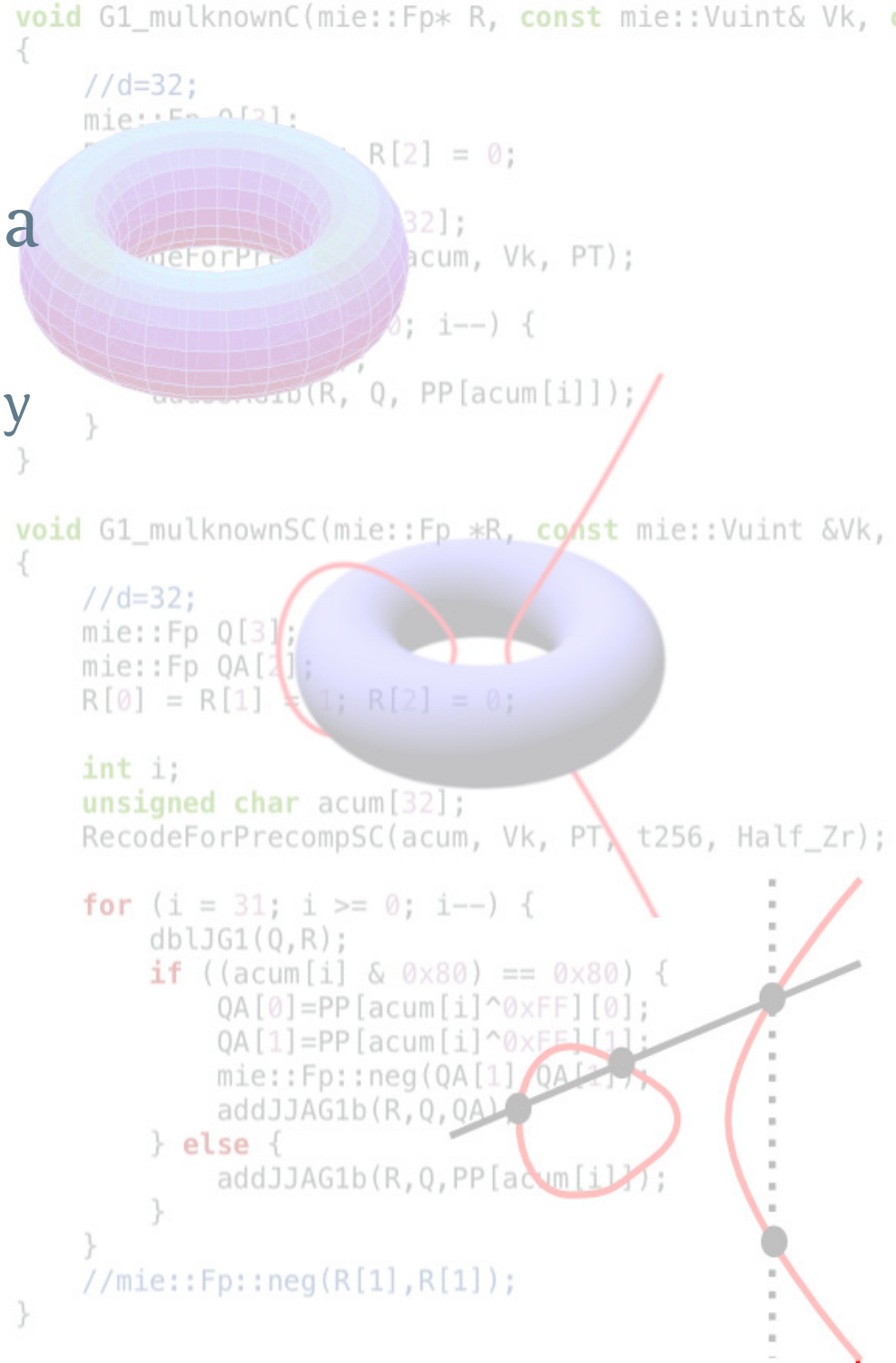
Matemáticas aplicadas a la criptografía

Unidad IV - Logaritmos discretos y Diffie-Hellman

Dr. Luis J. Dominguez Perez

Universidad Don Bosco

Abril 25, 2013



Contenido de la sección 1

El nacimiento de la criptografía de llave pública

- 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

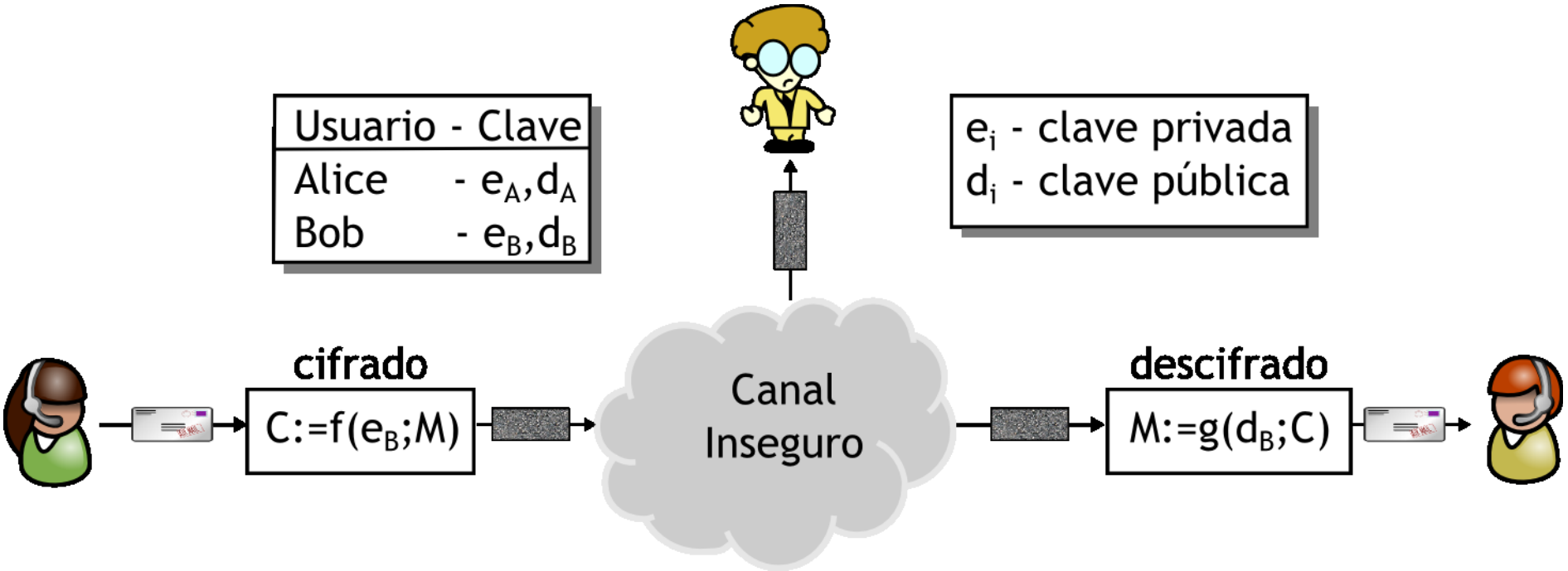
- Los algoritmos critográficos se conviertieron en ``municiones'' para el gobierno
- La gente se hacía playeras con código RSA amenzando con salir del país
- Hubo quien incluso se hizo tatuajes
- Las penas por viajar de los E.E.U.U. a Europa de esta manera llegaban a los 10 años de prisión.
- Ahora es posible comprar implementaciones seguras

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

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 públicas; para esto, se utilizan los certificados digitales.

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 de la sección 2

Función de un solo sentido

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.

Di e–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	
$a \in_R \mathbb{Z}_p^\star$		$b \in_R \mathbb{Z}_p^\star$
$A_{\text{priv}} = a$		$B_{\text{priv}} = b$
$A_{\text{pub}} \equiv \alpha^a \bmod p$		$B_{\text{pub}} \equiv \alpha^b \bmod p$
	$A_{\text{pub}} \longrightarrow$	
	$\longleftarrow B_{\text{pub}}$	
$k_{AB} \equiv (B_{pub})^a \bmod p$		$k_{AB} \equiv (A_{pub})^b \bmod p$

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 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:**
  $M = C^d \bmod n = 85^{113} \bmod 143 = 50$

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

... 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.

ElGamal

- El cifrado Elgamal fue propuesto por Taher Elgamal en 1985.
- Es una extensión del intercambio de llaves de Diffie-Hellman (DHKE)

Cifrado Elgamal

Alice

Bob

p, α

$b \in_R \mathbb{Z}_p^\star$

$\beta = \alpha^b$

$\xleftarrow{p,\alpha,\beta}$

$a \in_R \mathbb{Z}_p^\star$

$k_E = \alpha^a \bmod p$

$k_M = \beta^a \bmod p$

$x \in \mathbb{Z}_p^\star$

$y \equiv x \cdot k_M \bmod p$

$\xrightarrow{k_E,y}$

$k_M = (k_E)^b \bmod p$

$x \equiv (k_M)^{-1} \bmod p$

La llave es **efímera**, se tiene que generar en cada transmisión.

Contenido de la sección 3

Servicios Básicos de seguridad

Los servicios más importantes son:

- Confidencialidad
- Integridad
- Autenticación de mensajes
- No repudiación

Otros servicios de seguridad

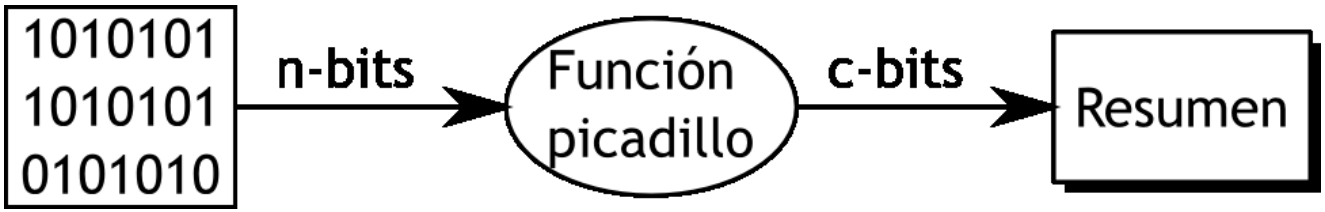
Existen otros servicios opcionales que dependen de la aplicación:

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

Funciones Picadillo

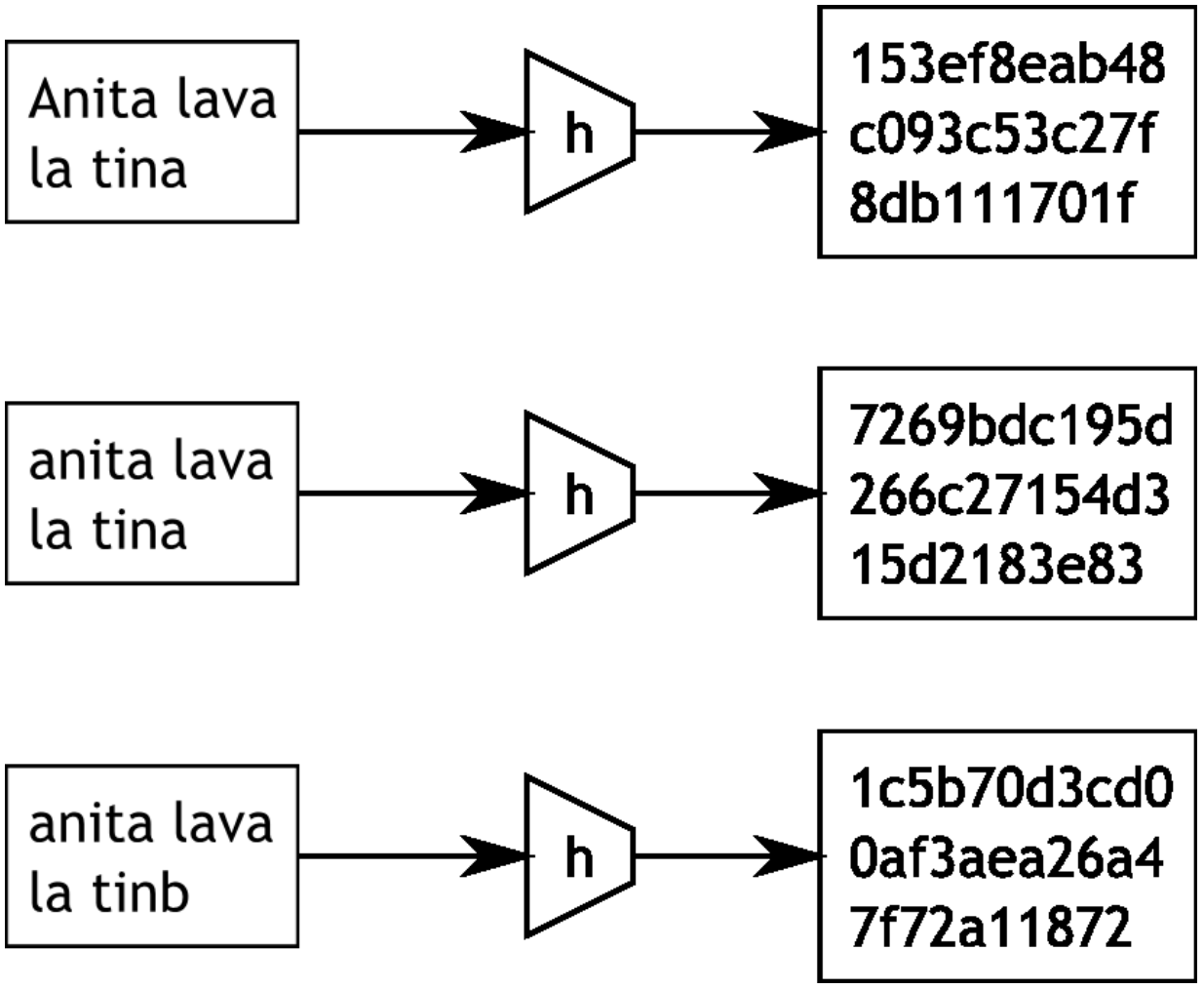
- Producen huellas digitales de longitud fija a partir de documentos de longitud arbitraria.
- Una pequeña variación en el texto original se ve reflejada en una huella digital totalmente diferente
  - Convierten contraseñas a salidas de longitud fija
  - Se utilizan para generar números aleatorios
  - Proveen autenticación básica a través de los MAC (Message autentication code)
  - Bloques básicos para firmas digitales.

Función picadillo - caja negra



La función picadillo recibe un texto de tamaño indefinido (normalmente se redondea el tamaño con ceros), y la salida es de un tamaño fijo.

Función picadillo - caja negra 2



Una función picadillo útil en criptografía da resúmenes totalmente diferentes incluso ante cambios menores.

