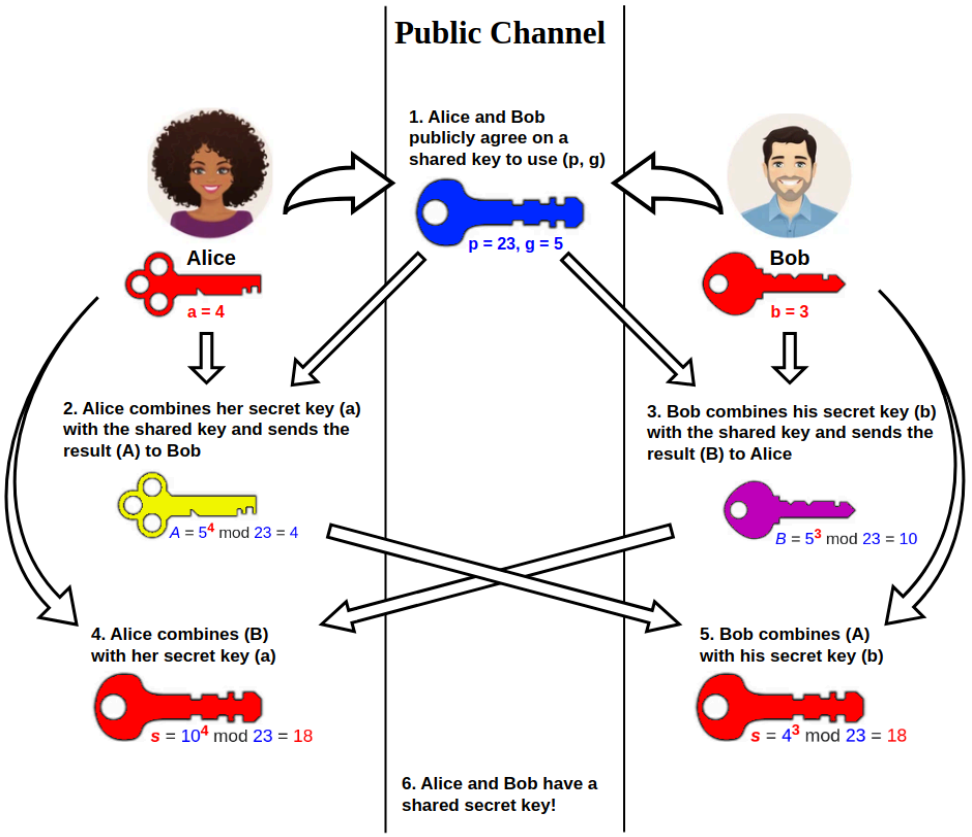
Key Agreement - Acordul de Chei

Key agreement-ul este un proces prin care două sau mai multe părți ajung la un acord comun privind o cheie secretă pe care o pot folosi pentru a comunica în mod securizat. Scopul acestui proces este de a permite părților să genereze o cheie comună, astfel încât să poată comunica în mod confidențial, asigurând că informațiile transmise sunt protejate împotriva accesului neautorizat.

Diffie-Hellman

- Este o metodă pentru schimbul securizat de chei criptografice pe canale de comunicare nesigure, fără a compromite securitatea și integritatea transmisiei datelor.
- A fost unul dintre primele protocoale de cheie publică, conceput de Ralph Merkle și numit după Whitfield Diffie și Martin Hellman.
- Folosește conceptul de exponențiere modulară în grupuri ciclice (de obicei, grupuri de numere prime).
- Nu asigură autentificarea părților, de aceea este adesea combinat cu alte protocoale pentru a asigura autentificarea și confidențialitatea.



Valorile roșii reprezintă valorile **secrete**, iar valorile albastre valorile **cunoscute**.

