Explicación RSA.cpp

Función gcd(): Calcula el máximo común divisor (GCD) entre dos números a y b utilizando el algoritmo de Euclides.

Argumentos:

- a (long long): Primer número.
- b (long long): Segundo número.

Retorna el GCD de ambos números.

Función modInverse(): Calcula el inverso modular de e respecto a phi. Esto es necesario para encontrar el valor de d, que es la clave privada en RSA.

Argumentos:

- e (long long): El exponente público.
- phi (long long): El valor de ϕ(n) (phi de n).

Retorna el valor de x, el cual corresponde al inverso modular de e.

Función modExp(): Esta función realiza la exponenciación modular, lo que significa calcular (base^exp) % mod de manera eficiente, utilizando el método de exponenciación binaria. Esta operación es esencial para el cifrado y descifrado en RSA.

Argumentos:

- base (long long): La base de la potencia.
- exp (long long): El exponente.
- mod (long long): El módulo.

Retorna el resultado de (base^exp) % mod.

Función encryptOneChar(): Cifra un solo carácter usando la clave publica (e, n).

Argumentos:

- ch (char): El carácter a cifrar.

Retorna el valor cifrado del carácter ch.

Función encrypt(): Cifra un texto completo (una cadena de caracteres) utilizando la función encryptOneChar para cada carácter.

Argumentos:

- plain_Text (const char): Cadena de texto plano que se desea cifrar.
- cipher_Text (long long): Arreglo donde se almacenará el texto cifrado.

No retorna ningún valor debido a que el cifrado se almacena directamente en cipher_Text.

Función decryptOneChar(): Descifra un solo carácter usando la clave privada (d, n).

Argumentos:

- encryptedChar (log long): Valor cifrado que se desea descifrar.

Retorna el carácter descifrado.

Función decrypt(): Descifra un arreglo de texto cifrado y lo convierte de nuevo en texto plano.

Argumentos:

- cipher_Text (const char): Arreglo de valores cifrados.
- decrypt_Text (long long): Arreglo donde se almacenará el texto descifrado.

No retorna ningún valor debido a que el descifrado se almacena directamente en decrypt_Text.

Función generateRSAKeys (): Genera las claves pública y privada necesarias para el cifrado RSA utilizando dos números primos p y q.

Argumentos:

- p (long long): Primer número primo.
- q (long long): Segundo número primo.

No retorna ningún valor.

Explicación BasicUsageRSA.ino

Este código de ejemplo utiliza la librería RSA para cifrar y descifrar un texto en una placa Arduino. A continuación, se destacan los puntos importantes:

Incluir Bibliotecas: Se incluyen Arduino.h para funciones básicas de Arduino y RSA.h para las operaciones de cifrado RSA.

Variables Globales:

Se definen dos números primos (p = 991 y q = 997) que se usan para generar las claves pública y privada.

plain_Text es el texto que se desea cifrar.

cipher_Text es un arreglo de tipo long long que almacenará el texto cifrado.

decrypted_Text es un arreglo de tipo char que almacenará el texto descifrado.

Instancia de RSA: Se crea una instancia de la clase RSA para manejar las operaciones de cifrado y descifrado.

Función setup():

Genera las claves RSA utilizando p y q.

Cifra el texto plain_Text y almacena los resultados en cipher_Text.

Descifra el texto cifrado y lo almacena en decrypted_Text.

Muestra el texto original, el texto cifrado y el texto descifrado en el monitor serial.

Cifrado y Descifrado:

Cifrado: Convierte cada carácter del texto en un número grande usando la clave pública.

Descifrado: Convierte cada número grande de vuelta al carácter original usando la clave privada.

Tipos de Datos:

long long se usa para manejar los grandes números requeridos por el algoritmo RSA.

char[] se usa para manejar las cadenas de texto.