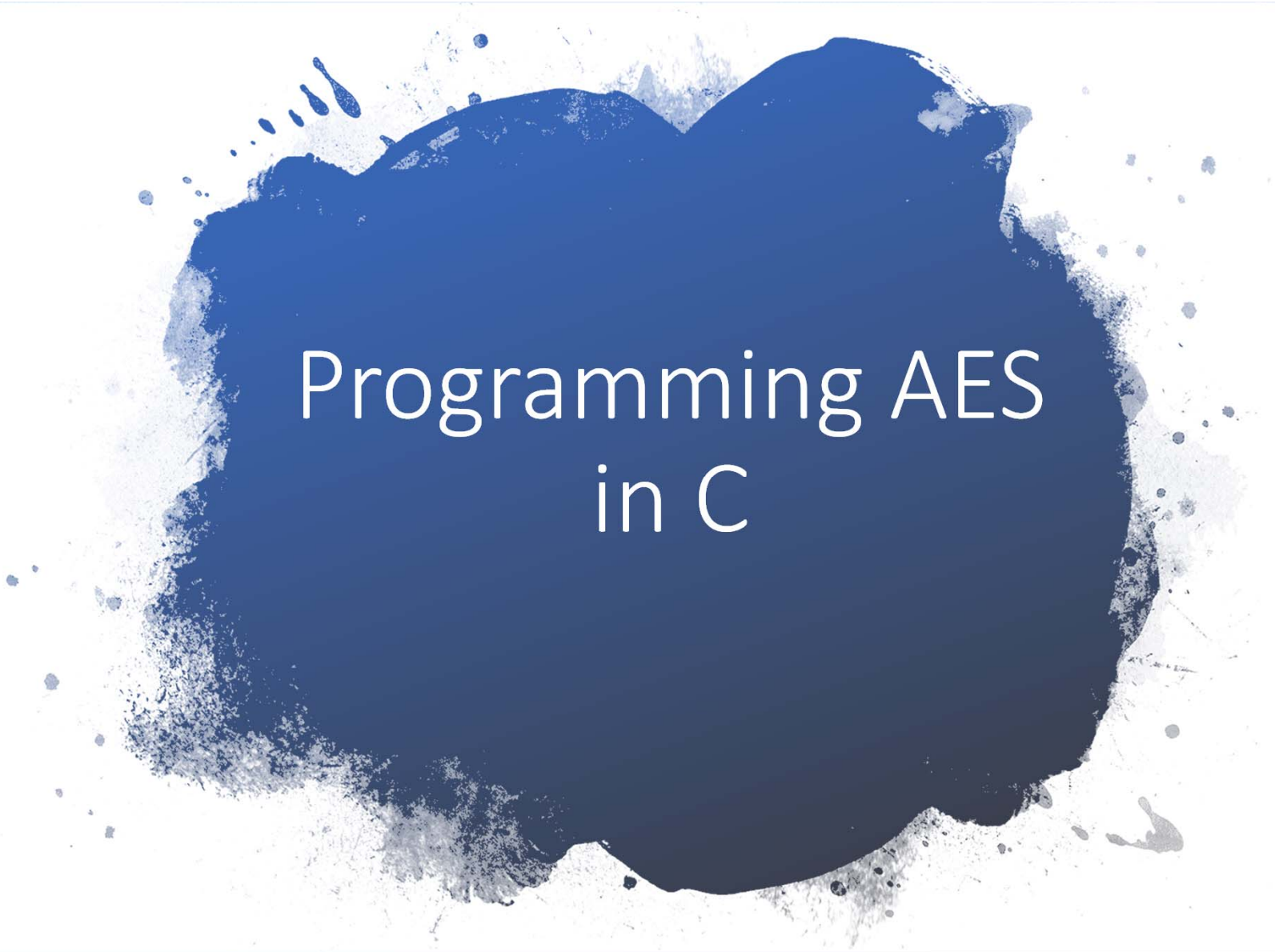

Programming AES
in C

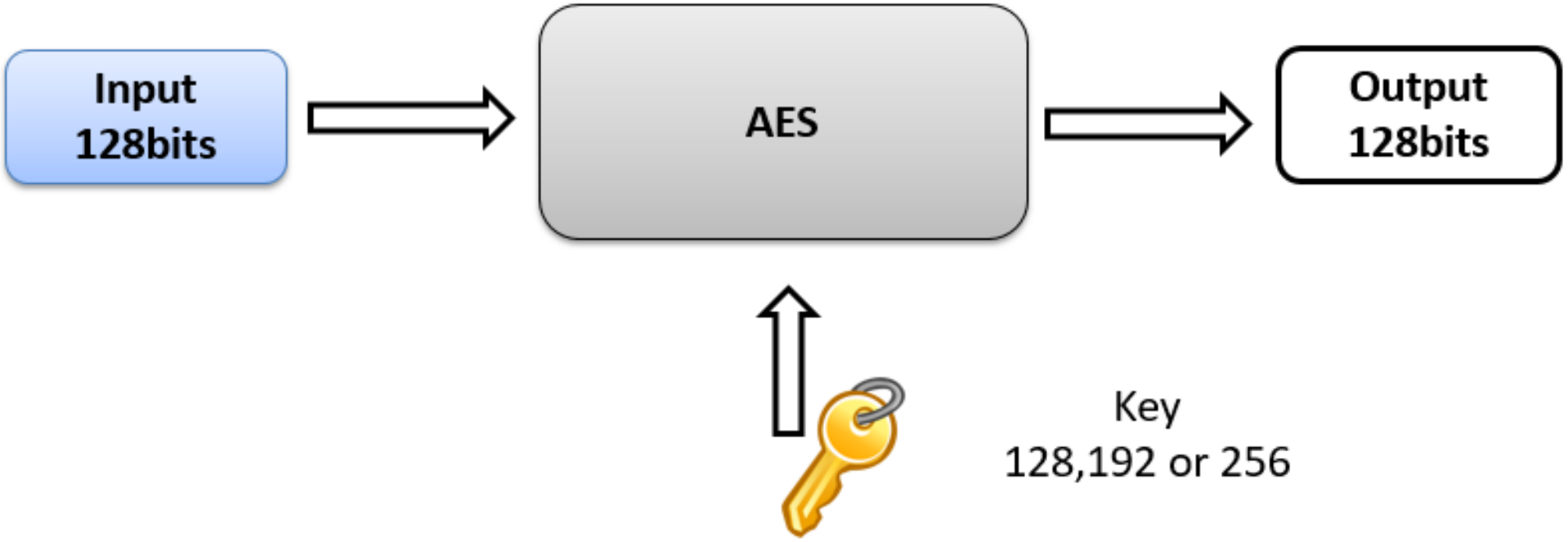
Programming AES in C

- Step 1:
  - Write a simple program that will print “HELLO world!” and compile it with the following command line
    - ./gcc –Wall first.c –lcrypto –o first
    - In your program add the following libraries
      - #include <stdio.h>
      - #include <openssl/evp.h>
      - #include <openssl/aes.h>
      - #include <openssl/err.h>
      - #include <string.h>

- int main(int arc, char *argv[])

```
{
        printf (“Hello WORLD\n”);
        return 1;
}
```

Advanced Encryption Standard



Initialization

- int main(int arc, char *argv[])

```
{
    OpenSSL_add_all_algorithms();  //load all cipher algorithms
    ERR_load_crypto_strings(); //load human readable errors

    //the above two lines are needed for initialization and to be able to use the
    libcrypto library
```

Calling **encrypt()** function

- What do we need for the encrypt function?
  - Plaintext
  - Length of the plaintext
  - Key
  - IV?
- Encrypting consists of the following stages:
  - Setting up a context
  - Initializing the encryption operation
  - Providing plaintext bytes to be encrypted
  - Finalizing the encryption operation

