

## 1. Цель работы

Реализовать алгоритм электронной цифровой подписи DSA.

## 2. Алгоритм

### 2.1. Общее описание алгоритма

Алгоритм DSA основывается на двух вычислительных задачах, связанных с дискретным логарифмированием. Одной задачей является сложность вычисления логарифма в  $Z_p^*$ , другая задача - сложность логарифмирования в циклической подгруппе порядка  $q$ . Алгоритм является частным случаем цифровой подписи Эль-Гамала (ElGamal) и был представлен как стандарт FIPS PUB 186-94 (DSS).

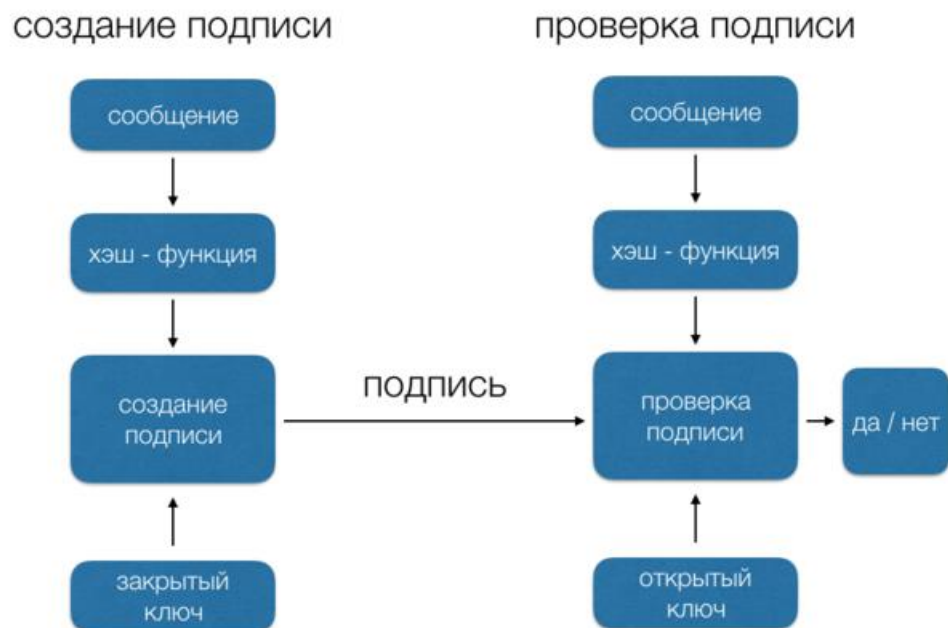


Рис.1 – схема алгоритма

### 2.2. Генерация ключей DSA

- Выбирается простое число  $q$ , такое, что  $2^{159} < q < 2^{160}$ .
- Выбирается  $t$ , т.ч.  $0 \leq t \leq 8$  и выбирается простое число  $p$  так, что  $2^{511 + 64t} < p < 2^{512 + 64t}$ , причем  $q$  должно делить  $(p-1)$ .
- Находится производящий элемент  $a$  для циклической группы  $Z_p^*$  порядка  $q$  (для этого выбирается  $g \in Z_p^*$  и вычисляется  $\alpha = g^{(p-1)/q} \bmod p$ , Если  $\alpha \neq 1$ , то пр. элемент найден).
- Выбирается случайное целое  $a$ , т. ч.  $1 \leq a \leq q-1$ .
- Вычисляется  $y = a^a \bmod p$ .

Секретным ключом является  $a$ , открытым ключом -  $(p, q, \alpha, y)$

### 2.3. Подпись сообщения

Имеется сообщение  $m$ . Подпись сообщения секретным ключом выглядит следующим образом.

- Выбирается случайное секретное число  $k$ ,  $0 < k < q$  (разовый секретный ключ).
- Вычисляется  $r = (a^k \bmod p) \bmod q$ .
- Вычисляется  $k^{-1} \bmod q$ .
- Вычисляется  $s = k^{-1}\{h(m) + ar\} \bmod q$ , где  $h(m)$  - значение хэш-функции от сообщения  $m$ .

Подписью для сообщения  $m$  является пара  $(r, s)$ .

### 2.4. Проверка подписи

Имеется открытый ключ  $(p, q, \alpha, y)$ , сообщение  $m$ , подпись сообщения  $(r, s)$ .

- Проверить, что  $0 < r < q$  и  $0 < s < q$ . Если это не так, отвергнуть подпись.
- Вычислить  $w = s^{-1} \bmod q$  и  $h(m)$ .
- Вычислить  $u_1 = wh(m) \bmod q$  и  $u_2 = rw \bmod q$ .
- Вычислить  $v = (a^{u_1} y^{u_2} \bmod p) \bmod q$ .
- Подпись верна, только если  $v = r$ .

### 2.5. Доказательство корректности подписи

Если  $(r, s)$  является корректной подписью для сообщения  $m$ , тогда должно выполняться  $h(m) = -ar + ks \pmod q$ . Умножим обе части равенства на  $w$  и получим, что  $wh(m) + arw = k \pmod q$ . А это есть  $u_1 + au_2 = k \pmod q$ . Т.е. получаем, что  $(a^{u_1} a^{au_2} \bmod p) \bmod q = (a^k \bmod p) \bmod q$ . Или  $(a^{u_1} y^{u_2} \bmod p) \bmod q = (a^k \bmod p) \bmod q$ . Это есть  $v = r$ , что и требовалось доказать.

