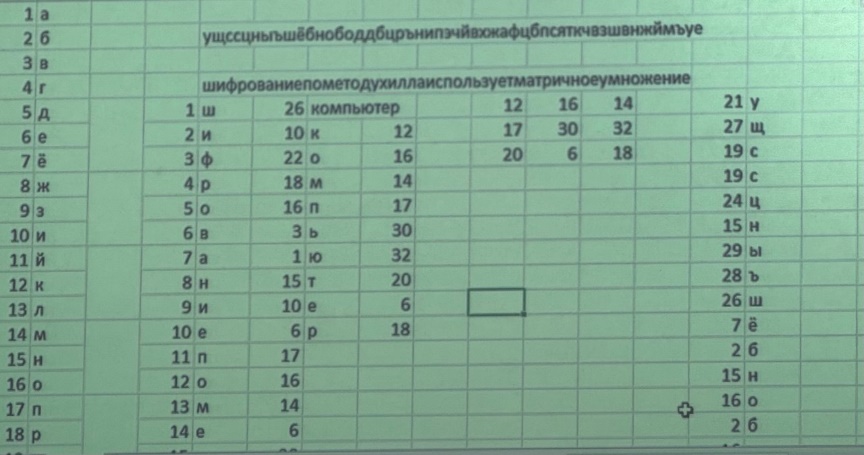
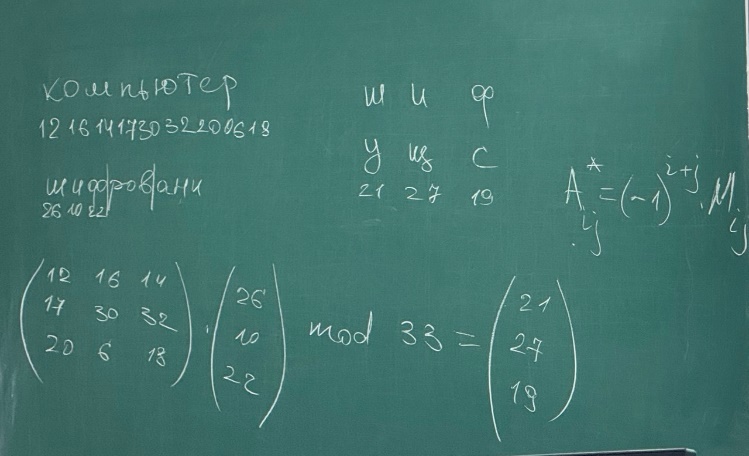
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| м | ы | \_ | б | у | д |
| е | м | \_ | ш | и | ф |
| р | о | в | а | т | ь |
| \_ | т | е | к | с | т |
| \_ | м | е | т | о | д |
| о | м | \_ | в | е | р |
| т | и | к | а | л | ь |
| н | о | й | \_ | п | е |
| р | е | с | т | а | н |
| о | в | к | и | . | \_ |

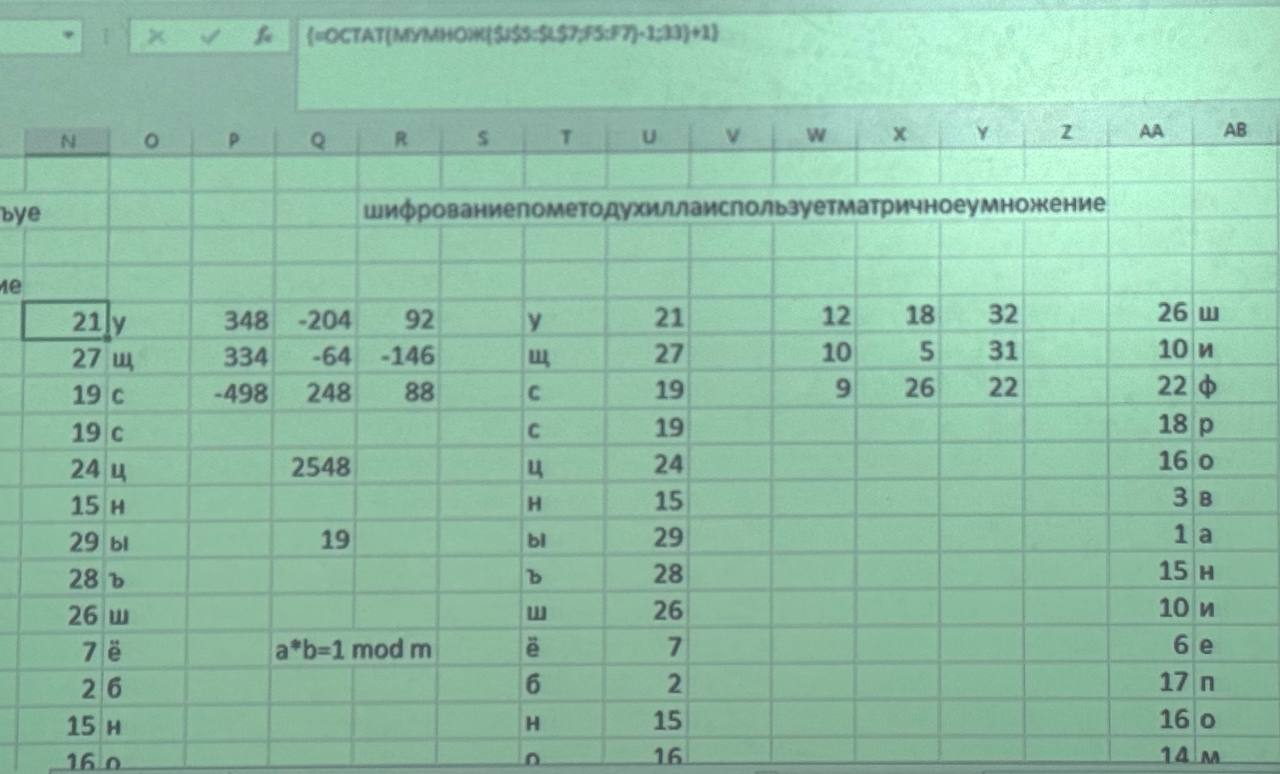
Чтобы получить шифр текст необходимо по порядке следования символов ключа в алфавите выполнять прочтения текста из таблицы по столбцам.

бшактва\_тиуитсоелпа.дфьтдрьен\_мер\_\_отнроымотммиоев\_\_вее\_кйск

**Шифр Хила**

был изобретён 1529 году американским математиком Лестером Хилом который основан на линейной алгебре и модульной арифметике. Для шифрования текстов методом Хила используется матрица и выполнение некоторых операций с матрицами для этого задаётся некая матрица которая является квадратной и в нём записаны числовые обозначения символов ключа, далее необходимо разбивать текст на группу символов где размер каждой группы должен быть равным размеру матрицы после этого необходимо выполнять умножение матрицы ключа с числовыми обозначениями каждой группы по модулю равное количествам символов в используемом алфавите. Для расшифрования необходимо для заданного ключа составить матрицу и определить обратную матрицу и выполнять аналогичные операции с обратной матрицей и шифро-текстом. Надо отметить что для нахождения обратной матрицы по шифру Хила используется не простой способ нахождения обратной матрицы а нахождение обратной матрицы в кольце — это означает что для матрицы ключа находятся алгебраические дополнения с их помощью составляется транспонированная матрица и эта матрица умножается на обратную матрицу ключа, полученный результат умножается с группами числовых обозначений шифра-текста и в результате получается расшифрованный текст



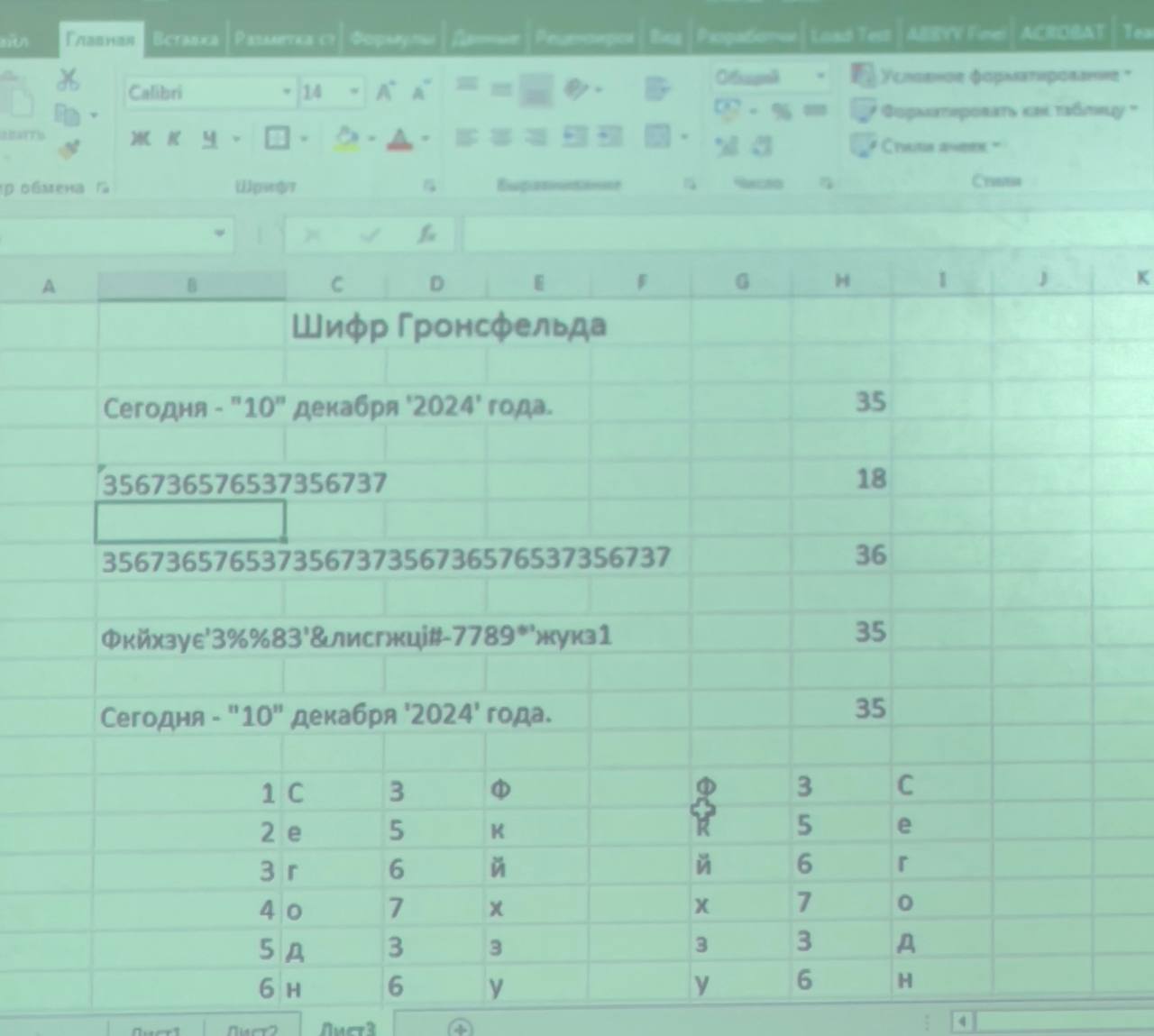


**Шифр Алгебры матрицы**

похож на шрифт Хила только в нём обратная матрица ключа находится обычным образом т при выполнении операции с матрицами не используются операции деления по модулю, имеется только ограничение в выборе ключа. Ключ должен быть подобран таким образом что при выполнении операции не получились числа выходящие за пределы числовых обозначений символов в компьютере

