Algoritmos, Secretos y Números Primos

Amalia Pizarro Madariaga
Instituto de Matemáticas
Universidad de Valparaíso

14 Abril 2023

Algoritmos

Algoritmos

Un algoritmo es una secuencia finita de pasos lógicos y ordenados con los cuales podemos solucionar un problema.

Algoritmos

Un algoritmo es una secuencia finita de pasos lógicos y ordenados con los cuales podemos solucionar un problema.

Están presentes tanto en la tecnología como en nuestra vida diaria, cada vez que debemos resolver un problema.

Algoritmos

Un algoritmo es una secuencia finita de pasos lógicos y ordenados con los cuales podemos solucionar un problema.

Están presentes tanto en la tecnología como en nuestra vida diaria, cada vez que debemos resolver un problema.

Los algoritmos están en todas partes!

Algoritmos

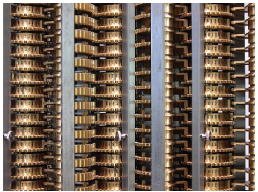
¿Cuál fue el primer algoritmo de la historia? ¿Te imaginas quien lo escribió?

Algoritmos

¿Cuál fue el primer algoritmo de la historia? ¿Te imaginas quien lo escribió?

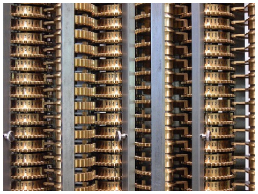
Ada Byron Lovelace (1815-1852)

Máquina Analítica



En una fiesta, conoció a Charles Babbage, matemático e ingeniero que estaba diseñando una calculadora mecánica para resolver automáticamente funciones numéricas: La máquina analítica

Máquina Analítica



En una fiesta, conoció a Charles Babbage, matemático e ingeniero que estaba diseñando una calculadora mecánica para resolver automáticamente funciones numéricas: La máquina analítica

“La Máquina Analítica no tiene ninguna pretensión de originar nada. Es capaz de hacer cualquier cosa, siempre que sepamos ordenarle cómo hacerla.”

Telar Jaqcuard

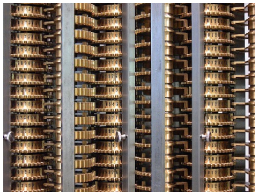
Sus ideas fueron fruto de una visita a las fábricas en donde se utilizaba el famoso “telar de Jacquard”.

Telar Jaqcuard

Sus ideas fueron fruto de una visita a las fábricas en donde se utilizaba el famoso “telar de Jacquard”.

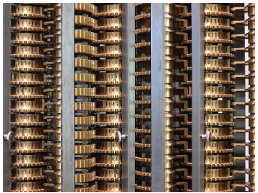
Estas máquinas permitían a usuarios inexpertos la elaboración de complejos diseños en tejidos de seda, mediante tarjetas perforadas que automatizaban el proceso del posterior tejido

El Primer Algoritmo

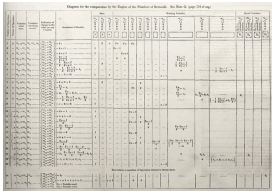


Ada Lovelace expresó con claridad las tres funciones que podía cumplir el invento de Babbage: a) procesar fórmulas matemáticas expresadas con símbolos, b) hacer cálculos numéricos (su objetivo primordial) y c) dar resultados algebraicos en notación literal.

El Primer Algoritmo



Ada Lovelace expresó con claridad las tres funciones que podía cumplir el invento de Babbage: a) procesar fórmulas matemáticas expresadas con símbolos, b) hacer cálculos numéricos (su objetivo primordial) y c) dar resultados algebraicos en notación literal.



En septiembre de 1843, las Notas de Ada Lovelace se publicaron en la revista “Scientific Memoirs”. Este trabajo fue olvidado pues la máquina nunca se construyó. En 1980, el Departamento de Defensa de EE.UU. desarrolló un lenguaje de programación y lo llamó ADA en su honor.

