Diffie - Hellman
&
RSA

Juan Lorthiois

Temas a abordar

- Sistemas simétricos vs asimétricos
- Cifrado de llave pública
- Describir el algoritmo de Diffie & Hellman
- Describir el algoritmo del RSA
- Determinar las diferencias entre Diffie & Hellman y RSA

El cifrado de Vigenère

- Este método de encripción aunque en desuso desde el siglo XIX ilustra bien un problema común al que los criptólogos deben enfrentarse.
- Para entender como funciona basta un ejemplo:

  Suponga que se quiere encriptar el mensaje: No Such Agency

  Entonces se utiliza la clave: Turing (puede ser cualquier palabra o frase)

  Y se disponen de la siguiente forma:

  Mensaje: NO SUCH AGENCY

  Clave: TU RINGT URINGT

Asignamos a cada par de letras de la combinación mensaje/clave con la letra correspondiente en la tabla:

Mensaje: NO SUCH AGENCY

Clave: TU RING TURING

Cifrado: GI JCPN TAVVPE

Para desencriptar, basta que el receptor del mensaje cifrado aplique el mismo proceso en sentido inverso.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Cifrado de Vigenère

- Por supuesto, este método presentado por el italiano Giovan Battista Bellaso en 1553, es obsoleto y un método de ataque y desencripción fue presentado en 1863 por el alemán Friedrich Kasiski.
- Sin embargo ilustra muy bien un problema: Después de encriptar un mensaje, ¿Cómo hago para transmitirle la clave al destinatario legítimo del mensaje?
- No puedo mandarle la clave junto con el mensaje, sería absurdo.
- Incluso durante la segunda guerra mundial era necesario mantener centros de distribución de claves (KDC) que se encargaban de transmitirle de manera “segura” las claves a todos los encargados de dicho trabajo en las distintas unidades de un ejército.
- En palabras de Diffie: ¿De qué sirve tener sistemas de encripción impenetrables si sus usuarios son forzados a utilizar un KDC que puede ser comprometido por un robo o la orden de un juez?

Características básicas de un sistema criptográfico

- **Transformaciones realizadas sobre el texto a encriptar:**

  Todo sistema de cifrado funciona mediante 2 principios:

  - Sustitución: Es decir un mapeo que asigna una letra o bit a otro elemento de otro conjunto.
  - Transposición: Es decir un reordenamiento de las letras del mismo texto.
  - El requisito principal es que toda operación sea reversible (invertible).

- **El numero de claves utilizadas:**
  - Si el emisor y el receptor de un mensaje utilizan la misma clave entonces se trata de un sistema simétrico.
  - Si el emisor del mensaje utiliza una clave para cifrar/encriptar diferente de la que utiliza el receptor para descifrar el mensaje, entonces se trata de un sistema asimétrico.

- **La manera en que se procesa el texto:**
  - Ya sea el cifrado se hace sobre un bloque de elementos a la vez (block cipher). O bien, se hace un elemento del texto la vez (stream cipher).

Sistemas simétricos

- El cifrado de Vigenère es un ejemplo típico de un sistema simétrico.
- La mayoría de algoritmos de encripción corresponden ya sea a la categoría de los sistemas simétricos o bien a los sistemas asimétricos.

