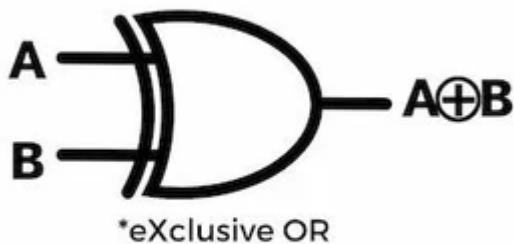


# [C] XOR Encryption

## Introduction

Le OU exclusif, ou le XOR, est très souvent utilisé dans les algorithmes de chiffrement et les logiciels malveillants.

C'est pourquoi je partage une fonction permettant de chiffrer en XOR.



2 input XOR gate

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

## Code

Voici un simple XOR avec un simple caractère :

```
#define KEY 'I';

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

unsigned char*shellcodeXor(unsigned char*shellcode, char key){
```

```

char*shellcodeXored = (char*)malloc(sizeof(shellcode));
// Xor each char until the nullbyte
for(int i=0;i<shellcode[i];i++) shellcodeXored[i] = shellcode[i] ^ key;
return shellcodeXored;
}

void*printShellcode(unsigned char*shellcode){
    // Print each char until the nullbyte
    for(int i=0;shellcode[i];i++) printf("\\x%x", shellcode[i]);
    printf("\\n");
}

int main(){

    char key = KEY;
    unsigned char sc[] = {"\x48\x31\xc9\x48\x48"};
    int scLen = strlen(sc);

    // XOR process
    unsigned char*scXor = shellcodeXor(sc, key);
    printShellcode(scXor);

    // XOR again (to unxor)
    unsigned char*scDec = shellcodeXor(scXor, key);
    printShellcode(scDec);

    printf("Sizeof shellcode : [%d]\\n", scLen);

    free(scDec);

    return 0;
}

```

Voici une alternative avec une clé de plusieurs caractères (la clé doit être une chaîne de caractère ASCII) :

```

#define KEY "14394e85c196c4274d621e3c9924df46e5218bab1450875c030d06f2f0991f2e";

#include <stdio.h>
#include <stdlib.h>

```

```
unsigned char*shellcodeXor(unsigned char*shellcode, char*key){
```

```
    char*shellcodeXored = (char*)malloc(sizeof(shellcode));
```

```
    // Parse and xor with all char of key
```

```
    for(int i=0;key[i];i++){
```

```
        // Xor each char until the nullbyte
```

```
        for(int j=0;j<shellcode[j];j++) {
```

```
            shellcodeXored[j] = shellcode[j] ^ key[i];
```

```
        }
```

```
    }
```

```
    return shellcodeXored;
```

```
}
```

```
void*printShellcode(unsigned char*shellcode){
```

```
    // Print each char until the nullbyte
```

```
    for(int i=0;shellcode[i];i++) printf("\\x%x", shellcode[i]);
```

```
    printf("\\n");
```

```
}
```

```
int main(){
```

```
    char*key = KEY;
```

```
    unsigned char sc[] = {"\x48\x31\xc9\x48\x72"};
```

```
    int scLen = sizeof(sc)-1; // Don't count the null byte
```

```
    // XOR process
```

```
    unsigned char*scXor = shellcodeXor(sc, key);
```

```
    printShellcode(scXor);
```

```
    // XOR again (to unxor)
```

```
    unsigned char*scDec = shellcodeXor(scXor, key);
```

```
    printShellcode(scDec);
```

```
    printf("Sizeof shellcode : [%d]\\n", scLen);
```

```
    free(scDec);
```

```
    return 0;
```

```
}
```

---

Revision #9

Created 30 July 2024 20:24:32 by Elieroc

Updated 31 July 2024 12:39:36 by Elieroc