2011년 종합설계1

졸업작품 최종 보고서

200011436 김국영
200511349 장기웅
200010639 김재홍

Contents.

Graduation Project

Title : 인터넷을 통한 음성통화(VOIP)

Members :

Name	Student ID	E-mail	Cell Phone
김국영	200011436	webmaster@game.re.kr	010-5305-9471
장기웅	200511349	wkdrldnd09@nate.com	010-7184-8499
김재홍	200010639	muscovite81@hotmail.com	010-7760-3360

담당 교수님 : 유준범 교수님

주제선정의 배경 및 목적

○ 배경

- 인터넷을 통한 무료 SMS 서비스(카카오톡,마이피플) 등의 관심이 높아지는 가운데 인터넷을 통한 무료 통화 서비스에도 많은 관심이 집중 되고 있다.

○ 목적

- 인터넷을 통해 기본적인 기능인 통화를 가능하게 구현하고 기존 유선통화보다 더 좋은 음질의 통화를 할 수 있게 하는 데에 목적이 있음.
- 보안을 통한 음성통화를 구현하도록 하고 인증서를 통해 사용자 계정의

도용을 막는 기술 도입을 시도한다.

개발 내용 및 최종 목표

○ 개발 내용

- 윈도우즈 기반의 어플리케이션
- 서버 및 클라이언트를 구현하여 서버는 각 클라이언트들간의 접속에 필요한 정보를 제공할 수 있게 구현한다.
- 클라이언트는 서버에서 접속할 다른 클라이언트(들)의 정보를 얻고 다른 클라이언트와 접속하여 음성통화를 할 수 있게 구현한다.
- 보안모드를 따로 운용해 보안모드 시에 음성데이터(패킷)을 암호화 하여 전달 할 수 있도록 한다.
- 인증서를 통해 사용자 자신의 핸드폰 번호와 어플리케이션 이용에 쓰는 사용자 자신의 번호를 동일화 시켜 사용자 계정 도용을 방지를 할 수 있도록 한다.
- 컴퓨터에 설치되어 있는 마이크 장치가 여러개일 경우를 대비하여 선택 운용 할 수 있도록 한다.

○ 최종 목표

- 인터넷을 통한 무료 통화 구현
- 안정적인 접속이 가능한 어플리케이션 구현
- 기존 유선통화보다 더 좋은 음질의 통화가 가능한 어플리케이션 구현
- 보안기술을 이용한 통화 구현
- 인증서를 통한 보안 기능 구현

개발 환경

- 시스템 : 마이크로소프트 윈도우 환경

- 서버 : PHP
- 서버 데이터베이스 : MYSQL
- 클라이언트 : C++
- 개발도구 : Visual Studio 6.0

요구사항 분석

○ 서버 동작

Ref	Function
R1.1	서버 관리자는 서버를 시작/정지가 가능하다.
R1.2	사용자의 회원가입 요청시 인증서를 통한 가입 절차를 수행할 수 있어야한다.
R1.3	가입 요청 시 ID 중복 및 입력 정보의 유효성 검사를 한다.
R1.4	서버는 데이터베이스와 연동하여 동작한다.
R1.5	서버는 등록된 회원의 관리가 가능하다.
R1.6	서버는 클라이언트의 정보 요청시 요청된 정보를 전달 가능해야 한다.

○ 클라이언트 동작

Ref	Function
R2.1	클라이언트는 서버와 접속이 가능해야 한다.
R2.2	클라이언트는 회원가입을 할 수 있어야 한다.
R2.3	클라이언트는 회원가입시 인증서를 생성하는 절차를 수행 할 수 있어야한다.

R2.4	클라이언트는 접속할 다른 클라이언트의 정보를 서버에게 요청하고 수신할 수 있어야한다..
R2.5	클라이언트는 사용할 마이크 장치의 선택이 가능해야한다.
R2.6	클라이언트는 접속할 다른 클라이언트의 번호를 마우스/키보드를 통해 입력 받을 수 있어야한다.
R2.7	클라이언트는 서버 접속 후 다른 클라이언트부터의 통화 요청을 받을 수 있도록 대기 할 수 있어야한다.
R2.8	클라이언트는 서버에서 수신된 정보를 통해 다른 클라이언트에 접속 요청을 할 수 있어야한다.
R2.9	클라이언트는 음성정보를 송수신 할 수 있어야한다.
R2.10	클라이언트는 수신된 음성정보를 스피커 장치를 통해 출력할 수 있어야한다.
R2.11	클라이언트는 보안통화 수행시 압축된 음성정보를 암호화/복호화 할 수 있어야한다.
R2.12	클라이언트는 스피커와 마이크의 볼륨을 조절할 수 있어야한다.

유스케이스

서버

Usecase	서버 시작	
Actor	서버 관리자	
Type	Primary	
사전조건	데이터베이스가 동작 중인 상태	
기본흐름	Actor	System
	1. 서버 동작 요청을 한다.	2. 데이터베이스와의 연결 상태를 검사한다.(E-1) 3. 데이터베이스 연결에 문제가 없는 경우 정상적으로 동작을 시작한다.
예외흐름	Actor	System
		E-1. 데이터베이스와 연결이 실패할 경우 에러 메시지를 출력한다.

Usecase	서버 중단	
Actor	서버 관리자	
Type	Primary	
사전조건	서버가 동작 중인 상태	
기본흐름	Actor	System
	1. 서버 종료 요청을 한다.	2. 현재 접속 중인 사용자에게 서버 종료 메시지를 보낸다. 3. 사용자와의 연결을 끊는다. 4. 서버의 동작을 중단한다.

Usecase	회원가입	
Actor	클라이언트	
Type	Primary	
사전조건	클라이언트와의 접속,데이터베이스 연결	
기본흐름	Actor	System
	1. 회원가입 요청 3. 회원 정보 전송	2. 요청받은 후 클라이언트에 다시 요청 승인 전송 4. 전송받은 회원 정보를 기존 데이터베이스와 비교 5. 전송받은 회원 정보를 데이터베이스에 저장 6. 클라이언트에 성공을 알림
예외흐름	Actor	System
		E-1. 기존 데이터베이스와 비교시 같은 정보를 가지고 있으면 클라이언트에 알림

Usecase	인증번호 발송	
Actor	클라이언트	
Type	Primary	
사전조건	클라이언트와의 접속,데이터베이스 연결	
기본흐름	Actor	System
	1. 인증번호 요청 4. 인증번호 전송	2. 인증번호 생성 3. 인증번호를 접속한 회원 정보의 전화번호로 발송 5. 전송받은 인증번호를 비교 후 성공확인 6. 클라이언트에 성공을 알림
예외흐름	Actor	System
	1. 인증번호 재요청	E-1. 잘못된 인증번호일시 클라이언트에 잘못된 인증번호임을 알림] 2.기본 흐름의 2~6의 단계를 시행

클라이언트

Usecase	클라이언트 시작	
Actor	사용자	
Type	Primary	
사전조건	없음	
기본흐름	Actor	System
	1. 프로그램 실행	2. 각 하드웨어 장치를 사용할 수 있도록 한다. 3. 회원로그 파일을 검사한다. 4. 로그파일에 정보가 있을 시 로그 파일의 정보대로 서버에 접속한다. 4.리스너 프로세스를 생성 대기한다.
예외흐름	Actor	System
		E-1. 로그파일이 없거나 정보가 없을 시 서버 접속을 수행하지 않는다. E-2. 로그파일의 정보로 서버 접속이 실패할 경우 그대로 대기한다.