**Шифр Гронсфельда**

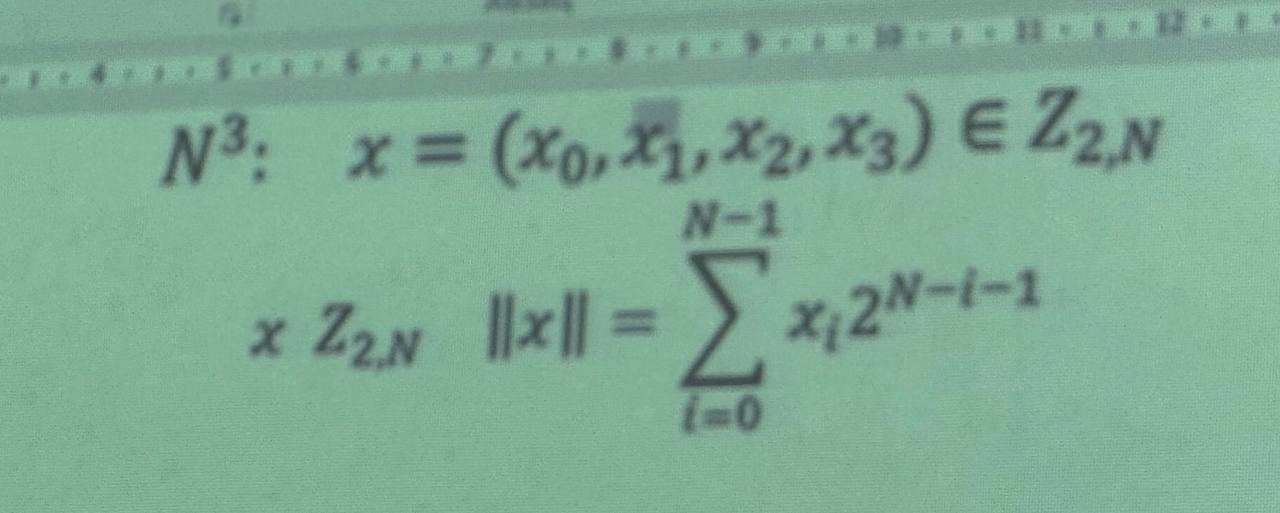
шифр Гронсфельда является поли алфавитным постановочным шифром созданным графом Гронсфельдом, которым являлся руководителем первой дешифровальной службы в Германии в 17 веке. Шифр Гронсфельда является усовершенствованным вариантом шифра Цезаря и Вижинера. Идея метода заключается в следующем. Для заданного текста используется операция сдвига равное значением ключа, при это длина ключа должно соответствовать количеством символов в тексте



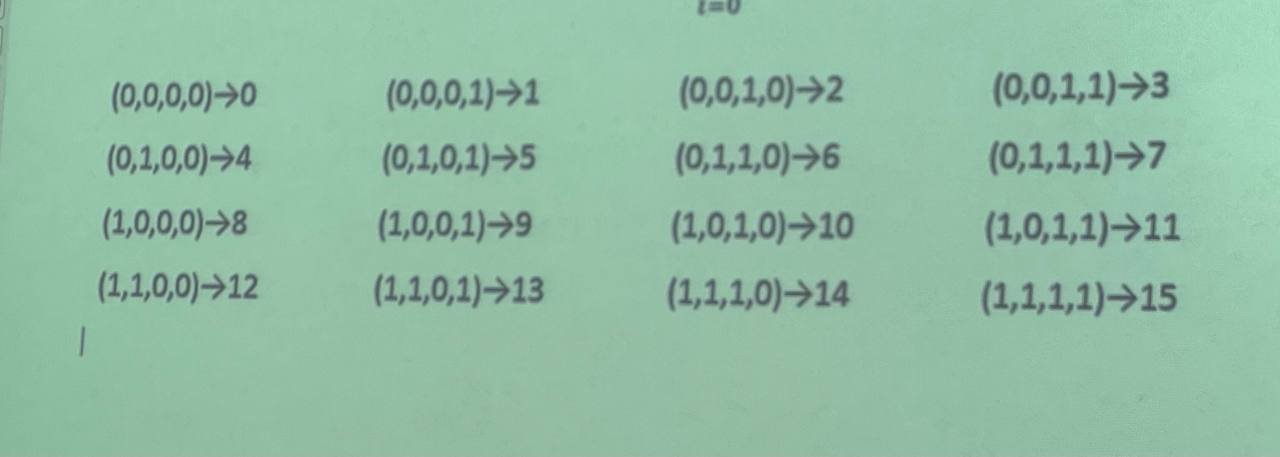
**Шифр DES**

Американский стандарт криптографического закрытия данных DES (Data Encryption Standart) принятый в 1978 году является типичным представителем семейства блочных шифров. Этот шифр допускает эффективную аппаратную и программную реализацию причём возможно достижение скоростей шифрования до нескольких мегабайт в секунду. Перед тем как ознакомится с алгоритмом DES приведём несколько основных понятий:

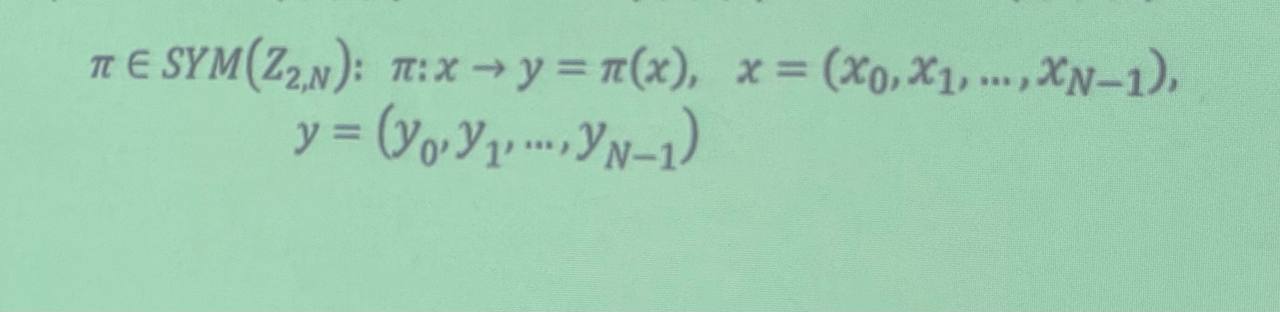
1. Под n-разрядным блоком понимается последовательность из 0 и 1 длины N3:x = (x0, x1, x2, x3) принадлежит Z2,N. X в пространстве Z2,N можно интерпретировать как вектор и как двоичное представление целого числа



Например N=4



Блочным шифром называют элемент



Так как блочные шифры являются частными случаями постановок их следует рассматривать особа

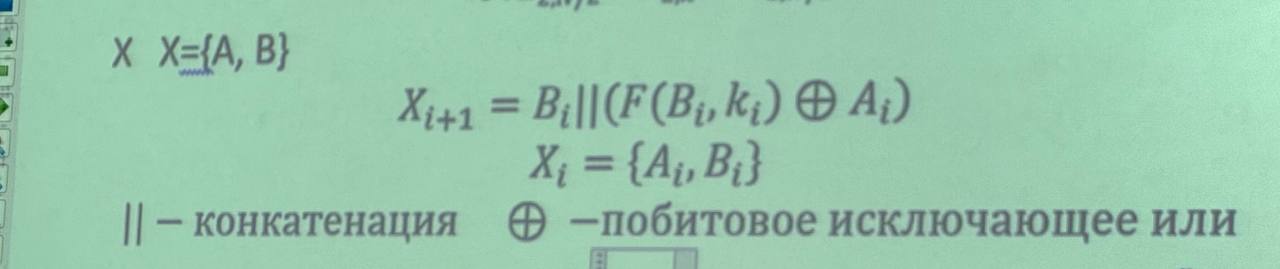
поскольку: во-первых большинство симметричных шифров используемых в системах передачи информации являются блочными и во вторых блочные шифры удобнее всего описывать в алгоритмическом виде а никак обычные постановки. Одним из наиболее распространённых способов задания блочных шифров является использование так называемых сетей

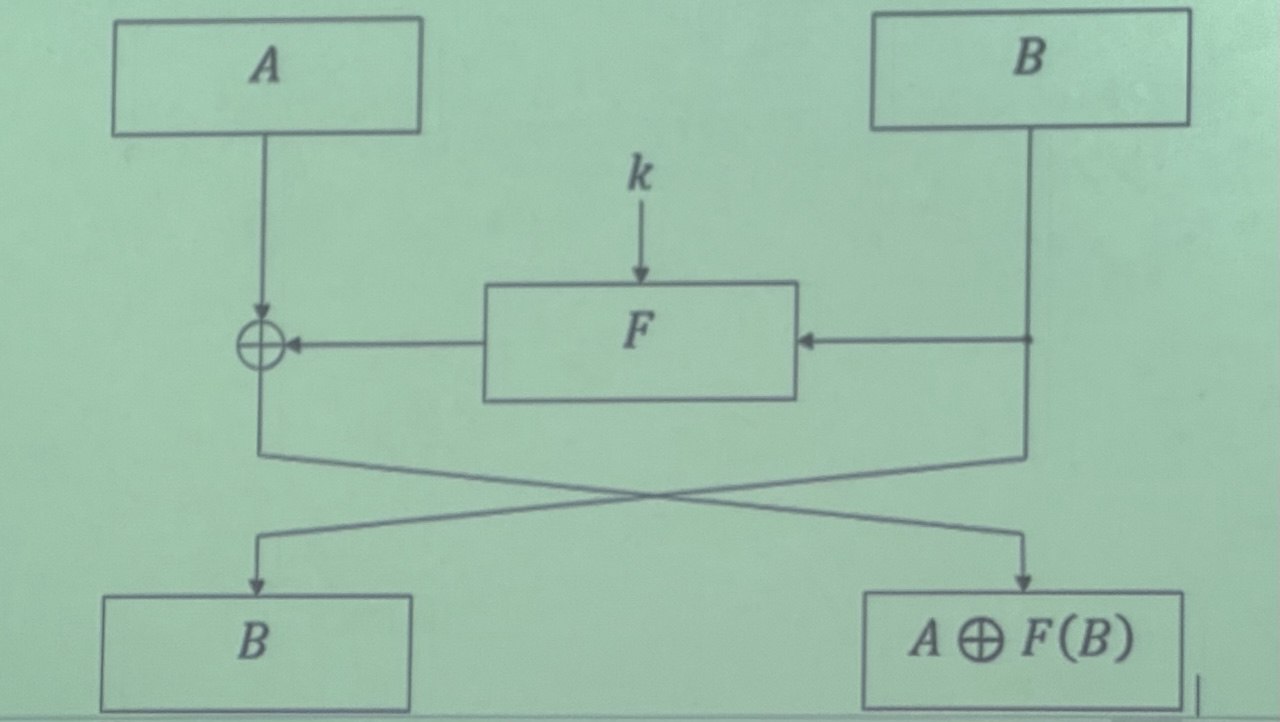
Фейстела. Сеть фейстала представляет собой общий метод преобразования произвольной функции F в перестановку на множестве блоков. Это конструкция была изобретена Хорстом Фестелом и была использована в большом количестве шифров включая DES и ГОСТ 28147-89. Функция Фестела представляющая собой основной строительный блок сети Фейстела всегда выбирается нелинейной и практически во всех случаях необратимой, формально F функцию можно представить в виде отображения



