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.
Доказательства с нулевым разглашением.
Протоколы аутентификации

15 февраля 2018 г.

Доказательства с нулевым разглашением: постановка задачи (1)

Пусть задана интерактивная система доказательства $\langle P,V,S\rangle$. В определении интерактивной системы доказательства ранее не предполагалось, что *V* может быть противником (предполагалась только возможность существования нечестного участника P'). Но *V* может оказаться противником, который хочет выведать у *P* какую-либо новую полезную информацию об утверждении *S*. В этом случае *P* может не хотеть, чтобы это случилось в результате работы протокола интерактивной системы доказательства $\langle P,V,S\rangle$. Таким образом приходим к идее протокола доказательства с *нулевым разглашением* (zero-knowledge proof). Нулевое разглашение подразумевает, что в результате работы протокола интерактивной системы доказательства *V* не увеличит свои знания об утверждении *S*, или, другими словами, не сможет извлечь никакой информации о том, почему *S* истинно.

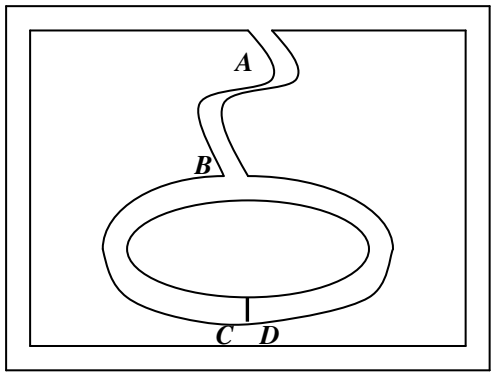
Как и ранее, в протоколе предварительно формулируется некоторое утверждение *S*, например, о том, что некоторый объект *w* обладает свойством *L*: $w \in L$. В ходе протокола *P* и *V* обмениваются сообщениями. Каждый из них может генерировать случайные числа и использовать их в своих вычислениях. В конце протокола *V* должен вынести свое окончательное решение о том, является ли *S* истинным или ложным.

Доказательства с нулевым разглашением: постановка задачи (2)

Цель *P* всегда состоит в том, чтобы убедить *V* в том, что *S* истинно, независимо от того, истинно ли оно на самом деле или нет, т.е. *P* может быть активным противником, а задача *V* – проверять аргументы *P*. Цель участника *V* заключается в том, чтобы вынести решение, является ли *S* истинным или же ложным. Как и ранее, *V* имеет полиномиально ограниченные вычислительные возможности, а именно время его работы ограничено некоторым полиномом от длины доказываемого утверждения: $t \le p(|w|)$. В силу этого он самостоятельно, без помощи *P*, не способен распознать истинность высказывания *S*. Вычислительные возможности *P* никак не ограничиваются.

«Задача о пещере Али-Бабы»

Это модельная задача, наглядно иллюстрирующая суть доказательств с нулевым разглашением. Имеется пещера, план которой показан на рисунке. Пещера имеет дверь с секретом между точками *C* и *D*. Каждый, кто знает волшебные слова, может открыть эту дверь и пройти из *C* в *D* или наоборот. Для всех остальных оба хода пещеры ведут в тупик.



Пусть *P* знает секрет пещеры. Он хочет доказать *V* знание этого секрета, не разглашая волшебные слова. Вот протокол их общения.

1. *V* находится в точке *A*.

2. *P* заходит в пещеру и добирается либо до точки *C*, либо до точки *D*.

3. После того, как *P* исчезает в пещере, *V* приходит в точку *B*, не зная, в какую сторону пошел *P*.

4. *V* зовет *P* и просит его выйти либо из левого, либо из правого коридора пещеры согласно желания *V*.

5. *P* выполняет это, открывая при необходимости дверь, если, конечно, он знает волшебные слова.

6. *P* и *V* повторяют шаги (1) – (5) *n* раз.

Если *P* не знает секрета двери, вероятность того, что *V* попросит его выйти из того же коридора, в который он вошел, равна $\frac{1}{2}$. После *n* раундов вероятность сократится до $\frac{1}{2^n}$.

