1. Шифры замены. Одноалфавитные системы.

В шифрах замены, также называемых подстановочными шифрами, буквы исходного сообщения заменяются на подстановки. Замены в криптотексте расположены в том же порядке, что и в оригинале. Если использование замен постоянно на протяжение всего текста, то криптосистема называется одноалфавитной (моноалфавитной). В многоалфавитных системах использование подстановок меняется в различных частях текста.

Шифром замены называется алгоритм шифрования, который производит замену каждой буквы открытого текста на какой-то символ шифрованного текста. Получатель сообщения расшифровывает его путем обратной замены.

В классической криптографии различают 4 разновидности шифров замены:

- Простая замена, или одноалфавитный шифр. Каждая буква открытого текста заменяется на один и тот же символ шифртекста.
- Омофонная замена. Аналогична простой замене с единственным отличием: каждой букве открытого текста ставятся в соответствие несколько символов шифртекста. Например, буква "А" заменяется на цифру 5, 13, 25 или 57 , а буква "Б" — на 7, 19, 31 или 43 и так далее.
- Блочная замена. Шифрование открытого текста производится блоками. Например, блоку "АБА" может соответствовать "РТК", а блоку "АББ" — "СЛЛ".
- Многоалфавитная замена. Состоит из нескольких шифров простой замены. Например, могут использоваться пять шифров простой замены, а какой из них конкретно применяется для шифрования данной буквы открытого текста, — зависит от ее положения в тексте.

Все упомянутые шифры замены легко взламываются с использованием современных компьютеров, поскольку замена недостаточно хорошо маскирует стандартные частоты встречаемости букв в открытом тексте.

Примером одноалфавитного шифра является Шифр Цезаря.

2. Шифр Цезаря. Шифр Цезаря с ключевым словом. Аффинная криптосистема. Шифр Полибия. Методы вскрытия одноалфавитных систем.

Шифр Цезаря В шифре Цезаря каждая буква замещается на букву, находящуюся **k** символами правее по модулю равному количеству букв в алфавите. $C_k(j)=(j+k)(\text{mod } n)$, n - количество букв в алфавите

Очевидно, что обратной подстановкой является: $C_k^{-1}(j)=C_{n-k}=(j+n-k)(mod\ n)$

Шифр Цезаря с ключевым словом

В данной разновидности шифра Цезаря ключ задается числом k (0<=k<=n-1) и коротким ключевым словом или предложением. Выписывается алфавит, а под ним, начиная с k-й позиции, ключевое слово. Оставшиеся буквы записываются в алфавитном порядке после ключевого слова. В итоге мы получаем подстановку для каждой буквы. Требование, чтобы все буквы ключевого слова были различными не обязательно - можно записывать ключевое слово без повторения одинаковых букв.

Аффинная криптосистема Обобщением системы Цезаря является аффинная криптосистема. Она определяется двум числами a и b, где 0<=a, b<=n-1. n - как и раньше, является мощностью алфавита. Числа a и n должны быть взаимно просты.

Соответств заменами яв-ся: $A_{a,b}(j)=(a*j+b)(mod\ n)$ и $A^{-1}_{a,b}(j)=(j-b)*a^{-1}(mod\ n)$

Обр замену также можно получить, просто поменяв местами строки в табл замен.

Взаимная простота a и n необходима для биективности отображения, в противном случае возможны отображения различных символов в один и неоднозначность дешифрирования.

Шифр Полибия. Его суть состоит в следующем: рассмотрим прямоугольник, часто называемый доской Полибия. По вертикали и горизонтали выписываются некоторые буквы алфавита, а внутри доски пишется построчно весь алфавит.Каждая буква может быть представлена парой букв, указывающих строку и столбец, в которых расположена данная буква. Так представления букв В, Г, П, У будут АВ, АГ, ВГ, ГБ соответственно.

Методы вскрытия одноалфавитных систем

При своей простоте в реализации одноалфавитные системы легко уязвимы. Одним из таких методов является **частотный анализ**. Распределение букв в криптотексте сравнивается с распределением букв в алфавите исходного сообщения. Буквы с наибольшей частотой в криптотексте заменяются на букву с наибольшей частотой из алфавита. Вероятность успешного вскрытия повышается с увеличением длины криптотекста.

Простейшая защита против атак, основанных на подсчете частот, обеспечивается в системе омофонов - однозвучных подстановочных шифров, в которых один символ открытого текста отображается на несколько символов шифротекста, их число пропорционально частоте появления буквы. Шифруя букву исходного сообщения, мы выбираем случайно одну из ее замен. Следовательно простой подсчет частот ничего не дает криптоаналитику.

Таблица частот биграмм отражает кол-во повт-й в тексте опр пар букв (биграмм).

Для шифра Цезаря им-ся более простой способ расш-ки - **метод полосок**. На каждую полоску наносятся по порядку все буквы алфавита. В криптограмме берется некоторое слово. Полоски прикл-ся друг к другу так, чтобы образовать данное слово. Двигаясь вдоль полосок нах-ся осмысленное словосочетание, которое и служит расшифровкой данного слова, одновременно находится и величина сдвига

3. Многоалфавитн системы (шифр Вернама, шифр с автоключом).

Полиалфавитные подстановочные шифры были изобретены Лином Баттистой (Lean Battista) в 1568 году. Основная идея многоалфавитных систем состоит в том, что на протяжении всего текста одна и та же буква может быть зашифрована по-разному. Т.е. замены для буквы выбираются из многих алфавитов в зависимости от положения в тексте. Это является хорошей защитой от простого подсчета частот, так как не существует единой маскировки для каждой буквы в криптотексте. В данных шифрах используются множественные однобуквенные ключи, каждый из которых используется для шифрования одного символа открытого текста. Первым ключом шифруется первый символ открытого текста, вторым - второй, и т.д. После использования всех ключей они повторяются циклически.

Шифр Вернам Идея шифра состоит в том, чтобы посимвольно суммировать код буквы откр текста с кодом буквы ключа. При этом необходимо иметь случайно сформированную последовательность симвллов ключа длиной, равной длине текста.Суммирование по модулю2. При получении шифрованного сообщения нужно произвести аналогичную операцию с тем же ключом.

+ Высокая стойкость шифра(при формировании случайной последовательности взломать его нефозможно)

- сложность формирования случайной последовательности
- распространение ключа

Шифр с автоключом

Для того, чтобы избежать формирования случайной ключевой последовательности длиной, равной длине текста, применяют автоключ. Формируют короткий случайный ключ и пишут его с начала ключа, остальное место заполняют исходным открытым текстом до длины исходного текста. Производят сложение по модулю *, равному количеству символов в исходном алфавите,* открытого текста и полученного ключа.

4. Шифры перестановки (простой столбцевой перестановочный шифр, перестановочный шифр с ключевым словом).

В шифре перестановки буквы открытого текста не замещаются на другие, а меняется сам порядок их следования. Например, в шифре простой колонной перестановки исходный открытый текст записывается построчно (число букв в строке фиксировано), а шифротекст получается считыванием букв по колонкам. Расшифрование производится аналогично: шифротекст записывается поколонно, а открытый текст можно затем прочесть по горизонтали.

Для повышения стойкости полученный шифротекст можно подать на вход второго шифра перестановки. Существуют еще более сложные шифры перестановки, однако почти все они легко взламываются с помощью компьютера.

Простой столбцевой перестановочный шифр

В данном виде шифра текст пишется на горизонтально разграфленном листе бумаги фиксированной ширины, а шифротекст считывается по вертикали. Дешифрирование заключается в записи шифротекста вертикально на листе разграфленной бумаги фиксированной ширины и затем считывании открытого текста горизонтально.

Перестановочный шифр с ключевым словом

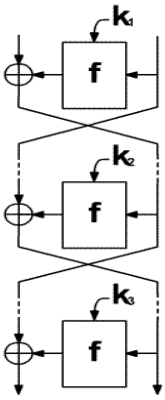
Буквы открытого текста записываются в клетки прямоугольной таблицы по ее строчкам. Буквы ключевого слова пишутся над столбцами и указывают порядок этих столбцов (по возрастанию номеров букв в алфавите). Чтобы получить зашифрованный текст, надо выписывать буквы по столбцам с учетом их нумерации:

Ключевое слово(последовательность столбцов) известно адресату, который легко сможет расшифровать сообщение.

Так как символы криптотекста те же, что и в открытом тексте, то частотный анализ покажет, что каждая буква встречается приблизительно с той же частотой, что и обычно. Это дает криптоаналитику информацию о том, что это перестановочный шифр. Применение к криптотексту второго перестановочного фильтра значительно повысит безопасность.

5. Схема Фейстеля. Режимы шифрования блочных шифров.

Подразумевала многокр вып-е одних и тех же операций. При этом на кажд этапе шифр-я блока текста исп-сь разные эл-ты ключа.



При шифр-и блок откр текста разб-ся на 2 равн части пра и лев. Прав часть пост-т на вход ф-ии шифр-я f, после вып-я кот эта часть сумм-ся с лев частью блока. Рез-т сложения и исходная правая часть после этого меняются местами. Эти действия вып-ся в рамках 1 этапа.Этапов м б произвольное количество. При этом ф-я f на всех этапах одинаковая, а ключи к-разные. Режимы шифр-я блочных шифров:

- Электронной кодовой книги (Electronic Code Book, ECB),

Исходный текст разбивается на блоки. Каждый блок независимо друг от друга шифр-ся по выбр алг-му. Из получ после преобр-я блоков формир-ся шифротекст.

+простота реализации, возможно распараллеливание

-треб-ся, чтобы длина текста кратна длине блоков

-блоки, состоящие из одинаковых бит, будут зашифр одинаково

- Сцепления блоков шифра (Cipher Block Chaining, CBC)

Исп-ся инициализация (IV). Это блок текста длиной, равной длине блока откр текста. IV шифр-ся и побитово сумм-ся по модулю 2 с 1-м модулем откр текста. Получ блок шифр-ся, давая 1-й блок шифротекста. Этот же блок С1 исп-ся для сложения со 2-м блоком откр.текста.

+Кажд новый блок маскир-ся предыдущим, т.о. любая подделка м.б.обнаружена

-треб-ся, чтобы длина текста кратна длине блоков

-эффект распр-я ошибки

- Обратной связи по шифротексту (Cipher Feedback, CFB)

i-тый блок шифротекста форм-ся путем шифр-я i-1 блока шифротекста и его суммирования с i-тым блоком открытого текста. Работа начинается с заполнения сдвигового регистра случ.посл-ю бит.Далее блок из регистра шифр-ся ф-ей F и рез-т шифрования побитово сумм-ся с блоком открытого теста. Рез-т суммирования отпр в канал связи и он же помещается в сдвиговый регистр. При этом биты, которые были в регистре, сдвиг-ся влево и усекаются.

