Федеральное государственное автономное образовательное учреждение высшего образования
Национальный исследовательский ядерный университет «МИФИ»

Кафедра «Криптология и кибербезопасность»

Криптографические протоколы

курс лекций

Запечников Сергей Владимирович,
профессор кафедры «Криптология и кибербезопасность» НИЯУ МИФИ

Москва – 2018

Криптографические протоколы

курс лекций

Лекция 2.
Протоколы аутентификации, основанные на доказательствах с нулевым разглашением

12 февраля 2018 г.

Цель и требования к протоколам аутентификации

Участники протокола: *претендент P* и *проверяющий V*.

***Цель* проверяющего *V* в протоколе аутентификации заключается в том, чтобы подтвердить идентичность претендента, т.е. что он в самом деле является *P*, а не кем-то иным. Выбор происходит из конечного множества лиц – зарегистрированных участников системы.**

***Требования* к протоколу аутентификации:**

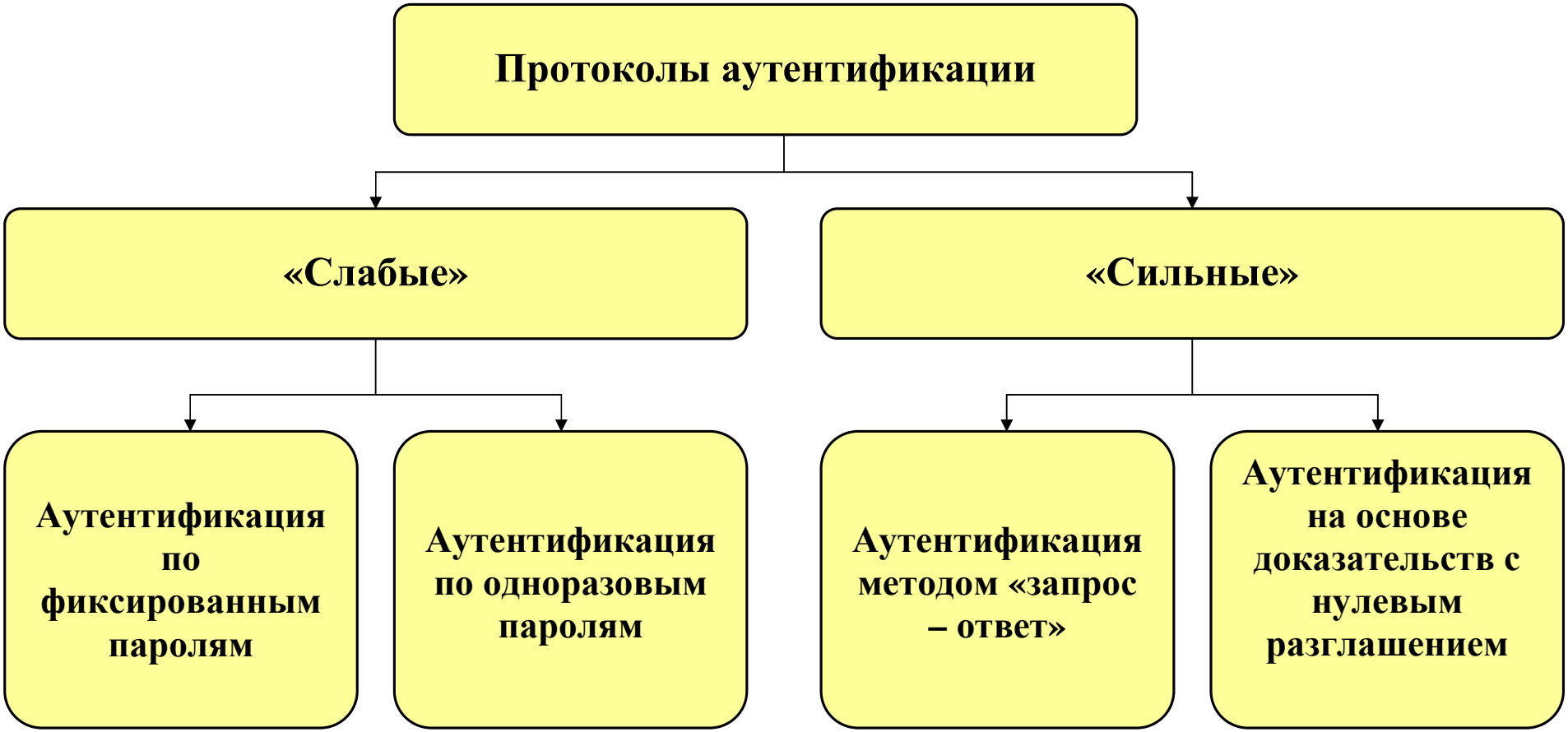
1) если *P* и *V* являются честными, *V* завершит протокол, приняв идентичность *P*;

2) *V* не может повторно использовать протокол, совершенный с *P*, для того, чтобы успешно деперсонифицировать *P* в протоколе с третьей стороной *M*;

3) вероятность того, что любая сторона *M*, отличная от *P*, проведя протокол и играя роль *P*, может заставить *V* завершить протокол с принятием идентичности *P*, пренебрежимо мала;

4) предыдущие свойства остаются справедливыми, даже если между *P* и *V* совершено большое, но полиномиально ограниченное число сеансов протокола аутентификации, противник *M* участвовал в предыдущих сеансах выполнения протокола, и несколько сеансов могли выполняться одновременно.

