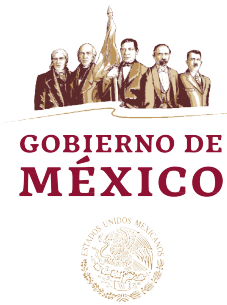
INTRODUCCIÓN A LA CRIPTOGRAFÍA

Escuela de Verano

[2019]

Luis J. Dominguez Perez



Lecture 2

Contenido, sección 1

Funciones de un solo sentido

Criptografia Pública

Ejemplos de protocolos

Función de un solo sentido

CIMAT

Definición

Una función $f()$ es una función de un solo sentido si:

- $y = f(x)$ es computacionalmente sencilla, y
- $x = f^{-1}(y)$ es computacionalmente impráctica.

Ejemplos de funciones de un solo sentido

- Logaritmo discreto
  - Dados x, a, y n, es fácil calcular $y = x^a \bmod n$; sin embargo, dados y, x, y n, encontrar a es muy difícil
- Factorización
  - Dados x y y, es fácil calcular $n = xy$; sin embargo, dada n, encontrar los factores x y y es muy difícil
- Raíz cuadrada discreta
  - Dados x y n, es fácil calcular $a = x^2 \bmod n$; sin embargo, dados a y n, encontrar x es muy difícil.

Contenido, sección 2

Funciones de un solo sentido

Criptografia Pública

Ejemplos de protocolos

El nacimiento de la criptografía de llave pública

CIMAT

- En 1976, Whitfield Diffie y Martin Hellman publicaron su famoso artículo: “Nuevas direcciones en criptografía” (New Directions in Cryptography)
- Poco antes, Ralph Merkle inventó una construcción de llave pública para sus clases. Su trabajo se tituló: “Comunicación segura sobre canales inseguros” en 1982 (Secure communication over insecure channels)

El nacimiento de la criptografía de llave pública

- El concepto fue originalmente descubierto por James Ellis, aunque se mantuvo en secreto ya que era infomación clasificada de la GCHQ de 1969 a 1997.
- Adicionalmente, Malcolm Williamson y Clifford Cocks de la GCHQ, descubrieron el intercambio de llave Diffie-Hellman, y el cifrado RSA.

El nacimiento

- Antes de la publicación de “New Directions. . .”, la investigación sobre cifrado en los E.E.U.U. era dominio de la Agencia Nacional de Seguridad (NSA).
- Hasta mediados de 1990’s, la exportación de algoritmos criptográficos era penada con traición.
- Después, solamente prohibía la exportación de algoritmos de seguridad alta si eran leíbles por máquinas (código fuente, ejecutables, etc.).

Curiosidades

- Los algoritmos critográficos se conviertieron en “municiones” para el gobierno
- La gente se hacía playeras con código RSA amenzando con salir del país
- Hubo quien incluso se hizo tatuajes

Curiosidades

- Los algoritmos critográficos se conviertieron en “municiones” para el gobierno
- La gente se hacía playeras con código RSA amenzando con salir del país
- Hubo quien incluso se hizo tatuajes

- Las penas por viajar de los E.E.U.U. a Europa de esta manera llegaban a los 10 años de prisión.
- Ahora es posible comprar implementaciones seguras

Criptografía Pública

- El término Criptografía de Clave Pública (o criptografía pública) es intercambiable con el de Criptografía Asimétrica, denotan lo mismo, y son sinónimos.
- Aunque fue hecha pública en 1976 por Diffie, Hellman, and Merkel, en 1997 se desclasificaron documentos británicos de 1972, en donde James Ellis, Clifford Cocks and Graham Williamson del Government Communications Headquarters (GCHQ, Reino Unido), descubrieron la criptografía pública.

