

2014년 정보보호 졸업작품 보고서

파일 유출 방지 시스템

지도 교수님 : 양환석 교수님

팀명 : LKC Security

팀원 : 이호영, 김상우, 최유탉

2014. 5

중부대학교 정보보호학과

목 차

1. 프로젝트 설명	1
2. 연구 계획	1
2.1 연구 목적	1
2.2 연구 방법	2
2.3 연구 개요	2
2.4 연구 구성도	3
3. 연구 프로그램 구축	5
3.1 주요 소스	5
3.2 실행 결과	8
4. 연구 결과	12
4.1 전체 소스	12
5. 결 론	29
6. 참고 문헌	29
7. Program PPT	30

1. 프로젝트 설명

Program은 API로 구현한다. Program은 사용자 Login 시 Log가 Server에 자동 저장이 되고, 파일 암호화, 복호화 할 수 있는 프로그램이 실행된다. 프로그램은 일반 사용자와 관리자로 나뉘는데 일반 사용자는 자신을 인증 할 수 있는 아이디와 비밀번호를 만들어 암호화 복호화를 할 수 있다. 이와 반면 관리자는 프로그램에서 암호화 또는 복호화 된 문서에 대한 권한을 가질 수 없고, 관리자는 사용자가 Login 하였을 때의 IP와 시간이 저장되는 Log 기록 관리와 사용자 아이디와 패스워드를 생성하여 주고, 암호화 · 복호화 기록을 관리한다. 일반사용자는 자신이 암호화, 또는 복호화 할 텍스트 문서를 불러와 일을 처리하고 자신이 편집한 문서를 저장한다. 이 때 중앙의 서버로 파일을 암호화 하였을 때, 파일을 복호화 하였을 때 사용 기록을 저장하여 혹여 중요한 파일이 유출이 되었을 때 어떤 사용자가 사용 중일 때 파일이 유출 되었는지 파악하는데 시간을 줄일 수 있습니다.

2. 연구 계획

2.1 연구 목적

최근 들어, 정보통신의 비약적 발달로 비밀 수준의 정보들이 컴퓨터에 저장되어 네트워크 혹은 저장매체를 통하여 불법적으로 정보를 탈취, 변조 또는 소유자의 실수에 의하여 비밀 정보가 누출되는 등의 보안문제가 심각해지고 있다. 따라서 컴퓨터 보안을 위하여 운영체제 상에서의 보안을 강화 할 시 제작 비용과, 새로운 파일 암호 시스템을 개발해야 하는 복잡한 단계를 거쳐야 한다. 이러한 점을 착안하여 보안성이 강화된 암호화 운영체제를 개발하지 않고도 컴퓨터 시스템에서 사용자가 비밀 정보를 사용 할 때 프로그램을 사용하여 쉽게 암호화 하여 비밀 정보가 누출이 되더라도 유출된 정보를 해독하는데 많은 시간과 어려움을 갖게 하여 정보를 보호하는데 의의를 두었습니다. 또한 관리자를 두어 일반 사용자가 Log 기록에 쉽게 접근할 수 없도록 하고, 일반 사용자의 아이디와 패스워드를 생성, 관리 한다.

2.2 연구 방법

암호 기법을 사용하여 사용자가 자신이 작성한 파일이 유출되더라도 복호화를 하기 어렵게 한다. 그리고 사용자의 정보와 파일의 암호화 된 정보를 서버에 기록함으로써 유출에 대한 추적 및 관리를 강화한다.

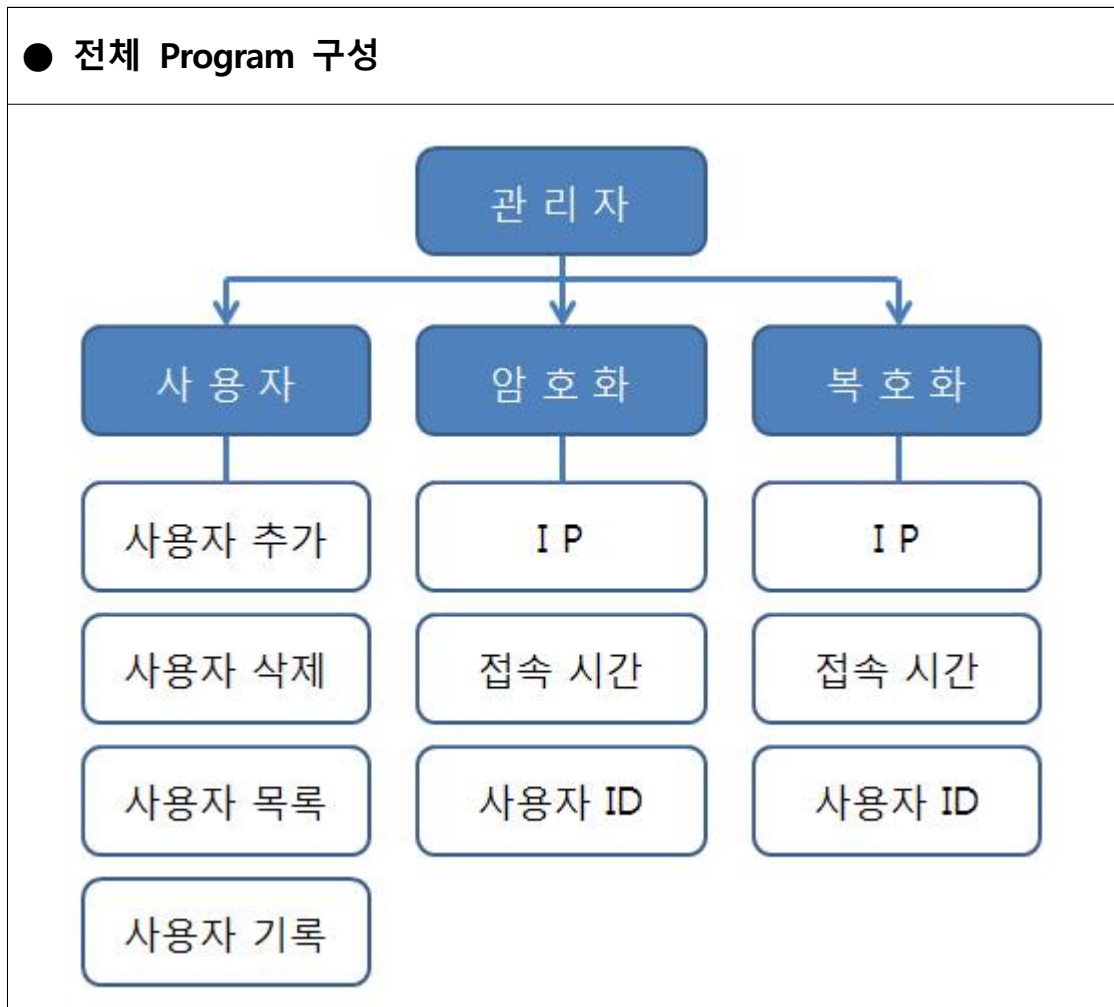
■ API : 프로그램 또는 애플리케이션이 운영 체제에 어떤 처리를 위해서 호출할 수 있는 서브루틴 또는 함수의 집합이다. 윈도우 API의 경우 C, C++, 파스칼 등과 같은 언어에서 윈도우를 만들고, 파일을 여는 것과 같은 처리를 할 수 있도록 1,000여개 이상의 함수로 구성되어 있다. 명령어의 집합으로 애플리케이션 프로그램에서 오퍼레이팅 시스템의 기본적인 기능을 사용할 필요가 있을 때에 여기에서 명령어를 호출한다. 이 명령어는 MS-DOS에서 말하면 시스템 콜에 해당한다. 이 API를 확실히 정하고 범용성을 갖게 하면 주변 기기의 인터페이스를 용이하게 사용할 수 있으며, 애플리케이션 프로그램의 개발이나 호환성의 면에서 개선된다.

■ MYSQL : 표준 데이터베이스 질의 언어인 SQL(Structured Query Language)을 사용하는 개방 소스의 관계형 데이터베이스 관리 시스템(RDBMS). 매우 빠르고, 유연하며, 사용하기 쉬운 특징이 있다. 다중 사용자, 다중 스레드를 지원하고, C, C++, Eiffel, 자바, 펄, PHP, Python 스크립트 등을 위한 응용 프로그램 인터페이스(API)를 제공한다. 유닉스나 리눅스, 윈도 운영 체제 등에서 사용할 수 있다. 상호 연동이 잘되면서도 오픈 소스로 개발되는 무료 프로그램이어서 홈 페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다.

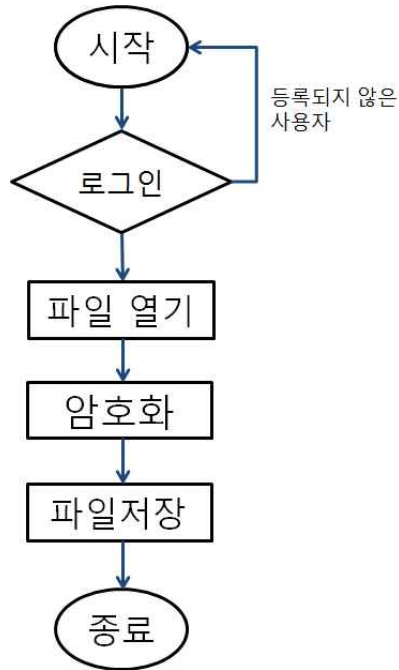
2.3 연구 개요

이번 프로젝트의 목적은 Application Program Interface(API)의 동작 원리를 이해하고, API를 구현하는 것이다. 사용자와 Log 기록을 저장하는 Server를 MYSQL을 사용하여 API와 MYSQL 사이에 연동하는 관계를 배우고 익히는데 목적을 두었다. 또한 Compiler는 Microsoft Visual Studio 2010 프로그램을 사용하였다. 프로그램은 일반 사용자와 관리자 Mod로 나뉘어 실행되고, 사용자가 프로그램에 로그인 또는 프로그램 실행 시 Server에 저장하기로 한다.

