Лекция 6. Электронная цифровая подпись

Вопросы:

1. Концепция цифровой подписи

2. Назначение, основные свойства хэш-функций

3. Алгоритм цифровой подписи RSA

4. Управление криптографическими ключами

1. Концепция цифровой подписи

Цифровая подпись (ЦП) является аналогом подписи, сделанной от руки. Она должна обеспечивать следующие возможности:

- возможность установить автора, а также дату и время подписи;
- возможность установить достоверность содержимого сообщения на время подписи:
- возможность проверки подписи третьей стороной на случай возникновения спора.

Требования к цифровой подписи:

- подпись должна быть двоичным кодом, зависящим от подписываемого сообщения;
- подпись должна использовать некоторую информацию, уникальную для отправителя, чтобы предотвратить возможность, как фальсификации, так и отрицания авторства;
- цифровую подпись можно относительно просто произвести;
- цифровую подпись можно относительно просто распознать и проверить;
- с точки зрения вычислений нереально фальсифицировать цифровую подпись ни с помощью создания нового сообщения для имеющейся цифровой подписи, ни с помощью создания фальшивой цифровой подписи для имеющегося сообщения.

При формировании ЦП отправитель первым делом вычисляет хэш-функцию $h(M)$ подписываемого текста M. Вычисленное значение хэш-функции $h(M)$ представляет собой короткий блок информации m, характеризующий весь текст M в целом.

Затем число m шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой электронную цифровую подпись для данного сообщения M.

При проверке ЦП получатель сообщения снова вычисляет хэш-функцию $m = h(M)$ принятого по каналу текста M, затем при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению m хэш-функции.

Без знания секретного ключа подписывания отправителя нельзя подделать ЦП. В качестве подписываемого документа можно использовать любой файл.

Подписанный файл создается из неподписанного путем добавления в него одной или более электронных подписей, которые должны содержать следующую информацию:

- дату подписи;
- срок окончания действия ключа данной подписи;
- информацию о лице, подписавшим файл (Ф.И.О., должность, краткое наименование фирмы);
- идентификатор подписавшего (имя открытого ключа);
- собственно цифровую подпись.

2. Назначение, основные свойства хэш-функций

Хэш-функция (англ. ***hash*** – мелко измельчать и перемешивать) предназначена для сжатия подписываемого документа до нескольких десятков или сотен бит. Хэш-функция h(·) принимает в качестве аргумента сообщение (документ) **M** произвольной длины и возвращает хэш-значение **h(M)=H** фиксированной длины. Обычно хэшированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Следует отметить, что значение хэш-функции **h(M)** сложным образом зависит от документа **M** и не позволяет восстановить сам документ **M**.

Хэш-функция должна удовлетворять целому ряду условий:

- хэш-функция должна быть чувствительна к всевозможным изменениям в тексте M, таким как вставки, выбросы, перестановки и т.п.;
- хэш-функция должна обладать свойством необратимости, то есть задача подбора документа **M'**, который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима;
- вероятность того, что значения хэш-функций двух различных

Большинство хэш-функций строится на основе однонаправленной функции f(·), которая образует выходное значение длиной **n** при задании двух входных значений длиной **n**. Этими входами являются блок исходного текста **M,** и хэш-значение $\mathbf{H_{i-1}}$ предыдущего блока текста.

Хэш-значение, вычисляемое при вводе последнего блока текста, становится хэш-значением всего сообщения **M**.

В результате однонаправленная хэш-функция всегда формирует выход фиксированной длины **n** (независимо от длины входного текста).

Общепринятым принципом построения хэш-функций является ***итеративная последовательная схема***. По этой методике ядром алгоритма является преобразование k бит в n бит. Величина n – разрядность результата хэш-функции, а k – произвольное число, большее n. Базовое преобразование должно обладать всеми свойствами хэш-функции т.е. необратимостью и невозможностью инвариантного изменения входных данных.

3. Алгоритм цифровой подписи RSA

Действия отправителя:

1. Вычисление секретного и открытого ключей, для чего отправитель электронных документов выбирает два больших простых числа P и Q, находит их произведение $N = P \cdot Q$.

