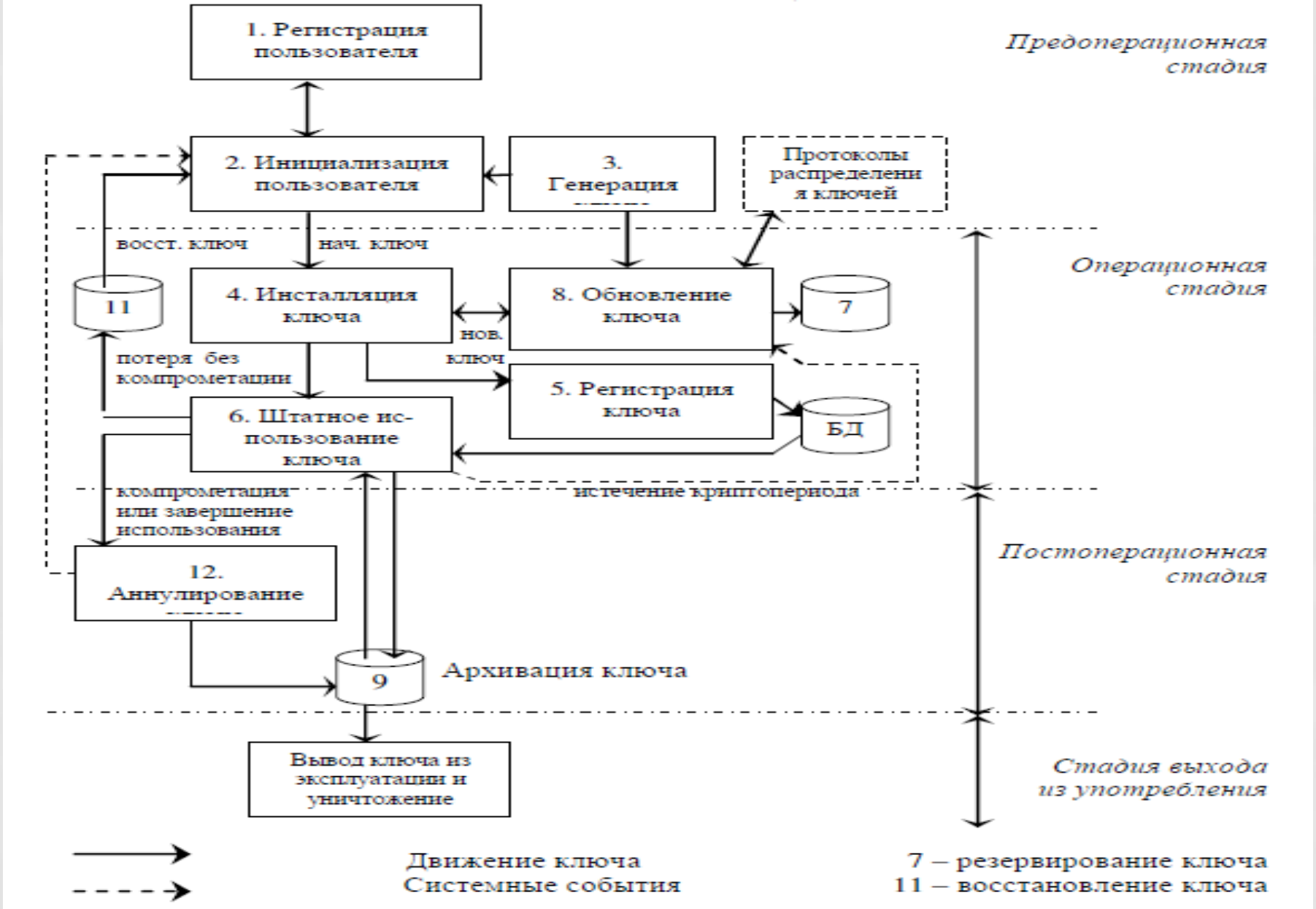
Управление криптографическими секретными ключами

Цель управления ключами

- Определена в международном стандарте ISO/IEC 11770 – Key management
- Цель управления – обеспечение секретности, подлинности, целостности криптографических ключей на всех этапах жизненного цикла
- Жизненный цикл – последовательность состояний, в которых пребывает ключевой материал за время своего существования в криптосистеме: генерация, хранение, распространение, уничтожение и др.
- Управление ключами - совокупность процедур и процессов, сопровождающих жизненный цикл ключей в криптосистеме

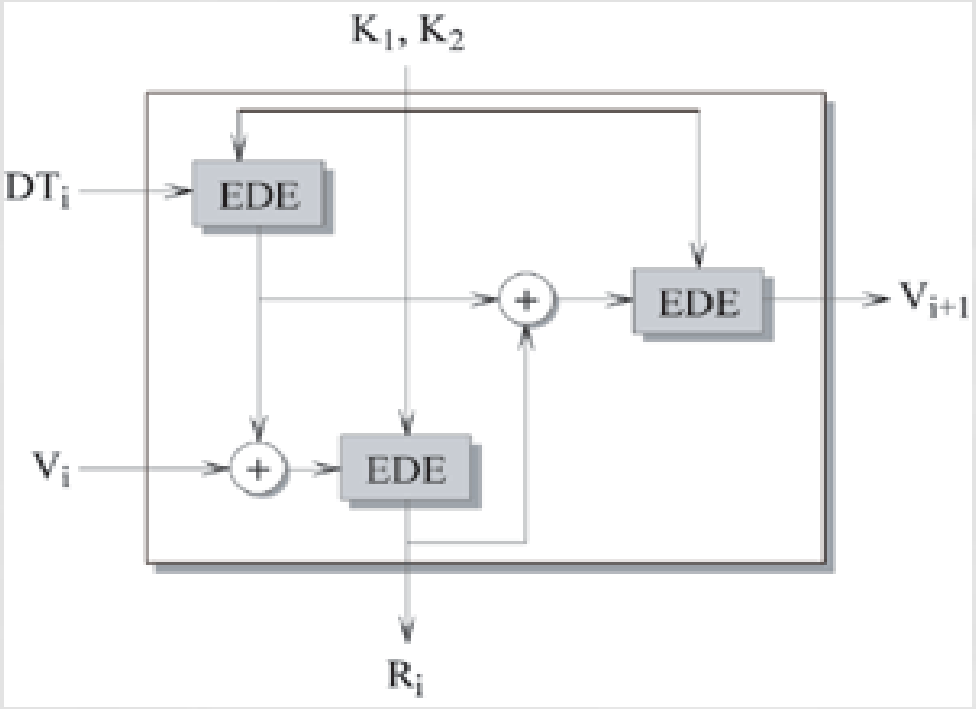
Схема жизненного цикла ключей



Задачи предоперационной стадии

- **Регистрация пользователя:** обмен первоначальной ключевой информацией с пользователем, такой, как общие пароли или PIN-коды, путём личного общения или пересылки через доверенного курьера
- **Инициализация:** пользователь устанавливает аппаратное оборудование и/или программные средства в соответствии с установленными рекомендациями и правилами
- **Генерация ключей:** *создание и обеспечение необходимых криптографических качеств ключей. Ключи могут генерироваться как самостоятельно пользователем, так и специальным защищенным элементом системы, а затем передаваться пользователю по защищенному каналу*

