Algoritmo Diffie-Hellman

- Diseñado en 1976, se denomina normalmente "**algoritmo Diffie-Hellman de intercambio de clave**"
- Permite a dos usuarios cualesquiera **intercambiar**, de forma confidencial, una **clave secreta *K* (o de sesión)** para, posteriormente, cifrar de forma simétrica los mensajes entre ellos dos
- La efectividad del algoritmo depende de la **dificultad de computar logaritmos discretos**
  - Si α es una *raíz primitiva* del número primo *q*, entonces los números

    $\alpha \bmod q,\ \alpha^2 \bmod q,\ \ldots,\ \alpha^{q-1} \bmod q$

    son distintos entre sí, y sus valores son los enteros *1* a *q-1*, en cualquier orden

  - Para cualquier entero *b* y una raíz primitiva α del número primo *q*, se puede encontrar un exponente único *i* tal que:

    $b \equiv \alpha^i \pmod q$ donde $0 \leq i \leq (q-1)$

    i es el logaritmo discreto de *b*, y se representa $dlog_{\alpha,q}(b)$

- Raíz primitiva para sistemas criptográficos:
  - Dados *m* y *n*, tales que *mcd(m,n) = 1*, el menor *g* tal que $n^g \equiv 1 \pmod{m}$ se denomina gaussiano de *n* respecto de *m*. Si *m* es un primo y *n* es un número cualquiera tal que m != n, y el gaussiano de *n* es ***m−1***, entonces se dice que *n* es una raíz primitiva de *m*. Un ejemplo es *m*=11 y *n*=2.

1	2	3	4	5	6	7	8	9	10
$2^1=2$	$2^2=4$	$2^3=8$	$2^4=16$	$2^5=32$	$2^6=64$	$2^7=128$	$2^8=256$	$2^9=512$	$2^{10}=1024$
2	4	8	5	10	9	7	3	6	1

  - Este tipo de raíces primitivas permiten cumplir los requisitos mostrados en la transparencia anterior. Existen otras raíces primitivas (p.ej. *m*=14, *n*=5) que no cumplen estos requisitos puesto que el gaussiano de *n* **no es** *m−1*

