ЛЕКЦИЯ 17.
Криптографические шифраторы.

17.1. Функциональные возможности и структура аппаратного шифратора.

17.2. Принцип действия аппаратного шифратора.

17.3. Основные типы современных шифраторов.

17.4. Основные направления развития технологии смарт-карт.

Классификация современных шифраторов

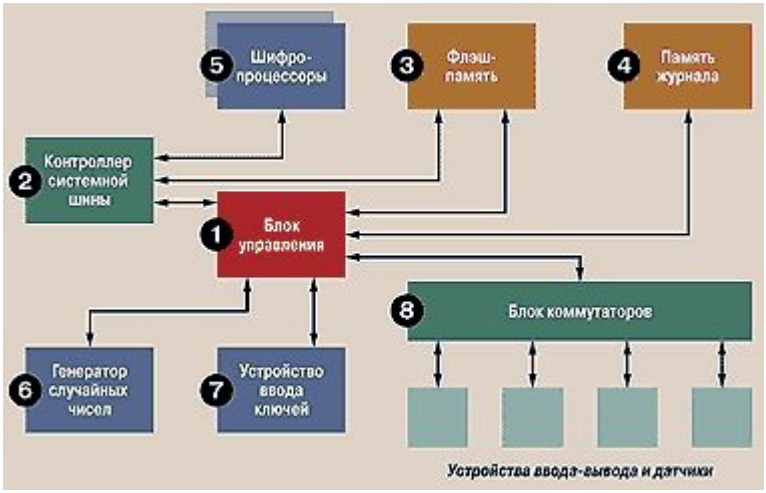
Функциональные возможности аппаратного шифратора:

1.Генерация случайных чисел.
2.Контроль входа на компьютер.
3.Контроль целостности файлов операционной системы.

Плата с перечисленными возможностями называется
устройством криптографической защиты данных — УКЗД.

Шифратор, выполняющий контроль входа на ПК и проверяющий целостность операционной системы, называют также
«электронным замком».

Структура аппаратного шифратора



1. Блок управления - основной модуль шифратора, реализуется на базе микроконтроллера.

2. Контроллер системной шины ПК (например, ***PCI***) - осуществляет основной обмен данными между ***УКЗД*** и компьютером.

3. Энергонезависимое запоминающее устройство (ЗУ) - микросхема *флэш-памяти*.
(достаточно ***емкая*** (несколько **мегабайт**) и допускает ***большое число циклов записи)***.
Здесь размещается программное обеспечение микроконтроллера, которое выполняется при инициализации устройства (т. е. когда шифратор перехватывает управление при загрузке компьютера).

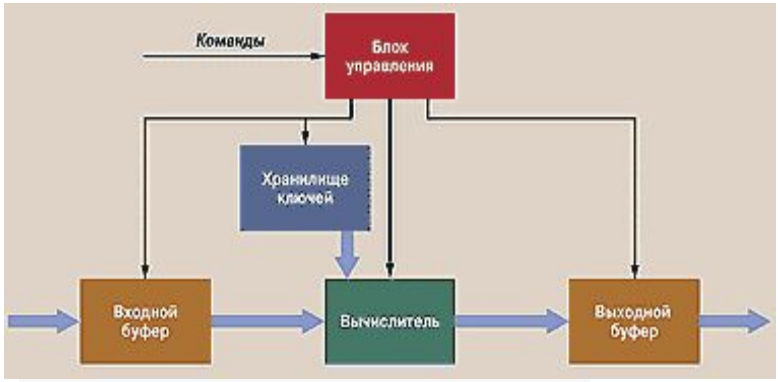
4. Память журнала - *энергонезависимое ЗУ*;
еще одна *флэш-микросхема*: память для программ и память для журнала **не должны объединяться.**

5.Шифропроцессор (или несколько) - это специализированная микросхема или микросхема ***программируемой логики PLD (Programmable Logic Device)***.

6. Генератор случайных чисел - устройство, дающее статистически случайный и непредсказуемый сигнал - *белый шум*.

7. Блок ввода ключевой информации. Обеспечивает ***защищенный прием ключей*** с ключевого носителя, через него также вводится ***идентификационная информация*** о пользователе, необходимая для решения вопроса "свой/чужой".

