Introducción a la tecnología blockchain

Gustavo A. Arellano

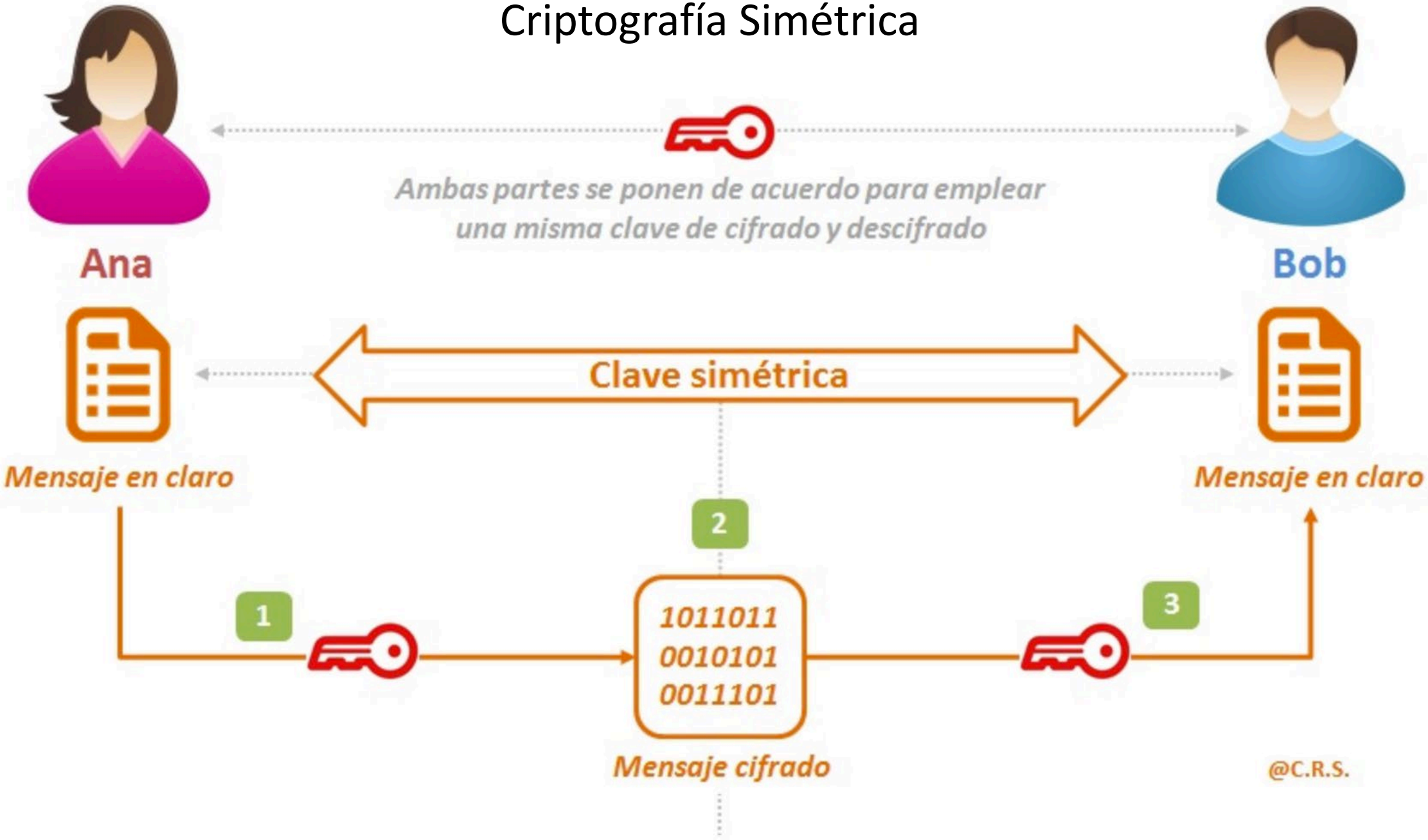
Abril, 2022

Twitter: @arellano_gus

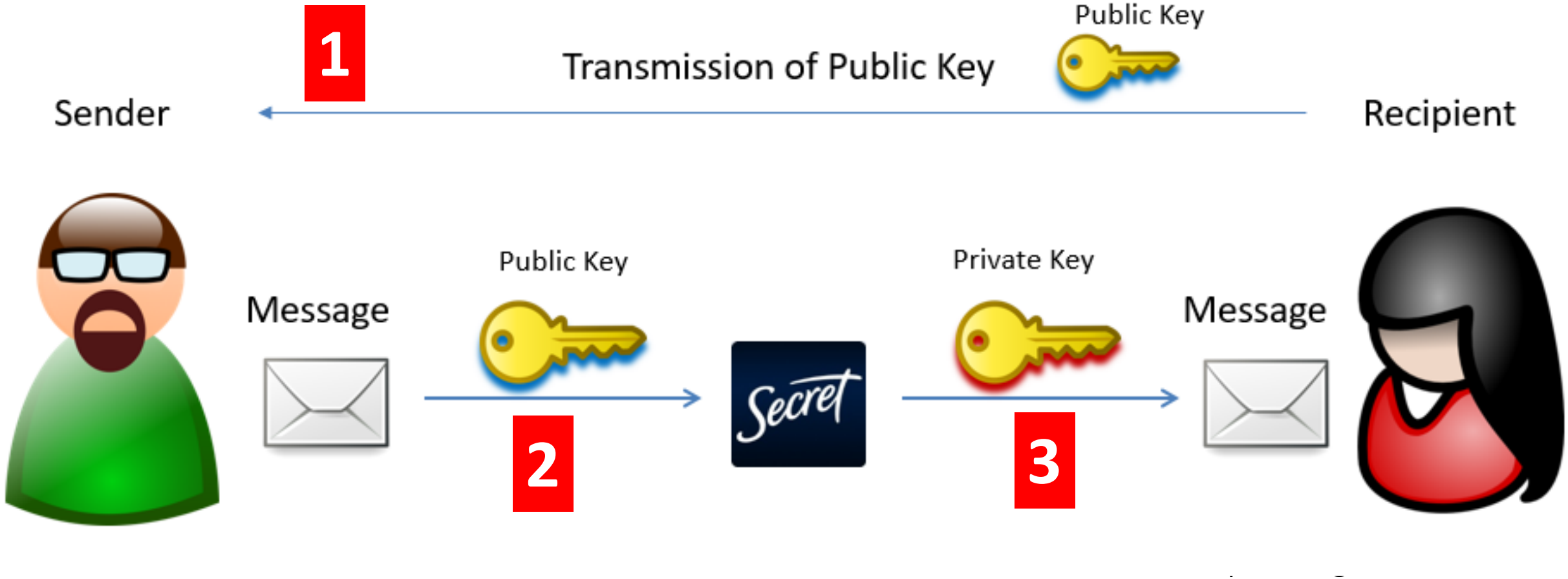
Introducción

- La Criptografía es una disciplina del área de la computación cuyo objetivo es alterar la estructura de un texto para impedir el acceso al texto original a receptores no autorizados y permitir el acceso de dicha información a receptores que demuestren su legítimo derecho de acceso.
- Los primeros intentos criptográficos se remiten a simples permutaciones de letras o a diccionarios de recolocación de símbolos.
- Posteriormente, se generaron algoritmos que permitían cifrar (o descifrar) el mensaje procesado mendiante una misma clave que sólo poseen el receptor y el emisor. Esto se conoce como “Criptografía Simétrica”.

Criptografía Simétrica



Criptografía Asimétrica



Algoritmo RSA

Supongamos que Bob quiere enviar a Alicia un mensaje secreto que solo ella pueda leer.

Alicia envía a Bob una caja con un candado abierto, del que solo Alicia tiene la llave. Bob recibe la caja, escribe el mensaje, lo pone en la caja y la cierra con su candado (ahora Bob no puede leer el mensaje). Bob envía la caja a Alicia y ella la abre con su llave. En este ejemplo, la caja con el candado es la «clave pública» de Alicia, y la llave del candado es su «clave privada».

Técnicamente, Bob envía a Alicia un «mensaje llano» M en forma de un número m menor que otro número n, mediante un protocolo reversible conocido como *padding scheme* («patrón de relleno»). A continuación genera el «mensaje cifrado» c mediante la siguiente operación:

$$c = m^e \pmod{n},$$

donde e es la clave pública de Alicia.

