ГОСТ Р 34.10—94

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

ПРОЦЕДУРЫ ВЫРАБОТКИ И ПРОВЕРКИ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ НА БАЗЕ АСИММЕТРИЧНОГО КРИПТОГРАФИЧЕСКОГО АЛГОРИТМА

Издание официальное

БЗ 3—94/130

ГОССТАНДАРТ РОССИИ
Москва

Предисловие

1 РАЗРАБОТАН Главным управлением безопасности связи Федерального агентства правительственной связи и информации и Всероссийским научно-исследовательским институтом стандартизации

ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационная технология» и Федеральным агентством правительственной связи и информации

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 23.05.94 № 154

3 ВВЕДЕН ВПЕРВЫЕ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Расширяющееся применение информационных технологий при создании, обработке, передаче и хранении документов требует в определенных случаях сохранения конфиденциальности их содержания, обеспечения полноты и достоверности.

Одним из эффективных направлений защиты информации является криптография (криптографическая защита), широко применяемая в различных сферах деятельности в государственных и коммерческих структурах.

Криптографические методы защиты информации являются объектом серьезных научных исследований и стандартизации на национальных, региональных и международных уровнях.

Настоящий стандарт определяет процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма с применением функции хэширования.

Электронная цифровая подпись обеспечивает целостность сообщений (документов), передаваемых по незащищенным телекоммуникационным каналам общего пользования в системах обработки информации различного назначения, с гарантированной идентификацией ее автора (лица, подписавшего документ).

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология. КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма.

Information technology.
Cryptographic Data Security.
Produce and check procedures of Electronic Digital Signature based on Asymmetric Cryptographic Algorithm.

Дата введения 1995—01—01

1 ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий стандарт устанавливает процедуры выработки и проверки электронной цифровой подписи (ЭЦП) сообщений (документов), передаваемых по незащищенным телекоммуникационным каналам общего пользования в системах обработки информации различного назначения, на базе асимметричного криптографического алгоритма с применением функции хэширования.

Внедрение системы ЭЦП на базе настоящего стандарта обеспечивает защиту передаваемых сообщений от подделки, искажения и однозначно позволяет доказательно подтвердить подпись лица, подписавшего сообщение.

2 НОРМАТИВНЫЕ ССЫЛКИ

В настоящем стандарте использованы ссылки на следующий стандарт:

ГОСТ Р 34.11—94 Информационная технология. Криптографическая защита информации. Функция хэширования.

3 ОБОЗНАЧЕНИЯ

В настоящем стандарте используются следующие обозначения.

β^* — множество всех конечных слов в алфавите $\beta=\{0,1\}$.

$|A|$ — длина слова $A \in \beta^*$.

$V_k(2)$ — множество всех бинарных слов длины k.

Издание официальное

$z \pmod n$ — наименьшее по значению неотрицательное число, сравнимое с z по модулю числа n.

$<N>_k$ — слово длины k, содержащее двоичную запись вычета $N \pmod{2^k}$ неотрицательного целого числа N.

$\hat{A}$ — неотрицательное целое число, имеющее двоичную запись A $(A \in \beta^*)$ (под длиной числа будем понимать номер старшего значащего бита в двоичной записи числа).

$A \| B$ — конкатенация слов A, $B \in \beta^*$ — слово длины $|A|+|B|$, в котором левые $|A|$ символов образуют слово A, а правые $|B|$ символов образуют слово B. Можно также использовать обозначение $A \| B = AB$.

A^k — конкатенация k экземпляров слова A $(A \in \beta^*)$.

M — передаваемое сообщение, $M \in \beta^*$.

M_1 — полученное сообщение, $M_1 \in \beta^*$.[1)]

h — хэш-функция, отображающая сообщение M в слово $h(M) \in V_{256}(2)$.

p — простое число, $2^{509} < p < 2^{512}$ либо $2^{1020} < p < 2^{1024}$.

q — простое число, $2^{254} < q < 2^{256}$ и q является делителем для $(p-1)$.

a — целое число, $1 < a < p-1$, при этом $a^q \pmod p = 1$.

k — целое число, $0 < k < q$.