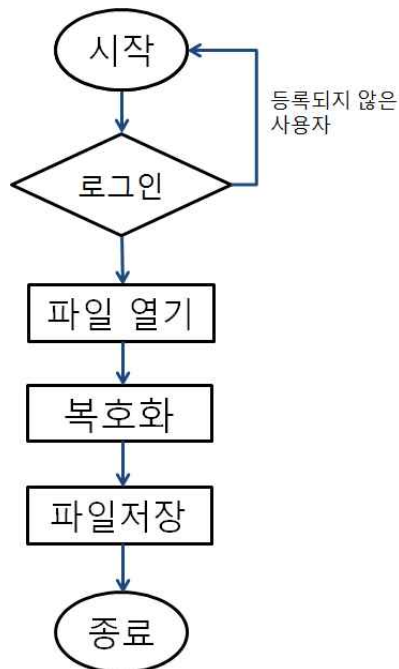
2.4 연구 구성도



● File 암호화 Program 설계



● File 복호화 Program 설계



3. 연구 프로그램 구축

3.1 주요 소스

● 로그인 부분

```
//로그인 함수
BOOL CALLBACK login(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    int i=0,num;
    char qstr[500]="";
    switch(iMessage) {
    case WM_COMMAND:
        switch(wParam) {
        case IDOK:
            query_stat = mysql_query(conn, "select *from login");
            RES = mysql_store_result(conn);
            num = mysql_num_fields(RES);
            GetDlgItemText(hDlg,IDC_ID,row_id,sizeof(row_id));
            GetDlgItemText(hDlg,IDC_PW,row_pw,sizeof(row_pw));
            while(ROW = mysql_fetch_row(RES)) {
                if((strcmp(ROW[0],row_id)
                strcmp(ROW[1],row_pw))==0) {
                    MessageBox(hDlg,"로그인 성공","로그인 성공",MB_OK);
                    i=1;
                }
            }
            if(i==0) {
                MessageBox(hDlg,"로그인 실패","로그인 실패",MB_OK);
                return 2;
            }
            EndDialog(hDlg,IDOK);
            break;
        case IDCANCEL:
            EndDialog(hDlg,IDCANCEL);
            exit(1);
        }
        break;
    }
    return 0;
}
```

● 암호화 부분

```
//암호화 버튼

case ID_BUT1:
    bSave = TRUE;
    if(bStart == TRUE) {
        for(i=0;i<lstrlen(Encry);i++) {
            if(Encry[i] + EnterKey + i == NULL)
                Encry[i] = Encry[i] - EnterKey;
            else Encry[i] = Encry[i] + EnterKey + i;
            if(Encry[i] == NULL)

//그래도 NULL문자이면

                Encry[i] += count;
            //count(사용자 정의)로 더해버림(250미만의 수)
        }

        //for문.
        SetWindowText(c4,Encry);

        MessageBox(hWnd,"정상적으로 암호화 되었습니다. Save
할 파일명을 입력하세요.,"알림",MB_OK);
        if(bSave == TRUE) {
            DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG1),hWnd,File_Name);

hFile1=CreateFile(SaveFile,GENERIC_WRITE,0,NULL,CREATE_ALWAYS,FILE_ATTRIBUTE_
NORMAL,NULL);

WriteFile(hFile1,Encry,lstrlen(Encry),&dwWrite,NULL);

            CloseHandle(hFile1);
            strcpy(enc,"insert into enc values (");
            strcat(enc,ip);
            strcat(enc,",");
            strcat(enc,time);
            strcat(enc,",");
            strcat(enc,SaveFile);
            strcat(enc,")");
            mysql_query(conn,enc);
        } else
            MessageBox(hWnd,"암호문이 없습니다.,"오류",MB_OK);
    }
    else
        MessageBox(hWnd,"암호화할 파일을 열어주세요.,"오류",MB_OK);
    bStart = FALSE;
    break;
```


● 복호화 부분

```
//복호화 버튼
        case ID_BUT2:
            GetWindowText(c4,Arr,30000);
//암호화 텍스트 얻어옴
            for(i=lstlen(Encry) -1;i>-1;i--) {
//복호화 작업시작
                if((2*(EnterKey)) + i + Encry[i] == NULL)
                    Arr[i] += EnterKey;
                else Arr[i] = Encry[i]-i-EnterKey; //문자열 출력
                if(Encry[i]-count==NULL && Arr[i] -i -(2*EnterKey)
                    == NULL) {
                    Arr[i] -= count;
//Encry에서 count을 빼면 NULL이 되는 문자를
                }
            }
            SetWindowText(c3,Arr);
            MessageBox(hWnd,"정상적으로 복호화 되었습니다. Save할 파일명
            을 입력하세요.", "알림",MB_OK);
            if(bSave == TRUE) {

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG1),hWnd,File_Name);

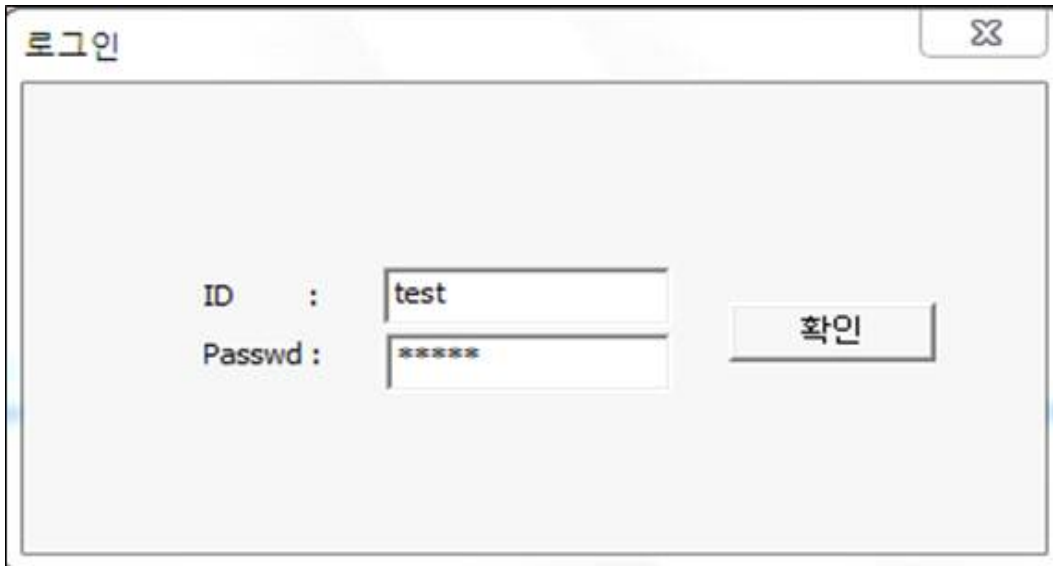
hFile2=CreateFile(SaveFile,GENERIC_WRITE,0,NULL,CREATE_ALWAYS,FILE_ATTRIBUTE_
NORMAL,NULL);

                WriteFile(hFile2,Arr,lstrlen(Arr),&dwWrite,NULL);
                CloseHandle(hFile2);
                strcpy(enc,"insert into ddec values (");
                strcat(enc,ip); strcat(enc,",");
                strcat(enc,time);
                strcat(enc,",");
                strcat(enc,SaveFile);
                strcat(enc,"");
                mysql_query(conn,enc);
            }
            else MessageBox(hWnd,"복호화할 파일을 열어주세요.", "오류
            ",MB_OK);
            bStart = FALSE;
            break;
        }
return 0;
```

3.2 실행 결과

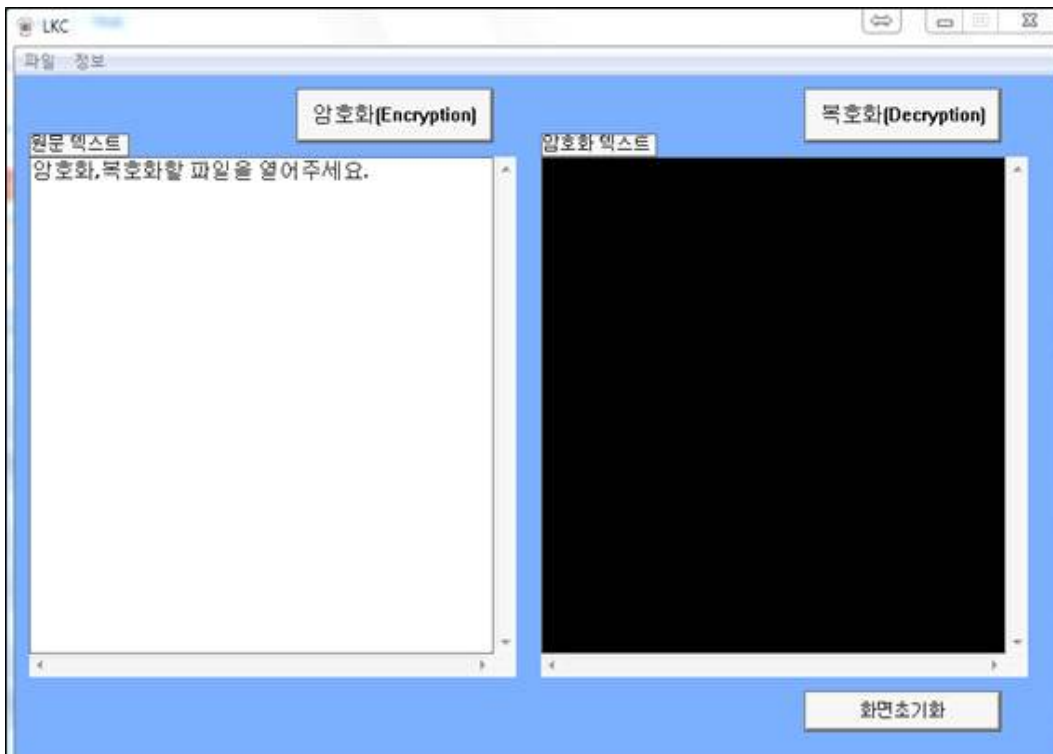
■ 로그인 화면

- 사용자는 자신의 ID와 패스워드로 접속이 가능하다.