- Обратной связи по выходу (Output Feedback, OFB)

Аналогичен предыдущему за исключением того, что обратная связь идет не от полученного шифротекста, а от формируемой ключ.посл-ти. Перед началом работы сдвиговый регистр заполняется вектором инициализации IV. Биты сдвигового регистрапопадают на вход шифрования, после чего рез-т шифр-я побитово сумм-ся с блоком откр. текста. При этом может быть взят неполный блок. В сдиг регистр помещается блок ключевой инф-ции, который использовался для суммирования. Это позволяе избежать эффекта распр-я ошибки.

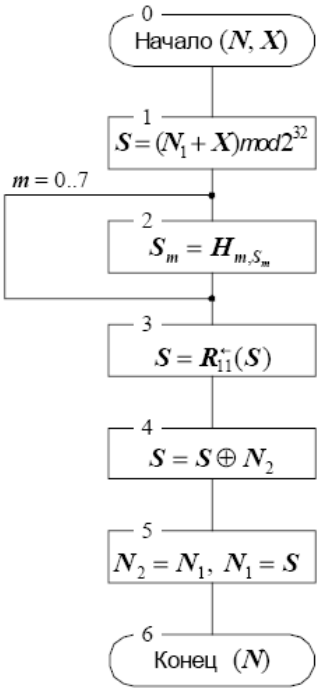
6. Алгоритм ГОСТ 28147-89. Основной шаг алгоритма.

Описание стандарта шифрования Российской Федерации содержится в документе «Алгоритм криптографического преобразования данных ГОСТ 28147-89». Помимо нескольких тесно связанных между собой процедур шифрования, в документе описан один построенный на общих принципах с ними алгоритм выработки имитовставки - криптографической контрольной комбинацией, то есть кодом, вырабатываемым из исходных данных с использованием секретного ключа с целью имитозащиты, или защиты данных от внесения в них несанкционированных изменений.

Условно ГОСТ можно разделить на 3 вложенных уровня:1. Осн.шаг криптопреоб-я.Здесь вып-ся операции над 1 блоком текста. 2. Циклы, построенные на повторении осн.шага. Они имеют следующие названия и обозначения:цикл зашифр-я (32-З);цикл расшифрования (32-Р);цикл выработки имитовставки (16-З). 3. Режимы шифр-я, построенные на комбинации циклов.

В ГОСТе исп-ся 2 вида ключ инф-и: 1 – Ключ, предст-й массив из 32-бит эл-ов.

2-Таблица замены-матрица 8 x 16. В ячейках записаны числа разрядностью 4 бита. В каждой строчке разбросаны неповторяющиеся значения от 0 до 16.



Основной шаг криптопреобразования является оператором, определяющим преобразование 64-битового блока данных. Доп-м параметром этого оператора является 32-битовый блок, в качестве кот используется какой-либо эл-т ключа.

Шаг 0. На вход поступает ***N*** – преобразуемый 64-битовый блок данных. ***X*** – 32-битовый элемент ключа.

Определяет исходные данные для основного шага криптопреобразования:, в ходе выполнения шага его младшая (N_1) и старшая (N_2) части обрабатываются как отдельные 32-битовые целые числа без знака. Таким образом, можно записать $N=(N_1,N_2)$.;

Шаг 1. Блок теста разбивается на 2 равные части и младшая (N_1) часть складывается по модулю 2^{32} с используемым на шаге элементом ключа. Получается ***S.***

Шаг 2. Поблочная замена. 32-битовое значение ***S***, полученное на предыдущем шаге, интерпретируется как массив из восьми 4-битовых блоков кода: $S = (S_0,S_1,S_2,S_3,S_4,S_5,S_6,S_7)$. Далее значение каждого из восьми блоков заменяется новым, которое выбирается по таблице замен следующим образом: значение блока ***Si*** меняется на ***Si***-тый по порядку элемент (нумерация с нуля) ***i***-того узла замен (т.е. ***i***-той строки таблицы замен, нумерация также с нуля). Другими словами, в качестве замены для значения блока выбирается элемент из таблицы замен с номером строки, равным номеру заменяемого блока, и номером столбца, равным значению заменяемого блока как 4-битового целого неотрицательного числа. Теперь становится понятным размер таблицы замен: число строк в ней равно числу 4-битовых элементов в 32- битовом блоке данных, то есть восьми, а число столбцов равно числу различных значений 4-битового блока данных, равному, как известно 2^4, шестнадцати.

Шаг 3 Новое значение ***S*** сдвигается циклически на 11 бит влево.

Шаг 4Побитовое сложение: значение, полученное на шаге 3, побитно складывается по модулю 2 со старшей половиной преобразуемого блока.

Шаг 5 Замена блоков младшей и старшей части.

Шаг 6 Склеивание старшей и младшей части.

7. Алгоритм ГОСТ 28147-89. Цикл алгоритма. Режим простой замены.

Базовые циклы закл в многократном выполнении осн шага с использованием разных эл-в ключа и отличаются друг от друга только числом повторения шага и порядком использования ключевых элементов. порядок для различных циклов.

1. Цикл зашифрования 32-З:

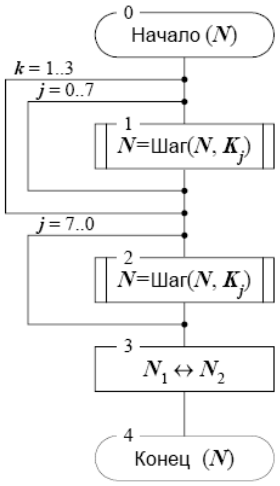
K0,***K***1,***K***2,***K***3,***K***4,***K***5,***K***6,***K***7,***K***0,***K***1,***K***2,***K***3,***K***4,***K***5,***K***6,***K***7,***K***0,***K***1,***K***2,***K***3,***K***4,***K***5,***K***6,***K***7,***K***7,***K***6,***K***5,***K***4,***K***3,***K***2,***K***1,***K***0.

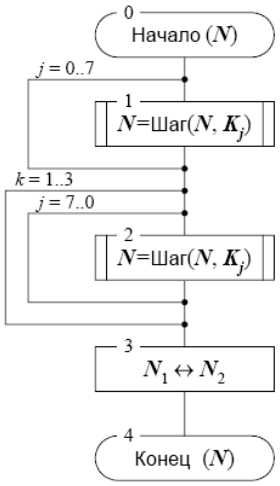
2. Цикл расшифрования 32-Р:

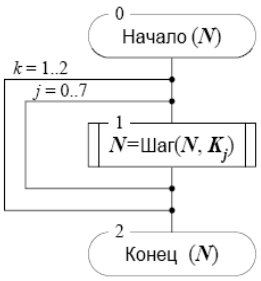
K0,***K***1,***K***2,***K***3,***K***4,***K***5,***K***6,***K***7,***K***7,***K***6,***K***5,***K***4,***K***3,***K***2,***K***1,***K***0,***K***7,***K***6,***K***5,***K***4,***K***3,***K***2,***K***1,***K***0,***K***7,***K***6,***K***5,***K***4,***K***3,***K***2,***K***1,***K***0.

3. Цикл выраб имитовставки 16-З: ***K***0,***K***1,***K***2,***K***3,***K***4,***K***5,***K***6,***K***7,***K***0,***K***1,***K***2,***K***3,***K***4,***K***5,***K***6,***K***7.

Каждый из циклов имеет собственное буквенно-цифровое обозначение, «*n*-*X*», где первый элемент обозначения (*n*), задает число повторений основного шага в цикле, а (*X*)-буква, задает порядок зашифрования («З») или расшифрования («Р»).



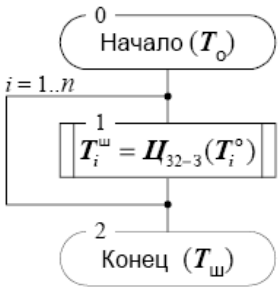




Цикл расшифр-я должен быть обратным циклу зашифр-я, то есть послед-е прим-е этих 2 циклов к произв блоку должно дать в итоге исх блок.

Цикл выраб имитовставки вдвое короче циклов шифр-я, порядок исп-я ключ эл-в в нем такой же, как в первых 16 шагах цикла зашифр-я.

ГОСТ 28147-89 предусматривает 3 след режима шифр-я данных:простая замена, гаммир-е, гаммир-е с обратной связью и один доп режим выработки имитовставки.



Простая замена Зашифрование в данном режиме заключается в применении цикла 32-З к блокам открытых данных, расшифрование – цикла 32-Р к блокам зашифрованных данных. Это наиболее простой из режимов, 64-битовые блоки данных обрабатываются в нем независимо друг от друга.

Размер массива открытых или зашифр данных должен быть кратен 64 битам: $|\boldsymbol{T}\text{о}| = |\boldsymbol{T}\text{ш}| = 64\cdot\boldsymbol{n}$, после выполнения операции размер полученного массива данных не изменяется.Режим шифрования простой заменой имеет следующие особенности:

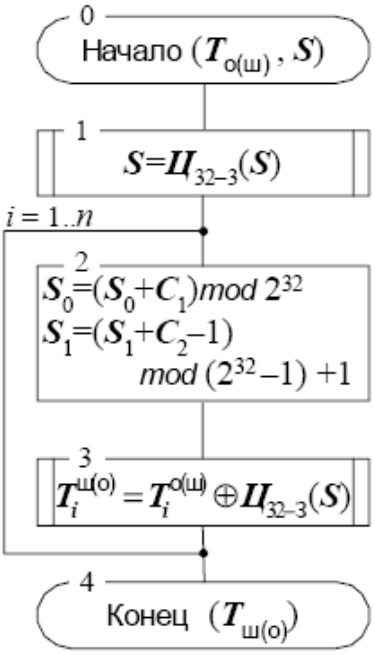
1. Так как блоки данных шифруются независимо друг от друга и от их позиции в массиве, при зашифровании двух одинаковых блоков открытого текста получаются одинаковые блоки шифртекста и наоборот.
2. Если длина шифруемого массива данных не кратна 8 байтам или 64 битам, возникает проблема, чем и как дополнять последний неполный блок данных массива до полных 64 бит. Очевидные решения типа «дополнить неполный блок нулевыми битами» или, более обще, «дополнить неполный блок фиксированной комбинацией нулевых и единичных битов» могут при определенных условиях дать в руки криптоаналитика возможность методами перебора определить содержимое этого самого неполного блока, и этот факт означает снижение стойкости шифра. Кроме того, длина шифртекста при этом изменится, увеличившись до ближайшего целого, кратного 64 битам, что часто бывает нежелательным.

