СЕРВИСЫ БЕЗОПАСНОСТИ. ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИЯ

Пестунова Тамара Михайловна

кандидат технических наук, доцент

ptm@ngs.ru

КРИПТОГРАФИЯ С ОТКРЫТЫМ КЛЮЧОМ

В основе теоретико-сложностный подход.

Гипотеза $P \neq NP$.

Односторонние функции $F: X \rightarrow Y$

а) существует полиномиальный алгоритм вычисления $y=F(x)$;

б) не существует полиномиального алгоритма инвертирования функции F, т.е. решения уравнения $F(x)=y$ относительно x .

Более сильное требование, чем принадлежность к NP-полным проблемам, т.к. требуется отсутствие полиномиального алгоритма почти всюду.

КРИПТОГРАФИЯ С ОТКРЫТЫМ КЛЮЧОМ

Функция с секретом $f_K : X \rightarrow Y$

а) **при любом k существует полиномиальный алгоритм вычисления f_K;**

б) **при неизвестном k не существует полиномиального алгоритма инвертирования данной функции;**

в) **при известном k существует полиномиальный алгоритм ее инвертирования.**

Дифи У., Хеллман М. Защищенность и имитостойкость. Введение в криптографию // ТИИЭР т.67, №3, 1979

RSA

Алгоритм шифрования, в основе сложная задача факторизации больших чисел

1. Абонент выбирает пару простых чисел: p и q

вычисляет и публикует $n=pq$

2.Функция Эйлера: $\varphi(n) = (p-1)(q-1)$

3. Случайное e

4. Вычисляем d: $ed=1 \pmod{\varphi(n)}$

5. Открытый ключ: (n,e)

6. Секретный ключ: $(p, q, \varphi(n), d)$

Шифрование: $s=t^e \pmod n$

Расшифрование: $t=s^d \pmod n$

RSA-пример

Задача. *Зашифровать аббревиатуру RSA при p=17, q=31.*

Решение. 1) Вычисляем модуль $n=p\cdot q=17\cdot 31=527$

2) Функция Эйлера $\varphi(n)=(p-1)(q-1)=480.$

3) Случайное e, т.ч. $(e, \varphi(n))=1$, например $e = 7.$

4) Вычисляем d, т.ч. $e\cdot d=1 (mod\ 480)$, $d=343$, *т.к.* $343\cdot 7=2401=480\cdot 5+1$

5) Переведем слово «*RSA*» *в числовой вид:*

$R \rightarrow 18_{10} = (10010)_2$

$S \rightarrow 19_{10} = (10011)_2$

$A \rightarrow 1_{10} = (00001)_2$

Общая последовательность (с учетом диапазона [0,526]):

$RSA \rightarrow (100101001100001) \rightarrow (100101001),(100001) = (M_1 = 297, M_2 = 33)$

RSA-пример (продолжение)

6) Шифруем последовательно M_1 и M_2

$$C_1 = E_k(M_1) = M_1^e \pmod{527} = 297^7 \pmod{527} = 474$$
$$C_2 = E_k(M_2) = M_2^e \pmod{527} = 33^7 \pmod{527} = 407$$

Получаем шифрограмму: **C= (474,407)**

7) Расшифрование

$$D_k(C_1) = C_1^d \pmod{527} = 474^{343} \pmod{527} = \mathbf{297}$$
$$D_k(C_2) = C_2^d \pmod{527} = 407^{343} \pmod{527} = \mathbf{33}$$

Для упрощения вычислений можно использовать соотношение:
343=256+64+16+4+2+1

Домашнее задание: написать программу на языке, реализующую RSA, проверить
Пример для теста – зашифровать и расшифровать последовательность, включающую свои имя и фамилию (без пробела) + 3 собственных примера.

ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

Число, зависящее от сообщения и от некоторого секретного, известного только подписывающему субъекту ключа.

Легко проверяема, проверку подписи может осуществить каждый без получения доступа к секретному ключу.

При возникновении спорной ситуации (отказ от подписи, подделка подписи), третья сторона должна иметь возможность разрешить спор.

Таким образом, решаются три задачи:

аутентификация источника сообщения,

установление целостности сообщения,

невозможность отказа от подписи конкретного сообщения.