Ahora Alicia descifra el mensaje en clave c mediante la operación inversa dada por

$$m = c^d \pmod{n},$$

donde d es la clave privada que solo Alicia conoce.

Generación de claves [editar]

1. Se eligen dos números primos distintos p y q .
   - Por motivos de seguridad, estos números deben escogerse de forma aleatoria y deben tener una longitud en bits parecida. Se pueden hallar primos fácilmente mediante test de primalidad.
2. Se calcula $n = p \cdot q$.
   - n se usa como el módulo para ambas claves, pública y privada.
3. Con φ es la función φ de Euler calcula $\varphi(n) = (p-1) \cdot (q-1)$ aprovechando las dos propiedades de la función de Euler siguientes:
   - $\varphi(p) = p - 1$ si p es primo.
   - Si *m* y *n* son primos entre sí, entonces $\varphi(mn) = \varphi(m)\varphi(n)$.
4. Se escoge un entero positivo e menor que $\varphi(n)$, que sea coprimo con $\varphi(n)$.
   - e se da a conocer como el exponente de la clave pública.
   - Si se escoge un e con una suma encadenada corta, el cifrado será más efectivo. Un exponente e muy pequeño (p. ej. $e = 3$) podría suponer un riesgo para la seguridad.[2]
5. Se determina un d (mediante aritmética modular) que satisfaga la congruencia $e \cdot d \equiv 1 \pmod{\varphi(n)}$, es decir, que d sea el multiplicador modular inverso de $e \mod \varphi(n)$
   - Expresado de otra manera, $d \cdot e - 1$ es dividido exactamente por $\varphi(n) = (p-1) \cdot (q-1)$.
   - Esto suele calcularse mediante el algoritmo de Euclides extendido.
   - d se guarda como el exponente de la clave privada.

La **clave pública** es (n, e) , esto es, el módulo y el exponente de cifrado. La **clave privada** es (n, d) , esto es, el módulo y el exponente de descifrado, que debe mantenerse en secreto.

Usando las propiedades de la función de Euler, el Teorema de Euler y el Teorema del resto chino se puede demostrar que $x \equiv x^{ed} \pmod{n}$, $\forall x \in \mathbf{Z}_n$ [3] [4]

In a nutshell….

- Con la propuesta RSA, se genera un par de llaves (llave pública y llave privada) que están matemáticamente relacionadas entre si, pero en dónde es incosteable derivar la llave privada a partir de la llave pública en un tiempo razonable, ya que con el poder de cómputo del que se dispone actualmente, tomaría varios millones de años obtener la llave privada a partir de la pública.
- La idea subyace en la complejidad matemática para factorizar un número que es la multiplicación de dos primos.
- Además . . .

Private versus Public keys in RSA Crypto (Rivest Shamir Adleman –MIT, 1979-)

In RSA crypto, when you generate a key pair, it's completely arbitrary which one you choose to be the public key, and which is the private key. If you encrypt with one, you can decrypt with the other - it works in both directions.