Протокол доказательства изоморфизма графов

P **хочет доказать** ***V*** **изоморфизм графов** G_0 **и** G_1**. Пусть** $G_1 = \varphi(G_0): G_0 \approx G_1$**, где** φ **- преобразование изоморфизма.** m **– мощность множества** N **вершин графов.**

	P		***V***	
1	π - случайная перестановка вершин, вычисляет $H = \pi G_1$	$\rightarrow$		m раз
2		$\leftarrow$	$\alpha = \{0,1\}$-случ.	
3	Посылает преобразование ψ, такое что: $\psi = \begin{cases} \pi, если(\alpha = 1), \\ \pi \circ \varphi, если(\alpha = 0). \end{cases}$	$\rightarrow$		
4			Вычисляет граф ψG_α и сравнивает: $H \overset{?}{=} \psi G_\alpha$.	
5			Принимает доказательство тогда и только тогда, когда для $\forall m \quad H^{(m)} = \psi G_\alpha^{(m)}$.	

Протокол доказательства знания дискретного логарифма

Перед началом работы протокола задаются открытые величины: *p, q* – простые числа, такие, что $q|(p-1)$, элемент $g \in Z_p^*$, число *X*. Доказывающему *P* известна секретная величина $x : x \in Z_q, g^x = X$, знание которой он должен доказать *V*, не разглашая самой секретной величины.

	P		***V***
1	$r \in_R Z_q$ $M = g^r \pmod p$	$\rightarrow$	
2		$\leftarrow$	$R \in_R Z_q$
3	$m = r + xR \pmod q$	$\rightarrow$	
4			$g^m \stackrel{?}{=} X^R \cdot M \pmod p$

Протокол доказательства знания представления числа в базисе

Перед началом работы протокола задаются открытые величины, известные всем участникам: простые числа *p*, *q*, элементы $y, g_1, g_2, \ldots, g_k \in G_q$. Доказывающему *P* известны секретные величины $\alpha_1, \alpha_2, \ldots, \alpha_k \in Z_q : y = g_1^{\alpha_1} \cdot g_2^{\alpha_2} \ldots \cdot g_k^{\alpha_k}$, знание которых он должен доказать *V*, не разглашая самих этих величин.

	P		***V***
1	$r_1, r_2, \ldots, r_k \in_R Z_q$ $M = g_1^{r_1} \cdot g_2^{r_2} \cdot \ldots \cdot g_k^{r_k}$	$\rightarrow$	
2		$\leftarrow$	$R \in_R Z_q$
3	$m_i = r_i + \alpha_i R, i = \overline{1,k}$	$\rightarrow$	
4			$g_1^{m_1} \cdot g_2^{m_2} \cdot \ldots \cdot g_k^{m_k} \stackrel{?}{=} y^R \cdot M$

Доказательство знания представления множества чисел в соответствующих базисах

Перед началом работы протокола задаются открытые величины, известные всем участникам: простые числа *p,q*, элементы $y^{(j)}, g_1^{(j)}, g_2^{(j)}, ..., g_k^{(j)} \in G_q$ для некоторых *(j)*. Доказывающему *P* известны секретные величины $\alpha_1, \alpha_2, ..., \alpha_k \in Z_q$, такие, что для $\forall j$ $y^{(j)} = \left(g_1^{(j)}\right)^{\alpha_1} \cdot \left(g_2^{(j)}\right)^{\alpha_2} \cdot ... \cdot \left(g_k^{(j)}\right)^{\alpha_k}$, знание которых он должен доказать *V*, не разглашая самих этих величин.

	P		*V*
1	$r_1, r_2, ..., r_k \in_R Z_q$, для $\forall j$ $M^{(j)} = \left(g_1^{(j)}\right)^{r_1} \cdot \left(g_2^{(j)}\right)^{r_2} \cdot ... \cdot \left(g_k^{(j)}\right)^{r_k}$	$\rightarrow$	
2		$\leftarrow$	$R \in_R Z_q$
3	$m_i = r_i + \alpha_i R, i = \overline{1,k}$	$\rightarrow$	
4			для $\forall j$ $\left(g_1^{(j)}\right)^{m_1} \left(g_2^{(j)}\right)^{m_2} \cdot ... \cdot \left(g_k^{(j)}\right)^{m_k} \stackrel{?}{=}$ $\stackrel{?}{=} \left(y^{(j)}\right)^R \cdot M^{(j)}$