ЦИФРОВАЯ ПОДПИСЬ: ОТЛИЧИЯ ОТ СОБСТВЕННОРУЧНОЙ

СП. Не зависит от подписываемого текст, всегда одинаковая
ЦП. Зависит от текста, почти всегда разная

СП. Неразрывна связана с подписывающим лицом,
ЦП. Определяется секретным ключом, может быть утеряна.

СП. Неотделима от носителя, каждый экземпляр подписывается отдельно.
ЦП. Верна для всех копий.

СП. Не требует доп. механизмов для реализации.
ЦП. Требует доп. механизмов (алгоритмы, вычисления)

СП. Не требует поддерживающей инфраструктуры.
ЦП. Нужна доверенная инфраструктура сертификатов открытых ключей

ЦИФРОВАЯ ПОДПИСЬ: структура и требования

ЭЦП включает два алгоритма:

Алгоритм вычисления подписи и Алгоритм проверки подписи.

Основные требования:

Исключить возможность получения подписи без знания секретного ключа

Гарантировать возможность проверки подписи без знания секретного ключа.

Надежность подписи обеспечивается сложностью трех задач:

- **Подделки подписи (нахождение значения подписи лицам, не являющимся владельцем ЭЦП)**
- **Создания подписанного сообщения (нахождение хотя бы одного сообщения с правильным значением подписи)**
- **Подмены сообщения (подбор двух разных сообщений с одинаковым значением подписи)**

ЦИФРОВАЯ ПОДПИСЬ: ОБЩИЕ СХЕМЫ

1. Схемы на основе симметричных систем шифрования.

2. Схемы на основе специально разработанных алгоритмов вычисления и проверки подписи.

3. Схемы на основе шифрования с открытыми ключами (с восстановлением текста.)

(E,D) - пара преобразований, А - автор, П- получатель, М - сообщение,
S - подпись автора.
E зависит от открытого ключа, D - от секретного.
А: S=D(M) П: E(S)= M.
Требования: M= E (D(M)) для всех M;
невозможно вычислить D(M) без знания секретного ключа.
Возможно: данные подписываются, потом шифруются

ЦИФРОВАЯ ПОДПИСЬ : ПРОТОКОЛ ЭЛЬ-ГАМАЛЯ.

Основан на вычислении логарифма в конечном поле.

p **- простое число,**
Z(p) **- конечное поле,**
w **- примитивный элемент в** ***Z(p).***

Выбрать случайное число $1 \le a \le p-2$
(a - секретный ключ).

Вычислить $b = w^a \bmod p$.
((p, w, b)- открытый ключ).

ЦИФРОВАЯ ПОДПИСЬ : ПРОТОКОЛ ЭЛЬ-ГАМАЛЯ.

Алгоритм подписи

1. Выбрать случайное число $1 \le r \le p-2$;
2. Вычислить $c = w^r \bmod p$;
3. Для $x=M$ вычислить $d = (x - a \cdot c) r^{-1} \bmod (p-1)$;
4. $S=(c, d)$.

Алгоритм проверки. $b^c c^d \equiv w^x \pmod p$.

ЦИФРОВАЯ ПОДПИСЬ : ПРОТОКОЛ ЭЛЬ-ГАМАЛЯ.

Замечания.

1. Число r должно уничтожаться сразу после вычисления подписи. Иначе секретный ключ вычисляется

$$a=(x- r\cdot d)c^{-1} \bmod (p-1)$$

2. Число ***r*** должно быть случайным, не должно повторяться для разных подписей . На шаге 3 реально обычно берется не ***x=M***, а ***x=h(M)*** - свертка, полученная с помощью хэш-функции.
3. На одном секретном ключе можно выработать ЭЦП для многих сообщений

ЦИФРОВАЯ ПОДПИСЬ: АЛГОРИТМЫ, ПОСТРОЕННЫЕ ПО ПРИНЦИПУ ПРОТОКОЛА ЭЛЬ-ГАМАЛЯ

Схема проверки подписи вида

$$\alpha^{A}\beta^{B} \equiv \gamma^{C} \pmod{p},$$

где (A,B,C) – перестановка элементов $(\pm x, \pm d, \pm c)$

заложена во многих стандартных алгоритмах ЭЦП, в том числе в **ГОСТ 34-10-94** и **DSS.**

