Criptografía y Seguridad en Redes

v.2015

Seminario #2

Prof.Ing.Miguel SOLINAS

mksolinas@gmail.com

Agenda

- *Introducción*
- *Criptografía histórica*
- *Fundamentos teóricos*
  - *Teoría de la información*
  - *Teoría de números*
  - *Complejidad computacional*
  - *Números aleatorios*

Teoría de la Información

En 1949, Shannon [Shan49] suministró una base teórica para la criptografía sobre los fundamentos elaborados en su trabajo sobre teoría de la información [Shan48].

Midió el secreto teórico de una cifra a través de la incertidumbre existente sobre el texto plano dado un texto cifrado interceptado. Si no se puede aprender nada sobre el texto plano, sin importar cuánto texto cifrado tengamos, la cifra logra un secreto perfecto.

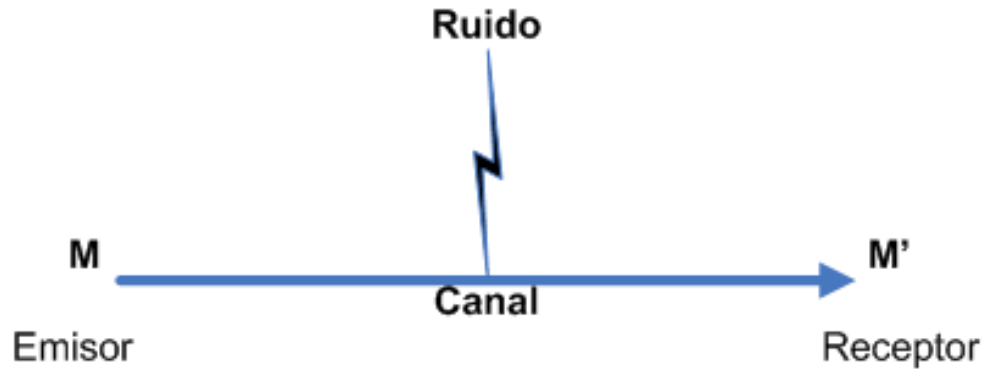
Shan48. Shannon, C. E., "A Mathematical Theory of Communication," Bell Syst. Tech. J. Vol. 27 pp. 379-423 (July), 623-656 (Oct.) (1948).

Shan49. Shannon, C. E., "Communication Theory of Secrecy Systems," Bell Syst. Tech. J. Vol. 28 pp. 656-715 (Oct. 1949).

Teoría de la Información

La Teoría de la Información se enfoca sobre dos problemas relacionados:

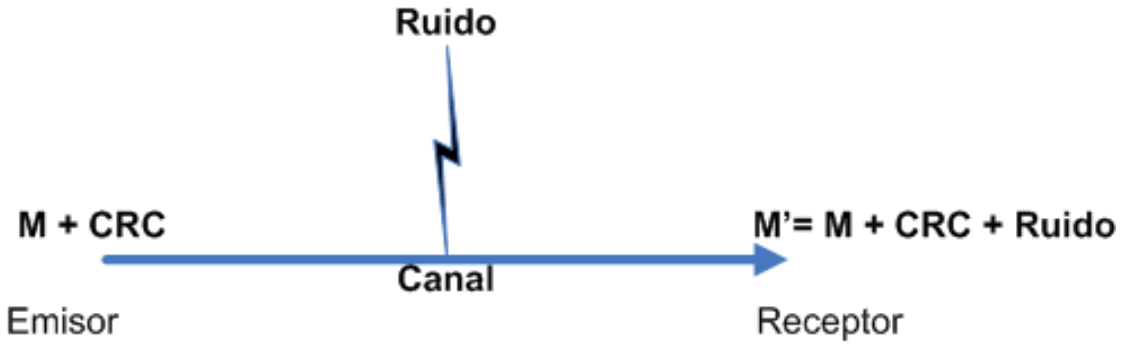
- El problema de ruido en un canal y
- El problema de los secretos



Teoría de la Información

La Teoría de la Información se enfoca sobre dos problemas relacionados:

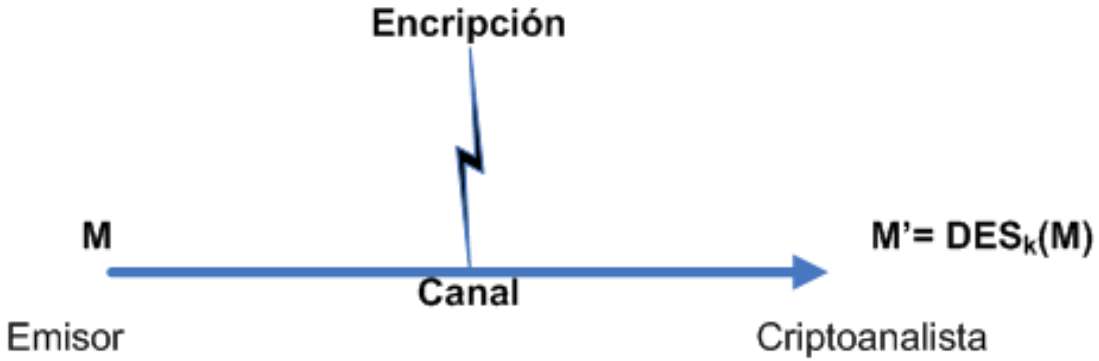
- El problema de ruido en un canal y
- El problema de los secretos



Teoría de la Información

La Teoría de la Información se enfoca sobre dos problemas relacionados:

- El problema de ruido en un canal y
- El problema de los secretos



Cantidad de Información

Vamos a introducir este concepto partiendo de su idea intuitiva. Para ello analizaremos el siguiente ejemplo: supongamos que tenemos una bolsa con NUEVE bolas negras y UNA blanca.

¿Cuánta información obtenemos si alguien nos dice que ha sacado una bola blanca de la bolsa?

¿Y cuánta obtenemos si después saca otra y nos dice que es negra?

Cantidad de Información

Sea V una variable aleatoria.

Sea x_i el *i-ésimo* suceso de V y sea $P(x_i)$ su probabilidad de ocurrencia.

Sea n el número de sucesos posibles.

$$I_i = - log_2 (P(x_i))$$

http://www.fooplot.com

Entropía

Sea V una variable aleatoria.

Sea x_i el *i-ésimo* suceso de V y sea $P(x_i)$ su probabilidad de ocurrencia.

Sea n el número de sucesos posibles.

$$H(V) = - \sum_{i=1}^{n} P(x_i) \log_2 [P(x_i)] = \sum_{i=1}^{n} P(x_i) \log_2 [1 / P(x_i)]$$

Entropía

$$H(V) = - \sum_{i=1}^{n} P(x_i)\ log_2\ [P(x_i)] \ = \ \sum_{i=1}^{n} P(x_i)\ log_2\ [1 / P(x_i)]$$

Propiedades

I. $0 \leq H(V) \leq log_2 N$

II. $H(V) = 0 \Leftrightarrow \exists\ i / P(x_i) = 1$ y $P(x_i) = 0$ para todo $i \neq j$

III. $H(x_1, x_2,..., x_n) = H(x_1, x_2,..., x_n, x_{n+1})$ si $P(x_{n+1}) = 0$

Entropía

La entropía como el número medio de bits necesarios para codificar todos los posibles sucesos de manera óptima.

Un codificador óptimo es aquel que utiliza el menor número de bits para codificar una fuente. Esto se hace pensando en el receptor si tiene que solicitar reenvío de mensajes.

- Cuántos bits de información son necesarios para guardar en una base de datos información del sexo de una persona..?
- Qué cantidad de información hay guardada en el campo “sexo” de una base de datos..?

Entropía

Supongamos que necesitamos codificar una fuente de información que tiene dos estados: cara y seca, ambos igualmente probables.

¿ Qué nos aporta en este caso la Entropía ?

Imaginemos ahora que queremos representar diez símbolos equiprobables, por ejemplo los diez dígitos decimales, utilizando secuencias de bits. Con tres bits no es suficiente, así que necesitaremos más, pero ¿cuántos más?

Entropía Condicionada

Supongamos que estudiamos dos características de un mismo elemento de la población (altura y peso, dos asignaturas, longitud y latitud, cifra y clave).

De forma general, si se estudian sobre una misma población y se miden por las mismas unidades estadísticas una variable X y una variable Y, se obtienen series estadísticas de las variables X e Y.

Considerando simultáneamente las dos series, se suele decir que estamos ante una variable estadística bidimensional.

Entropía Condicionada

Sea una variable aleatoria bidimensional ***(X;Y)***. Las distribuciones de probabilidad que podemos definir sobre dicha variable, con ***n*** posibles casos para ***X*** y ***m*** para ***Y*** :

1. Distribución conjunta de ***(X; Y)***:

$$P(x_i ; y_j)$$

2. Distribuciones marginales de ***X*** e ***Y*** :

$$P(x_i) = \sum_{j=1}^{m} P(x_i , y_j) \qquad P(y_j) = \sum_{i=1}^{n} P(x_i , y_j)$$

3. Distribuciones condicionales de ***X*** sobre ***Y*** y viceversa:

$$P(x_i / y_j) = P(x_i ; y_j) / P(y_j) \qquad P(y_j / x_i) = P(x_i ; y_j) / P(x_i)$$

Entropía Condicionada

Entropía de las distribuciones conjunta y condicionada:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i , y_j)\ log_2\,(P(x_i , y_j))$$

$$H(X/Y=y_j) = -\sum_{i=1}^{n} P(x_i / y_j)\ log_2\,(P(x_i / y_j))$$

Entropía condicionada de X sobre Y: suma ponderada de los $H(X/Y=y_j)$

$$H(X/Y)= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(y_j)P(x_i /y_j)\ log_2(P(x_i /y_j))= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\ log_2(P(x_i /y_j))$$

Entropía Condicionada

Así como existe una Ley de la Probabilidad Total, análogamente se define la Ley de Entropías Totales:

$$H(X, Y) = H(X) + H(Y/X)$$

cumpliéndose además, si X e Y son variables aleatorias independientes:

$$H(X/Y) = H(X) ;$$

$$H(Y/X) = H(Y);$$

$$H(X, Y) = H(X) + H(Y)$$

Teorema de Disminución de la Entropía

La entropía de una variable X condicionada por otra Y es menor o igual a la entropía de X, alcanzándose la igualdad si y sólo si las variables X e Y son independientes.

Esto una idea intuitiva bien clara: conocer algo acerca de la variable Y puede que nos ayude a saber más sobre X (lo cual se debería traducir en una reducción de su entropía) pero en ningún caso podría hacer que aumente nuestra incertidumbre.

Cantidad de Información entre dos variables

Shannon propuso una medida para la cantidad de información, que aporta sobre una variable, el conocimiento de otra.

Se define, la cantidad de información de Shannon que la variable X, contiene sobre la variable Y, como:

$$I(X,Y) = H(Y) - H(Y/X)$$

Inicialmente, poseemos un grado determinado de incertidumbre sobre la variable aleatoria Y. Si antes de medir una realización concreta de Y, medimos la de otra variable X, parece lógico que nuestra incertidumbre sobre Y se reduzca o permanezca igual.

Propiedades:

I. $I(X,Y) = I(Y,X)$

II. $I(X,Y) \geq 0$

Criptosistema seguro de Shannon

Diremos que un criptosistema es seguro si la cantidad de información que aporta el hecho de conocer el mensaje cifrado ***c*** sobre la entropía del texto claro ***m*** vale cero.

Sea una variable aleatoria ***C***, compuesta por todos los criptogramas posibles, y cuya observación corresponderá el valor concreto ***c*** del criptograma enviado, y otra variable ***M***, definida análogamente para los textos en claro ***m***. En ese caso, tendremos que:

$$I(C,M) = 0$$

El sistema que cumpla esta condición jamás se podría romper, ni siquiera empleando una máquina con capacidad de proceso infinita. Por ello los criptosistemas que cumplen la condición de Shannon se denominan también criptosistemas ideales..!!

Teoría de la Información

Redundancia

Según un etsduio de una uivennrsdiad ignlsea no ipmotra el odren en el que las ltears etsen ersciats, la uicna csoa ipormnte es que la pmrirea y la utlima ltera esten ecsritas en la psiocion cocrrtea. El rsteo peuden estar taotlmntee mal y aun prodas lerelo sin pobrleams. Etso es pquore no lemeos cada ltera en si msima, pero si la paalbra cmoo un todo. ¿No te parcee aglo icrneible?

Este es un claro ejemplo de que no necesitas tener un texto presentado de forma excelente para comprender lo que lees. Como te habrás dado cuenta este m mento, este artículo está escrito de f rma que se muestra sólo parte de las letras o las palabras. Aun así estás entendiendo prácticamente todo y, si ad m´s estás leyendo ráp dam nte, te res lt rá más fácil comprenderlo

Redundancia

El lenguaje natural es redundante, tiene exceso de información. Se podrá medir ? Se necesitan algunas definiciones.

Ratio de un lenguaje (r_k)

Definimos el ratio de un lenguaje L para mensajes de longitud k como:

$$r_k = H_k(M) / \ k$$

$H_k(M)$: Entropía de todos los mensajes de longitud k que tienen sentido en L.

Estamos midiendo el número de bits de información que transporta cada carácter en mensajes de una longitud determinada. Para idiomas como el Español, r_k suele valer alrededor de 1;4 bits por letra para valores pequeños de k.

Redundancia

Ratio absoluto de un lenguaje (R)

Es el máximo número de bits de información que pueden ser codificados en cada carácter, asumiendo que todas las combinaciones de caracteres son igualmente probables. Suponiendo m símbolos diferentes en nuestro alfabeto este índice vale:

$$R = log_2(m^k) / \ k = (k \ log_2 \ m \ )/ \ k = log_2 \ m = log_2 \ 27 = 4{,}7 \ [bits/letra]$$

Luego parece que el nivel de redundancia de los lenguajes naturales es alto.

Redundancia

Se define como la diferencia entre las dos magnitudes anteriores:

$$D = R - r$$

También se define el índice de redundancia como el siguiente cociente:

$$I = D / R$$

para medir la auténtica redundancia de un lenguaje, hay que tener en cuenta secuencias de cualquier número de caracteres, por lo que la expresión de r debería calcularse en realidad como:

$$r_{\infty} = lim_{\ n \to \infty} \ (H_n(M) / \ n)$$

Aplicación de la Redundancia

En la compresión de datos

Los algoritmos tratan de eliminar la redundancia dentro de un archivo, considerando cada byte como un mensaje elemental, y codificándolo con más o menos bits según su frecuencia de aparición. En este sentido se trata de codificar exactamente la misma información que transporta el archivo original, pero empleando un número de bits lo más pequeño posible.

En los Códigos de Redundancia Cíclica (CRC)

Permiten introducir un campo de longitud mínima en el mensaje, tal que éste proporcione la mayor redundancia posible. Así, si el mensaje original resultase alterado, la probabilidad de que el CRC añadido siga siendo correcto es mínima.

Ataques de fuerza bruta

Conocidos los patrones de redundancia de un lenguaje, es posible dar de forma automática una estimación de si una cadena de símbolos corresponde o no a dicho lenguaje. Opción: comprimir el mensaje antes de codificarlo..!!

Desinformación y Distancia de Unicidad

Definiremos desinformación de un sistema criptográfico como la entropía condicionada del conjunto M de posibles mensajes sobre el conjunto C de posibles criptogramas:

$$H(M/C) = -\sum_{m \epsilon M} \sum_{c \epsilon C} P(c)P(m/c)\ log_2(P(m/c))$$

Esta expresión permite conocer la incertidumbre que queda sobre cuál ha sido el mensaje enviado m, suponiendo que conocemos su criptograma asociado c.

- Qué se puede inferir si la incertidumbre se mantiene después de conocer c ?
- Qué valor de desinformación ($H(M/C) \approx 0$ ó $H(M/C) \approx H(M)$)sería deseable ?

Desinformación y Distancia de Unicidad

También se puede medir en función del conjunto K de claves, y entonces representará la incertidumbre que nos queda sobre k conocida c:

$$H(K/C) = -\sum_{k \epsilon K} \sum_{c \epsilon C} P(c)P(k/c)\ log_2(P(k/c))$$

Definiremos finalmente la distancia de unicidad de un criptosistema como la longitud mínima de mensaje cifrado que aproxima el valor $H(K/C)$ a cero. En otras palabras, es la cantidad de texto cifrado que necesitamos para poder descubrir la clave.

- Qué valor de distancia unicidad tienen los criptosistemas seguros de Shannon ?
- Qué valor debería buscar que tenga un criptosistema que vaya a diseñar ?

Confusión y Difusión

Dos técnicas básicas para ocultar la redundancia en un texto claro que a pesar de su antigüedad poseen una importancia clave en la Criptografía moderna.

Confusión

Trata de ocultar la relación entre el texto claro y el texto cifrado. Recordemos que esa relación existe y se da a partir de la clave k empleada, puesto que si no existiera jamás podríamos descifrar los mensajes. El mecanismo más simple de confusión es la sustitución.

Difusión

Diluye la redundancia del texto claro repartiéndola a lo largo de todo el texto cifrado. El mecanismo más elemental para llevar a cabo una difusión es la transposición.

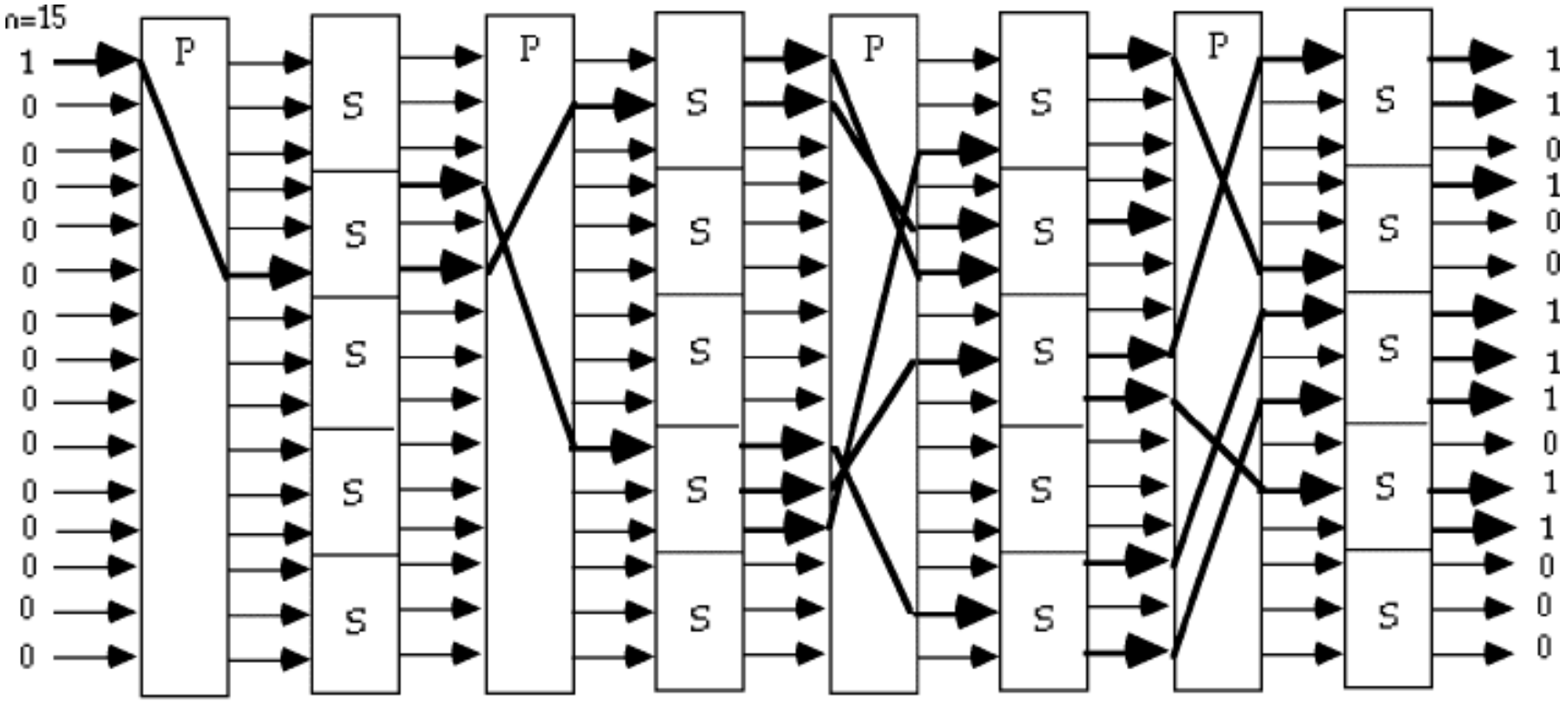
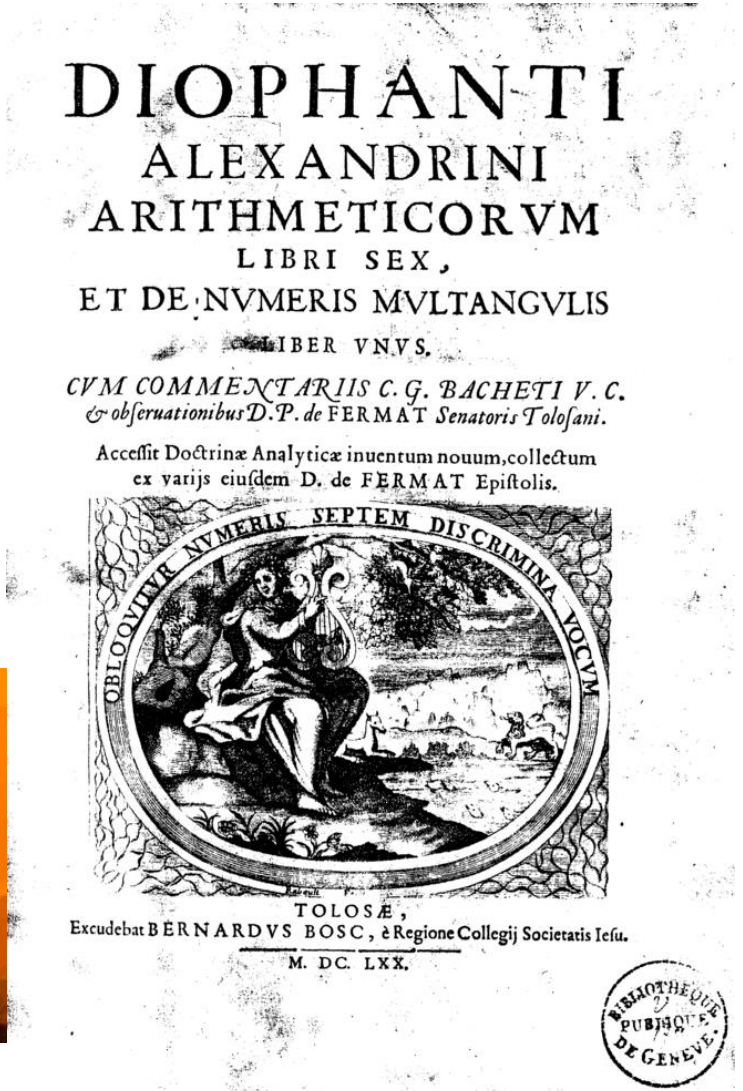


Fig 2.3 - Substitution-Permutation Network, with the Avalanche Characteristic

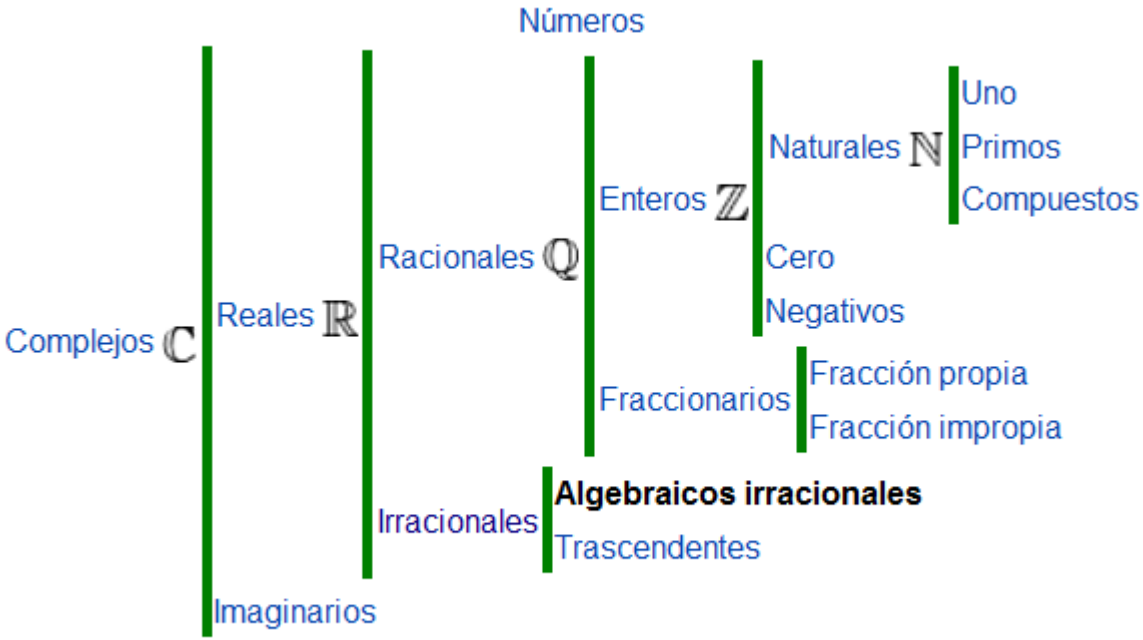
Teoría de números



Aritmética Modular

Qué es la Aritmética ?

La rama de la matemática cuyo objeto de estudio son los números y las operaciones elementales hechas con ellos: suma, resta, multiplicación y división.



Aritmética Modular

Definimos dos operaciones asociadas con el proceso de división. Dados ***a*** y ***b***, esas operaciones producen el cociente y el residuo del problema de la división.

Definición de división y módulo

Sean a y $b \in Z$ y $b > 0$. Se puede demostrar que ∃ un par único de números q y r tales que $a = qb + r$ y $0 \leq r < b$. Las operaciones ***div*** y ***mod*** se definen como sigue:

$$a \text{ div } b = q \qquad a \text{ mod } b = r$$

Ejemplos:

11 div 3 = 3

23 div 10 = 2

- 37 div 5 = -8

11 mod 3 = 2

23 mod 10 = 3

-37 mod 5 = 3

Aritmética Modular

La palabra mod tiene dos significados relacionados que son diferentes. Por un lado :

a ≡ ***b*** ***(mod*** n***)*** es que ***a – b*** es un múltiplo de n

Ejemplo:

$53 \equiv 3 \pmod{10} \rightarrow 53 - 3 = k10;$ con $k = 5$

$53 \bmod 10 = 3 \rightarrow$ residuo de la división.

Proposición

Con ***a, b*** y $n \in Z$ y $n > 0$, entonces:

$a \equiv b \pmod{n} \Leftrightarrow a \bmod n = b \bmod n$

$53 \equiv 23 \pmod{10} \Leftrightarrow 53 \bmod 10 = 23 \bmod 10 = 3$

Aritmética Modular

En lugar de aritmética con enteros o racionales, el nuevo conjunto en el que trabajaremos es $\boldsymbol{Z_n}$ siendo $\boldsymbol{n}$ es un entero positivo. El conjunto $\boldsymbol{Z_n}$ se define como sigue:

$$\boldsymbol{Z_n} = \{\ 0, 1, 2, \ldots , \boldsymbol{n-1}\ \}$$

Y llamaremos a este sistema como "enteros módulo n". Y definiremos las operaciones básicas:

⊕ suma mod n

⊗ multiplicación mod n

Θ resta mod n

∅ división mod n

Aritmética Modular

Suma, resta y multiplicación modular

Sea ***n*** un entero positivo, con ***a***, ***b*** $\in Z_n$, definiremos :

$$a \oplus b = (a + b) \bmod n$$

$$a \otimes b = (a\, b) \bmod n$$

Ejemplos: Sea n = 10, se cumple lo siguiente:

$5 \oplus 5 = 0$ $\qquad$ $9 \oplus 8 = 7$

$5 \otimes 5 = 5$ $\qquad$ $9 \otimes 8 = 2$

Suma, resta y multiplicación modular

Sea n un entero positivo, con $a, b \in Z_n$, definiremos $a \ominus b$ como la única $x \in Z_n$, tal que

$$a = b \oplus x$$

Que es equivalente a decir que

$$a \ominus b = (a - b) \bmod n$$

Aritmética Modular

Proposición: Sea n un entero tal que $n \geq 2$, se cumple:

$\forall\ a, b \in Z_n$,
$a \oplus b = b \oplus a$ (conmutación)
$a \otimes b = b \otimes a$

$\forall\ a, b, c \in Z_n$,
$a \oplus (b \oplus c) = (a \oplus b) \oplus c$ (asociatividad)
$a \otimes (b \otimes c) = (a \otimes b) \otimes c$

$\forall\ a \in Z_n$,
$a \oplus 0 = a$ (elem.identidad)
$a \otimes 1 = a$
$a \otimes 0 = 0$

$\forall\ a, b, c \in Z_n$,
$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ (distributiva)

División modular

Las ¾ partes de la aritmética modular es sencilla. La división modular es muy distinta a las demás operaciones. Por ejemplo, si a, b, c son enteros, y $a \neq 0$, entonces:

$$ab = ac \qquad \Rightarrow \qquad b = c$$

Pero ***en*** Z_{10}, que $5 \otimes 2 = 5 \otimes 4$ no implica que $2 = 4$, es más, $2 \neq 4$..!

A pesar que $5 \neq 0$, ***no podemos simplificar*** o dividir ambos lados por 5..!

Aritmética Modular

División modular

En **Q**, se puede expresar a/b como $a \cdot b^{-1}$ (multiplicación por el recíproco de b). Esto explica por qué la división por 0 es indefinida. No tiene recíproco.!!

El recíproco de un número racional x, es otro número racional y, tal que

$$x \cdot y = 1$$

Usamos este concepto para definir la división en Z_n…!

Debemos definir los recíprocos.

Definición

Sea n un entero positivo, y sea $a \in Z_n$. Un recíproco de a es un elemento $b \in Z_n$ tal que:

$$a \otimes b = 1.$$

Un elemento de Z_n que tenga un recíproco se llama ***invertible***.

Ejercicio:

Armar la tabla de producto, en Z_{10}. Ver qué valores tienen recíproco. Ídem en Z_9. Discutir conclusiones.

Aritmética Modular

División modular – Definición

Sea n un entero positivo, y sea b un elemento invertible de Z_n. Sea $a \in Z_n$ arbitrario. Entonces:

$a \oslash b$ se define como $a \otimes b^{-1}$

Ejemplo: En Z_{10}, $2 \oslash 7 = 2 \otimes 3 = 6$.

Para continuar debemos contestar las siguientes preguntas:

- ¿ Cuáles elementos son invertibles en Z_n ?
- ¿ Siendo b invertible en Z_n , cómo se calcula b^{-1} ?

Aritmética Modular

Teorema

Sea n un entero positivo, y sea $a \in Z_n$. Entonces a es invertible si y solo si a y n son primos relativos.

Definición

Sean dos enteros a y b. Se dice que a y b son primos relativos siempre y cuando su $mdc(a, b) = 1$.

Nota: sea n un entero positivo y sea $a \in Z_n$ arbitrario y supongamos que a sea invertible. Esto quiere decir que hay un elemento $b \in Z_n$ tal que $a \otimes b = 1$. En otras palabras: (ab) mod n = 1 es decir que $ab + kn = 1$ para algún entero k.

Definición de divisor común

Sean a y $b \in Z$. Se dice que un entero d es un divisor común de a y b siempre y cuando $d|a$ y $d|b$.

Definición de máximo divisor común (mcd)

Sean a y $b \in Z$. Un entero d es el mdc de a y b siempre y cuando:

- d sea divisor común de a y b, y
- Si c es divisor común de a y b, $d \geq c$.

El máximo divisor común de a y b se representa por: $mdc(a, b)$

Ejercicio: Calcular el mdc(30, 24).

Seguro que llegaremos a un algoritmo como el siguiente:

1. Suponer que ***a*** y ***b*** son enteros positivos
2. Para todo entero positivo ***k*** de 1 al menor de ***a*** y ***b***, ver si ***k***|***a*** y ***k***|***b***. Si es así poner ese número en una lista.
3. Elegir el número máximo de la lista. Es el ***mdc***.

Debemos pensar en un algoritmo para números moderadamente grandes como :

a = 34.902

b = 34.299.883

Aritmética Modular

Euclides, propuso una forma mas sencilla de obtener el mismo resultado:

Proposición: Si ***a*** y ***b*** son enteros positivos, y sea ***c*** = ***a*** mod ***b***, entonces

$$mdc(a, b) = mdc(b, c)$$

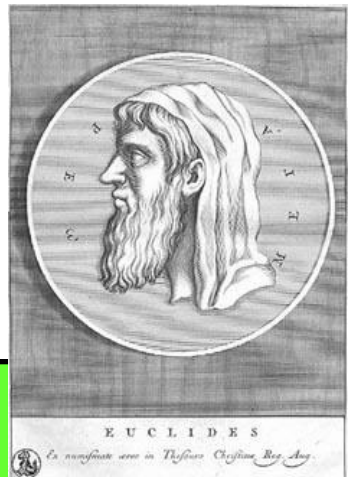
Ejemplo: Calcular el ***mdc(689, 234)***

mdc(689, 234) = ***mdc(234, c)***; $c = 689 \bmod 234 = 221$

mdc(234, 221) = ***mdc(221, c')***; $c' = 234 \bmod 221 = 13$

mdc(221, 13) = ***mdc(13, c'')***; $c'' = 221 \bmod 13 = 0$

$$mdc(689, 234) = mdc(234, 221) = mdc(221, 13) = 13$$



Teorema

Sean ***a*** y ***b*** enteros, ambos distintos de cero. El entero positivo mas pequeño de la forma ***a***x+***b***y donde x e y son enteros, es el ***mdc(a, b)***.

Ejercicio:

Con $\boldsymbol{a} = 30$ y $\boldsymbol{b} = 24$, construir una tabla de valores de $\boldsymbol{a}x+\boldsymbol{b}y$ para valores enteros de x e y entre –4 y 4. Determinar el valor mas pequeño de esta tabla.

Aritmética Modular

Ejercicio: Con ***a*** = 30 y ***b*** = 24, construir una tabla de valores de ***a***x+***b***y para valores enteros de x e y entre –4 y 4.

x/y	-4	-3	-2	-1	0	1	2	3	4
-4	-216	-186	-156	-126	-96	-66	-36	-6	24
-3	-192	-162	-132	-102	-72	-42	-12	18	48
-2	-168	-138	-108	-78	-48	-18	12	42	72
-1	-144	-114	-84	-54	-24	6	36	66	96
0	-120	-90	-60	-30	0	30	60	90	120
1	-96	-66	-36	-6	24	54	84	114	144
2	-72	-42	-12	18	48	78	108	138	168
3	-48	-18	12	42	72	102	132	162	192
4	-24	6	36	66	96	126	156	186	216

El valor que encontraremos es el 6..!

Si se ampliara la tabla, ¿ encontraría un valor menor ? NO..!

Aritmética Modular

Las preguntas:

Dados ***a*** y ***b***, cómo determinar los enteros x e y tal que ***a***x+***b***y = ***mdc(a, b)***

Quizás con algunas pruebas, podamos encontrar el resultado..!

Pero lo que nos interesa es resolver ***a***x + ***b***y = ***1*** ..!

Nuevamente Euclides nos da una pista:

Ejemplo:

Calcular x e y para 431x + 29y = 1; es el mdc(431, 29)

Ejemplo: Calcular x e y para 431x + 29y = mdc(431, 29) = 1

431 = 14 x 29 + 25	→	25 = 431 – 14 x 29
29 = 1 x 25 + 4	→	4 = 29 – 1 x 25
25 = 6 x 4 + 1	→	1 = 25 – 6 x 4
4 = 4 x 1 + 0		↓
	1 =	25 – 6 x (29 – 1 x 25) =
	=	25 – 6 x 29 + 6 x 25 =
	=	– 6 x 29 + 7 x 25 =
	=	– 6 x 29 + 7 x (431–14 x 29) =
	=	– 6 x 29 + 7 x 431– 7 x 14 x 29 =
	=	7 x 431 – 6 x 29 – 98 x 29 =
	=	7 x 431 – 104 x 29.-
		x = 7; y = -104

Aritmética Modular

Ejemplo: Calcular 29^{-1} en Z_{431}

Ya calculamos $431x + 29y = 1$

$x = 7;$

$y = -104$

$(-104 \times 29) \bmod 431 = 1,$

pero $-104 \notin Z_{431}$..!

En su lugar tomamos

$$b = -104 \bmod 431 = 327..!$$

Y tendremos:

$$29 \otimes 327 = 29 \,.\, 327 \bmod 431 = 9483 \bmod 431 = 1$$

$$29^{-1} = 327 \text{ en } Z_{431}$$

Aritmética Modular

Comenzamos acá en "División modular – Definición"

Sea n un entero positivo, y sea b un elemento invertible de Z_n. Sea $a \in Z_n$ arbitrario. Entonces:

$$a \oslash b \text{ se define como } a \otimes b^{-1}$$

Ejemplo: En Z_{10}, $2 \oslash 7 = 2 \otimes 3 = 6$.

Y pudimos contestar las siguientes preguntas:

- Cuáles elementos son invertibles en Z_n ..?
- Siendo b invertible en Z_n , cómo se calcula b^{-1},..?

Aritmética Modular

Función de Euler

Llamaremos conjunto reducido de residuos módulo n —y lo anotaremos como Z_n^*— al conjunto de números primos relativos con n. En otras palabras, Z_n^* es el conjunto de todos los números que tienen inversa módulo n.

Por ejemplo, si n fuera 12, su conjunto reducido de residuos sería {1, 5, 7, 11}.

Cómo calcular este valor ?

En el caso particular en que n fuese el producto de dos primos p y q,

$$\left| Z_n^* \right| = (p-1)(q-1)$$

Se define la función de Euler sobre n, a este cardinal.

$$\varphi(n) = \left| Z_n^* \right|$$

Teorema de Euler

Sea ***n*** un entero positivo, y sea ***a*** un entero primo relativo de ***n***. Entonces:

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

Ejemplo 39.5

Observemos que $\mathbf{Z}^*_9 = \{1, 2, 4, 5, 7, 8\}$ y que $\varphi(9) = 6$. Elevando los enteros del 1 al 9 a la potencia 6 (mod 9) se obtiene:

$$1^6 \equiv 1;\ 2^6 \equiv 1;\ \ 4^6 \equiv 1;\ 5^6 \equiv 1;\ 7^6 \equiv 1;\ \ 8^6 \equiv 1$$

Pequeño teorema de Fermat o primer teorema de Fermat

Se ***n*** un primo y sea ***a*** un entero. Entonces:

$$a^n \equiv a \pmod{n}$$

Aplicación a la criptografía asimétrica

Algoritmo RSA (Ronal **Rivest**, Adi **Shamir** y Leonard **Adleman**)

El criptosistema se basa en la extensión de Euler del primer teorema de Fermat.

Las funciones de cifrado y descifrado son:

$$E(M) = M^e \bmod n$$

$$D(N) = N^d \bmod n$$

Aplicación a la criptografía asimétrica

Algoritmo RSA (Ronal **Rivest**, Adi **Shamir** y Leonard **Adleman**)

Para generar un par de claves (K_P ;K_R), en primer lugar se eligen aleatoriamente dos números primos grandes, p y q. Luego se calcula el producto $n = pq$.

En segundo lugar elegimos un número e primo relativo con $(p - 1)(q - 1)$.

(e; n) será la clave pública.

Nótese que e debe tener inversa módulo $(p - 1)(q - 1)$, por lo que existirá un número d tal que:

$$ed \equiv 1 \bmod ((p - 1)(q - 1))$$

(d; n) será la clave privada.

Aplicación a la criptografía asimétrica

Roberto le revela los números $(\boldsymbol{e}; \boldsymbol{n})$, no sólo a Alicia, sino también a Eva. Mantiene en secreto $\boldsymbol{d}$.

Roberto	$\rightarrow$	$(\boldsymbol{e}; \boldsymbol{n})$	$\rightarrow$	Alicia
	$\leftarrow$	$\boldsymbol{N}$	$\leftarrow$	$\boldsymbol{N = E(M)}$

$$D(N) = D(E(M)) = M$$

$$D(E(M)) = D(M^e) = (M^e)^d = M^{ed}$$

Será posible que $M^{ed} = M$, trabajando en $\boldsymbol{Z_n}$ ..?

Cómo se logra esto..?

Aplicación a la criptografía asimétrica

Por Teorema de Euler, si $M \epsilon Z^*_n$, entonces

$$M^{\varphi(n)} = 1 \qquad ; en\ Z^*_n$$

$$(M^{\varphi(n)})^k = 1^k$$

$$M^{k\varphi(n)} = 1$$

$$M^{k\varphi(n)+1} = M$$

Luego, si $ed = k\varphi(n)+1$

se cumple $M^{ed} = M$

Aplicación a la criptografía asimétrica

Ejemplo 42.1 (Algoritmo RSA)

Roberto escoge los números primos p = 1231 y q = 337, y calcula

n = pq = 414847. También calcula $\varphi(n) = (p - 1)(q - 1) = 1230 \times 337 = 413280$.

Escoge $\boldsymbol{e}$ al azar en $\boldsymbol{Z}^*_{413280}$, por ejemplo $\boldsymbol{e}$ = 211243.

Por último, calcula, en $\boldsymbol{Z}^*_{413280}$, $\boldsymbol{d} = \boldsymbol{e}^{-1} = 166147$.

Las funciones de cifrado y descifrado de Roberto son:

$E(M) = M^{211243} \bmod 414847$ y $D(N) = N^{166147} \bmod 414847$

Suponemos el mensaje de Alicia M = 224455. En privado calcula lo siguiente:

$$E(M) = 224455^{211243} \bmod 414847 = 376682$$

y manda 376682 a Roberto. Él, en privado, calcula lo siguiente:

$$D(N) = 376682^{166147} \bmod 414847 = 224455$$

y conoce el mensaje de Alicia..!

Aplicación a la criptografía asimétrica

Algoritmo RSA

El atacante, si quiere recuperar la clave privada a partir de la pública, debe conocer los factores $\boldsymbol{p}$ y $\boldsymbol{q}$ a partir de $\boldsymbol{n}$, y esto representa un problema computacionalmente intratable, siempre que $\boldsymbol{p}$ y $\boldsymbol{q}$ —y, por lo tanto, $\boldsymbol{n}$— sean lo suficientemente grandes.

Aplicación a la criptografía asimétrica

Algoritmo RSA

El atacante, si quiere recuperar la clave privada a partir de la pública, debe conocer los factores $\boldsymbol{p}$ y $\boldsymbol{q}$ a partir de $\boldsymbol{n}$, y esto representa un problema computacionalmente intratable, siempre que $\boldsymbol{p}$ y $\boldsymbol{q}$ —y, por lo tanto, $\boldsymbol{n}$— sean lo suficientemente grandes.

¿ Computacionalmente intratable ?

¿Cuáles son las capacidades y limitaciones de los ordenadores ?

Hay tres áreas relacionadas con esta pregunta en la Teoría de la Computación:

- Autómatas,
- Complejidad,
- Computabilidad

Teoría de Autómatas

Se encarga de las definiciones y propiedades de los modelos matemáticos de computación (esenciales en áreas aplicadas de la informática).

Uno de estos modelos son los Autómatas Finitos, utilizados en:

- Procesamiento de textos
- Compiladores
- Diseño de Hardware.

Otro modelo son las Gramáticas Libres de Contexto, usadas en:

- Lenguajes de programación
- Inteligencia Artificial.

Teoría de la Complejidad

Trata de dar respuesta a la siguiente pregunta:

¿ Que hace a algunos problemas computacionalmente difíciles y a otros sencillos?

Tiene como finalidad la creación de mecanismos y herramientas capaces de describir y analizar la complejidad de un algoritmo y la complejidad intrínseca de un problema.

Teoría de la Computabilidad

Está muy relacionada con la Teoría de la Complejidad, ya que introduce varios de los conceptos que esta área utiliza. Su finalidad principal es la clasificación de diferentes problemas, así como formalizar el concepto de computar.

Estudia qué lenguajes son decidibles con diferentes tipos de "máquinas" y diferentes modelos formales de computación.

Brinda las bases para analizar los requerimientos computacionales de las técnicas de criptoanálisis. También ofrece las bases para estudiar las dificultades inherentes que se deben asumir para proveer de propiedades de seguridad a sistemas arbitrarios y para analizar la dificultad computacional de proteger la confidencialidad de ciertos datos en forma de estadísticas.

Complejidad algorítmica

La fortaleza de una cifra está determinada por la complejidad computacional de los algoritmos utilizados para resolver la cifra.

La complejidad computacional de un algoritmo se mide por sus requerimientos de tiempo (***T***) y espacio (***S***), donde ***T*** y ***S*** están expresado como una función de "***n***", donde "***n***" caracteriza el tamaño de la entrada.

Se puede decir que una función ***f(n)*** tiene un "orden de magnitud" de la forma ***O(g(n))*** (llamada notación "gran ***O***"), y se expresa como ***f(n) = O(g(n))*** cuando existen constante ***c*** y n_o tales que:

$$f(n) \leq c\,|g(n)| \qquad para \quad n \geq n_o$$

Complejidad algorítmica

Estimar los requerimientos de T y S de un algoritmo a partir del rendimiento de su "orden de magnitud" tiene la ventaja de independizarse del sistema.

No es necesario conocer el detalle de tiempos de ejecución de las diferentes instrucciones o el número de bits utilizados para representar diferentes tipos de datos.

Al mismo tiempo esto facilita la visualización de la relación entre los requerimientos de T y S y el valor de las entradas.

Por ejemplo, si $T = O(n^2)$, duplicando el tamaño de la entrada se cuadriplica el tiempo de corrida.

Complejidad algorítmica

Luego, es habitual clasificar los algoritmos por su complejidad de T (ó S).

Un algoritmo es polinomial (mas precisamente “de tiempo polinomial”) si su tiempo de corrida está dado por $T = O(n^t)$ para alguna constante t ; es constante si $t = 0$; lineal si $t=1$; cuadrático si $t=2$ y así.

Es exponencial si $T = O(t^{h(n)})$ para t constante y $h(n)$ un polinomio.

Luego para valores grandes de n, el comportamiento de un algoritmo puede tener enormes diferencias respecto a otros.

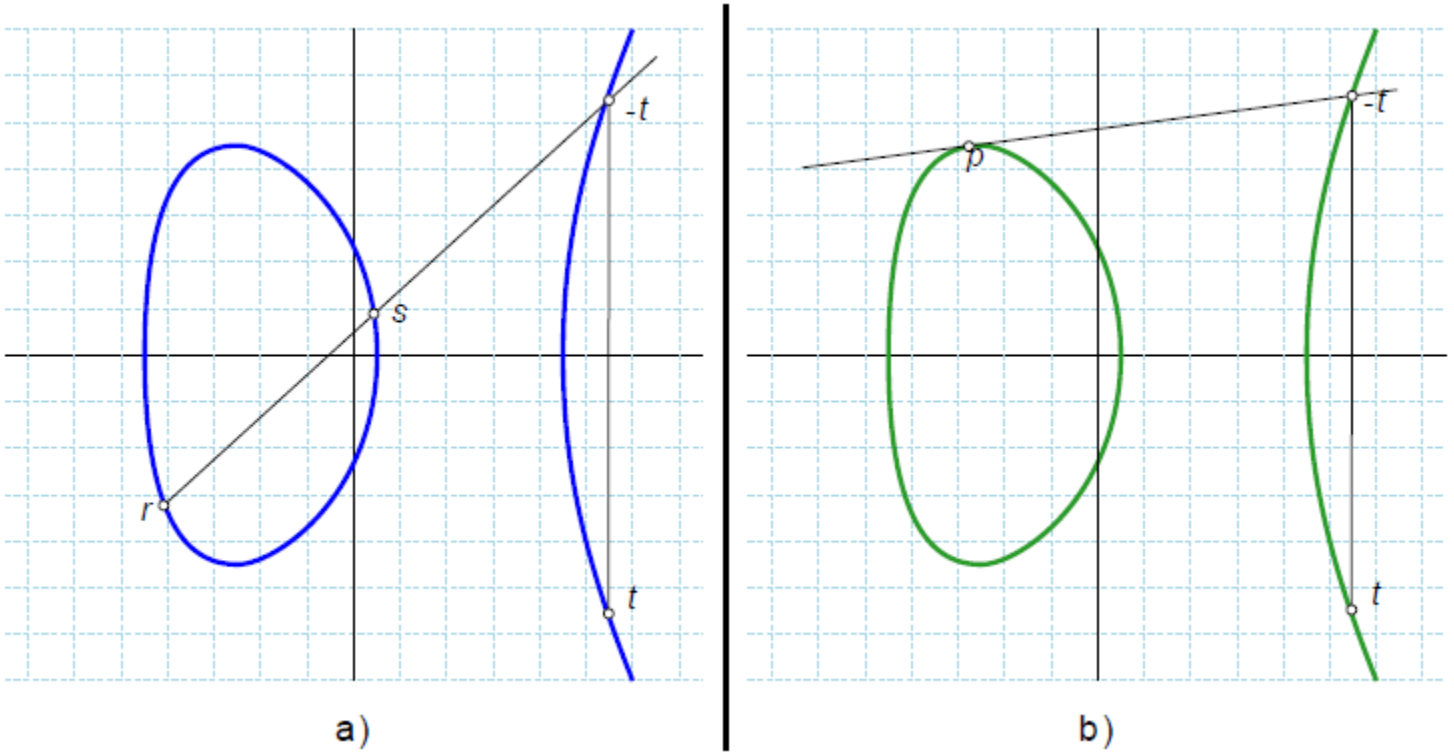
Complejidad algorítmica

Cuando n crece, la complejidad de un algoritmo puede presentar enormes diferencias..!!

CLASE	COMPLEJIDAD	# de op para $n=10^6$	T
Polinomial			
Constante	O(1)	1	1 μseg
Lineal	O(n)	10^6	1 seg
Cuadrática	$O(n^2)$	10^{12}	10 días
Cúbica	$O(n^3)$	10^{18}	23.397 años
Exponencial	$O(2^n)$	10^{301030}	10^{301016} años

Curvas elípticas en criptografía

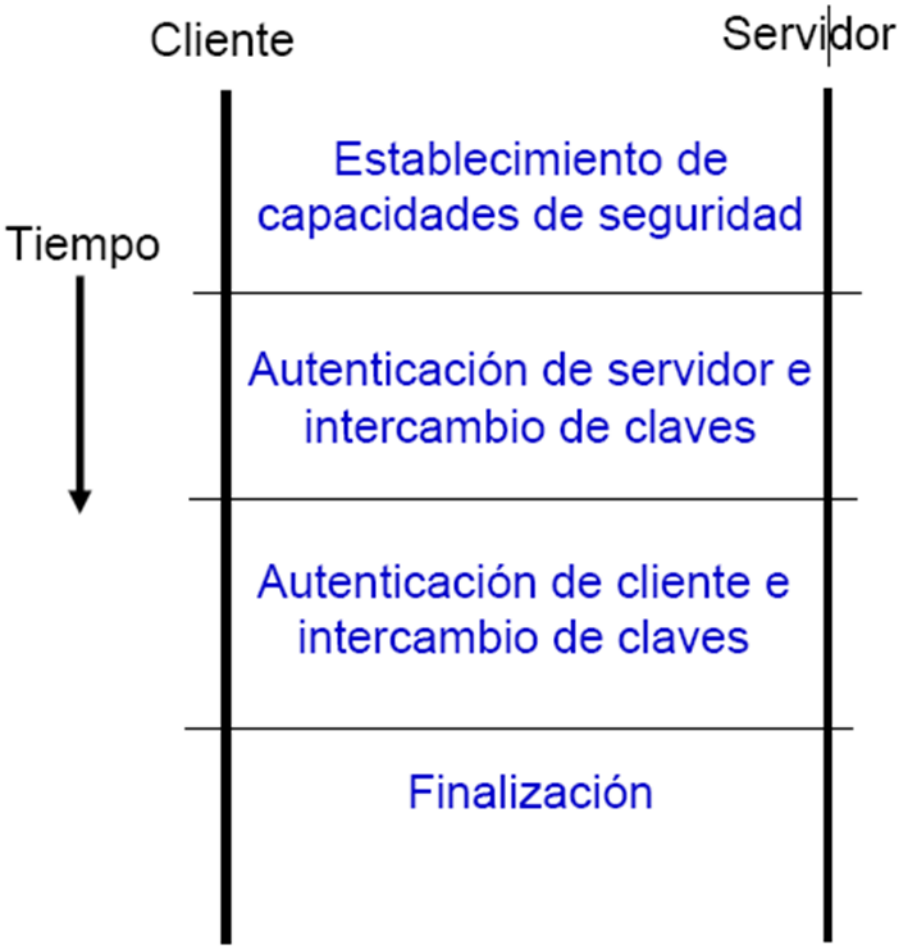
Las curvas elípticas constituyen un formalismo matemático conocido y estudiado desde hace más de 150 años, y presentan una serie de propiedades que dan lugar a problemas difíciles análogos a los que presentaba la aritmética modular, lo cual las hace válidas para aplicar algunos de los algoritmos asimétricos más conocidos.



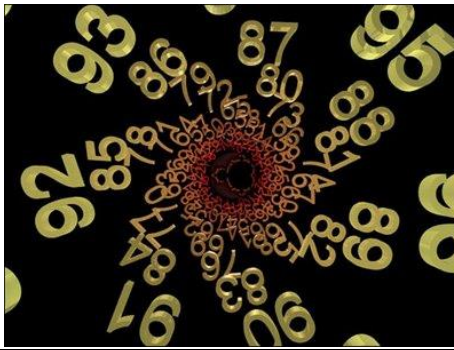
a) $y^2 = x^3 - 5x + 1$

b) $y^2 = x^3 - 3x + 4$

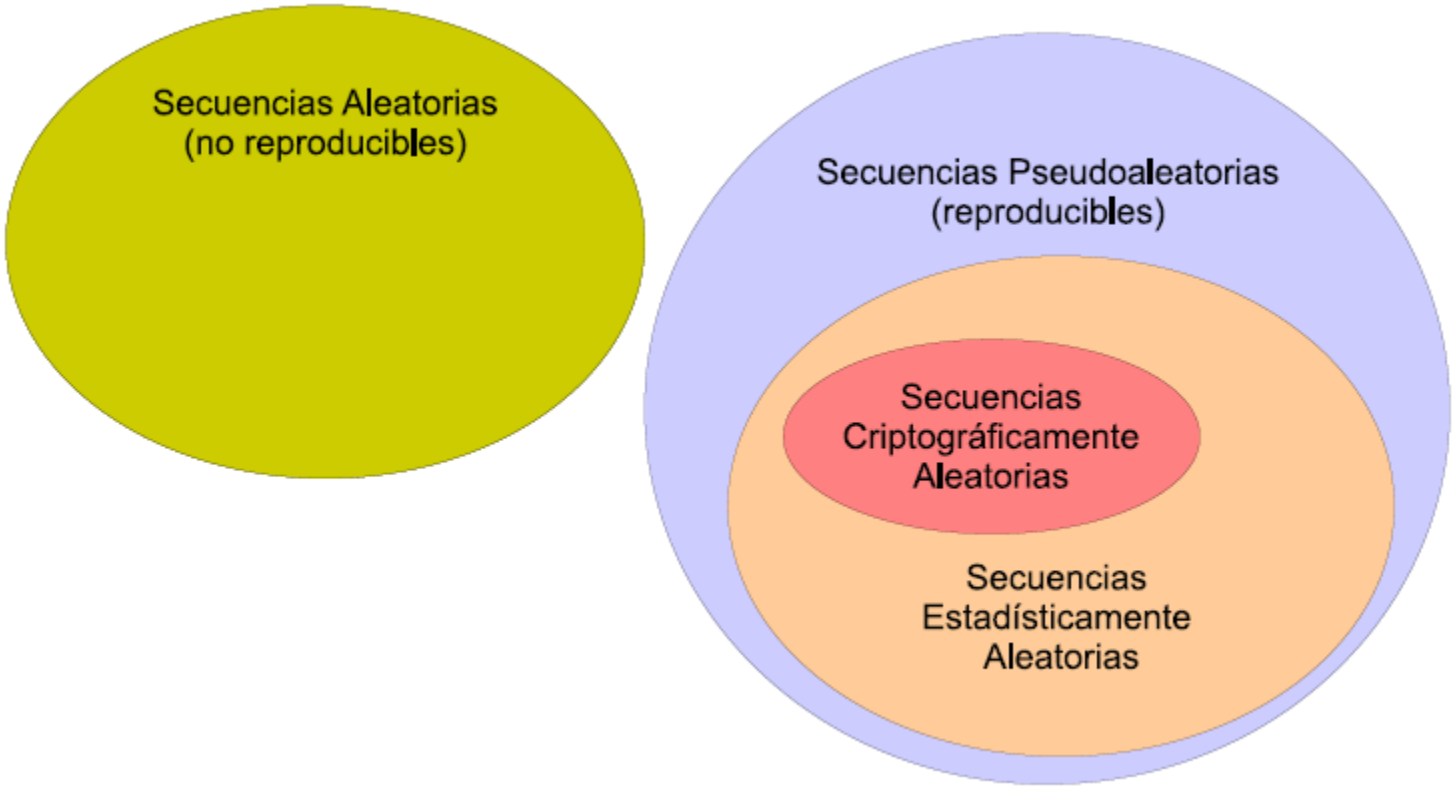
Números aleatorios



Para que una secuencia pseudoaleatoria sea criptográficamente aleatoria, ha de cumplir la propiedad de ser impredecible. Esto quiere decir que debe ser computacionalmente intratable el problema de averiguar el siguiente número de la secuencia, teniendo total conocimiento acerca de todos los valores anteriores y del algoritmo de generación empleado.



Números aleatorios



Clasificación de los distintos tipos de Secuencias Aleatorias


GRACIAS!