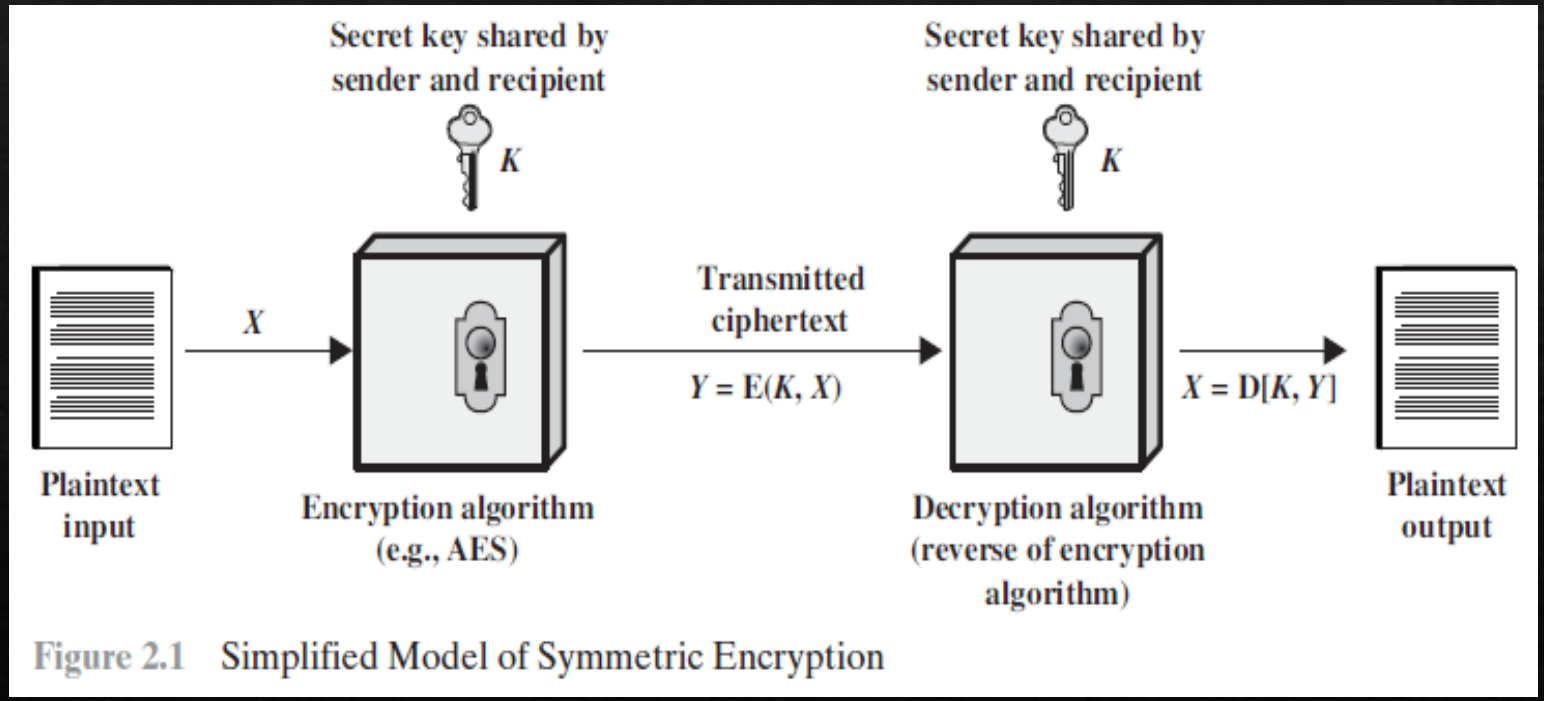
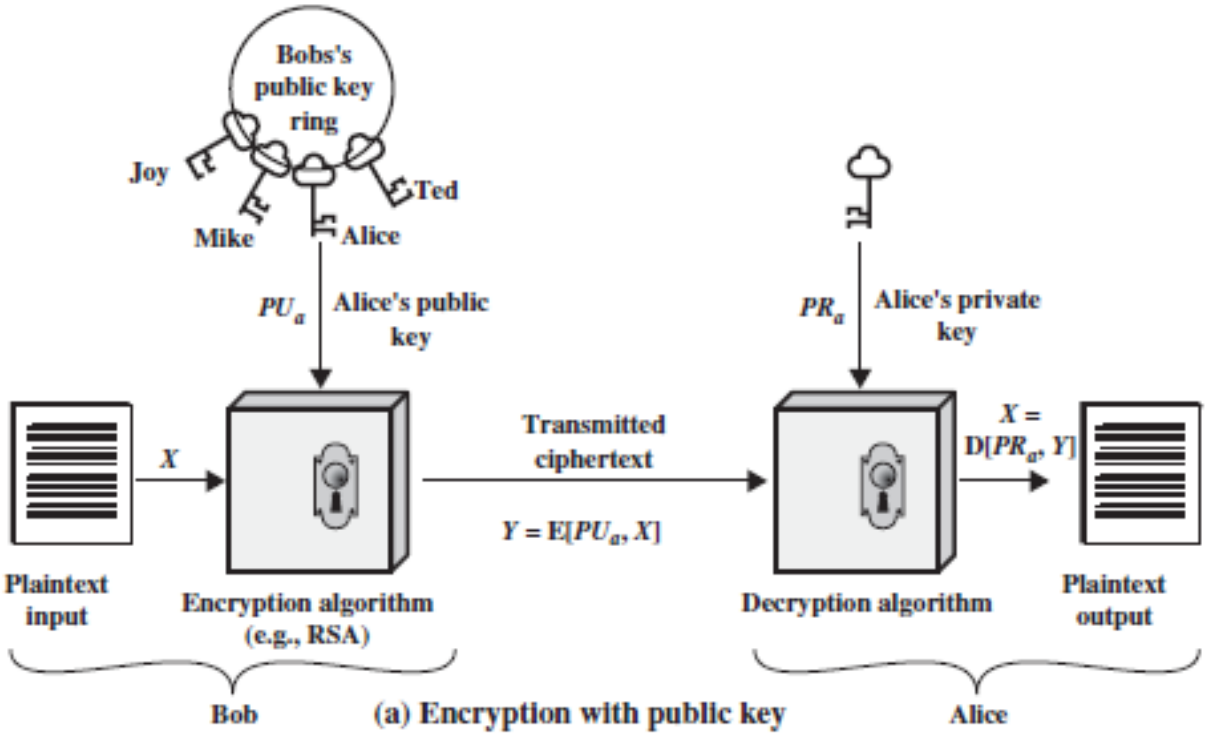