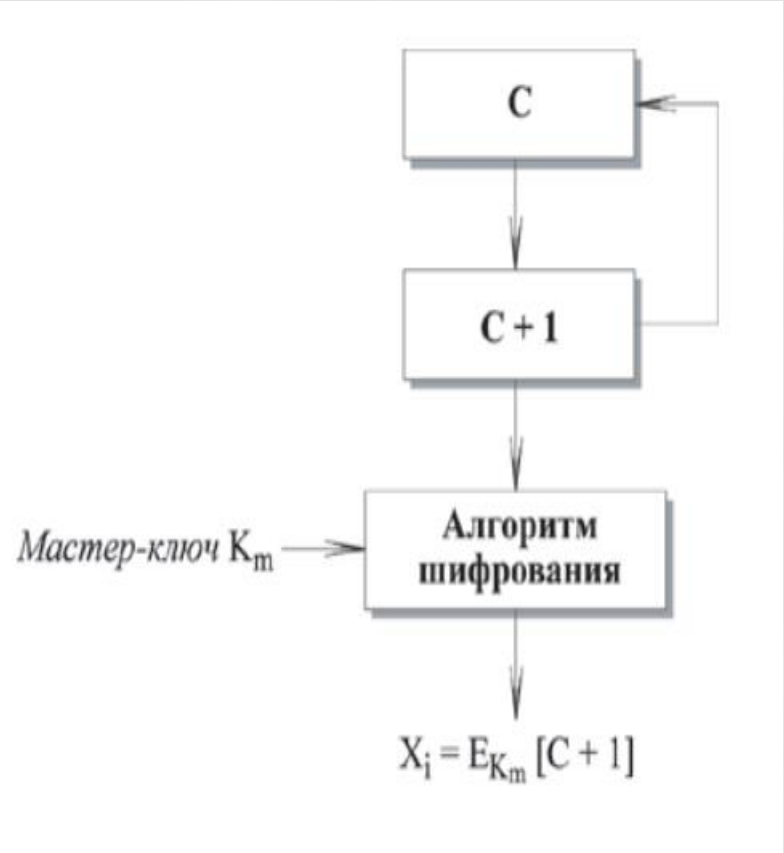
Генератор ключей стандарта ANSI X9.17



EDE – Encrypt-Decrypt-Encrypt

- Один из лучших генераторов. Применяется в приложениях финансовой безопасности и PGP
- DTi - значение даты и времени на начало i-ой стадии генерации
- Vi - начальное значение для i-ой стадии генерации.
- Ri - псевдослучайное число, созданное на i-ой стадии генерации.
- K1, K2 - ключи, используемые на каждой стадии.
- Тогда:
  - $R_i = EDE_{K_1K_2}[EDE_{K_1K_2}[\,DT_i] \oplus V_i\,]$
  - $V_{i+1} = EDE_{K_1K_2}[EDE_{K_1K_2}[\,DT_i] \oplus R_i\,]$

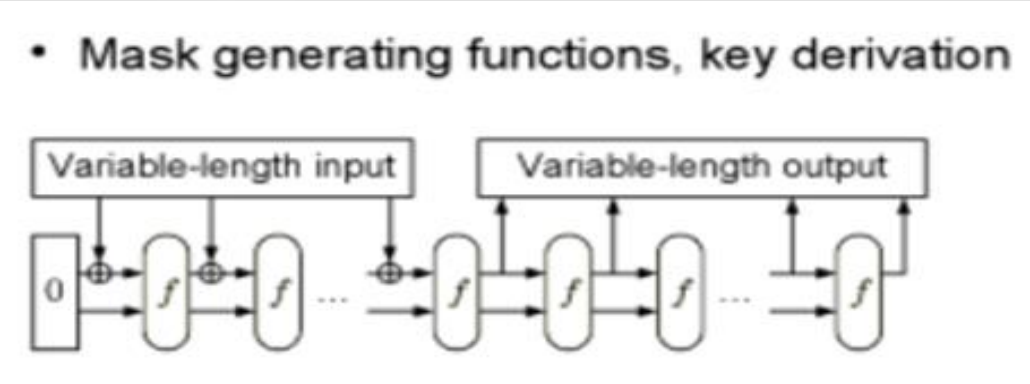
Режим счетчика CTR



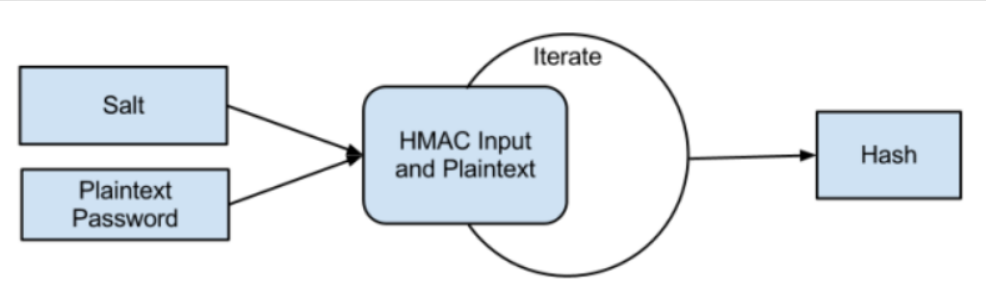
- Применяется для создания ключа сессии из мастер-ключа
- Счетчик инициализируется начальным значением C_0
- Период генератора определяется периодом счетчика
- Выход: k старших бит X_i

CTR - CounTeR

Генератор на основе хэш-функции

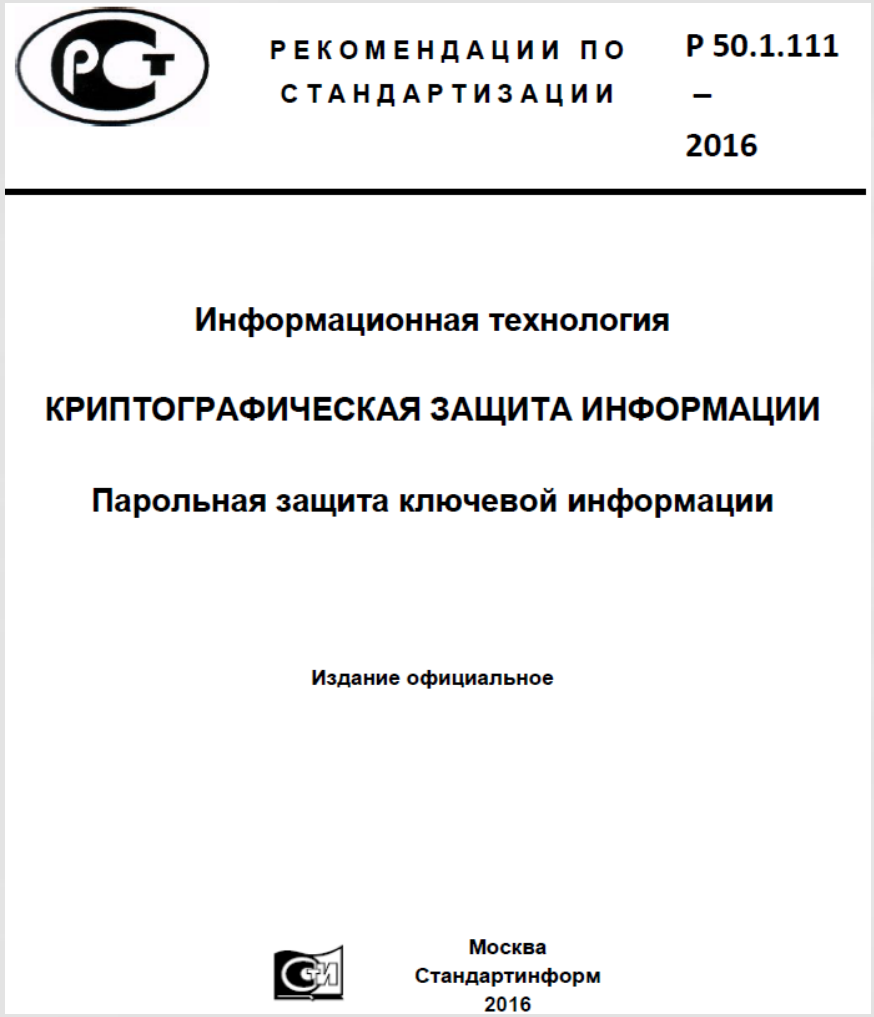


- Хэш-функции Keccak: на стадии «впитывания» входом является пароль, на стадии «отжатия» выход -ключ переменной длины. Может применяться для генерации симметричных ключей из паролей.