1. Alice and Bob publicly agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $a = 4$, then sends Bob $A = g^a \bmod p$
   - $A = 5^4 \bmod 23 = 4$ (in this example both A and a have the same value 4, but this is usually not the case)
3. Bob chooses a secret integer $b = 3$, then sends Alice $B = g^b \bmod p$
   - $B = 5^3 \bmod 23 = 10$
4. Alice computes $s = B^a \bmod p$
   - $s = 10^4 \bmod 23 = 18$
5. Bob computes $s = A^b \bmod p$
   - $s = 4^3 \bmod 23 = 18$
6. Alice and Bob now share a secret (the number 18).

Both Alice and Bob have arrived at the same values because under mod p,

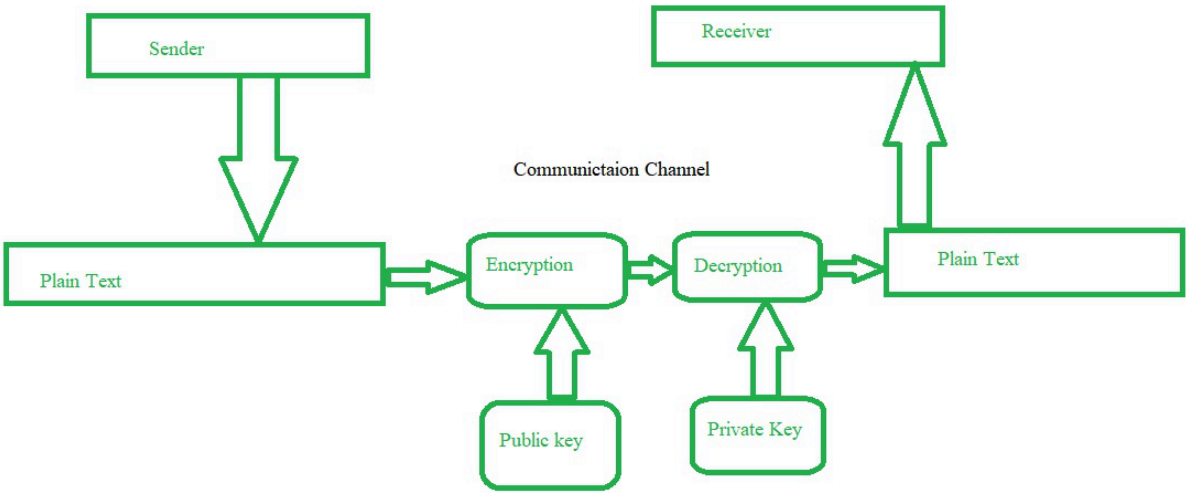
$$A^b \bmod p = g^{ab} \bmod p = g^{ba} \bmod p = B^a \bmod p$$

More specifically,

$$(g^a \bmod p)^b \bmod p = (g^b \bmod p)^a \bmod p$$

RSA

- RSA este un algoritm de criptare cu cheie publică conceput de Ron Rivest, Adi Shamir și Leonard Adleman Asta înseamnă că este folosit un set de două chei: o cheie publică pentru criptare și o cheie privată pentru decriptare. Mesajele criptate cu cheia publică pot fi decriptate numai cu cheia privată corespunzătoare și invers. Acest model de criptare asigură confidențialitatea datelor.
- Este folosit și pentru semnături digitale. În acest caz, semnătura digitală este generată cu cheia privată și poate fi verificată cu cheia publică corespunzătoare. Acest proces asigură autenticitatea și integritatea datelor, deoarece semnătura digitală este unică pentru mesajul dat și este foarte dificil de falsificat.
- Criptarea RSA se bazează pe dificultatea factorizării unui număr mare în factori primi. Cu cât cheile sunt mai mari, cu atât este mai dificil să se spargă criptarea RSA prin metode de forță brută sau prin factorizare. Cu toate acestea, avansul tehnologic poate face unele dimensiuni de cheie mai mici vulnerabile la atacuri.
- RSA poate fi folosit și pentru acordul de cheie, dar este mai rar folosit în acest scop datorită performanței sale slabe în comparație cu algoritmi specializați precum Diffie-Hellman.
- RSA este utilizat într-o gamă largă de aplicații, inclusiv comunicare securizată pe internet (HTTPS, SSL/TLS), semnături digitale pentru autentificare și autorizare, criptare a datelor stocate pe dispozitive și multe altele. Este unul dintre algoritmii de criptare cu cheie publică cei mai utilizați și este integrat în multe protocoale și standarde de securitate.



