Explicación ChaCha20.cpp

Constructor ChaCha20(): Inicializa la instancia, recibiendo como argumentos parámetros desde el archivo .ino, que se van a usar de manera global en la librería.

Argumentos:

- key (byte*): Puntero a un arreglo de bytes que contiene la clave de 256 bits.
- nonce (byte*): Puntero a un arreglo de bytes que contiene el nonce de 64 bits.
- counter (byte*): Puntero a un arreglo de bytes que contiene el contador inicial de 64 bits.

Función quarterRund(): Es una función de mezcla básica en el algoritmo para transformar los bloques de datos.

Argumentos:

- a (uint32_t &): Referencia a una de las 4 palabras (32 bits) de entrada que serán mezcladas.
- b (uint32_t &): Referencia a la segunda palabra.
- c (uint32_t &): Referencia a la tercera palabra.
- d (uint32_t &): Referencia a la cuarta palabra.

Función generateKeystream(): Genera el keystream (flujo de clave) a partir del estado inicial, que se usa para cifrar o descifrar el texto.

Argumentos:

- state (uint32_t [16]): Arreglo que representa el estado inicial de 512 bits (16 palabras de 32 bits).
- Keystream (uint32_t [16]): Arreglo donde se almacenará el keystream generado.

No retorna ningún valor debido a que modifica directamente el arreglo keystream.

Función encrypt(): Cifra el texto plano utilizando el keystream generado por el algoritmo ChaCha20.

Argumentos:

- plaintext (const char*): Puntero al texto plano que se desea cifrar.
- ciphertext (byte*): Puntero donde se almacenará el texto cifrado.
- length (size_t): Longitud del texto plano.

No retorna ningún valor debido a que modifica directamente el arreglo ciphertext.

Función decrypt(): Descifra un texto cifrado utilizando el keystream generado por el algoritmo ChaCha20.

Argumentos:

- ciphertext (const byte*): Puntero al texto cifrado que se desea descifrar.
- decryptedText (char*): Puntero donde se almacenará el texto descifrado.
- length (size_t): Longitud del texto cifrado.

No retorna ningún valor debido a que modifica directamente el arreglo decryptedText.

Función initializeState(): Inicializa el estado del algoritmo ChaCha20 con la constante, la clave, el contador y el nonce.

Argumentos:

- state(uint32_t [16]): Arreglo donde se almacenará el estado inicial.
- key (const byte*): Puntero a la clave de 256 bits.
- counter(const byte*): Puntero al contador de 64 bits.
- nonce (const byte*): Puntero al nonce de 64 bits.

No retorna ningún valor debido a que modifica directamente el arreglo state.

En el archivo **ChaCha20.h** se declaran state y keystream como privadas, lo que significa que estas variables solo pueden ser accedidas y modificadas directamente por las funciones miembro de la clase `ChaCha20`. Ningún código fuera de la clase puede acceder o alterar estas variables directamente.

state[16]: Es un arreglo de 16 elementos de tipo `uint32_t` que representa el estado interno del algoritmo ChaCha20. Este estado incluye la constante, la clave, el nonce y el contador. Se inicializa y se actualiza durante el proceso de generación del keystream y es fundamental para el funcionamiento del cifrado.

keystream[16]: Es un arreglo de 16 elementos de tipo `uint32_t` que almacena el keystream generado a partir del estado. Este keystream se utiliza para cifrar y descifrar los datos, aplicando una operación XOR con el texto plano o el texto cifrado.

Explicación BasicUsageCHaCha20.ino

Definición de Claves, Nonce y Counter

key: Arreglo de 32 bytes que representa la clave de cifrado.

nonce: Arreglo de 8 bytes que actúa como un valor único para evitar repeticiones.

counter: Arreglo de 8 bytes utilizado como contador para el cifrado, que se incrementa en cada bloque.

Estos tres elementos son fundamentales para el funcionamiento del algoritmo ChaCha20.

Texto Plano y Tamaño

plain_Text: Este es el texto que queremos cifrar, definido como un arreglo de tipo char.

size_Plain_Text: Calcula el tamaño del texto plano, incluyendo el carácter nulo \0.

Instancia de ChaCha20

ChaCha20 chacha20(key, nonce, counter): Crea una instancia de la clase ChaCha20 utilizando la clave, el nonce y el contador definido anteriormente. Esta instancia manejará las operaciones de cifrado y descifrado.

Función setup():

cipher_Text: Arreglo de bytes donde se almacenará el texto cifrado.

decrypted_Text: Arreglo de char donde se almacenará el texto desencriptado.

Cifrado y descifrado

Cifra el texto plano utilizando ChaCha20 y almacena el resultado en cipher_Text. Se resta 1 del tamaño para excluir el terminador nulo.

Descifra el texto cifrado y almacena el resultado en decrypted_Text.

Impresión de Resultados:

Texto plano: Se imprime el texto original en el monitor serial.

Texto cifrado: Se imprime el texto cifrado en formato hexadecimal para facilitar su visualización.

Texto desencriptado: Se imprime el texto desencriptado para verificar que coincide con el texto original.

Función printBytesAsHex

Esta función recorre el arreglo de bytes cipher_Text y convierte cada byte en su representación hexadecimal para imprimirlo en el monitor serial. Si el valor es menor que 16, se añade un '0' al principio para mantener el formato de dos dígitos.