Requerimientos de seguridad de las funciones picadillo

Los requerimientos de seguridad para el uso de las funciones en criptografía son los siguientes:

- Resistencia de preimagen (un solo sentido)
- Resistencia de segunda preimagen (débil resistencia a colisiones)
- Resistencia a colisiones (fuerte resistencia a colisiones)

Requerimientos de seguridad de las funciones picadillo - 2

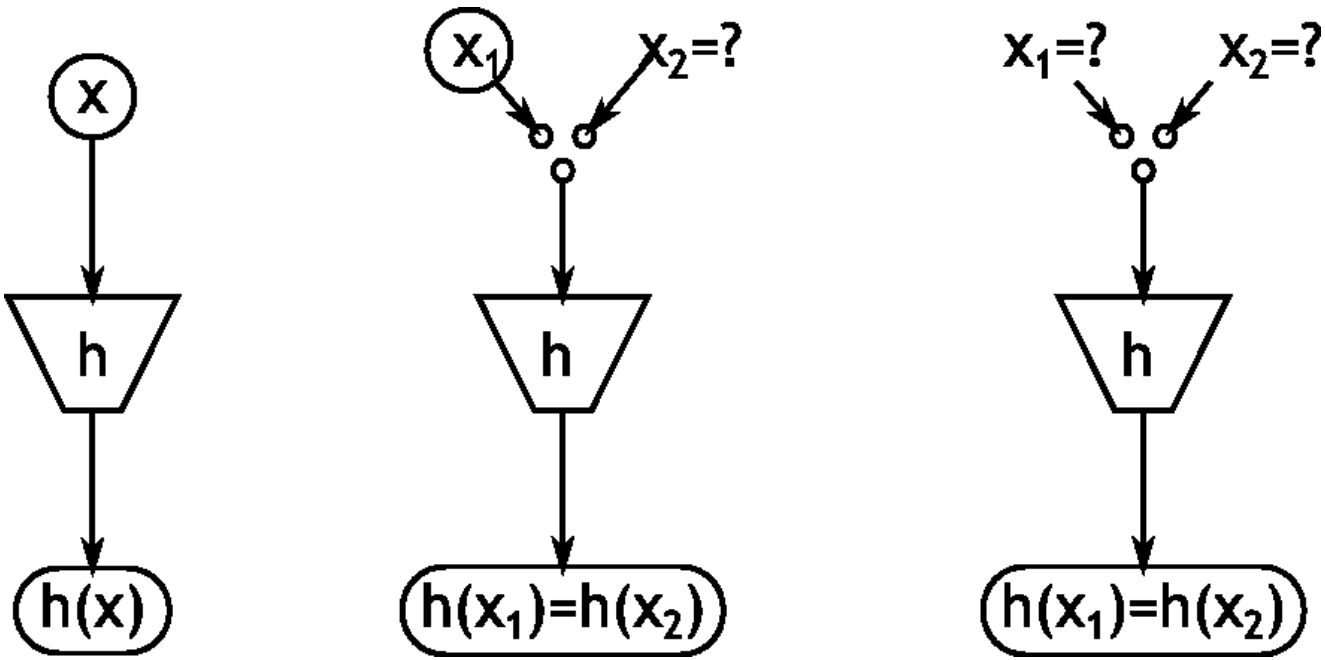
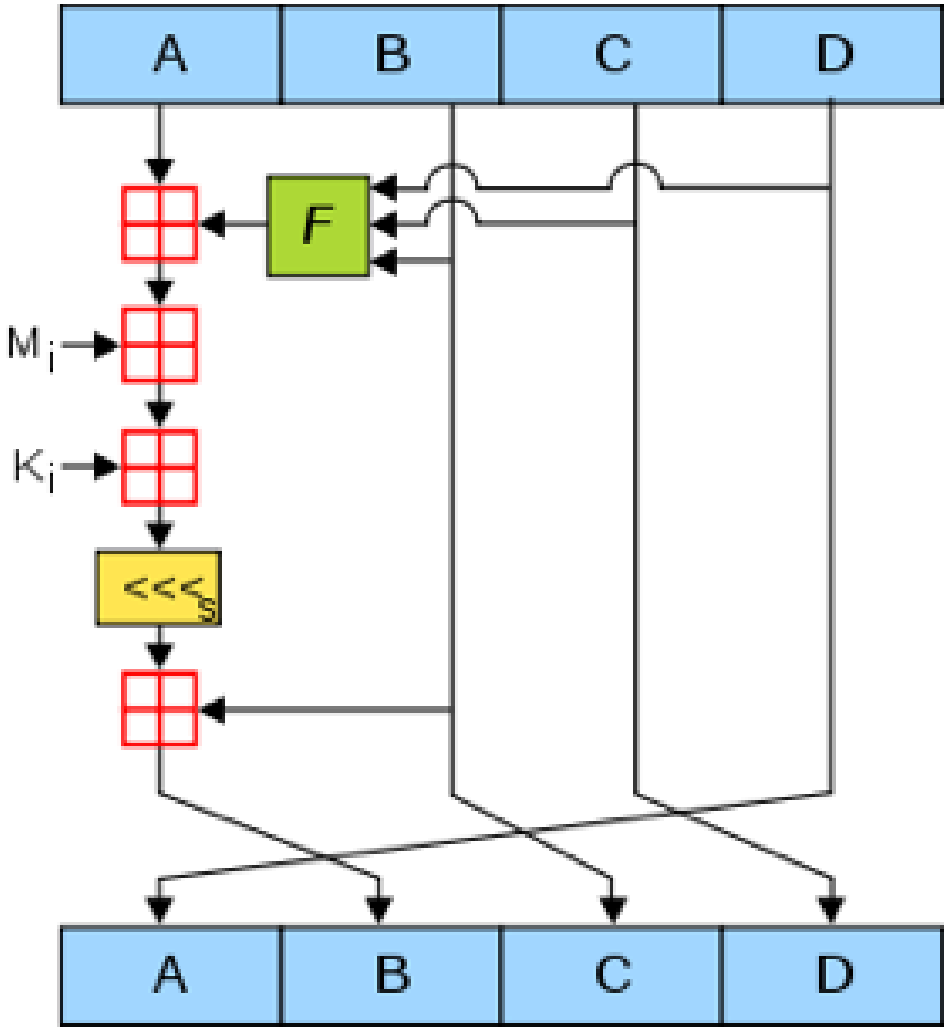


Diagrama MD5



Familias de funciones picadillo

Algoritmo		Salida	Entrada	Rondas	Colisiones
MD5		128	512	64	Sí
SHA-1		160	512	80	Aún no
SHA-2	**SHA-224**	224	512	64	No
	SHA-256	256	512	64	No
	SHA-384	384	1024	80	No
	SHA-512	512	1024	80	No

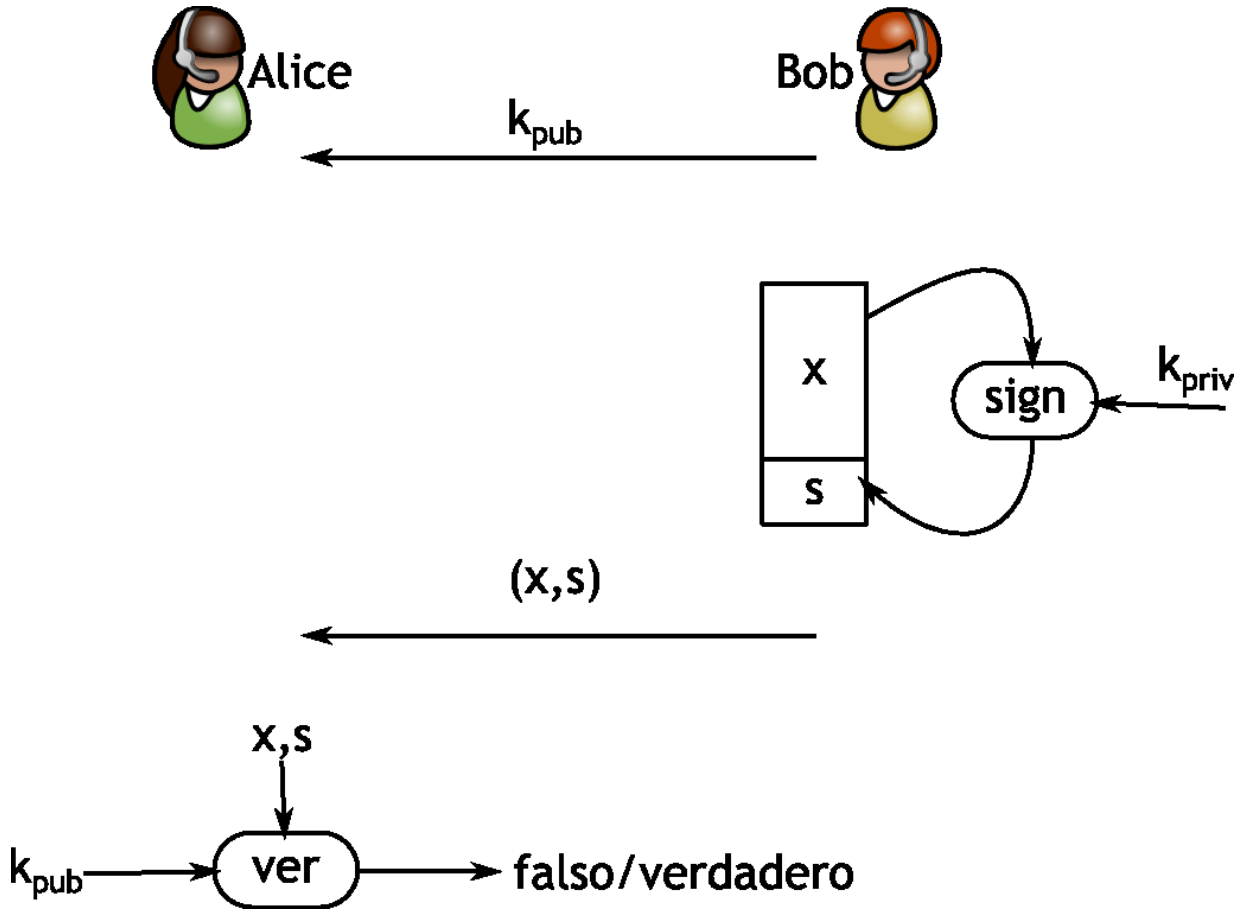
Recientemente, la función picadillo Keccak ganó el concurso para ser **SHA-3**; es una función *esponja*. Esta es la nueva recomendación del NIST.