- ***T***о,***T***ш – массивы соответственно открытых и зашифрованных данных;
- $\boldsymbol{Ti^{O}}$. $\boldsymbol{Ti^{Ш}}$ - *i*-тые по порядку 64-битовые блоки соответственно открытых и зашифрованных данных: $\boldsymbol{T}\text{о} = (\boldsymbol{T_1^{O}, T_2^{O}, T_3^{O}, \ldots T_n^{O}},), \boldsymbol{T}\text{ш} = (\boldsymbol{T_1^{Ш}, T_2^{Ш}, T_3^{Ш}, \ldots T_n^{Ш}},), \boldsymbol{0 \le i \le n}$, последний блок может быть неполным;
- ***n*** – число 64-битовых блоков в массиве данных;
- $\boldsymbol{Ц}_{\text{X}}$ – функция преобразования 64-битового блока данных по алгоритму базового цикла «X».

8. Алгоритм ГОСТ 28147-89. Режим гаммирования.

Для этого режима формируется специальная гамма, которая представляет собой случайную последовательность бит. Гамма побитого суммируется по модулю 2 с блоками открытого текста. Для выработки гаммы используется специальный рекуррентный генератор псевдослучайных чисел.

Гамма для этого режима получается следующим образом: с помощью некоторого алгоритмического рекуррентного генератора последовательности чисел (РГПЧ) вырабатываются 64-битовые блоки данных, которые далее подвергаются преобразованию по циклу 32-З, то есть зашифрованию в режиме простой замены, в результате получаются блоки гаммы. Благодаря тому, что наложение и снятие гаммы осуществляется при помощи одной и той же операции побитового исключающего или, алгоритмы зашифрования и расшифрования в режиме гаммирования идентичны.



Шаг 0 Определяет исходные данные для основного шага криптопреобразования:

- ***T***о(ш) – массив открытых (зашифрованных) данных произвольного размера, подвергаемый процедуре зашифрования (расшифрования), по ходу процедуры массив подвергается преобразованию порциями по 64 бита;
- ***S*** – синхропосылка, 64-битовый элемент данных, необходимый для инициализации генератора гаммы;

Шаг 1 Начальное преобразование синхропосылки, выполняемое для ее «рандомизации», то есть для устранения статистических закономерностей, присутствующих в ней, результат используется как начальное заполнение РГПЧ;

Шаг 2 Один шаг работы РГПЧ, реализующий его рекуррентный алгоритм. В ходе данного шага старшая (S_1) и младшая (S_0) части последовательности данных вырабатываются независимо друг от друга;

Шаг 3 Гаммирование. Очередной 64-битовый элемент, выработанный РГПЧ, подвергается процедуре зашифрования по циклу 32–З, результат используется как элемент гаммы для зашифрования (расшифрования) очередного блока открытых (зашифрованных) данных того же размера.

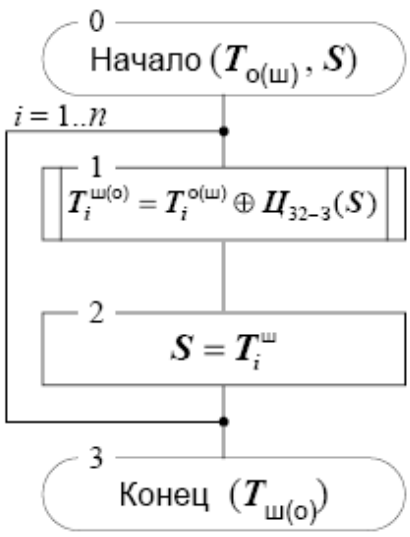
Шаг 4 Результат работы алгоритма – зашифрованный (расшифрованный) массив данных.

Данный алгоритм применим и для шифрования и для расшифрования текста при этом синхропосылку необходимо передать от отправителю к получателю, что выполняется обычно 2-мя способами.

1.её добавляют к открытому тексту

2.Либо не меняют(заранее предопределяются), либо её разрабатывают на каждой из сторон по некоему алгоритму

9. Алгоритм ГОСТ 28147-89. Режим гаммирования с обратной связью.



Гаммирование с обратной связью. Данный режим очень похож на режим гаммирования и отличается от него только способом выработки элементов гаммы – очередной элемент гаммы вырабатывается как результат преобразования по циклу 32-З предыдущего блока зашифрованных данных, а для зашифрования первого блока массива данных элемент гаммы вырабатывается как результат преобразования по тому же циклу синхропосылки. Этим достигается зацепление блоков – каждый блок шифртекста в этом режиме зависит от соответствующего и всех предыдущих блоков открытого текста. Поэтому данный режим иногда называется гаммированием с зацеплением блоков. На стойкость шифра факт зацепления блоков не оказывает никакого влияния. Схема алгоритмов за- и расшифрования в режиме гаммирования с обратной связью приведена на рисунке 2.6 и ввиду своей простоты в комментариях не нуждается.

Шифр-е в режиме гаммирования с обр связью обладает теми же особ-ми, что и шифр-е в режиме обычного гаммирования, за исключением влияния искажений шифртекста на соответствующий открытый текст. Если в режиме обычного гаммирования изменения в определенных битах шифртекста влияют только на соответствующие биты открытого текста, то в режиме гаммирования с обратной связью картина несколько сложнее. При расшифровании блока данных в режиме гаммирования с обратной связью, блок открытых данных зависит от соответствующего и предыдущего блоков зашифр данных. Поэтому, если внести искажения в зашифр блок, то после расшифр-я искаженными окажутся 2 блока откр-х данных – соответствующий и следующий за ним, причем искажения в первом случае будут носить тот же характер, что и в режиме гаммирования, а во втором случае – как в режиме простой замены. Др словами, в соответствующем блоке открытых данных искаженными окажутся те же самые биты, что и в блоке шифрованных данных, а в следующем блоке открытых данных все биты независимо друг от друга с вероятностью 1/2 изменят свои значения.

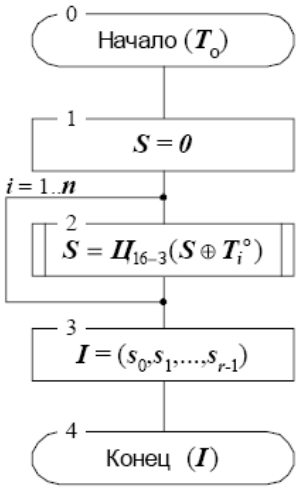
10.Алгоритм ГОСТ 28147-89. Режим выработки имитовставки.

При расшифровании в режиме простой замены соответствующий блок открытых данных оказывается искаженным непредсказуемым образом, а при расшифровании блока в режиме гаммирования изменения предсказуемы. В режиме гаммирования с обратной связью искаженными оказываются два блока, один предсказуемым, а другой непредсказуемым образом. При анализе данной ситуации необходимо учесть то, что непредсказуемые изменения в расшифрованном блоке данных могут быть обнаружены только в случае избыточности этих самых данных, причем чем больше степень избыточности, тем вероятнее обнаружение искажения.

Поскольку способность некоторых режимов шифрования обнаруживать искажения, внесенные в шифрованные данные, существенным образом опирается на наличие и степень избыточности шифруемых данных, эта способность не является имманентным свойством соответствующих режимов и не может рассматриваться как их достоинство.

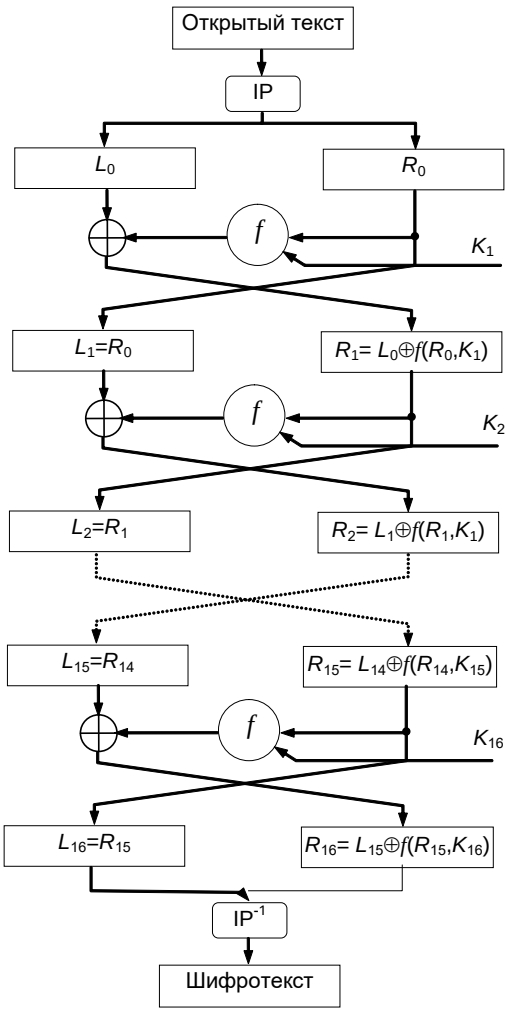
Для решения задачи обнаружения искажений в зашифрованном массиве данных с заданной вероятностью в ГОСТе предусмотрен дополнительный режим криптографического преобразования – выработка имитовставки. Имитовставка – это контрольная комбинация, зависящая от открытых данных и секретной ключевой информации. Целью использования имитовставки является обнаружение всех случайных или преднамеренных изменений в массиве информации. Проблема, изложенная в предыдущем пункте, может быть успешно решена с помощью добавления к шифрованным данным имитовставки. Для потенциального злоумышленника две следующие задачи практически неразрешимы, если он не владеет
ключевой информацией:

- вычисление имитовставки для заданного открытого массива информации;
- подбор открытых данных под заданную имитовставку;



В качестве имитовставки берется часть блока, полученного на выходе, обычно – 32 его младших бита. При выборе размера имитовставки надо принимать во внимание, что вероятность успешного навязывания ложных данных равна величине $2^{-|I|}$ на одну попытку подбора, если в распоряжении злоумышленника нет более эффективного метода подбора, чем простое угадывание. При использовании имитовставки размером 32 бита эта вероятность равна $2^{-32} \cong 0.23 \cdot 10^{-9}$.

11.DES, основной этап, описание принципа работы.



DES предст собой блочный шифр, он шифр-т данные 64-бит блоками. Вводится 64-битовый блок открытого текста, а выходит 64-битовый блок шифротекста. DES явл-я симметр алгоритмом: для шифр-я и дешифр-я исп-ся одинаковые алгоритм и ключ.