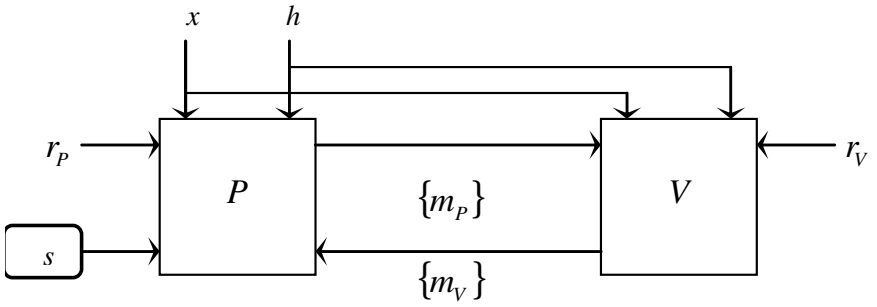
Структура протоколов доказательства с нулевым разглашением

В общем виде протокол интерактивного доказательства с нулевым разглашением состоит из четырех шагов:

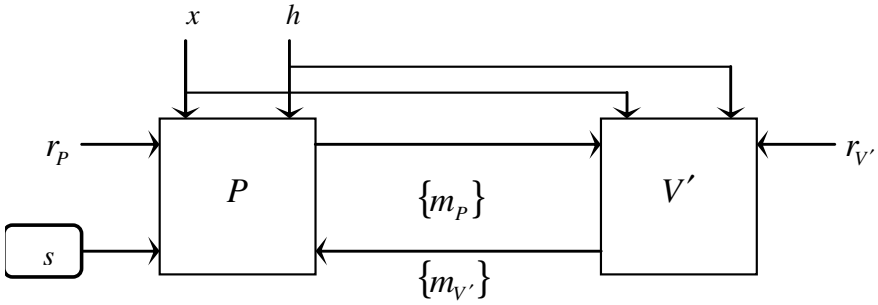
- **доказывающий передает проверяющему *W* – результат вычисления однонаправленной функции от секретной величины, знание которой он доказывает;**
- **проверяющий посылает ему случайный запрос;**
- **доказывающий отвечает на этот запрос, причем ответ зависит как от случайного запроса, так и от секретной величины, но из него вычислительно невозможно получить эту секретную величину;**
- **получая ответ, *V* проверяет его соответствие величине, переданной на первом шаге.**

	P	$S : x \in L$ – доказываемое утверждение, h – др. общедоступные параметры и величины, s – секретные данные доказывающего о том, почему S истинно, r – случ. число	***V***
1	r_P – случ., $W = f_1(x, r_P)$	$\rightarrow$	
2		$\leftarrow$	r_V – случ., $C = f_2(r_V)$
3	$R = f_3(C, x)$	$\rightarrow$	
4			$R \overset{?}{\approx} W$

Свойства доказательств с нулевым разглашением (1)



честный проверяющий



противник M

Пусть $\{m_P\},\{m_V\}$ – совокупность всех сообщений, передаваемых от *P* к *V* (соответственно от *V* к *P*), каждое из которых является случайной величиной, и таким образом, $\{x,h,r_V,\{m_P\},\{m_V\}\}=view_{P,V}(x,h)$ – это ансамбль случайных величин протокола, наблюдаемых извне (внешним наблюдателем), $\{x,h,r_{V'},\{m_P\},\{m_V\}\}=M_{V'}(x,h)$ – это ансамбль случайных величин, получаемых в результате работы полиномиального моделирующего алгоритма (simulator), который выполняется внешним наблюдателем (противником) самостоятельно.

Свойства доказательств с нулевым разглашением (2)

Если величины $view_{P,V}(x,h) \overset{C}{\approx} M_{V'}(x,h)$ ***вычислительно неразличимы*** **за полиномиальное время (т.е. не существует никакого алгоритма, который за полиномиальное время мог бы распознать эти два ансамбля случайных величин), то говорят, что протокол обеспечивает *вычислительно нулевое разглашение* (computationally zero-knowledge).**

Если величины $view_{P,V}(x,h) \approx M_{V'}(x,h)$ ***одинаково распределены*** **над множеством случайных величин, то говорят, что протокол обеспечивает *абсолютно нулевое разглашение* (perfect zero-knowledge).**

Система $\langle P,V,S \rangle$ **называется *интерактивной системой доказательства с нулевым разглашением* для языка *L*, если она:**

1) является интерактивной системой доказательства для языка *L* (т.е. обладает свойствами полноты и корректности);

2) обладает свойством нулевого разглашения.