Figure 2.1 Simplified Model of Symmetric Encryption

Stallings W. (2011) *Cryptohraphy and Network security Principles and Practice 5th ed.*

Sistemas asimétricos

- Para resolver el problema de la distribución de claves, los sitemas asimétricos utilzan 2 claves en vez de una.
- Una sirve exclusivamente para encriptar mensajes, es decir la llave pública.
- La otra clave (la privada) sirve exclusivamente para desencriptar mensajes cifrados mediante la llave pública y debe mantenerse secreta.



Stallings W. (2011) *Cryptohraphy and Network security Principles and Practice 5th ed.*

Primer intento: Diffie & Hellman

- Diffie y Hellman son 2 criptólogos norte americanos que publicaron en 1976 la primera implementación conocida de un sistema de llave pública.
- El método se basa en el uso de logaritmos discretos, y encuentra su fuerza en la dificultad de calcular dichos logaritmos para primos muy grandes.
- Este método permite generar un clave compartida entre 2 usuarios aunque intercambien información de manera pública

Algoritmo de Diffie Hellman

- Recordatorio: La raíz primitiva de un primo p es un número a tal que

$$(a) = \{\, a \mod p, a^2 \bmod \mathrm{p}, \ldots, a^{p-1} \bmod \mathrm{p} \,\} = \{1,2, \ldots, p-1\}$$

- Definición: (logaritmo discreto)

Para cualquier entero b y una raíz primitiva a de algún primo p es posible encontrar un único exponente i tal que:

$$b \equiv a^i \pmod{p}\ , \quad 0 \leq i \leq p-1$$

el exponente i se conoce como el logaritmo discreto de b con base a.

Notación: $\mathrm{dlog}_{a,p}(b)$

Algoritmo de Diffie Hellman

1. Seleccionar un número primo p (muy grande) y a una raíz primitiva de p. (ambos números son públicos).
2. El emisor selecciona un número $X_A < p$ y el receptor selecciona un número $X_B < p$. Estos números deben mantenerse secretos (no son publicados).
3. Luego el emisor (A) calcula el numero: $Y_A = a^{X_A} \pmod p$ y el receptor (B) hace lo mismo con su numero secreto: $Y_B = a^{X_B} \pmod p$
4. Ambos usuarios comparten los números Y_A y Y_B de manera publica y mantienen X_A y X_B secretos.
5. Finalmente, ambos generan la nueva clave secreta:

(A) calcula $\mathrm{K} = (Y_B)^{X_A} \pmod p$

(B) calcula $\mathrm{K} = (Y_A)^{X_B} \pmod p$

} Estas expresiones son iguales !

Algoritmo de Diffie Hellman

- Nótese que:

$$(Y_B)^{X_A} \pmod{p} = (a^{X_B} \pmod{p})^{X_A} \pmod{p} = a^{X_A X_B} \pmod{p} = (a^{X_B} \pmod{p})^{X_A} \pmod{p} = (Y_A)^{X_B} \pmod{p}$$

- ✓ De esta manera, vemos que (A) y (B) han intercambiado una clave secreta sin tener que enviarla a través de un canal no seguro.
- ✓ El algoritmo de Diffie Hellman en sí no encripta un mensaje, pero permite compartir de forma segura una clave.
- ✓ De una manera más exacta el algoritmo de Diffie Hellman es un criptosistema asimétrico, de llave pública que permite el intercambio de una clave que por ejemplo podría utilizarse en algún sistema de encripción simétrico.