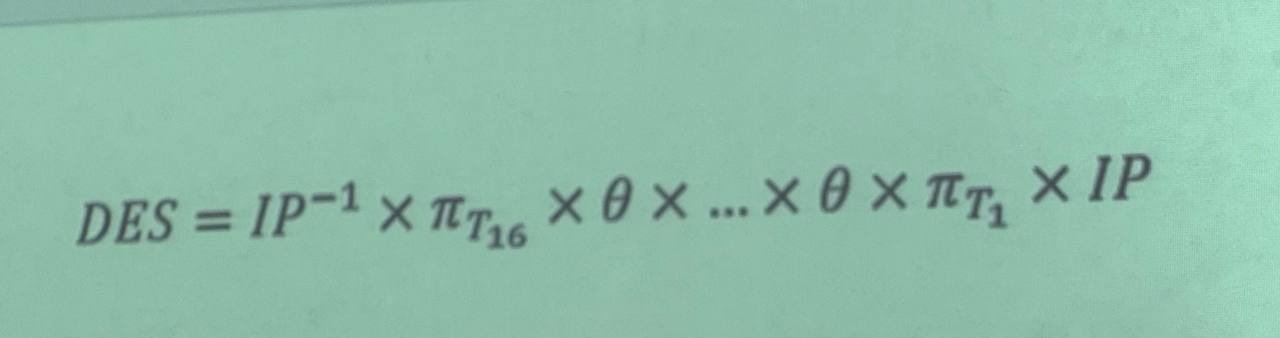
где N — длина преобразуемого блока текста(всегда чётно), к — длина использования блока ключевой информации

предположим что: пусть Х является блоком текста и мы его будем представить в виде 2х подблоков одинаковой дины следующего вида X={A, B}, тогда одна итерация или раунд сети Фейстела определяется как

Структура сети Фейстела имеет следующий вид

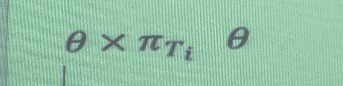


Шифр DES — представляет собой результат 33 отображений, которых можно описывать следующим образом

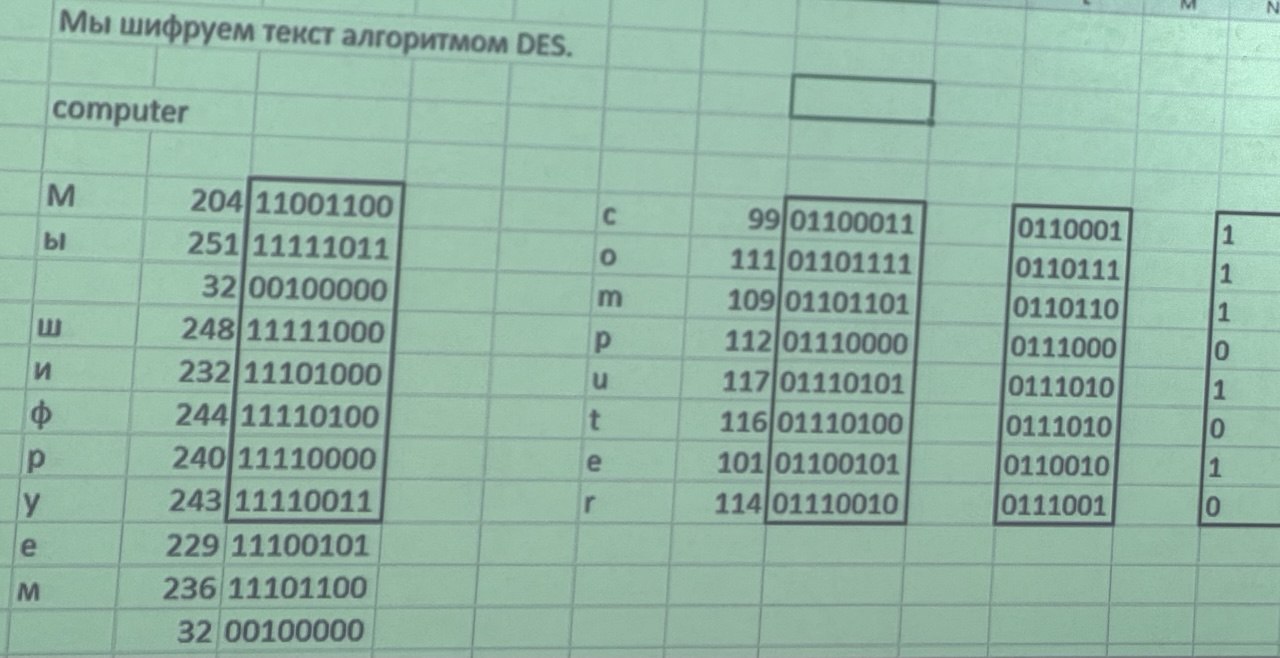


Здесь IP — это сокращение слов initial permutation (исходная перестановка) представляет собой проволочную коммутацию с инверсией IP-1

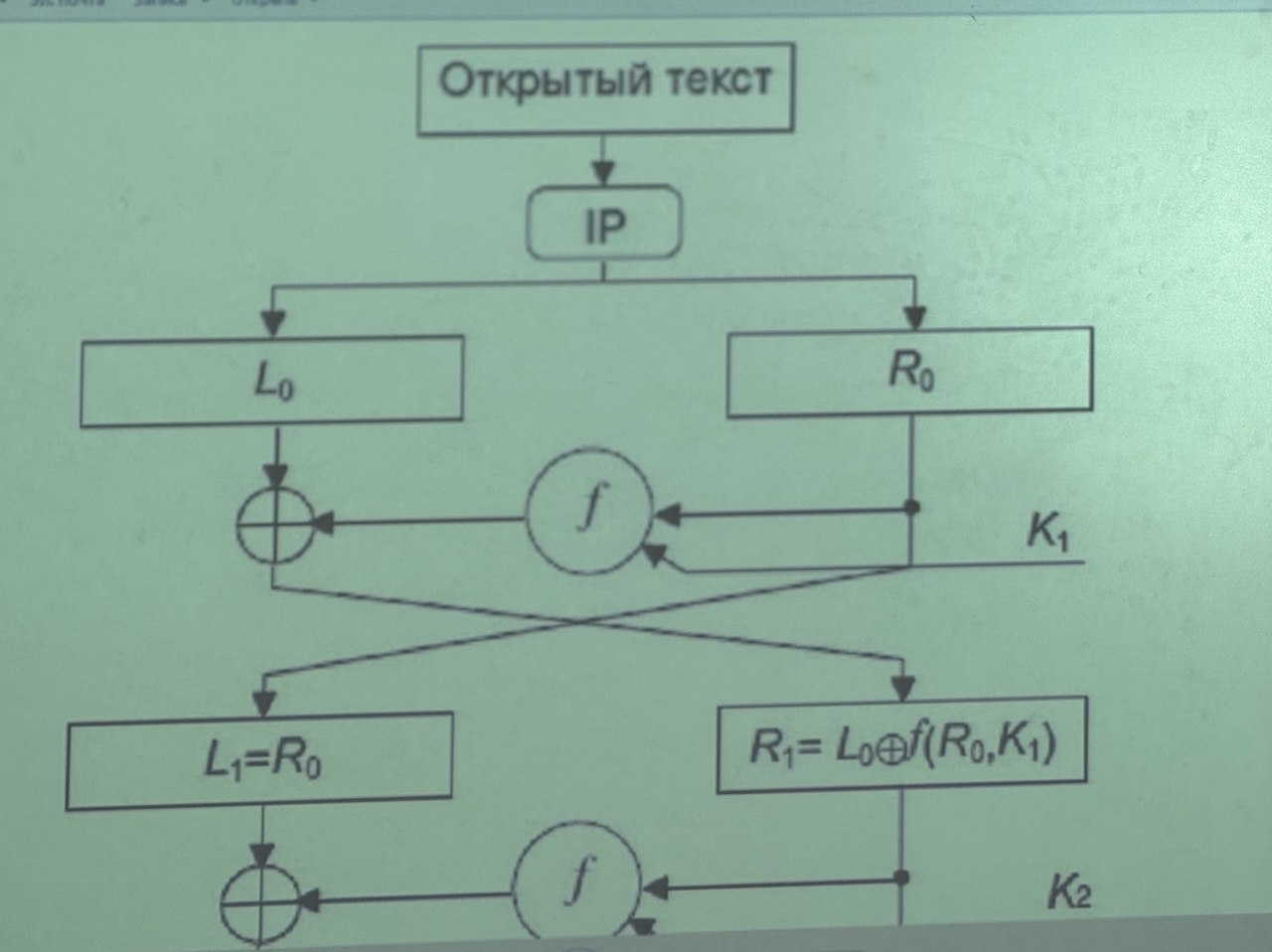
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 | 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |



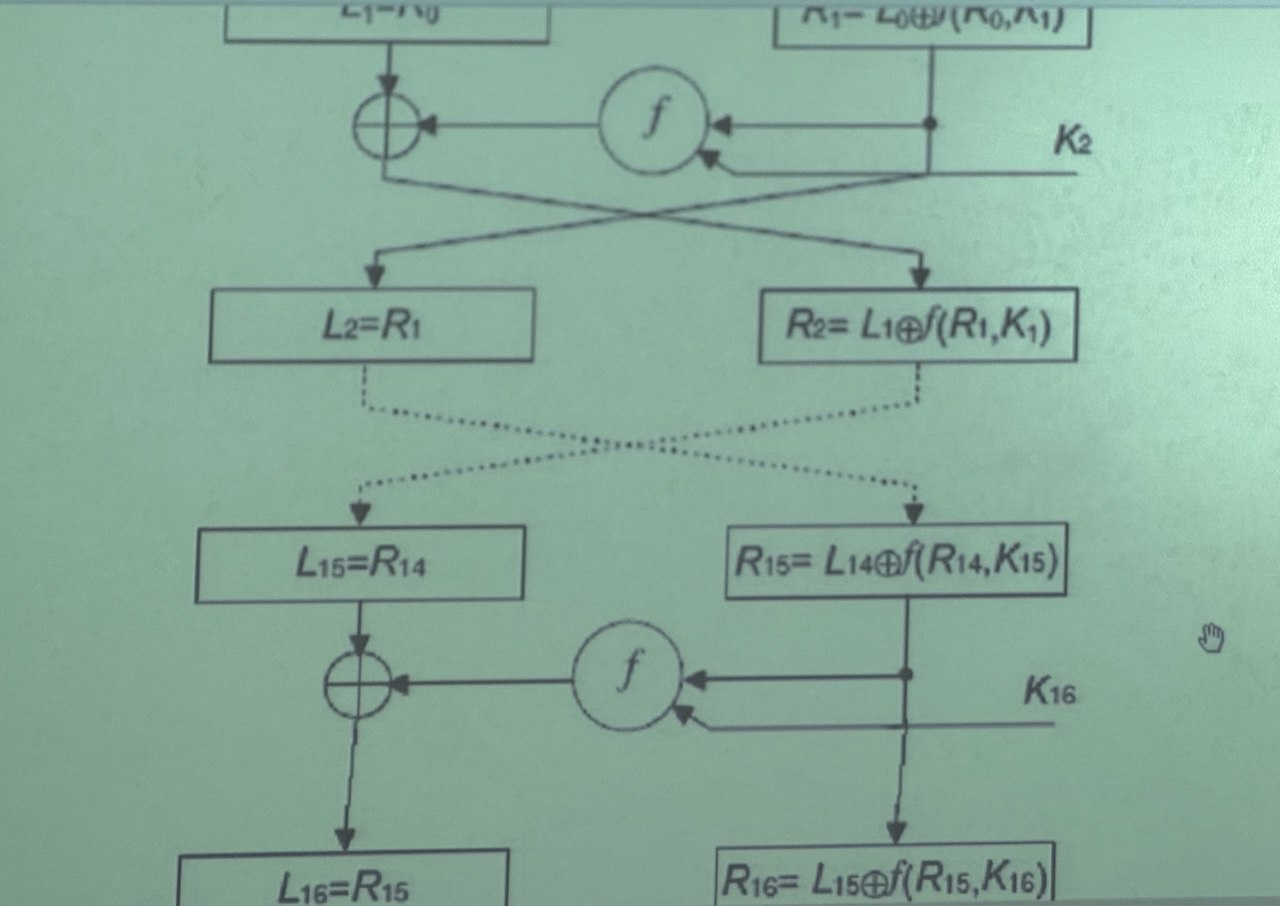
Композиция, где тета является перестановкой местами левой и правой половин блока данных представляет одну итерацию фейстела. Надо отметить что в последнем цикле шифрования по алгоритму DES перестановка местами половин блока не производится. Данные в алгоритмом DES шифруются с разбиением на блоков размерностью 64 бита. При шифровании каждого блока из 64-х бит в результате тоже выходит блок шифротекста из 64 битов. Ключ шифрования тожее задаётся в виде блока из 64-х бит, но из него используется блок из 56 битов, в котором каждый 8 бит используется для проверки чётности и игнорируется