Теорема 1. **(*Goldreich O., Krawchyk H.*) Последовательное выполнение двух протоколов с нулевым разглашением является протоколом с нулевым разглашением.**

Теорема 2. **(*Goldreich O., Krawchyk H.*) Параллельное выполнение протоколов с нулевым разглашением не обязательно приводит к протоколу с нулевым разглашением.**

Другие виды вероятностных доказательств

Среди всех протоколов доказательства с нулевым разглашением выделяют класс протоколов *доказательства знания* (*proof of knowledge*).

Например, доказательство знания чисел *p*, *q*, таких, что $p \cdot q = n$ есть доказательство знания, но доказательство того, что *n* – составное число, доказательством знания не является – это так называемое *доказательство обладания* (*proof of possession*).

Но доказательство знания не обязательно должно быть доказательством с нулевым разглашением, так как можно просто сообщить секрет другой стороне протокола: при этом сообщивший докажет знание секрета, но тем самым разгласит секрет. В различных приложениях криптографии, в частности, в протоколах аутентификации и в схемах электронных платежей, встречаются протоколы *доказательства знания с нулевым разглашением* (*zero-knowledge proof of knowledge – ZKPK*). Существуют специальные разновидности протоколов доказательства знания с нулевым разглашением: протоколы группового и «скрытого» доказательства знания и др.

Неинтерактивные доказательства с нулевым разглашением **(*non-interactive zero-knowledge proofs*) – однораундовые протоколы доказательства с нулевым разглашением, в которых доказывающий формирует, а проверяющий проверяет доказательство, пользуясь общей ссылочной строкой (common reference string), которая служит заменой случайного запроса проверяющего к доказывающему на шаге (2) обычного интерактивного протокола.**

Протоколы аутентификации

Основные понятия и определения

- ✓ **Идентификация - однозначное именование (присвоение уникальных имён или признаков) компонентов автоматизированной системы и всех лиц (пользователей), взаимодействующих с системой.**
- ✓ **Аутентификация - установление подлинности этих лиц и компонентов системы путём проверки соответствия заявленным ими именам или признакам - идентификаторам.**
- ✓ **Протокол аутентификации – криптографический протокол, в ходе которого одна сторона удостоверяется в идентичности другой стороны, вовлеченной в протокол, а также убеждается в том, что вторая сторона активна во время или непосредственно перед моментом выполнения протокола.**
- ✓ **Принципы аутентификации:**
  - «субъект знает» (логическая);
  - «субъект обладает» (физическая);
  - «субъект есть» (биометрическая).