RSA

- Este sistema de encripción obtiene su nombre de las iniciales de sus creadores:

  Rivest – Shamir – Adleman

- Es una respuesta al paper publicado por Diffie & Hellamn en 1976 donde, además de detallar el método asimetrico para compartir una clave, explican cómo debería de funcionar un criptosistema asimétrico, de llave pública, que permita encriptar mensajes y no solo intercambiar una clave.
- RSA es entonces desarrollado en el transcurso del año 1977 y publicado en 1978.

Consideraciones en la elaboración del RSA

- Para un intercambio cifrado, mediante un algoritmo asimétrico entre un emisor (A) y un receptor (B), los requisitos planteados por Diffie & Hellman en su paper de 1976 son los siguientes:

1. Es computacionalmente "fácil" (tipo **P**) para B generar una pareja (**clave pública** PU_b, **c**lave privada PR_b).
2. Es computacionalmente "fácil" para A, cifrar un mensaje al conocer la clave pública.
3. Es computacionalmente "fácil" para B, descifrar el mensaje utilizando la clave privada.
4. Es computacionalmente "infactible" para un atacante que conoce PU_b determinar PR_b
5. Es computacionalmente "infactible" para un atacante que concoce PU_b y el texto cifrado, concocer el mensaje original.
6. Ambas claves pueden aplicarse en cualquier orden.

Funciones escotilla de 1 dirección:
(Trap door one-way functions)

$Y = f_k(X)$	fácil, si k es conocido
$X = f_k^{-1}(Y)$	fácil, si k es conocido
$X = f_k^{-1}(Y)$	infactible, si k no es conocido

Buscando una función escotilla
(Trap door one-way function)

1. Recordatorio: *(Euler-Fermat)*

Sea M y n , 2 enteros tales que $\mathrm{mcd}(\mathrm{M},\mathrm{n}) = 1$, entonces $M^{\varphi(n)} \equiv 1 \pmod{\mathrm{n}}$.

2. Corolario: Si $M < n$ y $\mathrm{mcd}(\mathrm{M},\mathrm{n}) = 1$, entonces $M^{\varphi(n)} = 1$.

3. Definase:

$$n = p \cdot q, \ \ p, q \text{ números primos,}$$

$$\mathbb{Z}_n = \{0,1, \dots, n-1\}$$

$$\mathbb{Z}_n{}^* = \{x \in \mathbb{Z}_n : \mathrm{mcd}(x,n) = 1\}$$

4. Claramente, si $x \in \mathbb{Z}_n{}^* \Rightarrow x^{\varphi(n)} = 1$

5. Además, nótese que:

$$|\mathbb{Z}_n{}^*| = \varphi(n) = (p-1)(q-1) = n - p - q + 1 \approx n - 2\sqrt{n} \approx n = |\mathbb{Z}_n|$$

Es decir que al tomar un elemento de $\mathbb{Z}_n$, es muy probable que este también pertenezca a $\mathbb{Z}_n{}^*$

Algoritmo RSA:

1. Seleccionar p y q, números primos, tales que $p \neq q$, y $p \cdot q$ tenga más de 309 dígitos (1024 bits).
2. Calcular $n = p \times q$
3. Calcular el totiente de n: $\varphi(n) = \varphi(p \times q) = (p-1)(q-1)$
4. Seleccionese un entero e tal que: $\text{mcd}(\varphi(n), e) = 1$ y $1 < e < \varphi(n)$
5. Determinese d de modo que se cumpla: $d \cdot e \equiv 1 \pmod{\varphi(n)}$

   i.e. $\exists\, k \ni d \cdot e = k \cdot \varphi(n) + 1$

   Nota: en un momento veremos que el paso anterior es necesario para que el proceso de desencripción sea posible mediante aplicación del teorema de Euler-Fermat.
6. Clave publica es la pareja: $PU = \{e, n\}$
7. Clave privada es la pareja: $PR = \{d, n\}$

Requisitos para p y q:

1. *Ser muy grandes*
2. *Tener cantidades similares de dígitos*
3. *No compartirlos nunca*

Nótese que un atacante que conoce PU y desea encontrar PR deberá determinar la factorización prima de n, lo cual es computacionalmente difícil para primos muy grandes.