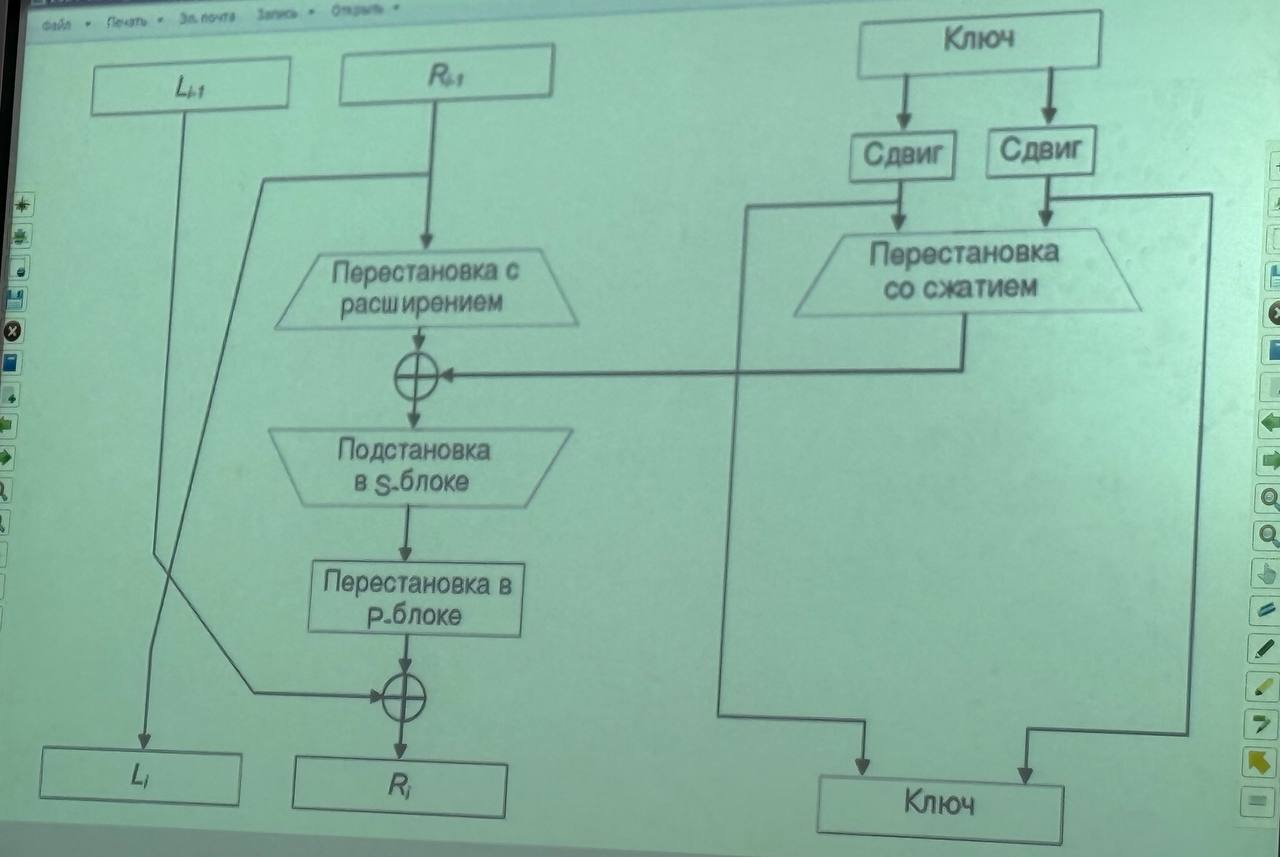


Цикл — один раз использовать алгоритм

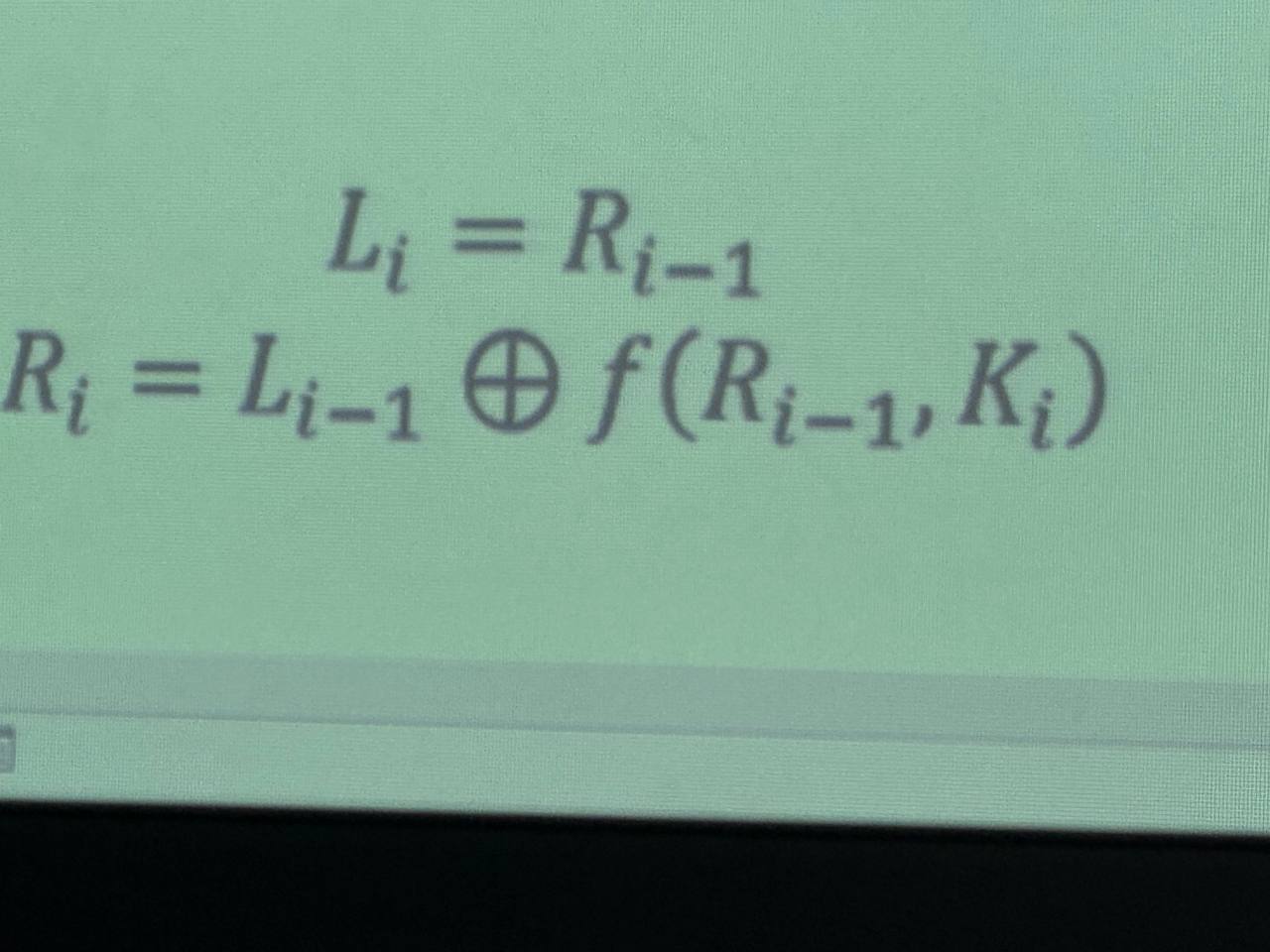


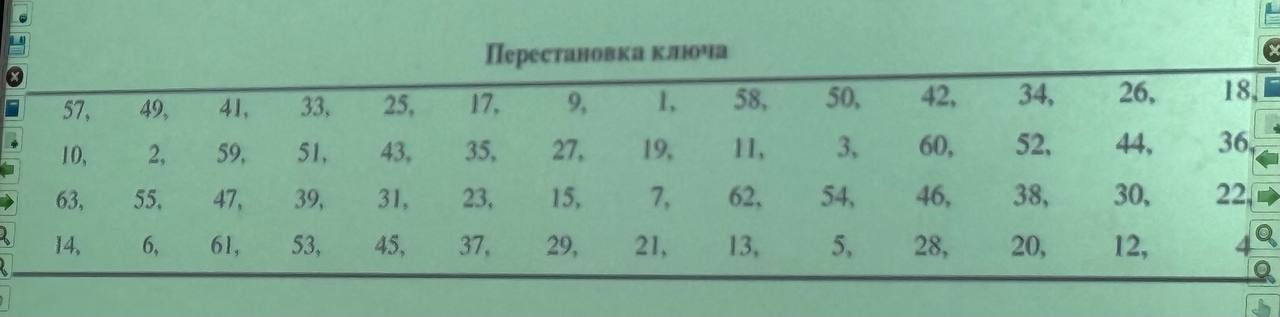
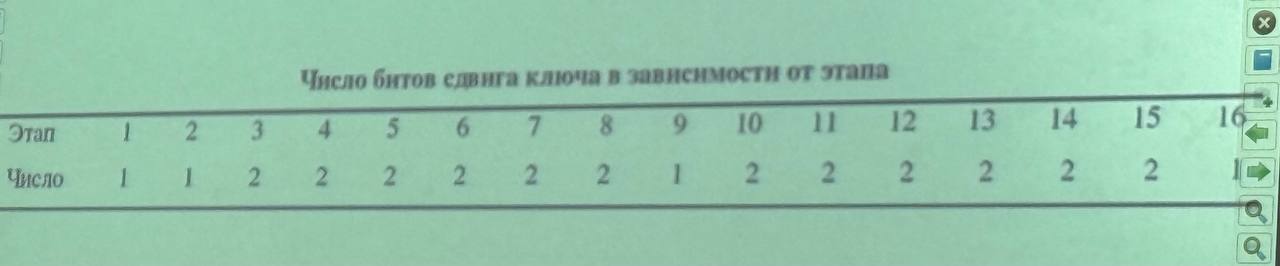


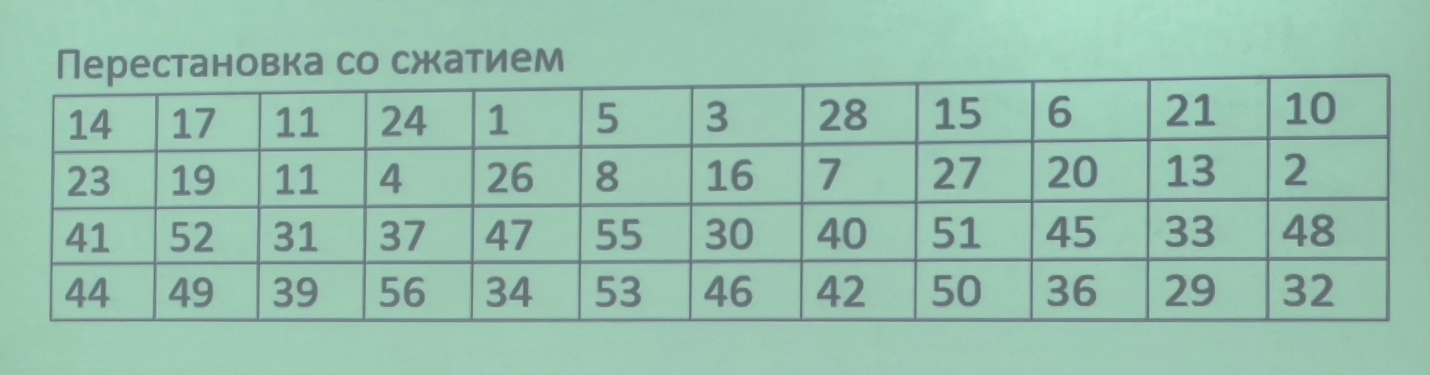




После первоначальной перестановки(IP) блок разбивается на левую и правую половины длиной по 32 бита, затем выполняется 16 этапов одинаковых действий называемых функцией F, в которых данные объединяются с ключом. Послее 16 этапа левая и правая половины объединяются и алгоритм завершается заключительной перестановкой в которой является обратной по отношению к первоначальной. На каждом этапе биты ключа сдвигаются и затем из 56 битов ключа выбираеется 48 битов, правая половина данных увеличивается до 48 битов с помощьюп ерестановки с расширением, объединяется по средством операции XOR с 48 битами смещённого и переставленного ключа проходит через 8 s блоков образуя 32 новых бита переставляется снова, эти 4 операции выполняются функцией F. Затем результат функции F объединяется с левой половиной с помощью другого XOR в итоге этих действий появляется новая правая половина, а старая правая половина становится новой половиной, эти действия повторяются 16 раз образуя 16 этапов(циклов или итераций) DES перестановки с расширением, объединяется по средством операции XOR с 48 битами смещённого и переставленного ключа проходит через 8 s блоков образуя 32 новых бита переставляется снова, эти 4 операции выполняются функцией F. Затем результат функции F объединяется с левой половиной с помощью другого XOR в итоге этих действий появляется новая правая половина, а старая правая половина становится новой половиной, эти действия повторяются 16 раз образуя 16 этапов(циклов или итераций) DES

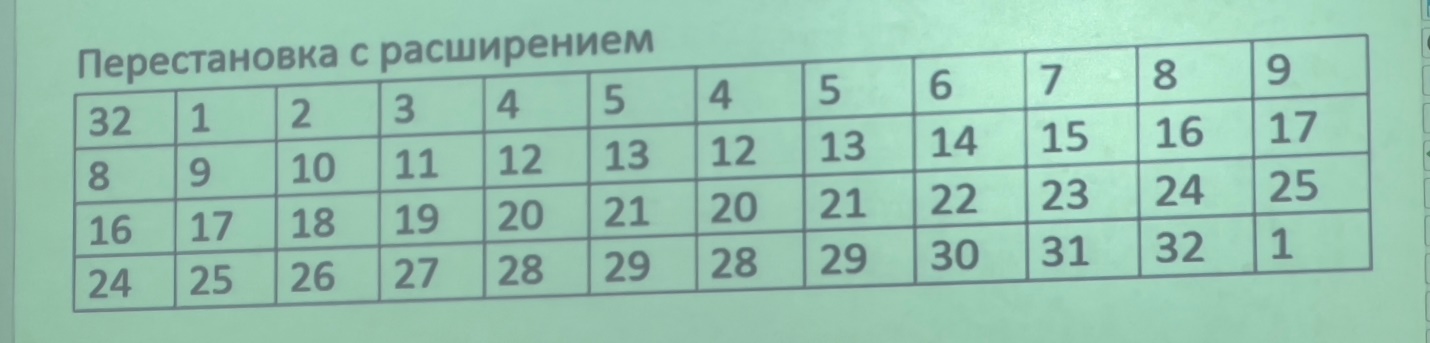
Преобразование ключа: сначала 64-х битовый ключ уменьшается до 56 ключа путём отбрасывания каждого 8 бита, после извлечеения 56 битового ключа для каждого из 16 этапов генерируется новый 48 битовый подключ. Эти подключи. Эти подключи определяются следующим образом. Во-первых 56 битовый ключ делится на 2 28-ти битовых потовинок, затем половинки циклически сдвигаются на лево на 1 или 2 бита в зависимости от этапов

После сдвига выбирается 48 из 56 битов так как при этом не только выбирается подмножество битов но и изменяется их порядок это операция называется перестановкой со сжатием и результатом является набор из 48 битов.

Из за сдвига используется отличное подмножество бита подключа .каждый бит используется приблизительно 14 из 16 подключей хотя не все биты используется в точности одинаковые число раз.

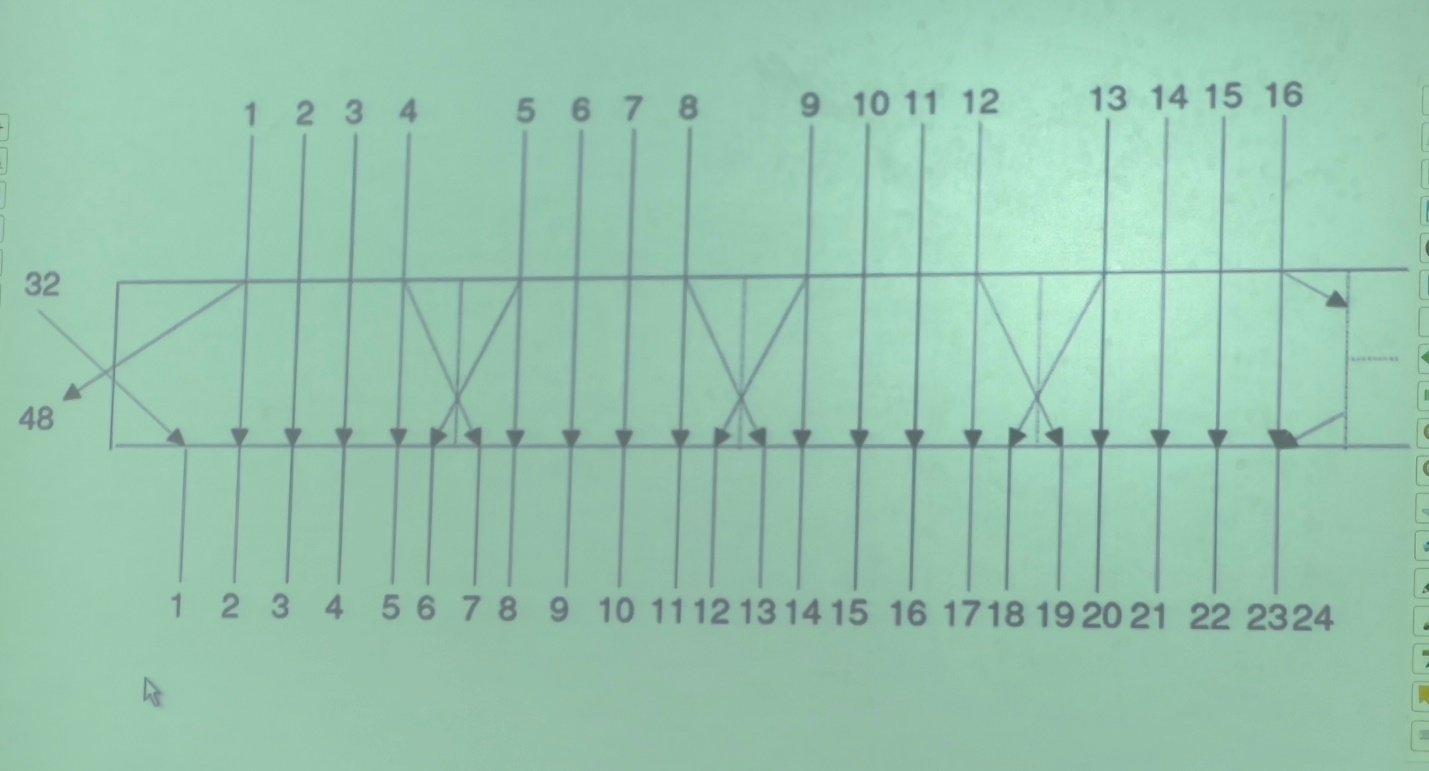
Перестановка с расширеениеем

Это операция расширяет правую половину данных Ri от 32 до 48 битов так как при этом не просто повторяются определенные биты но и изменяется их порядок . это операция называется перестановкой с расширениям.



