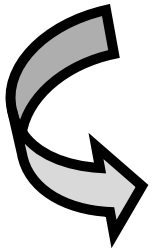
Fundamentos de la Seguridad Informática

Agenda

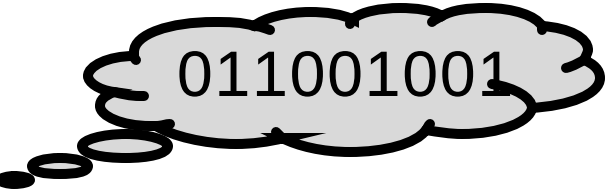
asdc
Argentina Software Development Center

- Funciones de Hash
- Criptografía Asimétrica
- RSA
- Certificados Digitales X-509 V3

- Funciones de Hash



Sirven para garantizar la **integridad** de los textos



Seguridad asociada a una función hash

Tamaño del digesto 128/160/256/384.....

Suponga que hemos creado una función hash de forma que el resumen es sólo de 4 bits, independientemente del tamaño del mensaje de entrada.

Propiedades de las funciones hash

h(M) será segura si tiene las siguientes características:

1. **Unidireccionalidad:** conocido un resumen h(M), debe ser computacionalmente imposible encontrar M a partir de dicho resumen.
2. **Compresión:** a partir de un mensaje de cualquier longitud, el resumen h(M) debe tener una longitud fija. Lo normal es que la longitud de h(M) sea menor que el mensaje M.
3. **Facilidad de cálculo:** debe ser fácil calcular h(M) a partir de un mensaje M.
4. **Difusión:** el resumen h(M) debe ser una función compleja de todos los bits del mensaje M: si se modifica un solo bit del mensaje M, el hash h(M) debería cambiar la mitad de sus bits aproximadamente.

5. **Colisión simple:** será computacionalmente imposible conocido M, encontrar otro M' tal que h(M) = h(M'). Esto se conoce como **resistencia débil a las colisiones**.

6. **Colisión fuerte:** será computacionalmente difícil encontrar un par (M, M') de forma que h(M) = h(M'). Esto se conoce como **resistencia fuerte a las colisiones.**

Diferentes funciones hash

MD5: 128 bits.
SHA-1: 160 bits. Existen otras propuestas conocidas como SHA-256 y SHA-512,
RIPEMD: 160 bits.
N-Hash: 128 bits.
Snefru: 128 y 256 bits.
Tiger: 192 bits.
Panama: 256 bits
Haval: 256 bits.
SHA 3: La fase preliminar del concurso (presentación de candidatos) se cerró el 30 de Octubre de 2008.

	Algorithm Name	Principal Submitter*
1	** Abacus [9M]	Neil Sholer
2	ARIRANG [18M]	Jongin Lim
3	AURORA [12M]	Masahiro Fujita (Sony)
4	BLAKE [9M]	Jean-Philippe Aumasson
5	Blender [168M]	Dr. Colin Bradbury
6	Blue Midnight Wish [18M]	Svein Johan Knapskog
7	** BOOLE [13M]	Greg Rose
8	Cheetah [9M]	Dmitry Khovratovich
9	CHI [8M]	Phillip Hawkes
10	CRUNCH [18M]	Jacques Patarin
11	CubeHash [276M]	D. J. Bernstein
12	** DCH [7M]	David A. Wilson
13	Dynamic SHA [12M]	Xu Zijie
14	Dynamic SHA2 [14M]	Xu Zijie
15	ECHO [8M]	Henri Gilbert
16	ECOH [10M]	Daniel R. L. Brown
17	EDON-R [9M]	Danilo Gligoroski

18	EnRUPT [9M]	Sean O'Neil
19	ESSENCE [37M]	Jason Worth Martin
20	FSB [43M]	Matthieu Finiasz
21	Fugue [8M]	Charanjit S. Jutla
22	Gröstl (New spelling: Grøstl) [7M]	Lars Ramkilde Knudsen
23	Hamsi [9M]	Ozgul Kucuk
24	JH [7M]	Hongjun Wu
25	Keccak [14M]	Joan Daemen
26	** Khichidi-1 [10M]	M Vidyasagar
27	LANE [13M]	Sebastiann Indesteege
28	Lesamnta [13M]	Hirotaka Yoshida
29	Luffa [13M]	Dai Watanabe
30	LUX [9M]	Ivica Nikolic
31	MCSSHA-3 [30M]	Mikhail Maslennikov
32	MD6 [13M]	Ronald L. Rivest
33	** MeshHash [29M]	Björn Fay
34	NaSHA [13M]	Smile Markovski

asdc
Argentina Software Development Center

35	SANDstorm [15M]	Rich Schroeppel
36	Sarmal [17M]	Kerem VARICI
37	Sgàil [19M]	Peter Maxwell
38	Shabal [9M]	Jean-Francois Misarsky
39	** SHAMATA [9M]	Orhun Kara
40	SHAvite-3 [9M]	Orr Dunkelman
41	SIMD [12M]	Gaetan Leurent
42	Skein [14M]	Bruce Schneier
43	Spectral Hash [9M]	Cetin Kaya Koc
44	** StreamHash [7M]	Michal Trojnara
45	SWIFFTX [15M]	Daniele Micciancio
46	** Tangle [14M]	Rafael Alvarez
47	TIB3 [14M]	Daniel Penazzi
48	Twister [9M]	Michael Gorski
49	Vortex [58M]	Michael Kounavis
50	** WaMM [73M]	John Washburn
51	** Waterfall [8M]	Bob Hattersley

Diferentes forma de calcularlo

Forma de quebrarlo

•Criptografía Asimétrica

La operación característica de la cifrado asimétrico es mediante un cifrado exponencial.

Cifrado: $C = M^e \bmod n$

Descifrado: $C^d \bmod n = (M^e)^d \bmod n$

Por lo tanto, la operación $M=C^d \bmod n$ recuperará el número M.

Clave Pública Kpu
Clave Privada Kpr

DH
RSA

asdc
Argentina Software Development Center

Funcionalidad >>>>>

Transporte / intercambio de clave

Intercambio de clave de Diffie y Hellman

Protocolo de Intercambio de Claves de Diffie y Hellman

- **A** y **B** seleccionan un grupo multiplicativo (con inverso) p y un generador α de dicho primo, ambos valores públicos.
- **A** genera un número aleatorio **a** y envía a B $\alpha^{\mathbf{a}}$ mod p
- B genera un número aleatorio b y envía a **A** αb mod p
- B calcula $(\alpha^{\mathbf{a}})^{b}$ mod p = $\alpha^{\mathbf{a}b}$ mod p y luego destruye b
- **A** calcula $(\alpha^{b})^{\mathbf{a}}$ mod p = $\alpha^{b\mathbf{a}}$ mod p y luego destruye **a**
- El secreto compartido por **A** y B es el valor $\alpha^{\mathbf{a}b}$ mod p

Ejemplo de intercambio de clave de DH

Adela (**A**) y Benito (B) van a intercambiar una clave de sesión dentro del cuerpo primo p = 1.999, con $\alpha = 33$. El usuario **A** elegirá a = **47** y el usuario B elegirá b = 117.

Algoritmo:

- **A** calcula $\alpha^a \bmod p = 33^{\mathbf{47}} \bmod 1.999 = 1.343$ y se lo envía a B.
- B calcula $\alpha^b \bmod p = 33^{117} \bmod 1.999 = 1.991$ y se lo envía a **A**.
- B recibe 1.343 y calcula $1.343^{117} \bmod 1.999 = 1.506$.
- **A** recibe 1.991 y calcula $1.991^{\mathbf{47}} \bmod 1.999 = 1.506$.

