REPORT

메시지 인증

과목명 :	컴퓨터보안
담당교수 :	정준호 교수님
제출일 :	2021 년 05 월 20 일
공과 대학	컴퓨터공학 과
학번 : 2016112154	이름: 정동구

1. 코드

```
std::string key()
{
    char input[] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7',
'8', '9', '!',
'@', '#', '$', ' % ', ' ^ ', ' & ', ' * ', '(', ')', ' - ', '_', ' = ', ' + ', '[', ']', '{',
'}', ';',':',',','<','.','>','/','?' };
    std::string output = "";
    for (int i = 0; i < 10; i++)
    {
        output += input[rand() % 62];
    }

    return output;
        }
```

랜덤한 키를 생성하는 함수이다.

```
std::string timespan()
{
    time_t start, end,timer;
    struct tm user_stime;
    struct tm* curr_stime;
    int tm_day, tm_hour, tm_min, tm_sec;
    double diff;

    timer = time(NULL);
    curr_stime = localtime(&timer);
    int my_year=curr_stime->tm_year, my_month=curr_stime->tm_mon, my_day=curr_stime->tm_mday;

    user_stime.tm_year = my_year - 1900; // 년도가 1900년부터 시작하기 때문
    user_stime.tm_mon = my_month - 1; //월이 0부터 시작하기 때문
    user_stime.tm_mday = my_day;
    user_stime.tm_hour = 0;
    user_stime.tm_min = 0;
    user_stime.tm_sec = 0;
    user_stime.tm_isdst = 0; //썸머타임 사용안함

    start = mktime(&user_stime);
    time(&end);

    diff = difftime(end, start);

    return std::to_string(diff);
}
```

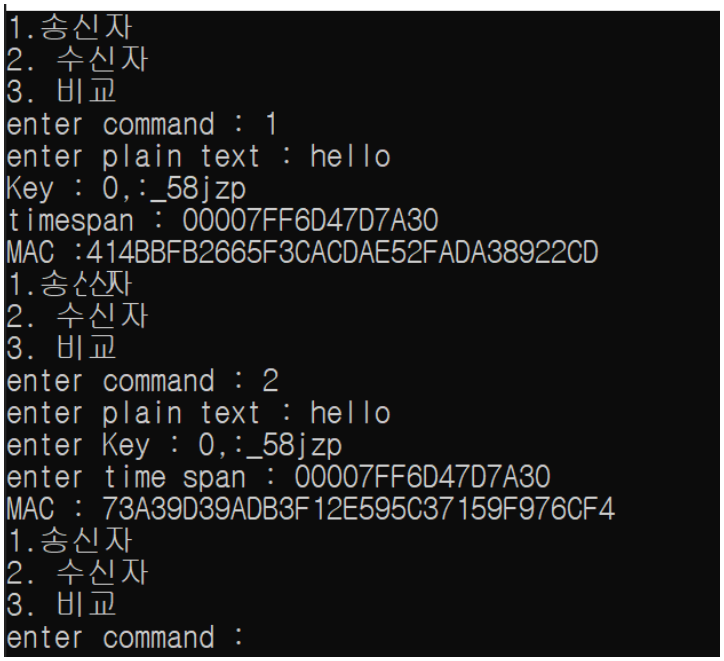
1900 년부터 현재 시각 까지의 시간차이를 초 단위로 나타내는 함수이다. DMAC 을 만들기 위해 사용된다.

```
std::string DMAC(std::string plain_text,std::string key, std::string timespan)
{
    std::string usr_str = key;
    std::string time_span = timespan;
    std::string input = plain_text;
    input += usr_str;
    input += time_span;
    CTR ctr(input);
    ctr.encrypt();
    std::string output = ctr.output();
    return output;
}
```

