

OpenSSL 라이브러리 헤더 및 매뉴얼

1. Open SSL : BIGNUM(BN)

<https://github.com/openssl/openssl/blob/master/include/openssl/bn.h>

2. Open SSL : AES

<https://github.com/openssl/openssl/blob/master/include/openssl/aes.h>

3. BIGNUM 매뉴얼

<https://www.openssl.org/docs/man1.0.2/man3/bn.html>

BN(BIGNUM) 변수 선언 및 메모리 할당 / type casting / free memory 예제

```
1 ▾ #include <stdio.h>
2   #include <openssl/bn.h>
3
4 ▾ int main(int argc, char* argv[]) {
5     BIGNUM *key = BN_new(); // BIGNUM 변수 선언 및 메모리 할당
6     BN_set_word(key, 400); // BN변수 key에 400 대입
7
8     printf("BN_bn2dec: %s\n", BN_bn2dec(key)); // 10진수 형태 문자열로 반환 ==> "400"
9     printf("BN_bn2hex: %s\n", BN_bn2hex(key)); // 16진수 형태 문자열로 반환 ==> "0190"
10
11     unsigned char to[32];
12     int len = BN_bn2bin(key, to); // BIGNUM에서 binary 형태로 변환 ==> to = {0x01, 0x90}
13     printf("BN_bn2bin: ");
14 ▾ for (int i=0; i<len; i++) {
15     |     printf("%x", to[i]);
16     | }
17     printf("\n");
18
19     BN_free(key); ➡ ※ 동적할당된 메모리 해제
20
21     return 0;
22 }
```

출력 결과 :


```
BN_bn2dec: 400
BN_bn2hex: 0190
BN_bn2bin: 190
```

주의!!

BIGNUM *BN_bin2bn(const unsigned char *s, int len, BIGNUM *ret);
같은 경우, len 에 strlen(s)를 넣어야 잘 돌아간다.

BN(BIGNUM) 난수 생성 / 덧셈, 곱셈 연산 예제

```
1 #include <stdio.h>
2 #include <openssl/bn.h>
3
4 int main(int argc, char* argv[]) {
5     BIGNUM *key = BN_new();
6     BIGNUM *A = BN_new();
7     BIGNUM *B = BN_new();
8     BIGNUM *C = BN_new();
9     BN_set_word(A, 5);
10    BN_set_word(B, 6);
11
12    // 최대 128bit 난수를 key에 저장
13    BN_rand(key, 128, BN_RAND_TOP_ANY, BN_RAND_BOTTOM_ANY);
14    printf("key: %s\n", BN_bn2hex(key));
15
16    // A > B => return 1, A == B => return 0, A < B => return -1
17    printf("BN_cmp(A, B): %d\n", BN_cmp(A, B));
18
19    // C = A + B
20    BN_add(C, A, B);
21    printf("A + B = %s\n", BN_bn2dec(C));
22
23    // C = A * B
24    BN_CTX *ctx = BN_CTX_new();
25    BN_mul(C, A, B, ctx);
26    BN_CTX_free(ctx);
27    printf("A * B = %s\n", BN_bn2dec(C));
28
29    BN_free(key);
30    BN_free(A);
31    BN_free(B);
32    BN_free(C);
33
34
35    return 0;
36 }
```



```
key: E77110FD6FD97DDAD828F0BE26014910
BN_cmp(A, B): -1
A + B = 11
A * B = 30
```

출력 결과 (key는 실행시마다 바뀜)

BN_sub(), BN_sqr(), BN_div, BN_mod() 등의
다양한 연산 지원

BN_copy(), BN_swap() 등 편의성 함수 존재

※ 매뉴얼(또는 헤더 파일) 참조

BN_xor 구현 예제

```
int BN_xor(BIGNUM *b_r, int bits, const BIGNUM *b_a, const BIGNUM *b_b)
{
    //error
    if(b_r==NULL || b_a == NULL || b_b == NULL)
        return 0;
    //bytes = bits / 8
    int i, bytes = bits >> 3;
    //calloc for type casting(BIGNUM to U8)
    U8 *r = (U8*)calloc(bytes,sizeof(U8));
    U8 *a = (U8*)calloc(bytes,sizeof(U8));
    U8 *b = (U8*)calloc(bytes,sizeof(U8));
    //BN_num_bytes(a) : return a's bytes
    int byte_a = BN_num_bytes(b_a);
    int byte_b = BN_num_bytes(b_b);
    //difference between A and B
    int dif = abs(byte_a-byte_b);
    //minimum
    int byte_min = (byte_a < byte_b)? byte_a : byte_b;
    //type casting(BIGNUM to U8)
    BN_bn2bin(b_a,a);
    BN_bn2bin(b_b,b);
    //xor compute
    for(i=1;i<=byte_min;i++)
        r[bytes - i] = a[byte_a - i] ^ b[byte_b - i];
    for(i=1;i<=dif;i++)
        r[bytes - byte_min - i] = (byte_a>byte_b)? a[dif-i] : b[dif-i];
    //type casting(U8 to BIGNUM)
    BN_bin2bn(r,bytes,b_r);
    //Free memory
    free(a);
    free(b);
    free(r);
    return 1;//correct
}
```

AES : KEYGEN / ENC / DEC 예제

```
1 #include <stdio.h>
2 #include <openssl/bn.h>
3 #include <openssl/aes.h>
4
5 int main(int argc, char* argv[]) {
6     BIGNUM *key = BN_new();
7     unsigned char user_key[16];
8     int size = 128; // bit length
9
10    AES_KEY enc_key; // AES encryption key
11    AES_KEY dec_key; // AES decryption key
12
13    BN_rand(key, size, BN RAND TOP ANY, BN RAND BOTTOM ANY); // get random BN key
14    BN_bn2bin(key, user_key); // convert BN key to binary form
15
16    AES_set_encrypt_key(user_key, size, &enc_key);
17    AES_set_decrypt_key(user_key, size, &dec_key);
18
19    unsigned char m[16] = "hello"; // key size = message size
20    unsigned char enc[16];
21    unsigned char dec[16];
22
23    AES_encrypt(m, enc, &enc_key);
24    printf("enc: %s\n", enc);
25    AES_decrypt(enc, dec, &dec_key);
26    printf("dec: %s\n", dec);
27
28    BN_free(key);
29
30    return 0;
31 }
```

$F_k(r)$

$F_k^{-1}(r)$

AES는 대칭키이기 때문에
encrypt key와 decrypt key가 같음

enc:
dec: hello

출력 결과
(enc는 실행시마다 바뀜)