Criptografía simétrica vs. asimétrica

Simétrica

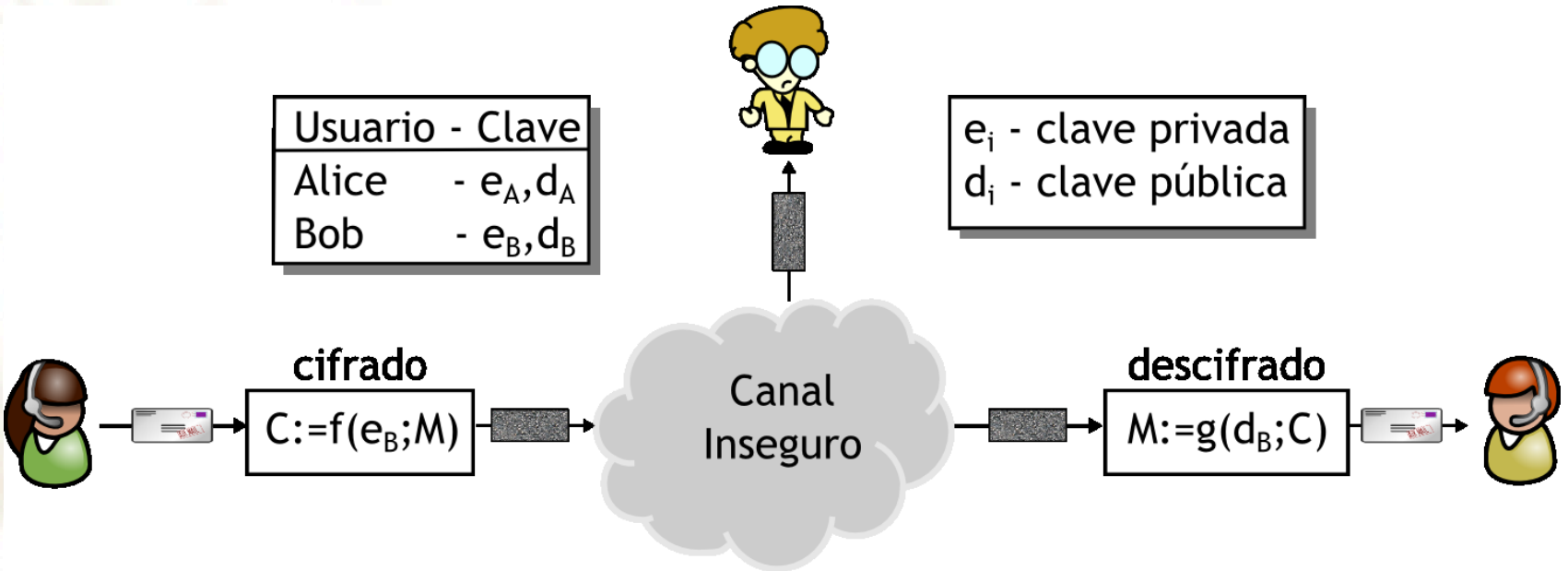
- La misma llave secreta se utiliza para cifrar y descifrar
- La función de cifrado y descifrado son muy parecidas, la misma en algunos casos
- Distribución de llaves
- Número de llaves
- Colusión

Criptografía simétrica vs. asimétrica

Asimétrica

- Sólo se publica la parte pública de la llave, cada quien la baja sobredemanda
- Que haya muchas claves es irrelevante
- Se puede proteger de la colusión
- Ahora se tiene un par de llaves: pública y privada
- La función de cifrado y descifrado son diferentes

Esquema General de la Criptografía Pública



Usos de la criptografía de clave pública

Además de cifrar datos, la criptografía de clave pública puede utilizarse en lo siguiente:

- **Establecimiento de llaves**. Existen protocolos para la compartición de claves secretas sobre medios inseguros (para utilizar criptografía simétrica).
- **No repudiación**. Proveen no repudiación e integridad de mensajes mediante firmas digitales: RSA, DSA, ECDSA
- **Identificación**. Mediante mecanismos de desafío-respuesta junto con firmas digitales: para smart cards y teléfonos móviles.
- **Cifrado**.

El único pendiente es la autenticación de las claves

Algoritmos relevantes

Los tres tipos relevantes en la criptografía de clave pública son:

- **Esquemas de factorización de enteros**. Basados en la dificultad de factorizar números enteros grandes
- **Esquemas de Logaritmos Discretos**. Basados en la dificultad del problema del logaritmo discreto sobre campos finitos
- **Esquemas de Curvas Elípticas**. Generalización del problema anterior a esquemas basados en curvas elípticas

Nivel de seguridad

Dado que los ataques a los sistemas criptográficos son más eficientes en los esquemas asimétricos, se define como nivel de seguridad equivalente: el número de bits en un criptosistema asimétrico que equivalen en resistencia a uno simétrico.

Esto es: esto es, si rompemos una clave simétrica de 80 bits en 1 segundo con 2^{80} computadoras, ¿cuántos bits debe de tener una clave asimétrica para que son 2^{80} computadoras también nos tardemos 1 segundo?