Usecase	회원가입	
Actor	사용자	
Type	Primary	
사전조건	프로그램 실행	
기본흐름	Actor	System
	1. 회원가입 요청 4. 회원 가입 양식창에 회원 정보를 기입한다.	2. 서버에 접속한다. 3. 회원 가입 양식 입력 창을 띄운다. 5. 입력된 회원 정보를 서버로 전송하여 서버 데이터베이스로 저장

		할 수 있게끔한다. 6. 성공 메시지를 출력한다.
예외흐름	Actor	System
		E-1. 잘못된 양식의 회원정보를 입력시 에러 메시지를 출력한다.

Usecase	인증번호를 통한 사용자 인증	
Actor	사용자	
Type	Primary	
사전조건	회원정보를 통한 서버로 로그인 완료	
기본흐름	Actor	System
	1. 인증을 위한 인증번호를 요청 3. 인증번호를 입력	2. 로그인을 한 회원의 전화번호 정보를 이용한 인증번호 발송을 서버에 요청 4. 입력된 인증번호를 서버에 전송 5. 성공 확인 정보를 서버로부터 수신 6. 성공 메시지를 출력한다.
예외흐름	Actor	System
	1. 인증번호를 재요청	E-1. 서버에서 실패 메시지 수신시 실패 메시지를 출력한다. 2. 기본흐름의 2~6에 단계를 시행

Usecase	통화 요청	
Actor	사용자	
Type	Primary	
사전조건	회원정보를 통한 서버로 로그인 완료	
기본흐름	Actor	System
	1. 통화를 할 상대방의 전화번호 입력후 통화를 요청	2. 상대방의 전화번호를 서버에 전

		송 3. 상대방의 IP 정보를 서버로부터 전송 받음 4. 전송받은 IP 정보를 통해 상대 클라이언트에 접속 요청 5. 화면에 전화를 걸고 있음을 출력 6. 상대로부터의 접속 확인을 받음 7. 화면에 상대와 연결 되었음을 출력함
예외흐름	Actor	System
		E-1. 서버로부터 상대방이 서버에 접속이 되어 있지 않음을 통보 받을 때 화면에 출력 E-2. 상대방이 수신을 거부(또는 실패) 시 화면에 출력

Usecase	통화 받음	
Actor	사용자	
Type	Primary	
사전조건	회원정보를 통한 서버로 로그인 완료, 상대의 통화 요청이 들어옴	
기본흐름	Actor	System
	3. 전화 요청을 받아들임	1. 리스너가 상대의 접속 요청을 받아들임 2. 전화 요청이 있음을 화면에 출력 4. 상대방에게 접속 승인을 알림 5. 화면에 접속이 이루어졌음을 출력
예외흐름	Actor	System
	E-1. 전화요청을 거부함	E-2. 전화 거부를 상대 클라이언트에게 통보

Usecase	통화 요청 (보안모드)	
Actor	사용자	
Type	Primary	
사전조건	회원정보를 통한 서버로 로그인 완료	
기본흐름	Actor	System
	1. 통화를 할 상대방의 전화번호 입력후 통화를 요청	2. 상대방의 전화번호를 서버에 전송 3. 상대방의 IP 정보를 서버로부터 전송 받음 4. 전송받은 IP 정보를 통해 상대 클라이언트에 접속 요청 5. 요청시 openssl에서 생성된 자신의 공개키를 같이 전송 6. 화면에 전화를 걸고 있음을 출력 7. 상대로부터의 접속 확인과 암호화된 보안대칭키, 상대의 openssl 공개키를 받음 8. 암호화된 보안대칭키를 상대의 공개키로 복호화한 후 자신의 비밀키로 복호화하여 보안대칭키를 얻음 9. 화면에 상대와 연결 되었음을 출력함
예외흐름	Actor	System
		E-1. 서버로부터 상대방이 서버에 접속이 되어 있지 않음을 통보 받을 때 화면에 출력 E-2. 상대방이 수신을 거부(또는 실패) 시 화면에 출력

Usecase	통화 받음 (보안모드)	
Actor	사용자	
Type	Primary	
사전조건	회원정보를 통한 서버로 로그인 완료, 상대의 통화 요청이 들어옴	
기본흐름	Actor	System
	3. 전화 요청을 받아들임	1. 상대의 접속 요청과 상대의 openssl 공개키를 받음 2. 전화요청이 있음을 화면에 출력 4. 보안대칭키를 생성하여 자신의 openssl 비밀키로 암호화 한 후 상대의 openssl 공개키로 암호화함. 5. 상대방에게 접속 승인과 암호화된 보안대칭키를 전송함 6. 화면에 상대가 연결되었음을 출력함
예외흐름	Actor	System
	E-1. 전화요청을 거부함	E-2. 전화 거부를 상대 클라이언트에게 통보

Usecase	일반통화(발신)	
Actor	사용자	
Type	Primary	
사전조건	통화 연결	
기본흐름	Actor	System
	1. 마이크에 대고 음성을 입력	2. 입력된 음성을 음성정보로 변환 3. 패킷으로 쪼갠후 상대 클라이언트에 전송
예외흐름	Actor	System

Usecase	일반통화(수신)	
Actor	사용자	
Type	Primary	
사전조건	통화 연결	
기본흐름	Actor	System
		1. 수신된 상대 클라이언트의 음성 패킷을 음성정보로 변환 2. 변환된 음성정보를 스피커를 통해 출력
예외흐름	Actor	System

Usecase	보안통화(발신)	
Actor	사용자	
Type	Primary	
사전조건	통화 연결	
기본흐름	Actor	System
	1. 마이크에 대고 음성을 입력	2. 입력된 음성을 음성정보로 변환 3. 음성정보를 보안대칭키를 이용하여 암호화 하여 전송
예외흐름	Actor	System

Usecase	보안통화(수신)	
Actor	사용자	
Type	Primary	
사전조건	통화 연결	
기본흐름	Actor	System
		1. 수신된 정보를 보안대칭키를 이용하여 복호화 2. 복호화된 정보를 음성정보로 변환 3. 음성정보를 스피커에 출력
예외흐름	Actor	System