## 3. Пример:

Результат работы программы:

### 3.1. Пример 1

Генерация простых чисел и ключей

Простое число  $P$

$P = 006b\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ 3365$

160-ти битовое простое число  $Q$

$Q = 0000\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ ffff\ fe1b$

Случайное число  $H$ , приведенное по модулю  $P$

$H = 0005\ ff66\ ffff\ 4fff\ ffbb\ 624f\ f4fc\ fdf6\ 64ff\ f26f\ fdf3$

Секретный ключ, приведенный по модулю Р  
0000 ffff ffff ffff 664f f4ff fff6 f4ff f2ff ffd8

вычисление  $G = H_{\text{nach}}^{[(P-1)/Q]} \pmod{P}$   
 $G = 0059\ c1b3\ f0ca\ c78e\ bb04\ 588e\ 0b47\ c04f\ b9cc\ 276d\ 5d32$

проверка модуля P:  $G^Q \equiv 1 \pmod{P}$   
 $G^Q = 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001$

Открытый ключ  $Y = G^x \pmod{P}$   
 $Y = 0066\ f23a\ 628b\ 9fb2\ 2a46\ be6d\ d3a9\ ea34\ 91bf\ 48c7\ 024b$

Введите сообщение, которое необходимо подписать:  
cryptography

---Выработка подписи---

[illegible]

Значение хэш-функции  
H = 0000 7e88 16e1 b779 fa1d c673 5e3a 05c6 9f08 3c95 9b3d

-Подпись сообщения-  
Компонента R = 0000 1d82 3690 104e f26b aa1e e308 c446 9fb1 a0b5 2a77  
Компонента S = 0000 986a 826b 948c 5f5c 8c28 1ce1 9f16 034b 3b2a f072

---Проверка подписи---

Открытый ключ  
Y = 0066 f23a 628b 9fb2 2a46 be6d d3a9 ea34 91bf 48c7 024b

# Подписываемое сообщение

## cryptography

Вычисление компонент A,B,V  
A=0000 da31 d4f1 f792 8d22 45df f701 f4b3 72d9 514d 1fd8

B=0000 6687 3ec0 db3c fd11 7067 21ad c3bb c65d 761d fa37  
V=0000 1d82 3690 104e f26b aa1e e308 c446 9fb1 a0b5 2a77

Подпись верна так как вычисленное значение  $V$  равно подписанному значению  $R$

### 3.2. Пример 2

## Генерация простых чисел и ключей

Простое число  $P$ 

P = 006b ffff ffff ffff ffff ffff ffff ffff ffff ffff 3365

160-ти битовое простое число  $Q$

**Q =** 0000 ffff ffff ffff ffff ffff ffff ffff ffff ffff fe1b

Случайное число  $N$ , приведенное по модулю  $P$

H = 0005 ff66 ffff 4fff ffbb 624f f4fc fdf6 64ff f26f fdf3

Секретный ключ, приведенный по модулю Р

```
0000 ffff ffff ffff ffff 664f f4ff fff6 f4ff f2ff ffd8
```

вычисление  $G = Hn^{ach^{[(P-1)/Q]}} \pmod{P}$

G = 0059 c1b3 f0ca c78e bb04 588e 0b47 c04f b9cc 276d 5d32

проверка модуля P:  $G^Q = 1 \pmod P$

$$G^A Q = 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001$$

Открытый ключ  $Y=G^x \pmod P$

Y = 0066 f23a 628b 9fb2 2a46 be6d d3a9 ea34 91bf 48c7 024b

Введите сообщение, которое необходимо подписать:

# cryptography

---Выработка подписи---

## Подписываемое сообщение

## cryptography

Сообщение, переведенное в коды ASCII

63 72 79 70 74 6f 67 72 61 70 68 79

Количество символов  $n = c$

Количество блоков по 64 символа  $n1 = 0$

Количество оставшихся символов  $n_2 = c$

63727970746f67726170687900

[illegible]

## Значение хэш-функции

H = 0000 7e88 16e1 b779 fa1d c673 5e3a 05c6 9f08 3c95 9b3d

-Подпись сообщения-

Компонента R = 0000 1d82 3690 104e f26b aa1e e308 c446 9fb1 a0b5 2a77

Компонента S = 0000 986a 826b 948c 5f5c 8c28 1ce1 9f16 034b 3b2a f072

---Проверка подписи---

Открытый ключ

Y = 6754 h43a 908b 6ab2 2a46 be6d d3a9 ea34 91bf 48c7 743a

Подписываемое сообщение

cryptography

Вычисление компонент A,B,V

A=0000 hf54 d4a2 a255 6543 45df f231 3fa2 652b 514d 1ff5

B=0000 9213 3ba0 dbec fd11 2345 21ad c3bb c43d 125d fa30

V=0000 2a46 91bf 645b a345 aa3c b767 c55c 9fb2 a0b5 2c01

Подпись не верна так как вычисленное значение V не равно подписанному значению R

## Вывод

В данном алгоритме предлагается использовать простое  $p$  размером от 512 до 1024 бит. Размер в 512 бит обеспечивает минимальную защищенность. Рекомендуемый размер - не менее 768 бит. Согласно FIPS 186, алгоритм не допускает простых чисел больше 1024 бит.

Совершенно не обязательно существование уникальных  $p$  и  $q$  для каждого пользователя алгоритма. FIPS допускает использование  $p$ ,  $q$  и  $a$  в качестве системных параметров для группы пользователей. Однако для повышения безопасности работы лучше использовать уникальные значения.

## 4. Список источников

1. [А.Л. Чмора "Современная прикладная криптография"](#)
2. [Черчхаус. Коды и шифры](#)