Классификация протоколов аутентификации



Общие положения

Общая идея протоколов аутентификации, основанных на доказательствах с нулевым разглашением, состоит в том, что законный пользователь *P*, имеющий открытый и секретный ключи, и проверяющий *V* выполняют совместный криптографический протокол интерактивного доказательства, в процессе которого *P*, выступающий в роли претендента, должен доказать свою подлинность. Для этого он должен продемонстрировать знание секретного ключа, но не разгласить его для проверяющего *V*, т.е. из информации, полученной *V*, ему вычислительно невозможно получить секретный ключ *P*.

Все протоколы имеют два этапа: *предварительный* и *рабочий*. На предварительном, который выполняется однократно, специфицируются некоторые параметры и вырабатываются величины, участвующие в рабочем этапе протокола, в частности, открытые и секретные ключи *P*. На рабочем этапе собственно выполняется доказательство аутентичности *P*.

Протокол аутентификации Фиата – Шамира

<table>
<tr><th colspan="4">Предварительный этап</th></tr>
<tr><th colspan="2">P</th><th>Центр доверия</th><th>V</th></tr>
<tr><td colspan="2">s: $(s,n)=1,\ 1\le s\le n-1$
$v=s^2 \pmod n$</td><td>p, q – большие простые числа, $n=pq$</td><td></td></tr>
<tr><td colspan="4">n, v</td></tr>
<tr><th colspan="4">Рабочий этап</th></tr>
<tr><th></th><th>P</th><th></th><th>V</th></tr>
<tr><td>1</td><td>$r\in_R \{1,2,\ldots,n-1\}$
$x=r^2 \pmod n$</td><td>$\rightarrow$</td><td></td></tr>
<tr><td>2</td><td></td><td>$\leftarrow$</td><td>$e\in_R \{0,1\}$</td></tr>
<tr><td>3</td><td>$y=rs^e \pmod n$</td><td>$\rightarrow$</td><td></td></tr>
<tr><td>4</td><td></td><td></td><td>Если (y=0), отклоняет доказательство, так как r=0. В противном случае:
$y^2 \stackrel{?}{\equiv} xv^e \pmod n$</td></tr>
</table>

Стойкость протокола

Рассмотрим подробнее структуру этого протокола. Запрос *e* на шаге (2) требует, чтобы *P* был способен ответить на два вопроса: один из них нужен для того, чтобы продемонстрировать знание *s* и, тем самым, предотвратить обман честного проверяющего нечестным претендентом, другой – чтобы предотвратить обман честного претендента нечестным проверяющим. Соответственно запросу претендент отвечает на шаге (3) либо *y=r*, либо $x = rs \pmod n$. Ни тот, ни другой ответ не несет никакой информации об *s*: в первом случае он от *s* вообще не зависит, во втором – замаскирован случайной величиной *r*, которая известна только *P*, так как на шаге (1) тоже была замаскирована при помощи ОНФ.

Противник, пытающийся деперсонифицировать *P*, может стремиться обмануть проверяющего, выбрав произвольное *r*, вычислив $x = \frac{r^2}{v} \pmod n$ и ответив *y=r* при *e=1*, но не сможет ответить при *e=0*, так как это требует знания $\sqrt{x} \pmod n$.

Противник, выступающий в роли проверяющего, может смоделировать пары сообщений *(x,y)* самостоятельно. Действительно, можно выбирать случайные *y*, задаваться случайными битами *e={0/1}* и вычислять в зависимости от этого $x = y^2 \pmod n$ либо $x = \frac{y^2}{v} \pmod n$.

Распределение вероятностей пар *(x,y)* не будет отличаться от распределения вероятностей тех величин, что сгенерировал бы *P* в реальном протоколе. Таким образом, протокол действительно обладает свойством нулевого разглашения.

Протокол аутентификации Файге – Фиата – Шамира

Предварительный этап		
P	***Центр доверия***	***V***
s_i: $(s_i, n) = 1$, $1 \le s \le n-1$ $v_i = s_i^2 \pmod n$ $v = (v_1, v_2, ..., v_k)$ $s = (s_1, s_2, ..., s_k)$	p, q – большие простые числа, $n = pq$	
n, v_1, v_2, ... , v_k		

Рабочий этап			
	P		***V***
	for (i=1,2, ... ,t)		
1	$r_i \in_R \{1,2,...,n-1\}$, $x_i = r_i^2 \pmod n$	$\rightarrow$	
2		$\leftarrow$	$(e_{i1}, e_{i2}, ..., e_{ik}) \in_R \{0,1\}^k$
3	$y_i = r_i (s_1^{e_{i1}} s_2^{e_{i2}} ... s_k^{e_{ik}}) \pmod n$	$\rightarrow$	
4			$x_i \overset{?}{=} y_i^2 (v_1^{e_{i1}} v_2^{e_{i2}} ... v_k^{e_{ik}}) \pmod n$