Codificación

- Los procesadores utilizan escritura binaria: $n = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_0 2^0$, con $a_i \in \{0, 1\}$, por ejemplo:

$$2020 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 = (11111100100)_2$$

  decimos en este caso que es un número de 11 bits.

Codificación

- Los procesadores utilizan escritura binaria: $n = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_0 2^0$, con $a_i \in \{0, 1\}$, por ejemplo:

$$2020 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 = (11111100100)_2$$

  decimos en este caso que es un número de 11 bits.
- Un arreglo de 8 bits es llamado byte.

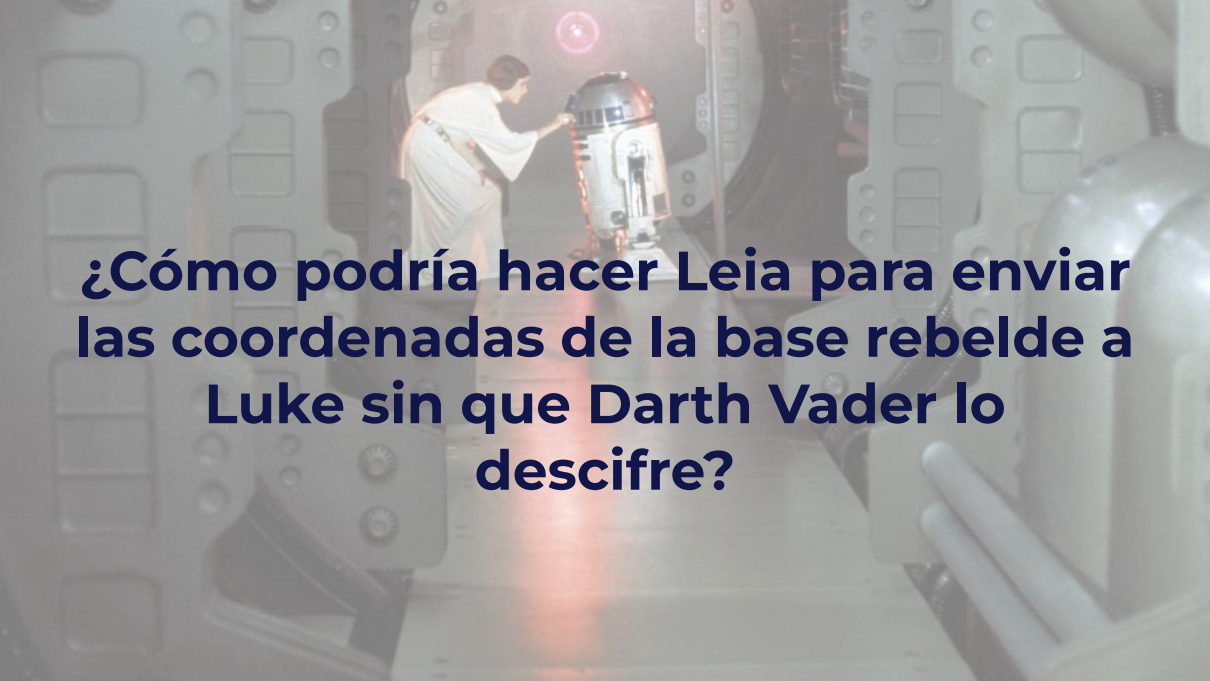
Codificación

Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	`
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(	01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41	)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledg	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[	01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	|
00011101	29	Group separator	00111101	61	=	01011101	93	]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

Las computadoras únicamente entienden números.
El código ASCII es un método de traducción de letras y símbolos a números.
ASCII utiliza códigos de 7 bits, mas un octavo bit de paridad

Lenguajes de Programación

Secretos



¿Cómo podría hacer Leia para enviar las coordenadas de la base rebelde a Luke sin que Darth Vader lo descifre?

Comunicaciones Seguras

- Leia envía el siguiente mensaje privado a Luke y Obi Wan: ***“eres mi última esperanza”***. Pero Darth Vader quiere conocer el mensaje.