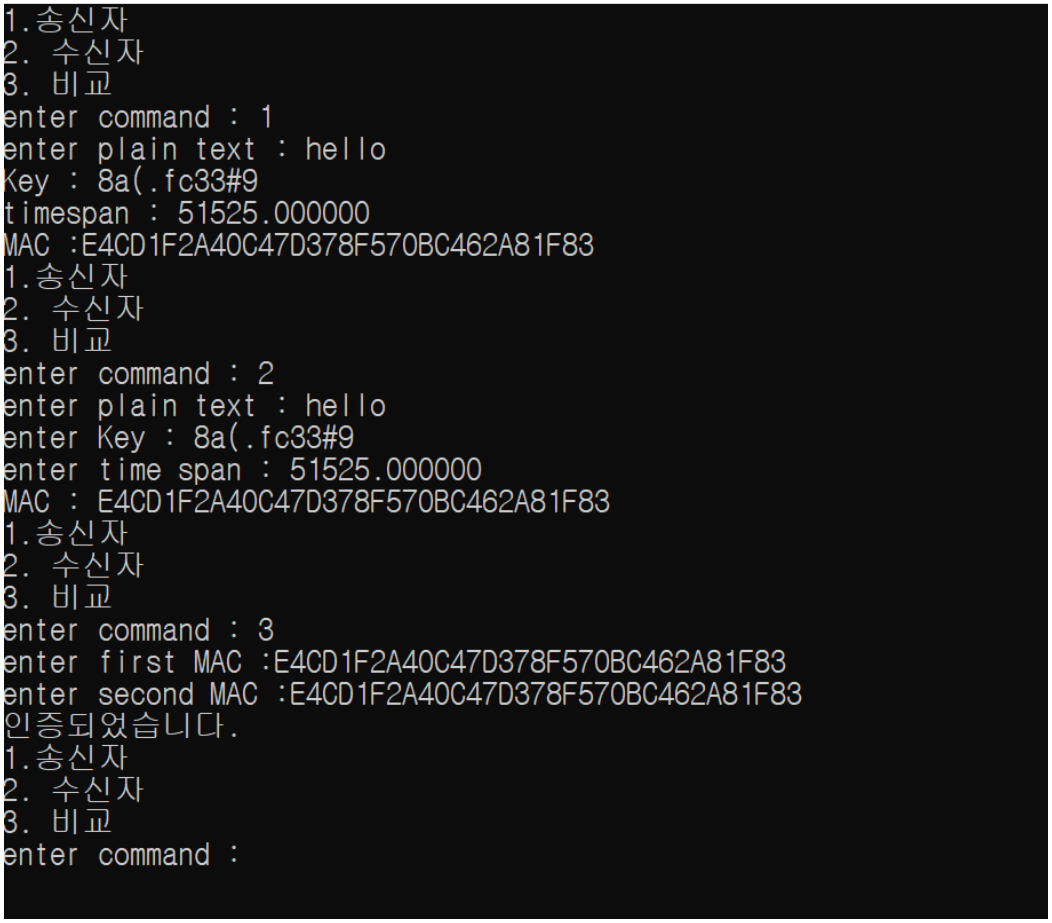
MAC 을 생성하는 함수로 평문+key+timespan 을 한 문자열을 가지고 DES 를 이용한 CTR 블록 암호로 암호화 한다.

전체 코드 : security/main.cpp at master · dsaf2007/security (github.com)

2. 결과



입력받은 평문을 바탕으로 key 와 timespan 을 조합하여 메시지 인증코드인 MAC 를 생성하는 것 까지는 정상적으로 잘 되었다. 이후 수신자가 해당 값을 통해서 동일한 MAC 을 생성이 되어야 정상이나 MAC 이 다르게 나왔다. 이 부분은 최초에 CTR 설계시에 nonce 가 CTR 클래스를 생성할 때 마다 랜덤하게 나왔던 점과 길이가 부족한 부분을 padding 할때에도 랜덤 문자열이 들어갔기 때문이다.



때문에 nonce 를 일정하게 하고, padding 이 되는 것도 “0”으로 일정하게 변경하고 난 후에는 평문+key+timespan 으로 이루어진 문자열을 CTR 을 통해 암호화 하였을 때 인증코드가 정확하게 일치되게 나옴을 확인할 수 있었다.

3. 공격방식과 방어

메시지 인증코드에 대한 공격은 크게 재전송 공격과 키 추측이 있다. 재전송 공격을 방어하는 방법에는 sequence number 을 부여하거나 현재 시간을 메시지에 넣는 timestamp 방법, 그리고 비표를 송신자에게 수신자가 전달하는 방법이 있다. 해당 과제에서는 timestamp 를 이용하여 재전송 공격을 방지하였다.

키 추측 공격의 경우 사람이 키를 입력할 경우에 추측이 쉬우므로 랜덤한 키값을 부여하는 것으로 어느정도 해결하였다. 구현하지 못한 HMAC 에서는 일방향 해시함수의 일방향성 및 충돌 내성을 통하여 MAC 값으로부터 키를 추측할 수 없도록 할 수 있다. 하지만 DES 를 이용하는 경우 전사공격 등을 통해 추측이 가능 할 수 있으므로 재전송 공격 방지와 비슷한 방식으로 비표를 수신자가 송신자에게 전송하고 메시지에 포함시키는 것으로 방지 할 수 있을 것이라 생각된다.

4. 소감

이전에 해시함수를 제대로 구현하지 못했기 때문에 DES 를 활용한 방법밖에 구현하지 못했다. 구현 못한 부분보다 아쉬운 것은 제출시각을 착각하여 이미 함수를 구현하고 보고서만 일부 마무리 했으면 됨에도 불구하고 제대로 제출을 하지 못한 부분이다. 꼼꼼하고 확실하게 확인하는 습관을 가져야겠다.