Encripción - desencricpión con RSA

- Habiendo generado $PU = \{e, n\}$ cualquier persona que dese enviarnos un mensaje puede cifrar con la operación:

$$C = M^e \pmod{n}$$

- Y luego para descifrar el mensaje basta utilizar la clave privada $PR = \{d, n\}$:

$$M = C^d \pmod{n}$$

- Donde M es un bloque numérico a cifrar tal que $M < n$ y $M \in \mathbb{Z}_n{}^*$.

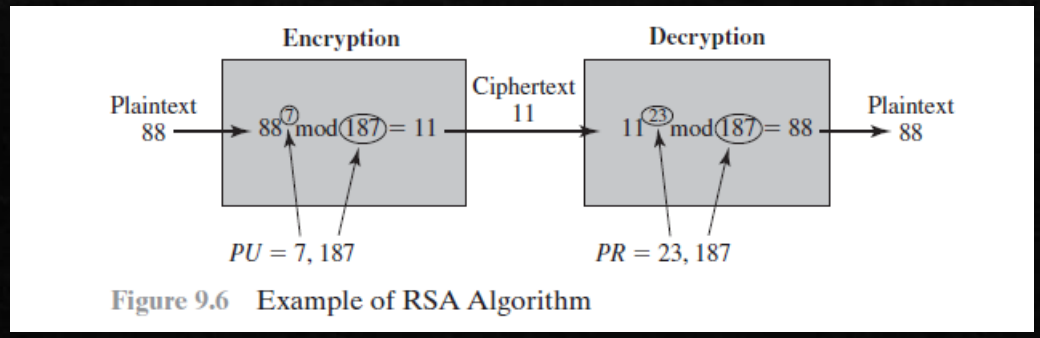


Figure 9.6 Example of RSA Algorithm

Stallings W. (2011) *Cryptohraphy and Network security Principles and Practice 5th ed.*

Verificando que el proceso de desencripción funciona

- Recapitulando tenemos una función "trap-door" $f_e\colon \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ definida por $f_e(x) = x^e \pmod{n}$
- Recordemos que d debe cumplir (paso 5):

$$d \cdot e \equiv 1 \pmod{\varphi(n)} \quad i.e \quad \exists\, k \ni d \cdot e = k \cdot \varphi(n) + 1$$

¿Porqué?

- Considere C un bloque cifrado mediante la función escotilla presentada anteriormente:

$$\Rightarrow C^d = (M^e)^d = \left(M^{ed}\right) = M^{k\cdot\varphi(n)+1} = M^{k\cdot\varphi(n)}M = M$$

- Nótese que $M \in \mathbb{Z}_n^* n \Rightarrow \text{mcd}(M,n) = 1 \Rightarrow$ por el teorema de Euler-Fermat, $(M^{\varphi(n)})^k = 1^k = 1$

Notas sobre RSA

- La versión que se presenta anteriormente, "*Textbook RSA*" es una versión simplificada del proceso real y se ha comprobado que es vulnerable a ciertos ataques, en particular si la función escotilla se aplica directamente sobre el bloque a cifrar.

- Una aplicación más realista es la del "Standard RSA":

1. Utilizar el algoritmo del RSA para generar una pareja PU y PR
2. Escoger un número x al asár en $\mathbb{Z}_n$ y aplicar $f_e(x) = x^e \pmod n = \mathrm{y}.$
3. Utilizar alguna función hash $H(x) = k$ para generar la clave de E_s algún sistema de encripción simétrico.
4. Luego, tras aplicar algún protocolo de "padding" acordado, aplicar sobre M, el mensaje a cifrar, el método de encricpión simétrico utilizando la clave *k* generada. El mensaje cifrado será una pareja (C,y).
5. Para desencriptar, utilizar ${f_d}^{-1}(y) = y^d = x$, y luego utilizar otra vez $H(x) = k$, para obtener la clave de E_s .

RSA pasa a ser un método para compartir claves o de autentificación.

