SGX

叶梓淳

2023 年 6 月 10 日

1 SGX

1.1 Enclave.edl

EDL 文件中声明了在 Enclave 中使用的 ECALL 和 OCALL 接口，trusted 部分是 ECALL，untrusted 部分是 OCALL。而 ECALL 中需要声明三个函数：

- ecall_sbox_generation()，用于生成 S 盒
- ecall_keystream_generation()，用于生成流密钥
- ecall_decryption([in, size=len] unsigned char* ciphertext, size_t len)，用于解密

Enclave.edl 代码如下：

```
enclave {

    from "Edger8rSyntax/Types.edl" import *;
    from "Edger8rSyntax/Pointers.edl" import *;
    from "Edger8rSyntax/Arrays.edl" import *;
    from "Edger8rSyntax/Functions.edl" import *;

    from "TrustedLibrary/Libc.edl" import *;
    from "TrustedLibrary/Libcxx.edl" import ecall_exception, ecall_map;
    from "TrustedLibrary/Thread.edl" import *;

    untrusted {
```

```
13          void ocall_print_string([in, string] const char *str);
14      };
15
16      trusted {
17          public void ecall_sbox_generation();
18          public void ecall_keystream_generation();
19          public void ecall_decryption([in, size=len] unsigned char*
                ciphertext, size_t len);
20      };
21  };
```

1.2 Enclave.cpp

Enclave.cpp 需要定义必要的全局变量和完善上述三个函数。这里将流密钥和明文均定义为全局变量，方便进行运算、输出。RC4 加密算法流程查找资料获得。

```
1   // RC4 decryption implementation in Intel SGX
2   const char* k = "gosecgosec";
3   unsigned char* key = (unsigned char*)k;
4   unsigned char K[256];
5   unsigned char S[256];
6   unsigned char keystream[256];
7   unsigned char plaintext[256];
8
9   void ecall_sbox_generation()
10  {
11      for (int i = 0; i < 256; ++i)
12      {
13          S[i] = i;
14          K[i] = key[i % 10];
15      }
16      int j = 0;
17      for (int i = 0; i < 256; ++i)
18      {
19          j = (j + S[i] + K[i]) % 256;
20          unsigned char temp = S[i];
21          S[i] = S[j];
```

```
22          S[j] = temp;
23      }
24  }
25
26  void ecall_keystream_generation()
27  {
28      int i = 0;
29      int j = 0;
30      for (int k = 0; k < 256; ++k)
31      {
32          i = (i + 1) % 256;
33          j = (j + S[i]) % 256;
34          unsigned char temp = S[i];
35          S[i] = S[j];
36          S[j] = temp;
37          int t = (S[i] + S[j]) % 256;
38          keystream[k] = S[t];
39      }
40  }
41
42  void ecall_decryption(unsigned char* ciphertext, size_t len)
43  {
44      const char* temp = reinterpret_cast<const char*>(ciphertext);
45      for (int i = 0; i < strlen(temp); ++i)
46      {
47          plaintext[i] = ciphertext[i] ^ keystream[i];
48      }
49      printf("plaintext: %s\n", plaintext);
50  }
```

这里 printf 会自动调用 App.cpp 中的 ocall_print_string 函数，通过 OCALL 输出明文。

1.3 App.cpp

App.cpp 需要修改主函数的内容：传入密文，并调用三个 ECALL 函数即可。

```
1
2  int SGX_CDECL main(int argc, char *argv[])
3  {
4      (void)(argc);
5      (void)(argv);
6
7
8      /* Initialize the enclave */
9      if(initialize_enclave() < 0){
10         printf("Enter a character before exit ...\n");
11         getchar();
12         return -1;
13     }
14
15     unsigned char ciphertext[] = {0x1c, 0x7b, 0x53, 0x61, 0x6e, 0x81,
           0xce, 0x8a, 0x45, 0xe7, 0xaf, 0x39, 0x19, 0xbc, 0x94, 0xab,
           0xa4, 0x12, 0x58};
16     ecall_sbox_generation(global_eid);
17     ecall_keystream_generation(global_eid);
18     ecall_decryption(global_eid, ciphertext, sizeof(ciphertext));
19
20     printf("Info: SampleEnclave successfully returned.\n");
21     printf("Enter a character before exit ...\n");
22
23     /* Utilize edger8r attributes */
24     edger8r_array_attributes();
25     edger8r_pointer_attributes();
26     edger8r_type_attributes();
27     edger8r_function_attributes();
28
29     /* Utilize trusted libraries */
30     ecall_libc_functions();
31     ecall_libcxx_functions();
32     ecall_thread_functions();
33
34     /* Destroy the enclave */
35     sgx_destroy_enclave(global_eid);
36     getchar();
```

```
37      return 0;
38  }
```

1.4 输出结果

按照指导文档编译、运行 app 文件，得到 flag:

```
test@9-19:~/sgxsdk/SampleCode/SampleEnclave$ ./app
plaintext: flag{Intel_SGX_TEE}
Info: SampleEnclave successfully returned.
Enter a character before exit ...
Checksum(0x0x7ffc515e9430, 100) = 0xfffd4143
Info: executing thread synchronization, please wait...
Info: SampleEnclave successfully returned.
Enter a character before exit ...
```

图 1: 运行结果