Criptografía asimétrica

- Se presupone siempre que hay un atacante escuchando la conversación (Eve)
- Para evitar compartir la clave secreta, esta se genera usando pares de claves pública/privada
- Clave pública: aquella que conocen todos los usuarios
- Clave privada: cada usuario tiene una propia, que no debe compartirse
- Las claves son inversas: lo que se cifra con una, se descifra con la otra

CIFRADO

- RSA
- ElGamal
- Cramer-Shoup

FIRMA DIGITAL

- DSA
- ECDSA/EdDSA
- ElGamal (Signature)

INTERCAMBIO DE CLAVES

- Diffie-Hellman
- ECDH

OTROS USOS

- Criptodivisas
- Certificados
- Smart Contracts
- …

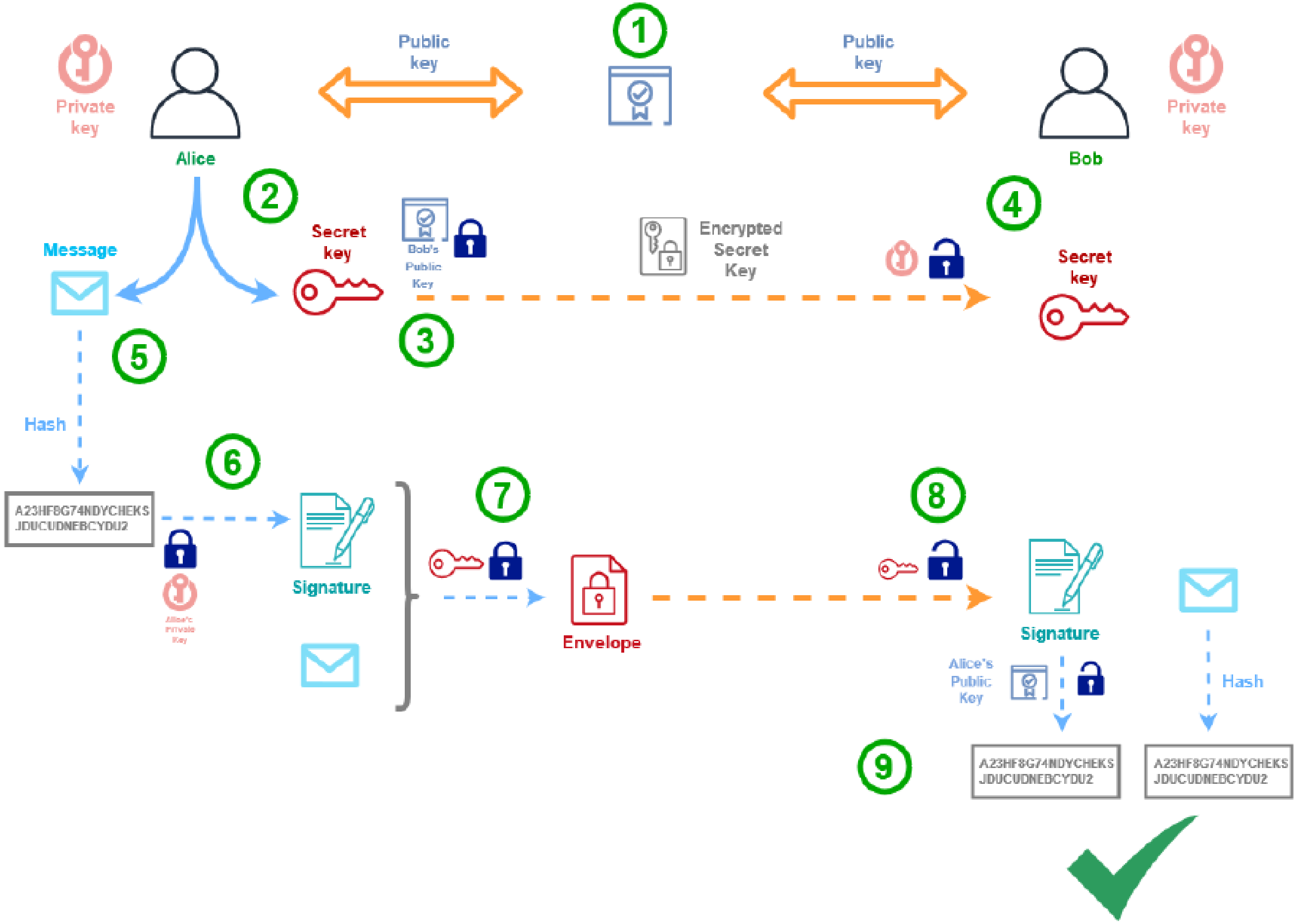
VENTAJAS

- Evita *eavesdropping*
- Fácilmente reversible
- Fácil de implementar

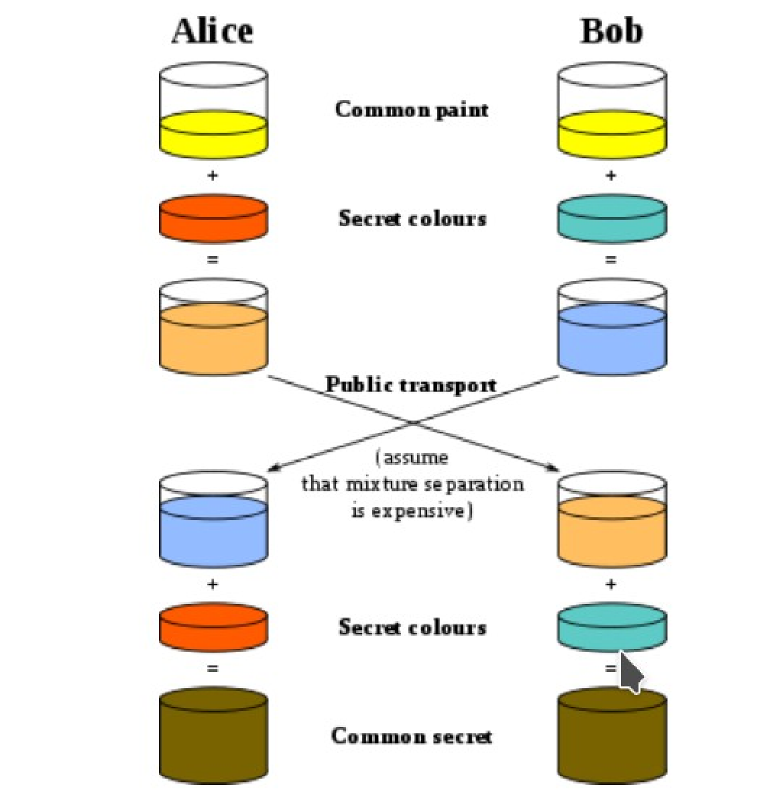
INCONVENIENTES

- Mucho más lento que los algoritmos simétricos
- Gran carga de procesador (cálculos matemáticos)

En general, se evita usar algoritmos asimétricos para cifrar.


1
Public key
Public key
Private key
Alice
Bob
Private key
2
Message
Secret key
Bob's Public Key
Encrypted Secret Key
4
Secret key
3
5
Hash
6
A23HF8G74NDYCHEKS JDUCUDNEBCYDU2
Signature
7
Envelope
8
Signature
Alice's Public Key
Hash
9
A23HF8G74NDYCHEKS JDUCUDNEBCYDU2
A23HF8G74NDYCHEKS JDUCUDNEBCYDU2

Diffie-Hellman