Firmas digitales

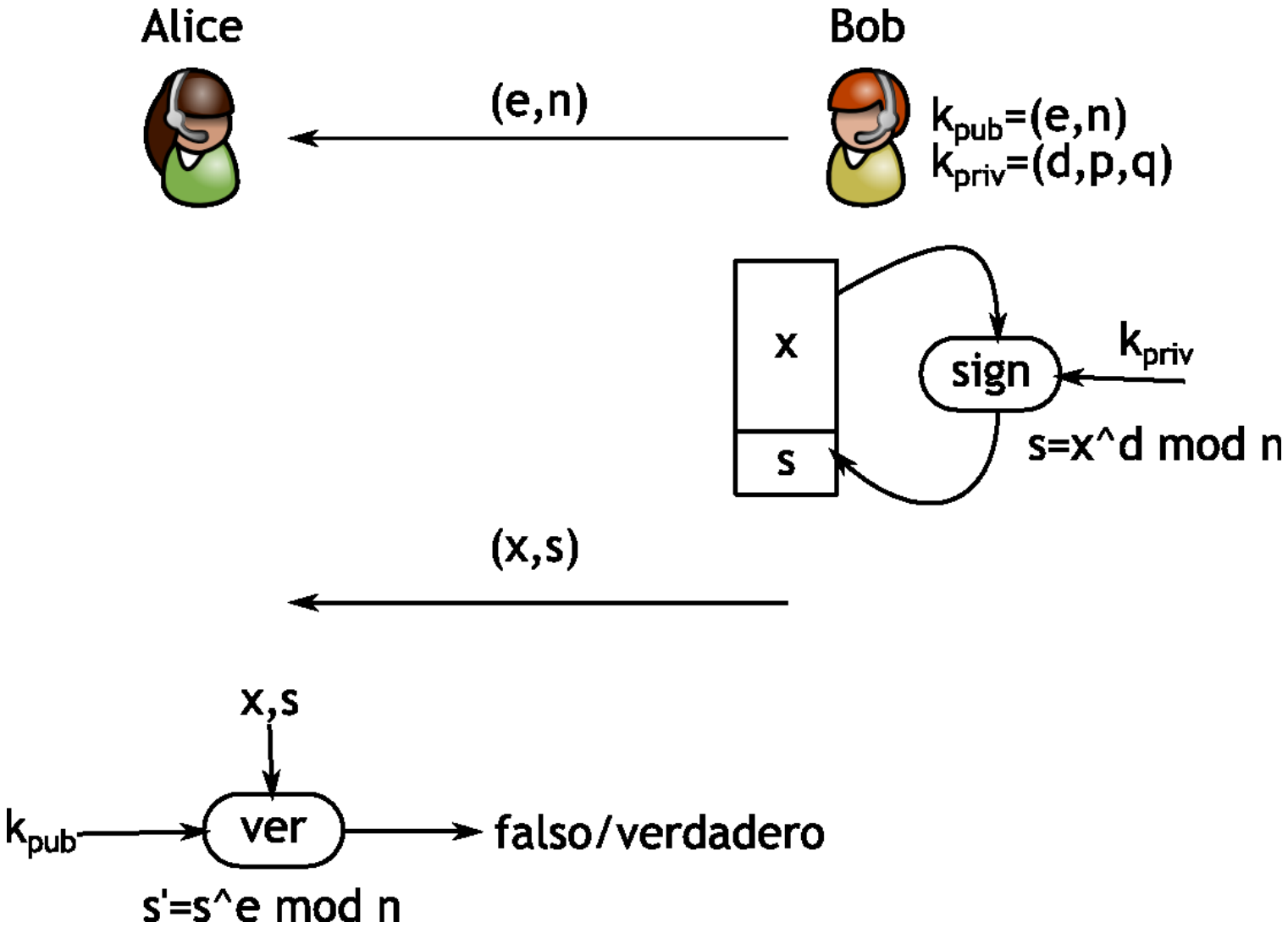
- La propiedad de demostrar que cierta persona generó un mensaje es crítica en muchas aplicaciones.
- En el mundo ``analógico'', se utilizan firmas a mano sobre papel.
- Sólo la persona que crea la firma, puede reproducirla.

