API Documentation

API Documentation

April 12, 2015

Contents

1 Module cifrados

Modulo con las clases de cifrado

1.1 Variables

Name	Description
___package___	**Value:** `None`

1.2 Class Cesar

Cifrado Cesar

1.2.1 Methods

___init___(*self*)

Constructor de clase

setK(*self*, *k*)

Modificar de k

Parameters

`k:` valor de la clave

(type=entero)

setTexto(*self*, *texto*)

Modificador del texto

Parameters

`texto:` texto a cifrar

(type=cadena)

cifrar(*self*)

Cifrado del texto por el método César

Return Value

texto cifrado

(type=cadena)

descifrar(*self*)

Descifrado del texto por el método César

Return Value

texto descifrado

(type=cadena)

1.3 Class Afin

Cifrado Afín

1.3.1 Methods

___init___(*self*)

Constructor de clase

setA(*self*, *a*)

Modificador de la clave a

Parameters

`a:` valor de a

(type=entero)

setD(*self*, *d*)

Modificador de la clave d

Parameters

`d:` valor de d

(type=entero)

setTexto(*self*, *texto*)

Modificar del texto

Parameters

`texto:` cadena a cifrar

(type=cadena)

cifrar(*self*)

Cifrado del texto por el método afín

Return Value

texto cifrado

(type=cadena)

descifrar(*self*)

Descifrado del texto por el método afín

Return Value

texto descifrado

(type=cadena)

1.4 Class Mochila

Cifrado mochila o de Merkle-Hellman

1.4.1 Methods

___init___(*self*)

Constructor de la clase

setMochila(*self*, *mochila*)

Modificador de la clave mochila

Parameters

`mochila:` valor de la clave mochila

(type=lista)

setM(*self*, *m*)

Modificador de la clave m

Parameters

`m:` valor de la clave m

(type=entero)

setW(*self*, *w*)

Modificador de la clave w

Parameters

`w:` valor de la clave w

(type=entero)

setPublic(*self*, *public*)

Modificador de la clave publica

Parameters

`public:` valor de la lista publica

(type=lista)

setTexto(*self*, *texto*)

Modificador del texto a cifrar/descifrar

Parameters

`texto:` texto a cifrar/descifrar

(type=cadena)

setClaves(*self*, *mochila*, *m*, *w*, *public*)

Modificador de todas las claves

Parameters

`mochila:` cadena con los valores de la mochila

(type=cadena)

`m:` valor de la clave m

(type=entero)

`w:` valor de la clave w

(type=entero)

`public:` cadena con los valroes de la clave publica

(type=cadena)

cifrar(*self*)

Cifrado por el método mochila

Return Value

lista con los valores del cifrado

(type=lista)

descifrar(*self*)

Descifrado por el método mochila

Return Value

texto descifrado

(type=cadena)

generarClave(*self*)

Genera las claves aleatoriamente y las modifica en la clase

descifraNumero(*self*, *texto*, *mochila*)

Consigue la cadena de binarios para descifrar

Return Value

cadena con los numeros binarios para descifrar el texto

(type=cadena)

1.5 Class RSA

Cifrado RSA

1.5.1 Methods

___init___(*self*)

Constructor de clase

setN(*self*, *n*)

Modificador de la clave n

Parameters

`n:` valor de la clave n

(type=entero)

setE(*self*, *e*)

Modificador de la clave e

Parameters

`e:` valor de la clave e

(type=entero)

setD(*self*, *d*)

Modificador de la clave d

Parameters

`d:` valor de la clave d

(type=entero)

setTexto(*self*, *texto*)

Modificador del texto a cifrar

Parameters

`texto:` texto a descifrar

(type=cadena)

cifrar(*self*, *e*, *n*, *texto*)

Cifrado por el método RSA

Parameters

`e:` valor de e

(type=entero)

`n:` valor de la clave n

(type=entero)

`texto:` texto a cifrar

(type=cadena)

Return Value

valores del texto cifrado

(type=lista)

descifrar(*self*, *d*, *n*, *cifrado*)

Descifrado por el método RSA

Parameters

`d:` clave d

(type=entero)

`n:` clave n

(type=entero)

`cifrado:` lista con los valores a descifrar

(type=lista)