Calling Encrypt() function

- int main(int arc, char *argv[])

```
{
        OpenSSL_add_all_algorithms();
        ERR_load_crypto_strings();
        /* A 256 bit key */
        static const unsigned char key[] = "01234567890123456789012345678901";
unsigned char plaintext[] = "CS475 is an awesome course about computer security in the University of Cyprus.";

        /* Buffer to store the ciphertext. The size may be different due to padding.  */
        unsigned char ciphertext[128];

    /* Buffer for the decrypted text for verifying decryption.  */
    unsigned char decryptedtext[128];
int decryptedtext_len = 0, ciphertext_len = 0;
```

calling encrypt() function

- /* Encryption. */
  ciphertext_len = encrypt(plaintext, strlen((char *)plaintext), key, ciphertext);
  printf("Ciphertext is:\n");
  BIO_dump_fp(stdout, (const char *)ciphertext, ciphertext_len);

- }

Function: encrypt() internals

- int encrypt(unsigned char *plaintext, int plaintext_len, const unsigned char *key, unsigned char *ciphertext)

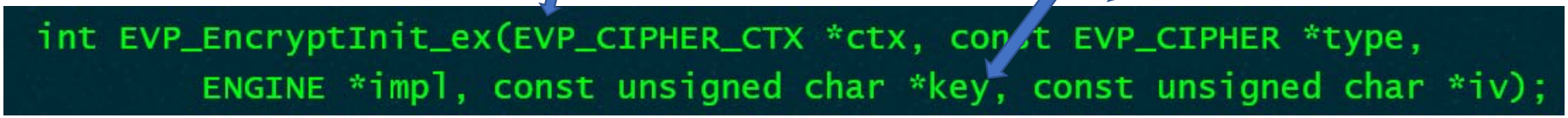
```
{
    EVP_CIPHER_CTX *ctx = NULL;
    int len = 0, ciphertext_len = 0;

    /* Create and initialize the context  and returns a pointer for success and null for failure*/
     if(!(ctx = EVP_CIPHER_CTX_new()))
            handleErrors();

     /* Initialize the encryption operation. */
    if(1 != EVP_EncryptInit_ex(ctx, EVP_aes_128_ecb(), NULL, key, NULL))
            handleErrors();
```

Function: encrypt()

```
/* Initialize the encryption operation. */
if(1 != EVP_EncryptInit_ex(ctx, EVP_aes_128_ecb(), NULL, key, NULL))
        handleErrors();
```



Function: EVP_EncryptUpdate

/* Provide the message to be encrypted, and obtain the encrypted output.
* EVP_EncryptUpdate can be called multiple times if necessary*/

```
int EVP_EncryptUpdate(EVP_CIPHER_CTX *ctx, unsigned char *out,
        int *outl, const unsigned char *in, int inl);
```

```
if(plaintext)
 {
if(1 != EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintext_len))
 handleErrors();
 ciphertext_len = len;
 }
```

encrypted version

actual number of bytes written

Function: EVP_EncryptFinal_ex

```
if(1 != EVP_EncryptFinal_ex(ctx, ciphertext + len, &len))
        handleErrors();
    ciphertext_len += len;
```

encrypted final data is written

```
int EVP_EncryptFinal_ex(EVP_CIPHER_CTX *ctx, unsigned char *out,
        int *outl);
```

- Encryptfinal_ex encrypts the final data that is any data that remains in a partial block.

number of bytes written

Function: EVP_CIPHER_CTX_free

```
/* Clean up */
 EVP_CIPHER_CTX_free(ctx);
return ciphertext_len;
}
```

```
void EVP_CIPHER_CTX_free(EVP_CIPHER_CTX *ctx);
```

EVP_CIPHER_CTX_FREE clears all information from a cipher context and free up any allocated memory associate with it, including **ctx** itself.

Resources

- https://www.openssl.org/docs/man1.1.0/crypto/EVP_EncryptInit.html
- https://wiki.openssl.org/index.php/Libcrypto_API