Длина ключа равна 56 битам. Ключ, который может быть любым 56-битовым числом, можно изменить в любой момент времени. Ряд чисел считаются слабыми ключами, но их можно легко избежать. Безопасность полностью определяется ключом.

На каждом этапе над 56-ю битами ключа выполняются след.операции:

1. Ключ разб-ся на 2 части и каждая из половинок циклически сдв-ся влево на 1 или 2 бита. Величина сдвига опр-ся № этапа и задана спец. таблицей.
2. Отбрас-ся часть бит у половинок ключа.

Оставш-ся биты перемеш-ся и половики скл-ся. В итоге получ-ся ключ дл-й 48 б.

Над блоком текста выполняются след.операции:

1. Перестановка с расширением. Правая половина исходного блока длиной 32 бита перемешивается и часть битов дублируется. В итоге блок длиной 48 бит.
2. Полученный блок побитово суммируется по модулю 2 с ключом.
3. S-подстановка в S-блок.

Здесь происходит преобразование исходного блока 48 бит в блок 32 бита. Используется 8 S-локов.Каждый предс собой табл 4 x 16. В ячейках записаны 4-битные числа, которые не повторяются в пределах одной строки. Исходный блок текста разбивается на 8 частей по 6 бит, которые поступают на вход отдельного S-блока. При этом для каждого S-блока первый и последний входные биты кодируют номер и строки, а остальные-номер столбца. По этим данным находится число в S-блоке, кот. Подается на выход. В итоге каждый S-блок выдает новый 4-битный элемент, который суммируется и образуется 32-битный блок данных.

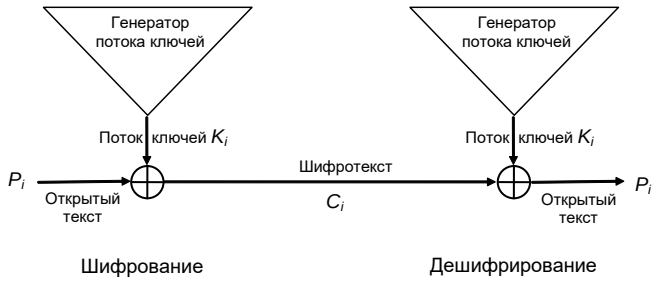
4. P2-перест-ка в P-блоке. Предст собой просто перемеш-е бит в блоке текста.
5. Побитовое сложение рез-та P2 и левой половины исх блока.

Расшифровка происх по тому же алг-му, но ключи форм-ся в обратном порядке. DES может работать во всех 4 режимах блочного шифрования.

- Слишом малая длина ключа

Большое количество статических таблиц для перестановок, что позволило накопить информацию по криптоанализу.

12.Потоковые шифры. Самосинхронизирующиеся потоковые шифры. Синхронные потоковые шифры.



Потоковые шифры преобразуют открытый текст в шифротекст по одному биту за операцию. В простейшей реализации шифрование предс т собой сложение по модулю 2 бита текста и бита ключа. При этом длина ключ последовательности должна быть равна длине текста.
Для получения длинной ключевой последовательности используются специальные генераторы ключевого потока. К ним предъявляются опр треб-я:
1. Последовательность вызываемой ими бит д.б. максимально приближена к случайной.
2. Генераторы, работающие на разных сторонах, д.б. синхронизированы
3. Ключевой поток, формируемый для разных сообщений д.б. разным
Потоковые шифры: 1. Самосинхронизирующиеся 2. Синхронные
В (1) каждый бит потока ключей является функцией фиксированного числа предыдущих битов шифротекста. Это м.б.реализовано след.образом:
В генераторе ключей имеется блок внутреннего состояния. Он предст собой сдвиговый регистр. Ф-я генерации ключевого потока зависит от состава бит в этом регистре. При одном и том же составе бит в этом регистре она выдаст одну и ту же последовательность. В этом случае не будет рассинхронизации, т.к.первый же поступивший на принимающую сторону бит шифротекста инициализирует регистр генератора ключ.потока и ф-я f выдаст строго определенный бит ключа.
- Эффект распространения ошибки. В случае искажения хотя бы 1 бита шифротекста при передаче этот бит неправильно проинициализирует генератор потока ключей. Ф-я f будет вызывать неверный ключ. Поток до тех пор пока этот искаженный бит не выйдет из регистра. После чего генерато самосинхр-ся.
-Повторные передачи текста. Злоумышленник может перехватить часть шифротекста и повторно отправить получателю Вначале может приходить некорректная информация, но затем генератор самосинхр-ся и начнет расшифр-ть все правильно.
Синхронные шифры. Генератор ключей не зависит от своего состояния, т.к. генераторы нужно синхр-ть самостоятельно. Это можно сделать по таймеру или специальной команде.
+ Отсутствие эффекта распространения ошибки
+Любые вставки шифротекста будут обнаружены, т.к. будет потеряна синхр-я
- При любых рассинхр-ях переача инф. Потока д.б.прерванаи генераторы д.б. снова синхронизированы.

13.Понятие хэш-функции. Передача информации с использованием криптографии с открытым ключом.

Хэш-функция - однонаправленная математическая процедура, позволяющая преобразовать набор произвольной длины в объём данных фиксированной небольшой длины.

При этом такая функция должна обладать следующими свойствами:

1.по открытому тексту легко осуществляется его хеш функция

2.зная хэш-функцию вычислительно невозможно за приемлемое время расчитать исходный текст из которого она была получена

3.не должно быть 2х разных сообщений, которые дают одно и то же значение хэш функции

Контроль целостности происх следующим образом:

1. Алиса формирует открытое сообщение и считает хэш-ф-ю от него
2. Алиса посылает Бобу и сообщение и его хэш-ф-ю
3. Боб повторно считает хэш-ф-ю от полученного сообщения и сравнивает с полученной от Алисы хэш-ф-ей.

Любые подделки или искажения открытого текста при передаче по каналу связи будут обнаружены Бобом, тк. Хэш-ф-я искаженного сообщения не совпадет с хэш-ф-ей исх.сообщения.

Передача информации с использованием криптографии с открытыми ключами

Используются 2 ключа-открытый и закрытый. Эти ключи формируются одновременно и обладают след.св-вами:

1. Ключи математически связаны и, зная один 1 ключ, невозможно вычислить другой
2. Информацию, зашифрованную 1 ключом, можно расшифровать только парным с ним ключом.

Схема работы ассиметричного алгоритма:

1. А и Б договариваются об алг-ме шифрования
2. А и Б формируют пары ключей
3. А и Б обмениваются открытыми ключами(закрытые остаются у них)
4. А форм-ет сообщение для Б, шифрует его открытым ключом Боба и отсылает ему
5. Б расшифр-т сообщение своим закрытым ключом

Криптография с открытыми ключами устраняет проблему распределения ключей, присущую симметричным криптосистемам. Раньше Алиса и Боб должны были тайно договориться о ключе. Алиса могла выбрать любой ключ, но ей нужно было передать его Бобу. Она могла бы послать ключ с секретным курьером, но для этого нужно время. Криптография с открытыми ключами все упрощает. Алиса может отправить Бобу секретное сообщение без каких-либо предварительных действий. У Евы, подслушивающей абсолютно все, есть открытый ключ Боба и сообщение, зашифрованное этим ключом, но она не сможет получить ни закрытый ключ Боба, ни текст сообщения.

14.Инфраструктура открытых ключей (Общие понятия, необходимость ее создания)

В криптографии с симметричными ключами для шифр-я и дешифр-я исп-ся один и тот же секретный ключ, то есть ключ шифр- совпадает с ключом дешифр-я. Стороны могут передавать друг другу данные, зашифр секретным ключом, только после того, как они обменяются этим общим ключом. Фунд-ным же св-вом шифр-я с открытым ключом является различие между ключом шифр-я и ключом дешифр-я. Открытый ключ позволяет зашифровать открытый текст, но не позволяет расшифровать его. Чтобы расшифровать текст, нужен ключ дешифр-я, кот отличается от ключа шифр-я, хотя и связан с ним. Т.о., при шифр-и с откр ключом у каждого пользователя должно быть два ключа — открытый и закрытый.

Немногие алгоритмы с откр ключом являются и безопасными, и практичными. Только 3 алг-ма хорошо работают как при шифр-и, так и для цифровой подписи: RSA, EIGamal и Rabin. Все эти алгоритмы медленны. Они шифруют и дешифр-т данные намного медленнее, чем симметричные алгоритмы. Обычно их скорость недостаточна для шифрования больших объемов данных.

Основные преимущества открытых ключей

+ Стороны не встречаются. Это преимущество особенно ощутимо при большом количестве пользователей, в случае необходимости связи двух конкретных

+ закрытый ключ и его разрядность не влияет на процесс пересылки, он не пересылается, а просто хранится в секретном месте.

Несмотря на эффективное решение проблемы распр-я ключей при ассиметр алг-мах появл новая проблема: подтверждение принадл-ти откр ключа его владельцу.

Атака «человек по середине»

1. А и Б договариваются об алг-ме шифр-я. А и Б формируют пары ключей
2. А и Б обмениваются открытыми ключами(закрытые остаются у них)
3. Человек по середине М перехватывает открытый ключ Боба, который был послан Алисе и заменяет его своим открытым ключом.
4. А шифр-т сообщение для Б открытым ключом М, думая, что это ключ Боба, и отсылает его по каналу связи.
5. М перехватывает сообщение для Б, расшифровывает его своим закрытым ключом.Получив открытый текст, М может делать с ним все, что угодно.
6. М шифр-т сообщение открытым ключом Боба и посылает ему.
7. Б расшифр-т сообщение своим закр ключом и получает неверный текст

Если бы Алиса на этапе получения откр ключа Боба могла проверить принадл-ть этого ключа, атака бы не удалась. Для решения проблемы исп-т:

1. Защищенный канал связи
2. Доступ всех пользователей в общую доверенную БД ключей
3. Использование доп криптосистем, гарантирующих подлинность ключей
4. Применение сертификации открытых ключей

15.Сертификат открытого ключа. Стандарт X.509

Под сертификацией понимается выдача каждому открытому ключу специального документа, заверенного легитимным органом, которому безоговорочно доверяют все участники взаимодействия.

Сертификат должен, как минимум, содержать открытый ключ и уникальную информацию ключа, позв-я его идентифицировать.

Под легитимным органом понимают Удостоверяющий Центр (УЦ).

Таким образом, выдавая сертификат, издатель удостоверяет подлинность связи между открытым ключом субъекта и информацией, его идентифицирующей.

Существует довольно большое количество форматов сертификатов, но наибольшее распространение в настоящее время получил формат x.509 v3. Эта базовая технология, используемая в инфраструктуре открытых ключей операционной системы Windows 2000/XP.