Return Value

texto descifrado

(type=cadena)

generarFirma(*self*, *texto*)

Genera una firma digital por RSA

Parameters

`texto:` texto con el que generar la firma

(type=cadena)

Return Value

valor de la firma digital

(type=entero)

generarClave(*self*)

Generacion de claves para RSA

2 Module funciones

Modulo con las funciones generales

2.1 Functions

letranumero(*texto*)

Convierte una letra a su numero correposdiente

Parameters

`texto:` cadena a convertir a numeros

(type=cadena)

Return Value

lista de numeros

(type=lista)

numeroLetra(*num*)

Convierte un numero a su letra correspondiente

Parameters

`num:` lista de numeros a convertir

(type=lista)

Return Value

cadena de los numeros convertidos a letras

(type=cadena)

abecedario()

Alfabeto a usar

Return Value

alfabeto a usar

(type=cadena)

egcd(*a*, *b*)

MCD de dos numeros

Parameters

`a:` primer elemento para calcular el mcd

(type=entero)

`b:` segundo elemento para calcular el mcd

(type=entero)

Return Value

Maximo común divisor y los coeficientes de Bézout

(type=lista)

modinv(*a*, *m*)

Calcula el inverso de un numero modulo m

Parameters

`a:` numero a calcular el inverso

(type=entero)

`m:` modulo en el que calcular el inverso

(type=entero)

Return Value

inverso de a modulo m

(type=entero)

numeroBinario(*texto*)

Convierte una cadena en su respectivos numeros binarios de 5 bits

Parameters

`texto:` texto a convertir

(type=cadena)

Return Value

cadena de numeros de 5 bits

(type=cadena)

binarioNumero(*texto*)

Convierte una cadena de numeros binarios de 5 bits a numeros decimales

Parameters

`texto:` cadena de numeros binarios de 5 bits

(type=cadena)

Return Value

cadena numeros decimales

(type=lista)

generarPrimos(*n*)

Genera un primo aleatorio

Parameters

`n:` escoge entre numeros primos de 0 a 5000 o de 0 a 150

(type=entero)

Return Value

numero primo

(type=entero)

listacadena(*lista*)

Convierte una lista en cadena

Parameters

`lista:` lista a convertir

(type=lista)

Return Value

lista convertida a cadena

(type=cadena)

cadenaLista(*cadena*)

Convierte una cadena de numeros separados por ',' en lista

Parameters

`cadena:` cadena a convertir

(type=cadena)

Return Value

cadena convertida a lista

(type=lista)

letraNumero2D(*texto*)

Convierte la cadena en una cadena de numeros de dos cifras

Parameters

`texto:` cadena a convertir en numeros

(type=cadena)

Return Value

cadena de numeros de dos cifras

(type=cadena)

dec2bin(*a*)

Convierte un entero a binario

Parameters

`a:` numero a convertir

(type=entero)

Return Value

numero binario

(type=cadena)

potencia(*c*, *d*, *n*)

Potencia modulo n

Parameters

`c:` numero a elevar

(type=entero)

`d:` potencia a la que elevar c

(type=entero)

`n:` modulo

(type=entero)

Return Value

c elevado a d modulo n

(type=entero)

reshape(*cadena*, *n*)

Divide una cadena en una lista de elementos de longitud n

Parameters

`cadena:` cadena a dividir

(type=cadena)

`n:` longitud de los elementos de la lista

(type=entero)

Return Value

lista con la cadena dividida

(type=lista)

prepa_num_cifrar(*n*, *texto*)

Prepara el texto para poder cifrarlo por RSA

Parameters

`n:` clave n

(type=entero)

`texto:` texto a cifrar

(type=cadena)

Return Value

lista con los numeros preparados para cifrar

(type=lista)

num_letra(*n*, *cifrado*)

descifra el vector para descifrarlo por RSA

Parameters

`n:` clave n

(type=entero)

`cifrado:` lista de numeros a descifrar

(type=lista)

Return Value

texto descifrado

(type=cadena)

hash(*s*, *M*)

Funcion Hash

Parameters

`s:` cadena con la que realizar el hash

(type=cadena)

`M:` modulo de la funcion

(type=entero)

Return Value

resumen del mensaje

(type=entero)

2.2 Variables

Name	Description
___package___	**Value:** `None`

Index