- Функция получения ключа на основе пароля (PBKDF) – использует итерационную схему на основе HMAC

Алгоритм выработки ключа из пароля



РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

Р 50.1.111 – 2016

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Парольная защита ключевой информации

Издание официальное

Москва
Стандартинформ
2016

- Функция диверсификации:

$$DK = \mathrm{PBKDF2}\,(P, S, c, dkLen).$$

- Алгоритм:

Вычисляют $n = \lceil dkLen / 64 \rceil$.

Для каждого i от 1 до n вычисляют набор значений:

$$\begin{aligned} U_1(i) &= \mathrm{HMAC\_GOSTR3411}\,(P, S \,||\, \mathrm{Int}(i)) \\ U_2(i) &= \mathrm{HMAC\_GOSTR3411}\,(P, U_1) \\ &\dots \\ U_c(i) &= \mathrm{HMAC\_GOSTR3411}\,(P, U_{c-1}). \end{aligned} \qquad (2)$$

$$T(i) = U_1(i) \oplus U_2(i) \oplus \dots \oplus U_c(i). \qquad (3)$$

Ключ DK вычисляют как конкатенацию байтовых строк $\{T(i)\}$ с последующим усечением полученной последовательности до длины $dkLen$ выходной последовательности:

$$DK = R^{n \cdot 64}_{dkLen}\,(T(1)||T(2)||\cdots||T(n)). \qquad (4)$$

Задачи операционной стадии

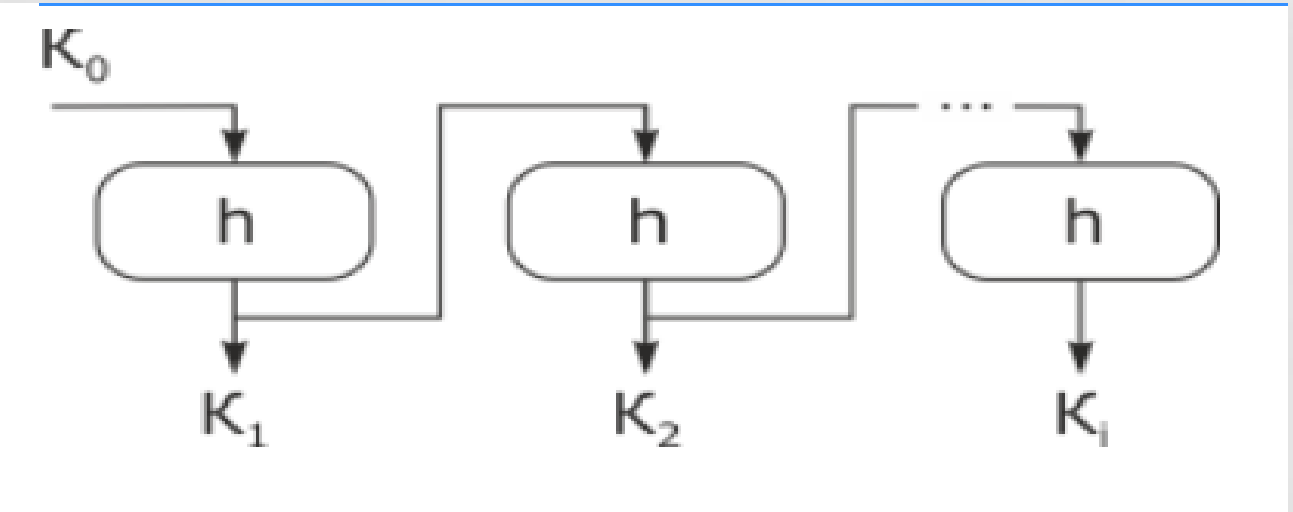
- **Инсталляция:** ключи устанавливаются в оборудование тем или иным способом. При этом первоначальная ключевая информация, полученная на стадии регистрации пользователей, может либо непосредственно вводиться в оборудование, либо использоваться для установления защищенного канала, по которому передается ключевая информация. Эта же стадия используется в последующем для смены ключевой информации
- **Регистрация ключа:** ключевая информация связывается регистрационным центром с именем пользователя и сообщается другим пользователям ключевой сети
- **Штатное использование**: шифрование и расшифрование данных и других ключей
- **Обновление:** *замена ключа осуществляется до истечения его срока действия и включает процедуры, связанные с генерацией ключей, протоколами обмена ключевой информацией между корреспондентами, а также с доверенной третьей стороной*

Ограничение срока жизни ключа

- Срок жизни ключа (***key lifetime***) — объем данных, который можно "безопасно" обработать на одном ключе, т.е. без возможности скомпрометировать любую конфиденциальную информацию
- Нагрузка на ключ — это объем данных (количество блоков размера n), обработанных на одном ключе
- Практика показывает, что обработка большого количества сообщений на одном ключе и накопление результатов обработки может привести к потере стойкости (к компрометации ключа, дешифрованию сообщений):
  - Методы криптоанализа, основанные на свойствах используемого шифра (например, дифференциальный метод, срабатывают при нагрузке на ключ 2^n
  - Методы криптоанализа, основанные на комбинаторных свойствах используемого режима работы шифра (например, атаки «парадокса дней рождения», срабатывают при нагрузке $2^{n/2}$
  - Методы, криптоанализа основанные на информации, полученной по побочным каналам (измерение энергопотребления, электромагнитного излучения, акустического шума, времени работы алгоритма шифрования). в случае обработке большого количества сообщений позволяют накапливать "опасную" информацию

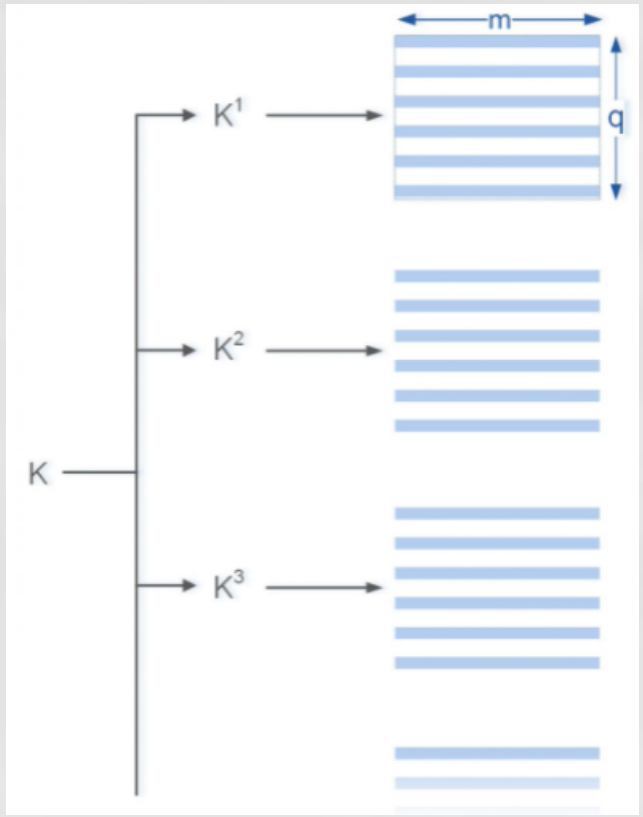
Преобразования ключа (re-keying)

- Это подход, основная идея которого заключается в зашифровании (расшифровании) данных с помощью последовательности ключей, получаемых из первоначально согласованного ключа (начального) путем применения специально подобранных детерминированных преобразований
- Ресурсоемким примером способа преобразования ключей является способ получении нового ключа путем хеширования старого.