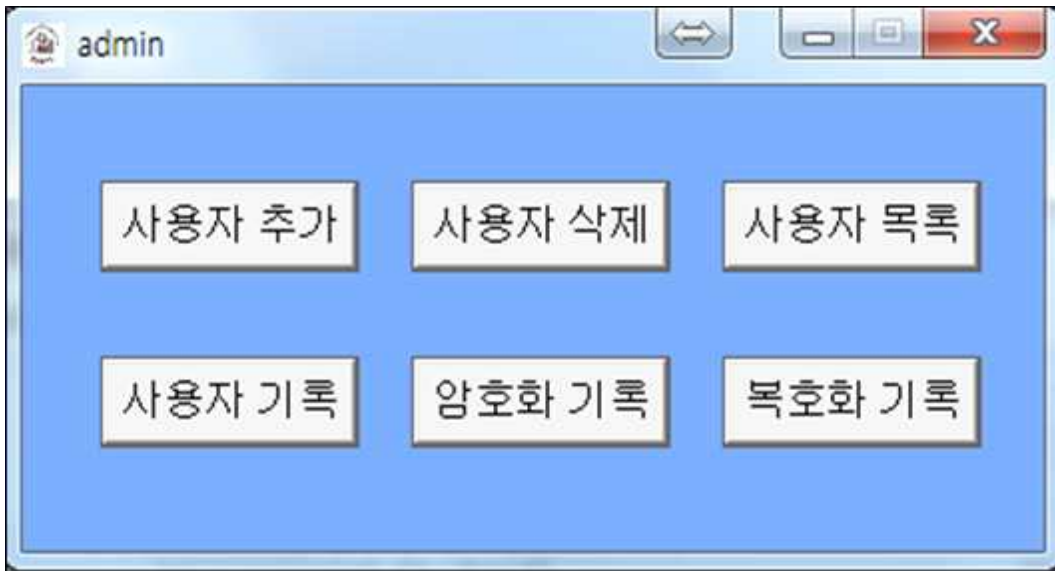
■ 암호복호화 화면

- 사용자는 파일을 선택하여 암호복호화를 진행할 수 있다.



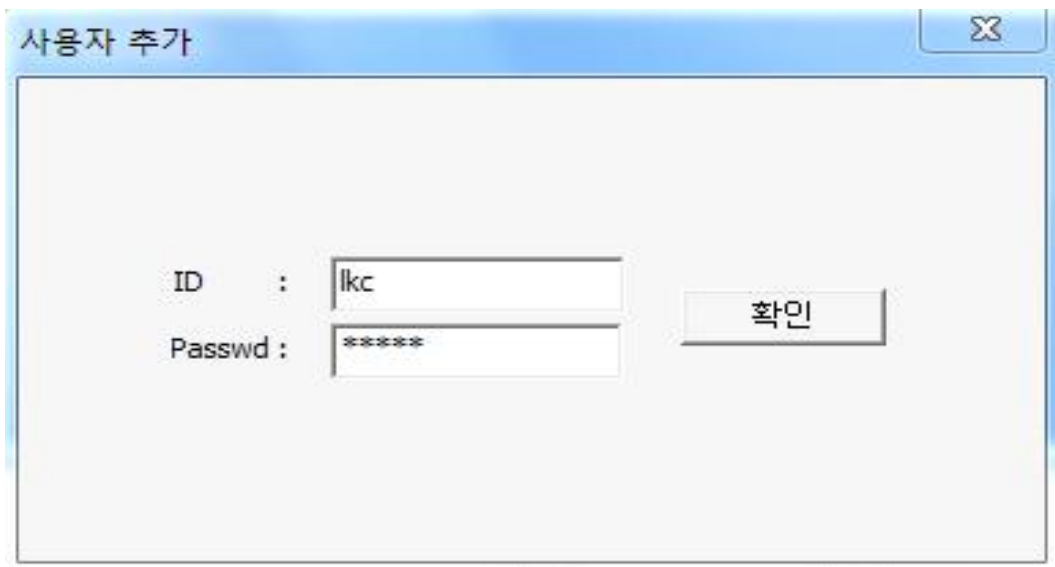
■ 관리자 화면

- 관리자는 사용자 관리 및 로그 기록을 볼 수 있다.



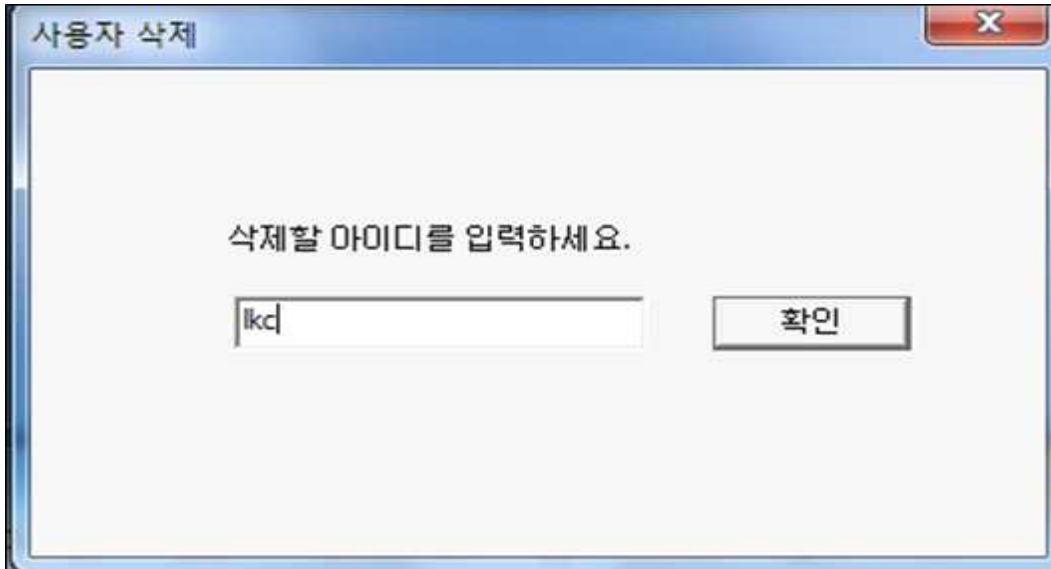
■ 사용자 추가 화면

- 관리자가 일반 사용자를 추가 할 수 있다.



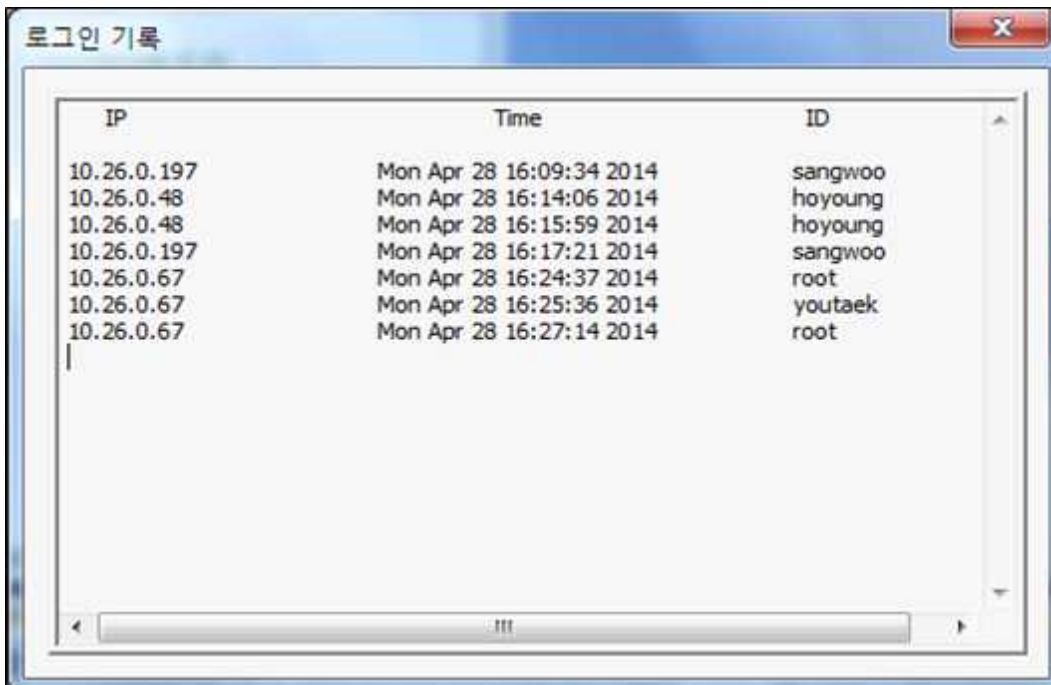
■ 사용자 삭제 화면

- 사용하지 않는 사용자를 삭제 할 수 있다.