Usecase	통화 끊김	
Actor	사용자	
Type	Primary	
사전조건	통화 연결	
기본흐름	Actor	System
	1. 통화 끊김을 요청	2. 통화 끊김을 상대 클라이언트에 통보 3. 연결을 끊음
예외흐름	Actor	System

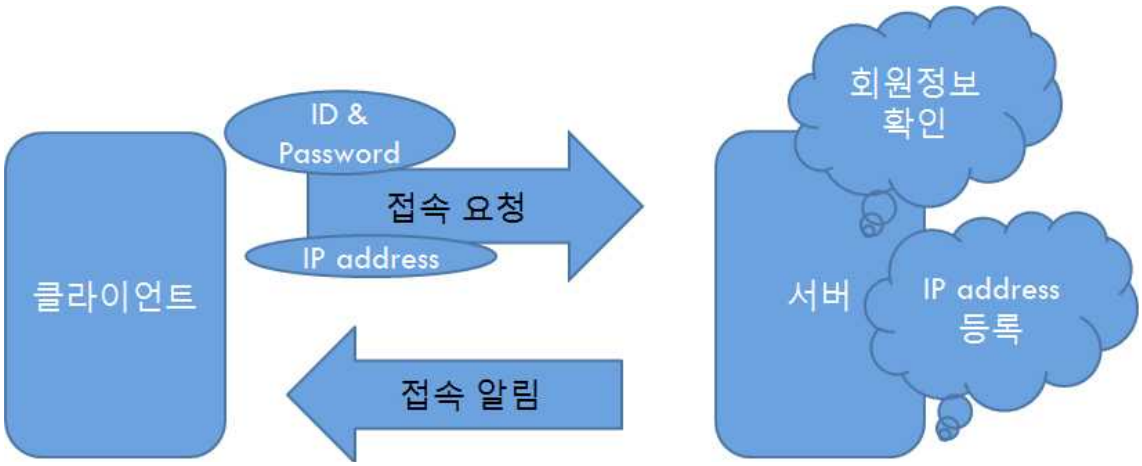
Usecase	마이크 장치 선택	
Actor	사용자	
Type	Secondly	
사전조건	통화 연결	
기본흐름	Actor	System
	1. 선택할 수 있는 마이크 장치 목록을 요청한다. 3. 사용할 마이크 장치를 목록에서 선택한다.	2. 컴퓨터에 설치된 마이크 장치 목록을 검색하고 그 목록을 출력한다. 4. 선택된 마이크 장치를 이용할 수 있도록 프로그램과 연동한다.
예외흐름	Actor	System
		E-1 선택된 장치와 연결할 수 없을 경우 에러 메시지를 출력한다.

Usecase	스피커 볼륨 조절	
Actor	사용자	
Type	Secondly	
사전조건	통화 연결	
기본흐름	Actor	System
	1. 스피커 볼륨조절을 바를 움직여 설정한다.	2. 입력받은 볼륨을 스피커 장치에 적용한다.
예외흐름	Actor	System
		E-1 선택된 장치와 연결할 수 없을 경우 에러 메시지를 출력한다.

Usecase	마이크 볼륨 조절	
Actor	사용자	
Type	Secondly	
사전조건	통화 연결	
기본흐름	Actor	System
	1. 마이크 볼륨조절을 바를 움직여 설정한다.	2. 입력받은 볼륨을 마이크 장치에 적용한다.
예외흐름	Actor	System
		E-1 선택된 장치와 연결할 수 없을 경우 에러 메시지를 출력한다.

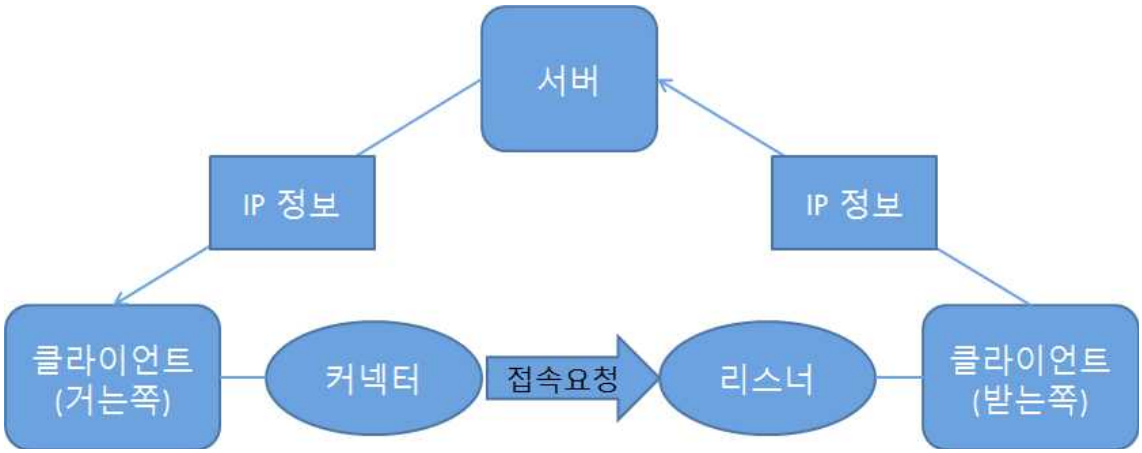
설계 구조 내용

○ 클라이언트와 서버간의 접속(회원정보가 서버에 있을 시)



- 클라이언트는 서버에 접속 요청시 사용자의 ID, Password, IP address 등의 정보를 전송한다.
- 서버는 전송받는 회원정보를 데이터베이스에서 확인 후 전송받은 IP address를 데이터베이스에 저장한다.
- 서버는 클라이언트에게 접속이 되었음을 알린다.
- 회원정보가 서버의 데이터베이스에 없거나 매치가 되지 않을 경우에는 접속 실패를 클라이언트에게 알린다.
- 이 때 저장된 IP address는 클라이언트간의 통화시에 사용된다.

○ 클라이언트 간의 접속



- 클라이언트의 IP 정보는 서버가 가지고 있을 수 있도록 한다.