Внешнее преобразование ключа (external re-keying)

- Отличием этого подхода является то, что ключ меняется не в процессе обработки одного сообщения, а после обработки некоторого количества целых сообщений
- Применение данного подхода не влияет на внутреннее строение режима и не меняет порядка обработки отдельных сообщений
- Пример способа получения последовательности ключей для любого режима шифра «Кузнечик»:

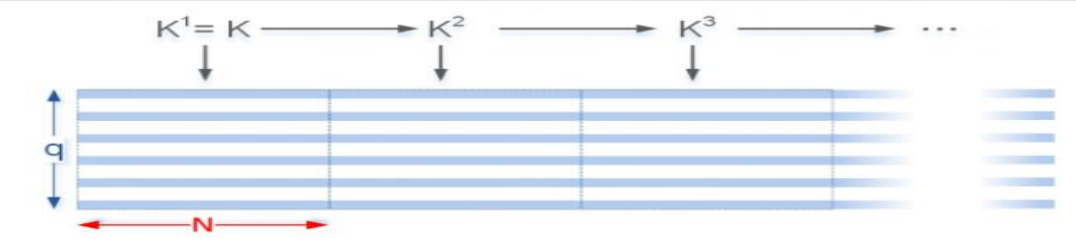


$$K^1 \mid K^2 \ldots \mid K^t = E_K([0]) \mid E_K([1]) \mid \ldots \mid E_K([2t\text{-}1]),$$

где $[i]$ — строка длины 128 бит, которая является двоичным представлением числа i.

Внутреннее преобразование ключа (Internal re-keying)

- Подход заключается в модификации какого-то конкретного режима работы блочного шифра так, чтобы ключ, периодически изменялся по ходу обработки одного файла данных.
- Секция – это последовательность блоков файла, обрабатываемая на одном ключе до его преобразования, при этом такой ключ называется секционным. Размер секции является параметром расширенного режима работы шифра.

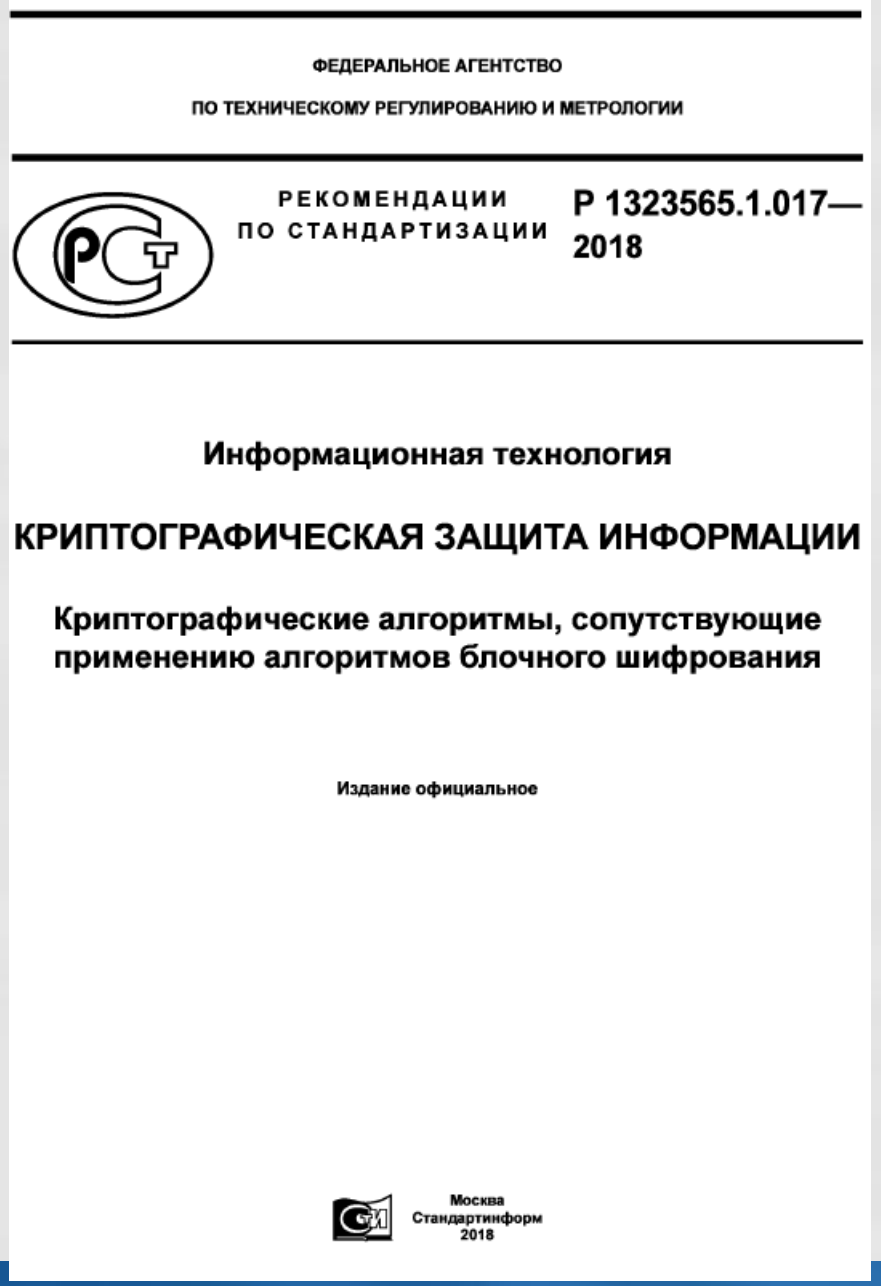


- Пример способа преобразования ключа ACPKM (Advanced Cryptographic Prolongation of Key Material), который применяется к режиму шифрования CTR шифра «Кузнечик»:

$$K^{i+1} = E_{K^i}(W_1) \mid E_{K^i}(W_2),$$

где W_1 и W_2 – некоторые константные строки, а операция "|" – конкатенация.

Первоисточник :



ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

Р 1323565.1.017— 2018

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Криптографические алгоритмы, сопутствующие применению алгоритмов блочного шифрования

Издание официальное

Москва
Стандартинформ
2018

Задачи операционной стадии (продолжение)

- **Резервирование:** создание копии ключевой информации для восстановления ключа на случай обстоятельств, не связанных с компрометацией
- **Восстановление**: восстановление ключа из хранимой копии, в случае, если ключевая информация была уничтожена, но не скомпрометирована (например, из-за неисправности оборудования или из-за того, что оператор забыл пароль)
- **Хранение ключа***: включает процедуры, необходимые для хранения ключа в надлежащих условиях, обеспечивающих его безопасность до момента его замены*

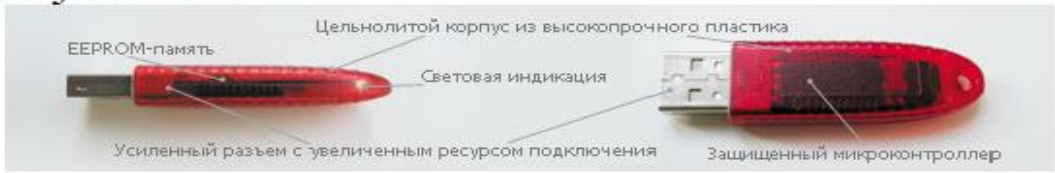
Аппаратные средства хранения ключей

Смарт-карта — устройство для одно- и двухфакторной аутентификации пользователей, хранения ключевой информации и проведения криптографических операций в доверенной среде.