Consideraciones computacionales

- Notese que el fase de encripción y desencripción del mensaje se debe exponenciar números potencilmente grandes con números que seguramente son enormes (en general n tiene más de 200 dígitos decimales) y luego reducirlos móduo n.
- ✓ La solución viene de propiedades aritmeticas:, en particular:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

- La otra consideración es la eficiencia de la exponenciación:
- ✓ Por ejemplo en vez de calcular directamente x^{16}, se puede utilizar la sucesión: $\{x, x^2, x^4, x^8, x^{16}\}$, para acelerar el proceso.

- En general se trata de utilizar los resultados de la teoría de números para acelerar el algoritmo y volverlo más eficiente. Constrarrestando de esta forma las complicaciones que surgen al emplear primos muy grandes.

Ataques al RSA

I. **Fuerza bruta:** i.e. probar todas las posibles claves

II. **Ataques matemáticos:** (todos toman aproximadamente la misma cantidad de tiempo)**:**

1. Factorizar n en sus factores primos. Entonces, podemos determinar $\varphi(n) = (p-1)(q-1)$, y por ende $d \equiv e^{-1} \bmod \varphi(n)$, i.e la parte secreta de PR
2. Determinar $\varphi(n)$ directamente.
3. Determinar d directamente.

III. **Ataques de tiempo:** dependen del tiempo requerido por el algoritmo de desencripción.

IV. **Ataque con texto cifrado escogido:** explotan propiedades del RSA y se escojen y comparan texto cifrados vs. su versión descifrada.

Contraataques o defensas

1. El problema de factorizar n, especialmente si es muy grande, es computacionalmente difícil, y se motró que el de determinar $\varphi(n)$ dado n, equivalente al de factorizar n.
2. Los ataques de tiempo pueden bloquearse con una o varias de las siguientes medidas:
    1. Cersiorarse de que el tiempo de exponenciación para encripción y desencripción de cada bit es el mismo.
    2. Agregar tiempos de espera aleatorios al algoritmo de desencripción – encripción
    3. Multiplicar el bit/bloque cifrado por un número aleatorio, de modo que que el atacante no pueda saber qué bits cifrados están siendo procesados.

¿Valor mínimo para n?

Por el momento se considera que un tamaño de n de 1024 bits ~ 309 digitos decimales a 2048 bits ~ 617 dígitos decimales es seguro.

Recomendaciones adicionales:

1. *p y q solo deben diferir por algunos dígitos.*
2. *$(p-1)$ y $(q-1)$ deben contener un factor primo grande.*
3. $\text{mcd}((p-1),(q-1))$ debe ser pequeño

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA-704 [b]	212	704	US$30,000	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA-220 [b]	220	729		May 13, 2016	S. Bai, P. Gaudry, A. Kruppa, E. Thomé and P. Zimmermann
RSA-230 [b]	230	762		August 15, 2018	Samuel S. Gross, Noblis, Inc.
RSA-232 [b]	232	768		February 17, 2020[13]	N. L. Zamarashkin, D. A. Zheltkov and S. A. Matveev.
RSA-768 [b]	232	768	US$50,000	December 12, 2009	Thorsten Kleinjung *et al.*[14]
RSA-240 [b]	240	795		Dec 2, 2019[15]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann
RSA-250 [b]	250	829		Feb 28, 2020[16]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann
RSA-260	260	862			
RSA-270	270	895			
RSA-896	270	896	US$75,000[d]		
RSA-280	280	928			

Estatus del desafío RSA

Processor / System	Dhrystone MIPS / MIPS	Year
Intel Core i9-9900K	412,090 MIPS at 4.7 GHz	2018
AMD Ryzen 9 3950X	749,070 MIPS at 4.6 GHz	2019
AMD Ryzen Threadripper 3990X	2,356,230 MIPS at 4.35 GHz	2020