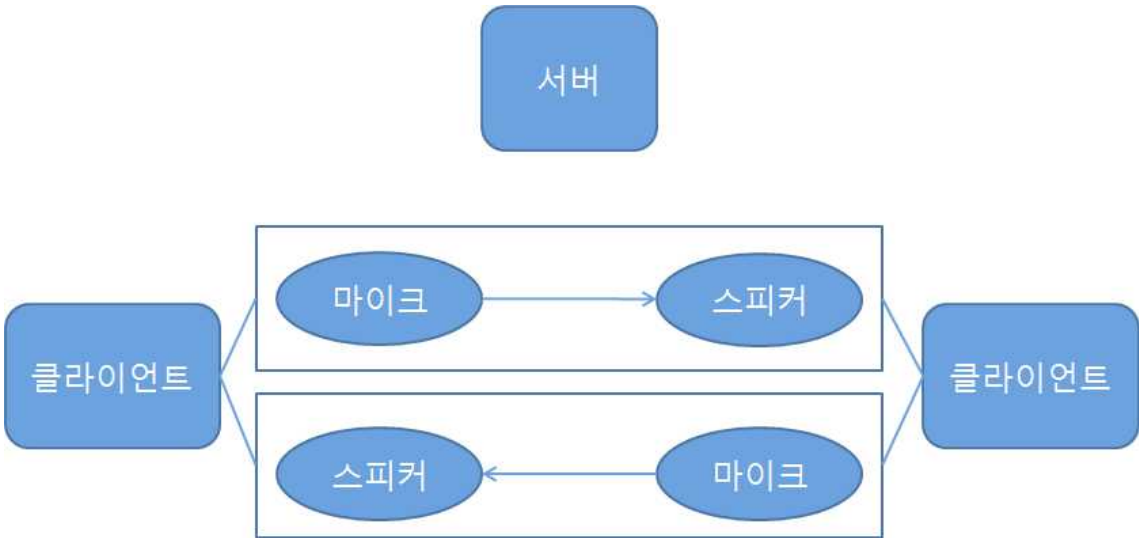
- 클라이언트는 접속할 다른 클라이언트의 IP 정보를 서버에게 요청하고 그 정보를 서버로부터 제공 받는다.

- 클라이언트의 커넥터는 그 IP 정보를 가지고 접속할 클라이언트에게 접속 요청을 한다.

- 접속요청을 받는 클라이언트의 리스너가 이 접속 요청을 받고 클라이언트 간의 접속을 시행한다.

- 클라이언트의 리스너는 항상 접속에 대비하여 대기하고 있도록 한다.

○ 접속 후 클라이언트 간의 관계

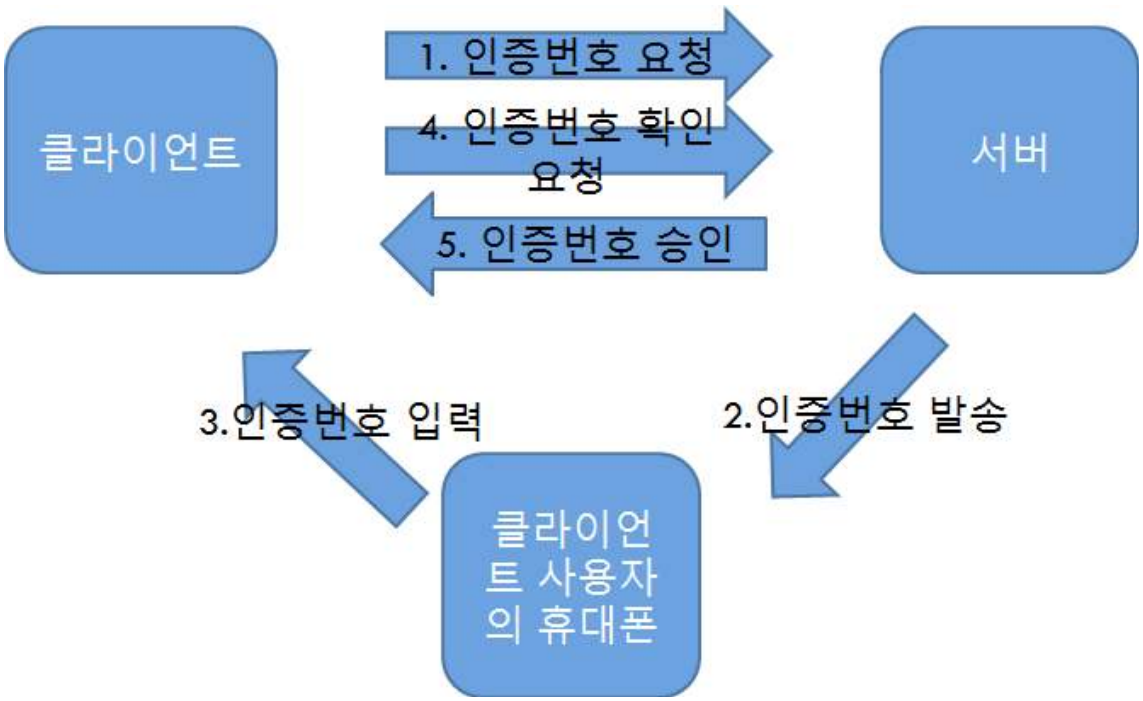


- 각 클라이언트는 통화 중에 혼선이 일어나면 안되므로 마이크와 스피커를 각각 쓰레드로 구현한다.

- 각 클라이언트간의 스피커와 마이크 사이에서 40kbyte/s의 속도로 음성정보에 대한 패킷을 각각 전달받을 수 있게 구현한다.

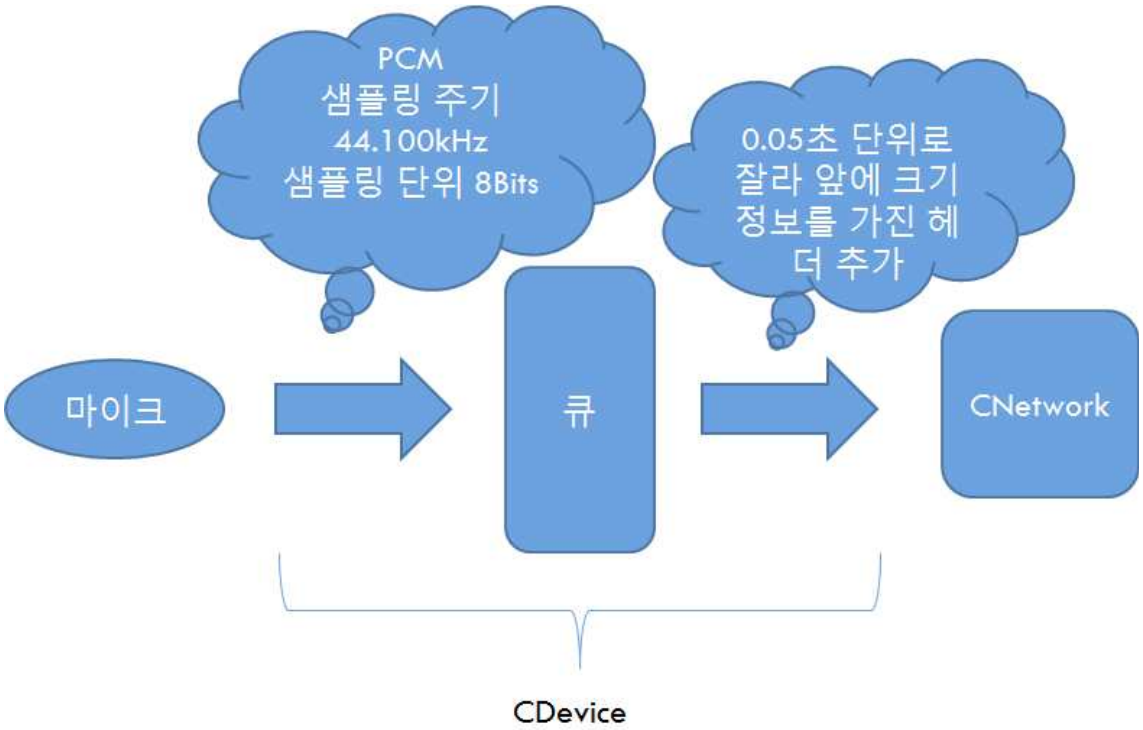
- 각 클라이언트의 마이크와 스피커는 윈도우의 저수준 API를 사용하여 구현한다.