Разница между использованием криптографических токенов или смарт-карт и стандартных флэш-накопителей в том, что при использовании криптографического оборудованием ключ генерируется на самом оборудовании и никогда не экспортируется !

Электронный идентификатор (токен) - компактное устройство в виде USB-брелока, которое служит для авторизации пользователя в сети или на локальном компьютере, защиты электронной переписки, безопасного удаленного доступа к информационным ресурсам, а также надежного хранения персональных данных.

«Рутокен»:



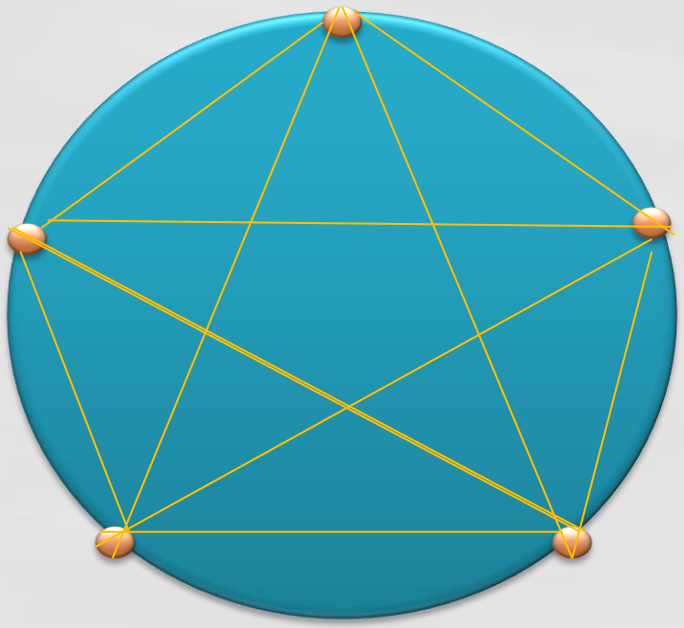
«eToken»:

Задачи постоперационной стадии

- **Архивирование: в** отдельных случаях ключевая информация после её использования для защиты информации может быть подвергнута архивированию для её извлечения со специальными целями (например, расшифровки материалов с грифом ДСП)
- **Аннулирование:** в случае компрометации ключевой информации возникает необходимость прекращения использования ключей до окончания срока их действия. При этом должны быть предусмотрены необходимые меры оповещения абонентов сети.
- **Вывод из эксплуатации:** после окончания сроков действия ключей они выводятся из обращения, и все имеющиеся их копии уничтожаются. При этом необходимо следить, чтобы тщательно уничтожалась и вся информация, по которой возможно их частичное восстановление.

Распределение симметричных ключей

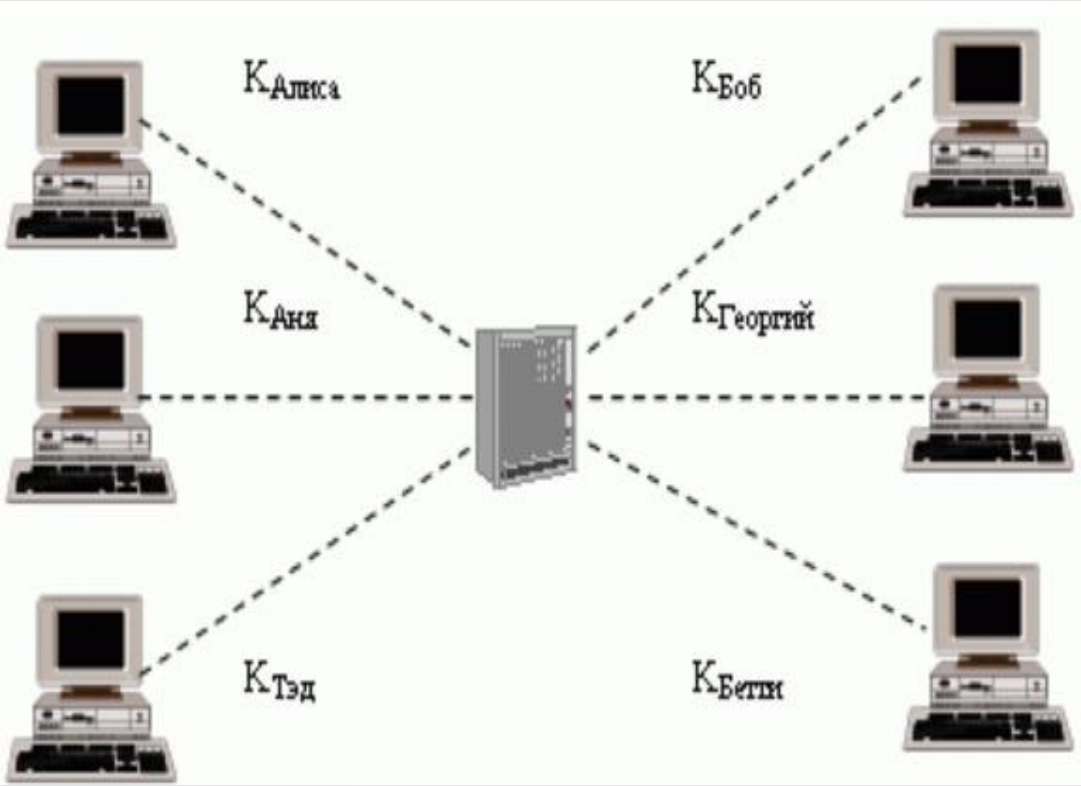
Проблемы распределения симметричных ключей



$$R \sim \frac{N*(N-1)}{2}$$

- Необходим надежный способ первоначального распределения ключей (обмен ключами при личной встрече, доставка спецкурьером, передача частями по разным каналам, по протоколу с центром распределения ключей)
- Ключи должны время от времени меняться для снижения вероятности их компрометации Оптимальным считается использование для каждого сеанса обмена зашифрованными сообщениями своего уникального ключа (*session key)*
- При большом числе взаимодействующих сторон *N* требуется значительные ресурсы *R* для рассылки, хранения и смене ключей