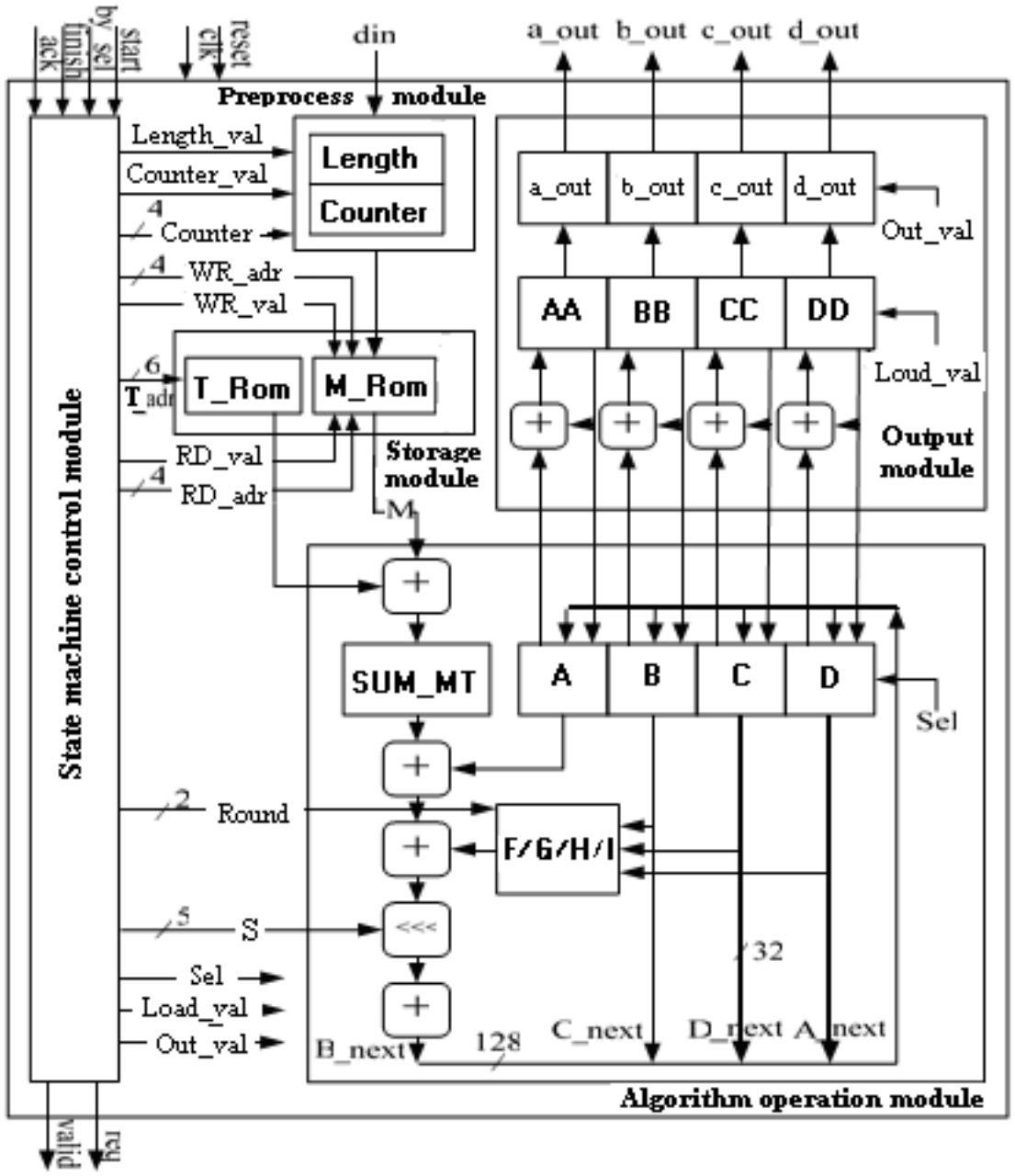
f(private, f(public, message)) === f(public, f(private, message)) === message

Parte II

- Algoritmos de digestión

Algoritmo MD5

- Los algoritmos de digestión toman como argumento una secuencia arbitraria de caracteres de cualquier tamaño y regresan una cadena hexadecimal de 32 caracteres que podría considerarse como la “huella digital” de la secuencia de caracteres original, dado que siempre generará la misma respuesta, si recibe el mismo argumento.



Características de los algoritmos de digestión

- Los algoritmos de digestión, como MD5, SHA-1 y SHA-256 (por ejemplo) son algoritmos de “sólo ida”, ya que una vez que procesan un argumento, no es posible reconstituir el argumento a partir del resultado, pero garantizan que dos argumentos idénticos siempre generan idénticos resultados.
- Evidentemente, pueden existir dos cadenas resultantes idénticas, a partir de dos argumentos distintos, pero es extremadamente complejo generar intencionalmente dos secuencias distintas de caracteres que arrojen el mismo MD5 y aún más difícil el mismo SHA256. A esto se le denomina “colisiones”.
- Google tardó 20 años en encontrar una manera de generar colisiones en SHA-1: https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

Ejemplos de MD5

Hola =
`8c8432c5523c8507a5ec3b1ae3ab364f`
hola =
`916f4c31aaa35d6b867dae9a7f54270d`
Adios =
`4764b0846460fe8fb4e4d38756ada998`
aDios =
`675ac99891373cfb42897d3705153f37`
Parangaricutirimicuaro =
`3729b3e1ce3b8e20809328c3a8c68822`

Parte III Generación de un UID

- Sea F = {1,2,…0, A, B…Z, a, b, …z} un conjunto de símbolos
- F tiene 64 símbolos
- El número total de cadenas distintas de longitud N en donde cada posición puede ser cualquier símbolo de F es 64^N.
- Para N razonablemente grande, digamos 16, el número total de cadenas distintas que se pueden formar empleando los símbolos de F es de: 64^16 = (2^6)^16=2^96
- De lo anterior, la posibilidad de generar dos cadenas identicas de longitud 16 en un proceso aleatorio es muy baja.