ОДНОРАЗОВАЯ ЦИФРОВАЯ ПОДПИСЬ
СХЕМА Диффи-Лампорта

Нужно подписать сообщение $M=(m_1 m_2 \ldots m_n)$, *где* m_i **из** $\{0,1\}$

Подписывающий

1) ***выбирает***

$2n$ **случайн. секретных ключей:** $K=[(k_{10},k_{11}), (k_{20},k_{21}),\ldots, (k_{n0},k_{n1})]$

$2n$ **случайных чисел из {0,1}:**

$$S=[(s_{10},s_{11}), (s_{20},s_{21}), \ldots , (s_{n0},s_{n1})]$$

2) ***вычисляет***

$R_{ij}=E_{kij}(s_{ij})$, **где** j **из** $\{0,1\}$, $i=1,2,\ldots,n$

3) ***Публикует наборы***

S **и** $R=[(R_{10},R_{11}), (R_{20},R_{21}), \ldots , (R_{n0},R_{n1})]$

Подпись для M **имеет вид** $(k_{1m1}, k_{2m2}, \ldots , k_{nmn})$

Проверка подписи: $R_{ij}=E_{kij}(s_{ij})$, **где** $j=m_i$, $i=1,2,\ldots,n$

ОДНОРАЗОВАЯ ЦИФРОВАЯ ПОДПИСЬ
СХЕМА Диффи-Лампорта (продолжение)

Недостатки:

1) Слишком большой размер ключа

Можно хранить только секретный ключ k , и на его основе формировать всю последовательность

$$k_{ij}=E_k(i,j)$$

2) После проверки весь секретный ключ или его часть становится известны проверяющему,

поэтому система одноразовая

ИНФРАСТРУКТУРА ОТКРЫТЫХ КЛЮЧЕЙ

ИОК необходима для исключения возможности подделки открытого ключа лицами, которые хотели бы выдать себя в качестве владельца секретного ключа.

ИОК включает в себя сеть центров сертификации открытых ключей.

Цель – обеспечить подтверждение достоверности принадлежности открытого ключа заявленному владельцу.

СЕРТИФИКАТЫ ЭЦП

Сертификат – набор данных, заверенный ЭЦП центра сертификации, включающий открытый ключ и дополнительны атрибуты.

Типовая структура сертификатов ключей определена спецификациями X509, имеющих статус международного добровольного стандарта.

Порядковый номер сертификата.

Идентификационный алгоритм ЭЦП.

Имя центра сертификации (удостоверяющего центра).

Срок действия ЭЦП.

Имя владельца сертификата.

Открытые ключи владельца сертификата.

Идентификационный алгоритм, ассоциированный с открытыми ключами владельца.

ЭЦП центра сертификации.

ТРЕБОВАНИЯ К СЕРТИФИКАТАМ

Удостоверяющий центр – это компонент глобальной службы каталогов, отвечающий за управление криптографическими ключами пользователей

Свойства сертификатов:

Любой пользователь, знающий ОК удостоверяющего центра (УЦ), может узнать ОК других клиентов УЦ и проверить целостность сертификата.

Никто, кроме УЦ, не может модифицировать информацию о пользователе, без нарушения целостности сертификата.

В X509 не описываются конкретные процедуры генерации ключей, но дается ряд общих рекомендаций.

РЕКОМЕДАЦИИ ПО ГЕНЕРИРОВАНИЮ КЛЮЧЕЙ

1) Ключи может генерировать сам пользователь. Тогда Секретный ключ не попадает в руки третьих лиц, но надо решить проблему безопасности связи с УЦ.

2) Ключи может генерировать доверенное лицо, тогда стоит задача безопасной доставки секретного ключа владельцу, а также предоставление достоверных данных в УЦ.

3) Ключи могут генерироваться УЦ, тогда должна быть решена задача безопасной передачи секретного ключа владельцу.

ЮРИДИЧЕСКИЕ АСПЕКТЫ ИСПОЛЬЗОВАНИЯ ЭЦП

Юридические аспекты использования ЭЦП обусловлены необходимостью разрешения споров, связанных с отказом от авторства, подделкой, возмещением убытков в спорных ситуациях.

- **Кто несет ответственность если подписанная сделка не состоялась;**
- **Кто несет ответственность, если система взломана и выявлен факт подделки секретного ключа;**
- **Какова ответственность уполномоченного по сертификатам, если открытый ключ сфальсифицирован;**
- **Какова ответственность владельца в случае утраты ОК;**
- **Кто несет ответственность при повреждении и разглашении секретного ключа;**
- **Каков порядок разрешения споров.**

КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ

Объекты - удаленные абоненты, взаимодействующие по открытой сети, в общем случае не доверяющие друг другу.

