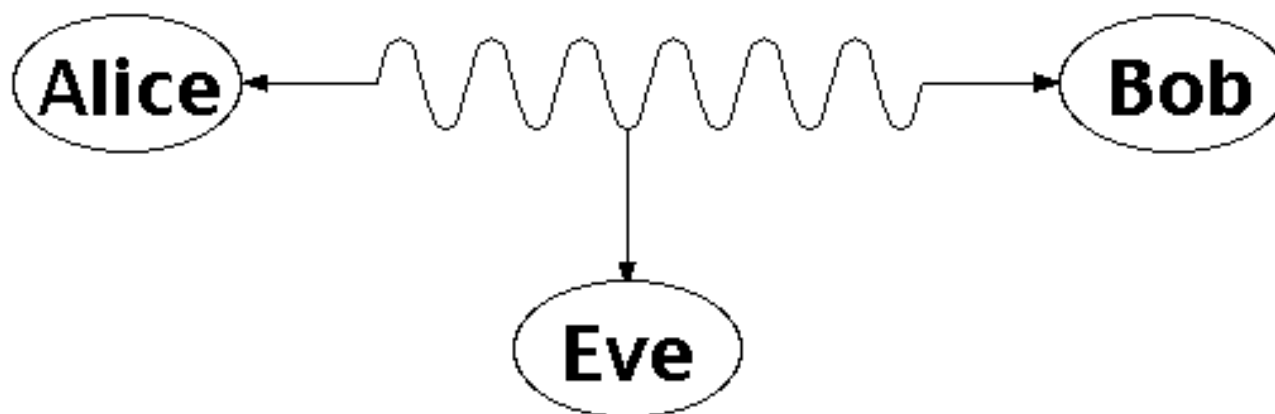
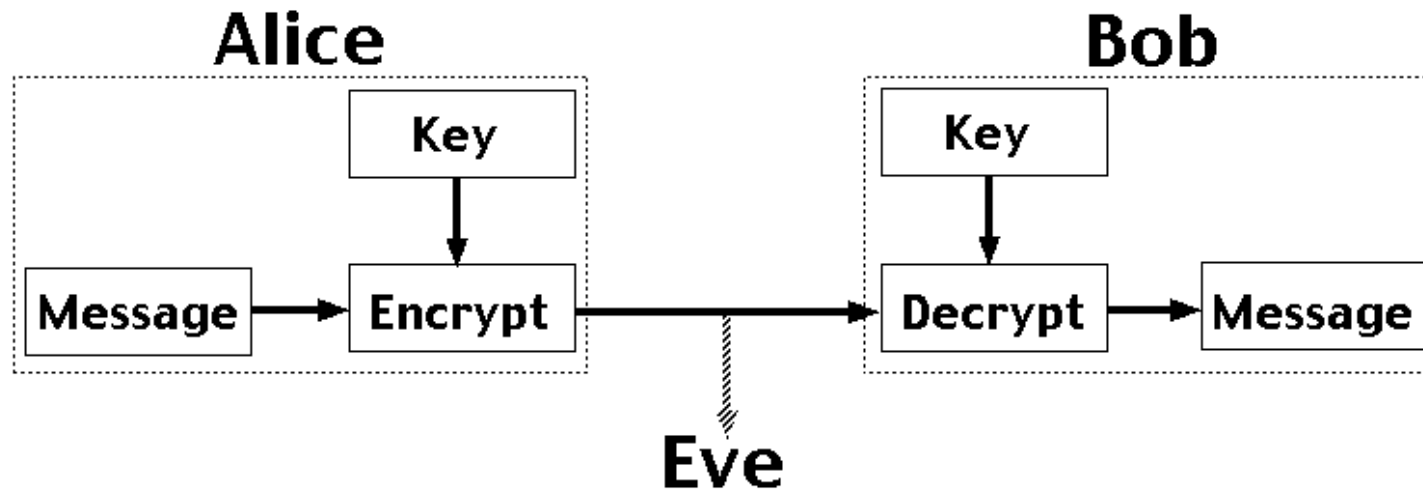


Public Key Cryptography and the RSA Cryptosystem

Two people, say Alice and Bob, would like to exchange secret messages; however, Eve is eavesdropping:



One technique would be to use an encryption technique based on an **encryption key**, but this poses a challenge: how do they exchange the encryption key without Eve receiving it?



Traditional Cryptography

In normal cryptography, both parties need to know a secret key k . The sender then encodes message m using key k via some method f to get the ciphertext c :

$$c = f(m, k).$$

Then, the receiver decodes the ciphertext c using key k via some method g to get back the original message m :

$$m = g(c, k).$$

The issue here is how to securely exchange the secret key k .

If we could find a method of encryption / decryption such that exchanging the necessary keys does not reveal to Eve how to decrypt intercepted messages, then we would avoid this problem altogether.