○ 인증번호를 통한 사용자 인증



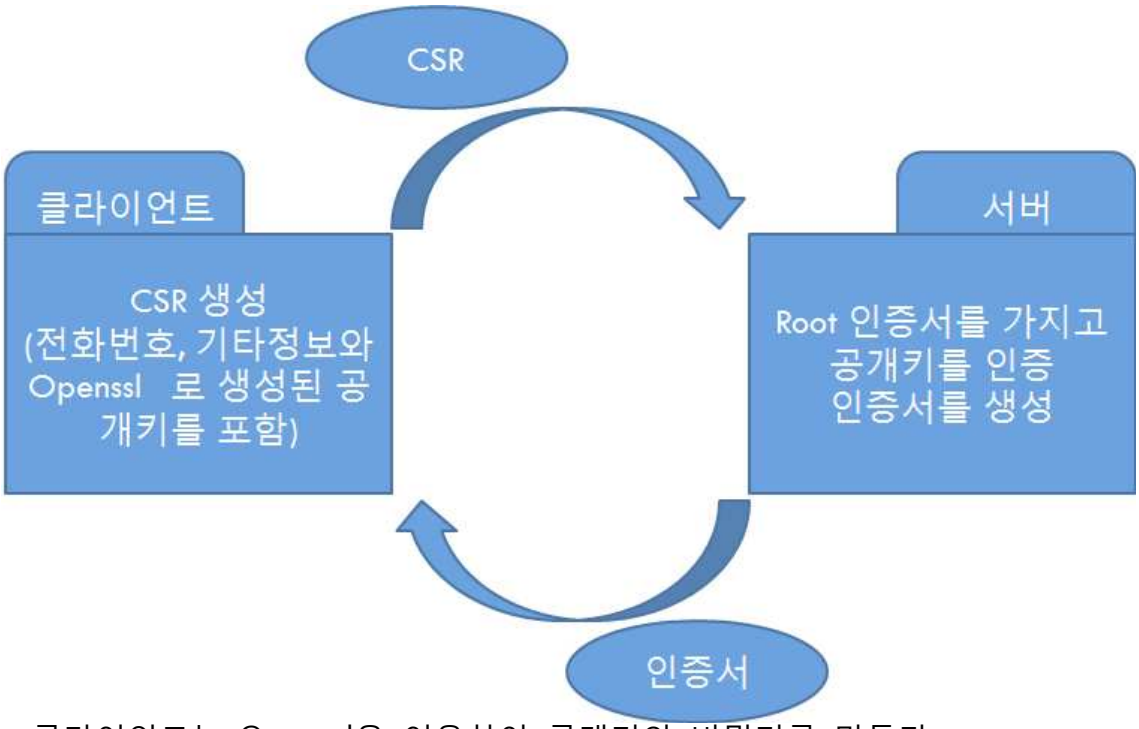
- 클라이언트가 서버에게 인증번호를 요청함
- 서버는 인증번호 요청을 받고 요청을 한 클라이언트의 회원 정보중 전화번호 정보를 데이터베이스에서 호출함
- 클라이언트의 전화번호로 인증번호를 발송함
- 클라이언트의 사용자가 사용자의 휴대폰으로 온 인증번호 SMS를 통해 인증번호를 확인한후 클라이언트에 인증번호를 입력해줌
- 클라이언트는 입력받은 인증번호를 서버에 확인 요청함
- 서버는 확인요청한 인증번호와 발송한 인증번호를 비교한후 승인여부를 클라이언트에게 통보함

○ 음성 정보 변환 과정



- 마이크 장치로부터 음성을 입력받는다.
- 입력된 음성을 PCM 샘플링 주기 44.100kHz 샘플링 단위 8Bits 로 음성정보로 변환하여 큐에 입력한다.
- 큐에 입력된 음성정보를 0.05초 단위로 잘라서 앞에 크기 정보를 가진 헤더를 추가하여 전송에 필요한 패킷을 만든다.
- 이 과정은 CDevice 클래스를 통해 이루어진다.
- 음성패킷은 또다시 큐로 저장된다.

○ 인증서 발급 과정



- 클라이언트는 Openssl을 이용하여 공개키와 비밀키를 만든다.

- 클라이언트는 인증서 발급 요청시 전화번호와 기타정보 그리고 Openssl을 이용하여 만든 공개키를 포함한 CSR을 생성한다.

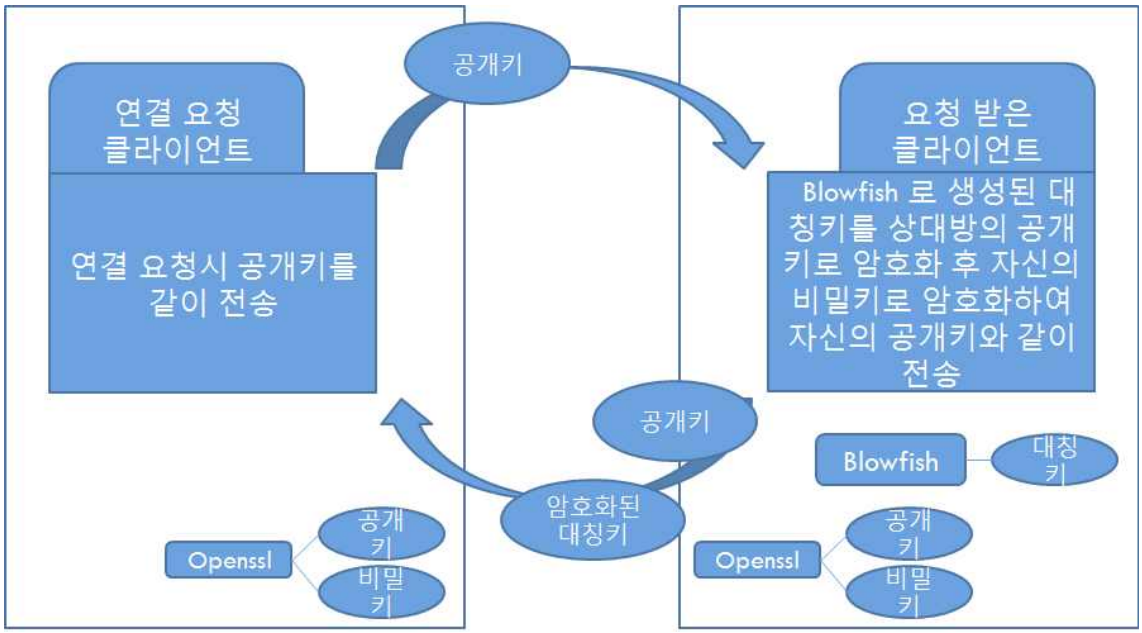
- 클라이언트는 생성된 CSR을 인증서 발급 요청과 함께 전송한다.