$\lceil d \rceil$ — наименьшее целое число, не меньшее чем d.

$\lfloor d \rfloor$ — наибольшее целое число, не большее чем d.

$e := g$ — присвоение параметру e значения g.

x — секретный ключ пользователя для формирования подписи, $0 < x < q$.

y — открытый ключ пользователя для проверки подписи, $y = a^x \pmod p$.

4 ОБЩИЕ ПОЛОЖЕНИЯ

Система ЭЦП базируется на методах криптографической защиты данных с использованием хэш-функции.

Алгоритм вычисления функции хэширования установлен в ГОСТ Р 34.11.

Процедуры цифровой подписи допускают как программную, так и аппаратную реализацию.

Система ЭЦП включает в себя процедуры выработки и проверки подписи под данным сообщением.

[1)] Отправляемые и получаемые последовательности, в том числе сообщения и подписи, могут отличаться друг от друга из-за случайных или преднамеренных искажений.

Цифровая подпись, состоящая из двух целых чисел, представленных в виде слов в алфавите β, вычисляется с помощью определенного набора правил, изложенных в стандарте.

Числа p, q и a, являющиеся параметрами системы, должны быть выбраны (выработаны) по процедуре, описанной в пункте 7. числа p, q и a не являются секретными. Конкретный набор их значений может быть общим для группы пользователей. Целое число k, которое генерируется в процедуре подписи сообщения, должно быть секретным и должно быть уничтожено сразу после выработки подписи. Число k снимается с физического датчика случайных чисел или вырабатывается псевдослучайным методом с использованием секретных параметров.

5 ПРОЦЕДУРА ВЫРАБОТКИ ПОДПИСИ

Текст сообщения, представленный в виде двоичной последовательности символов, подвергается обработке по определенному алгоритму, в результате которого формируется ЭЦП для данного сообщения.

Процедура подписи сообщения включает в себя следующие этапы:

1 Вычислить $h(M)$ — значение хэш-функции h от сообщения M.

Если $\hat{h}(M) \pmod q = 0$, присвоить $h(M)$ значение $0^{255}1$.

2 Выработать целое число k, $0 < k < q$.

3 Вычислить два значения:

$r = a^k \pmod p$ и $r' = r \pmod q$.

Если $r' = 0$, перейти к этапу 2 и выработать другое значение числа k.

4 С использованием секретного ключа x пользователя (отправителя сообщения) вычислить значение

$$s = (xr' + k\hat{h}(M)) \pmod q.$$

Если $s = 0$, перейти к этапу 2, в противном случае закончить работу алгоритма.

Подписью для сообщения M является вектор $<r'>_{256} \| <s>_{256}$.

Отправитель направляет адресату цифровую последовательность символов, состоящую из двоичного представления текста сообщения и присоединительной к нему ЭЦП.

6 ПРОЦЕДУРА ПРОВЕРКИ ПОДПИСИ

Получатель должен проверить подлинность сообщения и подлинность ЭЦП, осуществляя ряд операций (вычислений).

Это возможно при наличии у получателя открытого ключа отправителя, пославшего сообщение.

Процедура проверки включает в себя следующие этапы:

1 Проверить условия:

$0<s<q$ и $0<r'<q$.

Если хотя бы одно из этих условий не выполнено, то подпись считается недействительной.

2 Вычислить $h(M_1)$ — значение хэш-функции h от полученного сообщения M_1.

Если $\hat{h}(M_1) \pmod q = 0$, присвоить $h(M_1)$ значение $0^{255}1$.

3 Вычислить значение

$v = (\hat{h}(M_1))^{q-2} \pmod q$.

4 Вычислить значения:

$z_1 = sv \pmod q$ и

$z_2 = (q-r')\, v \pmod q$.

5 Вычислить значение

$u = (a^{z_1} y^{z_2} \pmod p) \pmod q$.

6 Проверить условие: $r' = u$.

При совпадении значений r' и u получатель принимает решение о том, что полученное сообщение подписано данным отправителем и в процессе передачи не нарушена целостность сообщения, т. е. $M_1 = M$. В противном случае подпись считается недействительной.