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:**

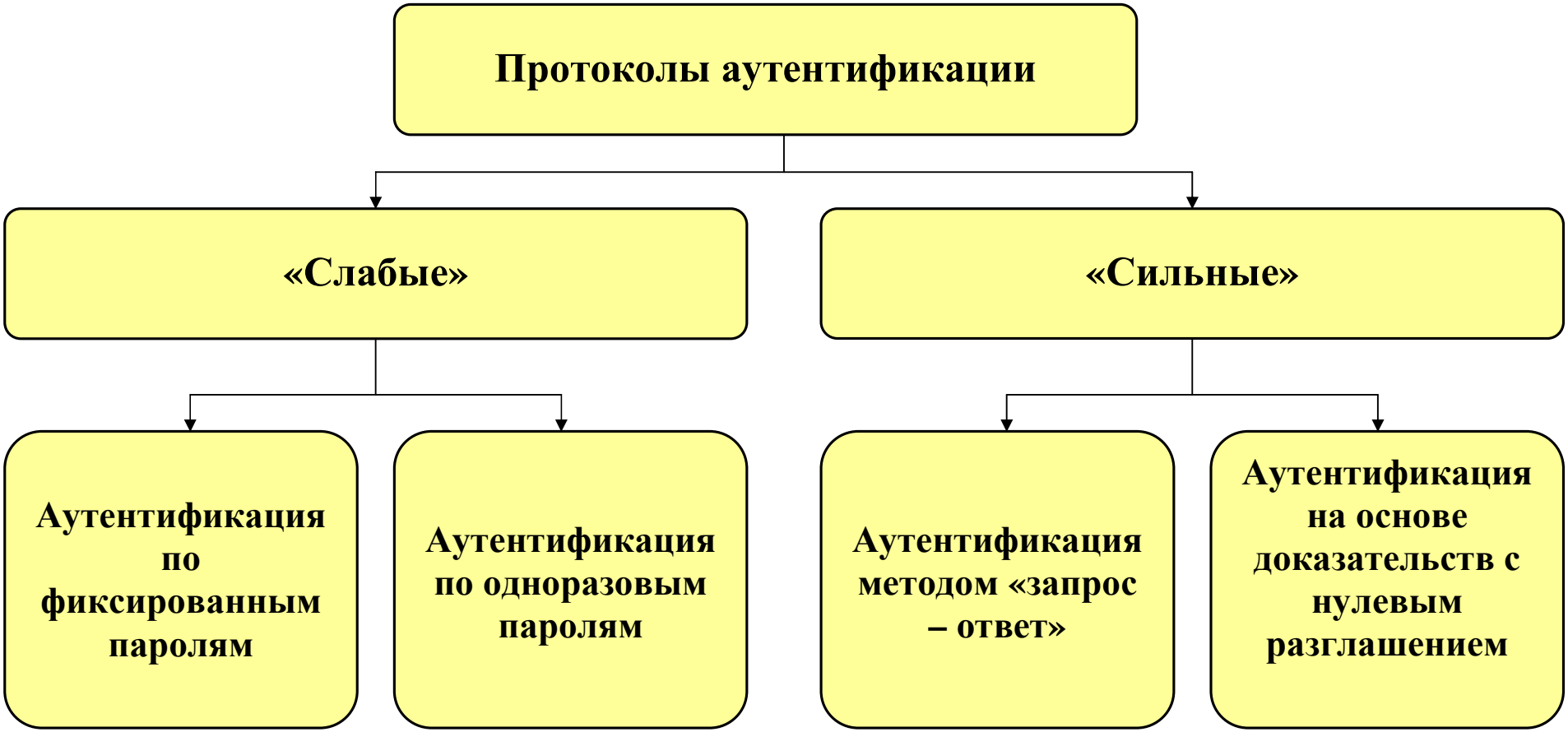
1) если *P* и *V* являются честными, *V* завершит протокол, приняв идентичность *P*;

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



Угрозы и атаки на протоколы парольной аутентификации

Угрозы **протоколам парольной аутентификации:**

- **разглашение пароля;**
- **прослушивание пароля во время выполнения протокола;**
- **угадывание пароля;**
- **восстановление пароля из системной информации.**

Атаки **на парольные протоколы:**

- **повтор паролей легальных пользователей злоумышленниками;**
- **полный перебор паролей;**
- **словарная атака на протокол аутентификации.**

Аутентификация по фиксированным паролям

Приёмы повышения стойкости протоколов аутентификации по фиксированным паролям:

- **хранение в компьютерной системе файлов паролей в защищенном режиме (с защитой от чтения/записи);**
- **хранение в системе не самих паролей, а их образов, полученных как результат вычисления однонаправленной функции от пароля, взятого в качестве аргумента;**
- **задание правил выбора паролей (минимальное количество символов, недопущение использования осмысленных слов, необходимость сочетания букв и цифр и т.п.), имеющих целью максимизировать энтропию пароля;**
- **искусственное замедление процесса ввода пароля в систему с целью резкого увеличения времени на перебор паролей;**
- **выбор в качестве пароля осмысленного предложения (фразы) с последующим преобразованием посредством хэш-функции в короткое сообщение, которое обычно обладает большей энтропией, чем пароль такой же длины, выбираемый человеком;**
- **добавление системой случайной величины к паролю перед обработкой его однонаправленной функцией – «метод солтинга»;**
- **многофакторная аутентификация.**

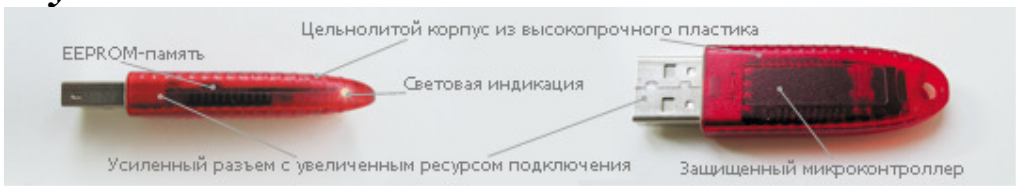
Многофакторная аутентификация

Частный случай фиксированного пароля – *PIN-код*. Используется в сочетании с физической аутентификацией посредством обладания смарт-картой или токеном.