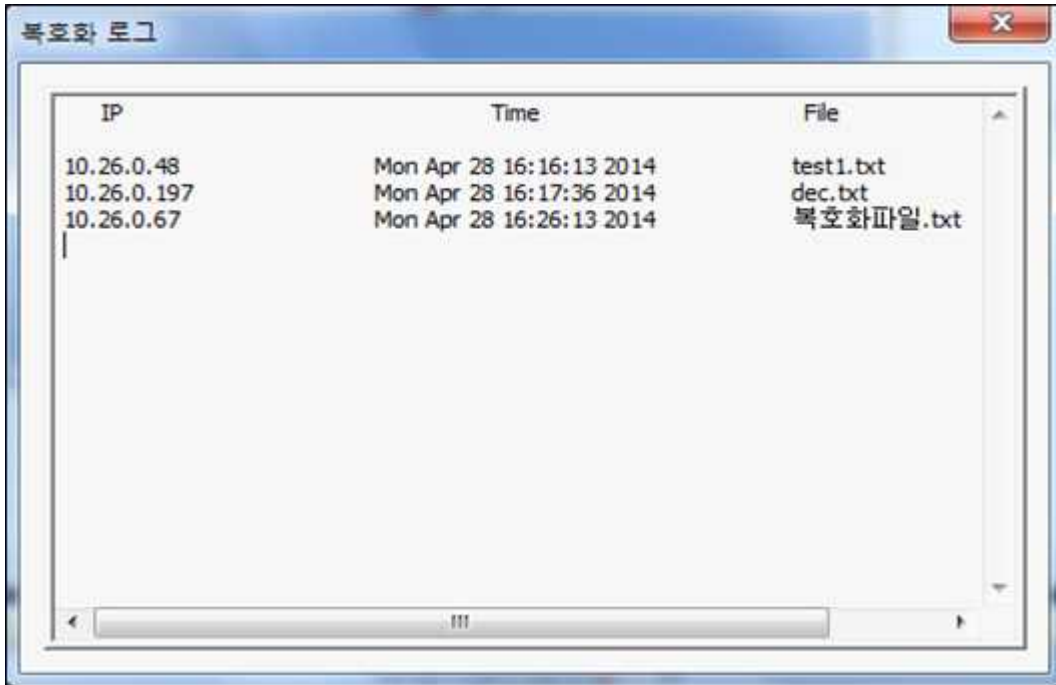
■ 로그인 기록 화면

- 모든 사용자의 로그인한 IP와 시간, ID를 확인이 가능하다.



■ 복호화 로그 화면

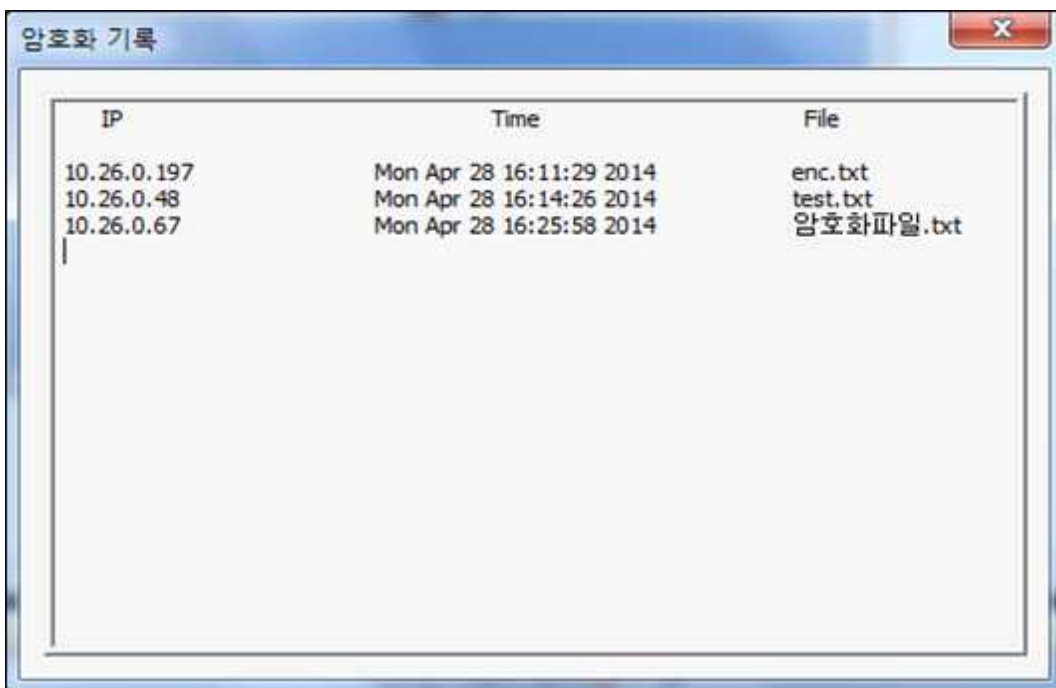
- 복호화를 한 사용자의 IP와 시간, 저장한 파일 이름을 확인이 가능하다.



IP	Time	File
10.26.0.48	Mon Apr 28 16:16:13 2014	test1.txt
10.26.0.197	Mon Apr 28 16:17:36 2014	dec.txt
10.26.0.67	Mon Apr 28 16:26:13 2014	복호화파일.txt

■ 암호화 로그 화면

- 암호화를 한 사용자의 IP와 시간, 저장한 파일 이름을 확인이 가능하다.



IP	Time	File
10.26.0.197	Mon Apr 28 16:11:29 2014	enc.txt
10.26.0.48	Mon Apr 28 16:14:26 2014	test.txt
10.26.0.67	Mon Apr 28 16:25:58 2014	암호화파일.txt

4. 연구 결과

4.1 전체 소스

Password code.cpp

```
#include <winsock2.h>
#pragma comment(lib, "ws2_32.lib")
#include <windows.h>
#include <mysql.h>
#include "resource.h"
#include <time.h>

#define MYSQL_HOST "10.26.0.48"
#define MYSQL_ID "root"
#define MYSQL_PW "12345"
#define MYSQL_DB "lkc"

LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);
HINSTANCE g_hInst;
HWND hWndMain;
HWND c1,c2,c3,c4,c5,c6,c7,c12;
LPCTSTR lpszClass=TEXT("LKC");
HBRUSH hBrushEdit;
HFONT hFontEdit,hFontEditT;
enum {ID_BUT1 = 20, ID_BUT2,ID_BUT3};

//암&복호화
static int EnterKey=1;
static int count=1;
TCHAR str[300]="암호화,복호화할 파일을 열어주세요.";
TCHAR Encry[30000]; //암호화 텍스트
TCHAR Arr[30000]=""; //복호화 텍스트
TCHAR SaveFile[200]=""; //저장 파일이름
BOOL bStart = FALSE;
BOOL bSave = FALSE;

//서버
MYSQL mysql,*conn;
MYSQL_RES* RES;
```

```

MYSQL_ROW ROW;
int    query_stat;
char  row_id[40]={NULL};
char  row_pw[40]={NULL};
char  *ip="";

//윈도우 프로그램의 시작점(윈도우 메인)
int  APIENTRY  WinMain(HINSTANCE  hInstance,HINSTANCE  hPrevInstance,LPSTR
lpzCmdParam, int  nCmdShow)
{
    HWND  hWnd;
    MSG  Message;
    WNDCLASS  WndClass;
    g_hInst=hInstance;

    WndClass.cbClsExtra=0;
    WndClass.cbWndExtra=0;
    WndClass.hbrBackground=CreateSolidBrush(RGB(121,170,255));
    WndClass.hCursor=LoadCursor(hInstance, IDC_ARROW);
    WndClass.hIcon=LoadIcon(hInstance,MAKEINTRESOURCE(IDI_ICON1));
    WndClass.hInstance=hInstance;
    WndClass.lpszClassName=lpzClass;
    WndClass.lpszMenuName=MAKEINTRESOURCE(IDR_MENU1);
    WndClass.style=CS_HREDRAW | CS_VREDRAW;
    RegisterClass(&WndClass);

//등록된 윈도우 클래스로 윈도우를 생성한다.
    hWnd = CreateWindow(lpzClass,lpzClass,WS_SYSMENU
|WS_MINIMIZEBOX,CW_USEDEFAULT,CW_USEDEFAULT,800,550,
NULL,(HMENU)NULL,hInstance,NULL);
    ShowWindow(hWnd,nCmdShow);

//메세지 루프
    while(GetMessage(&Message,NULL,0,0)) {
        TranslateMessage(&Message);
        DispatchMessage(&Message);
    }
    return  (int)Message.wParam;
}

//사용방법 부분
BOOL CALLBACK ProgramInfo(HWND  hDlg,UINT  iMessage,WPARAM  wParam,LPARAM  lParam)
{
    switch(iMessage) {

```

```

    case WM_INITDIALOG:
        return 0;
    case WM_COMMAND:
        switch(LOWORD(wParam)) {
            case IDOK:
                EndDialog(hDlg,IDOK);
                return 0;
            case IDCANCEL:
                EndDialog(hDlg,IDCANCEL);
                return 0;
        }
        break;
    }
    return 0;
}

//만든이 부분
BOOL CALLBACK Maker(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    switch(iMessage) {
        case WM_INITDIALOG:
            return 0;
        case WM_COMMAND:
            switch(LOWORD(wParam)) {
                case IDOK:
                    EndDialog(hDlg,IDOK);
                    return 0;
                case IDCANCEL:
                    EndDialog(hDlg,IDCANCEL);
                    return 0;
            }
            break;
        }
        return 0;
    }
}

//IP 부분
char *GetIPGET()
{
    WSADATA wdata;
    WSASStartup(MAKEWORD(2,2), &wdata);
    char hostname[255],ipaddr[50];
    struct in_addr addr;
    PHOSTENT h;
}

```



```

        memset(hostname,0,sizeof(hostname));
        memset(ipaddr,0,sizeof(ipaddr));
        h = gethostbyname(hostname);
        //memcpy(&addr, h->h_addr_list[0], sizeof(struct in_addr));
        memcpy(&addr, h -> h_addr, sizeof(struct in_addr));
        return inet_ntoa(addr);
    }

//파일이름 설정 부분
BOOL CALLBACK File_Name(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    switch(iMessage) {
    case WM_INITDIALOG:
        return 0;
    case WM_COMMAND:
        switch(wParam) {
        case IDOK:
            GetDlgItemText(hDlg, IDC_EDIT1, SaveFile, sizeof(SaveFile));
            bSave = FALSE;
            EndDialog(hDlg, IDOK);
            return 0;
        case IDCANCEL:
            EndDialog(hDlg, IDCANCEL);
            bSave = TRUE;
            return 0;
        }
        break;
    }
    return 0;
}

//로그인함수
BOOL CALLBACK login(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    int i=0,num;
    char qstr[500]="";
    switch(iMessage) {
    case WM_COMMAND:
        switch(wParam) {
        case IDOK:
            query_stat = mysql_query(conn, "select *from login");
            RES = mysql_store_result(conn);
            num = mysql_num_fields(RES);
            GetDlgItemText(hDlg, IDC_ID, row_id, sizeof(row_id));
            GetDlgItemText(hDlg, IDC_PW, row_pw, sizeof(row_pw));