Состав полей сертификата:

1. версия
2. серийный номер сертификата
3. идентификатор алгоритма подписи
4. имя издателя
5. период действия сертификата
6. имя субъекта
7. открытый ключ субъекта
8. уникальный идентификатор издателя (только для v2, v3)
9. уникальный идентификатор субъекта(только для v2, v3)
10. расширение (только для v3)
11. подпись издателя

В расширение сертификата могут вкл-ся произвольное колич-во полей.

Для обеспечения функционирования системы использования открытых ключей и сертификатов необходима организация специальной инфраструктуры(PKI- public key infrastruction).

PKI – мн-во аппаратуры, людей, ПО, политик, процедур, необходимых для создания, управления, хранения, отзыва сертификатов.

16.Механизмы проверки статуса сертификатов

Сертификаты, кот.выдаются пользователям УЦ действуют строго опр время. Это время явно прописано в сертификате и при необходимости всегда можно понять, действует ли сертификат или нет, однако, могут быть ситуации, когда сертификат становится недействительным до истечения этого времени:

1. компрометация закр ключа соответствующего сертификата
2. изменение полей в сертификате
3. изменение связи между субъектом и УЦ

В этих случаях нужны доп.мех-мы преобразования серт-та из действующего в недействительный. Этот мех-м называется отзывом (аннулированием) сертификата и сост.из след этапов:

1. Формирование заявки на отхыв сертификата, кот перед-ся в УЦ
2. УЦ помещает указанный сертификат в Список отозванных сертификатов (СОС), где указывается ноер отозванного сертификата и дата и время.
3. Публикация СОС и(или) донесение инф-и о СОС до пользователей.

Разработано большое кол-во способов доставки информации о СОС, но чаще всего исп-т 3:

1. Полный СОС. Наиболее простой вариант. Каждый пользователь получает полный список, кот ведется с момента начала работы УЦ
   +Есть инф-я обо всех отозванных серт-тах
   -С течением времени СОС становится объемным
   -величение периода рассылки для экономии трафика приводит к устарванию информации о СОС со стороны пользователя.
2. Дельта-СОС-неполный список, а приращение СОС за неоторый промежуток времени
   +Если выбрать промежуток времени маленьким, то дельта-СОС тоже будет небольшим, значит бысто закачается. Инф-я о СОС будет всегда актуальной.
   -Нужно постоянно закачивать эти обновления
3. Мех-м онлайновых запросов OCSP. Подразумевает отправку пользователем коротого запроса на сервер OCSP с вопросом, действителен ли конкретный сертификат, указанный в запросе(да, нет, не знаю)

17.Жизненный цикл ключа

Как известно, любой объект (например, любое промышленное изделие) имеет определенное, конечное время жизни, за которое он проходит определенные стадии своего развития от «рождения» до «гибели». Не являются в этом смысле исключением и криптографические ключи. Для любого объекта стандартизации стандартами ISO определяются четыре стадии жизненного цикла: предоперационная, операционная, постоперационная, стадия выхода из эксплуатации.

Применительно к криптографическим ключам эти стадии означают следующее. На предоперационной стадии ключ еще не доступен для штатного использования в криптосистеме. Находясь в операционной стадии жизненного цикла, ключ доступен пользователям криптосистемы и применяется ими в штатном режиме. На постоперационной стадии ключ более не используется в штатном режиме, но доступ к нему возможен в особом режиме для специальных целей. На стадии выхода из эксплуатации ключ более недоступен, а все записи, содержащие значение ключа, удалены из криптосистемы.

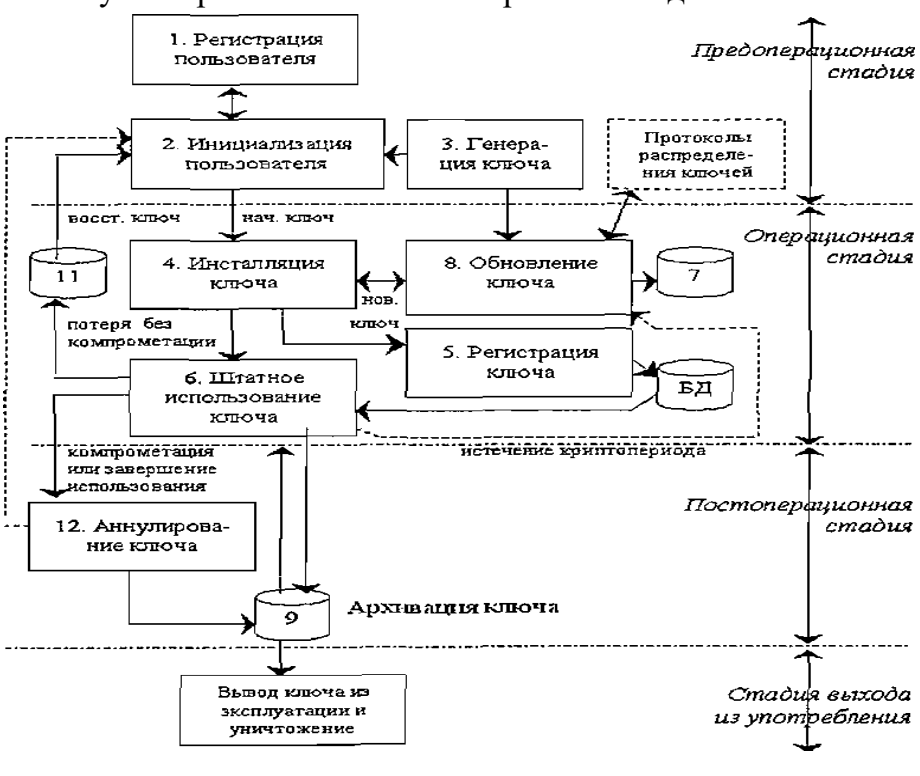
Внутри перечисленных общих стадий жизненного цикла можно более точно выделить различные состояния, или фазы, жизненного цикла, в которых пребывает ключ, а также условия переходов между ними (рис. 2.1).

Регистрация пользователя: субъект становится авторизованным пользователем криптосистемы, приобретает или создает безопасным одноразовым способом начальный ключевой материал (пароли, персональные коды и др.).

Инициализация пользователя: субъект криптосистемы инициализирует свои криптографические приложения, например, производит инсталляцию программного и аппаратного обеспечения, включая инсталляцию на него начального ключевого материала, полученного во время регистрации.

Генерация ключа: производится таким образом, чтобы гарантировать необходимые свойства для приложения или алгоритма и случайность (в смысле возможности предсказания его противником с пренебрежимо малой вероятностью). Субъект может генерировать свои собственные ключи или приобретать ключи от доверенного компонента криптосистемы.

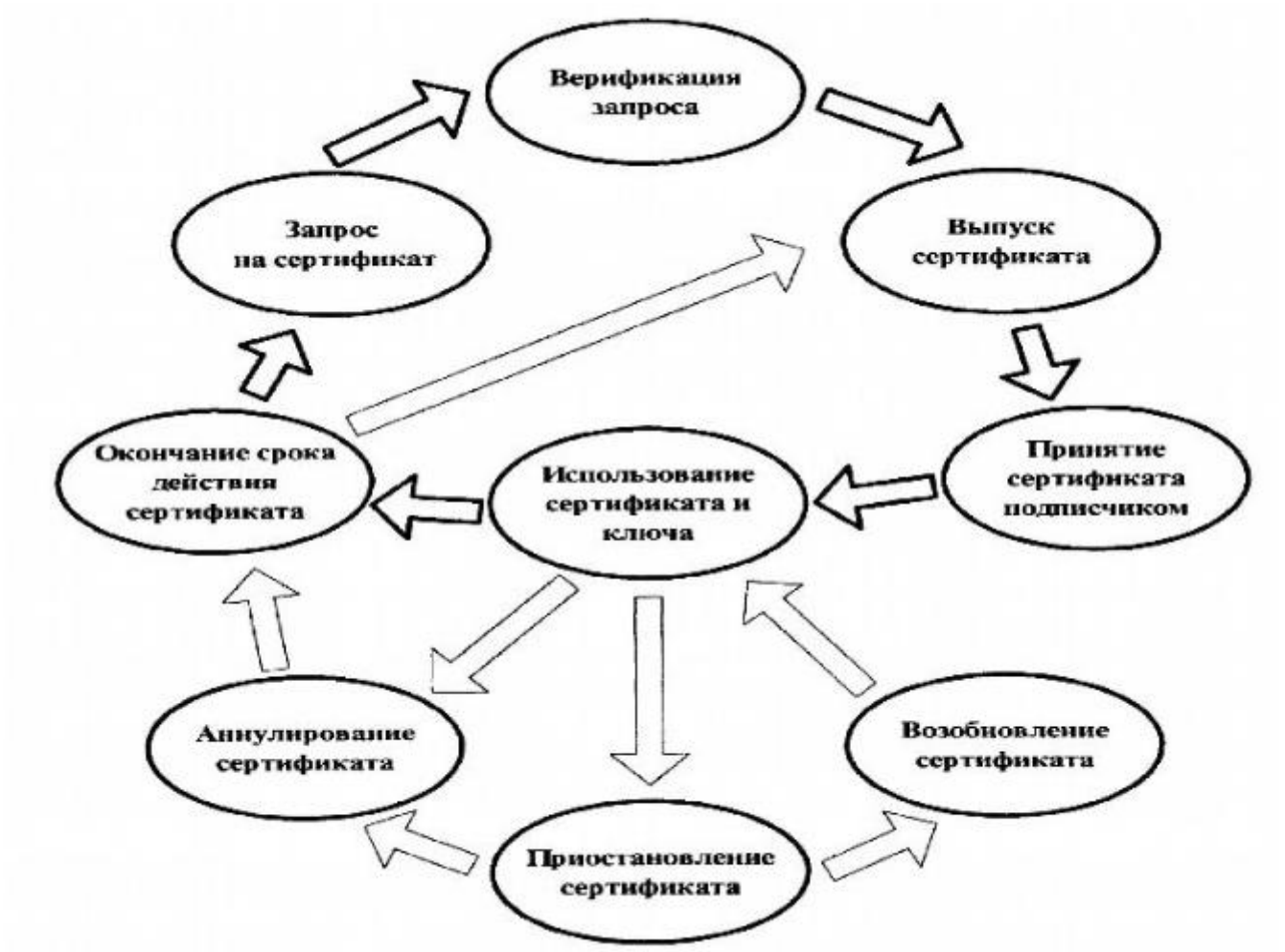
Инсталляция ключа: ключевой материал инсталлируется для функционального использования в программном или аппаратном обеспечении субъекта, включая ручной ввод пароля или персонального кода, запись на магнитный диск, в постоянную память, микроэлектронную схему или на другие носители. Начальный ключевой материал может служить для установления сеанса с доверенным компонентом криптосистемы, во время которого в реальном масштабе времени согласуются рабочие ключи. Во время последовательных обновлений новый заменяет используемый.