Public Key Cryptosystem

The idea is that the receiver publically publishes a **public key** k . Then anyone who wishes to encode a message m can do so:

$$c = f(m, k).$$

The receiver has a **private key** k' that is needed to decode the ciphertext c to receive the original message m :

$$m = g(c, k').$$

Both the encoding technique f and the decoding technique g are also publically known; the only secret information is k' .

While it is possible, it is computationally difficult to compute k' from k : the **key pair** can be chosen so that in the amount of time it takes to derive k' from k , the information m no longer has significant value.

RSA Cryptosystem

The most common form of public key cryptosystem is RSA, which stands for Rivest, Shamir, and Adleman, who invented it. It is based on modular arithmetic and large primes, and its security comes from the computational difficulty of factoring large numbers.

The idea is as follows: select p and q to be large primes (at least several hundred digits); the degree of security is dependent on the size of p and q . Take $n = pq$. Then the **public key** is a pair $k = (n, e)$ such that:

$$\gcd(e, (p - 1)(q - 1)) = 1.$$

The **encoding function** is:

$$f(m, k) \equiv m^e \pmod{n}.$$

This assumes that the message can be represented by an integer $m < n$ with $\gcd(m, p) = \gcd(m, q) = 1$; if not, we can break m down into smaller pieces and encode each individually.

The **private key** is a pair $k' = (n, d)$ such that:

$$de \equiv 1 \pmod{(p-1)(q-1)}.$$

The **decoding function** is:

$$g(c, k') = c^d \pmod{n}.$$

The security of the algorithm lies in the challenge of **prime factorization**: in order to calculate d , it is necessary to factor n to get p and q , which is very difficult (exponential in the number of digits in p and q).

We now proceed to show that RSA actually works.

Proof of the RSA Cryptosystem

Theorem (RSA Cryptosystem)

Let p, q be primes with $n = pq$ and let e be an integer such that $\gcd(e, (p-1)(q-1)) = 1$, with $ed \equiv 1 \pmod{(p-1)(q-1)}$. Let m be an integer with $m < n$ and $\gcd(m, p) = \gcd(m, q) = 1$. Define $k = (n, e)$ and $k' = (n, d)$, and the functions:

$$f(m, k) = m^d \pmod{n}$$

$$g(c, k) = c^e \pmod{n}.$$

Then we claim that:

$$g(f(m, k), k') = m.$$

Proof.

We have that:

$$g(f(m, k), k') = (m^e \bmod n)^d \bmod n = m^{ed} \bmod n.$$

By the choice of e and d , we have that: $ed \equiv 1 \pmod{(p-1)(q-1)}$, or, equivalently, for some integer s , $ed = 1 + s(p-1)(q-1)$. By Fermat's Little Theorem, $m^{p-1} \equiv 1 \pmod{p}$ and $m^{q-1} \equiv 1 \pmod{q}$, giving:

$$m^{ed} \equiv m^{1+s(p-1)(q-1)} \equiv m \cdot (m^{p-1})^{s(q-1)} \equiv m \cdot 1^{s(q-1)} \equiv m \pmod{p}.$$

Similarly, $m^{ed} \equiv m \pmod{q}$. Since $\gcd(p, q) = 1$, by the Chinese Remainder Theorem, $m^{ed} \equiv m \pmod{pq}$ as required. □

Note that we can apply the same argument to show that:

$$f(g(m, k'), k) = m.$$

Thus, the owner of the private key can encrypt a message m using the private key, which can then be decrypted by anyone using the public key, and prove that only the private key owner could have encrypted it. This is the basis of **digital signature systems**.

Example: Bob wants to receive messages from Alice, so he selects two primes, say $p = 43$ and $q = 59$. (We choose small primes for feasibility of the example; in reality, they would be vastly larger.) Then $n = pq = 2537$ and $(p - 1)(q - 1) = 2436$. He then picks $e = 13$, which has the property that:

$$\gcd(e, (p - 1)(q - 1)) = \gcd(13, 2436) = 1.$$

Bob then calculates $d = 937$, the inverse of e mod 2436:

$$de \equiv 937 \times 13 \equiv 12181 \equiv 5 \times 2436 + 1 \equiv 1 \pmod{2436}.$$

Bob publishes the **public key** $k = (2537, 13)$.

Alice wants to send message “STOP” to Bob using RSA. She encodes this: $S \rightarrow 18$, $T \rightarrow 19$, $O \rightarrow 14$, $P \rightarrow 15$, i.e. 1819 1415 grouped into blocks of 4. Thus, $m = m_1m_2 = 18191415$. Each block is encrypted:

$$1819^{13} \pmod{2537} = 2081$$

$$1451^{13} \pmod{2537} = 2182$$

Then the encrypted message is 20812182. Bob has **private key** $k' = (2537, 937)$, and computes:

$$2081^{937} \bmod 2537 = 1819 \rightarrow \text{ST}$$

$$2812^{937} \bmod 2537 = 1415 \rightarrow \text{OP}$$

Thus, the original message was STOP.