```

```

        while(ROW = mysql_fetch_row(RES)) {
            if((strcmp(ROW[0],row_id)
||
strcmp(ROW[1],row_pw))==0) {
                MessageBox(hDlg,"로그인 성공","로그인 성공",MB_OK);
                i=1;
            }
            if(i==0) {
                MessageBox(hDlg,"로그인 실패","로그인 실패",MB_OK);
                return 2;
            }
            EndDialog(hDlg,IDOK);
            break;
        case IDCANCEL:
            EndDialog(hDlg,IDCANCEL);
            exit(1);
        }
        break;
    }
    return 0;
}

```

```

LRESULT CALLBACK WndProc(HWND hWnd, UINT iMessage, WPARAM wParam, LPARAM
IParam)
{
    //HDC hdc;
    //PAINTSTRUCT ps;
    OPENFILENAME OFN;
    TCHAR lpstrFile[MAX_PATH]="";
    TCHAR *staticWord1="원문 텍스트";
    TCHAR *staticWord2="암호화 텍스트";
    HWND hMDlg1 = NULL, hMDlg2 = NULL;
    DWORD dwRead,dwWrite;
    HANDLE hFile1 = NULL;
    HANDLE hFile2 = NULL;
    TCHAR buf[30000] = "";
    int i=0;
    char enc[200]="", dec[200]="", log[200]="";

    time_t etime;    time(&etime);
    char *time;
    time=ctime(&etime);
    static HWND help;

```

```

switch(iMessage) {
case WM_CREATE:
    ip=GetIPGET();
    //MessageBox(hWnd,ip,"알림",MB_OK);
    mysql_init(&mysql);
    conn = mysql_real_connect(&mysql, MYSQL_HOST, MYSQL_ID,
MYSQL_PW, MYSQL_DB, 0, 0, 0);
    hWndMain = hWnd;
    ShowWindow(c12,SW_SHOW);

start :

    i = DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG3),hWnd,login);
    if(i==2) goto start;
    else {
        strcpy(log,"insert into login_log values (");
        strcat(log,ip);
        strcat(log,",");
        strcat(log,time);
        strcat(log,",");
        strcat(log,row_id);
        strcat(log,")");
        mysql_query(conn,log);
        if((strcmp(row_id,"root") || strcmp(row_pw,"12345"))==0) {
            ::ShellExecute(::GetDesktopWindow(), "open", "admin.exe" , 0, 0,
SW_SHOWNORMAL); //exe파일 실행
            exit(1);
        }
    }

//버튼

    c1 = CreateWindow(TEXT("button"),TEXT("암호화(Encryption)"),
        WS_CHILD | WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
        214,10,150,40,hWnd,(HMENU)ID_BUT1,g_hInst,NULL);
    c2 = CreateWindow(TEXT("button"),TEXT("복호화(Decryption)"),
        WS_CHILD | WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
        600,10,150,40,hWnd,(HMENU)ID_BUT2,g_hInst,NULL);

//에디터 창

    c3 = CreateWindow(TEXT("edit"),str,
        WS_CHILD | ES_MULTILINE | WS_VISIBLE | WS_VSCROLL |
        WS_HSCROLL| WS_BORDER,11,60,370,380,hWnd,(HMENU)0,g_hInst,NULL);
    c4 = CreateWindow(TEXT("edit"),NULL,
        WS_CHILD | ES_MULTILINE | WS_VISIBLE | WS_HSCROLL |
        WS_VSCROLL | WS_BORDER,400,60,370,380,hWnd,(HMENU)3,g_hInst,NULL);

//텍스트 이름

    c5 = CreateWindow(TEXT("static"),staticWord1,WS_CHILD| WS_BORDER
| WS_VISIBLE,11,43,76,17,hWnd,(HMENU)-1,g_hInst,NULL);
    c6 = CreateWindow(TEXT("static"),staticWord2,WS_CHILD| WS_BORDER

```

```

        | WS_VISIBLE,400,43,88,17,hWnd,(HMENU)-1,g_hInst,NULL);
        c7 = CreateWindow(TEXT("button"),TEXT("화면초기화"),
            WS_CHILD | WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
            600,450,150,30,hWnd,(HMENU)ID_BUT3,g_hInst,NULL);

hBrushEdit = CreateSolidBrush(RGB(0,0,0));
hFontEdit = CreateFont(13,0,0,0,0,0,0,HANGEUL_CHARSET,3,2,1,VARIABLE_PITCH,TEXT("평
문"));
hFontEditT = CreateFont(10,0,0,0,0,0,0,HANGEUL_CHARSET,3,2,1,VARIABLE_PITCH,TEXT("
궁서"));
SendMessage(c5,WM_SETFONT,(WPARAM)hFontEdit,MAKELPARAM(FALSE,0));
SendMessage(c6,WM_SETFONT,(WPARAM)hFontEdit,MAKELPARAM(FALSE,0));
SendMessage(c7,WM_SETFONT,(WPARAM)hFontEdit,MAKELPARAM(FALSE,0));
SetFocus(c3);
return 0;

    case WM_COMMAND:
        switch(LOWORD(wParam)) {
//암호화할 파일 열기
        case ID_MENU_1:
            SetWindowText (GetDlgItem (hWnd, 0),""); //화면 초기화
            SetWindowText (GetDlgItem (hWnd, 3),""); //화면 초기화
            memset(&OFN,0,sizeof(OPENFILENAME));
            OFN.IStructSize = sizeof(OPENFILENAME);
            OFN.lpszTitle = "암호화 할 파일 열기";
            OFN.hwndOwner = hWnd;
            OFN.lpszFilter = TEXT("텍스트문서(*.txt)W0*.txt");
            OFN.lpszFile = lpszFile;
            OFN.nMaxFile =MAX_PATH;
            GetOpenFileName(&OFN);
            hFile1 = CreateFile(OFN.lpszFile,GENERIC_READ |
GENERIC_WRITE,0,NULL,OPEN_EXISTING,
            FILE_ATTRIBUTE_NORMAL,NULL);

            if(hFile1 != INVALID_HANDLE_VALUE) {
                ReadFile(hFile1,buf,sizeof(buf),&dwRead,NULL);
                CloseHandle(hFile1);
                SetWindowText(c3,buf);
                bStart = TRUE;
                GetWindowText(c3,Encry,30000);
            }
            break;
        case ID_MENU_ProgramInfo:
            MessageBox(hWnd,"ASCII 코드를 이용한 치환 암호 기법을 사용한
프로그램입니다.Wn***암호화 방법***Wn1.원문 TEXT파일을 불러옵니다.Wn2.암호화 버튼을 누릅
니다.Wn***복호화 방법***Wn1.암호화 TEXT파일을 불러옵니다.Wn2.복호화 버튼을 눌러 원문 텍

```

```

스트로 변환시킵니다. ", "사용방법", MB_OK);
                break;
//복호화할 파일 열기
                case ID_MENU_2:
                    SetWindowText (GetDlgItem (hWnd, 0), ""); //화면 초기화
                    SetWindowText (GetDlgItem (hWnd, 3), ""); //화면 초기화
                    memset(&OFN, 0, sizeof(OPENFILENAME));
                    OFN.lStructSize = sizeof(OPENFILENAME);
                    OFN.lpszTitle = "복호화 할 파일 열기";
                    OFN.hwndOwner = hWnd;
                    OFN.lpszFilter = TEXT("텍스트문서(*.txt)W0*.txt");
                    OFN.lpszFile = lpstrFile;
                    OFN.nMaxFile = MAX_PATH;
                    GetOpenFileName(&OFN);
                    hFile2 = CreateFile(OFN.lpszFile, GENERIC_READ |
GENERIC_WRITE, 0, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
                    if(hFile1 != INVALID_HANDLE_VALUE) {
                        ReadFile(hFile2, buf, sizeof(buf), &dwRead, NULL);

                        CloseHandle(hFile2);
                        SetWindowText(c4, buf);
                        GetWindowText(c4, Encry, 30000);
                        bSave = TRUE;
                    }
                break;
//화면 클리어
                case ID_BUT3:
                    SetWindowText (GetDlgItem (hWnd, 0), "");
                    SetWindowText (GetDlgItem (hWnd, 3), "");
                    break;
//창 종료
                case ID_MENU_EXIT:
                    DestroyWindow(hWnd);
                    break;
//만든이
                case ID_MENU_Maker:
                    if(!IsWindow(hMDlg1)) {
                        hMDlg1 = CreateDialog(g_hInst, MAKEINTRESOURCE(IDD_DIALOG2), hWnd, Maker);
                        ShowWindow(hMDlg1, SW_SHOW);
                    }
                    break;
//암호화 버튼
                case ID_BUT1:
                    bSave = TRUE;

```

```

        if(bStart == TRUE) {
            for(i=0;i<Istrlen(Encry);i++) {
                if(Encry[i] + EnterKey + i == NULL)
                    Encry[i] = Encry[i] - EnterKey;
                else Encry[i] = Encry[i] + EnterKey + i;
                if(Encry[i] == NULL)

//그래도 NULL문자이면

                    Encry[i] += count;
                //count(사용자 정의)로 더해버림(250미만의 수)
            }

            //for문.
            SetWindowText(c4,Encry);

            MessageBox(hWnd,"정상적으로 암호화 되었습니다. Save
할 파일명을 입력하세요.","알림",MB_OK);
            if(bSave == TRUE) {
                DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG1),hWnd,File_Name);

hFile1=CreateFile(SaveFile,GENERIC_WRITE,0,NULL,CREATE_ALWAYS,FILE_ATTRIBUTE_NORMA
L,NULL);

WriteFile(hFile1,Encry,Istrlen(Encry),&dwWrite,NULL);

                CloseHandle(hFile1);
                strcpy(enc,"insert into enc values (");
                strcat(enc,ip);
                strcat(enc,"");
                strcat(enc,time);
                strcat(enc,"");
                strcat(enc,SaveFile);
                strcat(enc,"");
                mysql_query(conn,enc);
            } else
                MessageBox(hWnd,"암호문이 없습니다.","오류",MB_OK);
        }
        else
            MessageBox(hWnd,"암호화할 파일을 열어주세요.","오류",MB_OK);
        bStart = FALSE;
        break;

//복호화 버튼
        case ID_BUT2:
            GetWindowText(c4,Arr,30000);
//암호화 텍스트 얻어옴
            for(i=Istrlen(Encry) -1;i>-1;i--) {
//복호화 작업시작