Protocolo de Transporte de Claves RSA

En febrero de 1978 Ron **R**ivest, Adi **S**hamir y Leonard **A**dleman proponen un algoritmo de cifra de clave pública: **RSA**

Pasos para generar las claves

1. Cada usuario elige un grupo **n = p*q** (pueden y de hecho son distintos).
2. Los valores p y q no se hacen públicos.
3. Cada usuario calcula **f(n) = (p-1)(q-1).**
4. Cada usuario elige una clave pública **e** de forma que $1 < e < \phi(n)$ y que cumpla con la condición: **mcd [e, f(n)] = 1**.
5. Cada usuario calcula la clave privada **d = inv [e, f(n)].**
6. Se hace público el grupo **n** y la clave **e (Kpu)**.
7. Se guarda en secreto la clave **d (Kpr)**. También guardará **p** y **q**

Protocolo de Transporte de Claves RSA

Ejemplo numérico

Grupo n = 91 = 7∗13; $\phi(n) = \phi(7*13) = (7-1)(13-1) = 72$ **N = 48**

Elegimos e = 5 pues mcd (5,72) = 1 ∴ d = inv(5,72) = 29

CIFRADO:

$C = N^e \bmod n = 48^5 \bmod 91 = 5245.803.968 \bmod 91 = 55$

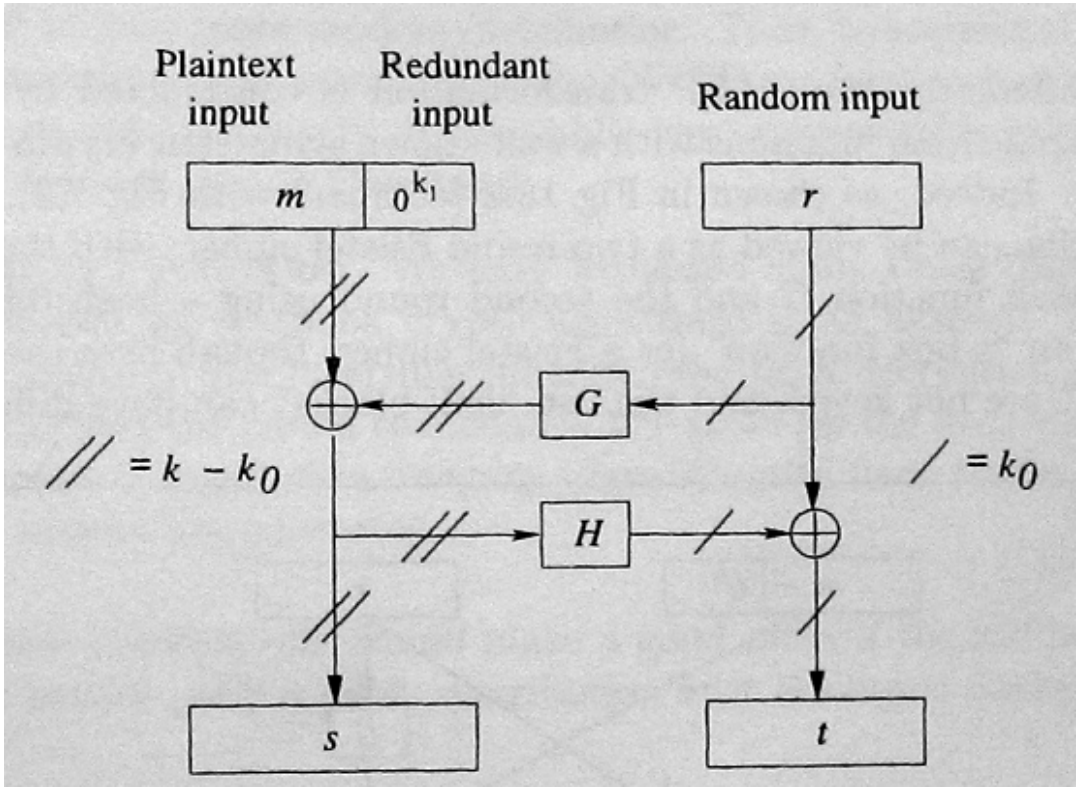
DESCIFRADO:

$N = C^d \bmod n = 55^{29} \bmod 91 = 48$

Debilidad Reportada del RSA

Recientemente el estándar PKCS#1 v. 1.5 ha sido actualizado para acercar a la función RSA a una función probabilistica y así evitar ataques del tipo de mensaje escogido (**Chosen-plaintext attack**), la modificación ha dado como resultado una nueva versión del PKCS #1, la versión 2. La principal modificación es agregar un OAEP Optimal Asymetric Encryption Padding que de algún modo confunde el mensaje original de forma diferente cada vez que se encripta, y así el resultado simule una función probabilistica.

OAEP, Optimal Asymmetric Encryption Padding o Rellenado Asimétrico Óptimo de la Encriptación es un esquema de rellenado de uso frecuente junto con la encriptación del RSA. OAEP fue introducido por Bellare y Rogaway



- Encryption
  - r = rand(k0); s = (m||0..0) ⊗ G(r); t = r ⊗ H(s)
  - c = $(s||t)^e$ mod N

- Decryption
  - s||t = c^d mod N; |s| = |m| + k1; |t| = k0
  - u = t ⊗ H(s); v = s ⊗ G(u)

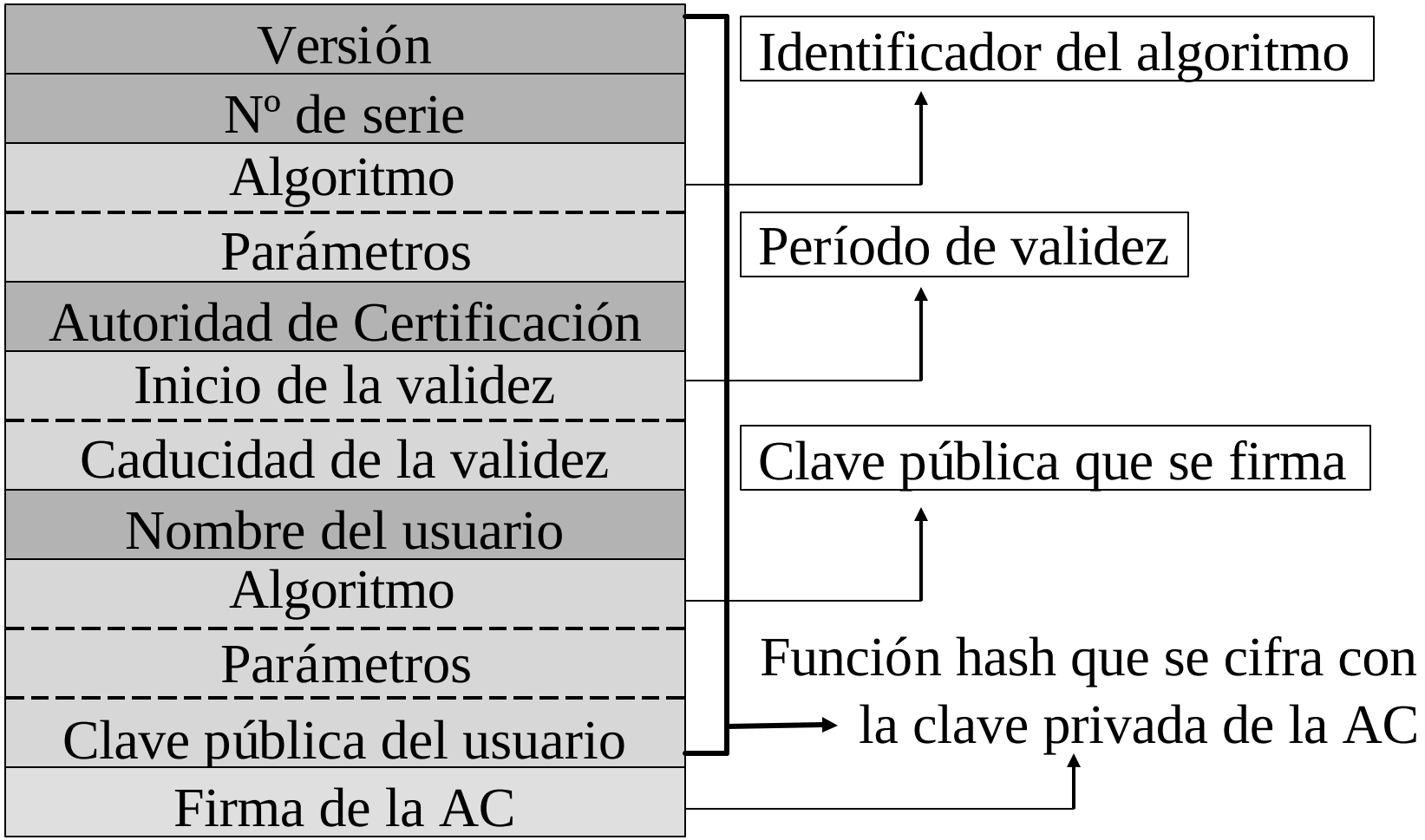
•Certificados Digitales X-509 V3

asdc
Argentina Software Development Center

- Un certificado digital es un documento que contiene diversos datos, entre ellos el nombre de un usuario y su clave pública, y que es firmado por una Autoridad de Certificación (AC).
- Como emisor y receptor confiarán en esa AC, el usuario que tenga un certificado expedido por ella se autenticará ante el otro, en tanto que su clave pública está firmada por dicha autoridad.