Цель - решение некоторой задачи.

Имеется противник, преследующий свои цели (противником может быть не только внешняя сторона, но один или несколько абонентов или сервер).

Протокол - некоторый распределенный алгоритм, определяющий последовательность действий каждой из сторон.

КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ (примеры задач)

Протокол подписания контракта (исключить ситуацию, когда один подписал контракт, а другой нет).

Протокол подбрасывания монеты (не допустить, чтобы подбрасывающий мог изменить результат подбрасывания после получения информации о догадке угадывающего).

Протокол идентификации абонента (Один абонент должен доказать другому, что он именно тот, за кого себя выдает).

Протокол электронной подписи (обеспечить аутентификацию, проверку целостности сообщения, невозможность отказа от факта подписи конкретного сообщения)

Протокол разделения секрета (содержание секрета разделяется таким образом, что каждый из пользователей обладает его частью, но без участия всех участников восстановить его невозможно)

ТЕХНОЛОГИЯ ОТКРЫТОГО РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ

Задача.

Организовать такую процедуру взаимодействия удаленных абонентов A и B, чтобы выполнить следующие условия:

а) вначале у A и B нет общей секретной информации, но в конце процедуры такая общая секретная информация вырабатывается (общий ключ) ;

б) пассивный противник, который перехватывает все передачи информации и знает, что хотят получить A и B, не может восстановить общий ключ.

ОТКРЫТОЕ РАСПРЕДЕЛЕНИЕ КЛЮЧЕЙ. ПРОТОКОЛ ДИФФИ-ХЕЛЛМАНА

Алгоритм Основан на общепризнанной трудной задаче дискретного логарифмирования,
т.е. инвертирования функции $a^X \pmod p$, где p – простое число .

Абоненты **A** и **B** выбирают натуральные числа x_A и x_B соответственно.

Вычисляют $y_A = a^{XA} \pmod p$ и $y_B = a^{XB} \pmod p$.

Обмениваются результатами по сети. При этом p и a общедоступны.

ОТКРЫТОЕ РАСПРЕДЕЛЕНИЕ КЛЮЧЕЙ. ПРОТОКОЛ ДИФФИ-ХЕЛЛМАНА

После обмена вычисляют новые значения

A: $(y_B)^{XA} = (a^{XB})^{XA} \pmod p$,

B: $(y_A)^{XB} = (a^{XA})^{XB} \pmod p$.

У абонентов появился общий элемент $a^{XA\,XB}$.

Противник знает p, a, a^{XA}, a^{XB} и хочет узнать $a^{XA\,XB}$

В настоящее время неизвестно более эффективных действий , чем дискретное логарифмирование - является труднорешаемой задачей.

ИНТЕРАКТИВНОЕ ДОКАЗАТЕЛЬСТВО С НУЛЕВЫМ РАЗГЛАШЕНИЕМ

Д -доказывающий,
П- проверяющий,
У - доказываемое утверждение

Д хочет доказать П, что У истинно. П без помощи Д не может проверить истинность У. Число раундов не ограничено. Требования к протоколу:

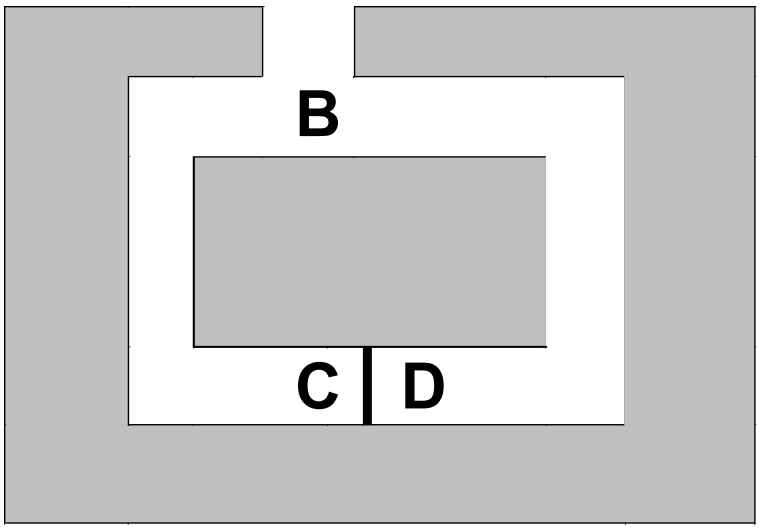
а) полнота (если У истинно, то Д убедит П признать это);

б) корректность (если У ложно, то Д вряд ли убедит П , что У истинно)

в) нулевое разглашение (в результате работы протокола П не увеличит свои знания об У, т.е. Не сможет извлечь никакой информации, почему У истинно).