Ejemplo

tarea.docx

+

Fred public key

1

Mensaje encriptado

Este mensaje sólo lo puede desencriptar Fred con su llave privada

+

My private key

2

Mensaje Firmado

Todo mundo puede verificar (con mi llave públca) que este mensaje sólo pudo ser enviado por mí.

Aplicaciones en el mundo real

- Crypto mercados (**Tecnología Blockchain**)
- Protocolos de comunicación (SSL, por ejemplo)
- Acceso a servidores con llave privada (AWS, por ejemplo)
- Firmas en cuestiones hacendarias (FIEL del SAT, por ejemplo)
- Seguridad y no repudio en correos electrónicos

¿QUÉ TAN FÁCIL (O DIFÍCIL) ES QUE SE FILTREN TUS CONVERSACIONES DEL WHATS*?

MUY MUY DÍFICIL. ASÍ ES COMO FUNCIONA:

¿Y SI LO INTERCEPTAN?

COMO SOLO **EL QUE ENVÍA TIENE LA LLAVE PARA FIRMAR**, NADIE MÁS PUEDE MANDAR MENSAJES HACIÉNDOSE PASAR POR ÉL...

Y COMO SOLO **EL QUE RECIBE TIENE LA LLAVE PARA DECRIPTAR**, NADIE MÁS (NI SIQUIERA WHATSAPP) PUEDE LEER EL MENSAJE.

*ACTUALMENTE WHATSAPP SE HA ACTUALIZADO PARA QUE ESTA ENCRIPTACIÓN FIN-A-FIN APLIQUE TAMBIÉN A LLAMADAS, GRUPOS Y ENVÍO DE IMÁGENES.

