Криптографические протоколы

Лекция 8

Протоколы распределения ключей (Часть 2)

Деркач Максим Юрьевич

October 11, 2018

Ссылки

1. ISO/IEC 11770-1:2010 – Information technology – Security techniques – Key management – Part 1: Framework
2. ISO/IEC 11770-2:2008 – Information technology – Security techniques – Key management – Part 2: Mechanisms using symmetric techniques
3. ISO/IEC 11770-3:2008 – Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques
4. ISO/IEC 11770-4:2006 – Information technology – Security techniques – Key management – Part 4: Mechanisms based on weak secrets
5. СТБ 34.101.45-2013 "Информационные технологии и безопасность. Алгоритмы электронной цифровой подписи и транспорта ключа на основе эллиптических кривых". http://apmi.bsu.by/assets/files/std/bign-spec19.pdf
6. СТБ 34.101.60-2014 "Информационные технологии и безопасность. Алгоритмы разделения секрета". http://apmi.bsu.by/assets/files/std/bels-spec12.pdf

Протоколы распределения ключей

Needham-Schroeder (NSSK)

1. $A -> S : ID_A || ID_B || R_A$
2. $S -> A : E_{K_{AS}}(R_A || ID_B || KS || E_{K_{BS}}(KS || ID_A))$
3. $A -> B : E_{K_{BS}}(KS || ID_A)$
4. $B -> A : E_{KS}(R_B)$
5. $A -> B : E_{KS}(R_B - 1)$

Протоколы распределения ключей

Needham-Schroeder (NSSK)

Атака

Если ключ KS скомпрометирован, возможна атака на протокол методом повтора сеанса: берутся сообщения из прошлого сеанса с ключом KS^*:

1. $A -> S : ID_A||ID_B||R_A$
2. $S -> A : E_{K_{AS}}(R_A||ID_B||KS||E_{K_{BS}}(KS||ID_A))$
3. $I(A) -> B : E_{K_{BS}}(KS^*||ID_A)$
4. $B -> I(A) : E_{KS^*}(R_B)$
5. $I(A) -> B : E_{KS^*}(R_B - 1)$

Протоколы распределения ключей

KERBEROS

Несколько видоизмененный протокол Needham – Schroeder был положен в основу программного средства аутентификации пользователей распределенных вычислительных систем Kerberos.

В целях исключения возможности осуществления атаки, описанной выше, клиент, пройдя аутентификацию на сервере аутентификации, должен предварительно, до того, как ему будет предоставлен доступ к серверам приложений, получить у специального сервера выдачи билетов так называемые билеты – структуры данных, в которых указывается срок полномочий клиента для доступа к серверам приложений. По истечении этого срока клиент должен получать новый билет. Эта мера ограничивает срок, в течение которого возможно осуществить атаку на протокол.

Протоколы распределения ключей
KERBEROS

AS - сервер аутентификации
TGS - сервер выдачи билетов
$ticket_1 = ID_{TGS}||E_{K_{AS,TGS}}(ID_A||ID_{TGS}||T_{AS}||L||K_{A,TGS})$ -
$auth_1 = E_{K_{A,TGS}}(ID_A||T_A||...)$
$ticket_2 = ID_B||E_{K_B}(ID_A||ID_B||ID_{TGS}||L^{'}||K)$
$auth_2 = E_K(ID_A||T_A||K_A)$
$auth_3 = E_K(ID_A||T_A + 1||K_B)$

Протоколы распределения ключей
KERBEROS

1. $A -> AS: \quad ID_A||ID_{TGS}||R_A$
2. $AS -> A: \quad E_{K_{A,AS}}(E_{K_{A,TGS}}||R_A||ticket_1)$
3. $A -> TGS: ID_B||ticket_1||auth_1$
4. $TGT -> A: E_{K_{A,TGS}}(K||ticket_2)$
5. $A -> B: \quad ticket_2||auth_2$
6. $B -> A: \quad auth_3$

Шаги (1) – (2) выполняются только во время первого входа клиента в систему.
Шаги (3) – (4) выполняются всякий раз , когда клиент A хочет обратиться к новому серверу B.
Шаг (5) выполняется всякий раз, когда A проходит аутентификацию для B.
Шаг (6) является необязательным и выполняется, когда A требует от B взаимную аутентификацию.

