

# RSA ENCRYPTION: BEHIND THE SCENES

PETRO KOLOSOV

ABSTRACT. Simple explanation of the main idea behind the RSA encryption.

## CONTENTS

|                                 |   |
|---------------------------------|---|
| 1. RSA Encryption: General Idea | 1 |
| 2. Mathematics behind RSA       | 3 |
| References                      | 5 |

## 1. RSA ENCRYPTION: GENERAL IDEA

The RSA algorithm is named after Ron **Rivest**, Adi **Shamir** and Len **Adleman** who invented it in 1977 [1]. The basic technique was first discovered in 1973 by Clifford Cocks [2] of CESG (part of the British GCHQ) but this was a secret until 1997. The patent taken out by RSA Labs has expired.

Historically, the process of encryption is considered to be symmetric one.

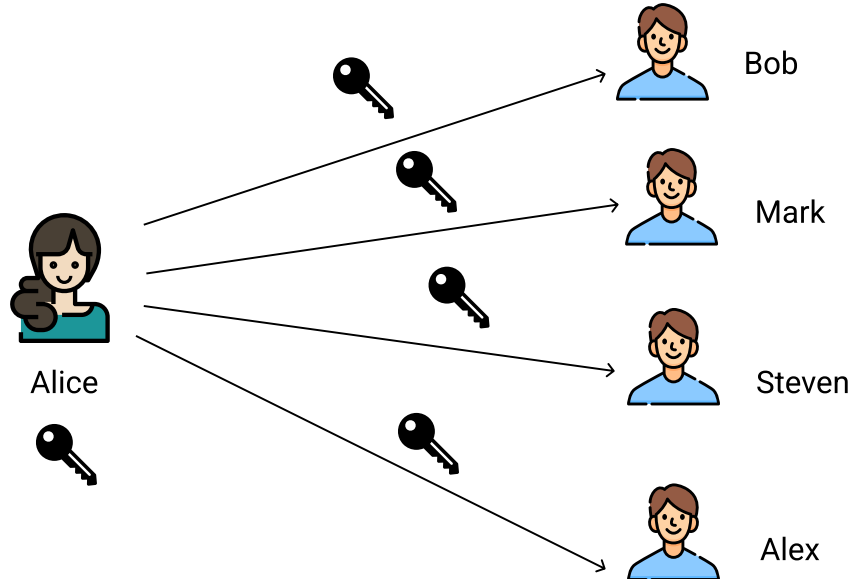
**Symmetric encryption** – is a type of encryption where only one secret key is used to both encrypt and decrypt information.

It means that prior the communication the sides must conclude and share the secret key to be used in both encryption and decryption. Such approach is highly cost since it requires to share the defined secret keys between each actor.

---

*Date:* March 13, 2022.

*Key words and phrases.* Algorithms, Computer science, Cryptography, RSA, Number Theory.



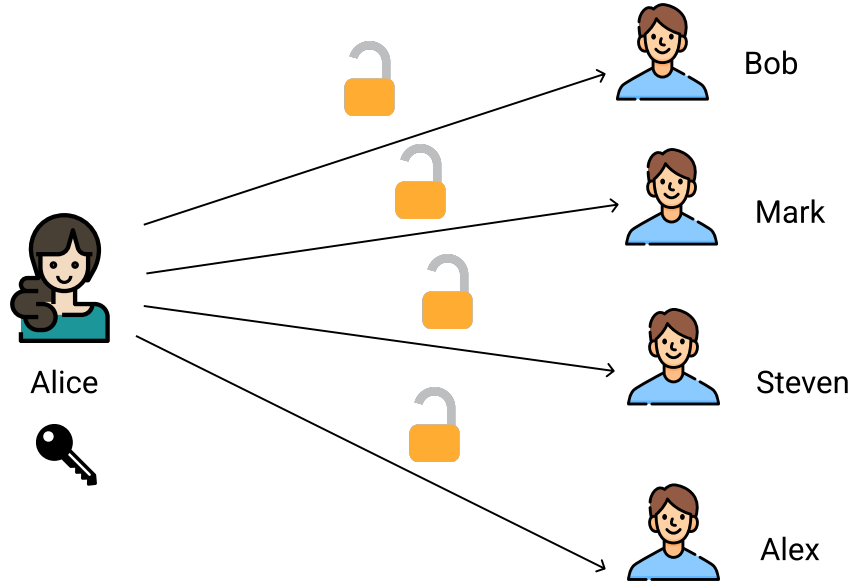
**Figure 1.** Symmetric encryption real life example.

The problem here is that Alice and Bob must exchange the keys securely, for instance by means of DH key exchange.

Much more simpler is to think about secured communication channel that in terms of asymmetric encryption.

**Asymmetric encryption** – is an encryption such that a message is encrypted using public key and decrypted using private key.

The real life example would be if Alice shares with all the actors not the secret key, but opened lock. Still Alice keeps the key with herself, but now she doesn't worry that anyone would lose it so that key exchange must be repeated.



**Figure 2.** Asymmetric encryption real life example.

So that the Bob, Mark, Steven and Alex have received an opened lock from Alice. Now anyone of them is able to write secret message to the Alice simply putting it to the chest closing by the lock received from Alice so that only Alice can open it. However, such a simple idea requires complex number theory approach. A concept of opened lock may be interpreted in terms of one-way functions.

**One-way function** – is a function that is easy to compute on every input, but hard to invert given the image of a random input.