Протокол аутентификации Гиллу - Кискатра (Guillou – Quisquater)

<table>
<tr><th colspan="4">Предварительный этап</th></tr>
<tr><th>P</th><th colspan="2">Центр доверия</th><th>V</th></tr>
<tr><td>I_P, B, J</td><td colspan="2">p, q — простые числа, $n=pq$,
I_P, $J = H(I_P)$,
$JB^v \equiv 1 \pmod n$</td><td></td></tr>
<tr><td colspan="4">n, J</td></tr>
<tr><th colspan="4">Рабочий этап</th></tr>
<tr><td></td><th>P</th><td></td><th>V</th></tr>
<tr><td>1</td><td>$r \in_R (1; n-1)$
$T = r^v \pmod n$</td><td>$\rightarrow$</td><td></td></tr>
<tr><td>2</td><td></td><td>$\leftarrow$</td><td>$d \in_R (0; v-1)$</td></tr>
<tr><td>3</td><td>$D = rB^d \pmod n$</td><td>$\rightarrow$</td><td></td></tr>
<tr><td>4</td><td></td><td></td><td>$T' = D^v J^d \bmod n$
$T \overset{?}{\equiv} T' \pmod n$</td></tr>
</table>

Протокол аутентификации Шнорра

<table>
<tr><th colspan="4">Предварительный этап</th></tr>
<tr><th colspan="2">P</th><th>Центр доверия</th><th>V</th></tr>
<tr><td colspan="2">$s \in_R \{1,\ldots,q-1\}$
$v = a^s \bmod p$</td><td>p, q — простые числа, $q \mid p-1$,
$a \in \mathrm{Z}_p : a^q \equiv 1 (\bmod p)$</td><td></td></tr>
<tr><td colspan="4">p, q, a, v</td></tr>
<tr><th colspan="4">Рабочий этап</th></tr>
<tr><th></th><th>P</th><th></th><th>V</th></tr>
<tr><td>1</td><td>$r \in_R \{1,\ldots,q-1\}$
$x = a^r \bmod p$</td><td>$\rightarrow$</td><td></td></tr>
<tr><td>2</td><td></td><td>$\leftarrow$</td><td>$e \in_R \{0,\ldots,2^{t-1}\}$</td></tr>
<tr><td>3</td><td>$y = r + se (\bmod q)$</td><td>$\rightarrow$</td><td></td></tr>
<tr><td>4</td><td></td><td></td><td>$x \overset{?}{=} a^y v^e \bmod p$</td></tr>
</table>

Протокол аутентификации Брикелла – МакКарли

Предварительный этап		
P	***Центр доверия***	***V***
$s < p$ – секр. ключ v: $v = a^{-s} \pmod p$	p, q, w – простые числа, $qw \mid p-1; q^2 \nmid p-1; q, w \geq 2^k$ $a: a^q \equiv 1 \pmod p$	
p, a, v		

Рабочий этап			
	P		*V*
1	$r \in_R \{1,\ldots, p-1\}$ $x = a^r \pmod p$	$\rightarrow$	
2		$\leftarrow$	$e \in_R \{0,\ldots,2^t\}$
3	$y = r + se \pmod{p-1}$	$\rightarrow$	
4			$x \overset{?}{=} a^y v^e \bmod p$

Асимметричные криптосхемы на основе математического аппарата спариваний (pairings)

Определение и свойства спариваний (1)

Пусть заданы: аддитивная группа *G*, мультипликативная группа G'.

Спариванием **(pairing) называется эффективно вычислимое невырожденное билинейное отображение $\hat{e}: G \times G \to G'$.**

Свойства:

1) **требование эффективной вычислимости означает, что для $\forall P, Q$ преобразование $\hat{e}(P, Q)$ вычислимо за полиномиальное время;**
2) **требование невырожденности означает, что если *P* – образующий элемент *G*, то $\hat{e}(P, P)$ – образующий элемент G'. Иными словами, $\hat{e}(P, P) \neq 1$;**
3) **свойство билинейности означает, что для $\forall R, S \in G$ $\hat{e}(Q, R+S) = \hat{e}(Q, R) \cdot \hat{e}(Q, S)$ и $\hat{e}(Q+R, S) = \hat{e}(Q, S) \cdot \hat{e}(R, S)$ (в левых частях равенств операции выполняются в группе *G*, в правых частях – в группе G').**

Определение и свойства спариваний (2)

Из билинейности сразу можно вывести следующее свойство парных отображений:

$$\hat{e}(2P,P)=\hat{e}(P+P,P)=\hat{e}(P,P)\cdot\hat{e}(P,P)=\hat{e}(P,P)^2=$$

$$=\hat{e}(P,P+P)=\hat{e}(P,2P) \quad .$$

Аналогично можно показать, что:

$$\hat{e}(3P,P)=\ldots=\hat{e}(P,3P)$$

$$\hat{e}(aP,bP)=\hat{e}(P,P)^{ab}=\hat{e}(abP,P)=\hat{e}(P,abP).$$

Отсюда следует *теорема*: задача распознавания Диффи–Хеллмана (DDHP) решаема в *G* за полиномиальное время тогда и только тогда, когда $\hat{e}(aP,bP)=\hat{e}(P,cP)$.