Протоколы распределения ключей

Протоколы основанные на ассиметричных криптосистемах

Needham-Schroeder Public Key (NSPK)

1. $A -> S : ID_A || ID_B$
2. $S -> A : E_{K_S^{sec}}(K_B^{pub} || ID_B)$
3. $A -> B : E_{K_B^{pub}}(K_A || ID_A)$
4. $B -> S : ID_B || ID_A$
5. $S -> B : E_{K_S^{sec}}(K_A^{pub} || ID_A)$
6. $B -> A : E_{K_A^{pub}}(K_B || K_A)$
7. $A -> B : E_{K_B^{pub}}(K_B)$
8. $A, B : \quad KS = f(K_A, K_B)$

$E_{K_S^{sec}}()$ - подпись на секретном ключе.
$E_{K_B^{pub}}()$ - шифрование на открытом ключе.
$f()$ - общеизвестная однонаправленная функция.

Протоколы распределения ключей

Протоколы основанные на ассиметричных криптосистемах

NSPK без 3-ей стороны

1. $A -> B : E_{K_B^{pub}}(K_A || ID_A)$
2. $B -> A : E_{K_A^{pub}}(K_A || K_B)$
3. $A -> B : E_{K_B^{pub}}(K_B)$
4. $A, B : \quad KS = f(K_A, K_B)$

Протоколы распределения ключей

Смешанные протоколы

EKE(Encrypted Key Exchange)

$K_{AB} = P$ - пароль

1. $A -> B : ID_A || E_P(K_A^{pub})$
2. $B -> A : E_P(E_{K_A^{pub}}(KS))$
3. $A -> B : E_{KS}(R_A)$
4. $B -> A : E_{KS}(R_A || R_B)$
5. $A -> B : E_{KS}(R_B)$

Bilateral Key Exchange with Public Key

1. $B -> A : ID_B || E_{K_A^{pub}}(R_B || ID_B)$
2. $A -> B : E_{K_B^{pub}}(h(R_B) || R_A || ID_A || KS)$
3. $B -> A : E_{KS}(h(R_A))$

Протоколы распределения ключей

Смешанные протоколы

SPX

a_A, a_B - сетевые адресса.
L, L_A, L_B - время жизни ключей (приватных и публичных) сеанса, пользователя A и пользователя B соответственно.
$m_A = (ID_A||ID_B||L_B||K_B^{pub})$
$m_B = (ID_B||ID_A||L_A||K_A^{pub})$

1. $A->T : ID_B$
2. $T->A : m_A||E_{K_{AT}}(h(m_a)) <-cert_{AB}$
3. $A->B : ID_A||E_{K_A^{sec}}(ID_A||K_A^{pub}||L)||$
   $E_{K_B^{pub}}(KS)||E_{K_A^{sec}}(E_{K_B^{pub}}(KS))||t||E_{KS}(t)||a_A$
4. $B->T : ID_A$
5. $T->B : m_B||E_{K_{BT}}(h(m_B)) <-cert_{BA}$
6. $B->A : E_{KS}(t||a_B)$

Протоколы распределения ключей

1. $A -> B : E_{K_B^{pub}}(KS||t)||sign_A(ID_B||KS||t)$
2. $A -> B : E_{K_B^{pub}}(KS||t||sign_A(ID_B||KS||t))$
3. $A -> B : t||E_{K_B^{pub}}(ID_A||KS)||sign_A(ID_B||t||E_{K_B^{pub}}(ID_A||KS))$

Сертификаты открытых ключей

$cert_A = (ID_A||K_A^{pub}||t||sign_T(ID_A||K_A^{pub}||t))$

Протоколы распределения ключей

X.509

$d_A = (T_A||R_A||ID_B||text_1||E_{K_B^{pub}}(K_A))$

$d_B = (T_B||R_B||ID_A||text_2||E_{K_A^{pub}}(K_B))$

1. $A->B : cert_A||d_A||sign_A(d_A)$
2. $B->A : cert_B||d_B||sign_B(d_B)$
3. $A->B : R_B||ID_B||sign_A(R_B||ID_B)$
4. $A, B : \quad KS = f(K_A, K_B)$

Шаг (3) необязателен, выполняется только если нужно подтверждение.