18.Жизненный цикл сертификата

Так как в цифровом сертификате, наряду с открытым ключом, присутствует также другая информация, имеющая определенный срок действия, жизненный цикл сертификата имеет ряд отличий от жизненного цикла ключа. Он включает в себя следующие основные фазы:

- Создание запроса в Удостоверяющий Центр на выпуск сертификата открытого ключа,
- Верификация запроса (проверка Удостоверяющим центром корректности данных),
- Выпуск сертификата в соответствии с данными, указанными в запросе, и действующей политикой сертификации,
- Распространение сертификата среди участников информационной системы,
- Хранение и выдача сертификата по запросу пользователей и владельцев сертификатов,
- Приостановка и возобновление действия сертификата,
- Обновление информации, содержащейся в сертификате, и ключевой пары
- Отзыв сертификата по запросу владельца или уполномоченного органа



19.CryptoAPI 1.0. CryptoAPI 2.0.

Как видно, криптографические функции поделены на два уровня CryptoAPI 2.0 и CryptoAPI 1.0. Это прикладные программные интерфейсы, позволяющие разработчикам добавлять криптографические функции в разрабатываемые продукты.

Уровень CryptoAPI 2.0 реализует криптографические функции более верхнего уровня. Как правило, именно эти функции вызываются из конечных приложений. Набор этих функций призван удовлетворить все запросы в области криптографии со стороны программиста. И в действительности этот набор функций очень сильно ускоряет процесс реализации функций конечных приложений связанных с криптографией и защищенной передачей данных.

Следующий уровень иерархии криптографических функций — CryptoAPI 1.0. Эти функции в свою очередь вызываются из функций уровня CryptoAPI 2.0.

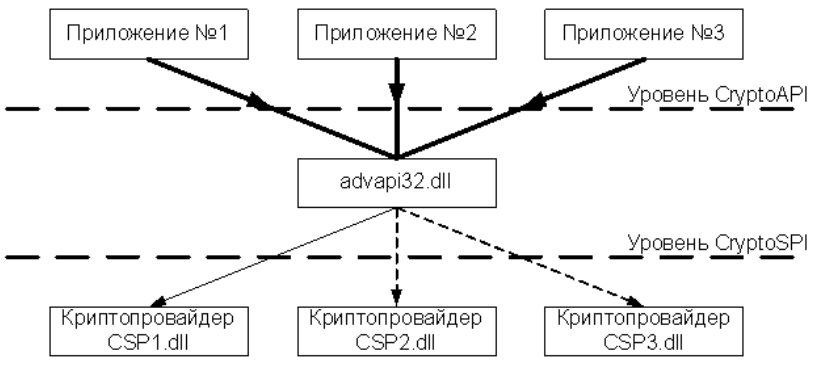
Приложение и функции CryptoAPI 2.0 вызывают не сами реализации криптографических функций, а промежуточные, «абстрактные» функции CryptoAPI 1.0, которые экспортируются системной библиотекой advapi32.dll.

Таблица 6.1. – Группы функций интерфейс криптопровайдера CryptoSPI

Группа функций	Описание
Инициализация и параметры криптопровайдера	Функции инициализации и параметры. Предназначены для установления сессии прикладного ПО с криптопровайдером и получения (установки) параметров сессии.
Функции генерации и работы с ключами	Функции генерации и работы с ключами предназначены для генерации парных и обработки различных типов ключей, используемых в криптопровайдере.
Функции шифрования/расшифрования данных	Функции шифрования/расшифрования данных предназначены для выполнения операций, шифрования и расшифрования данных.
Функции хэширования и ЭЦП	Функции хэширования и ЭЦП предназначены для вычисления значения хэш-функции и формирования/проверки ЭЦП данных.

Таблица 6.6. – Группы функций интерфейса CryptoAPI 2.0

Группа функций	Описание
Кодирующие/декодирующие функции	Предназначены для перевода объектов из структур языка С в ASN.1 структуры и обратно.
Функции работы с хранилищами сертификатов	Установка и отображение свойств хранилища, открытие для записи и закрытие, добавление хранилища в коллекцию и другие функции управления хранилищами сертификатов.
Функции работы с сертификатами	Создание и редактирование сертификата, проверка подписи, работа с хранилищем сертификатов,
Функции работы со списком отозванных сертификатов (COC или CRL)	Добавление информации о сертификате в COC, поиск сертификата в COC, работа с хранилищем
Список доверенных сертификатов (CTL)	Добавление информации о сертификате в CTL, поиск сертификата в CTL, работа с хранилищем



20.Состав и назначение программных компонент Удостоверяющего Центра. Центр сертификации.

Основная задача УЦ – поддержание жизненного цикла сертификатов открытого ключа с момента их выдачи до момента их уничтожения. В общем виде УЦ может состоять из след.компонентов:

Криптографическое ядро;

Центр Сертификации;

Центр Регистрации;

АРМ Абонента;

АРМ разбора конфликтных ситуаций;

Сетевой справочник или Репозиторий.

Центр Сертификации

Центр сертификации (ЦС) - компонент УЦ, предназначенный для формирования сертификатов открытых ключей пользователей и администраторов Удостоверяющего центра, списков отозванных сертификатов, хранения эталонной базы сертификатов и списков отозванных сертификатов.

ЦС взаимодействует только с Центром Регистрации или несколькими Центрами Регистрации по отдельному сегменту локальной сети с использованием защищенного сетевого протокола. При этом ЦС является наиболее критичным компонентом УЦ и для него должен быть обеспечен максимальный уровень защиты.

Функционально Центр Сертификации должен обеспечивать:

- генерацию ключей и сертификата открытого ключа уполномоченного лица УЦ;
- смену ключей и сертификата открытого ключа уполномоченного лица УЦ;
- формирование сертификата открытого ключа по запросам Центра Регистрации;
- ведение базы данных сертификатов с предоставлением доступа к ней ограниченному кругу компонентов системы;
- изменение базы данных сертификатов по запросам от Центра Регистрации (аннулирование (отзыв) сертификата, приостановление действия сертификата, возобновление действия сертификата);
- формирование списка аннулированных (отозванных) сертификатов открытых ключей по запросам Центра Регистрации;
- протоколирование работы Центра Сертификации.

21.Состав и назначение программных компонент Удостоверяющего Центра. Центр регистрации.

Основная задача УЦ – поддержание жизненного цикла сертификатов открытого ключа с момента их выдачи до момента их уничтожения. В общем виде УЦ может состоять из след.компонентов:

Криптографическое ядро;

Центр Сертификации;

Центр Регистрации;

АРМ Абонента;

АРМ разбора конфликтных ситуаций;

Сетевой справочник или Репозиторий.

Центр Регистрации

Центр Регистрации - компонент УЦ, предназначенный для хранения регистрационных данных пользователей, запросов на сертификаты и сертификаты пользователей, предоставления интерфейса взаимодействия пользователей с Удостоверяющим Центром.

Центр Регистрации взаимодействует с Центром Сертификации по отдельному сегменту локальной сети с использованием защищенного сетевого протокола. Взаимодействие пользователей с Удостоверяющим центром обеспечивается за счет использования программного модуля АРМ Абонента.

Центр Регистрации является единственной точкой входа (регистрации) пользователей в системе. Только зарегистрированный в Центре Регистрации пользователь может получить сертификат на свой открытый ключ в Удостоверяющем Центре.

К функциям Центра Регистрации относятся:

- обеспечение аутентификации приложений и пользователей при обращении к Центру Регистрации;
- ведение Базы Данных пользователей, которые зарегистрированы в УЦ
- управление шаблонами сертификатов
- взаимодействие с Центром Сертификации и внешними приложениями
- управление режимами работы Удостоверяющего Центра по регистрации и управлению ключами и сертификатами;
- обеспечение выполнения Центром Регистрации в автоматическом режиме следующих задач: оповещение пользователей об истечении срока действия сертификатов, оповещение пользователей о необходимости замены ключей, оповещение администратора о возникновении критических ситуаций, получение списка отозванных сертификатов от соответствующего Центра Сертификации, удаление зарегистрированных пользователей, не имеющих ни одного действующего сертификата открытого ключа;
- протоколирование работы Центра Регистрации.

22.Состав и назначение программных компонент Удостоверяющего Центра. АРМ абонента.

Основная задача УЦ – поддержание жизненного цикла сертификатов открытого ключа с момента их выдачи до момента их уничтожения. В общем виде УЦ может состоять из след.компонентов:

Криптографическое ядро;

Центр Сертификации;

Центр Регистрации;

АРМ Абонента;

АРМ разбора конфликтных ситуаций;

Сетевой справочник или Репозиторий.

АРМ Абонента

АРМ Абонента предназначен для выполнения организационно-технических мероприятий, связанных с управлением личной ключевой информацией и сертификатами (формирование рабочих ключей и сертификатов, отзыв сертификатов, установка списка отозванных сертификатов), а также организации защищенного документооборота между конечными пользователями Удостоверяющего Центра путем предоставления доступа к функциям шифрования/дешифрования информации и формированием/проверкой электронной цифровой подписи под данными.

АРМ Абонента непосредственно взаимодействует с ближайшим к нему по иерархии Центром Регистрации путем пересылки по каналу связи сообщений формата PKCS, содержащих следующую информацию:

заявки на сертификат, поступающие от АРМ Абонента в ЦР;

сертификаты открытого ключа, поступающие от ЦП в АРМ Абонента

сообщения о компрометации ключа от АРМ Абонента в ЦР;

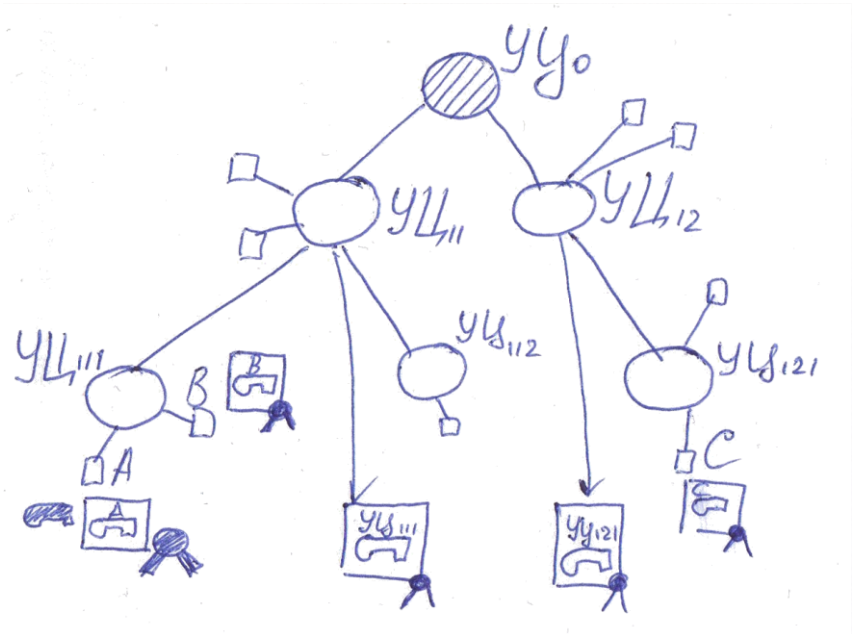
актуальный список отозванных сертификатов, регулярно пересылаемый ЦР на рабочие места абонентов.