Esquema

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

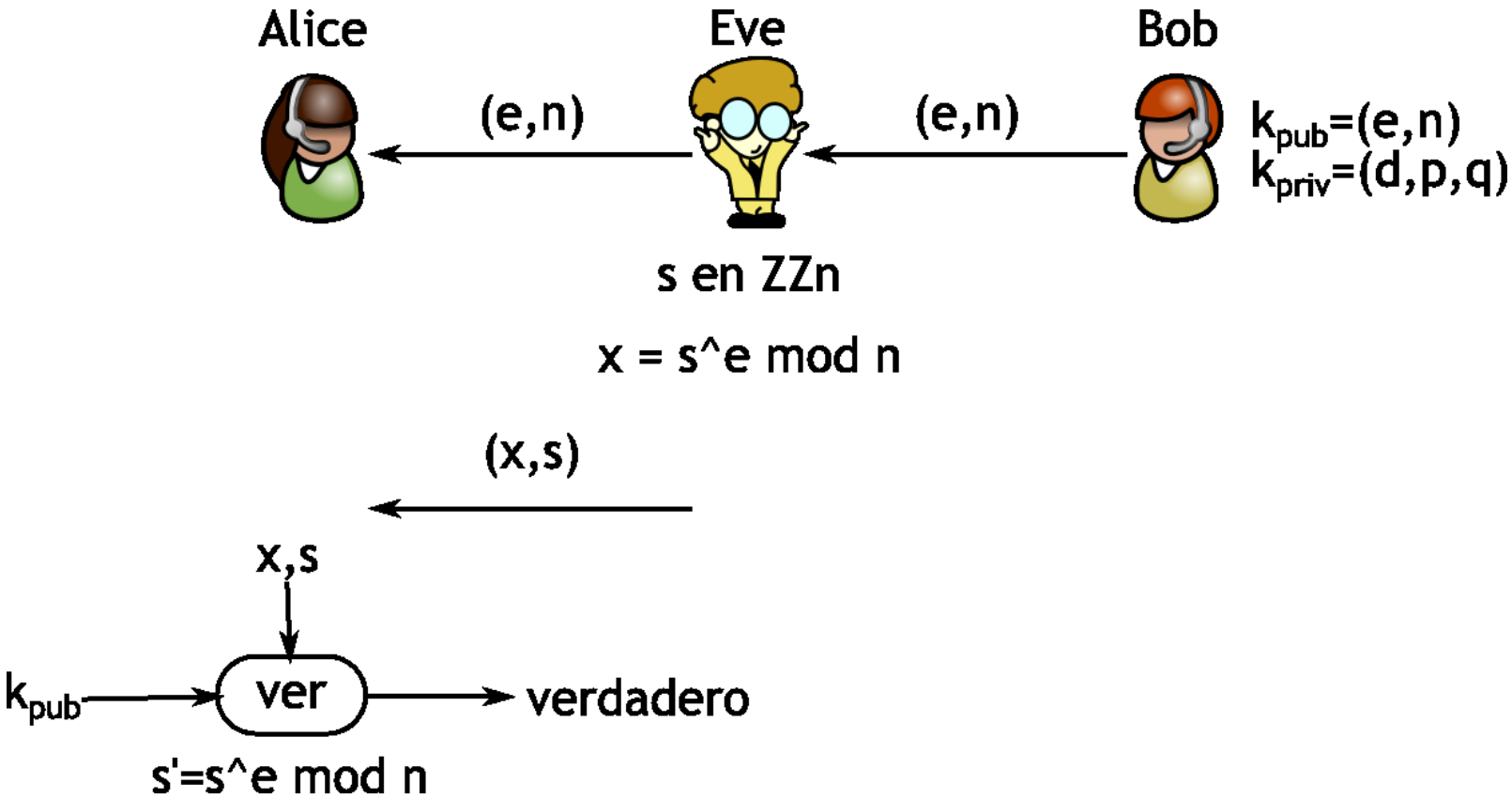


Firma RSA básica



Ataque a Firma RSA

Sobre validación de firmas ficticias



Formalizando un protocolo

- Los protocolos de seguridad contienen generalmente varios pasos para definirlos *formalmente*.

- Los pasos incluyen:
  - Setup
  - Key generation
  - Encryption
  - Decryption
  - Key delegation
  - Key revocation
  - . . .

Los pasos dependen del protocolo en particular

Firma Elgamal

- Generación de llaves:
  - Generar un primo p
  - Encontrar un elemento $\alpha \in \mathbb{Z}_p^\star$
  - Seleccionar un elemento aleatorio d, con $2 < d < p - 2$
  - Calcular $\beta = \alpha^d \bmod p$

Firma Elgamal de mensaje

- Firma de mensaje:
  - Dado un mensaje M
  - Seleccione una llave efímera k_E, con $0 < k_E < p - 2$, con $\text{MCD}(k_E, p-1) = 1$
  - Calcule $r \equiv a^{k_E} \bmod p$
  - Calcule $s \equiv (M - d \cdot r)k_E^{-1} \bmod p - 1$

- La firma de M es (r, s)

Verificación de Firma Elgamal

- Verificación de firma:
  - Calcule $t \equiv \beta^r \cdot r^s \bmod p$

- Si $t \equiv \alpha^x \bmod pq$, la firma verificó.

Ejemplo, M a firmar

$$p = 29,\ \alpha = 2$$
$$d = 12$$
$$\beta = \alpha^d \equiv 7 \bmod 29$$

$$\xleftarrow{k_{\text{pub}}(p,\alpha,\beta)=(29,2,7)}$$

$$k_E = 5\ (5, 28) = 1,\ x = 26$$
$$r = 2^5 \equiv 3 \bmod 29$$
$$s = -10 \cdot 7 \equiv 26 \bmod 29$$

$$\xleftarrow{(26,(3,26))}$$

$$t = 7^3 \cdot 3^{26} \equiv 22 \bmod 29$$
$$\alpha^x \equiv 2^26 \equiv 22 \bmod 29$$
$$t \equiv \alpha^x \bmod 29 \Rightarrow \text{OK}$$

Firma DSA

La firma estándar DSA contiene los siguientes pasos:

- Generación de claves:
  - Generar un primo p, con $2^{1023} < p < 2^{1024}$
  - Encontrar un primo q divisor de $p - 1$, con $2^{159} < q < 2^{160}$
  - Encontrar un elementos α, cuyo orden sea igual a q
  - Seleccionar un elemento aleatorio d, con $1 < d < q$
  - Calcular $\beta = \alpha^d \bmod p$

- Las claves son:
  - Pública: (p, q, α, β)
  - Privada: d

Firma DSA de mensaje

- Firma de mensaje:
  - Dado un mensaje M
  - Seleccione una llave efímera k_E, con $0 < k_E < q$
  - Calcule $r \equiv (a^{k_E} \bmod p) \bmod q$
  - Calcule $s \equiv (SHA(M) + d \cdot r)k_E^{-1} \bmod q$

- La firma de M es (r, s)

Verificación de Firma DSA

- Verificación de firma:
  - Calcule $w \equiv s^{-1} \bmod q$
  - Calcule $u_1 \equiv w \cdot SHA(M) \bmod q$
  - Calcule $u_2 \equiv w \cdot r \bmod q$
  - Calcule $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$

- Si $v \equiv r \bmod q$, la firma verificó.

Esquema, M a firmar

p-primo aleatorio
q factor de $(p-1)$
$\alpha \in_R \mathbb{Z}_q$, $\text{ord}(\alpha) = q$
secreto $d \in_R \mathbb{Z}q$
$\beta \equiv \alpha^d \bmod p$

$$\xleftarrow{k_{\text{pub}}(p,q,\alpha,\beta)}$$

$k_E \in_R \mathbb{Z}_p$
$r \equiv (\alpha^{k_E} \bmod p) \bmod q$
$s \equiv (\text{SHA}(M) + d \cdot r)(k_E)^{-1} \bmod q$

$$\xleftarrow{(M,(r,s))}$$

$w \equiv s^{-1} \bmod q$
$u_1 \equiv w \cdot \text{SHA}(M) \bmod q$
$u_2 \equiv w \cdot r \bmod q$
$v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$
$v \stackrel{?}{\equiv} r \bmod q$

Ejemplo, M a firmar

$$p = 59,\ q = 29$$
$$\alpha = 3,\ d = 7$$
$$\beta = \alpha^d \equiv 4 \bmod 59$$

$$\xleftarrow{k_{\text{pub}}(p,q,\alpha,\beta)=(59,29,3,4)}$$

$$k_E = 10$$
$$r = (3^{10} \bmod 59) \equiv 20 \bmod 29$$
$$s = (26 + 7 \cdot 20) \cdot 3 \equiv 5 \bmod 29$$

$$\xleftarrow{(M,(r,s))}$$

$$w = 5^{-1} \equiv 6 \bmod 29$$
$$u_1 = 6 \cdot 26 \equiv 11 \bmod 29$$
$$u_2 = 6 \cdot 20 \equiv 4 \bmod 29$$
$$v = 20 \equiv (3^{11} \cdot 4^4 \bmod 59) \bmod 29$$
$$v \equiv r \bmod 29 \Rightarrow \text{OK}$$

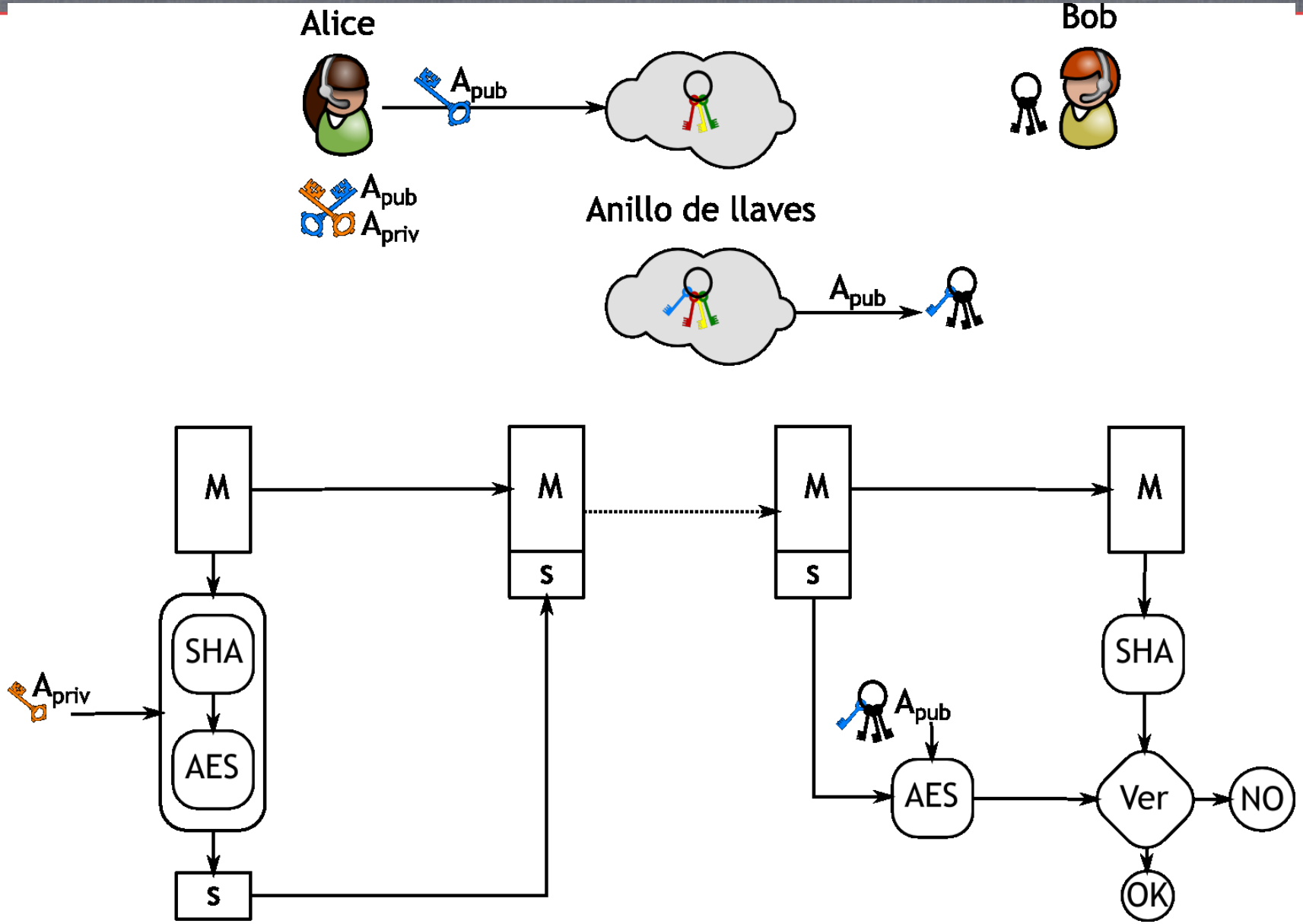
Valores de p y q, Firma DSA

Nivel de Seguridad	p	q	Tamaño de firma
80	1024	160	320
112	2048	224	448
128	3072	256	512
192	7680	384	768
256	15535	512	1024

Otras firmas

- Un método particularmente eficiente en aplicaciones en donde el consumo eléctrico o el espacio es restringido son las firmas cortas basadas en curvas elípticas.
- La criptografía basada en curvas elípticas es un tipo de criptografía de clave pública que requiere menos espacio de la clave que sus contrapartes.
- La criptografía de curvas elípticas es mucho más elaborada, pero permite la implementación eficiente de protocolos de seguridad más interesantes, o a un menor costo.

Firma Digital



Contenido de la sección 4

Certificados digitales

Es un documento que mediante una firma digital de una entidad de confianza, previamente almacenada en el equipo solicitante, asocia una clave pública con una identidad: nombre de la persona, organización, dirección, etc.

El certificado sirve para garantizar que una clave pública en particular pertenece al que dice ser el poseedor de la contraparte privada.

Los certificados son emitidos por una entidad de confianza, una Autoridad Certificadora... aunque en la práctica la relación de confianza se delega a Mozilla, Microsoft, Apple.

Certificados digitales

Es un documento que mediante una firma digital de una entidad de confianza, previamente almacenada en el equipo solicitante, asocia una clave pública con una identidad: nombre de la persona, organización, dirección, etc.

El certificado sirve para garantizar que una clave pública en particular pertenece al que dice ser el poseedor de la contraparte privada.

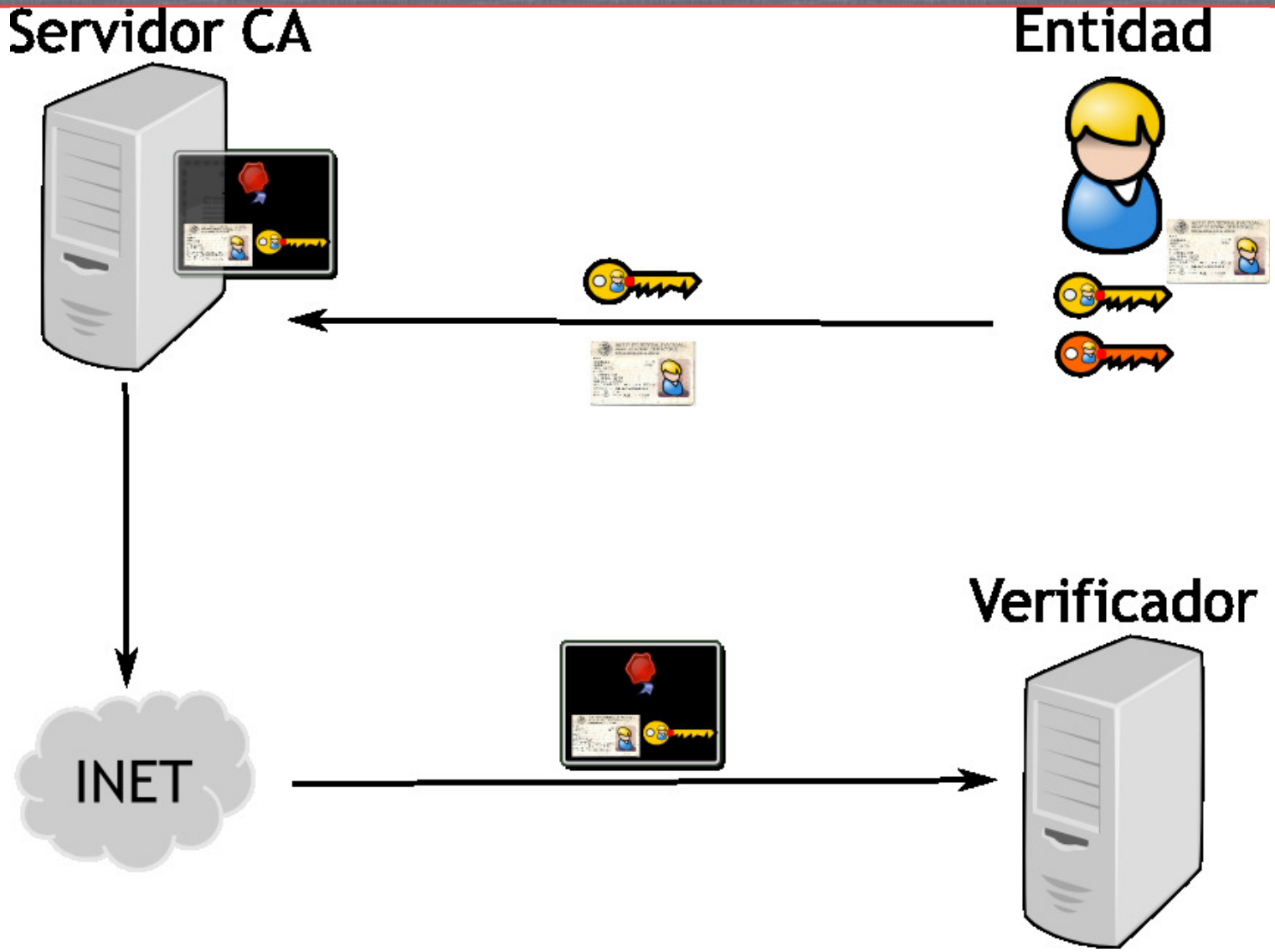
Los certificados son emitidos por una entidad de confianza, una Autoridad Certificadora... aunque en la práctica la relación de confianza se delega a Mozilla, Microsoft, Apple.