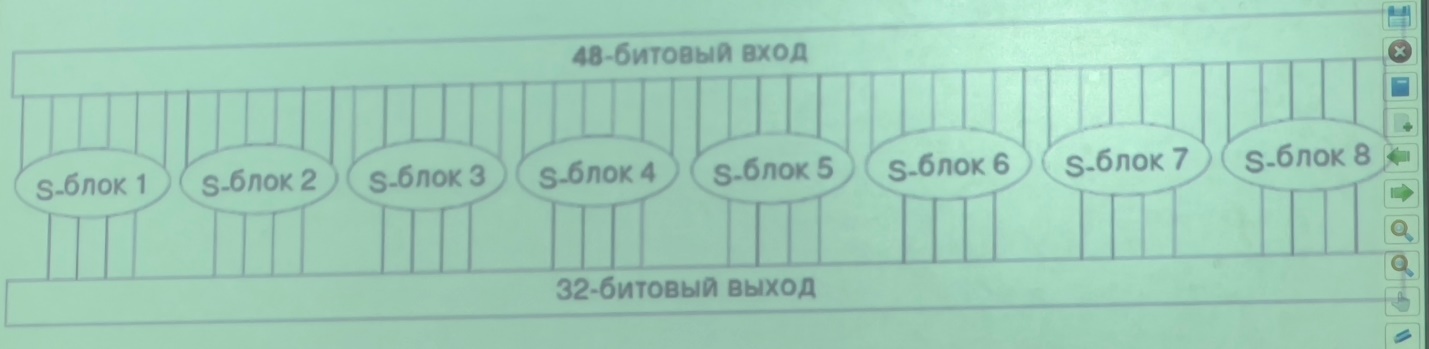
У этой операции две задачи:

Привести размер правой половины в соответсвии с ключом для операции Х OR(1 задача) и получить более длинный результат который можно будет сжать входе операции подставки (2 задача). Однако главный криптографический смысл совсем в другом. За счет влияния одного бита на две подстановки быстрее возрастает зависимость битов результата от битов исходных данных. Это называется лавинным эффектом. Алгоритм Дес спроектирован так чтобы как можно быстрее добиться зависимости каждого бита шифротекста от каждого бита открытого текста и каждого бита ключа. Перестановку с расширениям иногда называют Е-блоком (Expansion) то есть для каждого 4х битого входного блока первой и четвертый бит представляют собой два бита выходного блока а второй и третий один бит выходного блока.

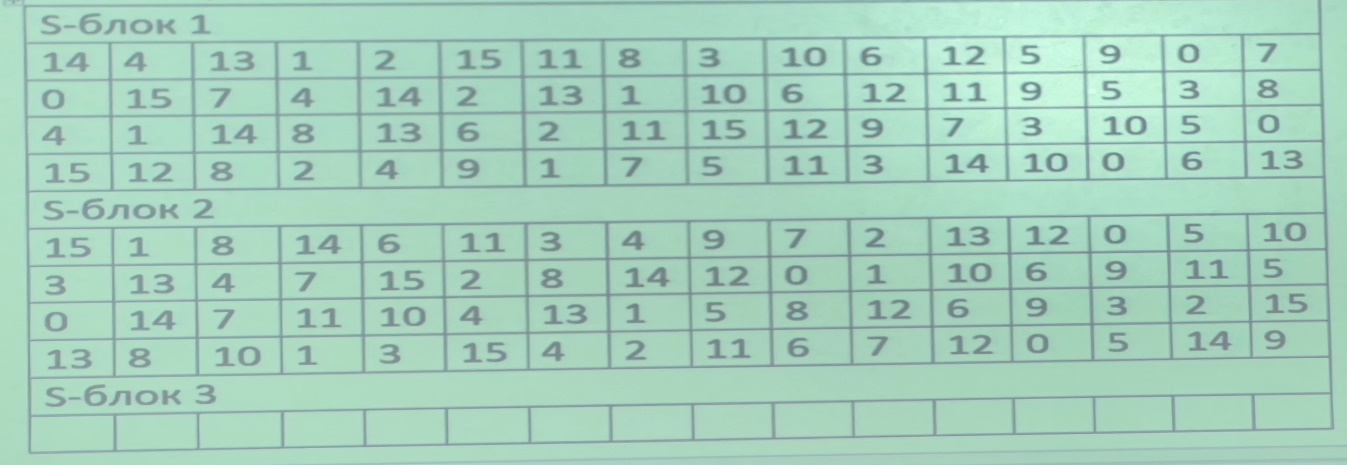
Хотя выходной блок больше входного каждый входной блок генерирует уникальный выходной блок.

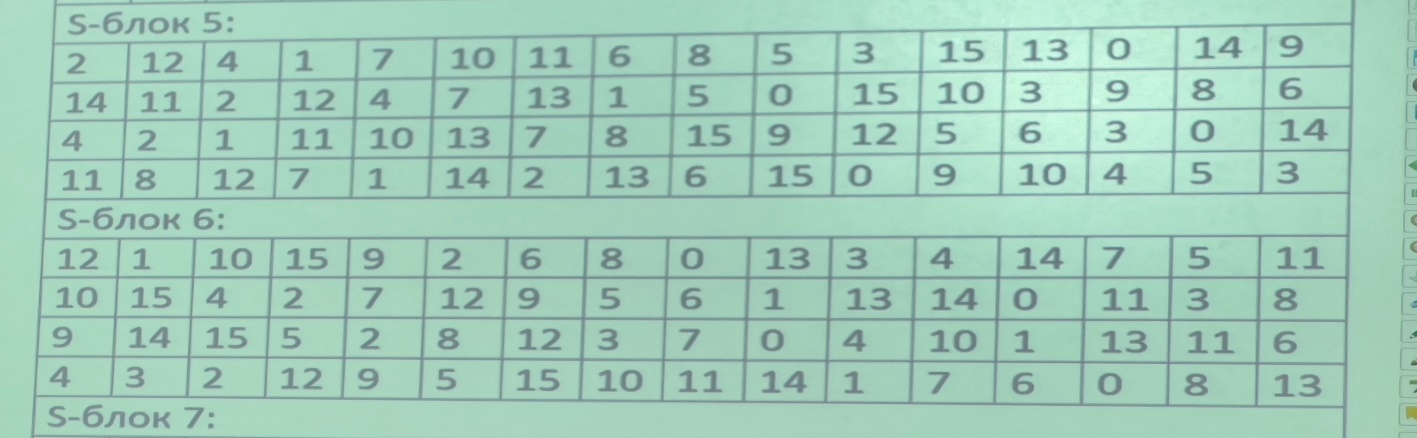
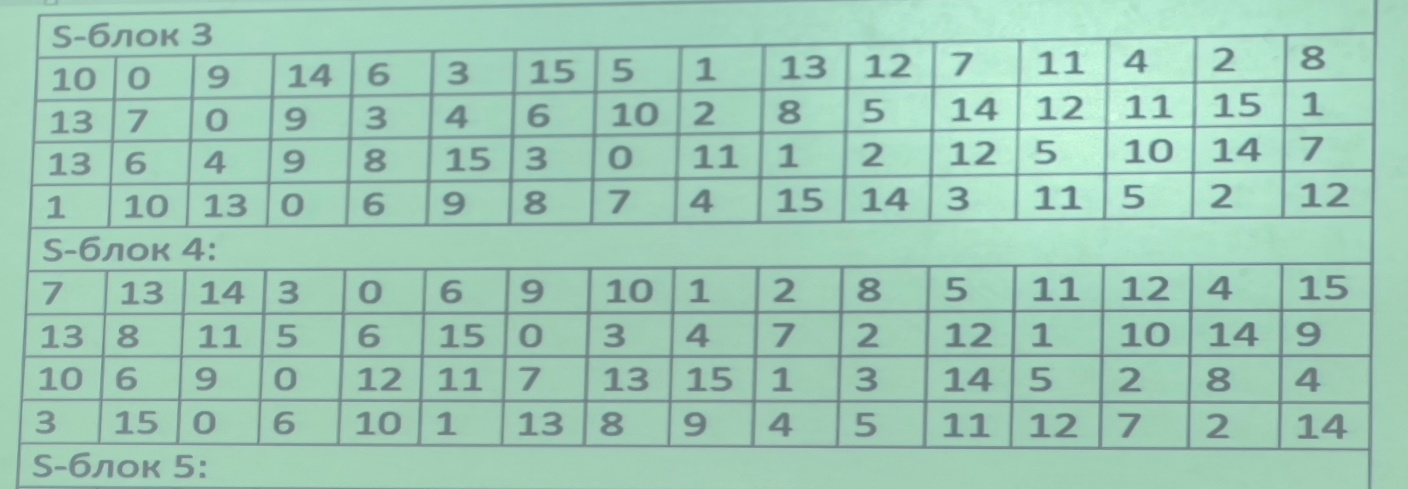
Подстановка с помощью S-блоков(Substitution)

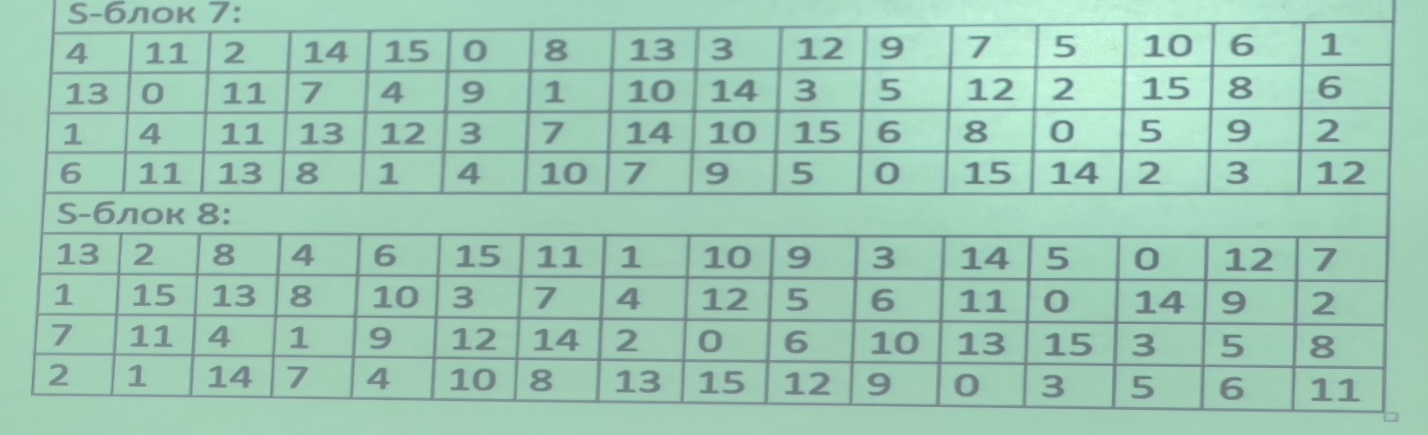
После объединения сжатого блока с расширенным блоком с помощью Х блок над 48 битовом результатом выполняется операция подстановки. Подстановки выполняются в 8 блоках под названием блоки подстановки или S блоки. У каждого S блока 6х битовой вход и 4х битовый выход всего используется 8 6ти битовых подблока. Для 8ми S блоков алгоритм Дес потребуется 256байтов памяти. 48 битов делятся на 8 6ти битовых подблока. Каждый отдельный подблок обрабатывается отдельным S блоком.