7 ПРОЦЕДУРЫ ПОЛУЧЕНИЯ ЧИСЕЛ p, q и a

Получение простых чисел осуществляется с использованием линейного конгруэнтного датчика по модулю 2^{16} или по модулю 2^{32} $(x_n = bx_{n-1} + c)$. При этом пользователь должен задавать начальное состояние x_0 и параметр датчика c.

Заданные величины необходимо зафиксировать (запомнить) для возможности проведения проверки того, что простые числа получены по установленной процедуре.

Ниже изложены процедуры получения параметров p, q и a.

7.1 П р о ц е д у р а А

Процедура позволяет получать простые числа p длины $t \geqslant 17$ битов с простым делителем q длины $[t/]$ битов числа p-1.

Получение чисел осуществляется с использованием линейного конгруэнтного датчика $x_n = (19381\, x_{n-1} + c) \pmod{2^{16}}$.

Задаются число x_0 с условием $0<x_0<2^{16}$ и нечетное число c с условием $0<c<2^{16}$.

Процедура вычисления включает в себя следующие шаги:

1 $y_0 := x_0$

2 Вычислить последовательность чисел $(t_0, t_1, \ldots, t_s)$ по правилу:

$t_0 := t$.

Если $t_i \geqslant 17$, то $t_{i+1} = \lfloor t_i/2 \rfloor$,

Если $t_i < 17$, то $s := i$.

3 Найти наименьшее простое число p_s длины t_s битов.

4 $m := s - 1$

5 Вычислить $r_m = \lceil t_{m+1}/16 \rceil$.

6 Вычислить последовательность $(y_1, \ldots, y_{r_m})$ по рекурсивному правилу $y_{i+1} = (19381\, y_i + c) \pmod{2^{16}}$.

7 Вычислить $Y_m = \sum_{i=0}^{r_m - 1} y_i 2^{16i}$.

8 $y_0 := y_{r_m}$.

9 Вычислить $N = \lceil 2^{t_m - 1}/p_{m+1} \rceil + \lfloor (2^{t_m - 1} Y_m)/(p_{m+1} 2^{16 r_m}) \rfloor$.

Если N нечетно, то $N := N + 1$.

10 $k := 0$.

11 Вычислить $p_m = p_{m+1}(N+k) + 1$.

12 Если $p_m > 2^{t_m}$, то перейти к шагу 6.

13 Проверить условия:

$2^{p_{m+1}(N+k)} \pmod{p_m} = 1$,

$2^{(N+k)} \pmod{p_m} \neq 1$.

Если хотя бы одно из условий не выполнено, то $k := k + 2$ и перейти к шагу 11.

Если оба условия выполнены, то $m := m - 1$.

14 Если $m \geqslant 0$, то перейти к шагу 5.

Если $m < 0$, то p_0 — искомое простое число p и p_1 — искомое простое число q.

7.2 Процедура A′

Процедура позволяет получать простые числа p длины $t \geqslant 33$ битов с простым делителем q длины $\lfloor t/2 \rfloor$ битов числа p-1.

Получение числа осуществляется с использованием линейного конгруэнтного датчика $x_n = (97781173\, x_{n-1} + c) \pmod{2^{32}}$.

Задаются число x_0 с условием $0 < x_0 < 2^{32}$ и нечетное число c с условием $0 < c < 2^{32}$.

Процедура вычисления включает в себя следующие шаги:

1 $y_0 := x_0$

2 Вычислить последовательность чисел $(t_0, t_1, \ldots, t_s)$ по правилу:

$t_0 := t$.

Если $t_i \geqslant 33$, то $t_{i+1} = \lfloor t_i/2 \rfloor$, Если $t_i < 33$, то $s := i$

3 Найти наименьшее простое число p_s длины t_s битов.

4 $m := s - 1$.

5 Вычислить $r_m = \lceil t_m / 32 \rceil$.

6 Вычислить последовательность $(y_1, \ldots, y_{r_m})$ по рекурсивному правилу $y_{i+1} = (97781173\ y_i + c) \bmod (2^{32})$.

7 Вычислить $Y_m = \sum\limits_{i=0}^{r_m - 1} y_i 2^{32i}$.

8 $y_0 := y_{r_m}$.

9 Вычислить $N = \lceil 2^{t_m - 1} / p_{m+1} \rceil + \lfloor (2^{t_m - 1} Y_m) / (p_{m+1} 2^{32 r_m}) \rfloor$.

Если N нечетно, то $N := N + 1$.

10 $k := 0$.

11 Вычислить $p_m = p_{m+1}(N + k) + 1$.

12 Если $p_m > 2^{t_m}$, то перейти к шагу 6.

13 Проверить условия:

$2^{p_{m+1}(N+k)} \pmod{p_m} = 1$,

$2^{(N+k)} \pmod{p_m} \neq 1$.

Если хотя бы одно из условий не выполнено, то $k := k + 2$ и перейти к шагу 11.

Если оба условия выполнены, то $m := m - 1$.

14 Если $m \geqslant 0$, то перейти к шагу 5.

Если $m < 0$, то p_0 — искомое простое число p и p_1 — искомое простое число q.

7.3 Процедура В

Процедура позволяет получать простые числа p длины $t_p = 1021 \div 1024$ битов с делителем q длины $t_q = 255 \div 256$ битов числа p-1.

Задаются число x_0 с условием $0 < x_0 < 2^{16}$ и нечетное число c с условием $0 < c < 2^{16}$.

Процедура вычисления включает в себя следующие шаги:

1 По процедуре А получить простое число q длины t_q битов.

2 По процедуре А получить простое число Q длины 512 битов, при этом пункт 1 процедуры А не выполнять, а сохранить значение y_0, полученное в конце работы шага 1.

3 Вычислить последовательность $(y_1, \ldots, y_{64})$ по рекурсивному правилу $y_{i+1} = (19381\ y_i + c) \pmod{2^{16}}$.

4 Вычислить $Y = \sum\limits_{i=0}^{63} y_i 2^{16i}$,

5 $y_0 := y_{64}$.

6 Вычислить

$N = \lceil 2^{t_p - 1}/(qQ) \rceil + \lfloor (2^{t_p - 1}Y)/(qQ2^{1024}) \rfloor$.

Если N нечетно, то $N := N + 1$.

7 $k := 0$.

8 Вычислить $p = qQ(N+k) + 1$.

9 Если $p > 2^{t_p}$, то перейти к шагу 3.

10 Проверить условия:

$2^{qQ(N+k)} \pmod p = 1$,

$2^{q(N+k)} \pmod p \neq 1$.

Если оба условия выполнены, то р и q — искомые простые числа.

Если хотя бы одно из условий не выполнено, то $k := k + 2$ и перейти к шагу 8.

Последовательность шагов повторить до выполнения условий на шаге 10.

7.4 Процедура B′

Процедура позволяет получать простые числа p длины $t_p = 1021 \div 1024$ битов с делителем q длины $t_q = 255 \div 256$ битов числа $p - 1$.

Задаются число x_0 с условием $0 < x_0 < 2^{32}$ и нечетное число c с условием $0 < c < 2^{32}$.

Процедура вычисления включает в себя следующие шаги:

1 По процедуре A′ получить простое число q длины t_q битов.

2 По процедуре A′ получить простое число Q длины 512 битов, при этом пункт 1 процедуры A′ не выполнять, а сохранить значение y_0, полученное в конце работы шага 1.

3 Вычислить последовательность $(y_1, \ldots, y_{32})$ по рекурсивному правилу $y_{i+1} = (97781173\, y_i + c) \pmod{2^{32}}$.

4 Вычислить $Y = \sum_{i=0}^{31} y_i 2^{32i}$.

5 $y_0 := y_{32}$.

6 Вычислить

$N = \lceil 2^{t_p - 1}/(qQ) \rceil + \lfloor (2^{t_p - 1}Y)/(qQ2^{1024}) \rfloor$.

Если N нечетно, то $N := N + 1$.

7 $k := 0$.

8 Вычислить $p = qQ(N+k) + 1$.

9 Если $p > 2^{t_p}$, то перейти к шагу 3.

10 Проверить условия:

$2^{qQ(N+k)} \pmod p = 1$,

$2^{q(N+k)} \pmod p \neq 1$.