1	2	3	4	5	6	7	8	9	10	11	12	13
5	11	13	9	3	1	5	11	13	9	3	1	5

Global Public Elements	
q	prime number
a	$a < q$ and a a primitive root of q

User A Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = a^{X_A} \bmod q$

User B Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = a^{X_B} \bmod q$

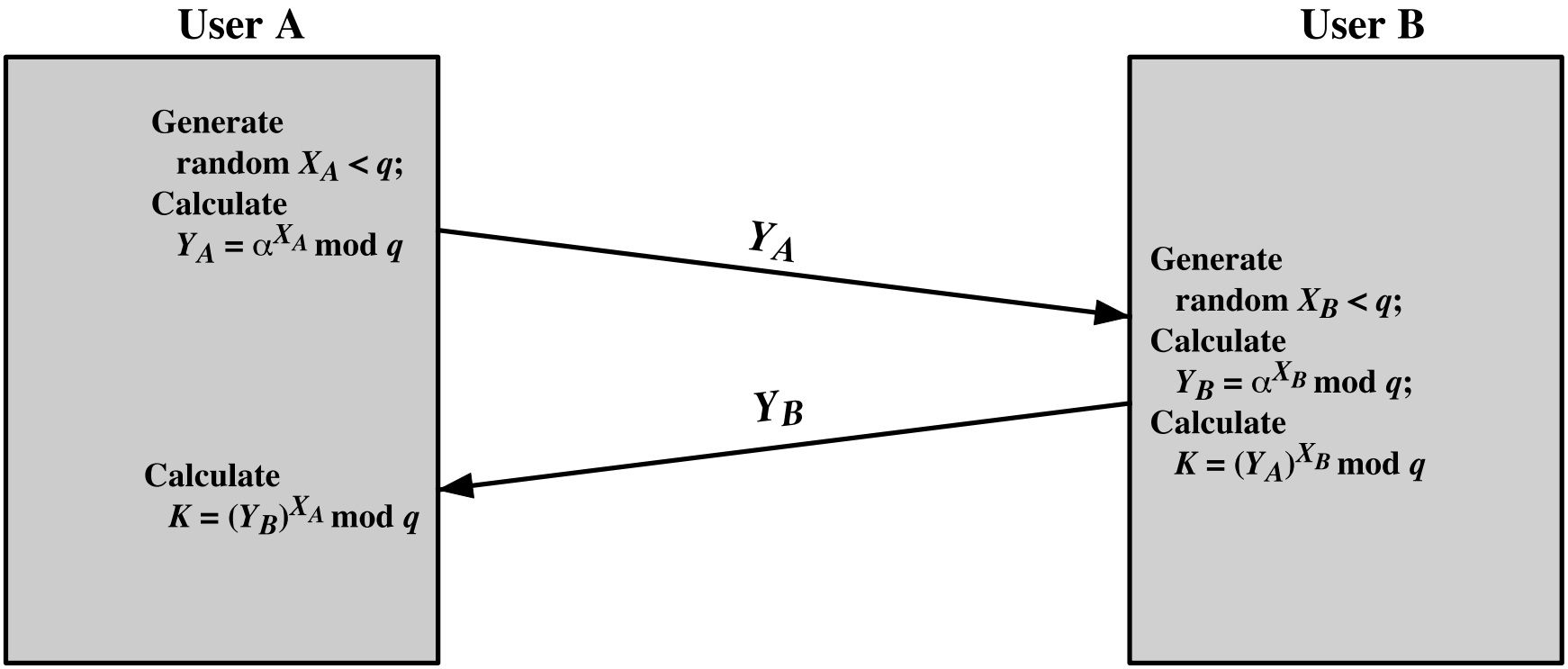
Calculation of Secret Key by User A

$K = (Y_B)^{X_A} \bmod q$

Calculation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

Algoritmo Diffie-Hellman de intercambio de clave



Secuencia ordenada de pasos del algoritmo

- Alice:
  - Valores públicos: q, α
  - Xa < q: clave privada
  - $Ya = \alpha^{Xa} \bmod q$
  - $Yb = \alpha^{Xb} \bmod q$
  - $K = (Yb)^{Xa} \bmod q$
  - $K = (\alpha^{Xb})^{Xa} \bmod q$
- Bob:
  - Valores públicos: q, α
  - Xb < q: clave privada
  - $Yb = \alpha^{Xb} \bmod q$
  - $Ya = \alpha^{Xa} \bmod q$
  - $K = (Ya)^{Xb} \bmod q$
  - $K = (\alpha^{Xa})^{Xb} \bmod q$

$$K = \alpha^{Xb\,Xa} \bmod q = \alpha^{Xa\,Xb} \bmod q$$

- Ejemplo:
  ① $q = 353;\ \alpha = 3$ (3 es raíz primitiva de 353)

  ② *A* y *B* seleccionan sus correspondientes claves privadas: $X_A = 97,\ X_B = 233$