## 2. MATHEMATICS BEHIND RSA

For instance, the function

$$f(m) = m^e \bmod N = C, \quad (e, N) \text{ are public constants, } m \text{ is secret}$$

is one-awy function because it is easy to compute  $C$  given  $m$ , but it is hard to compute  $m$  given  $C$ . The constants  $(e, N)$  may be interpreted as an Alice's opened lock, whereas  $m$  is a secret message from Bob. Note that constants

- $e$  is stands for encryption, public key
- $d$  is stands for decryption, private key

Now only one problem remains for the Alice – is to define a pair of the keys  $(e, d)$ . Alice knows that Bob encrypts his message  $m$  as follows

$$m^e \bmod N = C$$

To decrypt the message  $C$  Alice must fetch a constant  $d$  such that reverts the exponentiation of the secret message  $m$

$$\begin{aligned} C^d &= m \bmod N \\ m^{ed} \bmod N &= m \bmod N \end{aligned}$$

Firstly, Alice defines a public constant  $N$  as a product of two large prime numbers  $P, Q$

$$N = P \cdot Q$$

so that it is hard to compute a factorization.

But how to fetch the secret constant  $d$  to decrypt? The Euler's totient theorem helps. Given a number  $N$  and its prime factorization  $p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$ , the Euler's totient function  $\phi(N)$  is defined as

$$\phi(N) = (p_1^{e_1} - p_1^{e_1-1}) \cdot (p_2^{e_2} - p_2^{e_2-1}) \cdots (p_k^{e_k} - p_k^{e_k-1})$$

In particular, for the positive number  $N$  such that its factorization is  $p_1 \cdot p_2$ , the  $\phi(N)$  is

$$\phi(N) = (p_1 - 1) \cdot (p_2 - 1)$$

Euler's theorem relates the modular division and exponent as follows. Given number  $m$  then

$$m^{\phi(N)} = 1 \bmod N$$

It means that remainder of division  $m^{\phi(N)}$  by  $N$  is always 1. By the equality  $1^K = 1$

$$m^{K \cdot \phi(N)} = 1 \bmod N$$

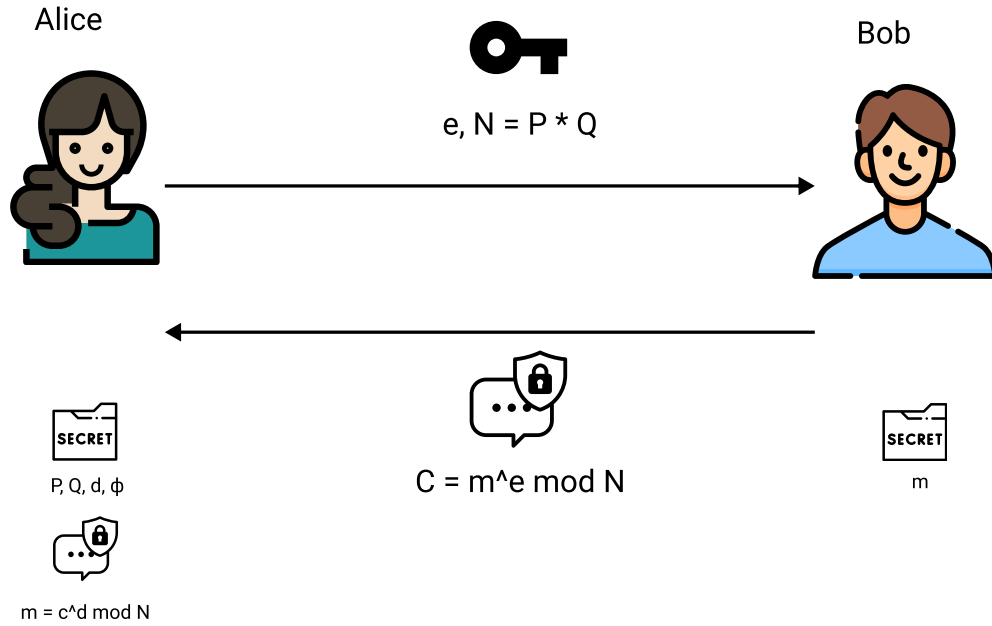
If we multiply both parts by  $M$ , we get

$$m \cdot m^{K \cdot \phi(N)} = m^{K \cdot \phi(N) + 1} = m \bmod N$$

It follows that Alice is able to define the secret  $d$  as follows

$$\begin{aligned} e \cdot d &= K \cdot \phi(N) + 1 \\ d &= \frac{K \cdot \phi(N) + 1}{e} \end{aligned}$$

The following image demonstrates the concept of RSA approach



**Figure 3.** RSA algorithm concept diagram.

To summarize, the process by the steps is as follows

- Alice defines the large secret prime numbers  $P, Q$ .
- Alice computes  $N = P \cdot Q$  and  $\phi = (P - 1)(Q - 1)$
- Alice chooses an integer  $e$ ,  $1 < e < \phi$  such that  $\gcd(e, \phi) = 1$ .
- Alice computes secret exponent  $d$ ,  $1 < d < \phi$  such that  $ed \equiv 1 \bmod \phi$ .
- Alice shares public key  $(N, e)$  with Bob and keeps private key  $(d, p, q)$  is secret.
- Bob defines the message  $m$ , encrypts it as  $C = m^e \bmod N$ .
- Bob sends  $C$  to Alice.
- Alice decrypts  $C$  using her secret  $d$ , so she gets  $m$

$$m = C^d \bmod N$$

Security of the RSA approach is based on the complexity of fundamental problem of prime factorization, which takes decades to solve having enough large number.

## REFERENCES

- [1] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [2] Clifford C Cocks. A note on non-secret encryption. *CESG Memo*, 1973.