Затем вычисляет значение функции Эйлера $\varphi(N)$:

$$\varphi(N) = (P-1)(Q-1)$$

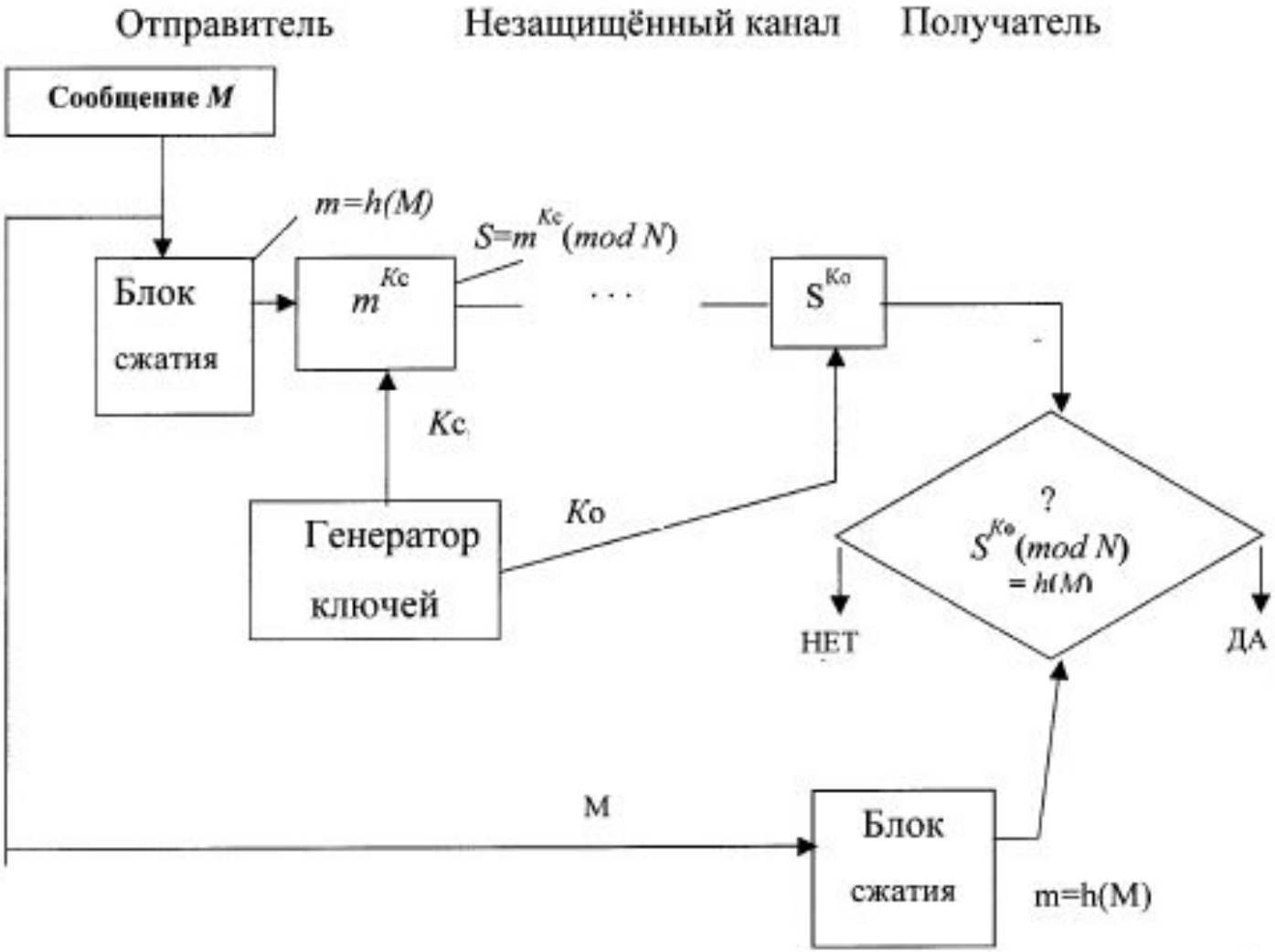
2. Вычисление открытого ключа K_o из условий:

$$K_o \le \varphi(N),\ \text{НОД}\ (K_o, \varphi(N)) = 1,$$

и секретного ключа K_c из условий:

$$K_c < N,\ K_o \cdot K_c \equiv 1 (mod\ \varphi(N)).$$

3. Передача пользователем сети пары чисел (K_o, N), которая является открытым ключом, партнерам по переписке для проверки его цифровой подписи. Число K_c сохраняется отправителем как секретный ключ для подписывания.



Обобщенная схема цифровой подписи *RSA*

Для формирования ЦП по схеме *RSA* необходимо:

1. Сообщение M (блок информации, файл и т.д.) сжать с помощью хэш-функции $h(M)$ в целое число m.

2. Вычислить цифровую подпись S под электронным документом M, используя хэш-значение m и секретный ключ K_c:

$$S = m^{K_c} \ (mod\ N).$$

3. Передать получателю электронный документ M, подписанный цифровой подписью S.

Действия получателя:

1. После приёма пары (M, S) получатель вычисляет хэш-значение сообщения M как $m = h(M)$ и восстанавливает хэш-значение m' цифровой подписи S по формуле

$$m' = S^{K_o} \ (mod\ N).$$

2. Если $S^{K_o}(mod\ N) = h(M)$, то получатель признаёт пару (M, S) подлинной.

Пример.

$$H_i = [(H_{i-1} \oplus M_i)^2] \ (mod\ N),$$

где H_o –вектор инициализации; $M_i = M_1, M_2, \ldots, M_n$.

Хешируемое сообщение «531». Выбираем числа $P = 7$, $Q = 3$, $H_o = 6$. Определяем $N = P \cdot Q = 7 \cdot 3 = 21$. Вычисляем хэш-код сообщения 531 поблочно по формуле (6.1).

1. $M_1 + H_o = 5 + 6 = 11;$

$[M_1 + H_o]^2 \ (mod\ N) = 11^2 \ (mod\ 21) = 16 = H_1;$

2. $M_2 + H_1 = 3 + 16 = 19;$

$[M_2 + H_1]^2 \ (mod\ N) = 19^2 (mod\ 21) = 4 = H_2;$

3. $M_3 + H_2 = 1 + 4 = 5;$

$[M_3 + H_2]^2 \ (mod\ N) = 5^2 \ (mod\ 21) = 4 = H_3.$

В итоге получаем хэш-значение сообщения «531», равное 4.

Пример.

Получить хэш-код для сообщения «HESHING» при помощи хэш-функции, вычисляемой по формуле (1). Выбираем числа $P = 17$, $Q = 19$.

Порядок вычисления хэш-кода:

1) вычисляем значение модуля $N = P \cdot Q = 323$;

2) представляем сообщение «HASHING» в виде символов *ASC* II:

3)

H	*A*	*S*	*H*	*I*	*N*	*G*
72	65	83	72	73	78	71

4) представляем коды ASC II битовой строкой:

72	65	83	72	77	78	71
01001000	01000001	01010011	01001000	01001001	01001110	01000111

5) разбиваем байт пополам, затем добавляем в начало полубайта единицы и получаем хэшируемые блоки M_i :

M_1	M_2	M_3	M_4	M_5	M_6	M_7
11110100	11111000	11110100	11110001	11110101	11110011	11110100
M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
11111000	11110100	11111001	11110100	11111110	11110100	11110111