```

```

if((2*(EnterKey)) + i + Encry[i] == NULL)
Arr[i] += EnterKey;
else Arr[i] = Encry[i]-i-EnterKey; //문자열 출력
if(Encry[i]-count==NULL && Arr[i] -i -(2*EnterKey)
== NULL) {
Arr[i] -= count;
//Encry에서 count을 빼면 NULL이 되는 문자를
}
}
SetWindowText(c3,Arr);
MessageBox(hWnd,"정상적으로 복호화 되었습니다. Save할 파일명
을 입력하세요.", "알림",MB_OK);
if(bSave == TRUE) {

```

```

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG1),hWnd,File_Name);

```

```

hFile2=CreateFile(SaveFile,GENERIC_WRITE,0,NULL,CREATE_ALWAYS,FILE_ATTRIBUTE_NORMA
L,NULL);

```

```

WriteFile(hFile2,Arr,lstrlen(Arr),&dwWrite,NULL);
CloseHandle(hFile2);
strcpy(enc,"insert into ddec values (");
strcat(enc,ip); strcat(enc,",");
strcat(enc,time);
strcat(enc,",");
strcat(enc,SaveFile);
strcat(enc,"");
mysql_query(conn,enc);

```

```

}
else MessageBox(hWnd,"복호화할 파일을 열어주세요.", "오류
",MB_OK);
bStart = FALSE;
break;

```

```

}
return 0;

```

```

case WM_CTLCOLOREDIT:

```

```

if((HWND)lParam==c4) {
SetTextColor((HDC)wParam,RGB(215,215,215));
SetBkColor((HDC)wParam,RGB(0,0,0));
return (LRESULT)hBrushEdit;
}

```

```

break;

```

```

case WM_DESTROY:

```

```

DeleteObject(hBrushEdit);
DeleteObject(hFontEdit);

```

```

        PostQuitMessage(0);
        return 0;
        mysql_close(conn);
    }
    return(DefWindowProc(hWnd,iMessage,wParam,lParam));
}

```

```

admin.cpp
#include <windows.h>
#include "resource.h"
#include <mysql.h>

#define MYSQL_HOST "10.26.0.48"
#define MYSQL_ID "root"
#define MYSQL_PW "12345"
#define MYSQL_DB "lkc"

LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);
HINSTANCE g_hInst;
HWND c1,c2,c3,c4,c5,c6;
HWND hMDI1 = NULL;
LPSTR lpszClass="admin";
enum{ID_BUT1 = 20,ID_BUT2,ID_BUT3,ID_BUT4,ID_BUT5,ID_BUT6};

//서버
MYSQL *conn, mysql;
MYSQL_RES* RES;
MYSQL_ROW ROW;
int query_stat;
char row_id[40]={NULL};
char row_pw[40]={NULL};

// 윈도우 프로그램의 시작점(윈도우 메인)
int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpszCmdParam,int nCmdShow)
{
    HWND hWnd; //메세지를 받을 윈도우 핸들
    MSG Message;
    WNDCLASS WndClass;
    g_hInst=hInstance;

    WndClass.cbClsExtra=0;
    WndClass.cbWndExtra=0;
    WndClass.hbrBackground=CreateSolidBrush(RGB(121,170,255));
    WndClass.hCursor=LoadCursor(NULL, IDC_ARROW);
    WndClass.hIcon=LoadIcon(hInstance,MAKEINTRESOURCE(IDI_ICON1));
    WndClass.hInstance=hInstance;

```



```

WndClass.lpfWndProc=WndProc;
WndClass.lpszClassName=lpszClass;
WndClass.lpszMenuName=NULL;
WndClass.style=CS_HREDRAW | CS_VREDRAW;
RegisterClass(&WndClass);

// 등록된 윈도우 클래스로 윈도우를 생성한다
hWnd=CreateWindow(lpszClass,lpszClass,WS_SYSMENU | WS_MINIMIZEBOX,
    450,450,400,200,NULL,(HMENU)NULL,hInstance,NULL);
ShowWindow(hWnd,nCmdShow); //윈도우를 보여준다.

while(GetMessage(&Message,0,0,0)) {
    TranslateMessage(&Message);
    DispatchMessage(&Message);
}
return (int)Message.wParam;
}

//ID,passwd 확인하는 부분
BOOL CALLBACK add(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    switch(iMessage){
    case WM_COMMAND:
        switch(LOWORD(wParam)){
        case IDOK:
            query_stat = mysql_query(conn, "select *from login");
            RES = mysql_store_result(conn);
            ROW = mysql_fetch_row(RES);
            GetDlgItemText(hDlg,IDC_ID,row_id,sizeof(row_id));
            GetDlgItemText(hDlg,IDC_PW,row_pw,sizeof(row_pw));
            EndDialog(hDlg,IDOK);
            break;
        case IDCANCEL:
            EndDialog(hDlg,IDD_DIALOG1);
            break;
        }
        break;
    }
    return 0;
}

//사용자 삭제하는 부분
BOOL CALLBACK del(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    switch(iMessage){
    case WM_COMMAND:
        switch(LOWORD(wParam)){
        case IDOK:
            query_stat = mysql_query(conn, "select *from login");
            RES = mysql_store_result(conn);
            ROW = mysql_fetch_row(RES);
            GetDlgItemText(hDlg,IDC_DEL,row_id,sizeof(row_id));

```

```

        case IDCANCEL:
            EndDialog(hDlg,IDD_DIALOG2);
            }
            break;
        }
        return 0;
    }
}
// 사용자 목록 부분
BOOL CALLBACK llist(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    int num;
    int i=0,j=0;
    char qstr[50000]="";
    switch(iMessage){

        case WM_INITDIALOG:

            query_stat = mysql_query(conn, "select *from login");
            RES = mysql_store_result(conn);
            num = mysql_num_fields(RES);
            strcat(qstr, " ID          Passwd ");
            strcat(qstr, "WrWn");
            strcat(qstr, "WrWn");
            while ((ROW = mysql_fetch_row(RES)))
            {
                for(j=0;j<num;j++){
                    strcat(qstr,ROW[j]);
                    strcat(qstr,"          ");
                }
                strcat(qstr,"WrWn");
            }
            SetDlgItemText(hDlg,IDC_LLIST,qstr);
            break;

        case WM_COMMAND:
            switch(LOWORD(wParam)){
        case IDCANCEL:
            EndDialog(hDlg,IDD_DIALOG3);
            break;
            }
        }
        return 0;
    }
}
//사용자 기록 부분
BOOL CALLBACK llog(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    int num;
    int i=0,j=0;
    char qstr[50000]="";
    switch(iMessage){

```

```

case WM_INITDIALOG:

    query_stat = mysql_query(conn, "select *from login_log");
    RES = mysql_store_result(conn);
    num = mysql_num_fields(RES);
    strcat(qstr, "      IP                               Time
ID ");
    strcat(qstr, "WrWn");
    strcat(qstr, "WrWn");
    while ((ROW = mysql_fetch_row(RES)))
    {
        for(j=0;j<num;j++){
            strcat(qstr,ROW[j]);
            strcat(qstr, "
");
        }
        strcat(qstr, "WrWn");
    }
    SetDlgItemText(hDlg, IDC_LLIST, qstr);
    break;

case WM_COMMAND:
    switch(LOWORD(wParam)){
case IDCANCEL:
    EndDialog(hDlg,IDD_DIALOG3);
    break;
    }
}
return 0;
}

// 암호화 기록 부분
BOOL CALLBACK enc1(HWND hDlg,UINT iMessage,WPARAM wParam,LPARAM lParam)
{
    int num;
    int i=0,j=0;
    char qstr[50000]="";
    switch(iMessage){
case WM_INITDIALOG:
    query_stat = mysql_query(conn, "select *from enc");
    RES = mysql_store_result(conn);
    num = mysql_num_fields(RES);
    strcat(qstr, "      IP                               Time
File ");
    strcat(qstr, "WrWn");
    strcat(qstr, "WrWn");
    while ((ROW = mysql_fetch_row(RES)))
    {
        for(j=0;j<num;j++){
            strcat(qstr,ROW[j]);
            strcat(qstr, "
");
        }
        strcat(qstr, "WrWn");
    }
}
}

```

```

        SetDlgItemText(hDlg, IDC_ENC1, qstr);
        break;
    case WM_COMMAND:
        switch(LOWORD(wParam)){
    case IDCANCEL:
        EndDialog(hDlg, IDD_DIALOG4);
        break;
        }
    }
    return 0;
}
// 복호화 기록 부분
BOOL CALLBACK dec(HWND hDlg, UINT iMessage, WPARAM wParam, LPARAM lParam)
{
    int num;
    int i=0, j=0;
    char qstr[50000]="";
    switch(iMessage){
    case WM_INITDIALOG:
        query_stat = mysql_query(conn, "select *from ddec");
        RES = mysql_store_result(conn);
        num = mysql_num_fields(RES);
        strcat(qstr, "      IP                               Time
File ");
        strcat(qstr, "WrWn");
        strcat(qstr, "WrWn");
        while ((ROW = mysql_fetch_row(RES)))
        {
            for(j=0; j<num; j++){
                strcat(qstr, ROW[j]);
                strcat(qstr, "          ");
            }
            strcat(qstr, "WrWn");
        }
        SetDlgItemText(hDlg, IDC_DEC1, qstr);
        break;
    case WM_COMMAND:
        switch(LOWORD(wParam)){
    case IDCANCEL:
        EndDialog(hDlg, IDD_DIALOG5);
        break;
        }
    }
    return 0;
}

LRESULT CALLBACK WndProc(HWND hWnd, UINT iMessage, WPARAM wParam, LPARAM lParam)
{