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

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»:

Аутентификация по одноразовым паролям (1)

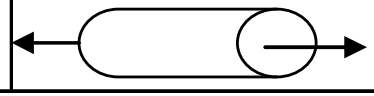
1. Разделяемые списки одноразовых паролей. **Пользователь и система имеют заранее определенную таблицу (список, карту) паролей, которую каждый из них хранит самостоятельно. При выполнении очередного сеанса протокола аутентификации выбирается пользователем и проверяется системой очередной пароль из этого списка.**

2. Последовательно обновляемые одноразовые пароли. **Первоначально пользователь и система имеют только один пароль, условно с номером *i*. Затем пользователь создает и передает системе пароль под номером *i*-1, зашифрованный на ключе, вычисленном из *i*-го пароля. Такой метод затруднительно реализовать при ненадежном канале связи (при возможности обрыва связи).**

3. Последовательности одноразовых паролей, основанные на однонаправленных функциях. **Этот метод наиболее эффективен по отношению к объему передаваемых данных. Примером является *протокол Лампорта* (RFC 1760 – The S/KEY One-Time Password System).**

Аутентификация по одноразовым паролям (2)

Протокол смены пароля для аутентификации по последовательно обновляемым одноразовым паролям

P		***V***
pwd_i		pwd_i
P и *V* в течение условленного времени используют пароль pwd_i в качестве общего секрета		
$k_i = f(pwd_i)$		$k_i = f(pwd_i)$
Выбирает pwd_{i-1}		
$C_{i-1} = E_{k_i}(pwd_{i-1})$	$\rightarrow$	$pwd_{i-1} = D_{k_i}(C_{i-1})$
P и *V* в течение условленного времени используют пароль pwd_{i-1} в качестве общего секрета		
$k_{i-1} = f(pwd_{i-1})$		$k_{i-1} = f(pwd_{i-1})$
Выбирает pwd_{i-2}		
$C_{i-2} = E_{k_{i-1}}(pwd_{i-2})$	$\rightarrow$	$pwd_{i-2} = D_{k_{i-1}}(C_{i-2})$
P и *V* в течение условленного времени используют пароль pwd_{i-2} в качестве общего секрета		
…	и т.д.	…

Аутентификация по одноразовым паролям (3)

Протокол Лампорта аутентификации по одноразовым паролям

<table>
<tr><th colspan="4">Предварительный этап</th></tr>
<tr><th colspan="2">P</th><th></th><th>V</th></tr>
<tr><td colspan="2">1. Выбираются: w – секрет пользователя P, H – однонаправленная хеш-функция, t – фиксированная константа, определяющая число разрешенных сеансов аутентификации, после чего P меняет свой секрет.
2. Вычисляет $w_0 = H^t(w)$ и передает ее V по секретному, аутентичному каналу.
3. Вычисляет $H^t(w) = H(H(...(H(w))...))$.</td><td>→</td><td>I_P=1 – счетчик для P</td></tr>
<tr><th colspan="4">Рабочий этап</th></tr>
<tr><th></th><th>P</th><th></th><th>V</th></tr>
<tr><td colspan="4">Для i=1,…,t:</td></tr>
<tr><td>1</td><td>Вычисляет $w_i = H^{t-i}(w)$ либо из w, либо из промежуточной величины, сохраненной во время вычисления $H^t(w)$.
Направляет V сообщение $[P, i, w_i]$, где i – номер сеанса аутентификации.</td><td>→</td><td></td></tr>
<tr><td>2</td><td></td><td></td><td>$\begin{cases} i \overset{?}{=} i_P, \\ H(w_i) \overset{?}{=} w_{i-1} \end{cases}$
Если да, то $i_P := i_P + 1$ (увеличивает</td></tr>
</table>

Протоколы аутентификации «запрос – ответ»

Идея, заложенная в основу *протоколов аутентификации типа «запрос – ответ» (challenge – response)*, заключается в том, что претендент доказывает свою идентичность проверяющему путем демонстрации знания некоторого секрета. В некоторых протоколах секрет известен проверяющему и используется для проверки ответа, в других – вообще нет необходимости, чтобы секрет был известен проверяющему. При выполнении протокола претендент должен ответить на запрос, меняющийся от сеанса к сеансу, причем ответ должен зависеть и и от запроса, и от известного ему секрета.