Кодовая таблица ASCII

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	null	▶	Space	0	@	P	‘	p	А	Р	а	▒	└	╨	р	Ё
1	☺	◀	!	1	A	Q	a	q	Б	С	б	▒	┴	╤	с	ё
2	☻	↕	"	2	B	R	b	r	В	Т	в	▓	┬	╥	т	Є
3	♥	‼	#	3	C	S	c	s	Г	У	г	│	├	╙	у	є
4	♦	¶	$	4	D	T	d	t	Д	Ф	д	┤	─	╘	ф	Ї
5	♣	§	%	5	E	U	e	u	Е	Х	е	╡	┼	╒	х	ї
6	♠	_	&	6	F	V	f	v	Ж	Ц	ж	╢	╞	╓	ц	Ў
7	•	↨	`	7	G	W	g	w	З	Ч	з	╖	╟	╫	ч	ў
8	Bsp	↑	(	8	H	X	h	x	И	Ш	и	╕	╚	╪	ш	°
9	Tab	↓	)	9	I	Y	i	y	Й	Щ	й	╣	╔	┘	щ	•
A	◙	→	*	:	J	Z	j	z	К	Ъ	к	║	╩	┌	ъ	·
B	♂	Esk	+	;	K	[	k	{	Л	Ы	л	╗	╦	█	ы	√
C	♀	∟	,	<	L	\	l	|	М	Ь	м	╝	╠	▄	ь	№
D	♪	↔	-	=	M	]	m	}	Н	Э	н	╜	═	▌	э	¤
E	♫	▲	.	>	N	^	n	~	О	Ю	о	╛	╬	▐	ю	■
F	☼	▼	/	?	O	_	o	⌂	П	Я	п	┐	╧	▀	я	blank

$$H_i = [(H_{i-1} \oplus M_i)^2] \ (mod\ N),$$

1. Вычисляем H_1

M_1	$= 11110100$
$\oplus$	
H_o	$= \underline{00000000}$
$H_o \oplus M_1$	$= 11110100_2 = 244_{10}$
$[(H_o \oplus M_1)]^2 \ (mod\ 323)$	$= 244^2 \ (mod\ 323) = 104$
H_1	$= 104_{10} = 01101000_2$

2. Вычисляем H_2

M_2	$= 11111000$
$\oplus$	
H_1	$= \underline{01101000}$
$H_1 \oplus M_2$	$= 10010000_2 = 144_{10}$
$[(H_1 \oplus M_2)]^2 \ (mod\ 323)]$	$= 144^2 \ (mod\ 323) = 64$
H_2	$= 64_{10} = 01000000_2$

. . . . . .

Вычисляем H_i. . . . . и так далее.

4. Управление криптографическими ключами

Управление ключами состоит из процедур, обеспечивающих:

1) включение пользователей в систему;
2) выработку, распределение и введение в аппаратуру ключей;
3) контроль использования ключей;
4) смену и уничтожение ключей;
5) архивирование, хранение и восстановление ключей.

Целью управления ключами является нейтрализация таких угроз, как:

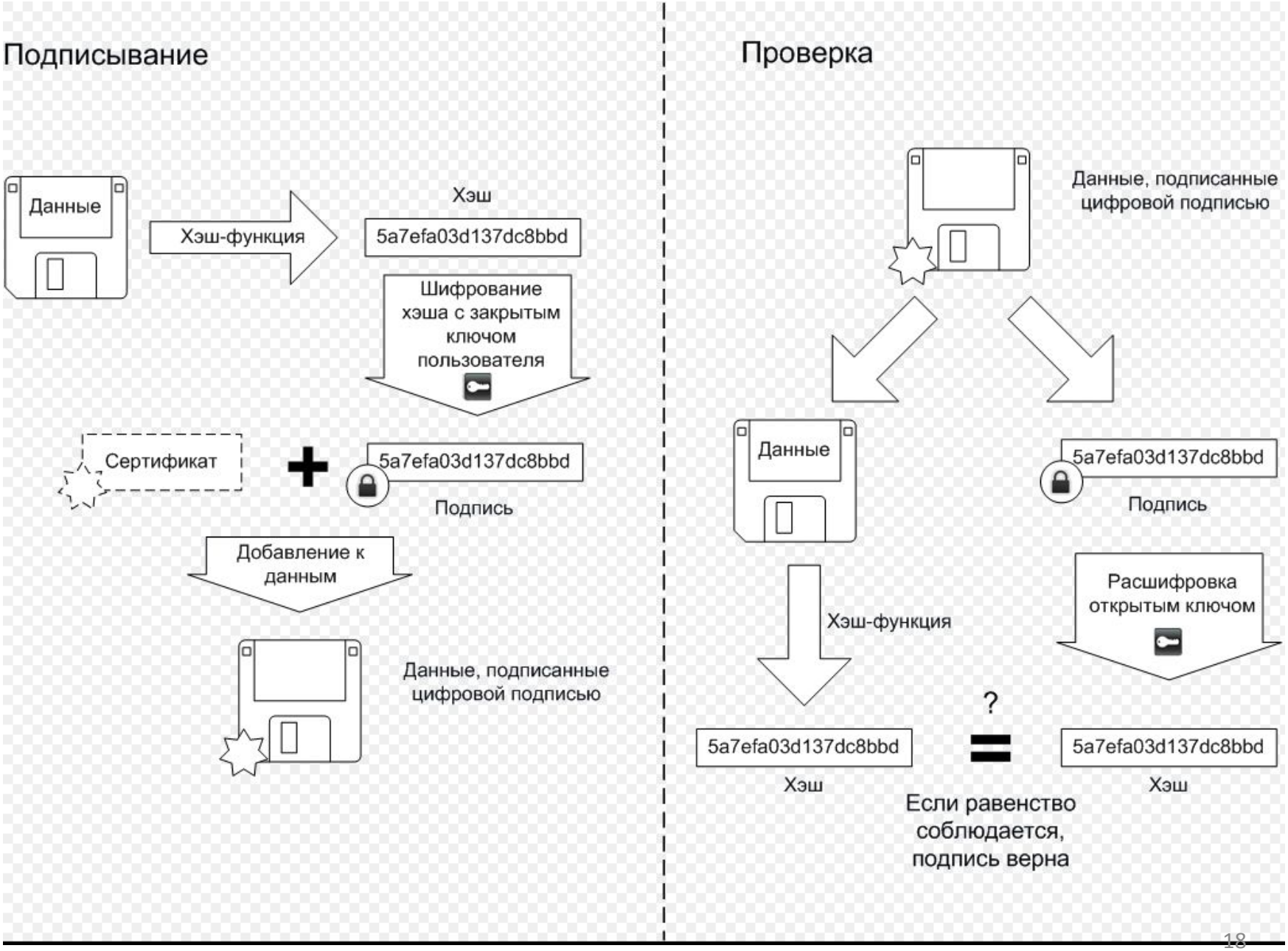
- компрометация конфиденциальности закрытых ключей;
- компрометация аутентичности закрытых или открытых ключей. При этом под аутентичностью понимается знание или возможность проверки идентичности корреспондента, для обеспечения конфиденциальной связи с которым используется данный ключ;
- несанкционированное использование закрытых или открытых ключей, например использование ключа, срок действия которого истек.

Главный ключ – высший ключ в иерархии, который не защищается криптографически. Его защита осуществляется с помощью физических или электронных средств.

Ключи для шифрования ключей – закрытые или открытые ключи, используемые для засекречивания перед передачей или при хранении других шифровальных ключей. Эти ключи сами могут быть зашифрованы с помощью других ключей.

Ключи для шифрования данных – используются для защиты данных пользователей.

Ключи более высоких уровней используются для защиты ключей или данных на более низких уровнях, что уменьшает ущерб при раскрытии ключей и объём необходимой информации, нуждающейся в физической защите.


Подписывание
Данные
Хэш-функция
Хэш
5a7efa03d137dc8bbd
Шифрование хэша с закрытым ключом пользователя
Сертификат
5a7efa03d137dc8bbd
Подпись
Добавление к данным
Данные, подписанные цифровой подписью
Проверка
Данные, подписанные цифровой подписью
Данные
5a7efa03d137dc8bbd
Подпись
Хэш-функция
Расшифровка открытым ключом
?
5a7efa03d137dc8bbd
Хэш
5a7efa03d137dc8bbd
Хэш
Если равенство соблюдается, подпись верна