Comunicaciones Seguras

- Leia envía el siguiente mensaje privado a Luke y Obi Wan: ***“eres mi última esperanza”***. Pero Darth Vader quiere conocer el mensaje.
- **Confidencialidad**: ¿Vió Vader el mensaje?

Comunicaciones Seguras

- Leia envía el siguiente mensaje privado a Luke y Obi Wan: ***“eres mi última esperanza”***. Pero Darth Vader quiere conocer el mensaje.
- **Confidencialidad**: ¿Vió Vader el mensaje?
- **Autenticidad**: ¿Es el mensaje realmente de Leia?

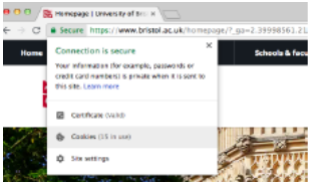
Comunicaciones Seguras

- Leia envía el siguiente mensaje privado a Luke y Obi Wan: ***“eres mi última esperanza”***. Pero Darth Vader quiere conocer el mensaje.
- **Confidencialidad**: ¿Vió Vader el mensaje?
- **Autenticidad**: ¿Es el mensaje realmente de Leia?
- **Integridad**: ¿Es el mensaje exactamente lo que Leia quería enviar?

¿Qué es la criptografía?

¿Qué es la criptografía?

- Por ejemplo en la RAE encontraremos lo siguiente: *“Arte de escribir con clave secreta o de un modo enigmático.”*
- Actualmente consiste en el estudio de métodos de envío de mensajes en forma encubierta, de manera que solo el destinatario pueda remover el contenido y leer el mensaje.



Criptografía

Criptografía de Clave Privada

Supongamos que Leia y Luke se encontraron en persona y en ese momento, acordaron una clave secreta S para poder comunicarse.

Criptografía de Clave Privada

Supongamos que Leia y Luke se encontraron en persona y en ese momento, acordaron una clave secreta S para poder comunicarse.

Eres mi última esperanza

$$\downarrow S$$

$$kofg567pTloGT678ffACAB1312$$

$$\downarrow S^{-1}$$

Eres mi última esperanza

Criptoanálisis

Darth Vader quiere leer el mensaje que Leia le envió a Luke.

Criptoanálisis

Darth Vader quiere leer el mensaje que Leia le envió a Luke.

Para esto, podría intentar conocer la clave secreta de ambos. El estudio y desarrollo de técnicas para ”quebrar” un criptosistema es llamado **Criptoanálisis.**

Criptografía de Clave Pública

- Si Leia y Luke no se han visto en persona y quieren mantener una comunicación confidencial, no podrán establecer una clave secreta para comunicarse.
- Para esto, será útil el concepto de **Criptografía de Clave Pública**

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.
- Es un sistema que permite que cualquier par de usuarios se comuniquen por un canal inseguro **sin un intercambio previo de claves.**

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.
- Es un sistema que permite que cualquier par de usuarios se comuniquen por un canal inseguro **sin un intercambio previo de claves.**
- Está basado en una función $f : A \to B$ tal que:

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.
- Es un sistema que permite que cualquier par de usuarios se comuniquen por un canal inseguro **sin un intercambio previo de claves.**
- Está basado en una función $f : A \to B$ tal que:
- $f(x)$ es fácil de calcular

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.
- Es un sistema que permite que cualquier par de usuarios se comuniquen por un canal inseguro **sin un intercambio previo de claves.**
- Está basado en una función $f : A \to B$ tal que:
- $f(x)$ es fácil de calcular
- $f^{-1}(x)$ computacionalmente difícil.

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.
- Es un sistema que permite que cualquier par de usuarios se comuniquen por un canal inseguro **sin un intercambio previo de claves.**
- Está basado en una función $f : A \rightarrow B$ tal que:
- $f(x)$ es fácil de calcular
- $f^{-1}(x)$ computacionalmente difícil.
- clave=(clave privada, clave pública).