  ③ *A* calcula su clave pública: $Y_A = 3^{97} \bmod 353 = 40$

  ④ *B* calcula su clave pública: $Y_B = 3^{233} \bmod 353 = 248$

  ⑤ *A* y *B* se intercambian sus claves públicas

  ⑥ *A* calcula $K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$

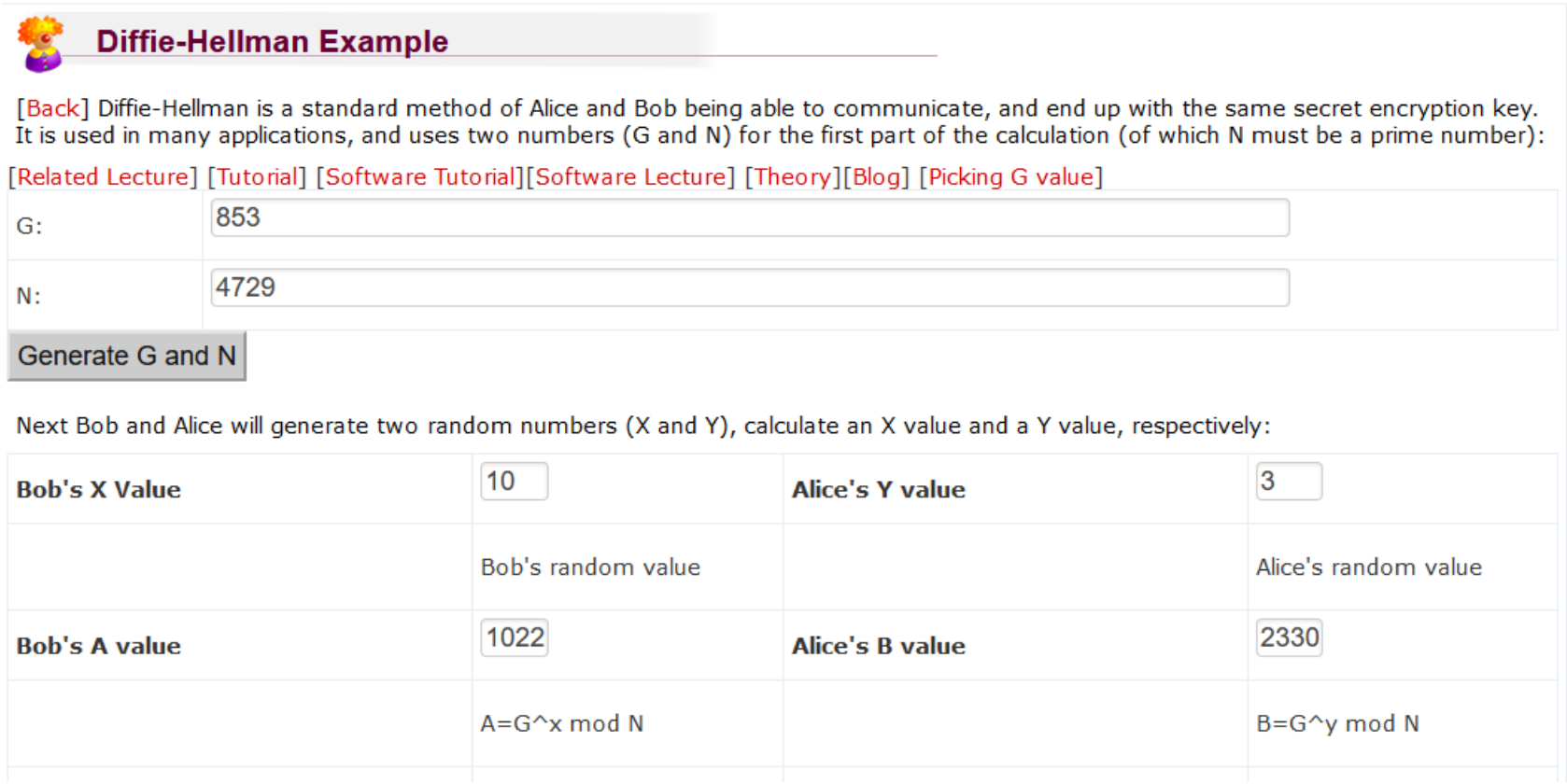
  ⑦ *B* calcula $K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$

- Ejemplo (continuación):
  - Se asume que el atacante tendrá la siguiente información:

    $q = 353; \quad \alpha = 3; \quad Y_A = 40; \quad Y_B = 248$

  - En este ejemplo simple, sería posible usar un ataque exhaustivo para determinar que $K = 160$
    - El atacante calcula fácilmente $3^{Xa} \bmod 353 = 40$ o bien $3^{Xb} \bmod 353 = 248$
  - Con números más grandes el problema no es resoluble en un tiempo razonable

- https://asecuritysite.com/encryption/diffie



Diffie-Hellman Example

[Back] Diffie-Hellman is a standard method of Alice and Bob being able to communicate, and end up with the same secret encryption key. It is used in many applications, and uses two numbers (G and N) for the first part of the calculation (of which N must be a prime number):

[Related Lecture] [Tutorial] [Software Tutorial][Software Lecture] [Theory][Blog] [Picking G value]