Если оба условия выполнены, то p и q — искомые простые числа.

Если хотя бы одно из условий не выполнено, то $k := k+2$ и перейти к шагу 8.

Последовательность шагов повторить до выполнения условий на шаге 10.

7.5 Процедура C

Процедура позволяет получить число a при заданных p и q.

1 Произвольно выбрать число d, $1 < d < p-1$.

2 Вычислить $f = d^{\frac{p-1}{q}} \pmod{p}$.

3 Если $f=1$, то перейти к шагу 1.

Если $f \neq 1$, то $a := f$.

Конец работы алгоритма.

Проверочные примеры для вышеизложенных процедур получения чисел p, q и a, выработки и проверки подписи приведены в приложении А.

Приложение А

(справочное)

ПРОВЕРОЧНЫЕ ПРИМЕРЫ

Значения параметров x_0, c, d, x, y, k, указанные в приложении, рекомендуется использовать только в проверочных примерах для настоящего стандарта.

А.1 Представление чисел и векторов

Длины чисел и векторов, а также элементы последовательности t записывают в десятичной системе счисления.

Последовательности двоичных символов записывают как строки шестнадцатеричных цифр, в которых каждая цифра соответствует четырем знакам ее двоичного представления.

А.2 Примеры к процедурам получения чисел p, q и числа a для реализации ЭЦП

А.2.1 *Процедура А*

Необходимо получить простое число p длины 512 битов с простым делителем q длины 256 битов числа p—1.

Задают числа x_0=5EC9 и c=7341.

Вычисляют последовательность t=(512, 256, 128, 64, 32, 16).

Тогда в процессе выполнения процедуры будет получена последовательность простых чисел:

$t_5=16$,	$p_5=$	8003			
$t_4=32$,	$p_4=$	AD4B0FAB			
$t_3=64$,	$p_3=$	B25D28A7	1A62D775		
$t_2=128$,	$p_2=$	9C992766	8E6E4908	964A9AE1	3773AE75
$t_1=256$,	$p_1=$	98915E7E	C8265EDF	CDA31E88	F24809DD
		B064BDC7	285DD50D	7289F0AC	6F49DD2D
$t_0=512$,	$p_0=$	EE8172AE	8996608F	B69359B8	9EB82A69
		854510E2	977A4D63	BC97322C	E5DC3386
		EA0A12B3	43E9190F	23177539	84583978
		6BB0C345	D165976E	F2195EC9	B1C379E3

p_1 и p_0 — искомые числа q и p соответственно.

А.2.2 *Процедура А'*

Необходимо получить простое число p длины 512 битов с простым делителем q длины 256 битов числа p—1.

Задают числа x_0=3DFC46F1 и c=D.

Вычисляют последовательность t=(512, 256, 128, 64, 32).

Тогда в процессе выполнения процедур будет получена последовательность простых чисел:

$t_4=32$,	$p_4=$	8000000B			
$t_3=64$,	$p_3=$	9AAA6EBE	4AA58337		
$t_2=128$,	$p_2=$	C67CE4AF	720F7BBA	B5FEBF37	B9E74807
$t_1=256$,	$p_1=$	931A58FB	6F0DCDF2	FE7549BC	3F19F472
		4B56898F	7F921A07	6601EDB1	8C93DC75
$t_0=512$,	$p_0=$	8B08EB13	5AF966AA	B39DF294	538580C7
		DA26765D	6D38D30C	F1C06AAE	0D1228C3
		316A0E29	198460FA	D2B19DC3	81C15C88
		8C6DFD0F	C2C565AB	B0BF1FAF	F9518F85

p_1 и p_0 — искомые числа q и p соответственно.

А.2.3 *Процедура В*

Необходимо получить простое число p длины 1024 битов с простым делителем q длины 256 битов числа p—1.

Задают начальные значения x_0=A565 и c=538B.