Запрос – это обычно некоторая переменная величина, выбираемая проверяющим в начале протокола. Если линия связи между участниками протокола прослушивается противником, то ответ претендента не должен снабжать противника полезной для него информацией, которая может быть использована в последующих сеансах протокола. Для этого все запросы проверяющего обязательно должны отличаться друг от друга.

В качестве изменяющихся от сеанса к сеансу параметров запроса могут использоваться три типа величин: *случайные числа*, *числовые последовательности* и *метки времени*.

Стандарты по аутентификации «запрос – ответ»

- **ISO/IEC 9798-1: 2010 – General;**
- **ISO/IEC 9798-2: 2008 – Mechanisms using symmetric encipherment algorithms (к этой части стандарта имеются дополнения и поправки 2010 и 2012 гг.);**
- **ISO/IEC 9798-3: 1998 – Entity authentication using digital signature technique (к этой части стандарта имеются дополнения и поправки 2009 и 2010 гг.);**
- **ISO/IEC 9798-4: 1999 – Mechanisms using a cryptographic check function (к этой части стандарта имеются дополнения и поправки 2009 и 2012 гг.);**
- **ISO/IEC 9798-5: 2009 – Mechanisms using zero knowledge techniques;**
- **ISO/IEC 9798-6: 2010 – Mechanisms using manual data transfer.**

Протоколы «запрос – ответ» с использованием симметричных криптосхем (1)

Односторонняя аутентификация с использованием случайного числа:

P	K – общий секр. ключ	***V***
	$\leftarrow [r_V] -$	r_V – случ.
$C = E_K(r_V, V)$	$-[C] \rightarrow$	$M = D_K(C),\ M = (r_V', V'),$ If { $r_V' = r_V$ and $V' = V$ } then {P принимается как аутентичный}

Односторонняя аутентификация с использованием метки времени:

P	K – общий секр. ключ	***V***
$C = E_K(t_P, V)$	$-[C] \rightarrow$	$M = D_K(C),\ M = (t_P', V'),$ If { $V' = V$ and $\lvert t_P' - t_V \rvert \le w$, где w – окно принятия} then {претендент принимается как аутентичный}

Протоколы «запрос – ответ» с использованием симметричных криптосхем (2)

Протокол взаимной аутентификации («протокол рукопожатия») с использованием случайных чисел:

P	K – общий секр. ключ	***V***
	$\leftarrow [r_V] -$	r_V – случ.
r_P – случ. $C_1 = E_K(r_P, r_V, V)$	$-[C_1] \rightarrow$	$M_1 = D_K(C_1)$, $M_1 = (r_P', r_V', V')$, If { $r_V' = r_V$ and $V' = V$ } then {P принимается как аутентичный}
$M_2 = D_K(C_2)$, $M_2 = (r_P'', r_V'')$, If { $r_P'' = r_P$ and $r_V'' = r_V$ } then {V принимается как аутентичный}	$\leftarrow [C_2] -$	$C_2 = E_K(r_V, r_P)$

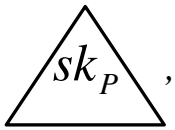
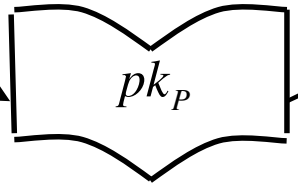
Протоколы «запрос – ответ» с использованием симметричных криптосхем (3)

Протокол взаимной аутентификации с использованием случайных чисел (вариант с хэш-функциями)

P	K – общий секр. ключ	***V***
	$\leftarrow [r_V] -$	r_V – случ.
r_P – случ. $H_1 = h_K(r_P, r_V, V)$	$-[r_P, H_1] \rightarrow$	$H_1' = h_K(r_P, r_V, V)$, If { $H' = H$ } then {P принимается как аутентичный}
$H_2' = h_K(r_V, r_P, P)$, If { $H_2' = H_2$ } then {V принимается как аутентичный}	$\leftarrow [H_2] -$	$H_2 = h_K(r_V, r_P, P)$

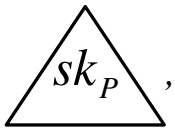
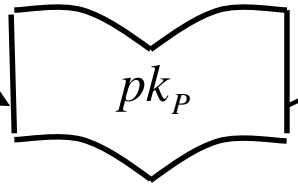
Протоколы «запрос – ответ» с использованием асимметричных криптосхем (1)