Протоколы распределения ключей

Протоколы с использованием ЭЦП

Деннига -Сакко

1. $A -> S : ID_A || ID_B$
2. $S -> A : cert_A || cert_B$
3. $A -> B : cert_A || cert_B || E_{K_B^{pub}}(KS || t_A || sign_A(kS || T_A))$
4. $A, B : \quad KS = f(K_A, K_B)$

Шаг (3) необязателен, выполняется только если нужно подтверждение.

Протоколы распределения ключей

Протокол MTI

1. Предварительный этап:
   Выбираются следующие параметры: p, α, где p - простое число, $a \in Z_p^*$
   A выбирает a, $1 \leq a \leq p-2$, $z_A = \alpha^a \ (mod\ p)$.
   B выбирает b, $1 \leq b \leq p-2$, $z_B = \alpha^b \ (mod\ p)$.
2. $A -> B : m_{AB} = \alpha^x \ (mod\ p)$, $1 \leq x \leq p-2$, x - случайное
3. $B -> A : m_{BA} = \alpha^y \ (mod\ p)$, $1 \leq y \leq p-2$, y - случайное

Варианты построения ключа:

№	$m_A B$	$m_B A$	K_A	K_B	K
1	α^x	α^y	$m_{BA}^a z_B^x$	$m_{AB}^b z_A^y$	α^{bx+ay}
2	z_B^x	z_A^y	$m_{BA}^{a^{-1}} \alpha^x$	$m_{AB}^{b^{-1}} \alpha^y$	α^{x+y}
3	z_B^x	z_A^y	$m_{BA}^{a^{-1}x}$	$m_{AB}^{b^{-1}y}$	α^{xy}
4	z_B^x	z_A^Y	m_{BA}^x	m_{AB}^y	α^{bxay}

Протоколы распределения ключей

Предварительное распределение ключей нужно для уменьшения объёма распределяемой и хранимой информации.

$A_1, ..., A_n$ - абоненты.
K - множество ключей.
P - множество исходных ключевых параметров (p_i - пароль каждого абонента).
Q - множество значений ключевых материалов абонентов (q_i - секрет каждого абонента).
R - множество значений открытой информации ($r_1, ..., r_n$ - в открытом доступе).

Протоколы распределения ключей

Схема предварительного распределения ключей:
$S(n) = (K, P, Q, R, A_0, A_1)$

1. $A_0 : P \times R -> Q$ - алгоритм формирования секретных ключевых материалов.
   $A_0(p_i, r_i) = q_i,\ 1 \leq i \leq n$
2. $A_1 : Q \times R -> K$ - алгоритм вычисления ключа парной связи.
   $A_1(q_i, r_j) = A_1(q_j, r_i)$,
   $K_{ij} = A_1(q_i, r_j), i = j$: либо не рассматривается либо некий личный секретный ключ.