С помощью процедуры А получают простое число q длиной l=256 битов:

BCC02CA0	CE4F0753	EC16105E	E5D530AA
00D39F31	71842AB2	C334A26B	5F576E0F

Затем вновь с помощью процедуры А получают простое число Q длиной l=512 битов:

CCEF6F73	87B6417E	C67532A1	86EC619C
A4DB132F	CA02621A	DE216F1D	F6F8114C
DB3D9209	7D978C6F	583C3301	4174AA1C
1AFCCEB2	843B1D35	0D2E5D16	855A7477

И, наконец, получают простое число p длиной l=1024 битов:

AB8F3793	8356529E	871514C1	F48C5CBC
E77B2F4F	C9A2673A	C2C1653D	A8984090
C0AC7377	5159A26B	EF59909D	4C984663
1270E166	53A62346	68F2A52A	01A39B92
1490E694	C0F104B5	8D2E1497	0FCCB478
F98D01E9	75A1028B	9536D912	DE5236D2
DD2FC396	B7715359	4D417878	0E5F16F7
18471E21	11C8CE64	A7D7E196	FA57142D

А.2.4 *Процедура В'*

Необходимо получить простое число p длины 1024 битов с простым делителем q длины 256 битов числа p—1.

Задают начальные значения x_0=3DFC46F1 и c=D.

С помощью процедуры А получают простое число q длиной l=256 битов:

931A58FB	6F0DCDF2	FE7549BC	3F19F472
4B56898F	7F921A07	6601EDB1	8C93DC75

Затем вновь с помощью процедуры А получают простое число Q длиной l=512 битов:

BB124D6C	255D373F	FA7D5DF5	5CE0DB44
96397506	6F8980B1	C7CB68DF	6C6E8D27
12D34BF3	3B536899	C7150C4D	F82FC171
D9529BC8	C9653929	D6682CF5	FBBA1B3D

И, наконец, получают простое число р длиной l=1024 битов:

E2C4191C	4B5F222F	9AC27325	62F6D9B4
F18E7FB6	7A290EA1	E03D750F	0B980675
5FC730D9	75BF3FAA	606D05C2	18B35A6C
3706919A	AB92E0C5	8B1DE453	1C8FA8E7
AF43C2BF	F016251E	21B28708	97F6A27A
C4450BCA	235A5B74	8AD386E4	A0E4DFCB
09152435	ABCFE48B	D0B126A8	122C7382
F285A986	4615C66D	ECDDF6AF	D355DFB7

А.2.5 *Процедура С*

Пусть заданы числа р и q, полученные в А.2.1 по процедуре А:

p=	EE8172AE	8996608F	B69359B8	9EB82A69
	854510E2	977A4D63	BC97322C	E5DC3386
	EA0A12B3	43E9190F	23177539	84583978
	6BB0C345	D165976E	F2195EC9	B1C379E3

q=	98915E7E	C8265EDF	CDA31E88	F24809DD
	B064BDC7	285DD50D	7289F0AC	6F49DD2D

Выбирают число d=2.

Вычисляют

$$f = d^{\frac{p-1}{q}} \pmod p =$$

9E960315	00C8774A	869582D4	AFDE2127
AFAD2538	B4B6270A	6F7C8837	B50D50F2
06755984	A49E5093	04D648BE	2AB5AAB1
8EBE2CD4	6AC3D849	5B1412AA6	CE23E21C

Так как $f \neq 1$, то f — искомое число a:=f

А.3 Примеры процедур выработки и проверки ЭЦП на базе асимметричного криптографического алгоритма

Пусть по процедуре А с начальными условиями x_0=5EC9 и c=7341 выработаны числа p, q и a:

p=	EE8172AE	8996608F	B69359B8	9EB82A69
	854510E2	977A4D63	BC97322C	E5DC3386
	EA0A12B3	43E9190F	23177539	84583978
	6BB0C345	D165976E	F2195EC9	B1C379E3

q=	98915E7E	C8265EDF	CDA31E88	F24809DD
	B064BDC7	285DD50D	7289F0AC	6F49DD2D

a=	9E960315	00C8774A	869582D4	AFDE2127
	AFAD2538	B4B6270A	6F7C8837	B50D50F2
	06755984	A49E5093	04D648BE	2AB5AAB1
	8EBE2CD4	6AC3D849	5B142AA6	CE23E21C

А.3.1 *Процедура подписи сообщения*

Пусть x=	30363145	38303830	34363045	42353244
	35324234	31413237	38324331	38443046