G:	853
N:	4729

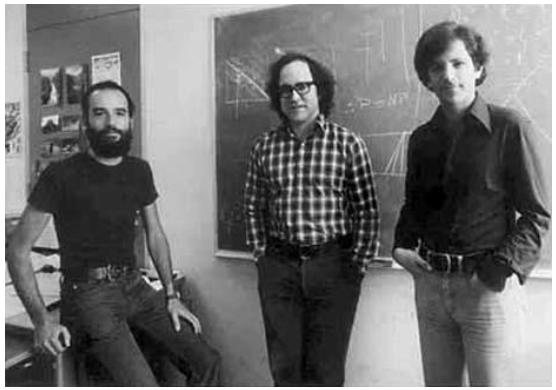
Generate G and N

Next Bob and Alice will generate two random numbers (X and Y), calculate an X value and a Y value, respectively:

Bob's X Value	10	**Alice's Y value**	3
	Bob's random value		Alice's random value
Bob's A value	1022	**Alice's B value**	2330
	A=G^x mod N		B=G^y mod N

Algoritmo RSA

- Es el criptosistema de clave pública más usado
  - Su nombre procede de sus inventores: *Rivest, Shamir y Adleman* del Instituto Tecnológico de Massachusetts (MIT)
- Estos se basaron para su invención, en 1.977, en una idea bastante "simple":
  - "***es muy fácil multiplicar dos números enteros primos grandes, pero extremadamente difícil hallar la factorización de ese producto***"
  - puede darse a conocer dicho producto sin que nadie descubra esos números de procedencia, a no ser que conozca al menos uno de ellos



Se piensa que RSA será seguro mientras no se conozcan "**formas rápidas**" de descomponer un número grande en producto de primos

- Parámetros necesarios:
  - encontrar dos números primos grandes *p* y *q*, y calcular

    $n = p * q \quad ; \quad p \neq q$

  - se calcula $\varphi(n)$, de forma que

    $\varphi(n) = (p - 1) * (q - 1)$

    Para extraer los primos, donde φ(n) se define como el número de enteros positivos menores o iguales a n y coprimos de n

  - se elige aleatoriamente un número grande *e* tal que

    $MCD\ (e, \varphi(n)) = 1; \quad e < \varphi(n)$

    (o sea, *e* y $\varphi(n)$ son primos relativos o coprimos)

  - Se determina el número *d* que cumple

    $e * d \equiv 1 \pmod{\varphi(n)}$

    (donde *d* debe ser el multiplicador inverso de *e mod* $\varphi(n)$)

Calcular las claves

- *n*, *d* y *e* (y por supuesto, *p* y *q*) son la base del sistema
  - n módulo
  - d clave privada
  - e clave pública

- Las claves tienen el siguiente uso:
  - la **clave privada**: para descifrar o firmar mensajes
  - la **clave pública**: para cifrar o verificar la firma

- Para *cifrar*: $C = M^e \pmod n$
- Para *descifrar*: $M = C^d \pmod n$

$$C^d = (M^e)^d \equiv M^{ed} \pmod n$$

- RSA se trata, por lo tanto, de un algoritmo exponencial

- Ejemplo del cálculo de la clave pública y privada:

① Seleccionar primos: $p = 17$, $q = 11$

① Calcular $n = pq = 17 \times 11 = 187$

② Calcular $\varnothing(n)=(p-1)(q-1) = 16 \times 10 = 160$

③ Seleccionar e: $mcd(e,160)=1$ y $e < 160$
se selecciona $e = 7$

⑤ Determinar d: $d.e \equiv 1 \mod 160$
$d = 23$ dado que $23x7 = 161 \mod 160 = 1$

⑥ Clave pública = 7 y módulo = 187

⑦ Clave privada = 23

- El texto original, *M* (y también el cifrado, *C*) debe tomarse como un número decimal. De esta forma, si se trabaja con el código ASCII:

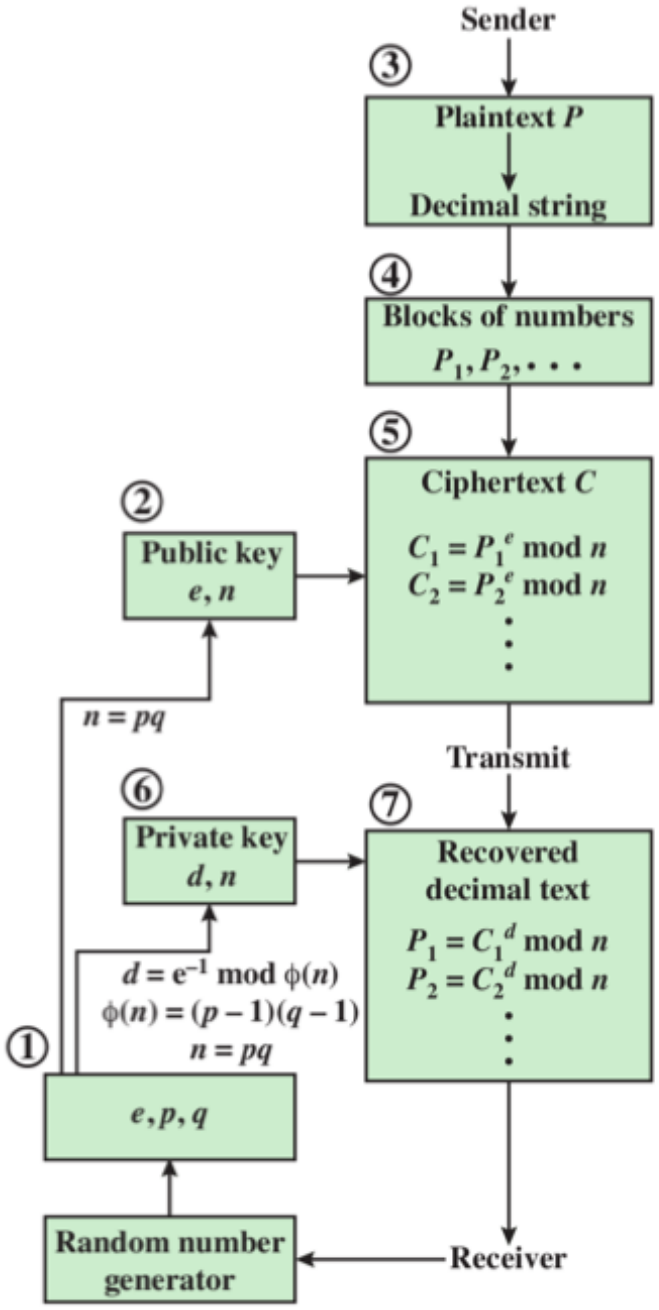
  M = “Hola” equivaldría al valor decimal 72 111 108 097