Каждый S блок представляет собой таблицу из 2х строк и 16 столбьцов. Каждый элемент в блоке является 4х битовым числом по 6 входным битом S блока определяется под какими номерами столбьцов и строк искать выходной значения.







Входные биты особым образом определяет элемент S блока. Расмотрим 6ти битовый вход S блока. b1, b2, b3, b4, b5, b6. Биты b1 и b6 объединяется образуя 2х битовой число от 0 до 3 . Биты от b2 до b5 объединяются образуя 4х битовый число от 0 до 15. Значения 2х битового числа соответствует строке таблице а значения 4х битового числа столбьца таблицы. Например предположим что допустим для 6 S блока попадаются 110011

11 =3 0..3

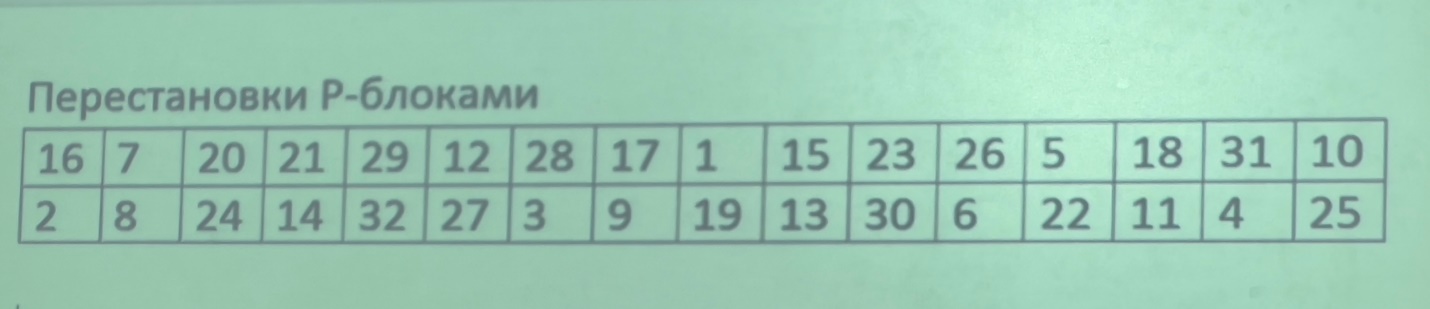
1001 =9 0..15

14

Подстановка с помощью S блоком является ключевым этапом Дес а другие действия алгоритма линейны и легко поддаются анализу. S блоки не линейны и именно они больше степени чем все остальные обеспечивает безопасность Дес. В результате этого этапа или операции подстановки получаются 8 4х битовых блоков которые вновь объединяются в единый 32х битовый блок. Этот блок поступает на вход следующего этапа под названием перестановки с помощью П-блоков

Перестановки P-блоками

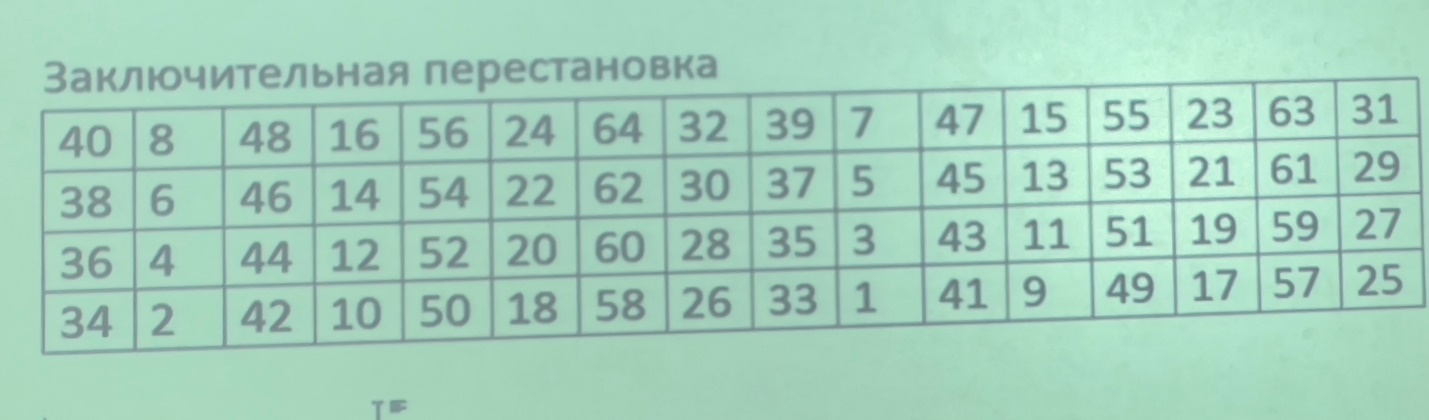
32х битовый выход подстановки с помощью S блоков перетасовывают соответствии с P блоком. Это перестановка перемещает каждый входной бит в другую позицию. Не один бит не используется дважды и не один бит не игнорируется. Этот процесс называется прямой перестановкой или просто перестановкой и выполняется с помощью след таблицы.



Наконец перестановка с помощью P блока объединяется посредством с Х OR с левый половины первоначального 64х битового блока. Затем левая и правая половины меняются местами и начинаются следующий этап. Это операция будет выполняться 16 раз.

Заключительная перестановка

Заключительная перестановка является обратной по отношению к начальной перестановки . Перед его выполнением левые и правые блоки после 16го этапа не меняются местами, объединяются в один единой 64х битовой блок и используется на вход заключительной перестановки. Заключительная перестановка выполняется по след таблицы.



Шифр Гамирования

Предположим что рассматриваемый алгоритм отождествлён с элементами кольца Zm выберем некоторую последовательность ɣ которая состоит из ɣ = ɣ1ɣ2 … ɣl . Также предположим что длина этой последовательности больше или равна длины открытого текста. Шифрование некоторого открытого текста заданной в виде последовательности x = x1x2 … xn (n ≤ l). Шифрование открытого текста х выполняется по правилам yi = xi + ɣi (i = 1, 2, …, n), здесь операция сложения выполняется по модулю m. Такой шифр называется шифром гамирования, последовательность ɣ называется гаммой, а операция поэлементного сложения символов с символами гаммы называется наложением гаммы. Дешифрование в данном случае выполняется по правилам yi = xi - ɣi (i = 1, 2, …, n), здесь операция вычитания выполняется по модулю m. Надо отметить что шифры цезаря и вижирования являются частными случаями шифрами гамирования с постоянной или периодической гаммой. Для использования шифра гамирования больше всего используют некоторую псевдо случайную последовательность(ПСП), которая вырабатывается по некоторому правилу. Шифр гамирования нельзя целиком выделить в отдельный класс криптографических преобразований, так как ПСП может вырабатываться с помощью блочного шифра. Если последовательность является истинно случайной и каждый её фрагмент используется только один раз то, данный случай относиться к криптосистемой с одноразовым ключом. Для генерации ПСП или гаммы можно использовать несколько различных способов, которыми являются:

1. Специально составленная и откорректированная на случайность таблица
2. Физический датчик
3. Формирования ПСП с помощью программы
4. программно аппаратный датчик ПСП на регистрах сдвига

Асимметричные криптосистемы

Как уже известно в асимметричных криптосистемах используется 2 ключа:

1. открытый используется для шифрования
2. закрытый используется для деешифрования

Примерами шифров асимметричных криптосистем являются криптосистемы Эль-Гамаля, криптосистема основанная на проблеме Диффи-Хеллмана, RSA (Ривеста-Шамира-Адлемана), криптосистема Меркля-Хеллмана и Хора-Ривеста, криптосистема основанная на эллиптических кривых

Алгоритм RSA

криптосистема RSA представляет собой криптосистему стойкость которой основана на сложности решения задачи разложения числа на простые сомножители. Алгоритм был разработан ы 1977 году, разработчиками алгоритма являются Рон Ривес, Ади Шамир и Лен Адлеман. Алгоритм RSA до сих пор широко применяется практически во всех приложениях используемых криптографию с открытым ключом. Идея алгоритма состоит в следующем: некий пользователь А выбирает пару различных простых чисел p, q и вычисляет значение n = p \* q и выбирает некоторое число d таким что НОД(d, ᵠ(n)) = 1, где фи функция Эйлера. n = p \* q, p, q — простые, тогда ᵠ(n) = (p — 1) \* (q - 1) Далее находим некоторое число е такое что d \* e = 1 (mod ᵠ(n)), тогда после выполнения вычислительный операций он размещает в общий доступ пару (e, n), которые являются открытым ключом и хранит в секрете пару (d, n), которые являются закрытым ключом. Если некоторый пользователь В желает предать некоторое сообщение пользователю А, который имеет вид х = (х1, х2, …, хn) принадлежащих Zn. Тогда он будет выполнять шифрование с использованием окрытого ключа по следующей схеме E(e, n) : c → ce mod n. После передачи имеенно шифрованного текста пользователю А, он его шифрует D(d, n) : c → cd mod n

Криптосистема Эль-Гамаля

Система Эль-Гамаля тоже является криптосистемой с открытым ключом, которая основана на проблеме логарифми. Система включает в себя как алгоритм шифрования и дешифрования, так и алгоритм цифровой подписи. Идея криптосистемы эль-гамаля заключается в следующем: множество параметров системы включая простое целое число p и целое число g. Степени которого по модулю p порождают большое число элементов множества Zp. Предположим у пользователя А есть некий секретный ключ а и открытый ключ у, где значение у = ga (mod p). предположим что пользователь B желает послать сообщение m пользователю А. В самом начале пользователь B выбирает некоторое случайное число к < p, затем он вычисляет (пользователь В) значение y1 = gk (mod p) и значение y2 = m (исключающее или плюс в кружочке) (yk (mod p)), где побитовое исключающее или. И отправляет пользователю А пару (у1, у2). После получения шифрованного текста пользователь А вычисляет значение m = (y1a mod p) (исключающее или плюс в кружочке) y2