8. Блок коммутаторов. ***УКЗД*** может по указанию администратора безопасности ***отключать возможность работы с внешними устройствами***: дисководами, CD-ROM, параллельным и последовательным портами, шиной USB и т. д.



Структура шифропроцессора

Вычислитель - набор регистров, сумматоров, блоков подстановки и т. п., связанных между собой шинами передачи данных.

Он выполняет ***криптографические действия***, причем, *максимально быстро*.

На вход вычислитель получает открытые данные, которые следует зашифровать, и **ключ шифрования** (случайное число).

1. **Блок управления -** *аппаратно реализованная* ***программа***, управляющая вычислителем.
2. **Буфер ввода-вывода** - для ***повышения производительности*** устройства: пока шифруется первый блок данных, загружается следующий и т. д.

Потоковая скорость обработки данных (***мегабайты в секунду)***:

$$V = F \times K / n,$$

где **F** — тактовая частота,

K — размер стандартного блока шифрования,

n — число тактов, требующееся на преобразование стандартного блока.

Алгоритм ***ГОСТ 28147—89*** имеет быстродействие **32** такта на **8-байтовый** блок: теоретически скорость шифрования стремится к **25 Мбайт/с** при тактовой частоте ***100 МГц.***

Программная реализация криптоГОСТа на современных ПК достигает **12—16 Мбайт/с** при тактовой частоте процессора **1 ГГц**. (В этом случае теоретическая скорость шифрования - около **250 Мбайт/с).**

Принцип действия аппаратного шифратора

У ***аппаратных шифраторов*** два основных режима работы:
начальной загрузки и **выполнения операций.**

Режим **начальной загрузки** начинается при загрузке компьютера. **Шифратор**:

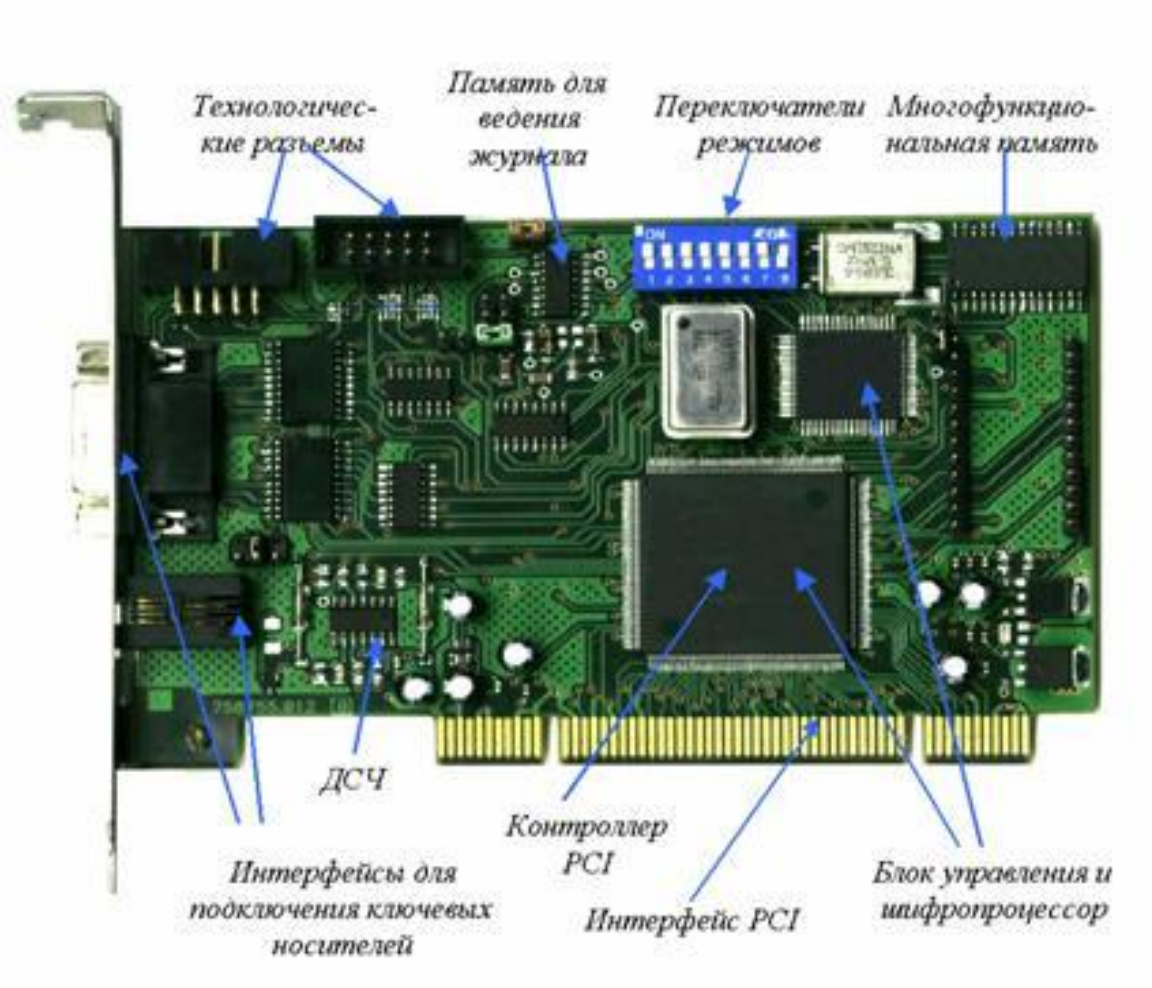
1. перехватывает управление и выполняет последовательность команд, предлагая пользователю прежде всего ввести ***главный ключ шифрования;***
2. выполняет различные операции с ***ключами шифрования***: их *загрузку* в шифропроцессор и *выгрузку* из него, а также *взаимное шифрование* ключей;
3. рассчитывает ***имитовставки*** для данных и ключей;
4. генерирует ***случайные числа*** по запросу.