- 서버는 Root 인증서를 가지고 받은 CSR 중 클라이언트의 공개를 인증하여 인증서를 생성한다.

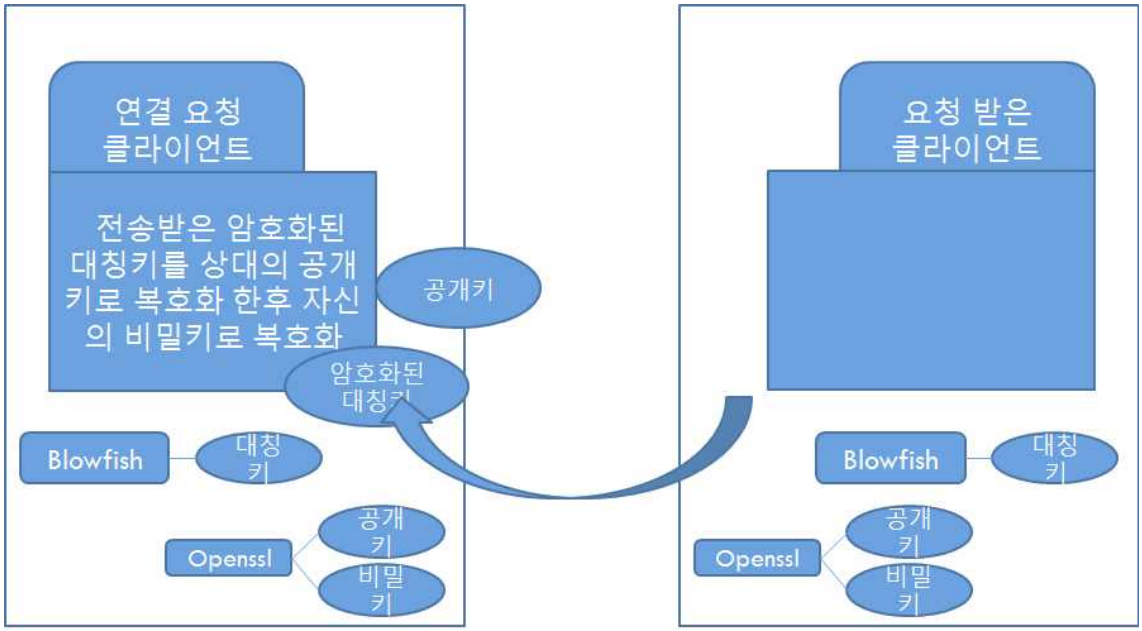
- 서버는 생선된 인증서를 클라이언트에게 전송한다.

- 클라이언트는 전송된 인증서를 저장한다.

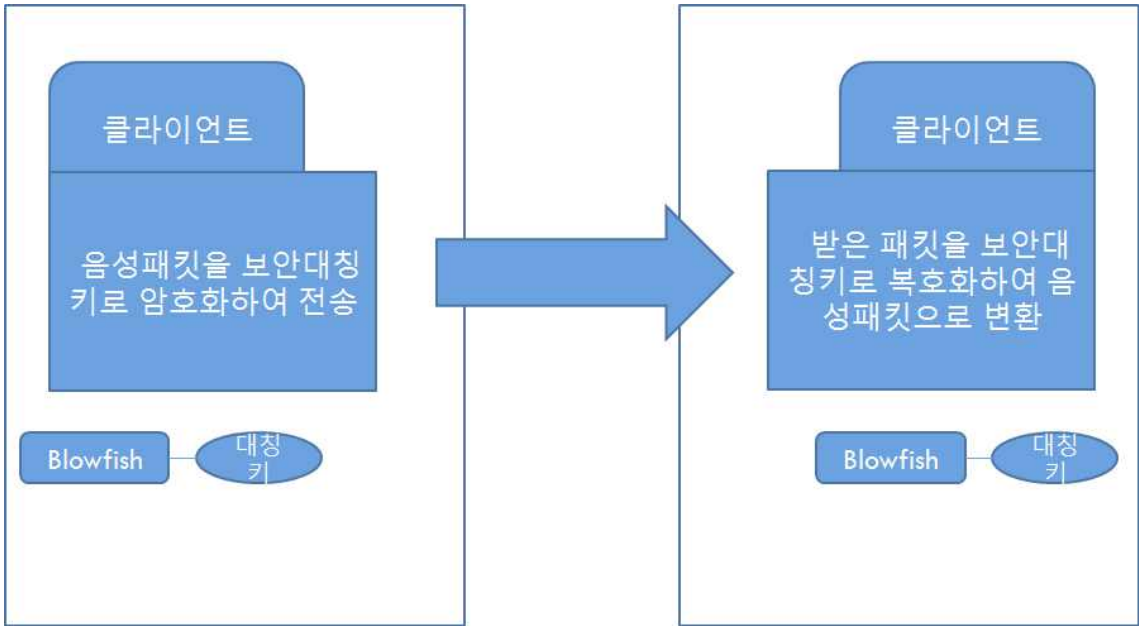
○ 보안



- 각 클라이언트는 Openssl를 통해 공개키와 비밀키를 갖는다.
- 연결을 요청하는 클라이언트는 상대 클라이언트에 연결을 요청할 때 생성된 자신의 공개키를 같이 전송한다.
- 연결 요청 받은 클라이언트는 Blowfish 알고리즘을 이용하여 생성된 대칭키를 전송받은 상대의 공개키로 암호화 한다.
- 연결 요청 받은 클라이언트는 상대의 공개키로 암호화 된 보안 대칭키를 자신의 비밀키로 다시한번 암호화 한다.
- 연결 요청 받은 클라이언트는 암호화된 대칭키와 자신의 공개키를 연결요청한 클라이언트에 전송한다.



- 연결 요청한 클라이언트는 전송받은 암호화된 대칭키를 상대의 공개키로 복호화한다.
- 연결 요청한 클라이언트는 상대의 공개키로 복호화 된 대칭키를 다시한번 자신의 비밀키로 복호화한다.
- 연결 요청한 클라이언트는 완전히 복호화된 보안대칭키를 저장한다.
- 결과적으로 각 클라이언트는 서로 같은 보안대칭키를 갖게된다.