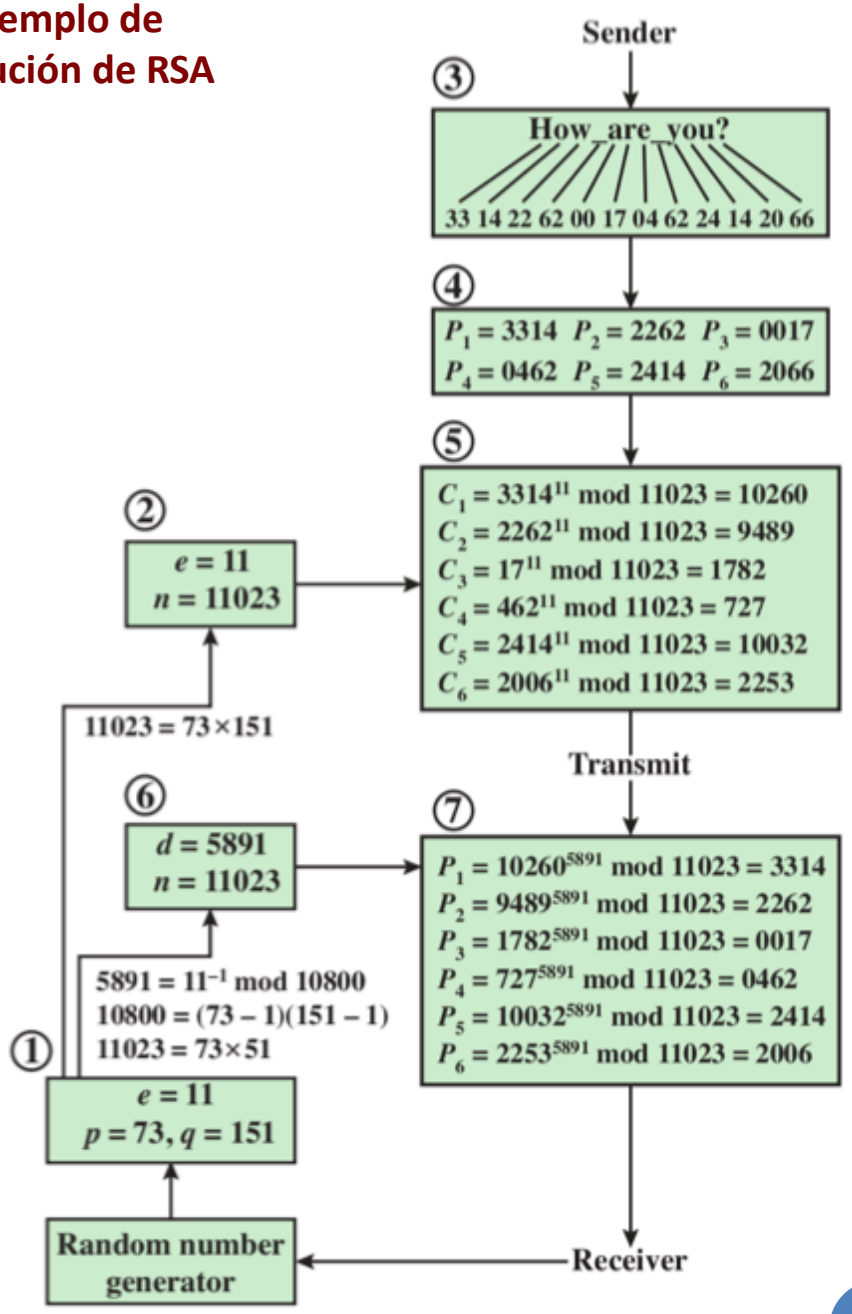
- Además, como se está trabajando en aritmética modular, todos los valores usados deben ser menores que el módulo *n*
  - Esto significa que si el valor decimal del mensaje es superior al módulo, $M > n$, entonces *M* debe ser troceado en otros menores que *n*
  - Suponiendo $n = 100000$, daría lugar a los bloques

    $m_0 = 72111$, $m_1 = 10809$ y $m_2 = 7$

- Los bloques m_i no deben/pueden ser pequeños, pues entonces el criptoanalista puede construirse una tabla donde tenga todos los posibles m_i y sus respectivos c_i
- Para seleccionar p ó q debe testearse si es primo o no con cualquiera de los tests existentes (se recomienda *Miller-Rabin*)
- También hay que tener cuidado al elegir el tamaño
  - **cuanto mayor sea, más seguro, pero también más lento** será el sistema en sus cálculos
- Se han propuesto sistemas en los que se busca un valor e muy pequeño, con lo cual hacer más rápido el proceso de cifrado

Ejemplo de ejecución de RSA





Comparativa: Criptografía Simétrica vs. Asimétrica

Atributo	Clave simétrica	Clave asimétrica
Años en uso	Miles	Menos de 50
Uso principal	Cifrado de grandes volúmenes de datos	Intercambio de claves; firma digital
Estándar actual	DES, Triple DES, AES	RSA, Diffie-Hellman, DSA
Velocidad	Rápida	Lenta
Claves	Compartidas entre emisor y receptor	Privada: sólo conocida por una persona Pública: conocida por todos
Intercambio de claves	Difícil de intercambiar por un canal inseguro	La clave pública se comparte por cualquier canal La privada nunca se comparte
Longitud de claves	56 bits (vulnerable) 256 bits (seguro)	1024 – 2048 (RSA) 172 (curvas elípticas)
Servicios de seguridad	Confidencialidad Integridad Autenticación	Confidencialidad Integridad Autenticación, No repudio