Протокол односторонней аутентификации с использованием схемы цифровой подписи (случайные числа)

P		*V*
Генерация пары ключей:  	 	Получение открытого ключа из справочника: 
	$\leftarrow [r_V] -$	r_V – случ.
r_P – случ., $Q = Sign_{sk_P}(r_P, r_V, V)$	$-[Q, V, r_P, cert_P] \rightarrow$	$Ver_{pk_P}(Q) \stackrel{?}{=} \{0/1\}$, If { $Ver_{pk_P}(Q) = 1$ } then {P принимается как аутентичный}

Протоколы «запрос – ответ» с использованием асимметричных криптосхем (2)

Протокол односторонней аутентификации с использованием схемы цифровой подписи (метка времени)

P		***V***
Генерация пары ключей:  получение $cert_P$	 	Получение открытого ключа из справочника: 
Получение t_P , $Q = Sign_{sk_P}(t_P, V)$	$-[Q, V, t_P, cert_P]\rightarrow$	$Ver_{pk_P}(Q) \stackrel{?}{=} \{0/1\}$, If { $Ver_{pk_P}(Q) = 1$ } then {P принимается как аутентичный}

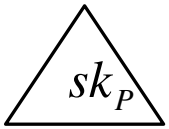
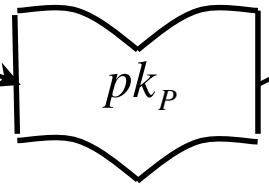
Протоколы «запрос – ответ» с использованием асимметричных криптосхем (3)

Протокол взаимной аутентификации с использованием схем цифровой подписи

P		*V*
Генерация пары ключей: sk_P, pk_P, получение $cert_P$	pk_P, pk_V	Генерация пары ключей: pk_V, sk_V, получение $cert_V$
Получение открытого ключа из справочника: pk_V		Получение открытого ключа из справочника: pk_P
	$\leftarrow [r_V] -$	r_V – случ.
r_P – случ., $Q_1 = Sign_{sk_P}(r_P, r_V, V)$	$-[Q_1, V, r_P, cert_P] \rightarrow$	$Ver_{pk_P}(Q_1) \stackrel{?}{=} \{0/1\}$, If $\{ Ver_{pk_P}(Q_1) = 1 \}$ then {*P* принимается как аутентичный}
$Ver_{pk_V}(Q_2) \stackrel{?}{=} \{0/1\}$, If $\{ Ver_{pk_V}(Q_2) = 1 \}$ then {*V* принимается как аутентичный}	$\leftarrow [Q_2, P, cert_V] -$	$Q_2 = Sign_{sk_V}(r_V, r_P, V)$

Протоколы «запрос – ответ» с использованием асимметричных криптосхем (4)

Протокол односторонней аутентификации с использованием схем открытого шифрования

P		*V*
Генерация пары ключей:  , pk_P , получение $cert_P$	 	Получение открытого ключа из справочника: 
$M = D_{sk_P}(C)$, $M = (r', V')$, $x' = h(r')$	$\leftarrow [x, V, C]-$	r – случ., $x = h(r)$, $C = E_{pk_P}(r, V)$
If { $V' = V$ and $x' = x$ } then {P продолжает протокол}	$-[r'] \rightarrow$	If { $r' = r$ } then {P принимается как аутентичный}

Протоколы «запрос – ответ» с использованием асимметричных криптосхем (5)

Протокол взаимной аутентификации с использованием схем открытого шифрования

P		*V*
Генерация пары ключей: sk_P, pk_P, получение $cert_P$	pk_P, pk_V	Генерация пары ключей: pk_V, sk_V, получение $cert_V$
Получение открытого ключа из справочника: pk_V		Получение открытого ключа из справочника: pk_P
r_1 – случ., $C_1 = E_{pk_V}(r_1, P)$	$-[r_1, C_1]\rightarrow$	$M_1 = D_{sk_V}(C_1)$, $M_1 = (r_1', P')$, If { $P' = P$ } then {V продолжает протокол}
$M_2 = D_{sk_P}(C_2)$, $M_2 = (r_1'', r_2')$,	$\leftarrow[r_2, C_2]-$	r_2 – случ., $C_2 = E_{pk_P}(r_1, r_2)$
If { $r_1'' = r_1$ } then {V принимается как аутентичный}	$-[r_2']\rightarrow$	If { $r_2' = r_2$ } then {P принимается как аутентичный}