Алгоритм Диффи-Хелмана для обмена ключей

Цель алгоритма состоит в том чтобы два пользователя могли безопасно обменяться ключом который в дальнейшем может использоваться в каком либо алгоритме симметричного шифрования. Алгоритм основан на трудности вычислений дискретных логарифмов. Дискретный логарифм определяется следующим образом: вводится понятие примитивного корня простого числа Q как числа чьи степени создают все целые числа в диапазоне от 1 до Q-1. Это означает что если некое число А является примитивным корнем простого числа Q то, числа A mod Q, A2 mod Q, A3 mod Q, …, AQ-1 mod Q являются различными и состоят из целых чисел в диапазоне от 1 до Q-1 с некоторыми перестановками. Для любого целого числа В < Q и примитивного корня А для числа Q, можно найти единственную экспоненту х такую что Y = AX mod Q , 0 ≤ X ≤ Q-1. Тогда в данном случае Х является открытым ключом а У открытым ключом

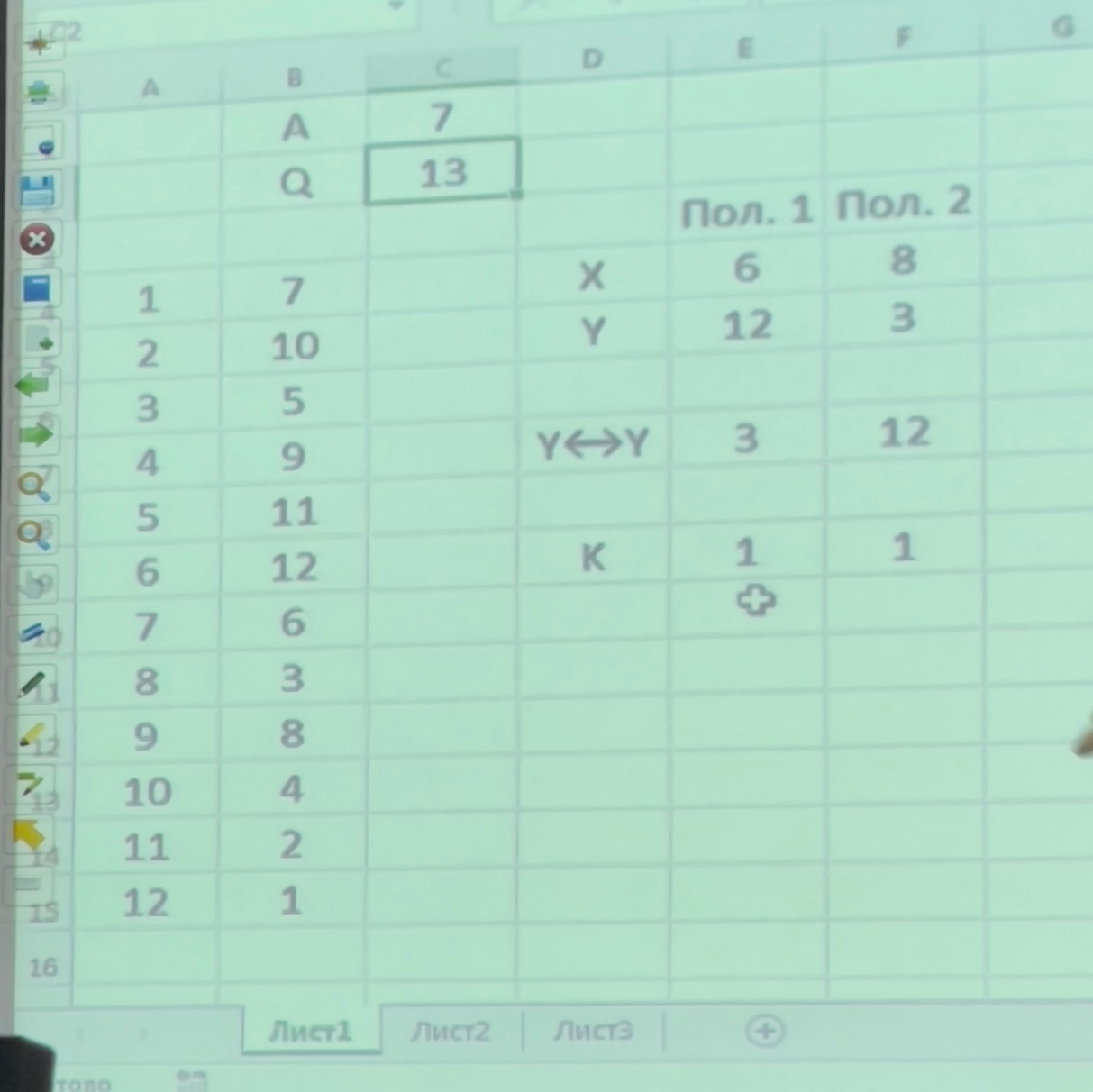
Х1 : Y1 = AX1 mod Q

Х2 : Y2 = AX2 mod Q

Y1 → ← Y2

K1 = Y2X1 mod Q

K2 = Y1X2 mod Q, где K1 = K2

пример (смотри алгоритм и потом пример, там везде деление мод Q )

Электронно цифровые подписи

Передача сообщения отправителем неким пользователем А получателю В предполагает передачу данных побуждающую пользователей к определённым действиям, передача данных может представлять собой передачу фондов между банками, продажу акций или облигаций на автоматизированном рынке, а также передачу сигналов или передачи по каналам электросвязи. Участники нуждаются в защите от множества злонамеренных действия к которым относятся:

1. Отказ (ренегатство) — отправитель в последствии отказывается от переданного сообщения
2. Фальсификация — получатель поделывает сообщение
3. Изменение — получатель вносит изменения в сообщение
4. Маскировка — нарушитель маскируется под другого пользователя

Для верификации (подтверждения) сообщение m который пользователь А передал пользователю В необходимо следующее:

1. Отправитель пользователь А должен внести в m подпись содержащую дополнительную информацию зависшую от m и в общем случае от получателя сообщения и известной только отправителю закрытой информации *к*А
2. Необходимо чтобы правильную подпись сообщения m которая выглядит таким образом m:SIG{*к*А, m, идентификатор В}нельзя было составить без *к*А
3. Для предупреждения повторного использования устаревших сообщений процедура составления подписи должна зависеть от времени
4. Пользователь В должен иметь возможность удостоверится, что подпись состоящий из данных есть правильная подпись для сообщения m пользователем А

Подпись сообщения определённый способ шифрования m путём криптографического преобразования, закрываемым элементом *к*А в преобразовании

<Идентификатор В, m>→SIG{*к*А,m, идентификатор В} является ключ криптопреобразования. Во всех практических криптографических системах *к*А принадлежит конечному множеству ключей к

1. исчерпывающая проверка всех ключей задаваемых соответствующими парами <m, Идентификатор В>→SIG{*к*А,m, идентификатор В} в общем должна привести к определению ключа kA злоумышленникам. Если множество К достаточно велико и ключ к определён методом случайного выбора то полная проверка ключей невозможна. Составить правильную подпись без ключа невозможно так как цифровая подпись ключом SIG{*к*А,m, идентификатор В} без ключа *к*А с вычислительной точки зрения эквивалентно поиску ключа
2. Доступ к аппаратуре программа и файлам системы обработки информации обычно контролируются паролями, подпись это вид пароля зависящий от отправителя получателя информации и содержания передаваемого сообщения
3. Подпись должна меняться от сообщения к сообщению для предупреждения её повторного использования с целью проверки нового сообщения. Цифровая подпись отличается от рукописной которая обычно не зависит от времени составления и данных. Цифровая и рукописная подпись идентичны в том смысле, что они характерны только для данного владельца
4. Хотя получатель информации не может составить правильную подпись, он должен иметь её подлинность. Этот процесс посредством которого каждая сторона устанавливает подлинность другой. Обязательным условием этого процесса является сохранение тайны.

В обычных коммерческих сделках процесс установления подлинности подписи выполняет третье независимые доверенные лица. Для того чтобы в системе обработки данных получатель мож установить подлинность отправителя необходимо выполнение следующих условий:

* + Отправитель А должен обеспечит В удостоверяющей информацией вида AUTH {*к*А,m, идентификатор В} зависящий от секретной информации *к*А известной только пользователю А
  + Необходимо чтобы удостоверяющую информацию AUTH {*к*А,m, идентификатор В} от пользователя А пользователю В можно было дать только при наличии ключа *к*А
  + Пользователь В должен располагать процедурой проверки того что AUTH {*к*А,m, идентификатор В} действительно подтверждает личность пользователя А
  + Для предупреждения использования предыдущей проверенной на достоверность информации процесс установления подлинности должен иметь некоторую зависимость от времени, алгоритм составления электронно цифровой подписи основанный на асимметричной криптосистеме, точнее с использование криптосистемы RSA выглядит следующим образом. Пользователь А вырабатывает цифровую подпись для пользователя В сообщения m с помощью следующего преобразования SIG(M) = EeB, nB (EdA,nA(M)). В данном преобразовании он использует своё секретное преобразование с ключом EdA,nA и открытое преобразование с ключом EeB, nB. Затем он передаёт пользователю В пару <m, SIG(m)>. пользователь В может верифицировать это подписанной сообщение сначала при помощи своего секретного преобразования EeB, nB с целью получения EdA,nA(M) = EeB, nB(SIG(M) = EeB, nB (EdA,nA(M))) и затем открытого преобразования EeB, nB пользователя А для получения сообщения m, затем пользователь В производит сравнение полученного сообщения m с тем которое он получил в результате проверки цифровой подписи и принимает решение об подлинности или подложности полученного сообщения. Данный алгоритм предназначен для проверки подлинности электронно цифровой подписи только одному пользователю В. Если требуется обеспечение возможности верификации электронно цифровой подписи произвольным пользователем, то алгоритм выработки электронно цифровой подписи упрощается и подпись вырабатывается по SIG(m) = EdA, nA (m) и другие пользователи осуществляют верификацию с использованием открытого преобразования пользователя А по формуле m = EeA, nA (SIG(m)) = EeA, nA (EdA, nA (m))

Вместо криптосистемы RSA для подписи сообщений можно использовать любую другую асимметричную криптосистему недостатком подобного подхода является то что производительность асимметричной криптосистемы можно оказаться недостаточной для удовлетворения предъявляемым требованиям, возможным решением является применение специальной эффективно вычисляемой функции называемой хэш-функцией или функцией хэширования. Входом этой функции является сообщение а выходом слово фиксированной длины, намного меньше чем длина исходного сообщения. Электронно цифровая подпись вырабатывается по той же схемой но при этом используется не само сообщение а значение хэш функции от него

**20.12.24**

**Использование сертификатов**

**24.12.24**

**Защита файлов**

В офисе есть что-то, при помощи которой реализуется защита от доступности информации в файлах. Данный способ предназначен только для установки пароля с целью ограничения доступа, а содержимое файлов остаётся в открытом виде. Т.е. если при помощи более продвинутых текстовых редакторов каким-то образом открывать эти файлы, тогда можно увидеть, что содержимая информация хранится в открытом виде.