- Una de las certificaciones más usadas y un estándar en la actualidad en infraestructuras de clave pública PKIs (Public-Key Infrastructure) es X.509.

- X.509 está basado en criptografía asimétrica y firma digital.
- En X.509 se define un framework (una capa de abstracción) para suministrar servicios de autenticación a los usuarios del directorio X.500.
- La autenticación se realiza mediante el uso de certificados.

Formato del certificado digital X.509



Formato del certificado digital X.509

asdc
Argentina Software Development Center

- **V:** Versión del certificado (actualmente V3).
- **SN:** Número de serie.
- **AI:** identificador del algoritmo de firma que sirve para identificar el algoritmo usado para firmar el paquete X.509.
- **CA:** Autoridad certificadora.
- **TA:** Periodo de validez.
- **A:** Propietario de la clave pública que se está firmando.
- **P:** Clave pública más identificador de algoritmo utilizado y más parámetros si son necesarios.
- **Y{I}:**Firma digital de Y por I usando la clave privada de la unidad certificadora.

IntelExternalBasicIssuingCA3A.crt

Certificate:

Data:

Version: 1 (0x0)

Serial Number: 1 (0x1)

Signature Algorithm: md5WithRSAEncryption

Issuer: O=Intel, OU=Seguridad en informatica/emailAddress=gjaiquel@gmail.com, L=Cordoba, ST=Cordoba, C=AR, CN=localhost

Validity

Not Before: Jun 8 03:35:14 2009 GMT

Not After : Jun 6 03:35:14 2019 GMT

Subject: C=AR, ST=Cordoba, O=IUA, OU=Seguridad, CN=localhost/emailAddress=gjaiquel@gmail.com

Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (1024 bit) Modulus (1024 bit):

