***Схема цифровой подписи RSA***

##### **Генерация ключей**

Генерация ключей в схеме цифровой подписи RSА точно такая же, как и генерация ключей в криптографической системе RSА.

Выбираем два простых числа **p** и **q** и вычисляем **n = p \* q**.

p = 823 и q = 953 n = 784319

Вычисляем Описание: \varphi  (n) = (p - 1) (q - 1). Значение Описание: \varphi  (n)- 782544.

Затем выбираем **e** = 313, для общедоступного ключа и вычисляем **d** для частного ключа, такое, что Описание: e x d = 1 mod \varphi \  (n)= d = 160009.

В схеме цифровой подписи RSA. d **является частным;**

e **и** n **- общедоступными**.

**Подписание**. На основе сообщения создаем подпись, используя частный (секретный) ключ, S = Md mod n, и передаем сообщение и подпись.

**Проверка**. Получатель принимает M = 19070 ->

S = (19070160009) mod 784319 = 210625313 mod 784319

Далее применяет общедоступный ключ к подписи, чтобы создать копию сообщения

М' = Se mod n.

Описание: М' = 210625313 \ mod \ 784319 = 19070 \ mod \ 784319 \ ? \ М 
\\
 М' \equiv \ mod \ n

После чего сравнивает значение М' со значением М. Т.к. два значения совпадают, принимаем сообщение.

#### Схема цифровой подписи Эль-Гамаля

##### Генерация ключей

Выберем достаточно большое простое число **p**, чтобы в поле **Z p\*** проблема дискретного логарифма была достаточно трудной. Пусть **e1** - простой элемент в **Z p\*.** Выбираем свой секретный ключ **d**, чтобы он был меньше, чем (**p – 1**). p = 3119, e1 = 2, d = 127

Вычисляем **e2 = e1d =** e2 = 2127 mod 3119 = 1702.

Открытый ключ - кортеж **(e1, e2, p)** ; секретный ключ - **d**.

**Подписывающийся** может подписать дайджест сообщения, направленный к любому объекту.

1. Выбираем секретное случайное число **r = 307**.
2. Вычисляем первую подпись **S1 = er mod p = 2307mod 3119 = 2083**.
3. Вычисляем вторую подпись **S2 = (M - d \* S1) \* r-1 = (320 - 127 \* 2083) \* 307-1 = 2105 mod 3118**, где **r** - мультипликативная инверсия **r** по модулю (**p – 1**).
4. Передаем **М = 320, S1 = 2083** и **S2** получателю.

**Проверка**. Получатель принимает **М, S1** и **S2** и может проверить их следующим образом.

1. Проверяет 0 < **S1** < p.
2. Проверяет 0 < **S2** < p - 1.
3. Вычисляет **V1 = e1M mod p =** **2320mod 3119** **= 3006 mod 3119**
4. Вычисляет **V2 = e2S1 \* e2S2 mod p** **= 17022083 \* 20832105 = 3006 mod 3119**
5. Если **V1** является конгруэнтным **V2**, сообщение принято; иначе оно будет отклонено.

#### Схема цифровой подписи Шнорра

##### **Генерация ключей**

Перед подписанием сообщения нужно сгенерировать ключи и объявить общедоступные ключи.

1. Выбираем простое число **p** = 2267 (p = 22 \* q + 1), которое обычно равно по длине 1024 битам.
2. Выбираем другое простое число ***q***  **= 103**, которое имеет тот же размер, что и дайджест, созданный функцией криптографического хэширования (160 битов). Простое число **q** должно делиться на **(p - 1)**. Т.е., **(p - 1) = 0 mod q**.
3. Выбираем **e1, q** -тый корень которого был бы равен **1 mod p**. Чтобы сделать это, выбираем примитивный элемент в **Zp, e0** = 2 и вычисляем **e1 = e0(p-1)/q mod p** = 222 mod 2267 = 354 ((p - 1) / q = 22).
4. Выбираем целое число, **d =** 30, как свой секретный ключ.
5. Вычисляем **e2 = e1d mod p** = 35430 mod 2267 = 1206.
6. Общедоступный ключ - **(e1, e2, p, q)**, секретный ключ - **(d)**.

**Подписание**

1. Выбираем случайное число **r =** 11, r должен иметь значение между 1 и q.
2. Вычисляем первую подпись **S1 = h(M | e1r mod p) =** 200 . Сообщение присоединяется (конкатенируется) спереди к значению **e1r mod p =** 35411 = 630 mod 2267, затем применяется хэш-функция, чтобы создать дайджест, функция получается из последовательного соединения М. и e1r mod p.
3. Вычисляем вторую подпись **S2 = r + d \* S1 mod q =** 11 + 1026\* 200 mod 103 = 11 + 24 = 35, по модулю q.
4. Передаем М =1000, S1 = 200 и S2 = 35.

**Верификация (проверка) сообщения**. Получатель принимает М, S1 и S2.

1. Вычисляет **V = h (М | e1S2 e2-S1 mod p) = 35**.
2. Если S2 конгруэнтно V по модулю p, сообщение принято; иначе оно отклоняется.