Шифратор может получать команды сразу от *нескольких* программ:

- программы шифрования файлов;
- программы шифрования данных и вычисления ***имитовставок*** от драйвера, выполняющего *прозрачное* (**автоматическое**) шифрование сетевых пакетов (например, реализующего механизмы виртуальных частных сетей);
- запросы на генерацию случайных чисел от программы-генератора криптографических ключей и т.д.

Программы **не имеют прямого доступа к шифратору** и управляют им с помощью специальных ***программных API-модулей.***

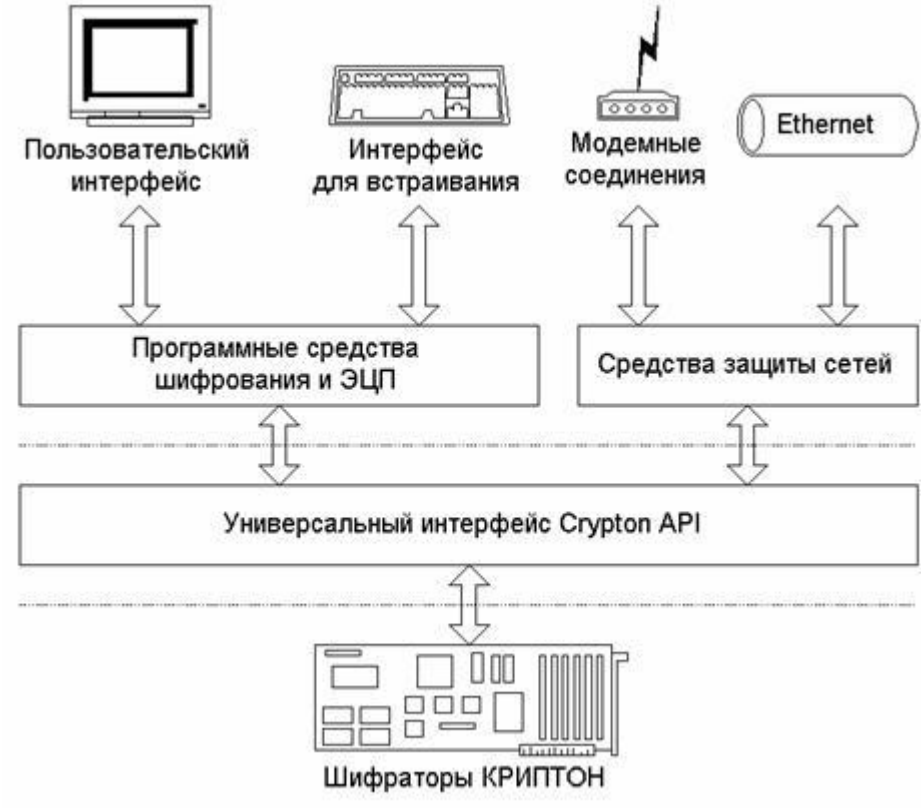
Функция *API* - обеспечение корректного последовательного ***выполнения шифратором команд, инициированных различными программами***.