```

```

char add1[200]="";
char del1[200]="";

switch(iMessage) {
case WM_CREATE:
    mysql_init(&mysql); // mysql 초기화
    conn = mysql_real_connect(&mysql, MYSQL_HOST, MYSQL_ID,
MYSQL_PW, MYSQL_DB, 3306, (char *)NULL, 0); // 서버 연결

    c1 = CreateWindow(TEXT("button"),TEXT("사용자 추가"),WS_CHILD |
WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
30,35,100,34,hWnd,(HMENU)ID_BUT1,g_hInst,NULL);
    c2 = CreateWindow(TEXT("button"),TEXT("사용자 삭제"),WS_CHILD |
WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
150,35,100,34,hWnd,(HMENU)ID_BUT2,g_hInst,NULL);
    c3 = CreateWindow(TEXT("button"),TEXT("사용자 목록"),WS_CHILD |
WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
270,35,100,34,hWnd,(HMENU)ID_BUT3,g_hInst,NULL);
    c6 = CreateWindow(TEXT("button"),TEXT("사용자 기록"),WS_CHILD |
WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
30,100,100,34,hWnd,(HMENU)ID_BUT6,g_hInst,NULL);
    c4 = CreateWindow(TEXT("button"),TEXT("암호화 기록"),WS_CHILD |
WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
150,100,100,34,hWnd,(HMENU)ID_BUT4,g_hInst,NULL);
    c5 = CreateWindow(TEXT("button"),TEXT("복호화 기록"),WS_CHILD |
WS_VISIBLE | WS_BORDER | BS_PUSHBUTTON,
270,100,100,34,hWnd,(HMENU)ID_BUT5,g_hInst,NULL);
    break;

case WM_COMMAND:
    switch(LOWORD(wParam)) {
    case ID_BUT1:

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG1),hWnd,add);
        if(row_id!=NULL){
            strcpy(add1,"insert into login(id,pw) values ('");
            strcat(add1,row_id);
            strcat(add1,",");
            strcat(add1,row_pw);
            strcat(add1,"')");
            mysql_query(conn,add1);
            MessageBox(hWnd,"사용자가 추가 되었습니다.", "성
공",MB_OK);
        }
        else
            MessageBox(hWnd,"아이디를 입력하세요.", "성공
",MB_OK);
        break;

    case ID_BUT2:

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG2),hWnd,del);

```

```
        strcpy(del1,"delete from login where id=");
        strcat(del1,row_id);
        strcat(del1,"");
        mysql_query(conn,del1);
        MessageBox(hWnd,"사용자가 삭제 되었습니다.", "성공",MB_OK);
        break;

        case ID_BUT3:

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG3),hWnd,Ilist);
        break;

        case ID_BUT4:

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG4),hWnd,enc1);
        break;

        case ID_BUT5:

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG5),hWnd,dec);
        break;
        case ID_BUT6:

DialogBox(g_hInst,MAKEINTRESOURCE(IDD_DIALOG6),hWnd,Ilog);
        break;
        }
        break;
case WM_DESTROY:
        PostQuitMessage(0);
        mysql_close(conn);
        return 0;
}
return(DefWindowProc(hWnd,iMessage,wParam,lParam));
}
```

5. 결 론

암호 기법을 사용하여 사용자가 자신이 작성한 파일을 암호화 하여 안전하게 저장 할 수 있다. 저장한 파일은 유출되더라도 암호화에 사용된 키 값을 모르기 때문에 복호화를 하기 어렵다. 또한 자신이 필요할 때마다 복호화를 하여 파일을 열고 수정이 가능하다. 그리고 사용자의 로그인한 기록과 파일을 암호화 한 기록이 서버에 기록되어 관리자가 확인이 가능해 파일 유출에 대해 추적 및 관리가 유용하다.

이번 프로젝트는 MySQL을 Database로 사용하는 프로그램을 구현하였다. Database를 선정하는 과정 중 Microsoft에서 나온 MSSQL과 운영체제에 직접 연동하여 사용 할 수 있는 ODBC를 알게 되었는데 MySQL과 같은 비슷하지만 다른 점들이 있었다. 따라서 이번 프로젝트를 향후 보완, 발전시킨다면 사용하였던 MySQL이 아닌 Microsoft에서 나온 모든 프로그램과 연동이 되는 MSSQL이나 ODBC를 사용하여 다양한 Database를 배워보아야겠다. 또한 API를 이용하여 프로젝트 활성창을 작성하였는데 프로젝트를 진행하면서 공부한 API를 이용하여 좀 더 심화된 MFC까지 개인 능력을 향상 시키고 이를 바탕으로 MFC 대한 이해도를 높여 활용 가능한 Program Language를 넓혀야겠다고 생각했다.

6. 참고 문헌

- 윈도우즈 API 정복 1 · 2 (김상형 지음/한빛 미디어)
- MySQL의 사용, 관리, 프로그래밍을 위한 완벽 가이드(폴 드보이/지앤선)
- 데이터베이스 시스템 = Database system : 설계, 구축 및 관리 (김성언/학현사)
- 네이버 지식 백과 <http://terms.naver.com/>
- MySQL <http://dev.mysql.com/downloads/connector/cpp/>