Niveles de seguridad

Familia	Criptosistema	Nivel de seguridad		
		128	192	256
Factorización entera	RSA	3072 bit	7680 bit	15360 bit
Logaritmo discreto	DH, DSA, Elgamal	3072 bit	7680 bit	15360 bit
Curvas elípticas	ECDH, ECDSA	256 bit	384 bit	512 bit
Clave simétrica		128 bit	192 bit	256 bit

Contenido, sección 3

Funciones de un solo sentido

Criptografia Pública

Ejemplos de protocolos

Servicios Básicos de seguridad

Los servicios más importantes son:

- Confidencialidad
- Integridad
- Autenticación de mensajes
- No repudiación

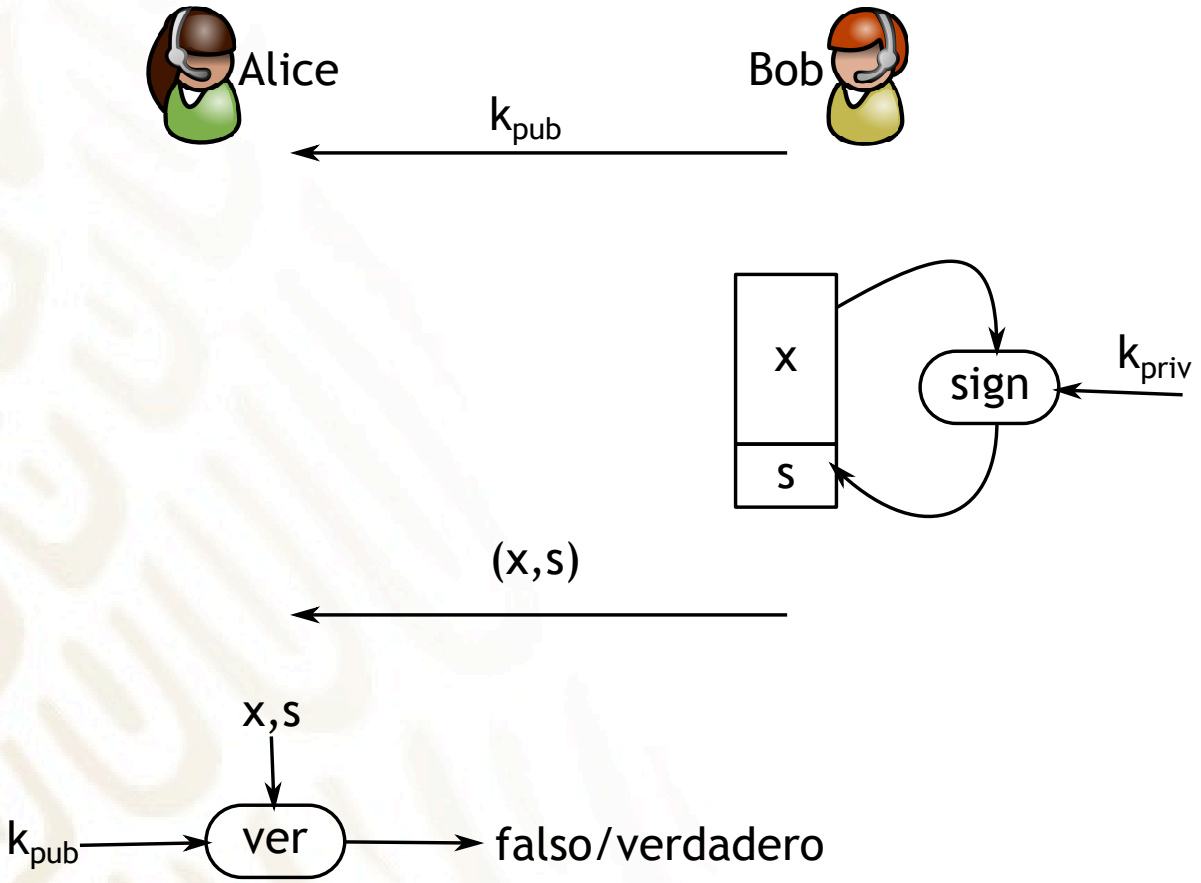
Otros servicios de seguridad

Existen otros servicios opcionales que dependen de la aplicación:

- Identificación o autenticación de entidades
- Control de acceso
- Disponibilidad
- Auditoría
- Seguridad física
- Anonimato

Esquema

Esto es posible mediante criptografía de clave pública. El signatario firma utilizando su clave privada, el receptor utiliza la clave pública para verificar.