Общий вид аппаратного шифратора типа "Криптон"

При обращении программы к ***УКЗД*** любая команда проходит **четыре** уровня (см. рис.):

- приложений,
- интерфейса между приложением и драйвером ***УКЗД***,
- ядра операционной системы - драйвера ***УКЗД*** ,
- **аппаратный** (собственно *уровень шифратора*).



Программный интерфейс ***Crypton API***.

Аппаратные шифраторы должны поддерживать *несколько уровней ключей* шифрования.
Трехуровневая иерархия ключей предусматривает использование:
сеансовых или **пакетных** ключей - **1-й уровень**,
долговременных пользовательских или ***сетевых*** ключей - ***2-й уровень*** ,
главных ключей - **3-й уровень**.

Шифрование данных выполняется *только* на ключах **первого уровня** (**сеансовых** или **пакетных**), остальные - для шифрования *самих ключей* при построении различных ключевых схем.

Упрощенный **пример** процесса шифрования файла:

1. На этапе **начальной загрузки** в ключевую ячейку ***№ 1*** заносится **главный** ключ. Но для *трехуровневого* шифрования необходимо получить еще два.
2. **Сеансовый** ключ генерируется в результате запроса к *датчику случайных чисел* (***ДСЧ***) шифратора на получение случайного числа, которое загружается в ключевую ячейку ***№ 2***, соответствующую **сеансовому** ключу. С его помощью шифруется содержимое **файла** и создается **новый файл**, хранящий зашифрованную информацию.
3. Далее у пользователя запрашивается ***долговременный*** ключ, который загружается в ключевую ячейку ***№ 3*** с расшифровкой посредством **главного** ключа, находящегося в ячейке ***№ 1.***
4. И, наконец, **сеансовый** ключ зашифровывается при помощи ***долговременного*** ключа, находящегося в ячейке ***№ 3***, выгружается из шифратора и записывается в заголовок зашифрованного файла.
5. При **расшифровке файла** сначала с помощью ***долговременного*** ключа пользователя расшифровывается **сеансовый** ключ, а затем с его помощью восстанавливается информация.

Многоключевая схема имеет **преимущества**:

Во-первых, **снижается нагрузка** на ***долговременный*** ключ - он используется только для шифрования *коротких* сеансовых ключей.

Во-вторых, при *смене* ***долговременного*** ключа можно очень быстро перешифровать файл: достаточно перешифровать **сеансовый** ключ со старого ***долговременного*** на новый.

В-третьих, разгружается *ключевой носитель* - на нем хранится только **главный** ключ, а все ***долговременные*** ключи (их может быть сколько угодно - для различных целей) могут храниться в зашифрованном с помощью **главного** ключа виде даже на жестком диске ПК.

Функция "электронного замка":

обеспечивает ПК защиту от несанкционированного доступа и позволяет контролировать целостность файлов операционной системы и используемых приложений.

Память шифратора, работающего в режиме "**электронного замка**", должна содержать :

- список пользователей, которым разрешен вход на защищаемый данным шифратором компьютер, и **данные**, необходимые для их **аутентификации**;
- список контролируемых файлов с рассчитанным для каждого из них ***хэш-значением*** ;
- журнал, содержащий список попыток входа на компьютер, как успешных, так и нет; в последнем случае - с указанием причины отказа в доступе.

Основные типы современных шифраторов.