ДОКАЗАТЕЛЬСТВО С НУЛЕВЫМ РАЗГЛАШЕНИЕМ

ПЕЩЕРА АЛИ-БАБЫ



<u>Цель:</u> Доказывающий (Д) должен убедить проверяющего (П) в наличии у него ключа от двери C-D

1) Начало: Д – в точке B, П – в точке A

2) Д переходит случайным образом в C или D.

3) П просит Д выйти из правого (левого) коридора.

4) Д выполняет просьбу П, при необходимости воспользовавшись имеющимся ключом.

При отсутствии ключа вероятность угадывания ½

5) Если Д правильно выполнил запрос, то итерация считается успешной, если не смог правильно выполнить – то доказательство отвергается.

6) Итерация 1)-5) повторяется *n* , пока не будет достигнута требуемая достоверность 1-(1/2)^*n*. При достижении заданного уровня достоверности доказательство принимается.

Доказательство с нулевым разглашением

Формальное определение в терминах машин Тьюринга (МТ)

Интерактивным доказательством для языка L

называется пара интерактивных МТ (P(x), V(x)),

где P – моделирует доказывающего,

V – моделирует проверяющего,

x – входное слово допустимого МТ языка L,

[P(x), V(x)] – слово на выходе

Доказательство с нулевым разглашением

Полнота $\forall x \; \text{Вер}\{[P(x), V(x)] = 1\} = 1$

Если оба участника следуют протоколу, то доказательство будет принято

Корректность $\forall P^*$ и полинома p
при $x \notin L$ достаточно большой длины

$$\text{Вер}\{[P^*(x), V(x)] = 1\} < 1 / p(|x|)$$

Если противник будет пытаться доказывать ложное утверждение, как угодно отклоняясь от протокола, то вероятность успеха для него пренебрежимо мала

Доказательство с нулевым разглашением

Нулевое разглашение

Для любой полиномиальной вероятностной МТ V^*, существует вероятностная МТ M_V^* работающая в среднем за полиномиальное время, т.ч

$$\forall x \in L \qquad MV^*(x) = [P(x), V^*(x)]$$

Защищает доказывающего от нечестного проверяющего:

Как бы ни отклонялся проверяющий от действий, предписанных протоколом (использует V^* вместо V),

он сможет при этом получить только такую информацию, которую и сам может самостоятельно вычислить в среднем за полиномиальное время

ZK – доказательство «Изоморфизм графов»

Рассмотрим графы

$G_1 = (X,U_1)$ и $G_0 = (X,U_0)$,

т.ч. $G_1 = \phi(G_0)$, где ϕ – изоморфизм

Изоморфизм – перестановка на множестве вершин X_0, такая что ребро (x,y) существует в графе G_0 тогда и только тогда, когда ребро $(\phi(x), \phi(y))$ существует в графе G_1.

(обозначается $G_1 \cong G_0$)

ZK – доказательство «Изоморфизм графов»

1. ***Доказывающий***

выбирает случайную перестановку **π** *на множестве вершин* **X**, *вычисляет* $H = \pi(G_1)$ *и посылает* **H** *проверяющему.*

2 . ***Проверяющий***

выбирает случайный бит **α** *и посылает его доказывающему*.

3. ***Если*** $\alpha = 1$, *то*

доказывающий отсылает проверяющему перестановку **π** ,

иначе **– π ○ ф.**

4. ***Если*** *полученная проверяющим перестановка не является изоморфизмом между* G_α ***и*** **H**, *то доказательство отвергается.*

Иначе *выполняется следующая итерация протокола.*

ZK – доказательство «Изоморфизм графов»

Доказательство принимается,

если все проверки ш. 4 выполнены
достаточное количество раз
(с учетом заданной вероятности достоверности)

и всегда давали положительный результат.