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.
- Es un sistema que permite que cualquier par de usuarios se comuniquen por un canal inseguro **sin un intercambio previo de claves.**
- Está basado en una función $f : A \rightarrow B$ tal que:
- $f(x)$ es fácil de calcular
- $f^{-1}(x)$ computacionalmente difícil.
- clave=(clave privada, clave pública).
- Para **cifrar** un mensaje se utiliza la **clave pública** del destinatario.

Criptografía de Clave Pública

- Este concepto fue introducido por Whiffield Diffie y Martin Hellman en 1976.
- Es un sistema que permite que cualquier par de usuarios se comuniquen por un canal inseguro **sin un intercambio previo de claves.**
- Está basado en una función $f : A \rightarrow B$ tal que:
- $f(x)$ es fácil de calcular
- $f^{-1}(x)$ computacionalmente difícil.
- clave=(clave privada, clave pública).
- Para **cifrar** un mensaje se utiliza la **clave pública** del destinatario.
- Para **descifrar** un mensaje cifrado, el destinatario utiliza su **clave privada.**

Secretos y Números Primos

Algoritmo RSA

El Algoritmo RSA fue el primer algoritmo de cifrado con clave pública y surgió en el año 1978, fue desarrollado por *Rivest, Shamir* y *Adleman*.

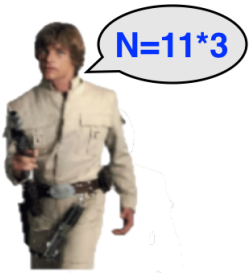
¿Cómo funciona RSA?

- Luke genera una **clave pública** formada por dos números enteros positivos **(N,e)**.

¿Cómo funciona RSA?

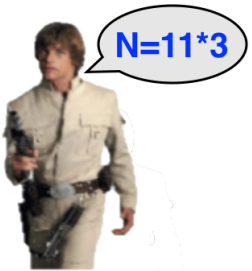
- Luke genera una **clave pública** formada por dos números enteros positivos **(N,e)**.
- Los demás usuarios utilizan esta clave para cifrar los mensajes que quieren enviar a Luke.

Para elegir la clave pública de RSA

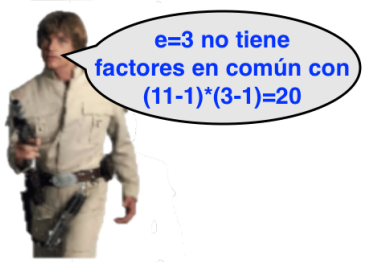


N será un número muy grande y de la forma **N** $= p \cdot q$, donde p y q son números primos . Luke hace público **N** pero mantiene **secretos** los primos.

Para elegir la clave pública de RSA



N será un número muy grande y de la forma **N** $= p \cdot q$, donde p y q son números primos . Luke hace público **N** pero mantiene **secretos** los primos.



e se elige al azar, pero debe verificar que no tenga factores comunes con $(p-1)\cdot(q-1)$, es decir, m.c.d$(e,(p-1)\cdot(q-1))$=1.

Para elegir la clave secreta de RSA

Además, Luke tiene una ***clave privada*** **d**, que sólo él conoce. Este número verifica lo siguiente:

$$e \cdot d \quad : \quad (p-1) \cdot (q-1) = 1$$

Para elegir la clave secreta de RSA

Además, Luke tiene una ***clave privada*** **d**, que sólo él conoce. Este número verifica lo siguiente:

$$e \cdot d \quad : \quad (p-1) \cdot (q-1) =$$
$$1$$

$(11-1) \cdot (3-1) = 20$ y $e = 3$, así que $e \cdot d = 3 \cdot 7 = 21$

$$21 \quad : \quad 20 =$$
$$1$$

¿Cómo enviar un mensaje cifrado con RSA?

Si Leia quiere enviar las coordenadas de la base rebelde a Luke se debe hacer lo siguiente. Supongamos que su mensaje está codificado como M con $1 < M < N$.