$A_0(p, r_i) = Q_i \subseteq K^{t_i} \subseteq Q,\ 1 \leq i \leq n$

Протоколы распределения ключей

Предложение 1

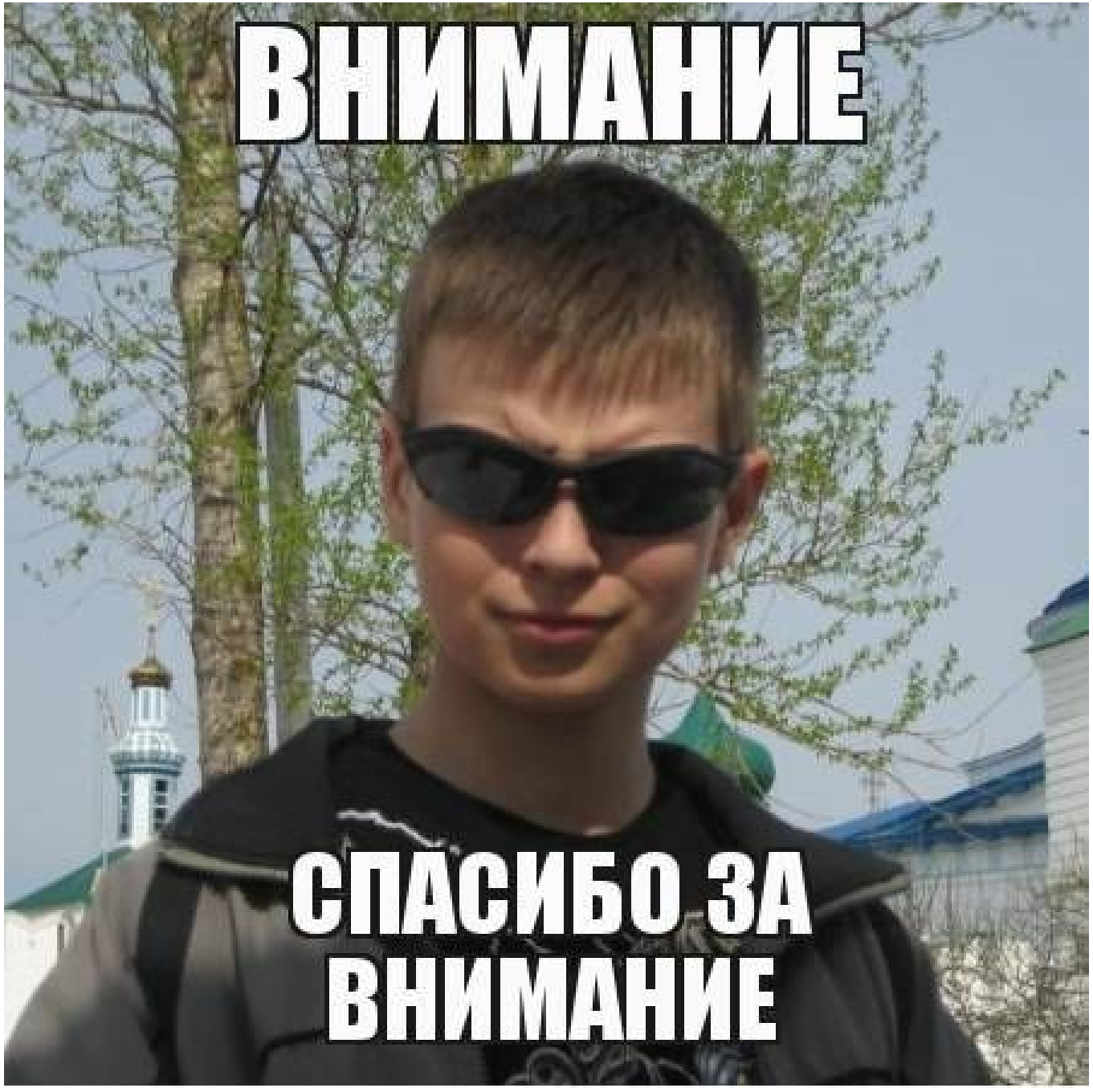
$\forall r_i \in R,\ q_i \in Q,\ 1 \leq i \leq n$

$A_0(p, r_i) = q_i$ - имеет одинаковое число решений относительно $p \in P$.

Предложение 2

$\forall r_i \in R,\ k \in K,\ 1 \leq i \leq n$

$A_1(q_i, r_i) = k$ - имеет одинаковое число решений относительно $q_i \in Q$.


ВНИМАНИЕ
СПАСИБО ЗА ВНИМАНИЕ