Это наблюдение приводит к постановке *билинейной проблемы Диффи – Хеллмана*.

Вычислительно сложные задачи, производные от задачи дискретного логарифмирования

Наименование задачи	**Условия задачи**	
	для аддитивных групп G	**для мультипликативных групп G'**
Дискретное логарифмирование DLOG	Дано: $P, aP \in G$. Найти: a.	Дано: $P, P^a \in G'$. Найти: a.
Вычислительная задача Диффи–Хеллмана CDHP – Computational Diffie – Hellman problem	Дано: $P, aP, bP \in G$. Найти: abP.	Дано: $P, P^a, P^b \in G'$. Найти: P^{ab}.
Задача распознавания Диффи–Хеллмана DDHP – Decisional Diffie – Hellman problem	Дано: $P, aP, bP, cP \in G$. Распознать: $c \stackrel{?}{=} ab \pmod{\lvert P \rvert}$.	Дано: $P, P^a, P^b, P^c \in G'$. Распознать: $c \stackrel{?}{=} ab \pmod{\lvert P \rvert}$.

Вычислительно сложные задачи, порождаемые спариваниями

Наименование задачи	Условия задачи	Примеры криптосистем
Билинейная задача Диффи–Хеллмана BDHP – Bilinear Diffie – Hellman problem	Дано: $P, aP, bP, cP \in G$. Найти: $\hat{e}(P,P)^{abc}$.	Boneh–Franklin (открытое шифрование), Joux (протокол открытого распределения ключей)

Таким образом, спаривание – это функция, которая отображает пары элементов $Q, R \in G$ в элементы $\hat{e}(Q,R) \in G'$. Спаривания удобны тем, что все аналитические выражения, в которых они используются, достаточно просты. Вместе с тем они носят формальный характер, поэтому нужно ещё выбрать соответствующий математический аппарат и найти такие отображения, которые позволили бы практически реализовать криптосистемы. Самыми известными и употребительными являются преобразования, известные как спаривания Вейля (Weil) и Тейта (Tate) над группами точек эллиптических кривых. Оба вычислимы с помощью алгоритма Миллера, но отображение Тейта обычно реализуется более эффективно.

Реализация спариваний

Let E be an elliptic curve containing n points over a field $\mathbb{F}_q$. Let G be a cyclic subgroup of $E(\mathbb{F}_q)$ of order r with r, q coprime. Let k be the smallest positive integer such that $r \mid q^k - 1$. For brevity write $K = \mathbb{F}_{q^k}$. An equivalent characterization is that $K = \mathbb{F}_{q^k}$ is the smallest extension of $\mathbb{F}_q$ containing the rth roots of unity.

The Tate (or Tate-Lichtenbaum) pairing

$$e : E[r] \cap E(K) \times E(K)/rE(K) \to K^*/K^{*r}$$

is defined as follows.

Let f_P be a rational function with divisor $(f_P) = (P)^r$. Choose an $R \in E(K)$ such that $R \neq P, P - Q, O, -Q$. The define

$$f(P, Q) = f_P(Q + R)/f_P(R)$$

Одна из самых известных библиотек – PBC (Pairing-based cryptography) Library (для разработчиков на C): https://crypto.stanford.edu/pbc/

Библиотека для разработки прототипов криптографических конструкций на Python: http://charm-crypto.com/

Протокол трёхстороннего обмена ключами Жу (Joux) (1)

Протокол позволяет трем участникам выработать общий секретный ключ за один раунд, заключающийся в одновременном обмене сообщениями всеми тремя участниками. Все известные до 2001 г. протоколы, начиная с протокола Диффи – Хеллмана, требовали двух или более раундов.