— секретный ключ, M — подписываемое сообщение, причем значение хэш-функции h от сообщения M есть

h(M)=m=	35344541	32454236	44313445	34373139
	43363345	37414342	34454136	31454230

Пусть целое число

k=	90F3A564	439242F5	186EBB22	4C8E2238
	11B7105C	64E4F539	0807E636	2DF4C72A

Тогда

$r=a^k \pmod p$ =	47681C97	4373B065	3C6CA965	C8F86127
	D07A7E02	E311846E	97A8C126	3F8A76AF
	FF0AD188	02643B5C	6C998775	0C6B0458
	98E4AD8C	FC689817	76BA8216	3ADBC988

$r'=r \pmod q$ =	3E5F895E	276D81D2	D52C0763	270A4581
	57B784C5	7ABDBD80	7BC44FD4	3A32AC06

$s=xr'+km \pmod q$ =	3F0DD5D4	400D47C0	8E4CE505	FF7434B6
	DBF72959	2E37C748	56DAB851	15A60955

Таким образом, цифровая подпись для сообщения M есть

$\langle r' \rangle_{256} \Vert \langle s \rangle_{256}$ =	3E5F895E	276D81D2	D52C0763	270A4581
	57B784C5	7ABDBD80	7BC44FD4	3A32AC06
	3F0DD5D4	400D47C0	8E4CE505	FF7434B6
	DBF72959	2E37C748	56DAB851	15A60955

А.3.2 Процедура проверки подписи

Пусть дано сообщение M_1 (в данном случае $M_1 = M$), его цифровая подпись

$\langle r' \rangle_{256} \| \langle s \rangle_{256} =$	3E5F895E	276D81D2	D52C0763	270A4581
	57B784C5	7ABDBD80	7BC44FD4	3A32AC06
	3F0DD5D4	400D47C0	8E4CE505	FF7434B6
	DBF72959	2E37C748	56DAB851	15A60955

и открытый ключ подписавшего сообщение

$y =$	EE1902A4	0692D273	EDC1B5AD	C55F9112
	8E35F9D1	65FA9901	CAF00D27	018BA6DF
	324519C1	1A6E2725	26589CD6	E6A2EDDA
	AFE1C308	1259BE9F	CEE667A2	701F4352

Замечание

Данный открытый ключ y соответствует секретному ключу x, использованному в примере подписи сообщения M
$y = a^x \pmod{p}$.

Пусть

$m =$	35344541	32454236	44313445	34373139
	43363345	37414342	34454136	31454230

— значение хэш-функции h для сообщения M_1.

Условия $0 < r' < q$ и $0 < s < q$ выполняются.

Вычисляют

$v = m^{q-2} \pmod{q} =$	72515E01	DDFA6507	E3682C01	CD285CBF
	89E462EE	E37B3865	918B6730	DEA77050

$z_1 = sv \pmod{q} =$	776DC3C6	4E83B73B	02B78826	6873EAFF
	B87DAED5	8686009B	5D387CC4	EAF5B744

$z_2 = (q - r')\ v \pmod{q} =$	18B04C46	C1D9E875	571FDA9E	95354DDE
	3AFD0A8D	FCADB67C	505C7F03	A5185DFD

$u = (a^{z_1} y^{z_2} \pmod{p}) \pmod{q} =$	3E5F895E	276D81D2	D52C0763	270A4581
	57B784C5	7ABDBD80	7BC44FD4	3A32AC06

Таким образом:

$r' =$	3E5F895E	276D81D2	D52C0763	270A4581
	57B784C5	7ABDBD80	7BC44FD4	3A32AC06

$u =$	3E5F895E	276D81D2	D52C0763	270A4581
	57B784C5	7ABDBD80	7BC44FD4	3A32AC06

Условие $r' = u$ выполнено. Это означает, что подпись подлинная.

УДК 681.3.06:006.354 П85 ОКСТУ 5002

Ключевые слова: информационная технология, криптографическая защита информации, электронная цифровая подпись, ассимметричный криптографический алгоритм, системы обработки информации, защита сообщений, подтверждение подписи, хэш-функция, функция хэширования