Diffie-Hellman

1. Alice y Bob acuerdan un módulo *p* (primo) y una base *q*.
Suponemos p = 23 y q = 5.

2. Alice elige un entero secreto, $a = 4$ y le envía a Bob: $A = g^a \bmod p$

   $A = 5^4 \bmod 23 = 4$

3. Bob elige su propio secreto, $b = 3$ y repite el proceso, enviándoselo a Alice.

   $B = 5^3 \bmod 23 = 10$

4. Ahora Alice calcula el secreto común como: $S = B^a \bmod p$.

   $S = 10^4 \bmod 23 = \underline{18}$

5. Bob puede calcular el mismo secreto: $S = A^b \bmod p$.

   $S = 4^3 \bmod 23 = \underline{18}$

6. Listo. Alice y Bob han obtenido un secreto común.

Diffie-Hellman

- Sabemos que $S = g^{a*b} \bmod p$

- Un atacante solo conoce *A*, *B*, *g* y *p*

- $18 = 5^{a*b} \bmod 23$ ¿Cuáles son *a* y *b*?

- Este problema es conocido como *Problema del Logaritmo Discreto*

- Se trata de un problema *NP* y no existe algoritmo eficiente


RSA

RSA

- Rivest – Shamir – Adleman (1983)
- Se usan claves algebraicamente inversas
- Álgebra avanzada (cerveza y os lo cuento)
- Problema de factorización de enteros (NP)

RSA – Generación de claves

1. Se toman dos primos p y q de gran tamaño. $N = p*q$

2. $\Phi(N) = \Phi(p * q) = \Phi(p) * \Phi(q) = (p – 1)(q – 1)$

3. Se toma un número e coprimo con $\Phi(N)$. Normalmente 65537.

4. (e,N) será la clave pública.