- 클라이언트는 통화시에 음성패킷을 보안대칭키로 암호화하여 상대 클라이언트에 전송한다.
- 암호화된 패킷을 받은 클라이언트는 받은 패킷을 보안대칭키로 복호화하여 음성패킷으로 변환한다.
- 위의 과정은 클라이언트간의 통화가 이루어질 때 일어나는 과정이다.

서버 데이터베이스 스키마

```
create table kfvp_info (
uid int(7) unsigned NOT NULL auto_increment,
id varchar(20) default NULL,
password varchar(32) default NULL,
name varchar(100),
phonenumber varchar(100),
ip varchar(100),
port int,
is_login varchar(2) default 'N',
PRIMARY KEY (`uid`)
);
```

열이름	데이터형식	NULL	내용
uid	int(7)	NOT NULL	PK
id	varchar(20)	default NULL	아이디
password	varchar(32)	default NULL	비밀번호
name	var(100)		이름
phonenumber	varchar(100)		전화번호
ip	var(100)		IP Address
port	int		포트
is_login	varchar(2)		접속여부 default 'N'

소스코드 오버뷰

서버 (php 각 소스로 설명)

○ sign_req.php – 인증번호 제공 관련

```
<?
include_once "db.inc";
session_start();
$id=$_GET['id'];
$password=$_GET['password'];
$query="select * from kfvp_info where id='$id' and password='$password' limit 1";
$r=$dbc->query($query);
if ($dbc->num_rows($r)!=1) {
        echo "NotAuth";
        exit; }
$d=$dbc->fetch_array($r);
$cn=$d[phonenumber];
$key=substr(md5(uniqid(time())),5,10);
$_SESSION['pin']=$key;
$_SESSION['cn']=$cn;
send_mphone("010-1234-1004",$cn,"인증키 [$key]");
echo "SUCCESS";
function send_mphone($sendno,$recvno,$msg)
        {
                $sp=explode("-",$sendno);
                $sms_url = "http://sslsms.cafe24.com/sms_sender.php"; //전송요청 URL
           $sms['user_id'] = base64_encode("smsaeris"); //SMS 아이디.
     $sms['secure'] = base64_encode("3ae3e60b525c6d7afa727da9383c4e6a") ;//인증키
     $sms['msg'] = base64_encode(iconv('utf-8','euc-kr',$msg));
     $sms['rphone'] = base64_encode($recvno);
     $sms['sphone1'] = base64_encode($sp[0]);
     $sms['sphone2'] = base64_encode($sp[1]);
```

```
    $sms['sphone3'] = base64_encode($sp[2]);
    $sms['mode'] = base64_encode("1"); // base64 사용시 반드시 모드 값을 1로 주셔야 합니다.
    $host_info = explode("/", $sms_url);
    $host = $host_info[2];
    $path = $host_info[3]."/".$host_info[4];
    srand((double)microtime()*1000000);
    $boundary = "----------".substr(md5(rand(0,32000)),0,10);
    //print_r($sms);
    // 헤더 생성
    $header = "POST /".$path ." HTTP/1.0₩r₩n";
    $header .= "Host: ".$host."₩r₩n";
    $header .= "Content-type: multipart/form-data, boundary=".$boundary."₩r₩n";
    // 본문 생성
    foreach($sms AS $index => $value){
        $data .="--$boundary₩r₩n";
        $data .= "Content-Disposition: form-data; name=₩"".$index."₩"₩r₩n";
        $data .= "₩r₩n".$value."₩r₩n";
        $data .="--$boundary₩r₩n";    }
    $header .= "Content-length: " . strlen($data) . "₩r₩n₩r₩n";
    $fp = fsockopen($host, 80);
    if ($fp) {
        fputs($fp, $header.$data);
        $rsp = '';
        while(!feof($fp)) {
            $rsp .= fgets($fp,8192);
}
        fclose($fp);
            }
       }
?>
```

데이터베이스에서 ID 와 비밀번호 정보를 얻어옴

인증번호 생성 send_mphone($sendno, $recvno, $msg) - sms를 통한 인증번호 제공함수

○ sign_req.php – csr 검증 및 인증서 검사

```
<?
session_start();
$cn=$_SESSION['cn'];
$pin=$_SESSION['pin'];
$csr=$_POST['csr'];
if ($pin=='') {
        echo "error:먼저 인증번호를 받으십시오";
        exit; }//csr 검증
$ret=openssl_csr_get_subject($csr);

if ($ret['CN']!=$cn) {
        echo "error:CSR의 cn과 등록된 휴대폰번호가 틀립니다";
        exit; }
if ($_GET['pin']!=$pin) {
        echo "error:인증번호가 틀립니다.";
        exit; }
$home="/home/hosting_users/certplan56/kfvp";
$fp=fopen($home."/$cn.csr","wb");
fwrite($fp,$csr);
fclose($fp);
$cmd="openssl ca -batch -in $home/$cn.csr -out $home/$cn.crt -config $home/openssl.cnf";
system($cmd);
$fp=fopen($home."/$cn.crt","rb");
echo fread($fp,8192);
fclose($fp);
?>
```

csr 검증 과 등록된 휴대번호와 비교 인증번호를 확인 인증서 발급

○ reg.php – 클라이언트 접속 IP와 포트 등록

```
<?
include_once "db.inc";
$id=$_GET['id'];
$password=$_GET['password'];
$islocal=$_GET['islocal'];
$port=$_GET['port'];
$query="select * from kfvp_info where id='$id' and password='$password' limit 1";
$r=$dbc->query($query);
if ($dbc->num_rows($r)!=1) {
        echo "NotAuth";
        exit;
}
$d=$dbc->fetch_array($r);
$phone=$d[phonenumber];

if ($islocal==1) $Ip=$_GET['ip'];
else $Ip=$REMOTE_ADDR;
$query="update kfvp_info set ip='$Ip',
port=$port,
is_login='Y' where id='$id' and password='$password'";
$dbc->query($query);
echo $phone;
?>
```