00:d6:de:da:9a:80:72:33:22:0b:a9:53:0e:9c:3f: 07:d5:04:2d:5f:34:0e:d8:b2:57:46:82:4f:42:1e:
c4:8b:7a:77:80:83:fd:7a:3a:0d:e7:a7:a6:40:d3: 65:e2:dc:4e:7c:fa:75:25:e1:35:4a:2c:a4:80:55:
65:87:99:35:bd:6d:e6:4d:31:06:af:69:9c:f0:bb: f3:f1:7e:e9:8f:b7:f7:51:41:31:57:08:b2:71:e9:
df:b4:61:6e:3e:a7:cc:67:82:e4:fe:ba:14:6b:d6: 9d:16:6c:58:9e:d3:07:cf:30:50:85:d3:5e:8e:c6: d2:f1:1a:01:80:49:8b:59:1d

Exponent: 65537 (0x10001) Signature Algorithm: md5WithRSAEncryption

2d:28:25:80:99:81:d4:9b:02:0f:32:7b:ee:14:56:a3:89:fb: 3a:e5:b1:12:8b:75:d0:08:68:b1:e9:a4:64:b6:03:91:dd:5e:
8d:ee:61:6c:d9:47:cd:5d:57:73:c8:7d:15:52:43:31:7e:e0: 6a:cb:5f:49:62:b6:e6:99:26:e8:d4:41:f6:7c:df:53:d0:d5:
ca:59:e4:c5:35:ef:b8:ac:58:cf:01:7a:bb:a9:53:dd:4a:0e: 31:40:1f:a8:a6:6f:62:cd:51:33:50:ff:fc:f5:03:da:47:96:
40:85:2f:50:02:17:5f:36:ce:78:0f:b7:e5:3c:e9:76:41:45: 53:c1

-----BEGIN CERTIFICATE-----
MIICgTCCAeoCAQEwDQYJKoZIhvcNAQEEBQAwgZkxDDAKBgNVBAoTA0lVQTEhMB8GA1UECxMYU2VndXJpZGFkIGVuIGluZm9ybWF0aWNhMSEwHwYJKoZIhvc
NAQkBFhJnamFpcXVlbEBnbWFpbC5jb20xEDAOBgNVBAcTB0NvcmRvYmExEDAOBgNVBAgTB0NvcmRvYmExCzAJBgNVBAYTAkFSMRIwEAYDVQQDEwlsb2Nh
bGhvc3QwHhcNMDkwNjA4MDMzNTE0WhcNMTkwNjA2MDMzNTE0WjB4MQswCQYDVQQGEwJBUjEQMA4GA1UECBMHQ29yZG9iYTEMMAoGA1UEChMDSVVB
MRIwEAYDVQQLEwlTZWd1cmlkYWQxEjAQBgNVBAMTCWxvY2FsaG9zdDEhMB8GCSqGSIb3DQEJARYSZ2phaXF1ZWxAZ21haWwuY29tMIGfMA0GCSqGSIb3D
QEBAQUAA4GNADCBiQKBgQDW3tqagHIzlgupUw6cPwfVBC1fNA7YsldGgk9CHsSLeneAg/16Og3np6ZA02Xi3E58+nUl4TVKLKSAVWWHmTW9beZNMQavaZz
wu/PxfumPt/dRQTFXCLJx6d+0YW4+p8xnguT+uhRr1p0WbFie0wfPMFCF016OxtLxGgGASYtZHQIDAQABMA0GCSqGSIb3DQEBBAUAA4GBAC0oJYCZgdSbAg8
ye+4UVqOJ+zrlsRKLddAlaLHppGS2A5HdXo3uYWzZR81dV3PIfRVSQzF+4GrLX0lituaZJujUQfZ831PQ1cpZ5MU177isWM8BerupU91KDjFAH6imb2LNUTNQ//z
1A9pHlkCFL1ACF182zngPt+U86XZBRVPB

-----END CERTIFICATE-----

Diferentes formas de generarlo

Windows -------- Linux

http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html

http://www.rsa.com/

http://triptico.com/docs/diffie_hellman.html

http://www.ietf.org/html.charters/pkix-charter.html

https://digitalid.verisign.com/client/enroll.htm

asdc
Argentina Software Development Center

Q&A

asdc
Argentina Software
Development Center

Tks

applause please