5. Se calcula d tal que $d * e \equiv 1 \mod \Phi(N)$. Se dice que d es el <u>inverso modular</u> de e.

6. (d, N) será la clave privada.

RSA – Cifrado

Alice le quiere enviar un mensaje *m* a Bob. Conoce la clave de Bob *(e, N).*

1. $c = m^e \bmod N$

    (c es el mensaje cifrado usando la clave pública de Bob)

2. Bob calcula ahora:

    $c^d \bmod N = m^{e*d} \bmod N =$ (<u>*este paso vale una cerve*</u>) $= m$

    (Bob eleva el texto cifrado a su propia clave privada, que es la inversa de la pública con la que se cifró)

3. Bob puede leer el mensaje.

RSA – Encoding

Los mensajes no suelen ser números enteros. Queremos poder cifrar texto.

El procedimiento más común en CTFs es:

Cifrado:

mensaje => hexadecimal => 0110101… => int(0110101…) => 167485...

Descifrado:

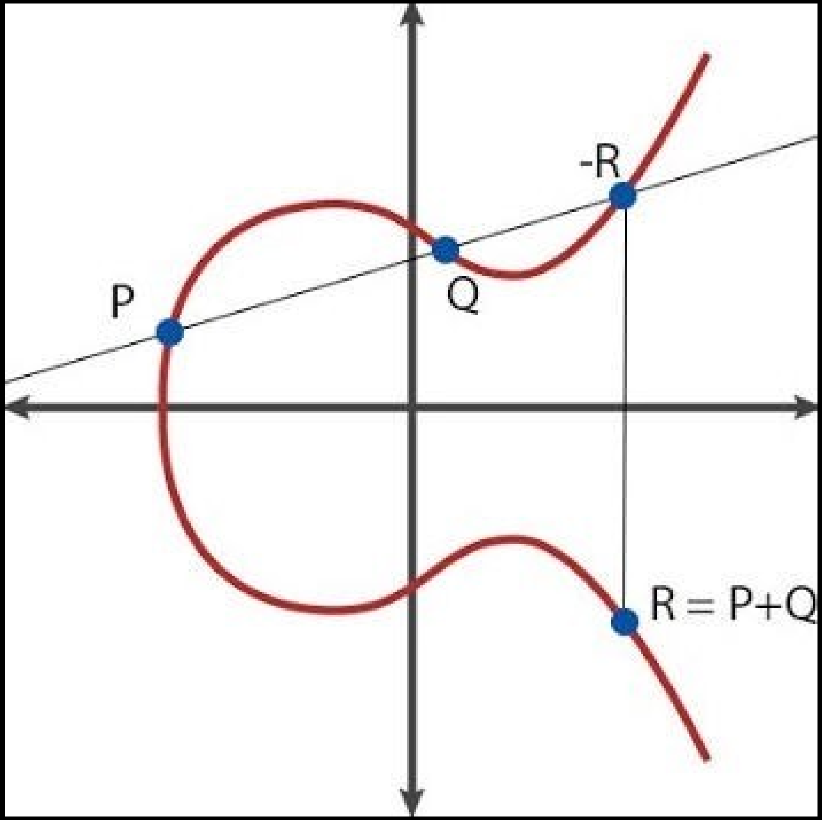
37821… => 101101… => hexadecimal (ceros por la izquierda)

Sin embargo, existen otros, como *base64*

DEMO TIME

HACKMADRID %27

Elliptic Curve Diffie-Hellman
ECDH



La info extra ahora cuesta DOS cervezas (y unos cacahuetes)

¿Qué es una curva elíptica?

Conjunto de puntos definido por una ecuación del tipo:

$y^2 = x^3 + ax + b$ *Tales que* $4a^3 + 27b^2 \neq 0$

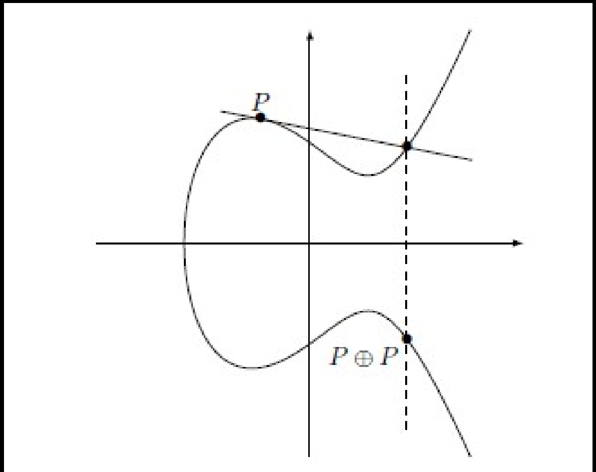
Un punto de la curva será $P = (x, y)$ donde x e y satisfacen la ecuación.