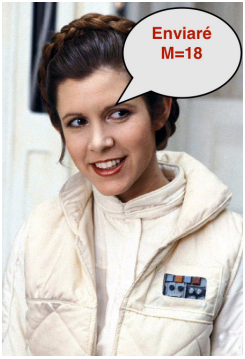
¿Cómo enviar un mensaje cifrado con RSA?

Si Leia quiere enviar las coordenadas de la base rebelde a Luke se debe hacer lo siguiente. Supongamos que su mensaje está codificado como M con $1 < M < N$.

Utilizando la **clave pública** de Luke, Leia calcula el mensaje cifrado C de la siguiente manera:

$$M^e \quad : \quad N = C$$

¿Cómo enviar un mensaje cifrado con RSA?



Si Leia quiere cifrar $M = 18$, utilizando la clave pública $(N, e) = (33, 3)$, debe calcular

$$18^3 \quad : \quad 33 = \\ 24$$

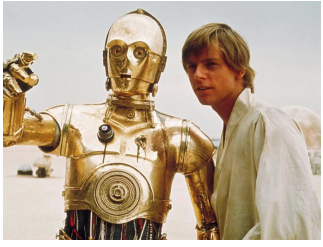
El mesaje cifrado que enviará a Luke es $C = 24$.

Para Descifrar el Mensaje...

Para descifrar el mensaje, Luke hace lo siguiente:

$$C^d \quad : \quad N = M$$

Para Descifrar el Mensaje...



Para descifrar el mensaje, Luke hace lo siguiente:

$$C^{d} \quad : \quad N = M$$

Como recibió $C = 24$, entonces calcula: $24^{7} = 4586471424$ y

$$4586471424 \quad : \quad 33 = 18,$$

recuperando el mensaje original $M = 18$ que le envió Leia.

RSA


Mensaje
Clave Pública
Texto cifrado
Clave Privada
Mensaje
Descifraré su mensaje!!

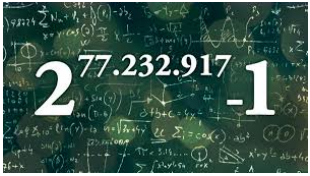

¿Qué tan seguro es RSA?

¿Qué tan seguro es RSA?

Para descifrar el mensaje Darth Vader requiere el valor de la clave privada d, es decir, conocer p y q.

¿Qué tan seguro es RSA?

Para descifrar el mensaje Darth Vader requiere el valor de la clave privada d, es decir, conocer p y q.



Sería suficiente, dado N, factorizarlo. Parece un problema sencillo, pero si N es grande (más de 100 dígitos), no existe un algoritmo que lo haga en un tiempo razonable.

Comentarios

- Los procesadores utilizan escritura binaria: $n = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_0 2^0$, con $a_i \in \{0, 1\}$, por ejemplo:

$$2020 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 = (11111100100)_2$$

  decimos en este caso que es un número de 11 bits.

Comentarios

- Los procesadores utilizan escritura binaria: $n = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_0 2^0$, con $a_i \in \{0, 1\}$, por ejemplo:

$$2020 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 = (11111100100)_2$$

  decimos en este caso que es un número de 11 bits.
- Actualmente, RSA utiliza primos de 2048 bits. Su tamaño es aproximadamente 2^{2048}, y tienen:

$$\log(2^{2048}) \approx 616 \text{ dígitos.}$$

Comentarios

- Los procesadores utilizan escritura binaria: $n = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_0 2^0$, con $a_i \in \{0, 1\}$, por ejemplo:

$$2020 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 = (11111100100)_2$$

  decimos en este caso que es un número de 11 bits.
- Actualmente, RSA utiliza primos de 2048 bits. Su tamaño es aproximadamente 2^{2048}, y tienen:

$$\log(2^{2048}) \approx 616 \text{ dígitos.}$$

- El mejor algoritmo de factorización que se conoce factoriza en dos primos en 900 core-years (900 CPU trabajando por un año)

Problemas abiertos I

- Determinar test de primalidad eficientes. Dato: primo mas grande conocido $2^{82589933} - 1$ con 24.862.048 dígitos.