Responsabilidades de una CA

Las responsabilidades básicas son:

- Generación de llaves (Intercambio seguro)
- Emisión de Certificados (¿Qué son?)
- Emisión de CRL's (¿Para qué sirven?)

Certificados



Contenido de un certificado

El estándar X.509 establece el formato ASN1 para los certificados digitales, que contienen:

- Número serial
- Sujeto: Persona o entidad identificada
- Algoritmo de firma digital
- Firma digital
- Emisor
- Inicio validez
- Fin validez
- Propósito de la llave: cifrado, firma digital, firma de certificados
- LLave pública
- Algoritmo de huella digital
- Huella digital

Ejemplo

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            07:23:53:8d:87:6d:b6:27:fc:1e:08:aa:49:96:d9:60
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert High Assurance CA-3
        Validity
            Not Before: Oct  8 00:00:00 2012 GMT
            Not After : Dec 16 12:00:00 2015 GMT
        Subject: C=MX, ST=Distrito Federal, L=Mexico, O=Centro de Investigacion... CN=*.cinvestav.mx
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d8:dc:9d:1a:7e:d4:6f:49:5b:7a:95:6a:57:6c:
                    05:8a:c1:0b:3f:b1:03:e0:1a:53:e5:22:8f:bd:6c:
                    c1:59:ec:13:68:5e:f2:6f:44:55:21:36:8c:82:d9:
                    84:4a:e7:97:55:84:f2:cf:71:ad:e4:e5:a6:73:5c:
                    be:5c:23:2d:ab:3b:5d:b7:c3:de:2f:0a:35:74:84:
                    46:23:39:20:78:d4:8b:47:eb:e1:d4:b4:c2:ab:59:
                    8d:7d:33:98:b3:f7:bf:3a:07:c0:64:8a:4f:a6:78:
                    55:87:13:a5:54:b5:e7:be:15:dc:da:9d:61:8c:06:
                    1f:e6:29:01:1e:ab:61:5d:bf:06:cb:ec:48:89:b0:
                    88:6f:e5:b0:4b:bf:83:bd:a0:58:bf:ff:33:0d:f8:
                    c7:73:ff:00:0b:64:f2:2b:9a:69:3f:d5:74:d3:12:
                    0f:e9:15:70:f8:7c:f1:2b:5c:70:d4:49:ce:01:c9:
                    65:47:5f:a2:8f:8f:fa:af:2a:00:c9:ec:20:fd:33:
                    90:12:5c:1c:46:2b:44:24:04:77:44:82:98:26:93:
                    d3:f3:53:a1:5e:a0:f5:f0:1f:f5:6b:22:27:94:a9:
                    2a:45:7d:73:6d:68:39:cf:d2:d2:60:3a:fd:6a:89:
                    2b:a5:22:06:22:46:c2:90:a6:8b:dd:95:61:7b:89:
                    b6:a7
```

```
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:50:EA:73:89:DB:29:FB:10:8F:9E:E5:01:20:D4:DE:79:99:48:83:F7

    X509v3 Subject Key Identifier:
        37:92:15:14:C3:5C:87:5F:C4:63:E2:F3:20:C1:8F:0C:92:B7:BC:7D
    X509v3 Subject Alternative Name:
        DNS:*.cinvestav.mx, DNS:cinvestav.mx, DNS:www.tamps.cinvestav.mx,
         DNS:webmail.tamps.cinvestav.mx, DNS:noc.tamps.cinvestav.mx
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl3.digicert.com/ca3-g15.crl

        Full Name:
          URI:http://crl4.digicert.com/ca3-g15.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.1.1
          CPS: http://www.digicert.com/ssl-cps-repository.htm
          User Notice:
            Explicit Text:

    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers - URI:http://cacerts.digicert.com/DigiCertHighAssuranceCA-3.crt

    X509v3 Basic Constraints: critical
        CA:FALSE
```

```
Signature Algorithm: sha1WithRSAEncryption
     89:72:14:45:fc:52:d2:46:12:ff:fa:f4:c5:4f:fd:7b:0e:e4:
     a7:d9:a1:6d:d4:4e:09:aa:c0:30:2f:1a:92:eb:0c:5b:6a:8f:
     58:26:59:bc:95:d7:73:28:36:47:d1:14:6e:e5:95:d1:ae:35:
     57:3d:2e:c2:9e:86:9f:08:47:a4:31:61:5d:4b:d6:3f:0a:60:
     0d:e4:f3:11:aa:69:9d:c1:6b:ed:ea:53:82:e0:b3:f7:cd:c4:
     d2:b5:5e:60:ef:35:d2:bb:19:68:84:c9:c0:82:8d:e1:80:e8:
     e8:0a:d0:d4:b0:b7:13:4f:43:24:e6:6f:37:4d:8b:f0:b9:0e:
     af:3c:d7:61:89:24:6b:8a:88:88:82:7e:de:4c:12:8a:64:2b:
     75:ca:18:e9:11:8f:7a:c4:0a:55:2a:d6:6a:a8:84:2e:6d:d9:
     f9:f5:fc:48:96:bf:e3:87:2c:02:41:ab:1a:6b:ce:e3:16:65:
     0a:08:56:a2:be:28:ea:47:d2:03:bb:28:ab:f1:b4:ec:62:44:
     cd:c4:14:5d:2c:13:21:6a:d0:6e:6c:29:ba:80:9c:08:a2:50:
     bb:7c:ac:56:41:c0:64:3e:2a:c3:e1:44:38:a0:31:2a:68:4b:
     43:02:27:eb:a5:87:71:e6:79:09:51:a6:82:83:28:30:0f:9a:
     d7:3d:5f:c6
```

Demo

Ejercicio de creación de certificados.

- Generación de Parámetros DSA

```
openssl dsaparam 2048 -out dsaparams.pem
```

- Generación de Llaves

```
openssl gendsa -out dsarootkey.pem dsaparams.pem
```

- Generación de certificado raíz auto-firmado

```
openssl req -newkey dsa:dsaparams.pem -keyout
  dsarootkey.pem -new -x509 -days 365 -out
  rootcert.pem
```

- Examinando el certificado

```
openssl x509 -text -in rootcert.pem | more
openssl asn1parse -in rootcert.pem | more
```

- Generando certificado para el cliente

```
openssl req -newkey dsa:dsaparams.pem -keyout
  dsakey.pem -new -days 365 -out dsareq.pem
```

- Expedición del Certificado

```
openssl x509 -days 180 -CA rootcert.pem -CAkey
  dsarootkey.pem -req -CAcreateserial -CAserial
  ca.srl -in dsareq.pem -out newcert.pem
```

- Examinando el certificado emitido