К основным функциям АРМ Абонента относятся:

- графический пользовательский интерфейс, позволяющий использовать базовые криптографические функции, реализованные в том криптоядре, которое выбрано для использования в Удостоверяющем Центре
- формирование заявки на сертификат по стандарту RSA PKCS#10;
- обеспечение возможности получения пользователем выпущенных сертификатов;
- вывод электронного сертификата открытого ключа пользователя на бумажный носитель;
- работа с ключевой информацией: генерация рабочих ключевых шифрования и ЭЦП пользователя, а также сессионных ключей симметричного шифрования;
- работа с локальным хранилищем сертификатов других абонентов: ведение хранилища сертификатов открытых ключей, импорт сертификатов в справочник сертификатов, экспорт собственных сертификатов на внешний носитель, проверка ликвидности сертификата, поиск сертификата в хранилище;
- формирование заявки на отзыв сертификата по стандарту RSA PKCS#7;
- выполнение функций администрирования списка отозванных сертификатов;
- запрос сертификата требуемого абонента и информации о статусе сертификата из Репозитория или Центра Регистрации;
- обеспечение взаимодействия с программными модулями Центр Регистрации, Репозиторий и Центр Сертификации в соответствии с регламентом Удостоверяющего Центра;
- обеспечение возможности выбора пользователем сертификата, с помощью которого будет осуществляться взаимодействие с ЦР;

23.Модели доверия УЦ. Иерархическая модель.

Понятие доверия является в PKI краеугольным, то есть вся система с ключами и сертификатами работает в случае доверяя пользователей некой 3 доверенной стороне (УЦ). Поскольку Удостоверяющих Центров может быть больше 1, то схема их взаимодействия порождает различные модели доверия(иерархическая, сетевая,мостовая).



Иерархическая модель.
УЦ0-корнвой УЦ сам себе выдает сертификат.
В случае, когда между собой взаимодейств-т пользователи одного и того же УЦ, например, А и В, проблем с проверкой подлинности сертификатов нет. А и В доверяют УЦ111 и имеют его корневой сертификат, поэтому могут проверить сертификат любого пользователя УЦ 111.
При взаимодействии а и С ситуация усложняется: А не доверяет УЦ 121, т.к.не я вяется его пользователем и не имеет его сертификата. Аналогично, С не доверяет УЦ 111.Решение:

1. А обращается в свой родной УЦ и узнает, какой УЦ выпускал для него сертификат: УЦ11 является родителем.
2. А запрашивает сертификат УЦ11.
3. Повторяя пункты 1 и 2 пользователь добирается до коренного УЦ0 и получает его сертификат. Процедура м.б.проще, если при регистрации А ему сразу выд-ся вся цепочка сертификатов.
4. А узнает, какая цепочка сертификатов нужна для проверки С. Эта информация содержится в самом сертификате С.
5. Зная корневой сертификат УЦ0, проверяется следующий сртификат в цепочке
6. Убедившись, что сертификат УЦ12 верен, с его помощью проверяется сертификат УЦ121, а затем и сертификат С.

Т.о. можно проверить подлинность сертификатов любого пользователя такой модели.

+Сравнительная простота реализации

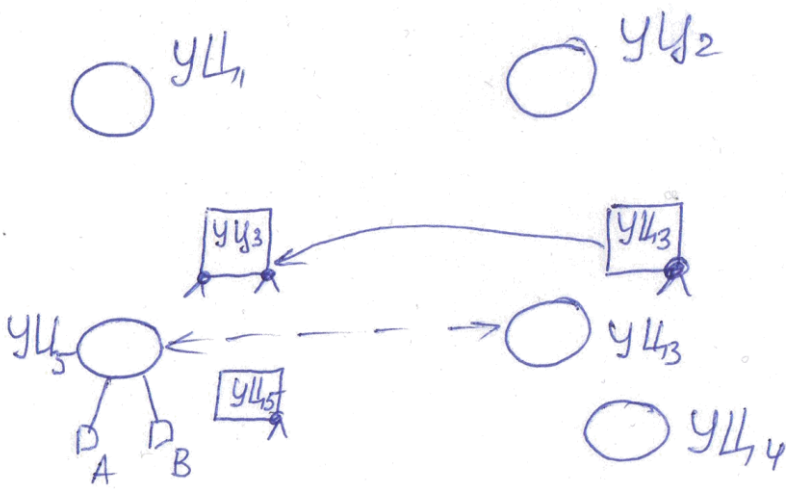
+Удобство наложения такой модели на структуру ведомств

-В случае компрометации любого УЦ, сертификаты всех стоящих ниже по иерархи УЦ и пользователей становятся недействительными.

24.Модели доверия УЦ. Сетевая модель.

Понятие доверия является в PKI краеугольным, то есть вся система с ключами и сертификатами работает в случае доверяя пользователей некой 3 доверенной стороне (УЦ). Поскольку Удостоверяющих Центров может быть больше 1, то схема их взаимодействия порождает различные модели доверия(иерархическая, сетевая,мостовая).

Сетевая модель.



В данной модели все УЦ являются независимыми, никто никому не подчиняется. Пользователи одного и того же УЦ легко друг с другом взаимодействуют, т.к.доверяют одному и тому же УЦ. Пользователи разных УЦ между собой взаимодействовать не могут, т.к.нет доверия между разными УЦ и проверить сертификаты другого УЦ невозможно. Решением проблемы явл-ся исп-е мех-ма кросс-сертификации.

Кросс-сертификация – процесс установки доверит отношений между ранее не связанными УЦ. Он заключается в использовании кросс-сертификатов:

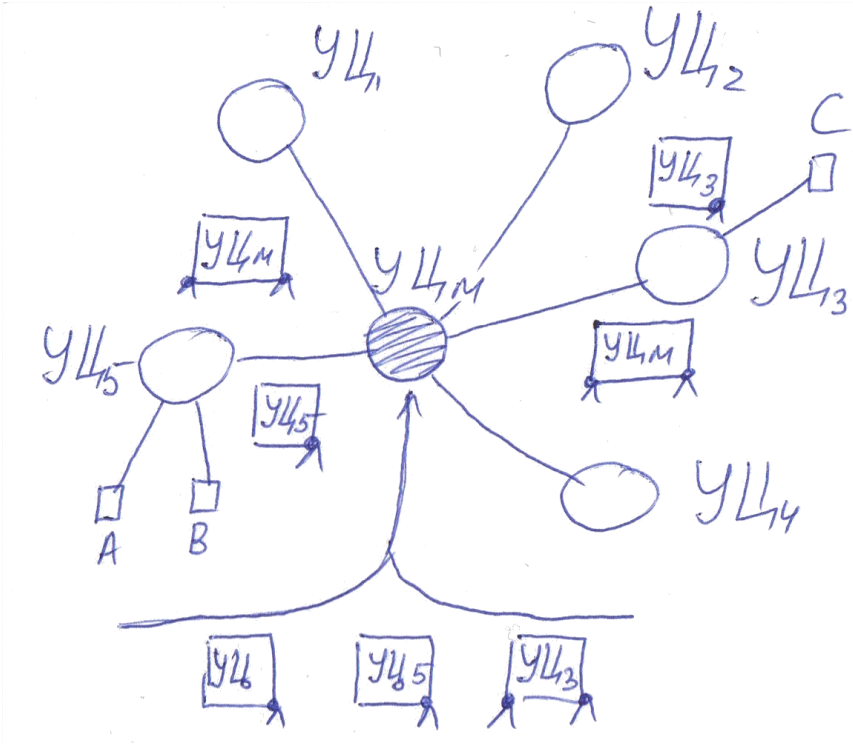
1. УЦ5 запрашивает у УЦ3 его самоподписанный сертификат.Данный серт-т должен быть получен защищенным способом(лично или по защищенному каналу связи)
2. Уц5 подпис-т полученный от УЦ3 сертификат своим закрытым ключом, тем самым подтверждая, что он доверяет УЦ3. Дважды подписанный сертификат и есть кросс-сертификат.
3. Аналогично поступает УЦ3.
4. Пользователь А желает проверить подлинность серт-та С. Для этого он берет кросс-сертификат в УЦ5, проверяет его подлинность с помощью сертификата УЦ5. Т.о.А убежд-ся, что открытый ключ УЦ3 достоверен и им можно пользоваться.
5. С помощью уже достоверного сертификата УЦ3 и ключа, который в нем лежит, проверяется подлинность сертификата С.

+ Компрометация любого из УЦ фатальна только для его пользователей и не касается других

- Для большой сети из N УЦ каждому УЦ придется получить N-1 кросс-сертификат.

25.Модели доверия УЦ. Мостовая модель.

Понятие доверия является в PKI краеугольным, то есть вся система с ключами и сертификатами работает в случае доверяя пользователей некой 3 доверенной стороне (УЦ). Поскольку Удостоверяющих Центров может быть больше 1, то схема их взаимодействия порождает различные модели доверия(иерархическая, сетевая,мостовая).



Мостовая модель.

1. УЦ3 и УЦ5 проходят процедуру кросс-сертификации с мостовым УЦ, каждый из них получает себе сертификат УЦм и подписывает его своим закрытым ключом, а УЦм получает себе сертификаты УЦ3 и УЦ5 и подписывает их. (Кросс-сертификация – процесс установки доверит отношений между ранее не связанными УЦ. Он заключается в использовании кросс-сертификатов).
2. А проверяет сертификат УЦм с помощью обычного сертификата УЦ5, убеждаясь, что УЦм достоверный.
3. Используя сертификат УЦм, А проверяет подлинность кросс-сертификата УЦ3 и убеждается, что сертификат УЦ3 достоверный.
4. С помощью сертификата УЦ3 проверяется сертификат С.