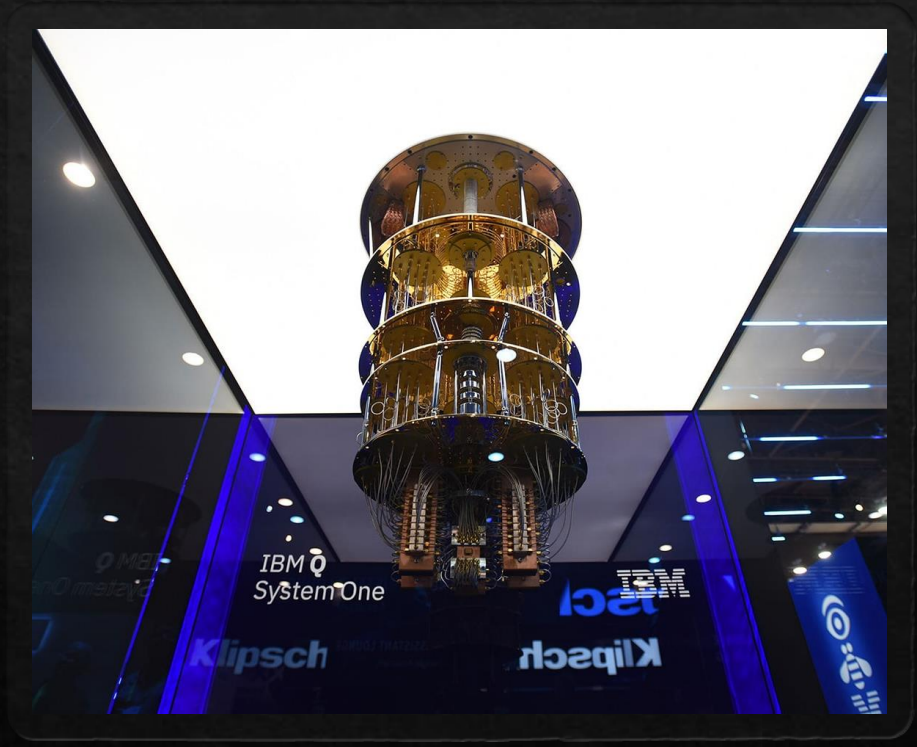
El procesador más rápido disponible para el público

Wikipedia

Quantum vs. RSA

Estado actual de la investigación:

1. En 1994, el matemático Peter Shor detalló un algoritmo eficiente con el que una computadora cuántica podría factorizar números grandes en sus factores primos.
2. Las computadoras cuánticas tienen el problema de que a medida que se incrementa el No. de qubits, tienen tendencia a perder información. Para compensar esto se requiere aún más poder computacional.
3. Según nuevas investigaciones por Craig Gidney (Google) y Martin Ekerå (KTH Royal Institute of Technology, Stockholm), el número de qubits requeridos para poder factorizar numeros de 2048 bits (actualmente el estándar para la mayoría de las implementaciones de RSA) bajó de más de 1 millardo a 20 millones.
4. Por el momento las mejores computadoras cuánticas tienen 70 qubits.
5. Sin embargo, es posible que en 25 años se disponga de una computadora cuántica con estos requisitos, finalizando la vida útil de RSA de 2048 bits cómo lo conocemos actualmente, ya que simplemente incrementar el tamaño de los primos utilizados podría volver la ejecución del algoritmo impráctica por una computadora clásica.
6. Ya existen algoritmos de clave pública “post.cuánticos” es decir resistentes a ataques con computadoras cuánticas, IBM presentó uno en 2019.



Fuentes y lecturas recomendadas

- Stallings W. (2011) *Cryptohraphy and Network security Principles and Practice 5th ed.* Pearson Education. Capítulos: 2.1, 9, 10.1-10.2
- Boneh D. *Online Cryptography Course* Week 6. Link: https://crypto.stanford.edu/~dabo/courses/OnlineCrypto/
- Pagina Wikipedia del desafío RSA: https://en.wikipedia.org/wiki/RSA_Factoring_Challenge
- Pagina Wikipedia con los procesadores catalogados por MIPS e información adicional: https://en.wikipedia.org/wiki/Instructions_per_second
- Bushwick S. (8/10/19) *New Encryption System Protects Data From Quantum Computers* Scientific American. Link: https://www.scientificamerican.com/article/new-encryption-system-protects-data-from-quantum-computers/
- Castelvecchi D. (30/10/20) *Quantum-computing pioneer warns of complacency over internet security.* Nature. Link: https://www.nature.com/articles/d41586-020-03068-9
- Emerging Technology from the arXiv (30/5/19) *How a quantum computer could break 2048-bit RSA encryption in 8 hours.* MIT Technology Review. Link: https://www.technologyreview.com/2019/05/30/65724/how-a-quantum-computer-could-break-2048-bit-rsa-encryption-in-8-hours/