파일 유출방지 시스템

2014.5.27

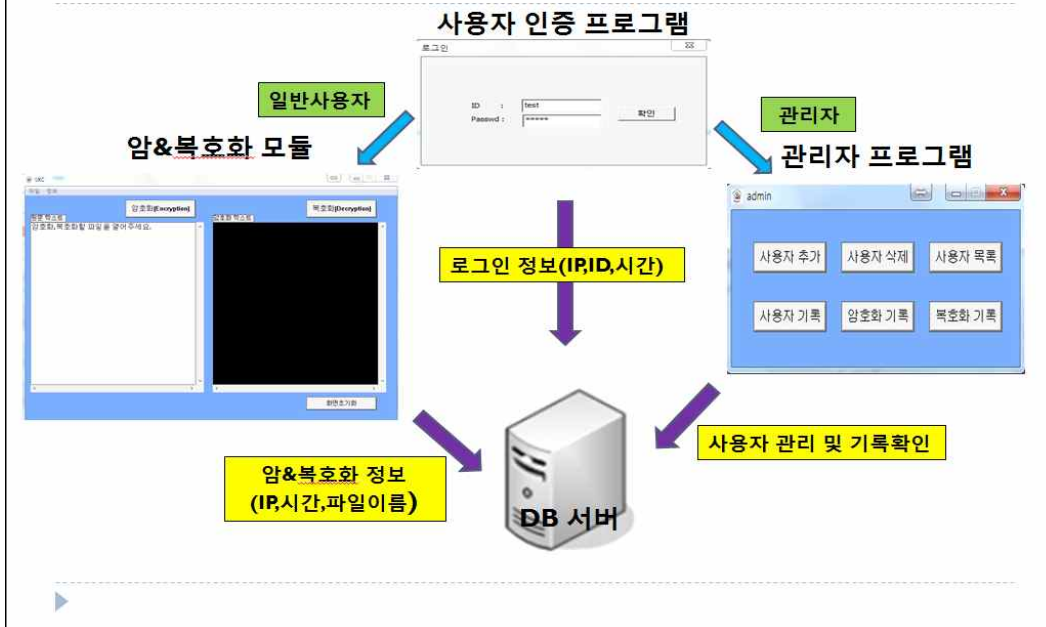
담당 교수님 : 양환석 교수님

1조 LKC security

목 차

- ▶ 조원 소개 및 역할
 - ▶ 주제 선정
 - ▶ 구상도
 - ▶ 개발 환경
 - ▶ 시스템 구현
 - ▶ 결론 및 의견
-

구상도

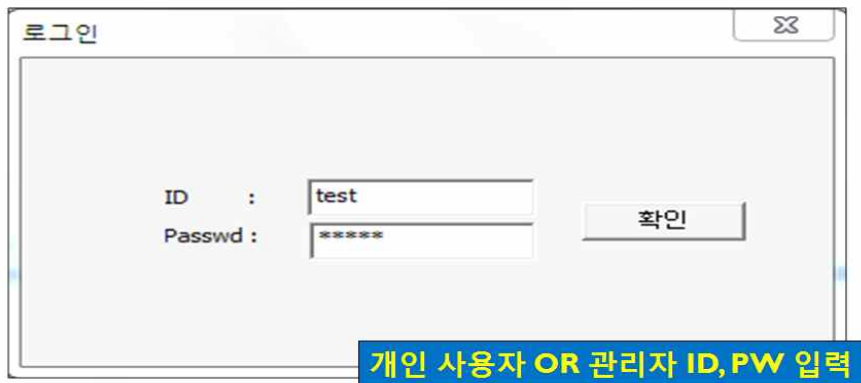


개발 환경

- ▶ Windows 7
- ▶ Visual Studio 2010
 - C++
 - API
- ▶ MYSQL 서버

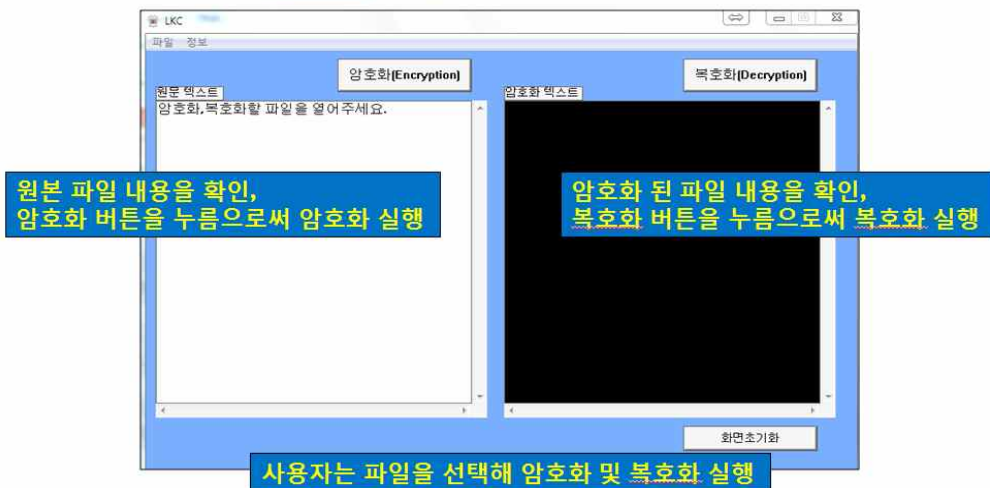
시스템 구현

◆ 사용자 인증

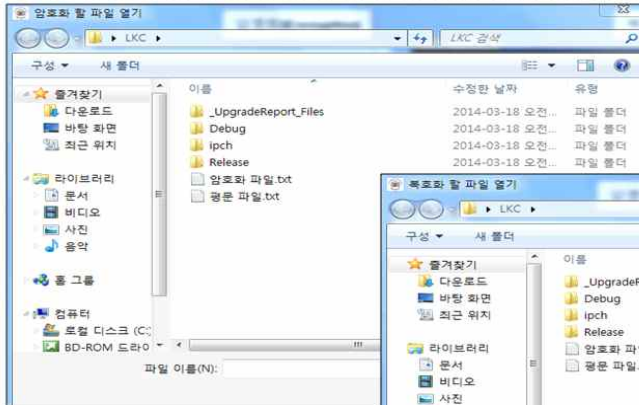


시스템 구현

◆ 암호화 및 복호화 화면

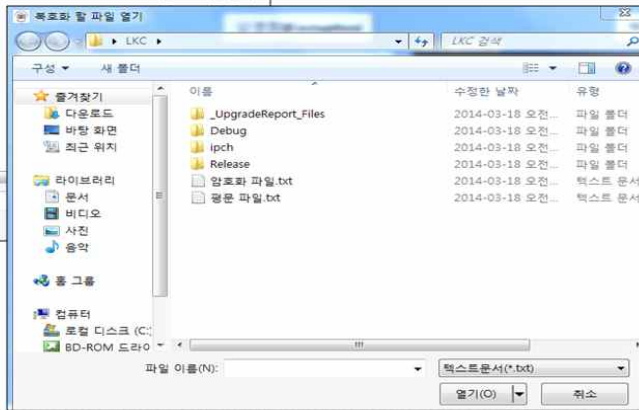


시스템 구현

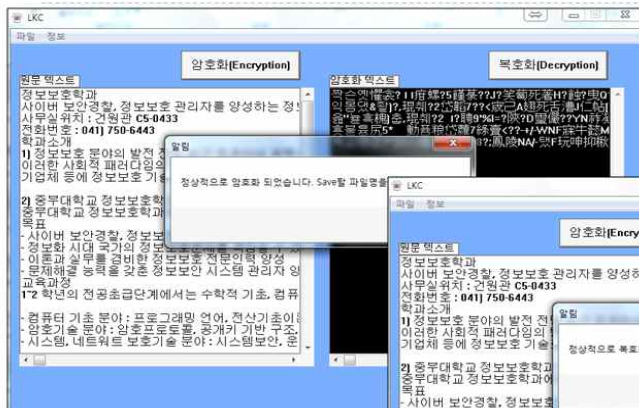


암호화 파일열기

복호화 파일열기

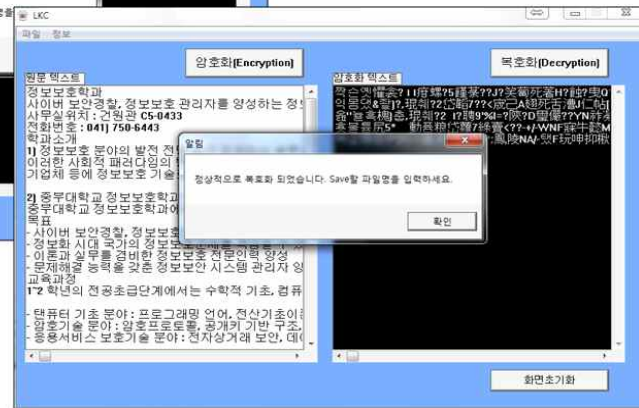


시스템 구현



암호화 실행

복호화 실행



시스템 구현

암호화 파일저장

복호화 파일저장


시스템 구현

◆ 관리자 화면

관리자는 사용자 관리 및 로그 기록 확인 가능

시스템 구현

◆ 사용자 추가



사용자 추가

ID : lkc

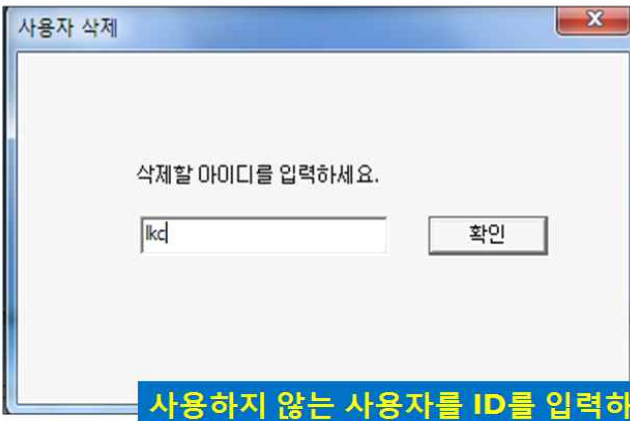
Passwd : *****

확인

관리자는 일반사용자의 ID, PW를 입력해 추가 가능

시스템 구현

◆ 사용자 삭제



사용자 삭제

삭제할 아이디를 입력하세요.

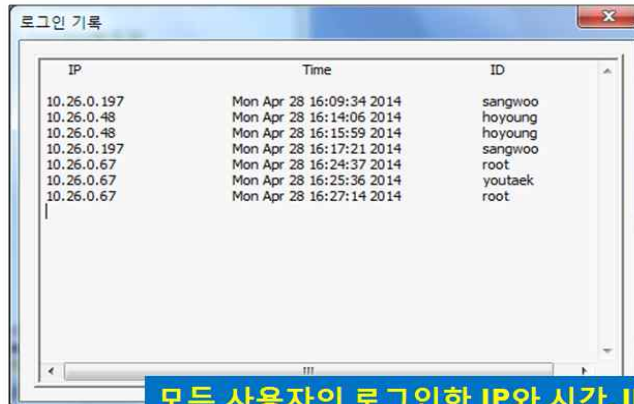
lkd

확인

사용하지 않는 사용자를 ID를 입력하여 삭제 가능

시스템 구현

◆ 로그인 기록

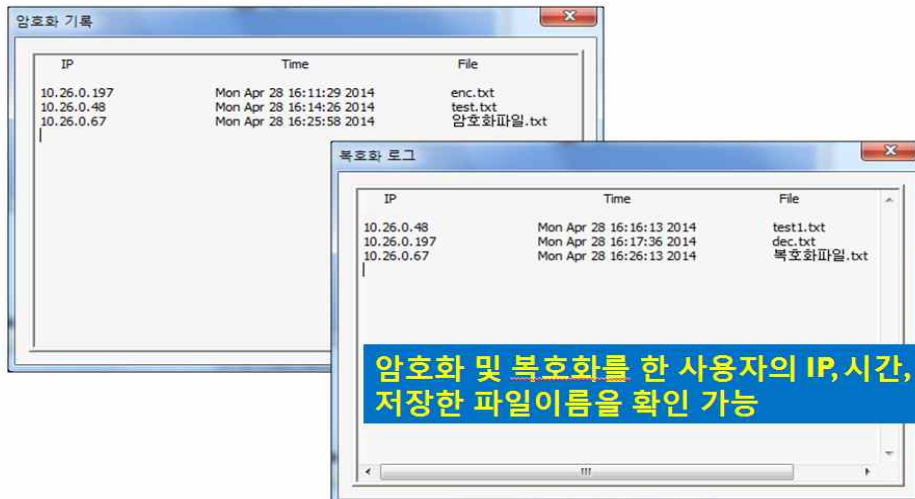


IP	Time	ID
10.26.0.197	Mon Apr 28 16:09:34 2014	sangwoo
10.26.0.48	Mon Apr 28 16:14:06 2014	hoyoung
10.26.0.48	Mon Apr 28 16:15:59 2014	hoyoung
10.26.0.197	Mon Apr 28 16:17:21 2014	sangwoo
10.26.0.67	Mon Apr 28 16:24:37 2014	root
10.26.0.67	Mon Apr 28 16:25:36 2014	youtaek
10.26.0.67	Mon Apr 28 16:27:14 2014	root

모든 사용자의 로그인한 IP와 시간, ID를 확인이 가능

시스템 구현

◆ 암호화 및 복호화 기록



IP	Time	File
10.26.0.197	Mon Apr 28 16:11:29 2014	enc.txt
10.26.0.48	Mon Apr 28 16:14:26 2014	test.txt
10.26.0.67	Mon Apr 28 16:25:58 2014	암호화파일.txt

IP	Time	File
10.26.0.48	Mon Apr 28 16:16:13 2014	test1.txt
10.26.0.197	Mon Apr 28 16:17:36 2014	dec.txt
10.26.0.67	Mon Apr 28 16:26:13 2014	복호화파일.txt

암호화 및 복호화를 한 사용자의 IP, 시간, 저장한 파일이름을 확인 가능

결론 및 의견

- ◆ 파일 유출 방지 시스템을 개발
 - 파일 암호화 및 복호화를 통해 파일 유출시에도 정보를 안전하게 보호
 - 특히 로그인, 암호화 및 복호화 과정에서 로그 기록을 저장, 유출상황 추적이 가능
- ◆ 개발시스템에서 암호화 및 복호화 Key의 가변화, MSSQL, ODBC 등 데이터베이스 운영환경 다변화 방안을 강구할 경우, 시스템의 활용도 극대화 가능



Thank you