```
openssl x509 -text -in newcert.pem | more
openssl asn1parse -in newcert.pem | more
```

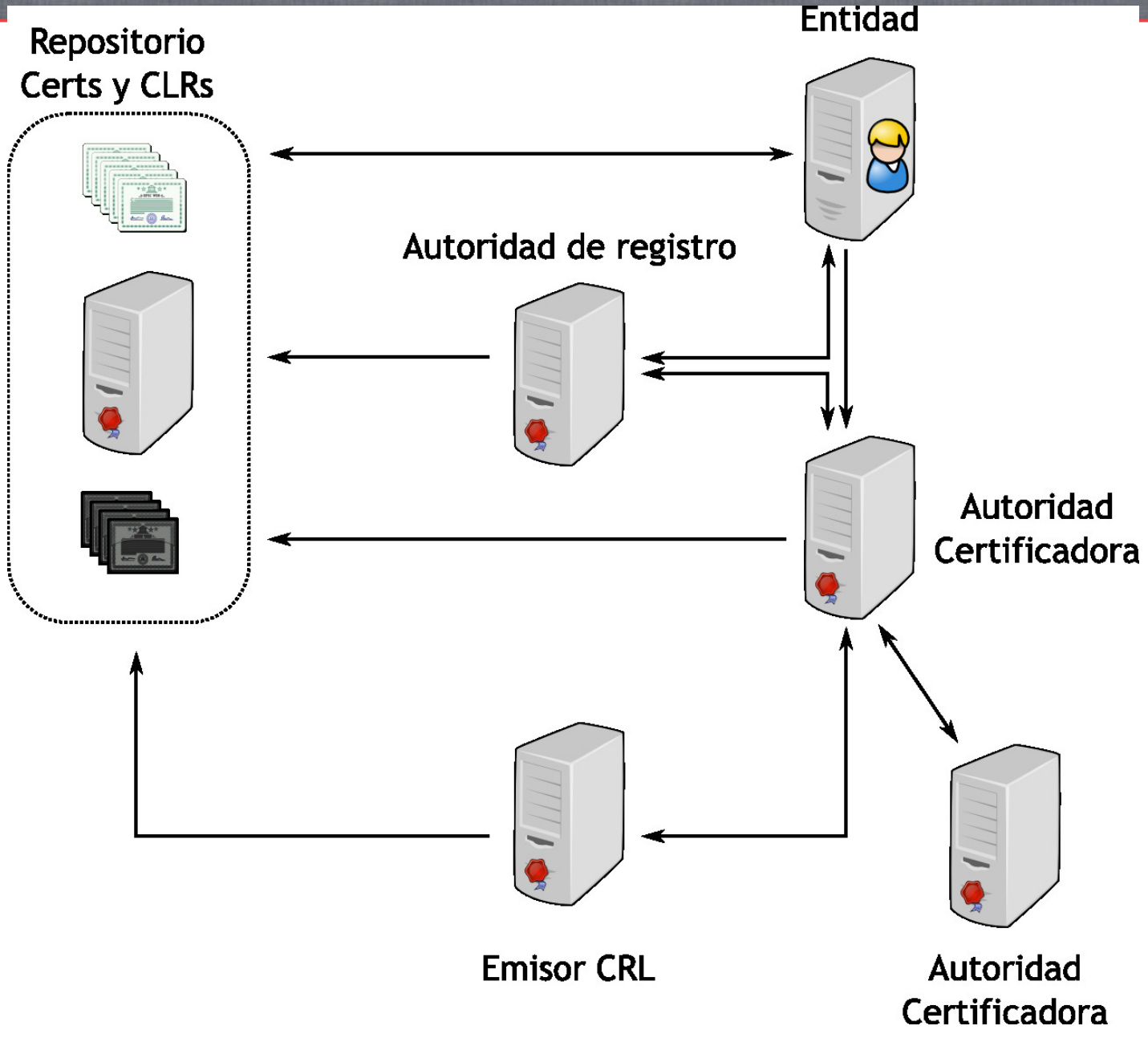
- Verificación del Certificado

```
openssl verify -CAfile rootcert.pem newcert.pem
```

PKI

- La *Infraestructura de Llave Pública (PKI)* es una combinación de software, tecnologías de cifrado, y servicios que permiten proteger la seguridad de las transacciones de información en un sistema distribuido.
- PKI integra (mas bien lo intenta) certificados digitales, criptografía de llave pública y autoridades de certificación en una arquitectura de seguridad unificada.

Diagrama PKI



Elementos en la PKI

- **Entidad final**. Término genérico para denotar a los usuarios finales o cualquier entidad que pueda ser identificada (personas, servidores, compañías, etc.) mediante un certificado digital expedido por una Autoridad Certificadora.

- **Autoridad Certificadora (AC)**. La AC es la entidad que expide los certificados digitales, así como la lista de revocación (CRL). Adicionalmente puede soportar funciones administrativas, aunque generalmente éstas son delegadas a una o varias Autoridades de Registro.

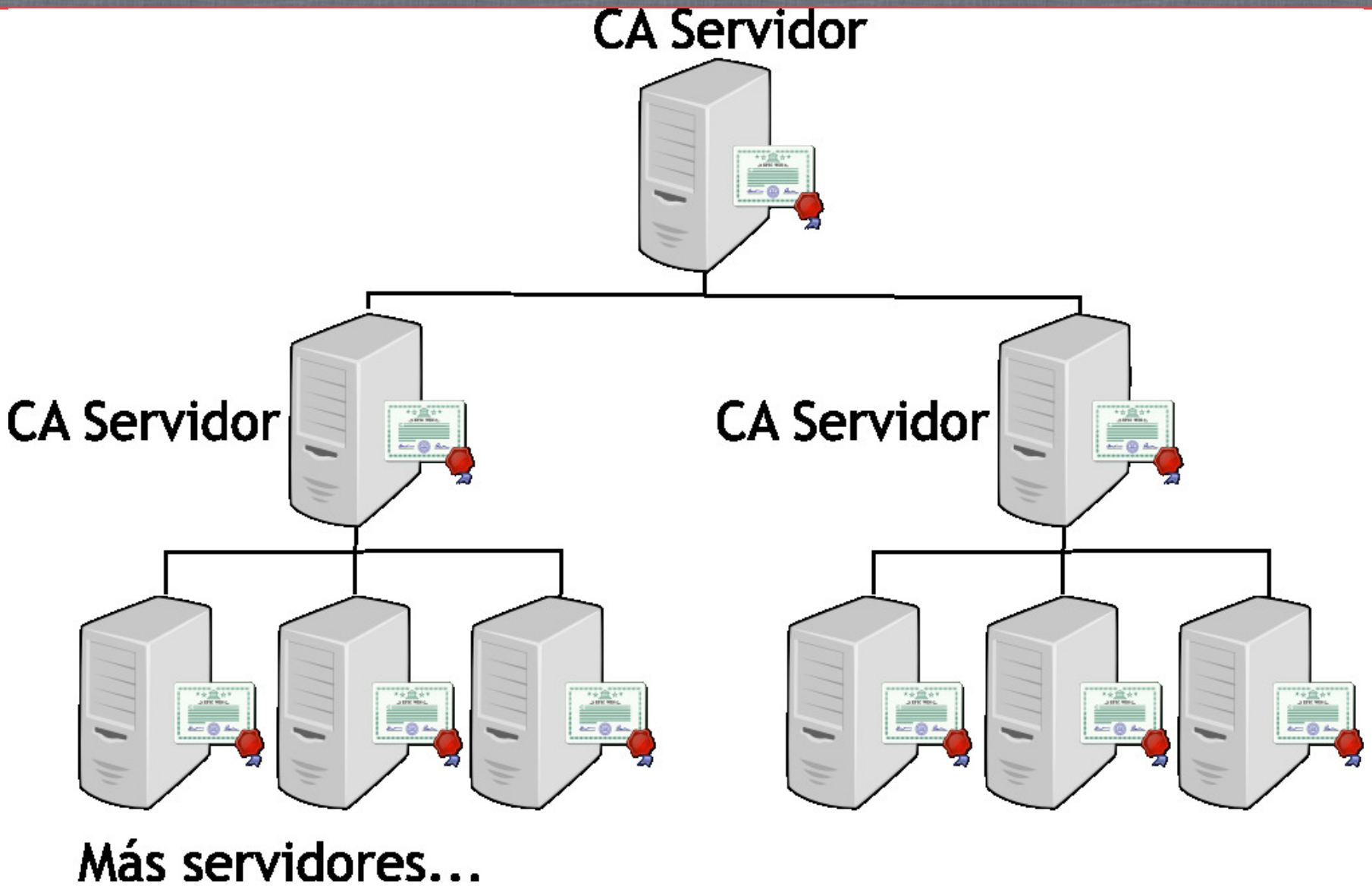
Elementos en la PKI

- **Autoridad de Registro (AR)**. Una AR es componente opcional que puede asumir funciones administrativas de la CA.

- **Repositorio**. El repositorio es el término genérico utilizado para denotar cualquier método para almacenamiento de certificados y listas de revocación (CRLs) que permita el acceso por parte de las entidades finales a dichos documentos.

- **Emisor CRL**. El emisor CRL es un componente opcional el cual puede ser utilizado por una AC para delegar las tareas de publicación de las listas de revocación.

Delegación de Autoridades Certificadoras



- Fin de la unidad 4