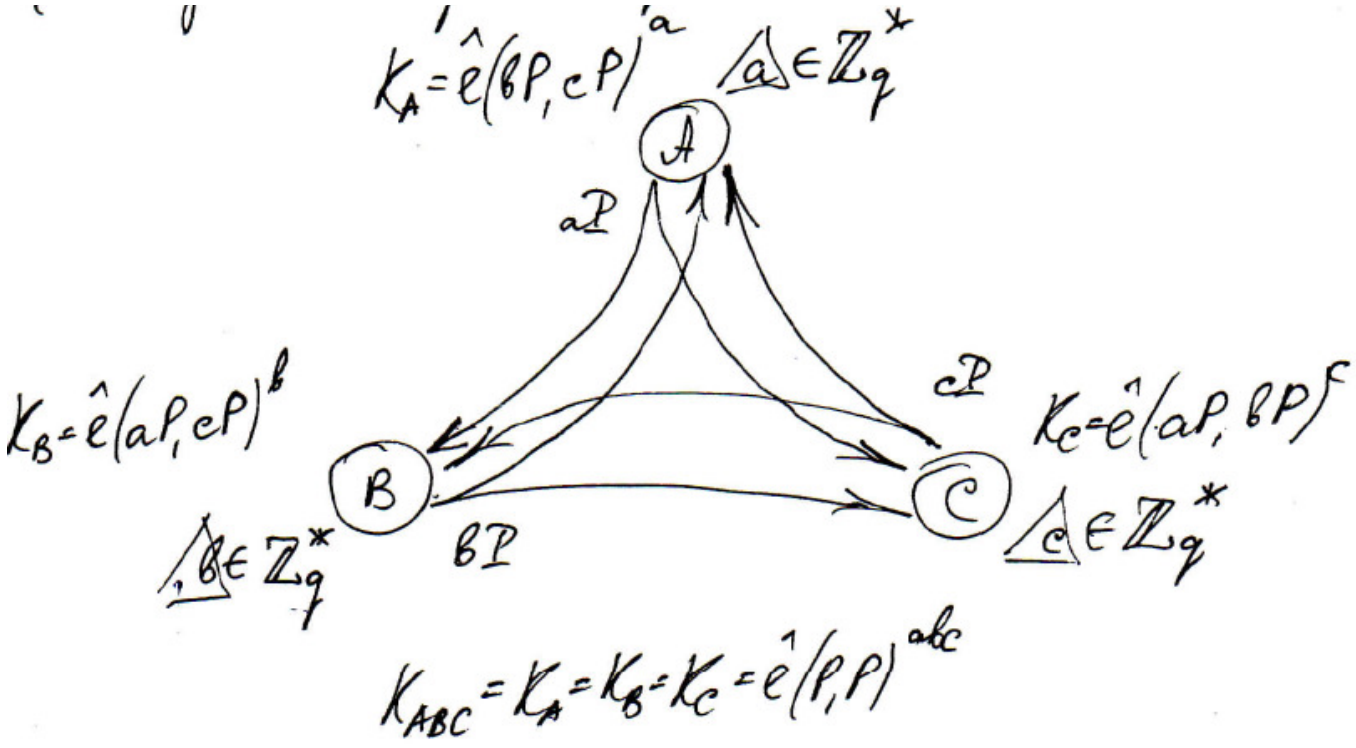
Обозначим участников протокола через *A, B, C*. Их функции в протоколе совершенно симметричны. Порядок выполнения протокола – следующий.

1. Предварительно каждый из участников *A, B, C* выбирает секретный ключ $a \in Z_q^*$, $b \in Z_q^*$, $c \in Z_q^*$ **соответственно.**

2. Каждый из участников *A, B, C* рассылает двум другим участникам свой открытый ключ: aP, bP, cP **соответственно.**

3. Каждый из участников *A, B, C* вычисляет общий для них секретный ключ по формулам $K_A = \hat{e}(bP, cP)^a$, $K_B = \hat{e}(aP, cP)^b$, $K_C = \hat{e}(aP, bP)^c$ **соответственно. При этом** $K_{ABC} = K_A = K_B = K_C = \hat{e}(P, P)^{abc}$.

Протокол трёхстороннего обмена ключами Жу (Joux) (2)



Протокол Жу обеспечивает стойкость только к пассивному противнику в предположении о вычислительной сложности задачи BDHP. Ни аутентификация участников протокола, ни аутентичное распределение ключей здесь не обеспечиваются.

Протокол трёхстороннего обмена ключами Жу (Joux) (3)

Вычислительная и коммуникационная сложность протокола Жу: при обмене сообщениями одновременно передаются 3 элемента группы G_1, следовательно, объем передаваемых в протоколе данных равен $|G_1|$, кроме того, каждый участник выполняет 1 скалярное умножение в группе G_1, 1 вычисление парного отображения и 1 экспоненцирование в группе G_2.

Схема цифровой подписи Boneh – Lynn – Shacham (1)

Особенности: **самая короткая цифровая подпись (160 битов) при стойкости, сравнимой со стандартной подписью DSA (320 битов) и RSA (1024 бита).**

Пусть $H:\{0,1\}^* \to G$ **– функция хэширования,** ***G*** **– аддитивная группа,** $|G|$ **– 160 битов.**

Алгоритм генерации ключей Gen**:**

$x \xleftarrow{R} \mathbf{Z}_q^*$ **– секретный ключ подписывающего;**

$P_{pub} = xP$ **– открытый ключ подписывающего;**

Return $sk = x,\ pk = P_{pub}$ **.**

Алгоритм генерации подписи $Sign_{sk}(M)$**:**

$\sigma = x \cdot H(M)$**;**

Return σ**.**

Алгоритм проверки подписи $Ver_{pk}(M,\sigma)$**:**

If $\hat{e}(P,\sigma) = \hat{e}\left(P_{pub}, H(M)\right)$ **then**

Return $d = 1$ **else Return** $d = 0$**.**

Схема цифровой подписи Boneh – Lynn – Shacham (2)

