Añadidos Tema 2.
¿Por qué / cómo funciona RSA?

Seguridad en Sistemas Informáticos
4^o Grado en Ingeniería Informática – ESEI

Septiembre-2018

Previo: Aritmética modular

Es equivalente:

(1) $\mathbf{a} \equiv \mathbf{b} \pmod{\mathbf{n}}$ — *a y b son* { **equivalentes** / *congruentes* } *módulo n*

(2) $\mathbf{a} \bmod \mathbf{n} = \mathbf{b} \bmod \mathbf{n}$ — *a* y *b* tienen el mismo resto *módulo n* (operaciones *módulo n*)

(3) $\exists\, \mathbf{k} \geq \mathbf{1}$ tal que $\mathbf{a} = \mathbf{k} \cdot \mathbf{n} + \mathbf{b}$ — *b* es el **resto** de la división entera de *a* entre *n*

[también: (*a* - *b*) es múltiplo de *n*]

Dadas las operaciones + ("suma") y · ("producto")
Si $a \equiv b \pmod{n}$ y $c \equiv c \pmod{n}$ se verifica que:

- $a + c \equiv b + d \pmod{n}$
- $a \cdot c \equiv b \cdot d \pmod{n}$

El conjunto de enteros *módulo n* ($\mathbb{Z}_n$) forma un **anillo conmutativo** algebraico de *n* elementos (+ y · verifican las propiedades asociativa, conmutativa y distributiva).

- Si *n* es primo, se tratará además de un **cuerpo finito** (todos los elementos tienen *inverso multiplicativo*)

Bases de RSA (I)

Suposición de partida

(1) Supongamos que existiera un *"número mágico"* **x** en $\mathbb{Z}_n$ que verificara

$$M^x \equiv M \pmod{n} \quad \forall\, M \in \mathbb{Z}_n$$
$$(\text{ó } M^x \bmod n = M)$$

Nota: { - en el caso de los números enteros sólo el 1 verifica esa propiedad
- en $\mathbb{Z}_n$ puede haber muchos elementos que sí la verifiquen

(2) Si además ese x se pudiera descomponer como $\mathbf{x \equiv e \cdot d \pmod{n}}$, por las propiedades de la aritmética modular, tendríamos

- una operación de cifrado:

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

- una operación de descifrado:

$$\begin{aligned} C^d \bmod n &= (M^e \bmod n)^d \bmod n = \\ &= M^{e\cdot d} \bmod n = M^x \bmod n = M \end{aligned}$$

En RSA, los exponentes e y d, junto con el módulo n, se seleccionan para que cumplan esas dos suposiciones.

Bases de RSA (II)

Teorema de Euler-Fermat

Si **a** y **n** son primos relativos (no tienen factores comunes excepto el 1) entonces se verifica

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

Ofrece las condiciones teóricas para caracterizar los exponentes e y d y el módulo n empleados en RSA.

$\varphi(n)$ es la función **totient de Euler**, que se corresponde con el número de enteros positivos menores o iguales que n que son *primos relativos* con n.

$$\varphi(n) = |\{i \in \mathbb{N} |\ i \leq n \text{ y } mcd(i, n) = 1\}|$$

En el caso de que n sea un número primo, $\varphi(n) = n - 1$.

Bases de RSA (III)

En RSA los exponentes e y d se seleccionan para que sean **inversos multiplicativos** *módulo* $\varphi(n)$.

- Es decir, e y d verifican $\boxed{e \cdot d \equiv 1 \pmod{\varphi(n)}}$
- Por lo tanto, tenemos que $\boxed{e \cdot d = k \cdot \varphi(n) + 1}$.

Esta restricción garantiza que el descifrado ”funcionará”:

- Partimos del cifrado de M usando la operación $\mathbf{C} = M^e \bmod n$.
- La operación de descifrado de C será:

$$
\begin{aligned}
C^d \bmod n &= (M^e \bmod n)^d \bmod n = && \textit{(por las propiedades de la exponenciación)} \\
&= M^{e \cdot d} \bmod n = && \textit{(aplicando que e y d son inversos multiplicativos)} \\
&= M^{k \cdot \varphi(n)+1} \bmod n = && \textit{(por las propiedades de la exponenciación)} \\
&= M^{k \cdot \varphi(n)} \cdot M^1 \bmod n = && \\
&= \left(M^{\varphi(n)}\right)^k \cdot M \bmod n = && \textit{(aplicando el teorema de Euler)} \\
&= 1^k \cdot M \bmod n = \mathbf{M} && \text{(descifrado)}
\end{aligned}
$$

Tiempos de factorización

Tabla extraída del artículo original: Tiempos estimados (1978) para diversos tamaños de clave expresados en dígitos decimales.

- R. Rivest, A. Shamir, L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of the ACM, Vol. 21 (2), pp.120–126. 1978.
  `http://people.csail.mit.edu/rivest/Rsapaper.pdf`

steps (here ln denotes the natural logarithm function). Table 1 gives the number of operations needed to factor n with Schroeppel's method, and the time required if each operation uses one microsecond, for various lengths of the number n (in decimal digits).

Table 1

Digits	*Number of operations*	*Time*
50	1.4×10^{10}	3.9 hours
75	9.0×10^{12}	104 days
100	2.3×10^{15}	74 years
200	1.2×10^{23}	3.8×10^{9} years
300	1.5×10^{29}	4.9×10^{15} years
500	1.3×10^{39}	4.2×10^{25} years

Tamaños de clave recomendados

Recomendaciones de 2012 del NIST sobre tamaños de clave (simétricas y asimétricas) y de resúmenes de funciones HASH en función del tiempo de vida de los datos.

Disponible en `http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf`

Keys length recommendations

Date	Minimum of Strength	Symmetric Algorithms	Asymmetric	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
2010 (Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1** SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
2011 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
> 2030	128	AES-128	3072	256	3072	256	SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
>> 2030	192	AES-192	7680	384	7680	384	SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512
>>> 2030	256	AES-256	15360	512	15360	512	SHA-512	SHA-256 SHA-384 SHA-512

All key sizes are provided in bits. These are the minimal sizes for security.

TDEA (Triple Data Encryption Algorithm) and AES are specified in [10].
Hash (A): Digital signatures and hash-only applications.
Hash (B): HMAC, Key Derivation Functions and Random Number Generation.

Fuente: `http://www.keylength.com/`

”Fortaleza” claves simétricas *vs.* claves asimétricas

Fortaleza relativa de claves simétricas y asimétricas (RSA, D-H, curva elíptica) según las estimaciones del informe 2012 del NIST.

Table 2: Comparable strengths

Bits of security	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
80	2TDEA[18]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

Fuente: `http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf`