Teoría de Números

UCR – ECCI

CI-0111 Estructuras Discretas

Prof. Kryscia Daviana Ramírez Benavides

Introducción

- Esta presentación brinda una breve revisión de nociones de la teoría elemental de números, concernientes al conjunto de números enteros $Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ y al conjunto de números naturales $N = \{0, 1, 2, 3, \ldots\}$.
- Además, se da una pequeña introducción a la teoría computacional de números, utilizada en las aplicaciones criptográficas.

Múltiplos y Divisores

- Dados dos números enteros $a,b \in Z$, se dice que a es **divisible** entre b, o b es **divisor** de a, o a es **múltiplo** de b, cuando existe algún entero $c \in Z$ tal que $a = b \bullet c$.
- Se utiliza la notación $b \mid a$ para indicar que b es divisor de a, y la notación $b \nmid a$ para indicar que b no es divisor de a. Además, se utiliza la notación "a es b" para indicar a es múltiplo de b.
- Si se que $b \mid a$ y el entero c tal que $a = b \bullet c$ es único, entonces se dice que c es el **cociente exacto** de la división del dividendo a entre el divisor b y se escribe $c = a / b$.

Múltiplos y Divisores (cont.)

- Dos casos especiales:
  - Puesto $0 = 0 \bullet c$ para cualquier $c \in Z$, resulta que $0 \mid 0$, pero $0/0$ está indefinido porque c no es único.
  - Para cualquier entero $a \neq 0$, se tiene que $a \neq 0 \bullet c$ sea cual sea $c \in Z$; por lo tanto, cuando $a \neq 0$ se tiene que $0 \nmid a$ y $a/0$ está indefinido.
- **Teorema de la División.** Dados un entero $a \in Z$ y un entero positivo $b \in Z$, $b > 0$, existen dos enteros unívocamente determinados $c,r \in Z$ tales que $a = b \bullet c + r$ y $0 \leq r < b$.
  - Se utiliza la notación $c = a \; div \; b$, $r = a \; mod \; b$ (o a veces también $(c,r) = a \; divmod \; b$) para indicar el cálculo de c y r a partir de a y b.

Múltiplos y Divisores (cont.)

- En el caso de un dividendo entero $a \in Z$ y un divisor negativo $b \in Z$, $b < 0$, el cociente (c) y el residuo (r) de la división entera se definen como $a \; div \; b = (-a) \; div \; (-b)$, $a \; mod \; b = (-a) \; mod \; (-b)$.
- La división entera con divisor 0 no está definida. Además, en el caso particular de que el residuo $a \; mod \; b$ valga 0, el cociente entero $a \; div \; b$ coincide con el resultado de la división exacta a/b.
- Todo número entero a es divisible por sus **divisores triviales** 1 y a. Los **divisores no triviales** de a se llaman **factores** de a.
  - Divisores de 20 son 1, 2, 4, 5, 10 y 20, los factores son 2, 4, 5 y 10.

Múltiplos y Divisores (cont.)

- Sean $a,b,c \in Z$, se tienen las siguientes propiedades básicas:
  - $a \mid a$ (Propiedad Refleja).
  - Si $a \mid b$ y $b \mid c$, entonces $a \mid c$ (Propiedad Transitiva).
  - Si $a \mid b$, entonces $|a| \leq |b|$.
  - Si $a \mid b$ y $a \mid c$, entonces $a \mid \beta b + \gamma c \ \forall \beta,\gamma \in Z$.
  - Si $a \mid b$ y $a \mid b \pm c$, entonces $a \mid c$.
  - Si $a \mid b$ y $b \mid a$, entonces $|a| = |b|$.
  - Si $a \mid b$ y $b \neq 0$, entonces $b/a \mid b$.
  - Para $c \neq 0$, $a \mid b$ si y sólo si $ac \mid bc$.

Múltiplos y Divisores (cont.)

- Sean $a,b,c \in Z$, se tienen las siguientes propiedades básicas (cont.):
  - Si $a \mid bc$ y $mcd(a,b) = 1$, entonces $a \mid c$.
  - Si $mcd(a,b) = 1$ y c cumple que $a \mid c$ y $b \mid c$, entonces $ab \mid c$.
- Como $0 = 0 \cdot n$ y $n = n \cdot 1$ se tiene que $n \mid 0$ y $1 \mid n$ para todo $n \in Z$.
- Ver criterios de divisibilidad: http://es.wikipedia.org/wiki/Divisibilidad.
- Ver la tabla de divisores: http://es.wikipedia.org/wiki/Anexo:Tabla_de_divisores.

Congruencias y Aritmética Modular

- La **aritmética modular** puede ser construida matemáticamente mediante la **relación de congruencia** entre enteros, que es compatible con las operaciones en el anillo de enteros: suma, resta y multiplicación.
- Para un determinado módulo n, ésta se define de la siguiente manera:
  - a y b se encuentran en la misma **clase de congruencia módulo *n***, si ambos dejan el mismo resto al dividirlos por n, o, equivalentemente, si $a - b$ es un múltiplo de n.

Congruencias y Aritmética Modular (cont.)

- Esta relación se puede expresar cómodamente utilizando la notación de Gauss:
  - $a \equiv b \ (mod\ n)$ ó $a \equiv_n b$.
- Por ejemplo: $63 \equiv 83 \ (mod\ 10)$, ya que 63 y 83 dejan en mismo resto (3) al dividir por 10, o, equivalentemente, $63 - 83$ es un múltiplo de 10.
  - Se lee: 63 es congruente con 83, módulo 10, o, 63 y 83 son congruentes uno con otro, módulo 10.
- La **clase de congruencia de *a* módulo *n*** está definida como:
  - $[a]_n = \{b \in Z \mid a \equiv b \ (mod\ n)\} = \{a + k \bullet n \mid k \in Z\}$.

Congruencias y Aritmética Modular (cont.)

- Por ejemplo, cuando el módulo es 12, entonces cualesquiera dos números que divididos por 12 den el mismo resto son equivalentes (o "congruentes") uno con otro.
  - Los números ..., −34, −22, −10, 2, 14, 26,... son todos “congruentes módulo 12” unos con otros, ya que cada uno deja el mismo resto (2) cuando se divide por 12.
- La colección de todos esos números es una clase de congruencia.
  - Se puede pensar en un “peine” (finito si se quiere ver sólo unos números equivalentes alrededor del cero, o infinito si se quiere todos de una vez).

Congruencias y Aritmética Modular (cont.)

- El **conjunto cociente** $Z/\equiv (mod\ n)$, que se representa habitualmente como $Z/(n)$, tiene como elementos todas las clases $[a]_n$, para los diferentes $a \in Z$.
- Si $n > 0$, se definen **operaciones aritméticas módulo *n*** en $Z/(n)$, de manera que:
  - $[a]_n +_n [b]_n = [c]_n$, c es un entero tal que $a + b \equiv_n c$, donde $0 \leq a, b < n$ y $c = (a + b)\ mod\ n$.
  - $[a]_n -_n [b]_n = [c]_n$, c es un entero tal que $a - b \equiv_n c$, donde $0 \leq a, b < n$ y $c = (a - b)\ mod\ n$.
  - $[a]_n \bullet_n [b]_n = [c]_n$, c es un entero tal que $a \bullet b \equiv_n c$, donde $0 \leq a, b < n$ y $c = (a \bullet b)\ mod\ n$.

Propiedades de las Congruencias

- Reflexividad $\rightarrow a \equiv a(mod\ m), m|a - a = 0, \forall a \in \mathbb{Z}$
- Simetría $\rightarrow a \equiv b(mod\ m) \rightarrow b \equiv a(mod\ m), m|a - b \rightarrow m|b - a, \forall a, b \in \mathbb{Z}$
- Transitividad $\rightarrow a \equiv b(mod\ m)$ y $b \equiv c(mod\ m) \rightarrow a \equiv c(mod\ m), \forall a, b, c \in \mathbb{Z}$
- Si a es coprimo con m y $a \equiv b(mod\ m) \rightarrow b$ es coprimo con m
- Si $a \equiv b(mod\ m)$ y $k \in \mathbb{Z} \rightarrow$
  - $a + k \equiv b + k(mod\ m)$
  - $ak \equiv bk(mod\ m)$
  - $a^k \equiv b^k(mod\ m), k > 0$

Propiedades de las Congruencias (cont.)

- Si k es coprimo con $m \rightarrow$ existe un entero h^{-1} que $kh^{-1} \equiv 1(mod\ m) \rightarrow \frac{a}{k} \equiv \frac{b}{k}(mod\ m)$
  - $a/k = ak^{-1}$
- Como consecuencia de lo anterior, si se tiene dos congruencias con igual módulo $a \equiv b(mod\ m)$ y $c \equiv d(mod\ m) \rightarrow$
  - $a + c \equiv b + d(mod\ m)$
  - $a - c \equiv b - d(mod\ m)$
  - $ac \equiv bd(mod\ m)$

Números Primos y Compuestos

- Un número entero $p > 1$ se dice que es **primo** cuando los únicos divisores positivos de p son 1 y el propio p (o sea, los divisores triviales).
  - Hay una cantidad infinita de números primos.
  - La relación de la cantidad de números primos que no exceda de x y $x/\ln(x)$ tiende a 1 cuando x crece sin límite.

Números Primos y Compuestos (cont.)

- Un número entero $x > 1$ se llama **compuesto** cuando no es primo, o lo que es lo mismo, si existe una descomposición $x = k \bullet l$ que expresa a x como producto de dos enteros k y l tales que $l < k < x$.
- Si no se puede encontrar ningún divisor d de x que sea mayor que 1 y menor o igual que la raíz cuadrada por defecto de x, se puede asegurar que x es primo.
  - Si n es un número entero compuesto, entonces n tiene un divisor primo menor o igual a $\sqrt{n}$.

Números Primos y Compuestos (cont.)

- El número 1 (elemento neutro de la operación producto) se considera que no es ni primo ni compuesto.
- Los números negativos se dividen en tres clases: -1, los opuestos a los números primos y los opuestos a los números compuestos.
- La propiedad de ser primo se denomina **primalidad**, y el término primo se puede emplear como adjetivo.
  - A veces se habla de **número primo impar** para referirse a cualquier número primo mayor que 2, ya que éste es el único número primo par.
  - Se denota el conjunto de todos los números primos por *P*.

Números Primos y Compuestos (cont.)

- Propiedades más importantes y útiles de los números primos:
  - Siempre que un número primo p cumpla que $p \mid x_1 \bullet x_2 \bullet \ldots \bullet x_n$, se puede concluir que $p \mid x_i$ para algún i con $1 \leq i \leq n$.
  - Cualquier número entero $x \geq 1$ se puede descomponer como producto de factores primos; esta descomposición es única, salvo el orden de los factores. Esta propiedad se conoce como **teorema fundamental de la aritmética**. Se utiliza la notación $x = {p_1}^{e1} \bullet {p_2}^{e2} \bullet \ldots \bullet {p_n}^{en}$, donde los p_i son números primos y los e_i son exponentes naturales. En el caso $x = 1$ la descomposición se obtiene tomando $n = 0$ (producto vacío).
    - http://es.wikipedia.org/wiki/Anexo:Tabla_de_factores_primos.

Números Primos y Compuestos (cont.)

- Propiedades más importantes y útiles de los números primos (cont.):
  - Si p es cualquier número primo y q es cualquier divisor primo de $1 + p!$, se puede asegurar que $q > p$. Como consecuencia de esta propiedad, para cualquier número primo existe otro mayor y, por lo tanto, el conjunto de los números primos es infinito.

Números Primos y Compuestos (cont.)

- El cálculo de los números primos menores o iguales que una cota superior *n* dada se puede hacer por un procedimiento conocido como la **criba de Eratóstenes**, la cual se ejecuta del siguiente modo:
  - Escribir en una lista el número 2, seguido de todos los números impares menores o iguales que el límite *n* dado (se supone que $n > 3$).

Números Primos y Compuestos (cont.)

- Cálculo de los números primos menores o iguales que n (cont.):
  - Repetir el siguiente proceso, para ir tachando algunos números de la lista:
    - Considerar el primer número primo p de la lista que sea mayor que 2 y no esté tachado (al inicio es el 3), se debe asegurar que p sea primo.
    - Si se tiene $p^2 \leq n$, tachar de la lista todos los múltiplos de p (excepto él mismo), y volver a repetir el proceso. En caso contrario, $p^2 > n$, se puede asegurar que todos los números de la lista que no están tachados son primos, y terminar.
  - Los números sin tachar serán los primos que se buscaban.

Divisores Comunes

- Si d es un divisor de a y b, entonces d es un **divisor común** de a y b.
  - Divisores de 30 son 1, 2, 3, 5, 6, 10, 15 y 30; y divisores de 24 son 1, 2, 3, 4, 6, 8, 12 y 24. Los divisores comunes de 24 y 30 son 1, 2, 3 y 6.
- El número 1 es divisor común de cualquier par de números enteros.
- Propiedad importante de los divisores comunes:
  - $d \mid a$ y $d \mid b \Rightarrow d \mid (a + b)$ y $d \mid (a - b) \Rightarrow d \mid (ax + by)\ \forall x,y \in Z$.
  - Si $a \mid b \Rightarrow \mid a \mid \leq \mid b \mid$ o $b = 0$, $a \mid b$ y $b \mid a \Rightarrow a = \pm b$.

Máximo Común Divisor (MCD)

- El **máximo común divisor** (MCD) de dos enteros a y b, ambos no pueden ser 0 (sólo alguno de ellos), es el divisor común más grande de a y b; el cual se denota como $mcd(a,b)$.
  - $mcd(24,30) = 6$, $mcd(5,7) = 1$, $mcd(9,0) = 9$.
- Propiedades elementales de la función MCD:
  - $mcd(a,b) = mcd(b,a)$.
  - $mcd(a,b) = mcd(-a,b)$.
  - $mcd(a,b) = mcd(|\ a\ |,|\ b\ |)$.
  - $mcd(a,0) = mcd(0,a) = |\ a\ |,\ a \neq 0$.
  - $mcd(a,ka) = |\ a\ |,\ \forall k \in Z$.

Máximo Común Divisor (MCD) (cont.)

- **Teorema.** Para cualquier par de números enteros a y b, ambos no pueden ser 0 (sólo alguno de ellos), su $mcd(a,b)$ es el elemento positivo más pequeño del conjunto de combinaciones lineales de a y b $\{ax + by : x,y \in Z\}$.
  - **Prueba.** Sea s el valor más pequeños positivo de la combinación lineal de a y b, y sea $s = ax + by$ para algunas $x,y \in Z$. Sea $q = \lfloor a/s \rfloor$. Las propiedades de los divisores comunes implican $a \bmod s = a - qs = a - q(ax + by) = a(1 - qx) + b(-qy)$, y $a \bmod s$ es una combinación lineal de a y b. Pero, si $a \bmod s < s$ se tiene que $a \bmod s = 0$, porque s es el elemento positivo más pequeño del conjunto de combinaciones lineales. Por lo tanto, $s \mid a$ y $s \mid b$.

Máximo Común Divisor (MCD) (cont.)

- **Prueba.** (cont.) Entonces, s es un divisor común de a y b, y el $mcd(a,b) \geq s$. Las propiedades de los divisores comunes implican que $mcd(a,b) \mid s$, por lo que $mcd(a,b) \mid a$ y $mcd(a,b) \mid b$, y s es una combinación lineal de a y b. Pero que $mcd(a,b) \mid s$ y $s > 0$ implica que $mcd(a,b) \leq s$. Combinando $mcd(a,b) \geq s$ y $mcd(a,b) \leq s$ nos queda que $mcd(a,b) = s$; por lo que se concluye que s es el máximo común divisor de a y b.

Máximo Común Divisor (MCD) (cont.)

- **Corolario 1.** $\forall a,b \in Z$, si $d \mid a$ y $d \mid b \Rightarrow d \mid mcd(a,b)$.
  - Ya que $mcd(a,b)$ es una combinación lineal de a y b.
- **Corolario 2.** $\forall a,b \in Z$ y $\forall n \in Z+$, $mcd(an,bn) = n\ mcd(a,b)$.
  - Si $n = 0$, el corolario es trivial. Si $n > 0 \Rightarrow mcd(an,bn)$ es el elemento más pequeño positivo del conjunto $\{anx + bny\}$, por lo que n es el elemento más pequeño positivo del conjunto $\{ax + by\}$.
- **Corolario 3.** $\forall n,a,b \in Z$, si $n \mid ab$ y $mcd(a,n) = 1 \Rightarrow n \mid b$.

Máximo Común Divisor (MCD) (cont.)

- Encontrar el MCD de dos números:
  - Descomponer cada número en factores primos.
  - Tomar los factores comunes con su menor exponente.
  - Multiplicar los factores anteriores.
- Otra manera de encontrar el MCD de dos números es utilizando el **algoritmo de Euclides**.

Mínimo Común Múltiplo (MCM)

- El **mínimo común múltiplo** (MCM) de dos enteros a y b, ambos no pueden ser 0 (sólo alguno de ellos), es el múltiplo común más pequeño de a y b; el cual se denota como $mcm(a,b)$.
  - $mcm(24,30) = 120$, $mcm(5,7) = 35$, $mcm(9,0) = 0$.
- Propiedades elementales de la función MCM:
  - $mcm(a,b) = mcm(b,a)$.
  - $mcm(a,b) = mcm(-a,b)$.
  - $mcm(a,b) = mcm(|a|,|b|)$.
  - $mcm(a,0) = mcm(0,a) = 0$, $a \neq 0$.
  - Para $a > 0$ y $b > 0$, $mcd(a,b) \bullet mcm(a,b) = a \bullet b$.

Mínimo Común Múltiplo (MCM) (cont.)

- Encontrar el MCM de dos números:
  - Calcular el MCD de los números.
  - Multiplicar los números.
  - Dividir el producto de los números por el MCD.
- Otra manera de encontrar el MCM de dos números:
  - Factorizar los números.
  - Tomar todos los factores (comunes y no comunes) elevados a los mayores exponentes.
  - Multiplicar los factores anteriores.

Primos Relativos

- Dos números enteros *a* y *b* son **números primos entre sí** (o ***coprimos***, o ***primos relativos***), si no tienen ningún factor primo en común, o, dicho de otra manera, si el único divisor común es 1; o sea, si y sólo si $mcd(a,b) = 1$.
  - Por ejemplo, 6 y 35 son primos relativos, pero 6 y 27 no lo son porque ambos son divisibles por 3.
  - El 1 es primo relativo respecto de todos los enteros, mientras que 0 sólo lo es respecto de 1 y -1.
- Un medio rápido para determinar si dos números enteros son primos relativos es el **algoritmo de Euclides**.

Primos Relativos (cont.)

- **Teorema.** $\forall a,b,p \in Z$, si $mcd(a,p) = 1$ y $mcd(b,p) = 1 \Rightarrow mcd(ab,p) = 1$.
  - **Prueba.** Se tienen las siguientes ecuaciones: $ax + py = 1$ y $bx' + py' = 1$. Al multiplicar y ordenar las ecuaciones se obtiene: $ab(xx') + p(axy' + bx'y + pyy') = 1$. Por lo que 1 es un elemento positivo de la combinación lineal de ab y p.
- Se puede decir que los enteros $n_1, n_2, \ldots, n_k$ son parejas de primos relativos si, $i \neq j$, $mcd(n_i, n_j) = 1$.
- **Teorema de Bézout.** Los números enteros a y b son primos relativos cuando existen dos enteros x y y tales que $ax + by = 1$. De forma equivalente, b tiene un inverso para el producto módulo a, existe un número entero y tal que $by \equiv 1 \pmod{a}$.

Factorización Única

- El **teorema fundamental de la Aritmética** o **teorema de factorización única** afirma que todo entero positivo se puede representar de forma única como producto de factores primos.
  - $\forall a \in Z, a = p_1^{e1} \bullet p_2^{e2} \bullet \ldots \bullet p_n^{en}$, donde los p_i son números primos y los e_i son exponentes enteros positivos.
    - Ejemplo: $6000 = 2^4 \bullet 3 \bullet 5^3$.
- **Teorema.** Para todos los primos p y $\forall a,b \in Z$, si $p \mid ab \Rightarrow p \mid a$ o $p \mid b$ o ambos.

Cálculo del MCD – Algoritmo de Euclides

- El **algoritmo de Euclides** es un método antiguo y eficaz para calcular el MCD.
- El **algoritmo de Euclides extendido** es una ligera modificación que permite además expresar al máximo común divisor como una combinación lineal.
- Este algoritmo tiene aplicaciones en diversas áreas como álgebra, teoría de números y ciencias de la computación entre otras.
- Con unas ligeras modificaciones suele ser utilizado en computadoras electrónicas debido a su gran eficiencia.

Cálculo del MCD – Algoritmo de Euclides (cont.)

- Si se factoriza dos números enteros *a* y *b*:

$$\left.\begin{aligned} a &= p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n} \\ b &= p_1^{f_1} p_2^{f_2} \cdots p_n^{f_n} \end{aligned}\right\} mcd(a,b) = p_1^{\min(e_1,f_1)} p_2^{\min(e_2,f_2)} \cdots p_n^{\min(e_n,f_n)}$$

- **Teorema de recursión del MCD.** Para cualquier par de números enteros positivos *a* y *b* (ambos no pueden ser 0, sólo alguno de ellos) $mcd(a,b) = mcd(b, a \bmod b)$.
  - Ejemplo: $mcd(30,21) = mcd(21,9) = mcd(9,3) = mcd(3,0) = 3$.
- El método funciona también si *a* y *b* son negativos. Basta trabajar con los valores absolutos de estos números.

Cálculo del MCD – Algoritmo de Euclides (cont.)

- La aplicación recursiva del **lema de Euclides** (o teorema de recursión del MCD) proporciona un método para calcular el MCD, y se llama **algoritmo de Euclides**.
- **Algoritmo de Euclides.** Dados dos enteros a y b tales que $a \geq b > 0$, el algoritmo va calculando valores a_i, b_i, c_i y r_i, asociados a valores crecientes de un índice $i \geq 0$.

Cálculo del MCD – Algoritmo de Euclides (cont.)

- El algoritmo funciona de la siguiente manera:
  - Se comienza calculando $a_0 = a$ y $b_0 = b$.
  - Calculados a_i y b_i, para un cierto subíndice i, puede ocurrir:
    - Si $b_i = 0$, el cálculo termina. Se toma $d = a_i$ y se puede asegurar que $d = mcd(a,b)$.
    - Si $b_i > 0$, se calcula $c_i = a_i \ div \ b_i$ y $r_i = a_i \ mod \ b_i$ y se continua con $a_{i+1} = b_i$, $b_{i+1} = r_i$.
- En la práctica, los cálculos necesarios para ejecutar el algoritmo se pueden organizar en una tabla con varias columnas, en las cuales se van registrando los valores de i, a_i, b_i, c_i y r_i.

Cálculo del MCD – Algoritmo de Euclides (cont.)

Algoritmo de Euclides tradicional implementado de manera recurrente

Función $\mathrm{mcd}\,(a, b)$:

Si $b = 0$ **entonces:**

El resultado es a

En otro caso:

El resultado es $\mathrm{mcd}\,(b, a \bmod b)$

Algoritmo de Euclides tradicional implementado de manera iterativa

Función $\mathrm{mcd}\,(a, b)$:

Mientras $b \neq 0$ **haga lo siguiente:**

$(a, b) \leftarrow (b, a \bmod b)$

El resultado es a

Cálculo del MCD – Algoritmo de Euclides (cont.)

```
int mcd(int a, int b)
{
   int r[1000]; //Reservamos 1000 espacios para el Array que va a guardar los restos sucesivos
   for(int i = 0; i < 1000; i++) //Inicializamos los valores a 0
   {
      r[i]=0;
   }
   int i = 1;
   r[0] = a; //Damos al resto r[0] el valor de a
   r[1] = b; //Damos al resto r[1] el valor de b
   while(r[i] != 0) //Vamos calculando los restos hasta llegar al mcd
   {
      r[i+1] = r[i-1] % r[i];
      i++;
   }
   int mcd = r[i-1]; //Damos el valor d r[i-1] al mcd
   return mcd; //Se retorna el mcd
}
```

Cálculo del MCD – Algoritmo de Euclides (cont.)

- **Teorema de Lamé.** $\forall k \in Z$ y $k \geq 1$, si $a > b \geq 1$ y $b < F_{k+1} \Rightarrow$ la función euclidiana $mcd(a,b)$ realiza k recursiva llamadas.
- El **teorema de Bézout** afirma que el MCD de dos números enteros se puede expresar como combinación lineal de dichos números con coeficientes enteros; es decir, dados $\forall a,b \in Z$ tales que $d = mcd(a,b)$, se pueden encontrar dos coeficientes enteros $\exists m,n \in Z$ de manera que se cumpla $d = am + bn$, o equitativamente $d = ma + nb$.
- Esto teoremas generan el algoritmo de Euclides extendido.

Cálculo del MCD – Algoritmo de Euclides (cont.)

- El algoritmo de Euclides extendido permite, además de encontrar el MCD de dos números enteros a y b, expresarlo como una combinación lineal, es decir, encontrar números enteros x y y tales que $d = mcd(a,b) = ax + by$, $\forall x,y \in Z$.
- Este algoritmo retorna una tripleta (d,x,y).
  - Ejemplo: $Euclides(99,78) = (3,-11,14) \Rightarrow mcd(99,78) = 3 = 99 \bullet -11 + 78 \bullet 14$.
- $\forall a,b \in Z+$, si $a > b > 0 \Rightarrow$ la función realiza $O(\log b)$ recursiva llamadas.
- En el cálculo práctico de m y n se parte de la tabla utilizada para el cálculo de $d = mcd(a,b)$ por el algoritmo de Euclides.

Cálculo del MCD – Algoritmo de Euclides (cont.)

- Suponga que k sea el valor del índice i con el que ha terminado el cálculo de $d = mcd(a,b)$, se van calculando valores m_i y n_i asociados a valores decrecientes de un índice i, comenzando por $i = k$, del siguiente modo:
  - Se comienza calculando $m_k = 1$ y $n_k = 0$.
  - Luego, los valores i comprendidos entre $k - 1$ y 0 se recorren en orden decreciente y para cada uno de ellos se calcula $m_i = n_{i+1}$ y $n_i = m_{i+1} - n_{i+1} \bullet c_i$.
  - Se toman $m = m_0$ y $n = n_0$.
- La tabla utilizada en el algoritmo de Euclides se le agrega dos columnas, en las que se van registrando los valores m_i y n_i.

Cálculo del MCD – Algoritmo de Euclides (cont.)

Algoritmo de Euclides extendido implementado de manera recurrente

Función $Euclides\ (a, b)$:

Si $b = 0$ **entonces:**

El resultado es $(a, 1, 0)$

En otro caso:

$(d, s, t) \leftarrow Euclides\ (b, a \bmod b)$

El resultado es $(d, t, s - (a \div b)\ t)$

Algoritmo de Euclides extendido implementado de manera iterativa

Función $Euclides\ (a, b)$:

$(s, t, s', t') \leftarrow (1, 0, 0, 1)$

Mientras $b \neq 0$ **haga lo siguiente:**

Divida a entre b para obtener un cociente q y un residuo r

$(a, s, t, b, s', t') \leftarrow (b, s', t', r, s - s'\ q, t - t'\ q)$

El resultado es (a, s, t)

Cálculo del MCD – Algoritmo de Euclides (cont.)

Algoritmo de Euclides extendido implementado de manera iterativa con matrices

Función $Euclides\ (a, b)$:

$$Q \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Mientras $b \neq 0$ **haga lo siguiente:**

Divida a entre b para obtener un cociente q y un residuo r

$$Q \leftarrow \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix} \times Q$$

$$(a, b) \leftarrow (b, r)$$

El resultado es $(a, Q_{1\ 1}, Q_{1\ 2})$

Cálculo del MCD – Algoritmo de Euclides (cont.)

- El algoritmo de Euclides extendido se aplica en:
  - Simplificar fracciones.
  - Representar fracciones continuas.
  - Calcular inversos modulares.

Cálculo del MCD – Algoritmo de Euclides (cont.)

i	Residuos r_i	Cocientes c_i	m_i	n_i	Combinación Lineal $d = mcd(a,b) = ax + by$
0	a	*	1	0	a
1	b	*	0	1	b
2	$r_{i-2}\ mod\ r_{i-1}$	r_{i-2} / r_{i-1}	$m_{i-2} - c_i * m_{i-1}$	$n_{i-2} - c_i * n_{i-1}$	$r_{i-2}\ mod\ r_{i-1} = a*m_i + b*n_i$

Cálculo del MCD – Algoritmo de Euclides (cont.)

i	Residuos r_i	Cocientes c_i	m_i	n_i	Combinación Lineal $d = mcd(a,b) = ax + by$
0	662	*	1	0	662 = 662*1 + 414*0
1	414	*	0	1	414 = 662*0 + 414*1
2	248	1	1	-1	248 = 662*1 + 414*-1
3	166	1	-1	2	166 = 662*-1 + 414*2
4	82	1	2	-3	82 = 662*2 + 414*-3
5	**2**	2	**-5**	**8**	2 = 662*-5 + 414*8
6	0	41	207	-331	0 = 662*207 + 414*-331

Cálculo del MCD – Algoritmo de Euclides (cont.)

i	Residuos r_i	Cocientes c_i	m_i	n_i	Combinación Lineal $d = mcd(a,b) = ax + by$
0	252	*	1	0	252 = 252*1 + 198*0
1	198	*	0	1	198 = 252*0 + 198*1
2	54	1	1	-1	54 = 252*1 + 198*-1
3	36	3	-3	4	36 = 252*-3 + 198*4
4	**18**	1	**4**	**-5**	18 = 252*4 + 198*-5
5	0	2	-11	14	0 = 252*-11 + 198*14

Exponenciación Modular

- En criptografía es importante encontrar b^n **mod** m de forma eficiente, cuando b, n y m son enteros muy grandes.
- Por lo que se utiliza el algoritmo de exponenciación modular para tal fin.

ALGORITHM 5 Modular Exponentiation.

procedure *modular exponentiation*(b: integer, $n = (a_{k-1}a_{k-2}\ldots a_1a_0)_2$,
m: positive integers)
$x := 1$
$power := b$ **mod** m
for $i := 0$ **to** $k - 1$
  if $a_i = 1$ **then** $x := (x \cdot power)$ **mod** m
  $power := (power \cdot power)$ **mod** m
return x {x equals b^n **mod** m}

Exponenciación Modular (cont.)

EXAMPLE Use Algorithm 5 to find 3^{644} **mod** 645.

Solution: Algorithm 5 initially sets $x = 1$ and $power = 3$ **mod** $645 = 3$. In the computation of 3^{644} **mod** 645, this algorithm determines 3^{2^j} **mod** 645 for $j = 1, 2, \ldots, 9$ by successively squaring and reducing modulo 645. If $a_j = 1$ (where a_j is the bit in the jth position in the binary expansion of 644, which is $(1010000100)_2$), it multiplies the current value of x by 3^{2^j} **mod** 645 and reduces the result modulo 645. Here are the steps used:

$i = 0$: Because $a_0 = 0$, we have $x = 1$ and $power = 3^2$ **mod** $645 = 9$ **mod** $645 = 9$;
$i = 1$: Because $a_1 = 0$, we have $x = 1$ and $power = 9^2$ **mod** $645 = 81$ **mod** $645 = 81$;
$i = 2$: Because $a_2 = 1$, we have $x = 1 \cdot 81$ **mod** $645 = 81$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;
$i = 3$: Because $a_3 = 0$, we have $x = 81$ and $power = 111^2$ **mod** $645 = 12{,}321$ **mod** $645 = 66$;
$i = 4$: Because $a_4 = 0$, we have $x = 81$ and $power = 66^2$ **mod** $645 = 4356$ **mod** $645 = 486$;
$i = 5$: Because $a_5 = 0$, we have $x = 81$ and $power = 486^2$ **mod** $645 = 236{,}196$ **mod** $645 = 126$;
$i = 6$: Because $a_6 = 0$, we have $x = 81$ and $power = 126^2$ **mod** $645 = 15{,}876$ **mod** $645 = 396$;
$i = 7$: Because $a_7 = 1$, we find that $x = (81 \cdot 396)$ **mod** $645 = 471$ and $power = 396^2$ **mod** $645 = 156{,}816$ **mod** $645 = 81$;
$i = 8$: Because $a_8 = 0$, we have $x = 471$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;
$i = 9$: Because $a_9 = 1$, we find that $x = (471 \cdot 111)$ **mod** $645 = 36$.

This shows that following the steps of Algorithm 5 produces the result 3^{644} **mod** $645 = 36$.

Inverso Aritmética Modular

- Si a y m son números enteros y primos relativos, y $m > 1$, entonces un inverso de $a \bmod m$ existe. Además, el inverso es único módulo m.
- En la solución de la ecuación: $ax \equiv b \ (mod\ m)$, donde $a,m > 0$, sería decir $aa^{-1} \equiv 1 \ (mod\ m)$, ya que la multiplicación de un número y su número es 1.
  - Z_m
  - $[r]_m = \{c \in Z \mid r \equiv c \ (mod\ m)\} = \{r + k \bullet m \mid k \in Z\}$.
  - $r < m$

Inverso Aritmética Modular (cont.)

- Ejemplo: Encontrar el inverso de 3 *mod* 7
  - 3 y 7 son primos relativos, sí hay inverso
  - $mcd(3,7) = 1$, por lo que $3x + 7y = 1$
  - Por el algoritmo de Euclides extendido: $x = -2$ y $y = 1$
  - $7 + -2 = 5$
  - $Z_7 \rightarrow [0]_7, [1]_7, [2]_7, [3]_7, [4]_7, [5]_7, [6]_7$.
  - $aa^{-1} \equiv 1 \ (mod\ m) \rightarrow 3*5 \equiv 1 \ (mod\ 7) \rightarrow 15 \equiv 1 \ (mod\ 7)$

Ecuaciones Lineales Modulares

- Se desea encontrar soluciones a la ecuación: $ax \equiv b \ (mod\ n)$, donde $a > 0$ y $n > 0$.
- Sea $\langle a \rangle$ el subgrupo de Z_n generado por a.
  - Ya que $\langle a \rangle = \{a^{(x)} : x > 0\} = \{ax \ mod\ n : x > 0\}$, la ecuación anterior se soluciona si y sólo si $b \in \langle a \rangle$.
  - El teorema de Lagrange dice que $|\langle a \rangle|$ puede ser un divisor de n.
- **Teorema.** Para cualesquier enteros positivos a y n, si $d = mcd(a,n)$, entonces $\langle a \rangle = \langle d \rangle = \{0, d, 2d, \ldots, ((n/d) - 1)d\}$, en Z_n, y así $|\langle a \rangle| = n/d$.

Ecuaciones Lineales Modulares (cont.)

- **Corolario 1.** La ecuación $ax \equiv b \pmod{n}$ se resuelve para un x desconocido si y sólo si $mcd(a,n) \mid b$.
- **Corolario 2.** La ecuación $ax \equiv b \pmod{n}$ tiene distintas soluciones d módulo n, donde $d = mcd(a,n)$, o no tiene soluciones.
- **Teorema.** Sea $d = mcd(a,n)$, y suponer que $d = ax' + ny'$ para cualesquier números x' y y' (como computando el algoritmo extendido de Euclides). Si $a \mid b$, entonces la ecuación tiene el valor x_0 como una de sus soluciones , donde $x_0 = x'(b/d) \bmod n$.

$$ax_0 \equiv ax'(b/d) \pmod{n}$$
$$ax_0 \equiv d(b/d) \pmod{n}, (\text{porque } ax' \equiv d \pmod{n})$$
$$ax_0 \equiv b \pmod{n}$$

Ecuaciones Lineales Modulares (cont.)

- **Teorema.** Suponga que la ecuación $ax \equiv b \pmod{n}$ se puede resolver (ya que $d \mid b$, donde $d = mcd(a,n)$) y que x_0 es una solución de la ecuación. Entonces, esta ecuación tiene exactamente d soluciones distintas, módulo n, que se obtienen por $x_i = x_0 + i(n/d)$ para $i = 0, 1, 2, \ldots, d - 1$.

$$ax_i \bmod n = a(x_0 + in/d) \bmod n$$
$$ax_i \bmod n = (ax_0 + ain/d) \bmod n$$
$$ax_i \bmod n = ax_0 \bmod n, (\text{porque } d|a)$$
$$ax_i \bmod n = b$$

Ecuaciones Lineales Modulares (cont.)

- **Corolario 1.** Para cualquier $n > 1$, si $mcd(a,n) = 1$, entonces la ecuación $ax \equiv b \ (mod\ n)$ tiene una única solución, módulo n.
- **Corolario 2.** Para cualquier $n > 1$, si $mcd(a,n) = 1$, entonces la ecuación $ax \equiv 1 \ (mod\ n)$ tiene una única solución, módulo n. En otro caso, no tiene solución.
  - Este corolario permite usar la notación $(a^{-1}\ mod\ n)$ para referirse a la multiplicación inversa de a, módulo n, cuando a y n son primos relativos. Si $mcd(a,n) = 1$, entonces una solución de la ecuación es un entero x calculado por el algoritmo extendido de Euclides, ya que la ecuación $mcd(a,n) = 1 = ax + ny$ implica $ax \equiv 1 \ (mod\ n)$.

Ecuaciones Lineales Modulares (cont.)

```
MODULAR-LINEAR-EQUATION-SOLVER(a,b,n)
1 (d,x',y') <-- EXTENDED-EUCLID(a,n)
2 if d|b
3    then x0 <-- x'(b/d) mod n
4         for i <-- 0 to d-1
5             do print(x0 + i(n/d)) mod n
6    else print "No solutions"
```

Ecuaciones Lineales Modulares (cont.)

- Ejemplo:
  - Sea la ecuación $14x \equiv 30 \ (mod\ 100)$, donde $a = 14$, $b = 30$ y $n = 100$.

$$\text{Línea } 1 \Rightarrow (d, x, y) = (2, -7, 1)$$

$$\text{Línea } 2 \Rightarrow d \mid b = 2 \mid 30 = 15 \Rightarrow \text{Se ejecutan las líneas } 3 - 5.$$

$$\text{Línea } 3 \Rightarrow x_0 = (-7 \bullet (30/2)) \bmod 100 = (-7 \bullet 15) \bmod 100 = 95$$

$$\text{Línea } 4 \Rightarrow \text{El ciclo de la líneas } 4 - 5 \text{ imprme dos soluciones : } 95 \text{ y } 45.$$

$$x_1 = (95 + 0 \bullet (100/2)) \bmod 100 = 95$$

$$x_2 = (95 + 1 \bullet (100/2)) \bmod 100 = (95 + 50) \bmod 100 = 145 \bmod 100 = 45$$

Ecuaciones Lineales Modulares
Algoritmo de Euclides Extendido e Inverso Aritmético Modular

- Se puede usar el algoritmo de Euclides Extendido para encontrar el inverso aritmético modular y así encontrar la solución de la ecuación lineal modular.
  - Sea la ecuación $11x \equiv 6 \ (mod\ 92)$.
  - El inverso de 11 $(mod\ 92) = -25 + 92 = 67$
    - $67*11 \equiv 1 \ (mod\ 92)$
  - Se multiplica por 67 a ambos lados de la congruencia lineal:
    - $67*11x = 67*6 \ (mod\ 92)$
    - $x = 402 \ (mod\ 92) \rightarrow x = 34$
  - $x = 34$ es el número entero positivo más pequeño que soluciona el sistema.
  - Las soluciones son todas las x tales que $x = 34 + 92k$, $k \in \mathbb{Z}$.

Teorema Chino del Resto

- El **Teorema Chino del Resto** establece que cuando los módulos de un sistema de congruencias lineales están entre pares de primos relativos, hay una solución única del sistema de congruencia módulo el producto de los módulos.

Teorema Chino del Resto (cont.)

- Supongamos que $m_1, m_2, \ldots, m_k$ son enteros positivos coprimos dos a dos. Entonces, para enteros dados $a_1, a_2, \ldots, a_k$, existe un entero x que resuelve el sistema de congruencias simultáneas

$$
\begin{aligned}
&x \equiv a_1 \pmod{m_1} && M_k = m/m_k, k = 1,2,\ldots,n \\
&x \equiv a_2 \pmod{m_2} && M_k y_k \equiv 1 \pmod{m_k} \\
&\ldots && x = a_1 M_1 y_1 + a_2 M_2 y_2 + \ldots + a_n M_n y_n \\
&x \equiv a_n \pmod{m_n} && x \equiv a_k M_k y_k \equiv a_k \pmod{m_k}, k = 1,2,\ldots,n
\end{aligned}
$$

- Todas las soluciones x de este sistema son congruentes módulo el producto $m = m_1 m_2 \ldots m_n$, donde $0 \le x \le m$ y otras soluciones son congruentes módulo m.

Teorema Chino del Resto (cont.)

- Se tiene el siguiente sistema:

$$x \equiv 2 (\mathrm{mod}\, 3)$$
$$x \equiv 3 (\mathrm{mod}\, 5)$$
$$x \equiv 2 (\mathrm{mod}\, 7)$$

$m = 3*5*7 = 105, M_1 = 105/3 = 35, M_2 = 105/5 = 21, M_3 = 105/7 = 15$

$mcd(35,3) = 1, \text{inverso } 2$

$mcd(21,5) = 1, \text{inverso } 1$

$mcd(15,7) = 1, \text{inverso } 1$

$x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 = 2*35*2 + 3*21*1 + 2*15*1 = 233$

$x = 233 (\mathrm{mod} 105) = 23$

  - Solución $x = 23$, número entero positivo más pequeño que soluciona el sistema.
  - Las soluciones son todas las x tales que $x = 23 + 105k$, $k \in \mathbb{Z}$.

Método Sustitución hacia Atrás

- Se tiene el siguiente sistema:
  - $x \equiv 1 \pmod 5 \rightarrow x = 5t + 1$
  - $x \equiv 2 \pmod 6 \rightarrow 5t + 1 \equiv 2 \pmod 6 \rightarrow t \equiv 5 \pmod 6$
  - $t \equiv 5 \pmod 6 \rightarrow t = 6u + 5$
  - $x = 5t + 1 \rightarrow x = 5(6u + 5) + 1 \rightarrow x = 30u + 26$
  - $x \equiv 3 \pmod 7 \rightarrow 30u + 26 \equiv 3 \pmod 7 \rightarrow u \equiv 6 \pmod 7$
  - $u \equiv 6 \pmod 7 \rightarrow u = 7v + 6$
  - $x = 30u + 26 \rightarrow x = 30(7v + 6) + 26 \rightarrow x = 210u + 206$
  - $x = 210u + 206 \rightarrow x \equiv 206 \pmod{210}$
  - Solución $x = 206$, número entero positivo más pequeño que soluciona el sistema.
  - Las soluciones son todas las x tales que $x = 206 + 210k$, $k \in \mathbb{Z}$

$$x \equiv 1 \pmod 5$$
$$x \equiv 2 \pmod 6$$
$$x \equiv 3 \pmod 7$$

Método Sustitución hacia Atrás

- $5t + 1 \equiv 2(mod6) \rightarrow t \equiv 5(mod6)$
  - $5t + 1 \equiv 2(mod6) \rightarrow 5t + 1 - 1 \equiv 2 - 1(mod6)$
  - $5t \equiv 1(mod6) \rightarrow$ Inverso es 5
  - $5t * 5 \equiv 1 * 5(mod6) \rightarrow t \equiv 5(mod6)$
- $30u + 26 \equiv 3(mod7) \rightarrow u \equiv 6(mod7)$
  - $30u + 26 \equiv 3(mod7) \rightarrow 30u + 26 - 26 \equiv 3 - 26(mod7)$
  - $30u \equiv -23(mod7)$
  - $30u \equiv 1(mod7) \rightarrow$ Inverso es 4
  - $30u * 4 \equiv -23 * 4(mod7) \rightarrow u \equiv -92(mod7)$
  - $u \equiv -92(mod7) \rightarrow u \equiv 6(mod7)$

Teorema de Fermat

- El **teorema de Fermat** se formula de la siguiente manera:
  - Si p es un número primo, entonces, para cada número natural a se tiene $a^p \equiv a \ (mod\ p)$.
- El teorema suele ser presentado de esta otra forma:
  - Si p es un número primo, entonces, para cada número natural a coprimo con p , $a^{p-1} \equiv 1 \ (mod\ p)$.
- Es decir, si se eleva un número a a la p-ésima potencia y al resultado se le resta a, lo que queda es divisible por p.

Teorema de Fermat (cont.)

- Resolver:

$$7^{222} \bmod 11$$

$$7^{10} \equiv 1 (\bmod 11)$$

$$(7^{10})^{k} \equiv 1 (\bmod 11), k \in Z^{+}$$

$$7^{222} = 7^{22*10+2} = (7^{10})^{22} 7^{2} \equiv (1)^{22} * 49 \equiv 5 (\bmod 11)$$

$$7^{222} \bmod 11 = 5$$

Teorema de Fermat (cont.)

- Se aplica al problema de la primalidad y en criptografía.
  - Se comprueba si n (número que se quiere saber si es primo) es divisor del número $2^{n-1} - 1$.

$$2^{n-1} \equiv 1 (\mathrm{mod}\, n)$$

- Sea b un número entero positivo. Si n es un número entero positivo compuesto, y $b^{n-1} \equiv 1 \pmod{n}$, entonces n se llama un **pseudoprimo** a la base b.

$$341 = 11 * 31$$

$$2^{340} \equiv 1 (\mathrm{mod}\, 341)$$

- El número 341 es psedoprimo a la base 2.

Teorema de Fermat (cont.)

- Un número entero compuesto *n* que satisface la congruencia $b^{n-1} \equiv 1 \pmod{n}$ para todos los enteros positivos *b* con $mcd(b, n) = 1$ se llama un número de *Carmichael*.

$$561 = 3 * 11 * 17$$

$$mcd(b,561) = 1$$

$$mcd(b,3) = mcd(b,11) = mcd(b,17) = 1$$

$$b^2 \equiv 1 (\mathrm{mod}\, 3), b^{10} \equiv 1 (\mathrm{mod}\, 11), b^{16} \equiv 1 (\mathrm{mod}\, 17)$$

$$b^{560} = (b^2)^{280} \equiv 1 (\mathrm{mod}\, 3)$$

$$b^{560} = (b^{10})^{56} \equiv 1 (\mathrm{mod}\, 11)$$

$$b^{560} = (b^{16})^{35} \equiv 1 (\mathrm{mod}\, 17)$$

$$b^{560} = 1 (\mathrm{mod}\, 561)$$

Teorema de Euler

- El **teorema de Euler**, también conocido como **teorema de Euler-Fermat**, es una generalización del teorema de Fermat, y como tal afirma una proposición sobre divisibilidad de números.
- El teorema establece que:
  - Si a y n son enteros primos relativos, entonces n divide al entero $a^{\varphi(n)} - 1$.
- Sin embargo, es más común encontrarlo con notación moderna en la siguiente forma:
  - Si a y n son enteros primos relativos, entonces $a^{\varphi(n)} \equiv 1 \pmod{n}$, donde $\varphi(n)$ es la **función φ de Euler**.

Teorema de Euler (cont.)

- La función φ de Euler se describe como:
  - Si *n* es un número entero, la cantidad de enteros entre 1 y *n* que son primos relativos con *n* se denota como φ(*n*):

Valor de *n*	Coprimos con *n* entre 1 y *n*	Función φ(*n*)
1	1	1
2	1	1
3	1,2	2
4	1,3	2
5	1,2,3,4	4
6	1,5	2
7	1,2,3,4,5,6	6
8	1,3,5,7	4
9	1,2,4,5,7,8	6
10	1,3,7,9	4

φ(n)	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0+		1	1	2	2	4	2	6	4	6
10+	4	10	4	12	6	8	8	16	6	18
20+	8	12	10	22	8	20	12	18	12	28
30+	8	30	16	20	16	24	12	36	18	24
40+	16	40	12	42	20	24	22	46	16	42
50+	20	32	24	52	18	40	24	36	28	58
60+	16	60	30	36	32	48	20	66	32	44
70+	24	70	24	72	36	40	36	60	24	78
80+	32	54	40	82	24	64	42	56	40	88
90+	24	72	44	60	46	72	32	96	42	60

Teorema de Euler (cont.)

- Tal función es multiplicativa: si m y n son primos relativos, entonces $\varphi(mn)=\varphi(m)\varphi(n)$.
  - Ejemplo: $\varphi(30) = \varphi(6)\varphi(5) = 2 \cdot 4 = 8$
- Las aplicaciones son numerosas:
  - En criptografía es muy utilizado.
  - En la resolución de ecuaciones de congruencia.
  - En matemáticas puras, sobretodo, relacionadas con el problema de la primalidad.
    - Si n es primo la congruencia se cumplirá siempre, en caso contrario n es compuesto.
  - En el análisis de la descomposición en producto de factores primos de ciertos enteros, en la divisibilidad.

Teorema de Euler (cont.)

- Por ejemplo, se desea encontrar todos los números x que satisfacen $5x \equiv 2 \ (mod\ 12)$, todos los números x tales que 12 divida a $5x - 2$.
- El teorema de Euler dice que $5^{\varphi(12)} = 5^4 \equiv 1 \ (mod\ 12)$ por lo que, multiplicando ambos lados de la ecuación por 5^3:
  - $5^3 \cdot 5x \equiv (5^3 \cdot 2 = 250) \equiv 10 \ (mod\ 12)$
  - $5^4\, x \equiv 10 \ (mod\ 12)$
  - $x \equiv 10 \ (mod\ 12)$

Teorema de Euler (cont.)

- Entonces, la conclusión es que, cualquier número que al dividirse por 12 tenga residuo 10, será una solución de la ecuación.
- Se puede verificar con un ejemplo.
  - Si se divide 34 entre 12, el residuo es 10, por lo que $x = 34$ debe funcionar como solución.
  - Para verificarlo, se divide 170 (34 * 5) entre 12, obtenemos un cociente 14 y un residuo 2, como se esperaba.

Raíces Primitivas y Discretos Logaritmos

- Una **raíz primitiva** módulo p (primo) es un número entero r en Z_p, tal que cada elemento no nulo de Z_p es una potencia de r. Z_p contiene enteros entre 1 y p-1.
- Ejemplo: Determine que 2 y 3 son raíces primitivas módulo 11
  - Potencias de 2 módulo 11 = 2, 4, 8, 5, 10, 9, 7, 3, 6, 1
    - Cada elemento de Z_{11} es una potencia de 2, 2 es una raíz primitiva de 11.
  - Potencias de 3 módulo 11 = 3, 9, 5, 4, 1, 3, 9, 5, 4, 1
    - 3 no es una raíz primitiva de 11.

Raíces Primitivas y Discretos Logaritmos (cont.)

- Un hecho importante en la teoría de números es que hay una raíz primitiva módulo p para cada primo p.
- Se supone que p es primo, r es una raíz primitiva módulo p y a es un entero entre 1 y p-1 (inclusive). Si $r^e \bmod p = a$ y $0 \leq e \leq p$-1, se dice que e es un logaritmo discreto de $a \bmod p$ a la base r y se escribe $log_r a = e$.
- Ejemplo: Encentre el logaritmo discreto de 3 y 5 módulo 11 a la base 2.
  - Potencias de 2 módulo 11 = 2, 4, 8, 5, 10, 9, 7, 3, 6, 1
    - $2^8 = 3$ y $2^4 = 5$ en $Z11$.
    - Los logaritmos discretos son 8 y 4:
    - $log_2 3 = 8$ y $log_2 5 = 4$ módulo 11

Criptografía

- La criptografía se ocupa de las técnicas de cifrado o codificado destinadas a alterar las representaciones lingüísticas de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados.
- Uno de los usos más antiguos conocidos de la criptografía fue el realizado por Julio César.
- Hizo mensajes secretos desplazando cada letra tres letras hacia adelante en el alfabeto. Por ejemplo: letra B se envía E y letra X se envía A.
  - Este es un ejemplo de cifrado, es decir, el proceso de hacer un secreto mensaje.

Criptografía (cont.)

- Para expresar proceso de cifrado de César matemáticamente, se sustituye primero cada letra por un elemento de Z_{26}, es decir, un número entero de 0 a 25 (igual a uno menos su posición en el alfabeto). Por ejemplo: el método de cifrado remplaza A por 0, K por 10, y Z por 25.
- La encriptación de César puede ser representada por la función f que asigna al número entero no negativo p, $p \leq 25$, el número entero $f(p)$ del conjunto $\{0, 1, 2, \ldots, 25\}$ con

$$f(p) = (p + 3) \bmod 26$$

- Así, la letra representada por p se remplaza por la letra representada por $(p + 3)$ *mod* 26.

Criptografía (cont.)

- El proceso de recuperación del texto original del texto cifrado, sin el conocimiento del método de cifrado y la clave, se conoce como **criptoanálisis** o **códigos de rotura (breaking codes)**.
  - En general, es un proceso difícil, especialmente cuando el método de cifrado es desconocido.
- Se explica cómo romper los mensajes que fueron cifrados usando un cifrado de desplazamiento.

Criptografía (cont.)

- Si se sabe el mensaje de texto cifrado y que fue producido por cifrado de desplazamiento, se puede tratar de recuperar el mensaje al cambiar todos los caracteres del texto cifrado por cada uno de los 26 posibles cambios (incluyendo un cambio de cero caracteres)
- La principal herramienta para realizar el procedimiento de código de cambio es el recuento de la frecuencia de las letras en el texto cifrado.
  - Las nueve letras más comunes en texto Inglés y sus frecuencias relativas aproximadas son E 13%, T 9%, A 8%, O 8%, I 7%, N 7%, S 7%, H 6%, y R 6%.

Criptografía (cont.)

- Para criptoanalizar el texto cifrado producido usando un cifrado de desplazamiento:
  - Se encuentra las frecuencias relativas de las letras en el texto cifrado. La hipótesis de que la letra más común en el texto cifrado se produce mediante la encriptación de E. Entonces, se determina el valor del cambio bajo esta hipótesis, digamos *k*.
  - Si el mensaje producido desplazando el texto cifrado por -*k* tiene sentido, se dice que la hipótesis es correcta y que se tiene el valor correcto de *k*.
  - Si no tiene sentido, se considerar la hipótesis de que la letra más común en el texto cifrado que siga. Así se continua con el proceso de la letra más común a menos común.

Criptografía (cont.)

- ZNK KGXRE HOXJ MKZY ZNK CUXS:
  - La letra con más frecuencia es K.
  - Se sustituye por E: $10 = 4 + k \ mod \ 26$
  - $k = 6$, todo se desplaza -6
  - THE EARLY BIRD GETS THE WORM
  - El mensaje tiene sentido, $k = 6$ es correcto

Criptografía (cont.)

- Se puede generalizar el método criptográfico de desplazamiento usando la siguiente función de cifrado:
  - $f(p) = (ap + b)\ mod\ 26;\ a, b \in \mathbb{Z}$
- La función de descifrado será la función biyectiva de la función anterior, lo cual ocurre si y solo si el mcd(a,26) = 1.
  - $f(c) = a'(c - b)\ mod\ 26;\ a, b \in \mathbb{Z}$
- Tal asignación se llama **transformación afín** y el resultando de cifrado se llama **cifrado afín**.
  - $c \equiv ap + b(mod\ 26) \rightarrow mcd(a, 26) = 1 \Longrightarrow$ encontrar el inverso modular de a' = a mod 26
  - $a'(c - b) \equiv a'ap(mod\ 26) \rightarrow \text{aa}' \equiv 1(mod\ 26)$
  - $p \equiv a'(c - b)\ mod\ 26$

Criptografía (cont.)

- ¿Cuál es la función de descifrado de acuerdo a la función de cifrado $f(p) = (7p + 3) \bmod 26$?
  - $a = 7$, $b = 3$, $m = 26$
  - mcd(7,26) = 1
  - a' = Inverso modular de 7 mod 26 = 15
  - $f(c) = a'(c - b)\ mod\ 26 = 15(c - 3)\ mod\ 26$

Criptografía (cont.)

- Los sistemas que proceden sustituyendo cada letra del alfabeto por otra letra del alfabeto. Debido a esto, estos cifrados se llaman carácter o sistemas de cifrado mono-alfabéticos.
  - Los métodos de cifrado de este tipo son vulnerables a ataques basados en el análisis de la frecuencia de letra en el texto cifrado.
- Se puede hacer más difícil de atacar con texto cifrado mediante la sustitución de bloques de letras con otros bloques de letras, en vez de sustituir caracteres de manera individual. Esto se llama cifrado de bloque.

Criptografía (cont.)

- Un tipo simple de cifrado de bloque se llama cifrado por transposición.
- Como clave se utiliza una permutación σ del conjunto {1, 2,…, *m*} para algún entero positivo *m*, es decir, una función de uno-a-uno de {1, 2,…, *m*} para sí mismo.
- Para cifrar un mensaje se divide las letras en bloques de tamaño *m*. Si el número de letras en el mensaje no es divisible por *m* añadimos algunas letras al azar al final para completar el bloque final. Luego, se encripta el bloque $p_1 p_2 \ldots p_m$ como $c_1 c_2 \ldots c_m = p_{\sigma(1)} p_{\sigma(2)} \cdots p_{\sigma(m)}$.

Criptografía (cont.)

- $p_{\sigma(m)}$ = Posición que ocupará en el bloque cifrado la letra de la posición m del bloque original de acuerdo a la permutación σ.
- Para descifrar un bloque de texto cifrado $c_1c_2\ldots c_m$, se transpone las letras usando la permutación σ^{-1}, la inversa de la permutación σ.
- Ejemplo: Use el método de cifrado basado en la permutación σ del conjunto {1,2,3,4} con σ(1) = 3, σ(2) = 1, σ(3) = 4 y σ(4) = 2.
  - a) Cifre el mensaje PIRATE ATTACK
  - b) Descifre el mensaje cifrado SWUE TRAE OEHS

Criptografía (cont.)

- Use el método de cifrado basado en la permutación σ del conjunto {1,2,3,4} con σ(1) = 3, σ(2) = 1, σ(3) = 4 y σ(4) = 2.
  - a) Cifre el mensaje PIRATE ATTACK
    - $m = 4$
    - PIRA TEAT TACK
    - Primer bloque:
      - $p_{\sigma(1)} = 3 \rightarrow$ Posición que ocupará en el bloque cifrado la letra de la posición 1 del bloque original: P, de acuerdo a la permutación σ(1)
      - $p_{\sigma(2)} = 1 \rightarrow$ Posición que ocupará en el bloque cifrado la letra de la posición 2 del bloque original: I, de acuerdo a la permutación σ(2)
      - $p_{\sigma(3)} = 4 \rightarrow$ Posición que ocupará en el bloque cifrado la letra de la posición 2 del bloque original: R, de acuerdo a la permutación σ(3)
      - $p_{\sigma(4)} = 2 \rightarrow$ Posición que ocupará en el bloque cifrado la letra de la posición 2 del bloque original: A, de acuerdo a la permutación σ(4)
    - Lo anterior con cada bloque
    - IAPR ETTA AKTC

Criptografía (cont.)

- Use el método de cifrado basado en la permutación σ del conjunto {1,2,3,4} con σ(1) = 3, σ(2) = 1, σ(3) = 4 y σ(4) = 2.
  - b) Descifre el mensaje cifrado SWUE TRAE OEHS
    - $m = 4$
    - SWUE TRAE OEHS
    - Primer bloque:
      - c_1 y $\sigma^{-1}(1) = 2$ → Posición que ocupará en el bloque original la letra de la posición 1 del bloque cifrado: S, de acuerdo a la permutación $\sigma^{-1}(1)$
      - c_2 y $\sigma^{-1}(2) = 4$ → Posición que ocupará en el bloque cifrado la letra de la posición 2 del bloque original: W, de acuerdo a la permutación $\sigma^{-1}(2)$
      - c_3 y $\sigma^{-1}(3) = 1$ → Posición que ocupará en el bloque cifrado la letra de la posición 2 del bloque original: U, de acuerdo a la permutación $\sigma^{-1}(3)$
      - c_4 y $\sigma^{-1}(4) = 3$ → Posición que ocupará en el bloque cifrado la letra de la posición 2 del bloque original: E, de acuerdo a la permutación $\sigma^{-1}(4)$
    - Lo anterior con cada bloque
    - USEW ATER HOSE
    - USE WATER HOSE

Criptografía (cont.)

- Un **sistema de criptografía** es una tupla de cinco elementos (*P,C,K,E,D*), donde *P* es el conjunto de caracteres del texto original, *C* es el conjunto de caracteres del texto cifrado, *K* es el *keyspace* (conjunto de posibles llaves o claves), *E* es el conjunto de funciones de encriptación, y *D* es el conjunto de funciones de descifrado.
- Se denota por E_k la función de cifrado en *E* correspondiente a la clave *k* y D_k la función de descifrado en *D* que descifra el texto cifrado que se cifró usando E_k, es decir, *Dk(Ek(p)) = p*, para todos los caracteres del texto original *p*.

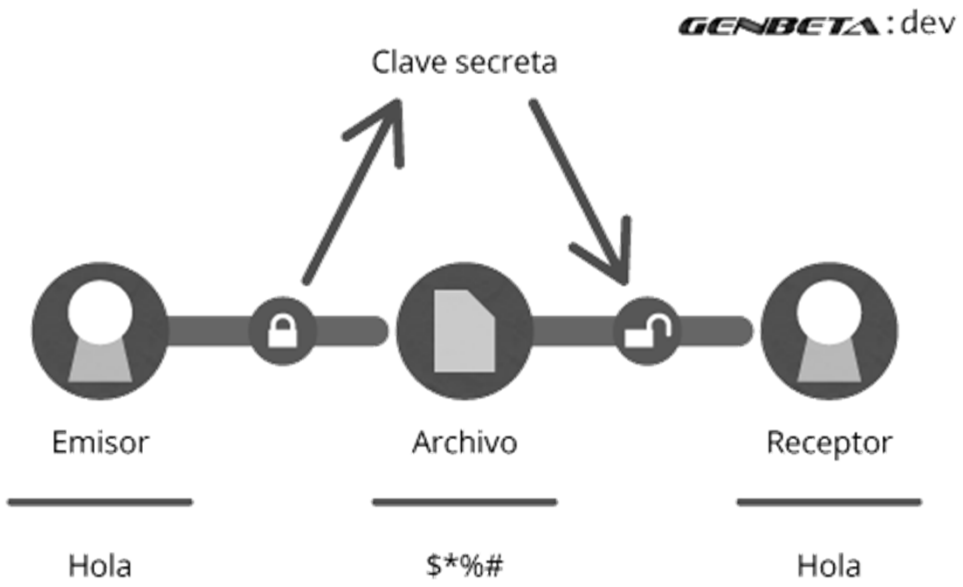
Criptografía (cont.)

- Ejemplo: Describa el sistema del cifrado de desplazamiento
  - P y C Enteros entre 0 y 25, es decir, elementos de Z_{26}
  - K conjunto de posibles desplazamientos
  - E es el conjunto de funciones $E_k(p) = (p + k) \; mod \; 26$
  - D es el conjunto de funciones $D_k(p) = (p - k) \; mod \; 26$

Criptografía (cont.)

- El **cifrado simétrico** o **cifrado de clave privada** (**cifrado de clave secreta**) consiste en utilizar la misma clave para el cifrado y el descifrado.
- En un sistema de cifrado de clave privada, una vez que sabes una clave de cifrado, se puede encontrar rápidamente la clave de descifrado.
  - Los cifrados clásicos, incluyendo los sistemas de cifrado de desplazamiento y cifrado afín, son ejemplos de sistemas criptográficos de clave privada.

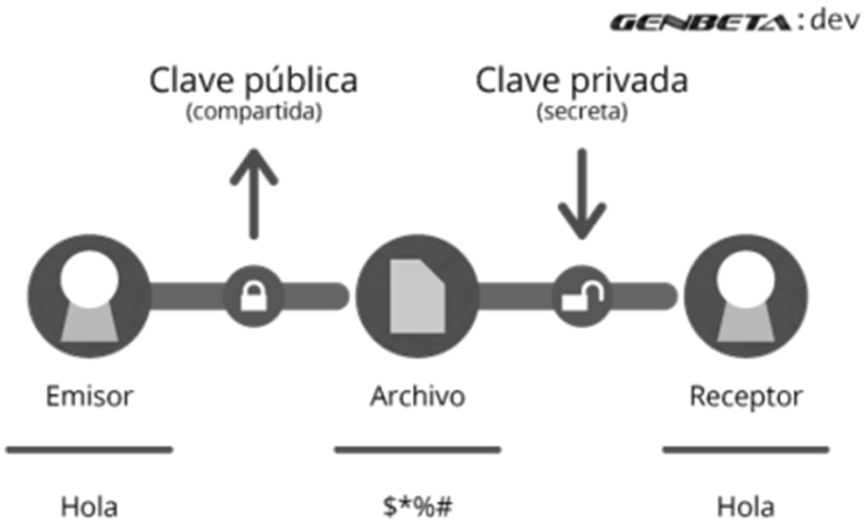
Criptografía (cont.)



Criptografía (cont.)

- El **cifrado asimétrico** o **cifrado de clave pública** apareció en 1976, con la publicación de un trabajo sobre criptografía por Whitfield Diffie y Martin Hellman.
- En un criptosistema asimétrico (o criptosistema de clave pública), las claves se dan en pares:
  - Una clave pública para el cifrado
  - Una clave secreta para el descifrado

Criptografía (cont.)



Criptografía (cont.)

- En un sistema de cifrado con clave pública, los usuarios eligen una clave aleatoria que sólo ellos conocen (ésta es la clave privada).
- A partir de esta clave, automáticamente se deduce un algoritmo (la clave pública). Los usuarios intercambian esta clave pública mediante un canal no seguro.
- Cuando un usuario desea enviar un mensaje a otro usuario, sólo debe cifrar el mensaje que desea enviar utilizando la clave pública del receptor (que puede encontrar, por ejemplo, en un servidor de claves como un directorio LDAP).
- El receptor podrá descifrar el mensaje con su clave privada (que sólo él conoce).

Criptografía (cont.)

- Este sistema se basa en una función que es fácil de calcular en una dirección (llamada *función trapdoor de único sentido*) y que, matemáticamente, resulta muy difícil de invertir sin la clave privada (llamada *trapdoor*).
  - Por ejemplo: si un usuario creara de forma aleatoria una pequeña llave metálica (clave privada) y luego produjera una gran cantidad de candados (claves públicas) que guarda en un casillero al que puede acceder cualquiera (casillero sería el canal no seguro). Para enviarle un documento, cada usuario puede usar un candado (abierto), cerrar con este candado una carpeta que contiene el documento y enviar la carpeta al dueño de la clave pública (dueño del candado). Sólo el dueño podrá abrir la carpeta con su clave privada.

Sistema de Cifrado RSA

- El sistema de cifrado RSA es el sistema de cifrado asimétrico más usado y más sencillo de entender e implementar.
- Una peculiaridad de este algoritmo es que sus dos claves sirven indistintamente tanto para cifrar como para autenticar.
- Debe su nombre a sus tres inventores: Ronald **Rivest**, Adi **Shamir** y Leonard **Adleman**, que publicaron por primera vez el método RSA en 1977.
- Se basa en la dificultad que presenta la factorización de números grandes.

Sistema de Cifrado RSA (cont.)

- Las claves **pública** y **privada** se calculan a partir de un número que se obtiene como producto de dos primos grandes.
- Un atacante que quiera recuperar un texto claro a partir del criptograma y de la clave pública, tiene que enfrentarse a dicho problema de factorización.
- El algoritmo consta de tres paso:
  - Generación de claves
  - Cifrado del mensaje
  - Descifrado del mensaje

Sistema de Cifrado RSA
Generación de Claves

- Cada usuario elige dos números primos distintos y grandes p y q (unas 200 cifras cada uno).
  - Por seguridad deben ser elegidos de forma aleatoria y tener una longitud en bits parecida. Se pueden hallar primos fácilmente mediante test de primalidad.
- Se calcula el producto $n = pq$
  - n se usa como el módulo para ambas claves: pública y privada
- Se calcula $\varphi(n) = (p-1)(q-1)$
  - φ es la función de Euler
- Se escoge un número entero positivo e menor que $\varphi(n)$, que sea primo relativo con $\varphi(n)$, e se usa como el exponente de la clave pública.

Sistema de Cifrado RSA
Generación de Claves

- Se determina un d (mediante aritmética modular) que satisfaga la congruencia $ed \equiv 1 \pmod{\varphi(n)} \rightarrow ed \equiv 1 \pmod{(p-1)(q-1)}$, d se usa como el exponente de la clave privada
  - d es el inverso modular de e mod $\varphi(n)$
  - Se calcula mediante el algoritmo de Euclides
  - Se cumple que $ed = 1 + k(p-1)(q-1)$ para cualquier entero k.
- La **clave pública** será el par de números (e,n), que pueden ser conocidos por cualquiera.
- La **clave privada** será el par de números (d,n), este número d debe mantenerse secreto y sólo será conocido por el propietario del par de claves.

Sistema de Cifrado RSA
Generación de Claves

- Para una mayor eficiencia los siguientes valores se calculan de antemano y se almacenan como parte de la clave privada:
  - Los primos para la generación de las claves: p y q.
  - $d \bmod (p-1)$ y $d \bmod (q-1)$
  - $q^{-1} \bmod p$
- URL: https://es.wikipedia.org/wiki/PKCS.

Sistema de Cifrado RSA
Cifrado y Descifrado del Mensaje

- Los mensajes que se cifran y descifran con este algoritmo son números enteros de tamaño menor que *n*, no letras sueltas como en el caso de los cifrados vistos antes.
- Para obtener el mensaje cifrado *C* a partir del mensaje original *M* se realiza la siguiente operación:
  $C = M^e \pmod n$
- Para recuperar el mensaje original *M* a partir del cifrado *C* se realiza la siguiente operación:
  $M = C^d \pmod n$

Sistema de Cifrado RSA
Ejemplo

- Cifrar STOP con RSA (use números primos pequeños)
  - $p = 43$ y $q = 59$, $n = 43*59 = 2537$
  - $mcd(e,42*58) = mcd(13,42*58) = 1$
  - Clave pública $Kp = (13,2537)$
  - STOP se pasa a números según la posición y se agrupan en bloques de cuatro dígitos: 1819 1415
  - Se usa la operación para cifrar:

$$C_1 = 1819^{13} \bmod 2537 = 2081$$

$$C_2 = 1415^{13} \bmod 2537 = 2182$$

  - El mensaje cifrado es: 2081 2182

Sistema de Cifrado RSA
Ejemplo

- Descifrar 0981 0461 con RSA (use números primos pequeños)
  - $d = 937$ (es el inverso de 13 módulo 42*58=2436)
  - Clave privada $KP = (937,2537)$
  - Se usa la operación para descifrar:

$$M_1 = 0981^{937} \bmod 2537 = 0704$$

$$M_2 = 0461^{937} \bmod 2537 = 1115$$

  - El mensaje descifrado es: 0704 1115
  - Se pasan las posiciones a letras: HELP

Protocolos de Criptografía
Intercambio de Claves

- Protocolo que permite intercambiar una clave secreta a través de un canal de comunicaciones inseguro sin haber compartido ninguna información en el pasado.
  - Generar una clave que se puede compartir es importante para muchas aplicaciones de la criptografía.
- Este protocolo se conoce como protocolo de claves Diffie-Hellman, Whitfield Diffie y Martin Hellman, descrito en 1976.

Protocolos de Criptografía
Intercambio de Claves (cont.)

- Alice y Bob desean compartir una clave común por un canal no seguro.
- El protocolo tiene los siguientes pasos:
  - (1) Alice y Bob deciden usar un primo p y una raíz primitiva a de p.
  - (2) Alice elige un número entero secreto k_1 y envía a^{k1} **mod** p a Bob.
  - (3) Bob elige un número entero secreto k_2 y envía a^{k2} **mod** p a Alice.
  - (4) Alice calcula $(a^{k2})^{k1}$ **mod** p.
  - (5) Bob calcula $(a^{k1})^{k2}$ **mod** p.

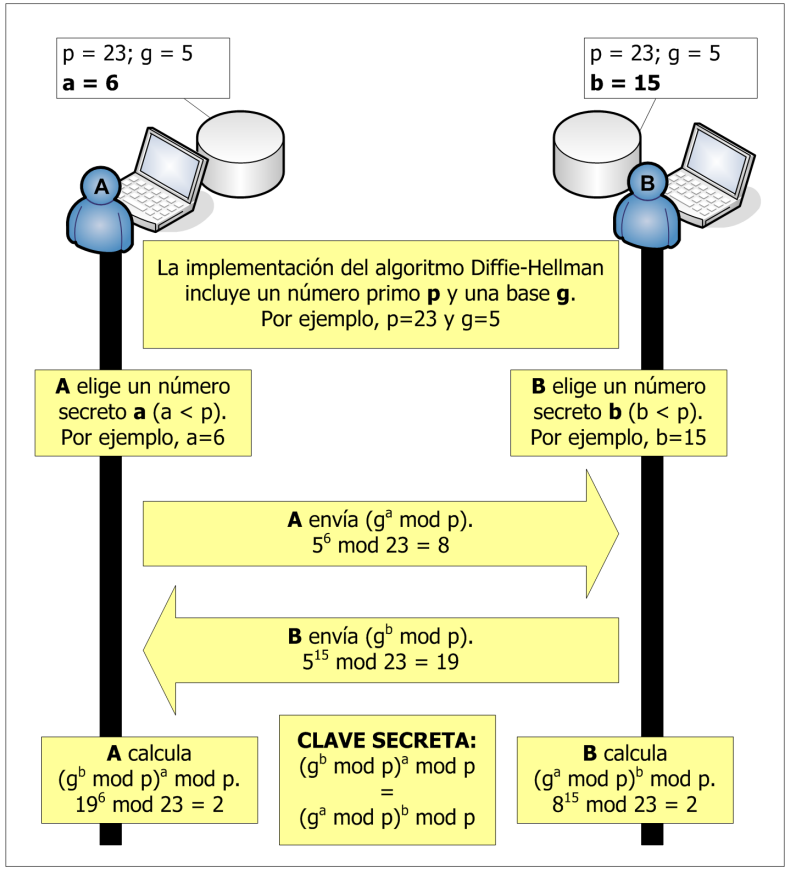
Protocolos de Criptografía
Intercambio de Claves (cont.)

- Al final del protocolo, Alice y Bob han calculado su clave compartida, es decir, $(a^{k2})^{k1}$ **mod** $p = (a^{k1})^{k2}$ **mod** p.
- El protocolo garantiza que los mensajes enviados en las etapas (1), (2) y (3) no se supone que se envían de forma segura, son información pública.
  - p, a, a^{k1} **mod** p, y a^{k2} **mod** p son información pública.
- El protocolo garantiza que k_1, k_2, y la clave común $(a^{k2})^{k1}$ **mod** $p = (a^{k1})^{k2}$ **mod** p se mantienen en secreto.
  - Para encontrar la información secreta de esta información pública requiere que se resuelve instancias del problema del logaritmo discreto, se tendría que encontrar k_1 y k_2 de a^{k1} **mod** p y a^{k2} **mod** p, respectivamente.

Protocolos de Criptografía
Intercambio de Claves (cont.)

- Por otra parte, ningún otro método es conocido para encontrar la clave compartida utilizando sólo la información pública.
- Este sistema es computacionalmente imposible cuando p y a son lo suficientemente grandes.
  - Con la potencia de cálculo disponible ahora, este sistema se considera irrompible cuando p tiene más de 300 dígitos decimales y, k_1 y k_2 deben tener más de 100 dígitos decimales cada uno.

Protocolos de Criptografía
Intercambio de Claves (cont.)



Protocolos de Criptografía
Firma Digital

- La criptografía se utiliza también para que el destinatario del mensaje esté seguro que la persona que envía el mensaje sea la cree que procede.
- En particular, se puede mostrar cómo esto puede llevarse a cabo utilizando el sistema de cifrado RSA para aplicar una firma digital a un mensaje.
- La firma digital puede ayudar en la autenticación e integridad de los datos enviados entre dos personas que tengan desconfianza mutua. Las firmas digitales deben tener las mismas características que las firmas manuales.

Protocolos de Criptografía
Firma Digital (cont.)

- Suponer que la clave pública RSA de Alice es (n,e) y su clave privada es d.
- Alice encripta un mensaje de texto original x utilizando la función de cifrado $E_{(n,e)}(x) = x^e$ **mod** n.
- Ella descifra un mensaje de texto cifrado usando la función de descifrado $D_{(n,e)} = x^d$ **mod** n.
- Alice quiere enviar el mensaje M de modo que todo el que recibe el mensaje sabe que ella lo envía.

Protocolos de Criptografía
Firma Digital (cont.)

- Al igual que en el cifrado RSA, que traduce las letras en sus equivalentes numéricos y se divide la cadena resultante en bloques $m_1, m_2, ..., m_k$, tal que cada bloque es el mismo tamaño lo más grande posible, de modo que $0 \leq m_i \leq n$ para $i = 1, 2, ..., k$.
- A continuación, se aplica la función de descifrado $D_{(n,e)}$ para cada bloque, obteniendo $D_{(n,e)}(m_i)$, $i = 1, 2, ..., k$.
- Ella envía el resultado a todos los destinatarios del mensaje.

Protocolos de Criptografía
Firma Digital (cont.)

- Cuando un destinatario recibe el mensaje de Alice, donde se aplicó la función de cifrado $E_{(n,e)}$ para cada bloque, todo el mundo tiene disponible la clave de Alice (n,e) porque es información pública.
- El resultado es el bloque de texto original porque $E_{(n,e)}(D_{(n,e)})(x)) = x$.
- Por lo tanto, Alice puede enviar su mensaje a tantas personas como ella quiera y mediante su firma, de esta manera, cada destinatario puede estar seguro de que venía de Alice.

Protocolos de Criptografía
Firma Digital (cont.)

- Ejemplo: Se tiene como clave pública RSA de Alice n = 43*59 = 2537 y e = 13. Su clave de descifrado es d = 937. Se quiere enviar el mensaje: "MEETAT NOON" a sus amigos para que estén seguros de que provenía de ella. ¿Qué se va a enviar?
  - Alice traslada el mensaje a bloques de dígitos: 1204 0419 0019 1314 1413
  - Ella aplica la transformación de descifrado $D_{(2537,13)}(x) = x^{937}$ **mod** 2537 a cada bloque. Obtuvo lo siguiente: 1204^{937} **mod** 2537 = 817, 419^{937} **mod** 2537 = 555, 19^{937} **mod** 2537 = 1310, 1314^{937} **mod** 2537 = 2173, and 1413^{937} **mod** 2537 = 1026.

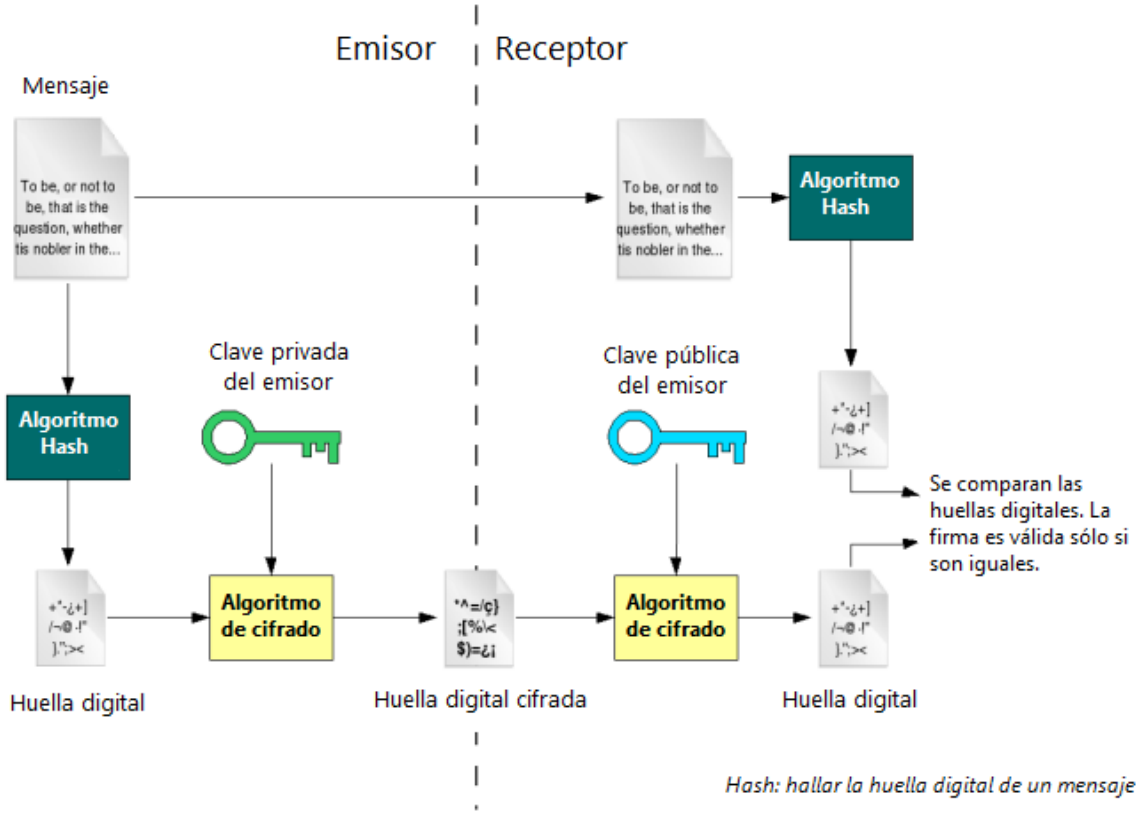
Protocolos de Criptografía
Firma Digital (cont.)

- Ejemplo: Se tiene como clave pública RSA de Alice n = 43*59 = 2537 y e = 13. Su clave de descifrado es d = 937. Se quiere enviar el mensaje: "MEETAT NOON" a sus amigos para que estén seguros de que provenía de ella. ¿Qué se va a enviar?
  - El mensaje que se envía es: 0817 0555 1310 2173 1026.
  - Cuando uno de sus amigos consigue este mensaje, aplican su transformación cifrado $E_{(2537,13)}$ para cada bloque. Obteniendo los bloques de dígitos del mensaje original que traducen de nuevo a las letras correspondientes.

Protocolos de Criptografía
Firma Digital (cont.)

- Los mensajes firmados pueden ser enviados mediante el sistema de cifrado RSA. Se puede extender el sistema de cifrado RSA para enviar mensajes secretos firmados.
- Para ello, el emisor aplica el cifrado RSA utilizando la clave pública de cifrado de un destinatario a cada bloque que se cifra mediante la transformación de descifrado del remitente.
- El receptor aplica la transformación de descifrado privada y luego la transformación de cifrado pública del remitente.

Protocolos de Criptografía
Firma Digital (cont.)



Hash: hallar la huella digital de un mensaje

Criptografía (cont.)

- Criptografía
  http://jwilson.coe.uga.edu/EMAT6680Fa2012/Warrayat/EMAT%206690/Essay2/Essay2.html
- Sistemas de cifrado (clave privada y clave pública)
  http://www.genbetadev.com/seguridad-informatica/tipos-de-criptografia-simetrica-asimetrica-e-hibrida
- RSA
  https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html

Códigos Binarios de Números Decimales

- Los números binarios son los más apropiados para los cálculos internos en un sistema digital, pero la mayoría de la gente todavía prefiere trabajar con los números decimales.
- Como resultado, las interfaces externas de un sistema digital pueden leer o exhibir números decimales.
- Un conjunto de cadenas de *n* bits en que las diferentes cadenas de bits representan diferentes números u otras cosas se llama **código**.
- Una combinación particular de valores de *n* bits se llama **palabra del código**.

Códigos Binarios de Números Decimales (cont.)

- Puede que en un código haya o no una relación aritmética entre los valores de los bits en una palabra de código y lo que representan.
- Además, un código que usa cadenas de n bits no necesita contener 2^n palabras de código.
- Al menos se necesitan 4 bits para representar los diez dígitos decimales.
  - Hay muchas maneras diferentes para elegir las 10 palabras código de 4 bits, los más comunes se muestran en la siguiente tabla.

Códigos Binarios de Números Decimales (cont.)

Dígito Decimal	BCD (8421)	Aiken (2421)	Exceso 3	Biquinario	1 de 10
0	0000	0000	0011	0100001	1000000000
1	0001	0001	0100	0100010	0100000000
2	0010	0010	0101	0100100	0010000000
3	0011	0011	0110	0101000	0001000000
4	0100	0100	0111	0110000	0000100000
5	0101	1011	1000	1000001	0000010000
6	0110	1100	1001	1000010	0000001000
7	0111	1101	1010	1000100	0000000100
8	1000	1110	1011	1001000	0000000010
9	1001	1111	1100	1010000	0000000001

Códigos Binarios de Números Decimales (cont.)

Palabras de código no usadas				
BCD (8421)	**Aiken (2421)**	**Exceso 3**	**Biquinario**	**1 de 10**
1010	0101	0000	0000000	0000000000
1011	0110	0001	0000001	0000000011
1100	0111	0010	0000010	0000000101
1101	1000	1101	0000011	0000000110
1110	1001	1110	0000101	0000000111
1111	1010	1111	…	…

Códigos Binarios de Números Decimales – Código BCD

- El código decimal más natural es el BCD (*binary-coded decimal*), que codifica los dígitos 0 a 9 por sus representaciones binarias sin signo en 4 bits, del 0000 al 1001.
- En un byte caben dos dígitos en BDC.
- Los códigos BCD más usados son:
  - Natural (8421).
  - Aiken (2421).
  - 5421.
  - Exceso 3.

Códigos Binarios de Números Decimales – Código BCD (cont.)

- En el BCD sólo se utilizan 10 de las 16 posibles combinaciones que se pueden formar con números de 4 bits, por lo que el sistema pierde capacidad de representación, aunque se facilita la compresión de los números.
  - El BCD solo se usa para representar **cifras** no números en su totalidad.
  - Esto quiere decir que para números de más de una cifra hacen falta dos números BCD para componerlo.
- Los pesos para los bits BDC son 8, 4, 2 y 1, por esta razón se le llama código 8421.

Códigos Binarios de Números Decimales – Código BCD (cont.)

- Desde que los sistemas informáticos empezaron a almacenar los datos en conjuntos de ocho bits, hay dos maneras comunes de almacenar los datos BCD:
  - Omisión de los cuatro bits más significativos(como sucede en el EBCDIC).
  - Almacenamiento de dos datos BCD, es el denominado BCD "empaquetado", en el que también se incluye en primer lugar el signo, por lo general con 1100 para el + y 1101 para el -.
- De este modo, el número 127 sería representado como (11110001, 11110010, 11110111) en el EBCDIC o (00010010, 01111100) en el BCD empaquetado

Códigos Binarios de Números Decimales – Código BCD (cont.)

- El BCD sigue siendo ampliamente utilizado para almacenar datos, en aritmética binaria o en electrónica. Los números se pueden mostrar fácilmente en visualizadores de siete segmentos enviando cada cuarteto BCD a un visualizador.
- La BIOS de un ordenador personal almacena generalmente la fecha y la hora en formato del BCD, probablemente por razones históricas se evitó la necesidad de su conversión en ASCII.

Códigos Binarios de Números Decimales – Código BCD (cont.)

- La ventaja del código BCD frente a la representación binaria clásica es que no hay límite para el tamaño de un número.
- Los números que se representan en formato binario están generalmente limitados por el número mayor que se pueda representar con 8, 16, 32 o 64 bits.
- Por el contrario utilizando BCD añadir un nuevo dígito sólo implica añadir una nueva secuencia de 4 bits.

Códigos Binarios de Números Decimales – Código BCD (cont.)

- La suma de dígitos BCD es similar a la suma de números binarios sin signo, excepto que se debe hacerse una corrección si el resultado excede 1001.
  - El resultado se corrige sumándole 6.

5	0101	5	0101	
+ 4	+ 0100	+ 9	+ 1001	
9	1001	14	1110	
			+ 0110	– Corrección
5	0000 0101	10+4	1 0100	
+ 11	+ 0001 0001			
16	0001 0110			

Códigos Binarios de Números Decimales – Código BCD (cont.)

- Otro conjunto de pesos resulta el **código 2421**, que es un *código autocomplementario*; o sea, la palabra código para el complemento a 9 de cualquier dígito puede obtenerse al complementar los bits individuales de la palabra código del dígito.
- El **código de exceso 3** es otro código autocomplementario, tiene una relación aritmética con el BDC; ya que la palabra código para cada dígito decimal es la correspondiente a la de BCD más 0011 (+3).

Códigos Binarios de Números Decimales – Código BCD (cont.)

- Los códigos decimales pueden tener más de 4 bits, como el **código biquinario** que usa 7 bits.
  - Los primeros dos bits en una palabra código indican si el número está en el rango 0-4 o 5-9 y los últimos 5 bits indican cuál de los cinco números del rango seleccionado está representado.
- El **código 1 de 10** es la codificación menos densa para los dígitos decimales, usando sólo 10 palabras de código de las 1024 posibles.

Códigos Binarios de Números Decimales – Código BCD (cont.)

- El término ***biquinario*** se refiere a que el código tiene una parte de dos estados (*bi*) y otra de cinco estados (*quin*).
- Existen varias representaciones de un decimal codificado en biquinario, ya que:
  - El componente de dos estados se puede representar tanto con uno como con dos bits.
  - El componente de cinco estados, tanto con tres como con cinco bits.

Código Gray

- Este es un código binario no ponderado y tiene la propiedad de que los códigos para dígitos decimales sucesivos difiere en un sólo bit, al código Gray también se le llama autorreflejado o cíclico.
  - Es un caso particular de sistema binario.
- Consiste en una ordenación de 2^n números binarios de tal forma que cada número sólo tenga un dígito binario distinto a su predecesor.
- Este código puede representar números o cosas.

Código Gray (cont.)

- Historia:
  - Esta técnica de codificación se originó cuando los circuitos lógicos digitales se realizaban con válvulas de vacío y dispositivos electromecánicos.
  - Los contadores necesitaban potencias muy elevadas a la entrada y generaban picos de ruido cuando varios bits cambiaban simultáneamente.
  - El uso de código Gray garantizó que en cualquier transición variaría tan sólo un bit.

Código Gray (cont.)

- El primer uso documentado de un código de estas características fue en una demostración del telégrafo del ingeniero francés Émile Baudot, en 1878.
- Pero no fueron patentados hasta 1953 por Frank Gray (que dio nombre al sistema de codificación), un investigador de los laboratorios Bell.
- Hay varios algoritmos para generar una secuencia de código Gray (y varios códigos posibles resultantes, en función del orden que se desee seguir), pero el más usado consiste en cambiar el bit menos significativo que genera un nuevo código.

Código Gray (cont.)

Número Decimal	Código Binario	Código Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

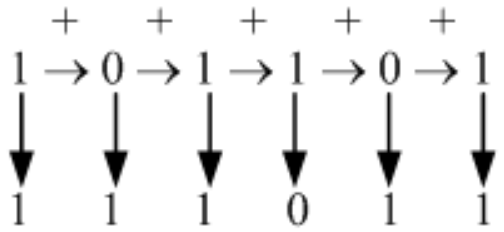
Código Gray (cont.)

Número Decimal	Código Binario	Código Gray
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Código Gray (cont.)

- Para convertir de Binario a Gray puede seguirse el siguiente procedimiento:
  - El MSB se deja igual.
  - Avanzando de MSB a LSB se suma cada bit con el siguiente despreciando el acarreo para obtener el siguiente bit del código Gray.
- Ejemplo: Pasar el número decimal 45 a código Gray.

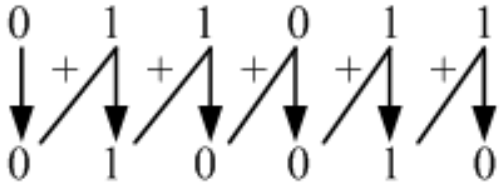
$$45_{10} = 101101_2$$



$$45_{10} = 111011_{gray}$$

Código Gray (cont.)

- Para convertir de Gray a Binario puede seguirse el siguiente procedimiento:
  - El MSB se deja igual.
  - Avanzando de MSB a LSB a cada bit obtenido en binario se le suma sin acarreo el siguiente bit de código Gray.
- Ejemplo: Obtener el equivalente decimal del siguiente código gray 011011_{gray}.



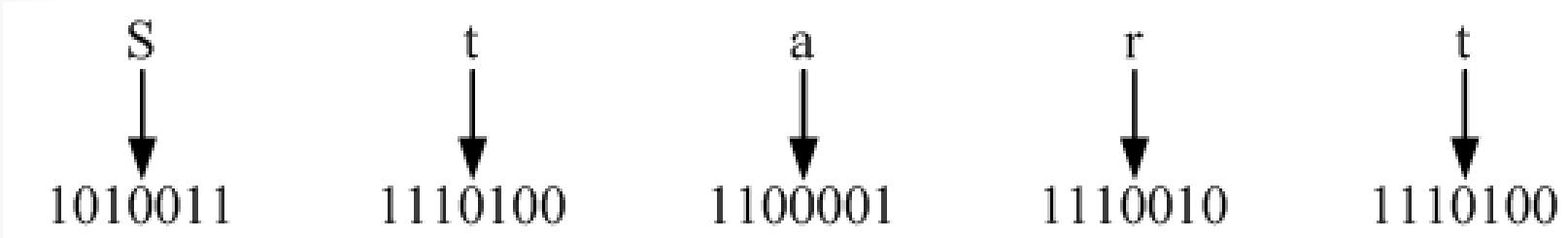
$011011_{gray} = 010010_2$
$010010_2 = 18_{10}$

Códigos de Caracteres o Alfanuméricos

- Muchas aplicaciones de sistemas digitales (especialmente las computadoras o la transmisión de textos) requieren del procesamiento de datos los como números, letras y símbolos especiales.
- Para manejar estos datos usando dispositivos digitales, cada símbolo debe estar representado por un código binario.
- El código alfanumérico más generalizado en la actualidad es el denominado ASCII (*American Standard Code for Information Interchange*). Este es un código de 7 bits.

Códigos de Caracteres o Alfanuméricos (cont.)

- Ver http://www.isa.cie.uva.es/proyectos/codec/teoria2.html.
- Ejemplo: La palabra "Start" se representa en código ASCII como sigue:



Códigos de Detección y Corrección de Errores

- Los sistemas digitales pueden cometer errores de vez en cuando.
- Aunque los dispositivos en circuito integrado que carecen de partes móviles tienen alta confiabilidad; pero los dispositivos que tienen interacción con partes móviles son menos confiables.
- Cuando se leen, escriben o transmiten caracteres de un sitio a otro, pueden producirse errores

Códigos de Detección y Corrección de Errores (cont.)

- Ejemplos:
  - Se pueden producir errores por polvo en las cabezas lectoras de una unidad de disco.
  - La ocurrencia de errores en la *transmisión de datos a distancia.*
  - Los datos que se transmiten por modem pueden recibirse incorrectamente si la línea tiene ruidos.
  - Las perturbaciones en el suministro de energía eléctrica pueden producir errores.
- En esta sección se ilustran dos códigos que permiten detectar errores y, en algunos casos, incluso corregirlos.

Códigos de Detección y Corrección de Errores – Código de Paridad

- Un método muy simple, pero ampliamente utilizado por su sencillez para detectar errores en transmisión de datos consiste en añadir un **bit de paridad a cada carácter**, normalmente en la posición más significativa.
  - En el código de **paridad par**, el bit de paridad se elige de manera que el número de bits 1 del dato sea un número par incluyendo el bit de paridad.
  - En el código de **paridad impar**, el bit de paridad se elige de modo que el número de bits 1 (incluyendo el de paridad) del dato sea impar.

Códigos de Detección y Corrección de Errores – Código de Paridad (cont.)

Bit de Paridad	Código ASCII							Carácter
0	1	0	1	0	0	1	1	S
0	1	1	1	0	1	0	0	t
1	1	1	0	0	0	0	1	a
0	1	1	1	0	0	1	0	r
0	1	1	1	0	1	0	0	t

Códigos de Detección y Corrección de Errores – Código de Paridad (cont.)

Bit de Paridad	Código ASCII							Carácter
1	1	0	1	0	0	1	1	S
1	1	1	1	0	1	0	0	t
0	1	1	0	0	0	0	1	a
1	1	1	1	0	0	1	0	r
1	1	1	1	0	1	0	0	t

Códigos de Detección y Corrección de Errores – Código de Paridad (cont.)

- De esta manera, cuando cambia un bit durante la transmisión, el número de unos en el carácter recibido tendrá la paridad equivocada y el receptor sabrá que se ha producido un error.
- Ejemplo:
  - Si un transmisor envía “Start” y hay errores en la transmisión, suponiendo que el receptor recibe la siguiente información, en la siguiente tabla se anota los datos que llegaron erróneos y si se detectó o no el error, agrega en la columna vacía cuantos bits cambiaron en la transmisión.

Códigos de Detección y Corrección de Errores – Código de Paridad (cont.)

Dato Enviado	Dato Recibido (supuesto)	Error	Paridad	Bits Erróneos
01010011 (S)	01010010 (R)	Sí	Mal	1
01110100 (t)	01100010 (>)	Sí	Mal	3
11100001 (a)	11100001 (a)	No	Bien	0
01110010 (r)	01110001 (G)	Sí	Bien	2
01110100 (t)	01110100 (t)	No	Bien	0

Códigos de Detección y Corrección de Errores – Código de Paridad (cont.)

Dato Enviado	Dato Recibido (supuesto)	Error	Paridad	Bits Erróneos
11010011 (S)	11010010 (R)	Sí	Mal	1
11110100 (t)	11100010 (>)	Sí	Mal	3
01100001 (a)	01100001 (a)	No	Bien	0
11110010 (r)	11110001 (G)	Sí	Bien	2
11110100 (t)	11110100 (t)	No	Bien	0

Códigos de Detección y Corrección de Errores – Código de Paridad (cont.)

- Como puede verse, el código de paridad No siempre resulta efectivo para detectar errores.
- ¿Qué tipo de errores si detecta y cuales no?
  - Detecta cuando hay una cantidad impar de errores.
  - Cuando hay una cantidad par de errores no detecta nada.
- Este código sólo detecta errores en una cantidad impar de bits, no corrige.

Códigos de Detección y Corrección de Errores – Código de Hamming

- Richard Hamming (1950) ideó un método no sólo para detectar errores sino también para corregirlos, y se conoce como **código Hamming.**
- Se añaden *k* bits de paridad a un carácter de *n* bits, formando un nuevo carácter de *n* + *k* bits. Los bits se numeran empezando por 1, no por 0, siendo el bit 1 el MSB.
- Todo bit cuyo número sea potencia de 2 es un bit de paridad y todos los demás se utilizan para datos.
- Para un carácter ASCII de 7 bits, se añaden 4 bits de paridad.
  - Los bits 1, 2, 4 y 8 son bits de paridad; 3, 5, 6, 7, 9, 10 y 11 son los 7 bits de datos.

Códigos de Detección y Corrección de Errores – Código de Hamming (cont.)

- Cada bit de paridad comprueba determinadas posiciones de bit y se ajusta de modo que el número total de unos en las posiciones comprobadas sea par, si se trata de paridad par; o sea impar, si se trata de paridad impar.
- En este código se pueden detectar errores en 1 o 2 bits, y también corregir errores en un solo bit.
  - Esto representa una mejora respecto a los códigos con bit de paridad, que pueden detectar errores en sólo un bit, pero no pueden corregirlo.

Códigos de Detección y Corrección de Errores – Código de Hamming (cont.)

- Las posiciones de los bits comprobados por los de paridad son:
  - El bit 1 comprueba los bits 1, 3, 5, 7, 9 y 11.
  - El bit 2 comprueba los bits 2, 3, 6, 7, 10 y 11.
  - El bit 4 comprueba los bits 4, 5, 6 y 7.
  - El bit 8 comprueba los bits 8, 9, 10 y 11.
- En general, el bit n es comprobado por los bits b_1, b_2,....,b_j, tales que $b_1 + b_2 + ... + b_j = n$.
  - Por ejemplo:
    - El bit 5 es comprobado por los bits 1 y 4 porque 1 + 4 = 5.
    - El bit 6 es comprobado por los bits 2 y 4 porque 2 + 4 = 6.

Códigos de Detección y Corrección de Errores – Código de Hamming (cont.)

- Ejemplo: Usando paridad par, construir el código de Hamming para el carácter "b“.

Código ASCII para “b”:

1	1	0	0	0	1	0

Código de Hamming para “b”:

1	2	3	4	5	6	7	8	9	10	11
		1		1	0	0		0	1	0

0	0	1	1	1	0	0	1	0	1	0

Códigos de Detección y Corrección de Errores – Código de Hamming (cont.)

- Considérese que pasaría si el bit 1 se modificara durante la transmisión.
  - El carácter recibido sería 10111001010 en lugar de 00111001010.
  - El receptor comprobaría los 4 bits de paridad con los resultados siguientes:
    - Bit de paridad 1 incorrecto: bits 1, 3, 5, 7, 9 y 11 contienen tres unos.
    - Bit de paridad 2 correcto: los bits 2, 3, 6, 7, 10 y 11 contienen dos unos.
    - Bit de paridad 4 correcto: los bits 4, 5, 6 y 7 contienen dos unos.
    - Bit de paridad 8 correcto: los bits 8, 9, 10 y 11 contienen dos unos.

Códigos de Detección y Corrección de Errores – Código de Hamming (cont.)

1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	0	0	1	0	1	0
1	**0**	1	1	1	0	0	1	0	1	0
1	0	1	**1**	1	0	0	1	0	1	0
1	0	1	1	1	0	0	**1**	0	1	0

Códigos de Detección y Corrección de Errores – Código de Hamming (cont.)

- El número total de unos en los bits 1, 3, 5, 7, 9 y 11 debería de ser par, ya que se está usando paridad par.
- El bit incorrecto debe ser uno de los bits comprobados por el bit de paridad 1, es decir, uno de los bits 1, 3, 5, 7, 9 u 11.
  - El bit de paridad 2 es correcto, sabemos que los bits 2, 3, 6, 7, 10 y 11 son correctos, de forma que el error no estaba en los bits 3, 7 u 11. Esto deja los bits 1, 5 y 9.
  - El bit de paridad 4 es correcto, lo cual significa que los bits 4, 5, 6 y 7 no contienen errores. Esto reduce la elección al 1 ó 9.
  - El bit de paridad 8 también es correcto y, por lo tanto, el bit 9 es correcto. Por consiguiente, el bit incorrecto debe ser el 1.
  - Dado que se recibió como un 1, debería haberse transmitido como un 0. En esta forma se pueden corregir los errores

Código de Huffman

- Código óptimo dentro de los códigos de codificación estadística, es el código de menor longitud media.
- A los símbolos con mayor frecuencia de aparición se les asignarán las palabras de código binario de menor longitud.
  - Se ordena el conjunto de símbolos del alfabeto fuente en orden creciente de probabilidades de aparición.
  - Se juntan los dos símbolos con menor probabilidad de aparición en un único símbolo, cuya probabilidad será la suma de las probabilidades de los símbolos que lo originaron.
  - Se repite este proceso hasta que sólo tengamos dos símbolos.

Código de Huffman (cont.)

- Se asigna un 1 a uno de los dos símbolos que tenemos y un 0 al otro.
- Recorreremos la estructura que hemos construido hacia atrás, cuando dos símbolos hayan dado origen a un nuevo símbolo, estos "heredarán" la codificación asignada a este nuevo símbolo.
- Se le añadirá un 1 a la codificación de uno de los símbolos y un 0 a la del otro símbolo.
- Sustituimos cada palabra del texto por el código respectivo y, una vez hecho esto, agrupamos los bits en grupos de ocho, es decir en bytes.

Ejemplo del Código de Huffman

- Se obtienen las frecuencias de cada palabra dentro del documento:

casa	29
nuevo	7
pesa	12
plato	5
sucio	4
tarde	8

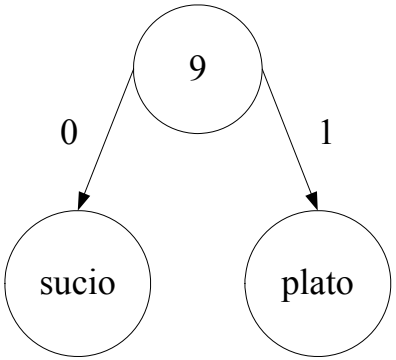
- Se ordenan las frecuencias en orden ascendente:

(sucio, plato, nuevo, tarde, pesa, casa)

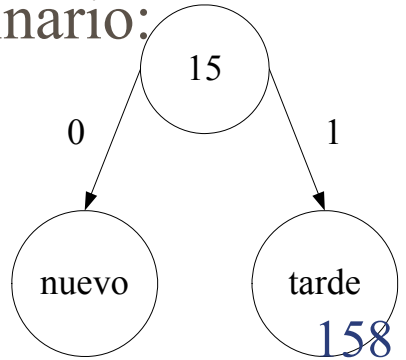
(4, 5, 7, 8, 12, 29)

Ejemplo del Código de Huffman (cont.)

- Luego se eligen los dos valores más pequeños y se construye un árbol binario con hojas etiquetadas:

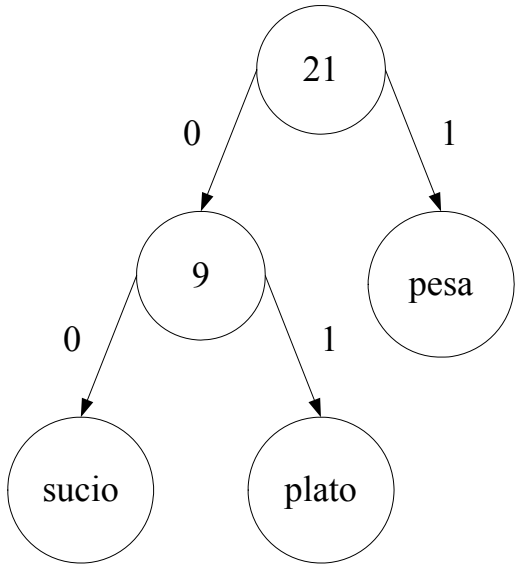


- Se reemplazan los dos valores por su suma, obteniéndose una nueva secuencia (7, 8, 9, 12, 29). De nuevo, se toman los dos valores más pequeños y se construye el árbol binario:



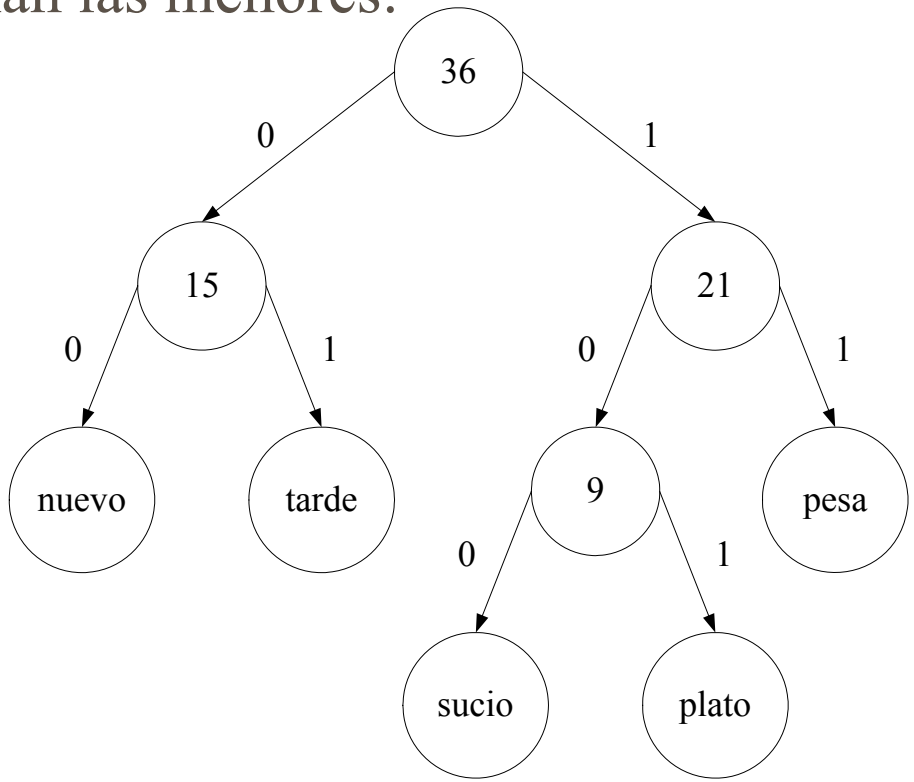
Ejemplo del Código de Huffman (cont.)

- Ahora se tienen las frecuencias (9, 12, 15, 29) y una vez más se seleccionan las menores:



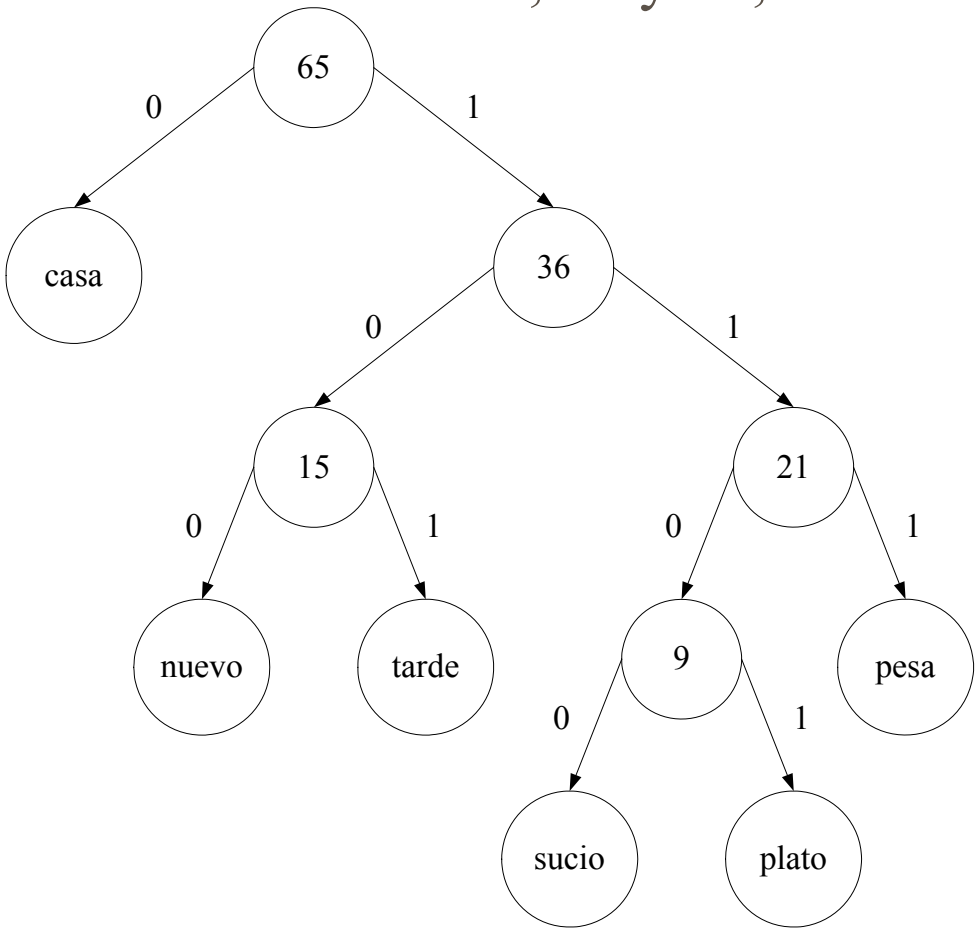
Ejemplo del Código de Huffman (cont.)

- Ahora se tienen las frecuencias (15, 21, 29) y una vez más se seleccionan las menores:



Ejemplo del Código de Huffman (cont.)

- Las dos frecuencias restantes, 29 y 36, se combinan en el árbol final:



Ejemplo del Código de Huffman (cont.)

- Del árbol anterior obtenemos el código para este alfabeto:

casa	0
nuevo	100
pesa	111
plato	1101
sucio	1100
tarde	101

- Sustituimos cada palabra del texto por el código respectivo y, una vez hecho esto, agrupamos los bits en grupos de ocho, es decir en bytes.

Referencias Bibliográficas

- Caballero Roldán, Rafael; Hortalá González, Teresa; Martí Oliet, Narciso; Nieva Soto, Susana; Pareja Lora, Antonio & Rodríguez Artalejo, Mario. "Matemática Discreta para Informáticos". Pearson Prentice Hall, Madrid. Primera Edición, 2007.
- Wikipedia. "Teoría de Números". URL: http://es.wikipedia.org/wiki/Teor%C3%ADa_de_n%C3%BAmeros. Modificado el 12 de julio del 2009.
- Wikipedia. "Divisibilidad". URL: http://es.wikipedia.org/wiki/Divisibilidad. Modificado el 16 de julio del 2009.

Referencias Bibliográficas (cont.)

- Wikipedia. “Tabla de Divisores”. URL: http://es.wikipedia.org/wiki/Anexo:Tabla_de_divisores. Modificado el 29 de abril del 2009.
- Wikipedia. “Tabla de Factores Primos”. URL: http://es.wikipedia.org/wiki/Anexo:Tabla_de_factores_primos . Modificado el 1de mayo del 2009.
- Código Binario Decimal. URL: http://es.wikipedia.org/wiki/C%C3%B3digo_binario_decimal.
- Tablas de Códigos. URL: http://www.isa.cie.uva.es/proyectos/codec/teoria2.html.

Referencias Bibliográficas

- Jonnsonbaugh, Richard. “Matemáticas Discretas”. Prentice Hall, México. Sexta Edición, 2005.
- Elizande, María Guadalupe. “Introducción a los Sistemas Computacionales”. URL: http://www.fismat.umich.mx/~elizalde/curso/curso.html.