Стойкость схемы: **схема обеспечивает стойкость к атаке по адаптивно выбираемым сообщениям при предположении о сложности решения задачи CDHP в группе *G*.**

Вычислительная сложность алгоритмов:

1) алгоритм генерации ключей *Gen* – 1 скалярное умножение в группе *G*;

2) алгоритм генерации подписи *Sign* – 1 вычисление функции хэширования + 1 скалярное умножение в группе *G*;

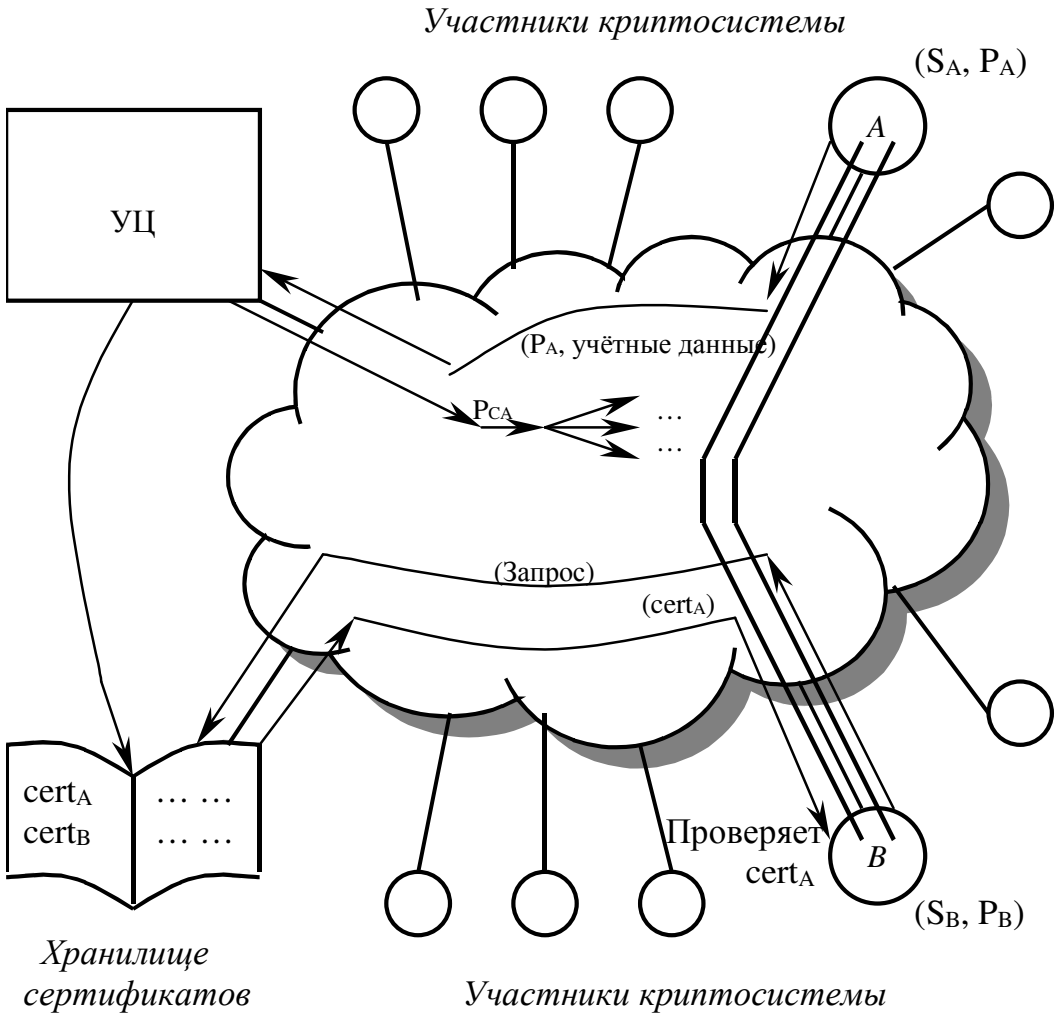
3) алгоритм проверки подписи *Ver* – 1 вычисление функции хэширования + 2 вычисления спариваний.

Идентификационные (identity-based) криптосистемы (1)