Данный протокол является протоколом с **абсолютно нулевым разглашением** для языка «изоморфизм графов»

ПРОТОКОЛ АУТЕНТИФИКАЦИИ ФИАТА - ШАМИРА

Относится к числу протоколов «нулевого разглашения». Основан на сложности задачи извлечения корня (аналогична факторизации)

1.Предварительный этап.

1.1. Доверенный центр T выбирает два простых числа ***p*** и ***q***, рассылает всем доказывающим число ***n=pq.***

1.2. ***A*** выбирает секрет ***s***, т.ч. ***(s,n)=1, 1≤s≤n-1***.

вычисляет $v = s^2 \pmod n$,

2. Итерация.

2.1. ***A*** выбирает случайное ***z***, т.ч. ***1<z<n-1***

вычисляет $x = z^2 \pmod n$,

отправляет ***x*** проверяющему ***B***

2.2. ***B*** выбирает случайный бит ***c*** и отправляет его ***A***

ПРОТОКОЛ АУТЕНТИФИКАЦИИ ФИАТА - ШАМИРА

2.3. ***A*** вычисляет ***y =z***, если ***c = 0***

y =zs, если ***c = 1***

и отправляет ***y*** проверяющему ***B***

2.2. ***B*** принимает итерацию, если $y^2 = xv^c \pmod n$.

Комментарий.

Полнота – непосредственно следует из вычисления формулы

Корректность. Например, пусть некто пытается выдать себя за А, подбирая значение ***x,*** не зная секрета: $x = z^2 / v$.

Тогда он сможет дать правильный ответ при ***c=1***,

но не сможет ответить правильно при ***c=0***,

(надо вычислить корень из ***V***, что является труднорешаемой задачей).

KERBEROS

Kerberos –это сервер аутентификации, (доверенная третья сторона) .

Функции: владеет секретными ключами обслуживаемых субъектов и помогает им в попарной проверке подлинности.

Предоставляет средства проверки подлинности субъектов,

Условия абонентов: незащищенная сеть,
возможность перехвата,
модификации,
дополнения пересылаемой информации.

KERBEROS

.

Kerberos не полагается

- *на средства аутентификации, операционных систем сетевых компьютеров,*
- *на подлинность сетевых адресов,*
- *на физическую защищенность сетевых компьютеров (кроме тех, на которых работает сервер Kerberos).*

Физическая реализация Kerberos —
один или несколько серверов проверки подлинности, функционирующих на физически защищенных компьютерах. Серверы поддерживают базу данных субъектов и их секретных ключей.

KERBEROS

1. **Клиент C Kerberos: c, s,...**
   {Клиент направляет Kerberos сведения о себе и о запрашиваемом сервере}
2. **Kerberos Клиент C: {d1}Kc, {Tc.s}Ks**
   {Kerberos возвращает билет, зашифрованный ключом сервера, и дополнительную информацию, зашифрованную ключом клиента}
3. **Клиент C Сервер S: d2, {Tc.s}Ks**
   {Клиент направляет на сервер билет и дополнительную информацию}

ОТ ЧЕГО KERBEROS НЕ ЗАЩИЩАЕТ

- **Атаки на доступность.** *отражение таких атак и реакция на "нормальные" отказы возлагается на пользователей и администраторов.*

- **Кража секретных ключей.** *Забота о сохранности своих ключей лежит на субъектах.*

- **Угадывание паролей.**

По слабому паролю можно вычислить секретный ключ и расшифровать полученную от Kerberos информацию.

При использовании "троянца" злоумышленник может узнать пароли многих пользователей.

След-но Kerberos все же предполагает некоторый базовый уровень защищенности обслуживаемых компьютеров.

ОТ ЧЕГО KERBEROS НЕ ЗАЩИЩАЕТ

- **Повторное использование идентификаторов субъектов.** *Т*

Теоретически возможно: новый субъект Kerberos получит тот же идентификатор, что был у ранее выбывшего субъекта.

Возможно, что этот идентификатор остался в списках управления доступом какой-либо системы в сети.

Тогда новый субъект унаследует права доступа выбывшего.

- **Рассогласование часов на компьютерах.**

Допустимая погрешность часов может устанавливаться индивидуально для каждого сервера.

Если синхронизация часов выполняется сетевыми средствами, соответствующий протокол должен сам заботиться о безопасности.

Более подробно о KERBEROS – см., например,
http://www.jetinfo.ru/1996/12-13/1/article1.12-13.1996.html