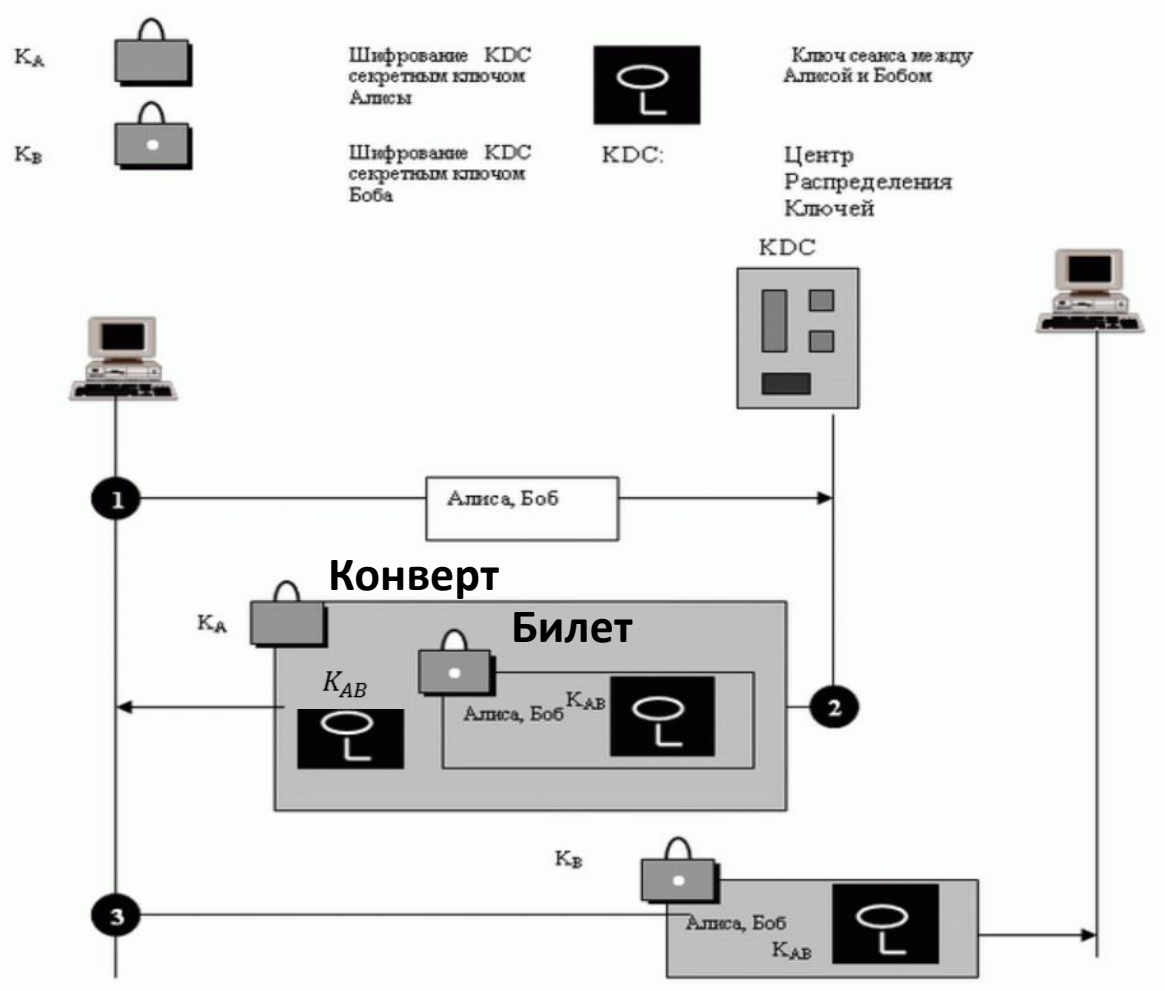
Центр Распределения Ключей: KDC



- KDC- это фактически привлечение третьего лица, которому доверяют
- Каждый абонент устанавливает ключ засекречивания с KDC
- Ключи засекречивания абонентов используется, чтобы подтвердить их подлинность