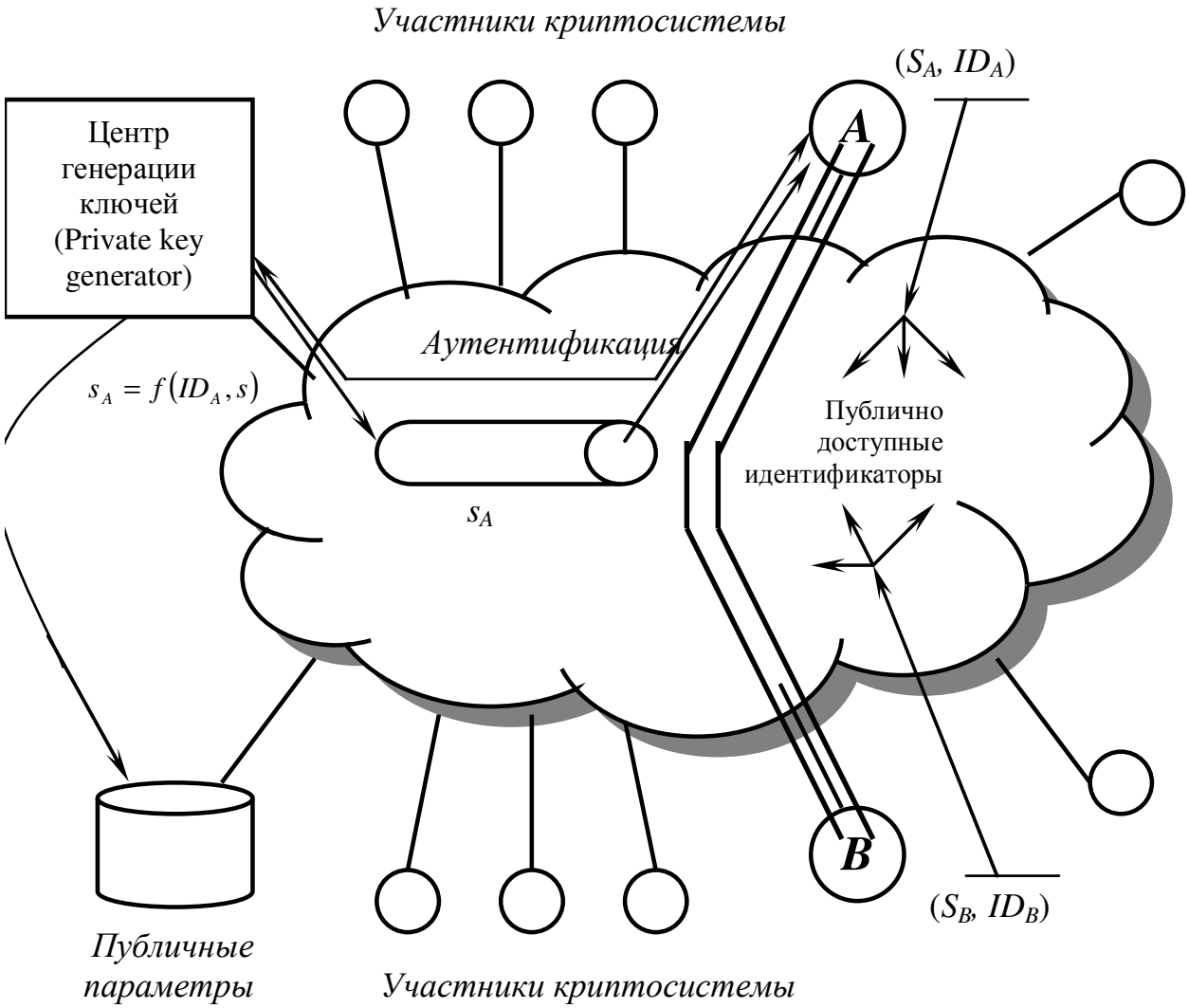
В традиционных схемах открытого шифрования самым большим неудобством является необходимость поддержания справочников аутентичных открытых ключей всех участников криптосистемы. Обычно это достигается при помощи инфраструктуры открытых ключей (PKI). Однако создание PKI само по себе является нетривиальной задачей.

Между тем, давно известная идея идентификационных криптосистем: она предложена Шамиром в 1984 г. Однако первая практически пригодная идентификационная схема открытого шифрования на основе спариваний была предложена Бонеем и Франклином лишь в 2001 г.

Для сравнения: криптосистемы на основе инфраструктуры открытых ключей (PKI)



Идентификационные (identity-based) криптосистемы (2)



Идентификационные (identity-based) криптосистемы (3)

Преимущества:

Нет необходимости поддерживать репозиторий сертификатов и CRL, вместо PKI поддерживается KMI.

Недостатки:

1) Центр генерации ключей PKG знает ключи всех своих клиентов;

2) необходима аутентификация клиентов при обращении к PKG (то же самое в PKIпри обращении к CA);

3) необходим защищенный канал для передачи секретного ключа от PKG к клиенту;

4) в случае многодоменной KMI необходима передача публичных параметров PKG между клиентами

Идентификационная схема открытого шифрования Boneh – Franklin (1)

***PKG* выбирает:**

- **две алгебраических группы:** G, G',
- **спаривание** $e: G \times G \to G'$,
- **функции хэширования** $H: \{0,1\}^* \to G$, $H_1: G' \to \{0,1\}^k$,
- $s \xleftarrow{R} \mathbf{Z}_q^*$ **– это его мастер-ключ,**
- $P \in G$ **– открытый параметр.**

PKG дополнительно вычисляет элемент $P_{pub} = sP \in G$ **и публикует системные параметры** $G, G', e, P, P_{pub}, H, H_1$.