Problemas abiertos I

- Determinar test de primalidad eficientes. Dato: primo mas grande conocido $2^{82589933} - 1$ con 24.862.048 dígitos.
- Determinar primos seguros para RSA.

Problemas abiertos I

- Determinar test de primalidad eficientes. Dato: primo mas grande conocido $2^{82589933} - 1$ con 24.862.048 dígitos.
- Determinar primos seguros para RSA.
- Determinar un algoritmo eficiente que permita factorizar en producto de primos.


Computación Cuántica

Bits versus Qubits

- **Bit** : unidad de memoria mas elemental de un computador clásico. Solo puede tener un estado, que es 0 o 1.

Bits versus Qubits

- **Bit** : unidad de memoria mas elemental de un computador clásico. Solo puede tener un estado, que es 0 o 1.
- Ejemplo: $76_2 = 1001100 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$, es decir 76 tiene aproximadamente 6 bits.

Bits versus Qubits

- **Bit** : unidad de memoria mas elemental de un computador clásico. Solo puede tener un estado, que es 0 o 1.
- Ejemplo: $76_2 = 1001100 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$, es decir 76 tiene aproximadamente 6 bits.
- Un registro de n bits puede representar 2^n estados diferentes, pero uno y solo uno de estos valores puede estar en un registro en un momento dado.

Bits versus Qubits

- **Bit** : unidad de memoria mas elemental de un computador clásico. Solo puede tener un estado, que es 0 o 1.
- Ejemplo: $76_2 = 1001100 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$, es decir 76 tiene aproximadamente 6 bits.
- Un registro de n bits puede representar 2^n estados diferentes, pero uno y solo uno de estos valores puede estar en un registro en un momento dado.
- **Qubit**: puede almacenar cual valor entero, real o complejo.

Bits versus Qubits

- **Bit** : unidad de memoria mas elemental de un computador clásico. Solo puede tener un estado, que es 0 o 1.
- Ejemplo: $76_2 = 1001100 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$, es decir 76 tiene aproximadamente 6 bits.
- Un registro de n bits puede representar 2^n estados diferentes, pero uno y solo uno de estos valores puede estar en un registro en un momento dado.
- **Qubit**: puede almacenar cual valor entero, real o complejo.
- Un registro cuántico de n qubits puede estar en una superposición arbitraria de hasta 2^n estados simultáneamente.

Criptografía Postcuántica

En 1997, Peter Shor encontró un algoritmo cuántico que resuelve el problema de Factorización de Enteros en tiempo polinomial.

Criptografía Postcuántica

En 1997, Peter Shor encontró un algoritmo cuántico que resuelve el problema de Factorización de Enteros en tiempo polinomial.

Si se logran fabricar procesadores cuánticos con suficiente capacidad para ejecutar este algoritmo, criptosistemas que basan su seguridad en estos problemas serían quebrados en algunas horas.

Bits versus Qubits

- Para factorizar un número entero de 1000 bits (es decir aprox. 2^{1000} se requerirá un procesador cuántico de aprox. 2^{23} qubits.

Bits versus Qubits

- Para factorizar un número entero de 1000 bits (es decir aprox. 2^{1000} se requerirá un procesador cuántico de aprox. 2^{23} qubits.
- 2^n bits (en un procesador clásico) equivalen aprox. a n qubits (en un procesador cuántico)

Problemas Abiertos II

Crear nuevos protocolos criptográficos cuya dificultad de quebrar no se base en factorización de enteros u otros problemas “fáciles’ ’ para un procesador cuántico.

Referencias

- J. Gómez, "Matemáticos, espías y piratas informáticos", El mundo es matemático,2012.
- M. Lucena, "Criptografía y seguridad en computadores", Universidad de Jaen, Escuela Politécnica Superior, 1999.
- L. Jiménez y C. Rojas, "La gran aventura del conocimiento"

Continuará....