Diffie–Hellman key exchange (DHKE)

- La idea básica detrás del DHKE es que la exponenciación en $\mathbb{Z}_p^\star$, con p primo, es una función de un solo sentido, y que la exponenciación es conmutativa:

$$x \equiv (\alpha^x)^y \equiv (\alpha^y)^x \bmod p$$

Diagrama DHKE

Alice		*Bob*
	Dados p y α	
$a \in_R \mathbb{Z}_p^\star$		$b \in_R \mathbb{Z}_p^\star$
$A_{\text{priv}} = a$		$B_{\text{priv}} = b$
$A_{\text{pub}} \equiv \alpha^a \bmod p$		$B_{\text{pub}} \equiv \alpha^b \bmod p$
	$A_{\text{pub}} \longrightarrow$	
	$\longleftarrow B_{\text{pub}}$	
$k_{AB} \equiv (B_{pub})^a \bmod p$		$k_{AB} \equiv (A_{pub})^b \bmod p$

Ejercicio

Pruebe en Python el intercambio de llaves DHKE:

- $p = 29$
- $\alpha = 2$

RSA

Diseñado en 1977 por Ron Rivest, Adi Shamir, y Leonar Adleman.

- Sean p y q dos diferentes grandes números primos aleatorios
- El módulo n es el producto de p y q
- La función $\Phi(n) = (p-1)(q-1)$
- Seleccionamos $1 < e < \Phi(n)$, tal que el $\text{MCD}(e, \Phi(n)) = 1$; $e = 2^{16} + 1$ típicamente
- Se calcula $d \equiv e^{-1} \bmod \Phi(n)$

La clave pública es e, y n. La clave privada es d, y los primos p y q.

Cifrado y descifrado RSA

Dado un mensaje $M < n$

- **Cifrado**. $C = M^e \bmod n$
- **Descifrado**. $M = C^d \bmod n$

Para mensajes más largos, se utiliza un modo de operación, como los vistos anteriormente.

Ejemplo

- $p = 11, q = 13$
- $n = p \cdot q = 11 \cdot 13 = 143$
- $\Phi(n) = (p-1)(q-1) = 10 \cdot 12 = 120$
- $\text{MCD}(e, \Phi(n)) = \text{MCD}(e, 120) = 1; e = 17$
- $d = e^{-1} \bmod \Phi(n) = 17^{-1} \bmod 120 = 113$

- Clave pública $= (e, n) = (17, 143)$
- Clave privada $= (d, p, q) = (113, 11, 23)$

... ejemplo

- Mensaje $M = 50$
- **Cifrado:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$
- **Descifrado:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

... ejemplo

- Mensaje $M = 50$
- **Cifrado:**
  $C = M^e \bmod n = 50^{17} \bmod 143 = 85$
- **Descifrado:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

parece sencillo, sin embargo, observe el 85^{113}, ¿qué pasaría con números grandes? Recuerde el tamaño en la tabla de Niveles de Seguridad.

Cifrado Elgamal

Alice		*Bob*
		p, α
		$b \in_R \mathbb{Z}_p^\star$
		$\beta = \alpha^b$
	$\xleftarrow{p,\alpha,\beta}$	
$a \in_R \mathbb{Z}_p^\star$		
$k_E = \alpha^a \bmod p$		
$k_M = \beta^a \bmod p$		
$x \in \mathbb{Z}_p^\star$		
$y \equiv x \cdot k_M \bmod p$		
	$\xrightarrow{k_E,y}$	
		$k_M = (k_E)^b \bmod p$
		$x \equiv (k_M)^{-1} \bmod p$

Utilización de SHA con PyCrypto

CIMAT

Hash

```
1 from Crypto.Hash import SHA256
2 h = SHA256.new()
3 h.update(b'Hello')
4 print h.hexdigest()
```

Code taken from PyCrypto Documentation.

Utilización de RSA con PyCrypto

CIMAT

Generación

```
1 from Crypto.PublicKey import RSA
2 key = RSA.generate(2048)
3 f = open('mykey.pem','w')
4 f.write(RSA.exportKey('PEM'))
5 f.close()
6
7 f = open('mykey.pem','r')
8 key = RSA.importKey(f.read())
```

Code taken from PyCrypto Documentation.

Utilización de RSA con PyCrypto

CIMAT

Cifrado

```
1 from Crypto.Cipher import PKCS1_OAEP
2 from Crypto.PublicKey import RSA
3
4 message = 'To be encrypted'
5 key = RSA.importKey(open('pubkey.der').
     read())
6 cipher = PKCS1_OAEP.new(key)
7 ciphertext = cipher.encrypt(message)
```

Code taken from PyCrypto Documentation.

Utilización de RSA con PyCrypto

CIMAT

Descifrado

```
1 key = RSA.importKey(open('privkey.der').
      read())
2 cipher = PKCS1_OAP.new(key)
3 message = cipher.decrypt(ciphertext)
```

Code taken from PyCrypto Documentation.