Каждый клиент *A* выбирает при регистрации произвольную двоичную строку ID_A**, не повторяющуюся с другими клиентами, которая будет служить его идентификатором (например, адрес электронной почты).**

***PKG* выдает ему его секретный ключ** $s_A = sQ_A$**, где** $Q_A = H(ID_A)$.

Идентификационная схема открытого шифрования Boneh – Franklin (2)

Алгоритм зашифрования $E_{ID_B}(M)$ **:**

Пусть *B* хочет отправить *A* текст *M*. Открытый текст *M* разбивается на блоки длиной *k* битов каждый: $M = M_1,...,M_n$**, где** $|M_i| = k$**.**

For $i = 1,...,n$ **do**

{ $r_i \overset{R}{\leftarrow} \mathbf{Z}_q^*$ **;**

$U_i = r_i P \in G$**;** $\quad V_i = M_i \oplus H_1\left(e\left(Q_A, P_{pub}\right)^{r_i}\right)$**;** $\quad C_i = (U_i, V_i)$ }

Return $C = C_1,...,C_n$**.**

Алгоритм расшифрования $D_{s_A}(C)$ **:**

Представить шифртекст в виде $C = C_1,...,C_n$**, где** $C_i = (U_i, V_i)$**.**

For $i = 1,...,n$ **do**

{ $W_i = H_1\left(e\left(s_A, U_i\right)\right)$**;**

$M_i = W_i \oplus V_i$ }

Return $M = M_1,...,M_n$**.**

Идентификационная схема открытого шифрования Boneh – Franklin (3)

Доказательство корректности схемы:

$$e\left(Q_A, P_{pub}\right)^{r_i} = e\left(Q_A, sP\right)^{r_i} = e\left(Q_A, P\right)^{r_i s} = e\left(sQ_A, r_i P\right) = e\left(sQ_A, U_i\right) = e\left(s_A, U_i\right).$$

Стойкость схемы:

Доказана теорема, что идентификационная схема открытого шифрования Вонея – Франклина стойка к атакам по адаптивно выбираемым шифртекстам в модели случайного оракула, если решение задачи BDHP в группах G, G' вычислительно сложно.

Вычислительная сложность алгоритмов:

Схема требует 1 вычисления спаривания, нескольких арифметических операций в группе и хэширования для зашифрования и расшифрования сообщения. Размер каждого блока открытого текста увеличивается на длину двоичного представления элемента группы *G*, например, на 160 битов.

Примечание:

Интересная особенность схемы – в том, что сообщение может быть зашифровано и отправлено получателю раньше, чем ему будет выдан секретный ключ.

Идентификационная схема цифровой подписи Hess (1)

Установка начальных параметров:

- $s \overset{R}{\leftarrow} \mathbf{Z}_q^*$ **– мастер-ключ;**
- $P_{pub} = sP \in G$ **– глобальный открытый ключ;**
- **функции хэширования** $H_1 : \{0,1\}^* \to G$ **,** $H : \{0,1\}^* \times G' \to \mathbf{Z}_q^*$**.**

Выработка ключей участника схемы ***Gen*****:**

- $ID \in \{0,1\}^*$ **– открытый идентификатор;**
- $Q_{ID} = H_1(ID)$ **– открытый ключ подписывающего;**
- $S_{ID} = sQ_{ID}$ **– секретный ключ подписывающего.**

Идентификационная схема цифровой подписи Hess (2)

Алгоритм генерации подписи $Sign_{S_{ID}}(M)$ **:**

$P_1 \overset{R}{\leftarrow} G^*$ **– выбирается произвольным образом;**

$k \overset{R}{\leftarrow} \mathbf{Z}_q^*$ **– разовый секретный ключ;**

$r = e(P_1, P)^k$ **;**

$v = H(M, r)$ **;**

$u = vS_{ID} + kP_1$ **;**

Return $\sigma = (u, v) \in G \times Z_q^*$ **.**

Алгоритм проверки подписи $Ver_{Q_{ID}}(M, \sigma)$ **:**

$r = e(u, P) \cdot e(Q_{ID}, P_{pub})^v$ **;**

If $v = H(M, r)$ **then**

Return $d = 1$ **else Return** $d = 0$**.**

Идентификационная схема цифровой подписи Hess (3)

Стойкость схемы:

Схема обеспечивает стойкость к атаке по адаптивно выбираемым сообщениям при предположении о сложности решения «облегчённой» задачи Диффи – Хеллмана в группе *G*:

Дано: (P, Q, sP) **для некоторых** $P, Q \in G$, $s \in \mathbf{Z}_q^*$. ***Найти:*** sQ **.**

Вычислительная сложность алгоритмов:

Установка начальных параметров: 1 скалярное умножение в *G*.

Выработка ключей: 1 вычисление функции хэширования + 1 скалярное умножение в *G*.

Генерация подписи: $e(P_1, P)$ **может быть вычислено предварительно, и тогда остаётся 1 экспоненцирование в группе** G' **+ 1 вычисление функции хэширования + 1 одновременное скалярное умножение в группе *G*.**

Проверка подписи: 1 экспоненцирование в группе G' **+ 1 вычисление функции хэширования + 2 вычисления спариваний, из которых 1 может быть сделано предварительно.**