Generarea de chei

Key generation [edit]

The keys for the RSA algorithm are generated in the following way:

1. Choose two large prime numbers p and q.
   - To make factoring harder, p and q should be chosen at random, be both large and have a large difference.[1] For choosing them the standard method is to choose random integers and use a primality test until two primes are found.
   - p and q should be kept secret.
2. Compute $n = pq$.
   - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
   - n is released as part of the public key.
3. Compute $\lambda(n)$, where λ is Carmichael's totient function. Since $n = pq$, $\lambda(n) = \operatorname{lcm}(\lambda(p), \lambda(q))$, and since p and q are prime, $\lambda(p) = \varphi(p) = p - 1$, and likewise $\lambda(q) = q - 1$. Hence $\lambda(n) = \operatorname{lcm}(p - 1, q - 1)$.
   - The lcm may be calculated through the Euclidean algorithm, since $\operatorname{lcm}(a, b) = \frac{|ab|}{\gcd(a, b)}$.
   - $\lambda(n)$ is kept secret.
4. Choose an integer e such that $1 < e < \lambda(n)$ and $\gcd(e, \lambda(n)) = 1$; that is, e and $\lambda(n)$ are coprime.
   - e having a short bit-length and small Hamming weight results in more efficient encryption – the most commonly chosen value for e is $2^{16} + 1 = 65\,537$. The smallest (and fastest) possible value for e is 3, but such a small value for e has been shown to be less secure in some settings.[15]
   - e is released as part of the public key.
5. Determine d as $d \equiv e^{-1} \pmod{\lambda(n)}$; that is, d is the modular multiplicative inverse of e modulo $\lambda(n)$.
   - This means: solve for d the equation $de \equiv 1 \pmod{\lambda(n)}$; d can be computed efficiently by using the extended Euclidean algorithm, since, thanks to e and $\lambda(n)$ being coprime, said equation is a form of Bézout's identity, where d is one of the coefficients.
   - d is kept secret as the *private key exponent*.

Exemplu

Example [edit]

Here is an example of RSA encryption and decryption:[b]

1. Choose two distinct prime numbers, such as
   $p = 61$ and $q = 53$.
2. Compute $n = pq$ giving
   $n = 61 \times 53 = 3233.$
3. Compute the Carmichael's totient function of the product as $\lambda(n) = \operatorname{lcm}(p-1, q-1)$ giving
   $\lambda(3233) = \operatorname{lcm}(60, 52) = 780.$
4. Choose any number $2 < e < 780$ that is coprime to 780. Choosing a prime number for e leaves us only to check that e is not a divisor of 780.
   Let $e = 17$.
5. Compute d, the modular multiplicative inverse of $e \pmod{\lambda(n)}$, yielding
   $d = 413,$
   as $1 = (17 \times 413) \bmod 780.$

The **public key** is $(n = 3233, e = 17)$. For a padded plaintext message m, the encryption function is

$$\begin{aligned} c(m) &= m^e \bmod n \\ &= m^{17} \bmod 3233. \end{aligned}$$

The **private key** is $(n = 3233, d = 413)$. For an encrypted ciphertext c, the decryption function is

$$\begin{aligned} m(c) &= c^d \bmod n \\ &= c^{413} \bmod 3233. \end{aligned}$$

For instance, in order to encrypt $m = 65$, one calculates

$$c = 65^{17} \bmod 3233 = 2790.$$

To decrypt $c = 2790$, one calculates

$$m = 2790^{413} \bmod 3233 = 65.$$

Mai multe informații

- Diffie–Hellman key exchange
- RSA (cryptosystem)
- Difference Between Diffie-Hellman and RSA