FUENTE: PANAYOTIS VRYONIS / BLOG.VRYPAN.NET

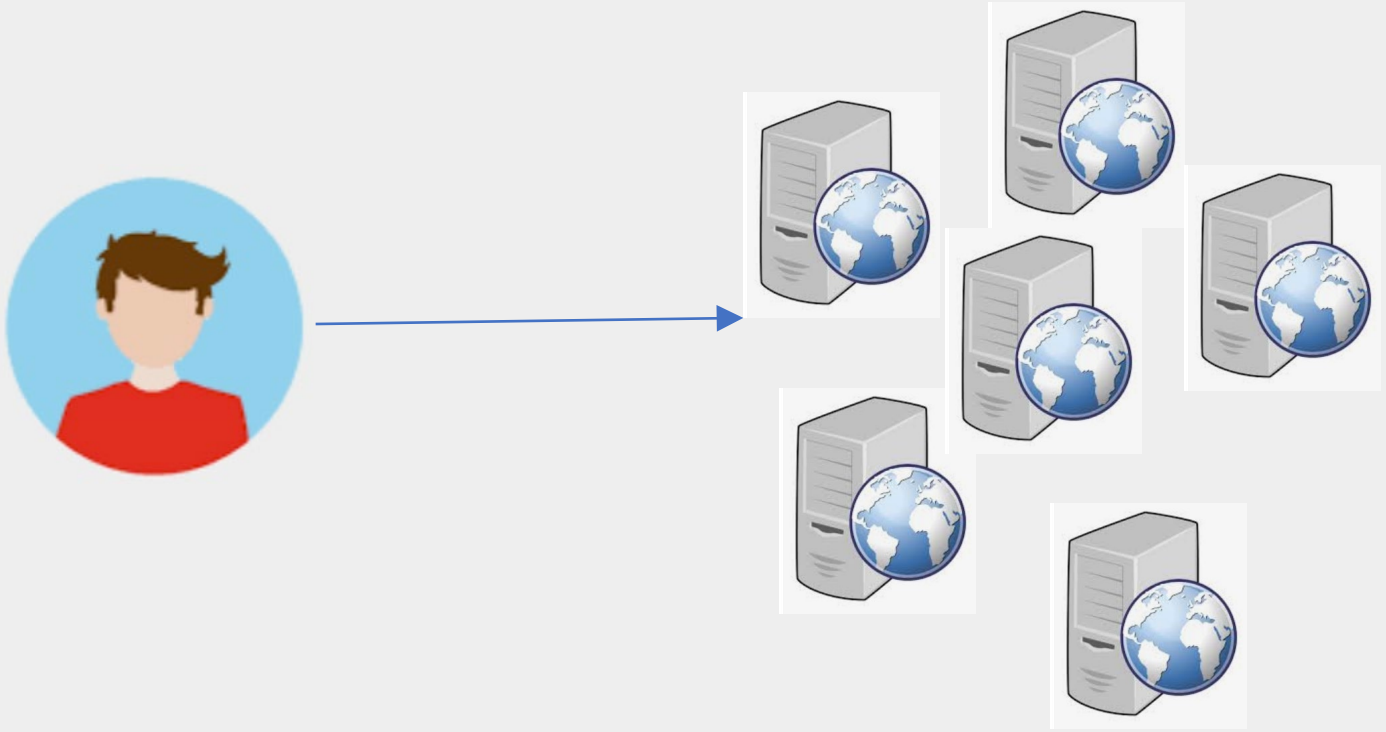
PICTOLINE.COM

Parte IV

- Transacciones centralizadas y transacciones descentralizadas

Esquema centralizado

Esquema descentralizado



Wallet BTC


English | Español | Français | ελληνικά | italiano | Deutsch
Česky | Magyar | 日本語 | 简体中文 | Русский | português
bitaddress.org
Generador de carteras Bitcoin de código abierto en lado de cliente con Javascript
20% 20% 20% Cartera mental
20% 20% Detalles de la cartera
Generando dirección Bitcoin...
Mueve un poco el ratón para crear entropía... 20%
OR type some random characters into this textbox
7b0d1dab1ed88951b6f17b3c22c0da9c246daad7c817e093eb837b74ad9a26d58
883706cfe65cc3e598627876fa648d3dc6083fb9a1e49891bf1dde1c4863646fb
cc14e3cee595acc1ee6b8c4253d651e4362443e983b27313877b7b528287af2cd
951717fcc5eed75905609a545889f5e1a441fb90a8074970c1185b7e924d1cd78
8748c881240668ab8bbfd53db430435c6750575c70498363763cc40a6cba655d0
3df9e99c4687539b651366d730afe18c4c68691f891536df51f7aa655899bfd68
30cbfc45768958acdad3b7629d1536f2187cba34e0daae6e60cde69b565a4a56d
054f55f385344eb932b940d4ead0d7092db8254cb42e85abe0ffdf1fa

English | Español | Français | ελληνικά | italiano | Deutsch
Česky | Magyar | 日本語 | 简体中文 | Русский | português

bitaddress.org

Open Source JavaScript Client-Side Bitcoin Wallet Generator

Single Wallet | Paper Wallet | Bulk Wallet | Brain Wallet
Vanity Wallet | Split Wallet | Wallet Details

Generate New Address | Print

Bitcoin Address

Private Key

SHARE

SECRET

1PvGLZBKf3ozQWnG67DDtpurM9smjihxUG

KxAPN1gJ3teGn5NzH5SUDDCXC8HjuHNpyXdamHH5oo9RKbJcuQLu

A Bitcoin wallet is as simple as a single pairing of a Bitcoin address with its corresponding Bitcoin private key. Such a wallet has been generated for you in your web browser and is displayed above.

To safeguard this wallet you must print or otherwise record the Bitcoin address and private key. It is important to make a backup copy of the private key and store it in a safe location. This site does not have knowledge of your private key. If you are familiar with PGP you can download this all-in-one HTML page and check that you have an authentic version from the

Parte V

- Base de datos de transacciones
- Nodos
- Prueba de Trabajo PoW
- Mineros, Recompensas y Halvings
- Exchanges

G R A C I A S

Abril, 2022

Gustavo A. Arellano

Twitter: @arellano_gus