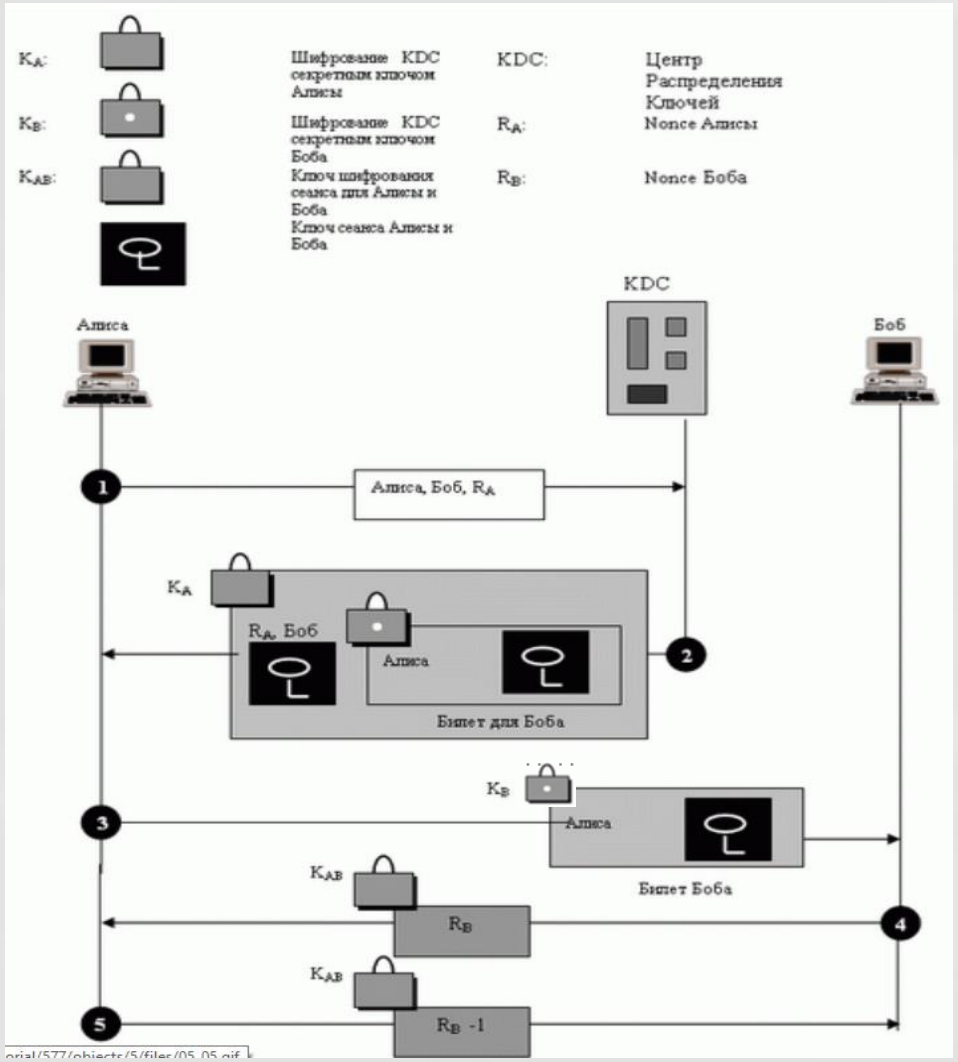
KDC - Key-Distribution Center

Простой протокол получения сеансового ключа



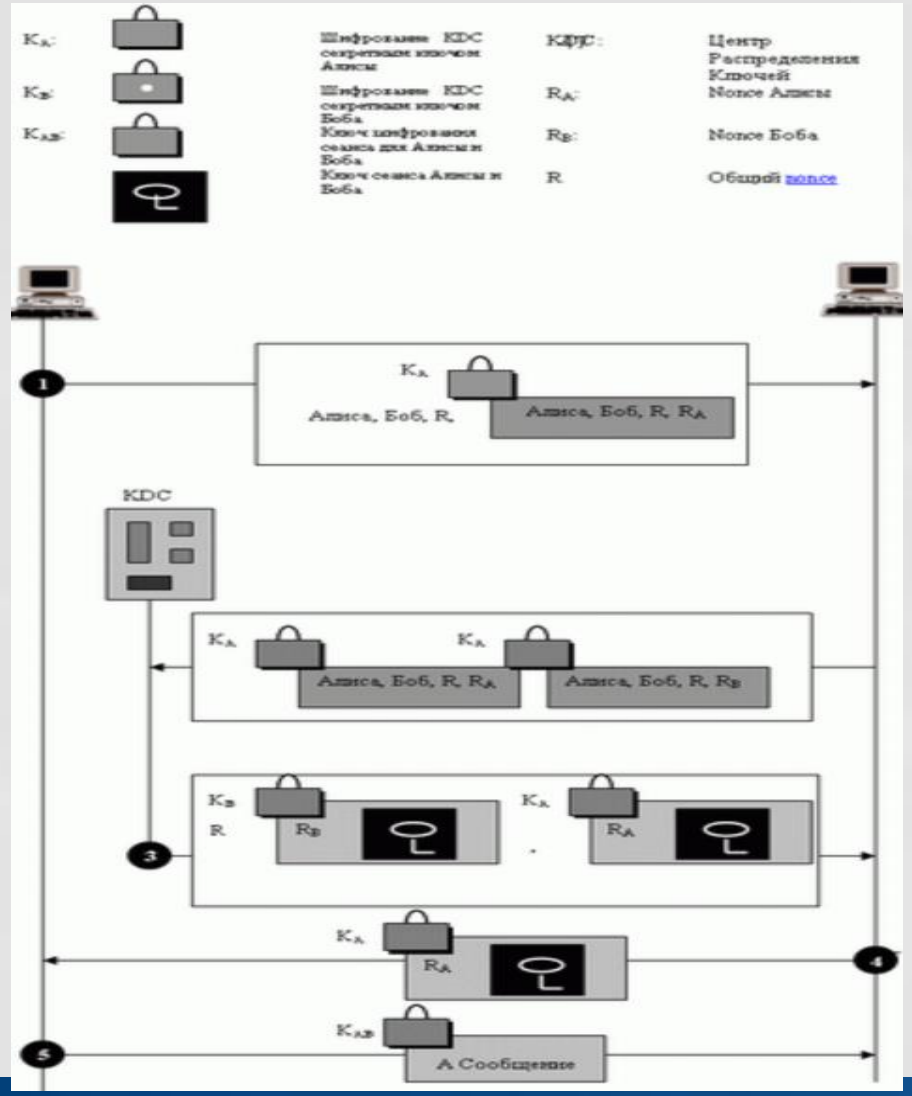
- (1) Открытый запрос в KDC на установление канала связи
- (2) KDC создает билет (на ключе получателя) и конверт (на ключе отправителя)
- (3) Билет с сеансовым ключом отправитель пересылает получателю
- (!) Возможна атака повтора ответа: можно сохранить сообщение шага 3 и использовать его позже

Протокол Ниидома-Шрёдера (Needham-Schreder)



- (1) Отправитель передает сообщение KDC, в которое включает свой *nonce* R_A, свой опознавательный код и опознавательный код получателя
- (2) KDC передает зашифрованное сообщение отправителю, которое включает *nonce* R_A , опознавательный код получателя, ключ сеанса и зашифрованный билет для получателя.
- (3) Отправитель передает билет получателя по адресу
- (4) Получатель передает свой *nonce* R_B отправителю, зашифрованный ключом сеанса K_{AB}
- (5) Отправитель отвечает на запрос получателя, передавая *nonce* R_B-1, зашифрованное сеансовым ключом K_{AB}

Протокол Отвея-Рисса (Otway-Rees)

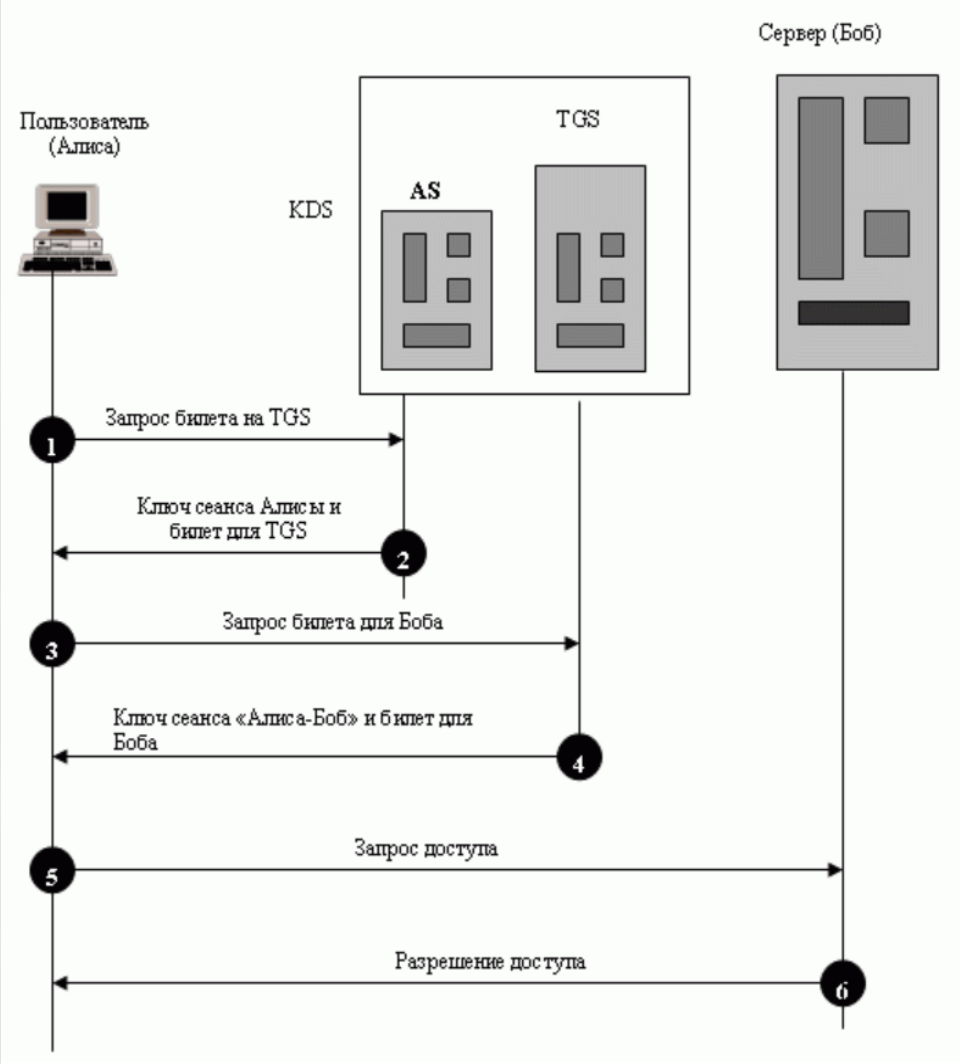


- (1) Отправитель передает сообщение получателю, включающее *nonce R*, идентификаторы отправителя и получателя и билет для KDC - в билет входят nonce R_A , копия общего *nonce R* и идентификаторы отправителя и получателя
- (2) Получатель создает подобный билет, но с собственным *nonce* R_B. Оба билета передают KDC
- (3) KDC создает и передает получателю сообщение, которое содержит общий *nonce R*, билет для отправителя и билет для получателя. Билеты содержат ключ сеанса K_{AB} и соответствующие *nonce* R_A и R_B
- (4) Получатель пересылает билет отправителю
- (5) Отправитель подтверждает получение сообщением зашифрованным на ключе сеанса

Протокол «Цербер» (Kerberos)

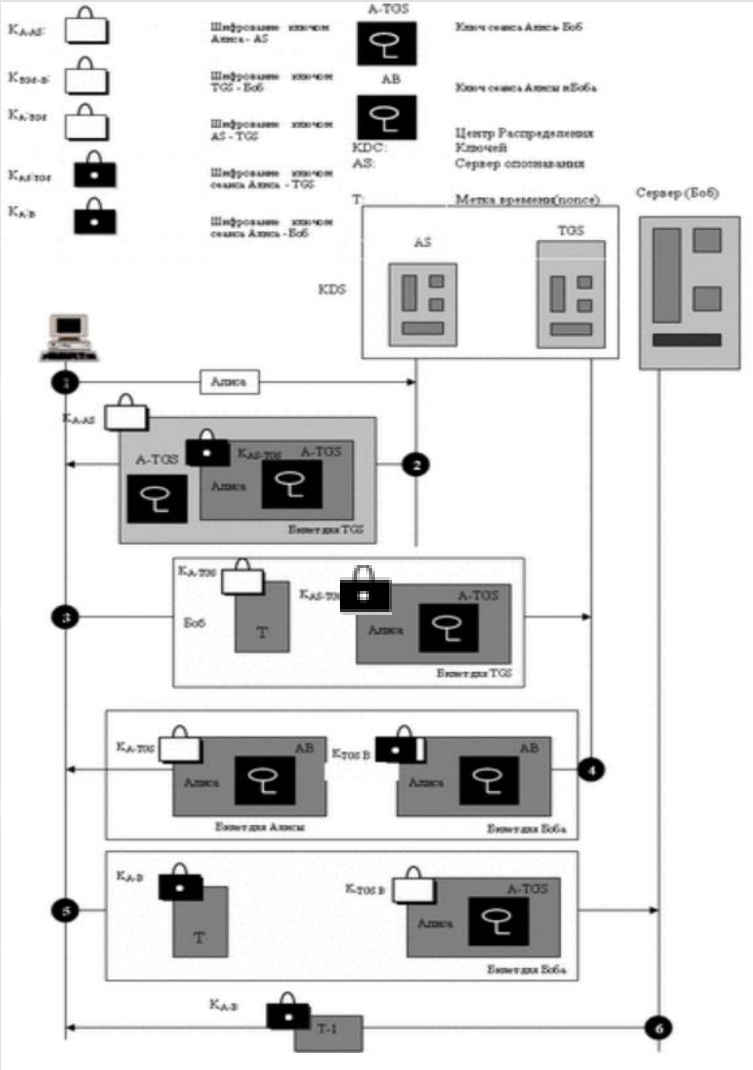
- Протокол проверки подлинности сторон (симметричной аутентификации), обеспечивающий безопасную передачу данных в незащищенных сетях
- Протокол разработан для использования в системах с «клиент-серверной» архитектурой (например, протокол передачи файлов FTP
- Протокол поддерживают, например, FreeBSD, Mac OS X, Red Hat Linux и прочие UNIX-подобные операционные системы

Протокол «Цербер»: взаимодействие серверов



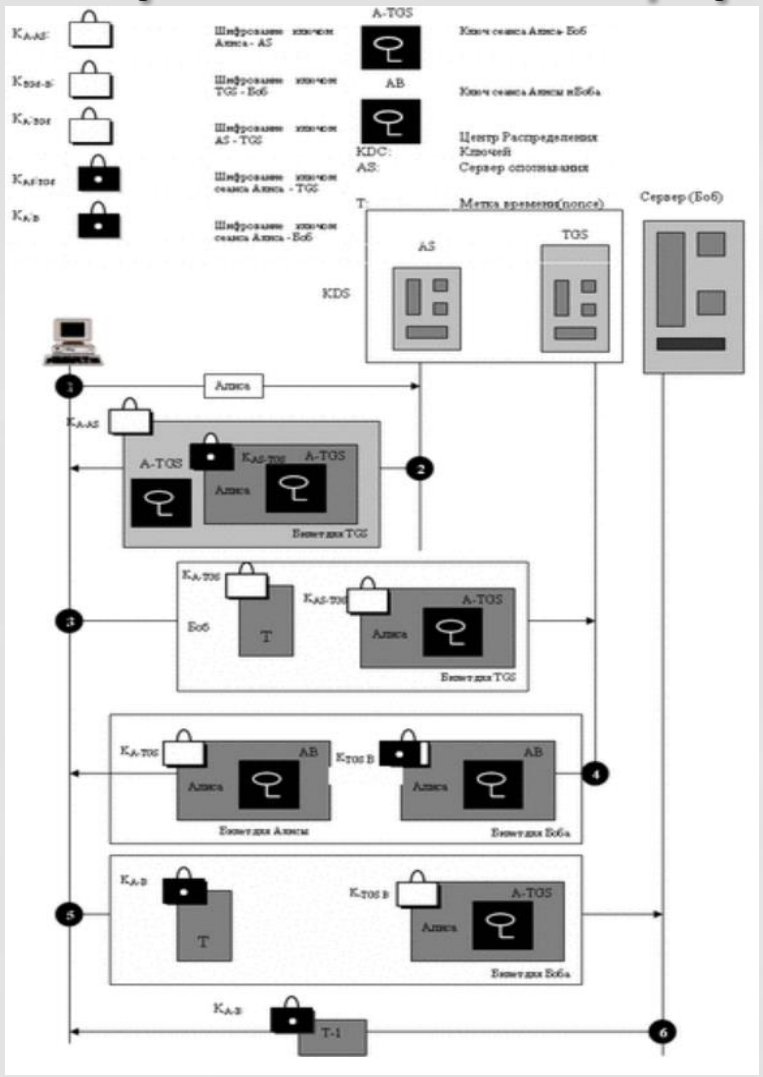
- Опознавательный сервер (Authentication Server). Каждому пользователю, зарегистрированному в AS, предоставляют пользовательский идентификационный код и пароль.. AS верифицирует пользователя, выдает ключ сеанса, который используется между клиентом и TGS, и передает билет для TGS
- Предоставляющий билет сервер (Ticket-Granting Server) вырабатывает билет для сервера услуг и обеспечивает ключ сеанса (K_{AB}) между клиентом и сервером услуг.
- Сервер услуг предоставляет сервисы для клиента.

Протокол «Цербер»: взаимодействие абонентов



- (1) Алиса передает свой запрос AS в открытом тексте, используя свой зарегистрированный код идентификации
- (2) AS передает сообщение, зашифрованное постоянным симметричным ключом клиента K_{A-AS}. Это сообщение содержит два объекта: ключ сеанса, K_{A-TGS}, который используется клиентом, чтобы войти в контакт с TGS, и билет для TGS, который зашифрован TGS-симметричным ключом K_{AS-TSG} . Клиент не знает K_{A-AS}, но когда сообщение прибывает, он сообщает свой пароль, который служит для создания K_{A-AS} и после уничтожается. Процесс использует K_{A-AS} для того, чтобы расшифровывать переданное сообщение с K_{AS-TSG} и билетом для TGS .
- (3) Клиент передает три объекта в TGS: билет, полученный от AS, имя сервера услуг, метку времени, которая зашифрована ключом K_{A-TGS}. Метка времени предотвращает ложный ответ нарушителя.

Протокол «Цербер»: взаимодействие абонентов



- (4) Теперь TGS передает клиенту два билета: каждый содержит ключ сеанса K_{A-B} между клиентом и сервером услуг , Билет для клиента - зашифрованный K_{A-TGS} , билет для сервера - зашифрованный с ключом K_{B-TGS}.
- (5) Клиент передает билет серверу услуг и метку времени, зашифрованной ключом K_{A-B}.
- (6) Сервер услуг подтверждает, что получил эту информацию, прибавляя 1 к метке времени. Сообщение шифруется ключом K_{A-B} и передается клиенту.

Спасибо за внимание !