+избегается N-1 кросс-сертификация

+Позволяет наладить связь между несколькими УЦ

+Повышает отказоустойчивость системы

-Компрометация УЦм заставит заново строить все связи, но при этом пользователи не пострадают.

26.Протокол TLS/SSL. Уровень диалога (рукопожатия).

Основная функция - обеспечение защиты и целостности данных между 2 взаимодействующими приложениями, одно из кот клиент, а другое – сервер.

Перечислим задачи протокола TLS в порядке их приоритета:

- Криптографическая безопасность: TLS должен использоваться для установления безопасного соединения между двумя участниками.
- Интероперабельность: независимые разработчики могут создавать приложения, которые будут взаимодействовать по протоколу TLS, что позволит устанавливать безопасные соединения.
- Расширяемость: TLS формирует общий каркас, в который могут быть встроены новые алгоритмы открытого ключа и симметричного шифрования.
- Отн эфф-ть: криптогр операции интенсивно используют ЦП, особенно операции с откр ключом. Для этого вводится понятие сессии, для кот опр-ся алг-мы и их пар-ры. В рамках 1 сессии м б создано неск соединений (например, TCP).

Протокол состоит из двух уровней. Нижним уровнем, расположенным выше некоторого надежного протокола (а именно, протокола TCP) является протокол Записи. Одним из протоколов более высокого уровня является протокол Рукопожатия, который использует протокол Записи в качестве транспорта для ведения переговоров о пар-рах без-ти.

Протокол Рукопожатия состоит из трех протоколов, использование кот позв-т уч-кам согласовать пар-ры без-ти для протокола Записи, аутентифицир-ть друг друга, дог-ся о пар-рах безоп-ти и сооб-ть друг другу о возн-и тех или иных ош-к.

Протокол изменения шифрования состоит из единств сообщения, кот зашифр-но и сжато, как определено в текущем состоянии соединения.

Alert протокол является одним из протоколов, лежащих выше протокола Записи. Содержимым протокола является либо фатальное, либо предупреждающее сообщение. Фатальное сообщение должно приводить к немедл разрыву данного соед-я.Протокол Рукопожатия включает следующие шаги:

1. Обмен сообщениями Hello для согл-я алг-мов, обмен случ знач-ми и проверка возобновл-ти сессии.
2. Обмен необходимыми криптограф пар-рами, кот позволяют клиенту и серверу согласовать премастер-секрет (клиент посылает премастер секрет серверу).
3. Обмен сертификатами и криптографической информацией, что позволяет клиенту и серверу аутентифицировать друг друга.
4. Предоставление параметров безопасности на уровень Записи.
5. Возможность клиенту и серверу проверить, что они вычислили одни и те же параметры безопасности и что Рукопожатие произошло без вмеш-ва злоум-ка.

В рез-те вып-я протокола Рукопожатия будут созданы след.элементы сессии:

Идентификатор сессии	Произвольная посл-ть байтов, выбираемая сервером для идентификации акт или возобновляемого состояния сессии
Серт-ат уч-ка	X.509 v3 сертификат участника. Этот элемент м б нулевым
Метод сжатия	Алг-м, используемый для сжатия данных перед шифрованием
Набор алгоритмов	Алгоритм симметр шифр-я данных (null, DES и т.д.), MAC-алг-м (MD5 или SHA) и разл криптогр атрибуты
Мастер-секрет	48-байтный секрет, разделямый клиентом и сервером

27.Протокол TLS/SSL. Уровень записи.

Основная функция - обеспечение защиты и целостности данных между 2 взаимодействующими приложениями, одно из кот клиент, а другое – сервер.
Протокол состоит из 2 уровней. Нижним уровнем, располож-м выше некоторого надежного протокола (а именно, протокола TCP) является протокол Записи. Одним из прот-в более высок ур-я яв-ся протокол Рукопожатия, кот исп-т прот-Записи в качестве транспорта для ведения переговоров о пар-рах безопасности.

Протокол Записи обеспечивает безопасность соединения, которая основана на следующих двух свойствах:

Конфиденциальность соединения. Для защиты данных используется один из алгоритмов симметричного шифрования. Ключ для этого алгоритма создается для каждой сессии и основан на секрете, о котором договариваются в протоколе Рукопожатия. Протокол Записи также может использоваться без шифрования.

Целостность соединения. Обеспечивается проверка целостности сообщения с помощью MAC с ключом. Для вычисления MAC используются безопасные хэш-функции SHA-1 и MD5. Протокол Записи используется для инкапсуляции различных протоколов более высокого уровня.

Протокол Записи фрагментирует сообщение на блоки нужной длины, осуществляет сжатие данных, применяет MAC и зашифровывает их, после чего результат передается по сети. На другом конце соединения полученные данные дешифруются, проверяется их целостность, далее они декомпрессируются, дефрагментируются и передаются протоколам более высокого уровня.

Состояния соединения.Вводится понятие состояния соединения, которое определяет параметры выполнения протокола Записи. Такими параметрами являются алгоритм сжатия, алгоритм шифрования и MAC-алгоритм, а также параметры этих алгоритмов. Для каждого направления (соответственно чтение или запись) параметры соединения могут различаться. Существует четыре состояния соединения: текущие состояния чтения и записи и ожидаемые состояния чтения и записи.

Фрагментация.Уровень Записи фрагментирует блоки в записи TLSPlaintext, поддерживая цепочки данных длиной не более 2^{14} байт. Границы записей протоколов более высокого уровня не сохраняются на уровне Записи, т.е. несколько сообщений протокола более высокого уровня некоторого ContentType могут быть размещены в одной записи TLSPlaintext или единственное сообщение может быть фрагментировано в несколько записей.

Компрессия и декомпрессия. Все записи сжимаются с использованием алгоритма сжатия, определенного в текущем состоянии сессии. Алгоритм сжатия преобразует TLSPlaintext-структуру в TLSCompressed-структуру. Если функция декомпрессии определяет, что длина декомпрессированного фрагмента превышает 2^{14} байтов, возникает фатальная ошибка декомпрессии.

Защита полезной информации записи. Функции шифрования и MAC преобразуют TLSCompressed-структуру в TLSCiphertext. Функции дешифр-я вып-т обратные преобразования. Применение MAC включает последовательный номер, поэтому потерю или повтор сообщений всегда можно обнаружить.

28.Алгоритм работы Ipsec.IP Security - это комплект протоколов, касающихся вопросов шифрования, аутентификации и обеспечения защиты при транспортировке IP-пакетов.IPSec предлагает стандартный способ аутентификации и шифрования соединений между сообщающимися сторонами. Чтобы обеспечить защиту связей, средства IPSec используют стандартные алгоритмы шифрования и аутентификации, называемые преобразованиями. Чтобы указать согласуемые параметры, в IPSec используются ассоциации защиты.
Ассоциация защиты (Security Association — SA) представляет собой согласованную политику или способ обработки данных, обмен которыми предполагается между двумя устройствами сообщающихся сторон. Действ пар-ры SA сохр в БД АЗ обеих сторон.
Протокол **IKE** (Internet Key Exchange — обмен Internet-ключами) является гибридным протоколом, обеспечивающим аутентификацию сторон IPSec, согласование параметров ассоциаций защиты IKE и IPSec, а также выбор ключей для алгоритмов шифрования, используемых в рамках IPSec.**Действие IPSec**:
1. Начало процесса IPSec. Трафик, которому требуется шифрование в соответствии с политикой защиты IPSec, согласованной сторонами IPSec, начинает IKE-процесс.
2. Первая фаза IKE. IKE-процесс выполняет аутентификацию сторон IPSec и ведет переговоры о пар-рах ассоциаций защиты IKE, в результате чего создается защищенный канал для ведения переговоров о пар-рах ассоциаций защиты IPSec в ходе второй фазы IKE.
3. Вторая фаза IKE. IKE-процесс ведет переговоры о параметрах ассоциации защиты IPSec и устанавливает соотв ассоциации защиты IPSec для устройств сообщающихся сторон.
4. Передача данных. Происходит обмен данными между сообщающимися сторонами IPSec, кот основывается на пар-рах IPSec и ключах, хранимых в БД ассоциаций защиты.
5. Завершение работы туннеля IPSec. Ассоциации защиты IPSec завершают свою работу либо в результате их удаления, либо по причине превышения предельного времени их сущ-я.

IPSec состоит из двух главных протоколов защиты и множества протоколов поддержки. Протоколы защиты:

Протокол AH (Authentication Header — заголовок аутентификации). Протокол зашиты, обеспечивающий аутентификацию и (в качестве опции) сервис выявления воспр-я. Протокол AH действует как цифровая подпись и гарантирует, что данные в пакете IP не будут несанкционированно изменены. Не обеспечивает сервис шифрования и дешифрования данных. Данный протокол может исп-ся или самост-но, или совместно с протоколом ESP.

Протокол ESP (Encapsulating Security Payload — включающий защиту полезный груз). Протокол защиты, обеспеч конфиденциальность и защиту данных, а также (в качестве опции) сервис аутентификации и выявления воспр-я. Используются для шифрования полезного груза IP-пакетов. Протокол ESP может использоваться самостоятельно или совместно с AH.

Протоколы поддержки:**Стандарт DES** (Data Encription Standard — стандарт шифрования данных). Алг-м шифр-я и дешифр-я данных пакетов. Для DES используется 56-битовый ключ, кот обеспеч более надежное шифрование. Алгоритм DES явл симметричным алг-ом шифрования, для кот требуются идентичные секретные ключи шифр-я в устройствах каждой из сообщающихся сторон IPSec.

"Тройной" DES (3DES). Вариант DES, основанный на использовании трех итераций стандартного DES с тремя разными ключами, что практически утраивает стойкость DES. Алгоритм 3DES используется в рамках IPSec для шифр-я и дешиф-я потока данных. Данный алгоритм использует 168-битовый ключ, что гарантирует высокую надежность шифр-я. IKE и IPSec используют алгоритм 3DES для шифр-я сообщений.При преобразовании IPSec исп-ся также два стандартных алгоритма хэширования, обеспечивающих аутентификацию данных.

Алгоритм MD5 (Message Digest 5). Алгоритм хэширования, применяемый для аутентификации пакетов данных. Хэширование предст собой процесс одностороннего (т.е. необратимого) шифр-я, в рез кот для поступающего на вход сообщения произвольной длины получается вывод фикс длины. IKE, AH и ESP используют MD5 для аутентификации данных.

Алгоритм SHA-1 Алгоритм хэширования, используемый для аутентификации пакетов данных.