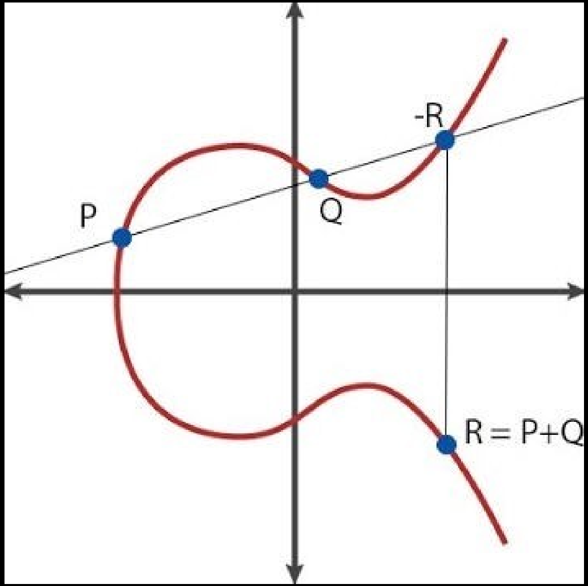
Hay tantas curvas elípticas como combinaciones de a y b existan

No todas son igual de seguras

Operaciones

El conjunto de puntos de una curva elíptica forma un *grupo algebraico*
Esto significa que existe una operación suma que “vive” dentro de la curva.





Operaciones

La operación se puede escribir de forma explícita. Llamamos R = P + Q

Si $P \neq Q$:

Sea $s = (y_P - y_Q) / (x_P - x_Q)$

$x_R = s^2 - x_P - x_Q$

$y_R = y_P + s(x_R - x_P)$

Si $P = Q$:

Ahora $s = (3x_P^2 + a) / 2y_P$

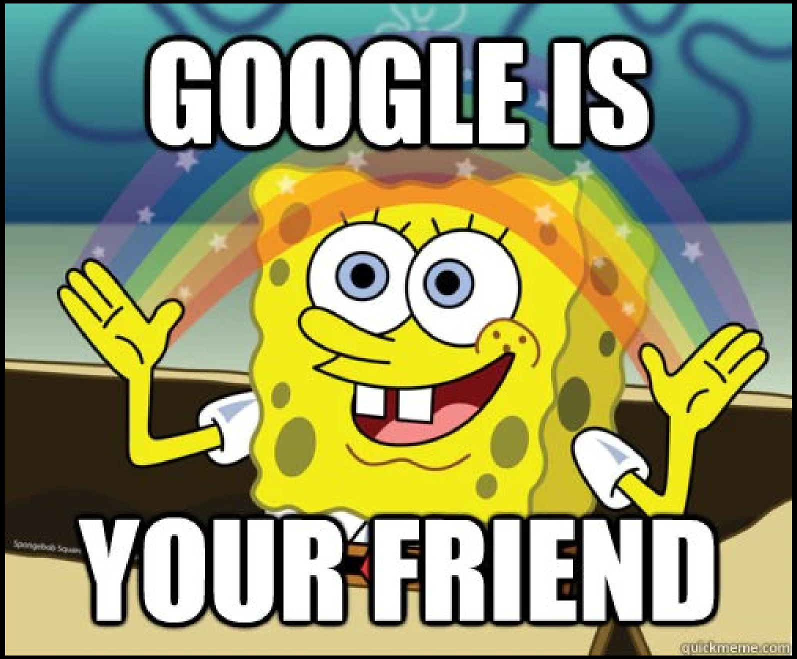
$x_R = s^2 - 2x_P$

$y_R = y_P + s(x_R - x_P)$

HACKMADRID %27

TRANQUILOS

No hay que aprendérselas. Están en Google.



ECDH

Muy parecido a Diffie-Hellman.

1. Alice y Bob acuerdan usar la misma curva y un punto base, G

2. Todo punto en la curva tiene orden finito. Digamos que el orden es N. Esto quiere decir que “N*G = 1”.

3. Ambos eligen un entero < N. a para Alice, b para Bob. Esta será su clave privada

4. Alice calcula $A = a*G$. Será su clave pública.

5. Bob calcula $B = b*G$. Será su clave pública.

ECDH

6. Alice calcula ahora el *secreto compartido*

   $S = a*B = a*b*G$

7. Bob hace lo mismo

   $S = b*A = b*a*G$

8. Ni Bob ni Alice conocen la clave privada de la otra persona, pero ambos pueden obtener un secreto en común.

9. El secreto *S* puede usarse para cifrar mensajes.