Внешний вид персонального шифратора Шипка-1.5

Шипка-1.5 – аббревиатура от слов “***Шифрование – Идентификация – Подпись – Коды Аутентификации”***– это ***USB***-устройство, в котором аппаратно реализованы:

1. Все стандартные **российские криптографические алгоритмы**:
шифрование (**ГОСТ 28147-89**);
вычисление *хэш-функции* (**ГОСТ Р 34.11-94**);
вычисление и проверка *ЭЦП* (**ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001**);
вычисление *защитных кодов аутентификации (**ЗКА**)*: чтобы убедиться, что данные правильно обрабатываются и нет нарушений в технологии, используются ***защитные коды аутентификации***; для этого в некоторых точках происходит проверка результата операций и, если он не совпадает с “правильным”, подается сигнал тревоги.

2. Ряд ***зарубежных алгоритмов***:
шифрование **RC2, RC4** и **RC5, DES, 3DES, RSA;**
хэш-функции **MD5 и SHA-1**;
ЭЦП **RSA, DSA**.

3. Два изолированных ***энергонезависимых*** блока памяти:
для хранения *критичной ключевой информации* – память объемом **4 кбайт**, размещенная непосредственно в вычислителе;
для хранения разнообразной ключевой информации, паролей, сертификатов и т.п. – память объемом до **2 Мбайт**.

4. Аппаратный **генератор случайных чисел.**

Устройство Шипка-1.5 обеспечивает решение самых разных задач защиты информации:

- шифрование и/или подпись файлов;
- защищенное хранилище паролей для различных web-сервисов;
- аппаратная идентификация пользователя в бездисковых решениях типа “тонкий клиент”;
- аппаратная идентификация пользователя для ПАК “Аккорд-NT/2000”, установленного на ноутбуках;
- аппаратная авторизация при загрузке ОС Windows на ПК;
- хранилище ключей и аппаратный датчик случайных чисел для криптографических приложений;
- использование смарт-карты в типовых решениях :
    1) авторизация при входе в домен Windows,
    2) шифрование и/или подпись сообщений в почтовых программах (например, Outlook Express),
    3) для получения сертификатов *Удостоверяющего Центра* для пар “***имя пользователя + открытый ключ***”.

Электронный идентификатор **ruToken**

имеет:

- свою собственную файловую систему,
- ***аппаратную*** реализацию алгоритма шифрования по **ГОСТ 28147-89**
- и содержит до **128 кбайт** защищенной энергонезависимой памяти.

Общий вид персональных идентификаторов типа ruToken

Электронный идентификатор **ruToken** позволяет обеспечить:

- надежную ***двухфакторную аутентификацию*** пользователей;
- ***хранение*** в памяти **ruToken** ключей шифрования, паролей и сертификатов;
- защиту электронной почты (***ЭЦП, шифрование***);
- сокращение эксплуатационных затрат,
- простоту использования.

Основные направления развития технологии смарт-карт

Цифровые интеллектуальные карты

– пластиковые карточки со *встроенным микроконтроллером* и защищенной памятью.

При создании *оптимизированных кристаллов* для интеллектуальных карт будут использованы:

- ***сверхмалогабаритные 16-*** или ***32-разрядные ядра процессоров;***
- более высокопроизводительные **8-разрядные** устройства, подобные ***RISC-процессорам***, выполняющим *одну команду* всего за **один цикл** (в отличие от **6 – 12 циклов** на команду для большинства распространенных микроконтроллеров, например, серии *8051);*
- *энергонезависимая*, но ***электрически перепрограммируемая память*** (**EEPROM**);
- ***флэш-память.***

Критерии выбора аппаратных шифраторов

Важнейшая характеристика –
реализуемый **алгоритм шифрования** и **размерность ключа**.

Другие параметры :

- **скорость** шифрования,
- **количество уровней** ключевой системы шифратора,
- **интерфейс** (***ISA/PCI/USB***),
- **набор поддерживаемых ключевых носителей** с возможностью прямой загрузки ключей шифрования,
- наличие функциональности "**электронного замка**",
- наличие **драйверов шифратора** для различных ОС,
- наличие **программного обеспечения**, позволяющего использовать функциональность шифратора.