Алгоритм создания ключей

1) **Выбираются 2 различных случайных простых числа** p **и** q заданного размера (например, 1024 бита каждое)

2) **Вычисляется их произведение** $N = p \cdot q$, которое называется *модулем*;

Шифрование RSA - это модульное экспонирование сообщения с экспонентой e и модулем N, который обычно является произведением двух простых чисел: $N = p \cdot q$.

> ☝ Очень важно, чтобы было 2 различных простых числа p и q. Потому что, если получится факторизовать N, то можно подсчитать приватный ключ из публичного.

3) **выбирается целое число** e $(1 < e < \phi(n))$, взаимно простое со значением функции $\phi(n)$

Вместе экспонента и модуль образуют "открытый ключ" RSA (N, e). Наиболее распространенное значение для e - 0x10001 или 65537.

Процесс шифрования:

```
p=17
q=23
e=65537
msg = 12
```

```
pow(msg,e,N)
```

- число e называется *открытой экспонентой* (англ. *public exponent*);
- обычно в качестве e берут простые числа, содержащие небольшое количество единичных бит в двоичной записи, например, простые из чисел Ферма: 17, 257 или 65537, так как в этом случае время, необходимое для шифрования с использованием быстрого возведения в степень, будет меньше;
- слишком малые значения e, например 3, потенциально могут ослабить безопасность схемы RSA.

RSA полагается на сложность факторизации модуля N. Если можно найти простые числа, то можно вычислить постоянную функцию Эйлера для N и таким образом расшифровать шифротекст.

```
def phi(p,q):
  return (p-1)*(q-1)
```

Закрытый ключ d используется для расшифровки шифротекстов, созданных с помощью соответствующего открытого ключа (он также используется для "подписи").

4) **Вычисляется число** d — мультипликативно обратное к числу e по модулю $\phi(n)$, то есть число, удовлетворяющее сравнению:

$$$$

число d называется *секретной экспонентой*; обычно оно вычисляется при помощи расширенного алгоритма Евклида

```
d = pow(e, -1, phi(p,q))
```

Процесс расшифрования:

$$m = D(c) = c^d \ mod \ N$$