*RSA (аббревиатура от фамилий Rivest, Shamir и Adleman)*

Первой и наиболее известной во всем мире конкретной системой ЭЦП стала система RSA, математическая схема которой была разработана в 1977 г. в Массачусетском технологическом институте США. Надежность алгоритма основывается на трудности факторизации больших чисел.

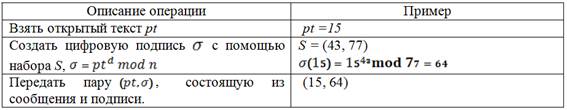
*Алгоритм создания открытого и секретного ключей RSA*



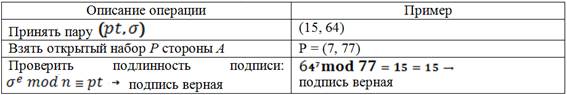
*Алгоритм цифровой подписи сообщений*

Предположим, что стороне A нужно отправить стороне B сообщение pt = 15, подтвержденное цифровой подписью.

*Алгоритм отправителя*



*Алгоритм получателя*

**

*Недостатки алгоритма цифровой подписи RSA*

- При вычислении модуля n, ключей e и d для системы цифровой подписи RSA необходимо проверять большое количество дополнительных условий, что сделать практически трудно. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение.

- Для обеспечения криптостойкости цифровой подписи RSA по отношению к попыткам фальсификации на уровне, например, национального стандарта США на шифрование информации (алгоритм DES), т.е. 1018, необходимо использовать при вычислениях n, d и e целые числа, не менее 2512 каждое, что требует больших вычислительных затрат, превышающих на 20-30% вычислительные затраты других алгоритмов цифровой подписи при сохранении того же уровня криптостойкости.

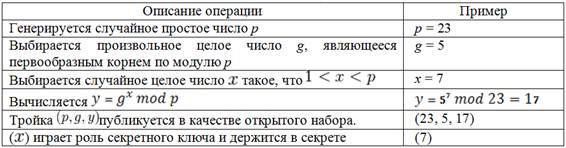
- Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позволяет злоумышленнику без знания секретного ключа d сформировать подписи под теми документами, у которых результат хэширования можно вычислить как произведение результатов хэширования уже подписанных документов.

*Elgamal (схема Эль-Гамаля)*

Более надежный и удобный для реализации на персональных компьютерах ЭЦП алгоритм был разработан в 1984 г. американцем арабского происхождения Тахером Эль Гамалем и получил название ElGamalSignatureAlgorithm (EGSA).

Идея EGSA основана на том, что для обоснования практической невозможности фальсификации ЭЦП может быть использована более сложная вычислительная задача, чем разложение на множители большого целого числа – задача дискретного логарифмирования. Кроме того, Эль Гамалю удалось избежать явной слабости алгоритма ЭЦП RSA, связанной с возможностью подделки ЭЦП под некоторыми сообщениями без определения секретного ключа.

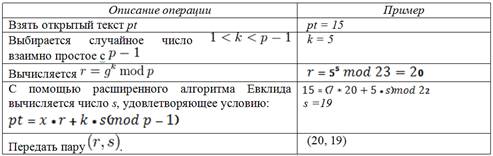
*Алгоритм создания открытого и секретного ключей Elgamal*



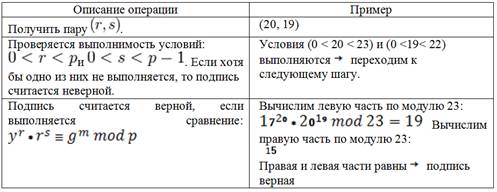
*Алгоритм цифровой подписи сообщений*

Предположим, что стороне A нужно отправить стороне B сообщение pt = 15, подтвержденное цифровой подписью.

*Алгоритм отправителя*



*Алгоритм получателя*

**

*Схема цифровой подписи Эль Гамаля имеет ряд преимуществ по сравнению со схемой цифровой подписи RSA:*

1) При заданном уровне стойкости алгоритма цифровой подписи целые числа, участвующие в вычислениях, имеют запись на 25% короче, что уменьшает сложность вычислений почти в два раза.

2) При выборе модуля *p* достаточно проверить, что это число является простым и что у числа (Р-1) имеется большой простой множитель.

3) Процедура формирования подписи по схеме Эль Гамаля не позволяет вычислять цифровые подписи под новыми сообщениями без знания секретного ключа (как в RSA).

Однако алгоритм цифровой подписи Эль Гамаля имеет и некоторые *недостатки по сравнению со схемой подписи RSA*. В частности, длина цифровой подписи получается в 1,5 раза больше, что, в свою очередь, увеличивает время ее вычисления.

Из-за низкой скорости шифрования, сообщения обычно шифруют с помощью более производительных симметричных алгоритмов со случайным *сеансовым ключом* (например, [AES](https://ru.wikipedia.org/wiki/Advanced_Encryption_Standard), [IDEA](https://ru.wikipedia.org/wiki/IDEA), [Serpent](https://ru.wikipedia.org/wiki/Serpent" \o "Serpent), [Twofish](https://ru.wikipedia.org/wiki/Twofish" \o "Twofish)), а с помощью RSA шифруют лишь этот ключ, таким образом реализуется [гибридная криптосистема](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%B1%D1%80%D0%B8%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0). Такой механизм имеет потенциальные уязвимости ввиду необходимости использовать [криптографически стойкий генератор псевдослучайных чисел](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8_%D1%81%D1%82%D0%BE%D0%B9%D0%BA%D0%B8%D0%B9_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB" \o "Криптографически стойкий генератор псевдослучайных чисел) для формирования случайного сеансового ключа симметричного шифрования.