데이터베이스에 접속
클라이언트의 IP와 포트를 데이터베이스에 등록함
is_login을 Y로 바꿈

○ logout.php – 로그아웃 시 동작

```
<?
include_once "db.inc";
$id=$_GET['id'];
$password=$_GET['password'];
$port=$_GET['port'];
$query="select * from kfvp_info where id='$id' and password='$password' limit 1";
$r=$dbc->query($query);
if ($dbc->num_rows($r)!=1) {
        echo "NotAuth";
        exit;
}
$d=$dbc->fetch_array($r);
$query="update kfvp_info set
is_login='N' where id='$id' and password='$password'";
$dbc->query($query);
echo "Success";
?>
```

is_login을 N으로 바꿈

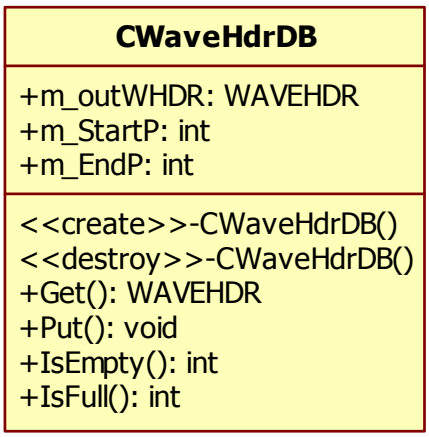
○ get.php – 클라이언트의 상대 클라이언트 IP 요청에 관련

```
<?
include_once "db.inc";
$id=$_GET['id'];
$phone=$_GET['phonecon_number'];
$query="select * from kfvp_info where phonenumber='$phone' limit 1";
$r=$dbc->query($query);
if ($dbc->num_rows($r)!=1) {
        echo "NotReg";
        exit;
```

```
}
$d=$dbc->fetch_array($r);
if ($d['is_login']!='Y') {
        echo "NotLogin";
        exit;
}
echo "OK:$d[ip]:$d[port]";
?>
```

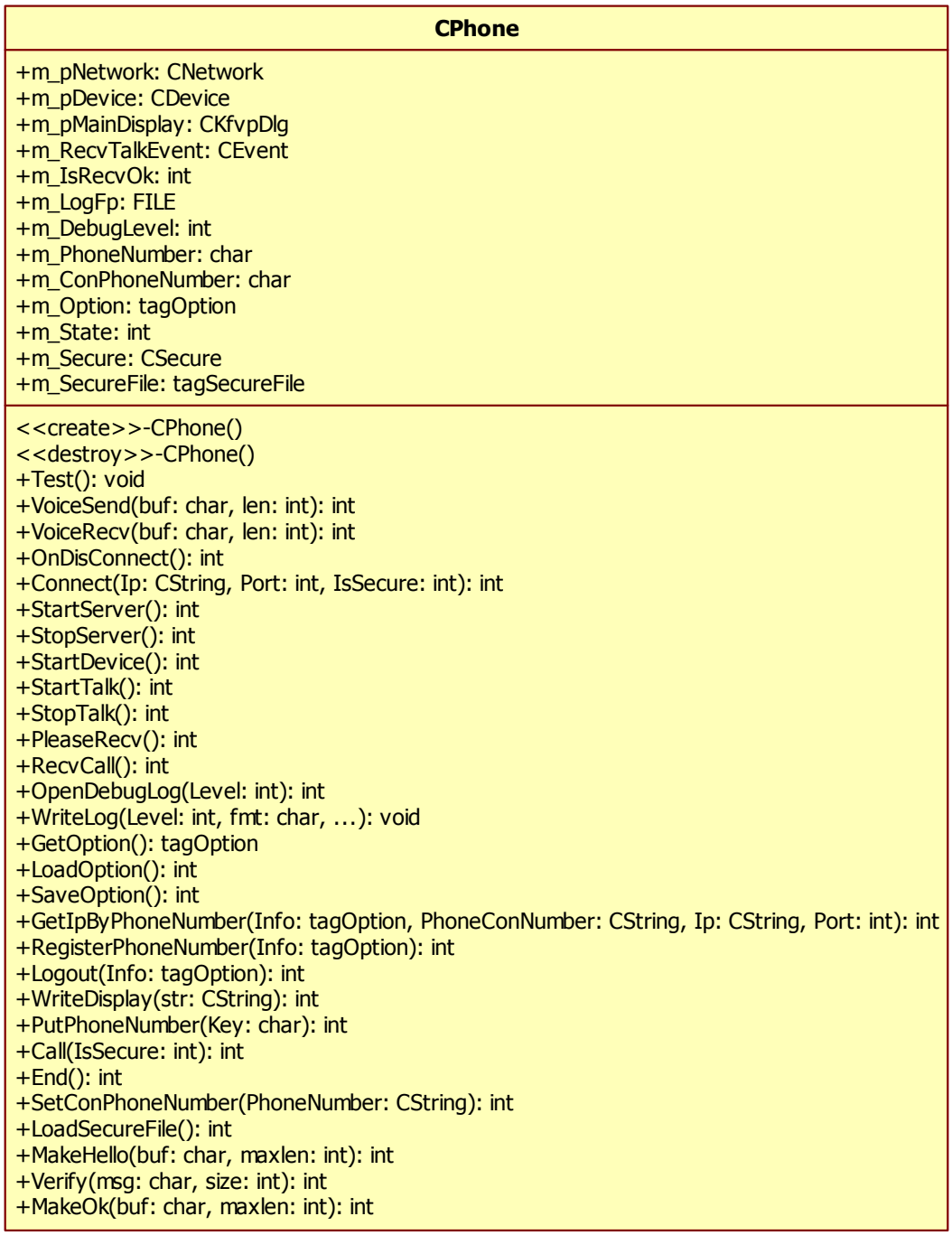
ip 요청시 is_login 검사 후 ip와 포트를 전송

클라이언트



○ CWaveHdrD -스피커 출력 시 필요한 정보인 웨이브 헤더 자료구조를 큐형식으로 구현함

m_outWHDR – 헤더 버퍼 M_StartP – 큐의 시작 포인트 M_EndP – 큐의 끝 포인트
Get – 비어있는 웨이브 헤더를 가져온다. Put – 사용한 웨이브 헤더를 반환한다. IsEmpty – 웨이브 헤더가 전부 비어있는지 확인한다. IsFull - 웨이브 헤더가 전부 사용중인지 확인한다.



○ CPhone - 통화에 필요한 기능들을 담당하는 클래스 CDevice 와 CNetwork 클래스를 중계하는 역할

m_pNetwork - CNetwork 클래스의 인스턴스
m_pDevice - CDevice 클래스의 인스턴스

m_pMainDisplay - CKfvpdlg 클래스의 인스턴스 m_RecvTalkEvent - 동기화를 위해 설정하는 이벤트 변수 m_IsRecvOk - 전화를 받았는지 확인하는 변수 0 : 받지않음 1 : 받음 m_LogFp - 로그파일의 파일 포인터 m_PhoneNumber - 자신의 전화번호 m_ConPhoneNumber - 상대의 전화번호 m_Option - 환경설정 변수 m_State - 0: 대기중 1:거는중 2: 받는중 3: 통화중 m_Secure - 현재 보안통신중인지 여부 m_SecureFile – 키파일
VoiceSend - 음성패킷을 네트워크를 통해 상대방에게 보낸다. VoiceRecv - 음성패킷을 네트워크를 통해 상대방으로부터 받는다. OnDisConnect - 연결이 끊어질시 호출되는 함수 Connect - 통화할 상대와 연결한다. StartServer - 통화수신시 필요한 서버를 시작한다. StopServer - 통화수신시 필요한 서버를 정지한다. StartDevice - 마이크 및 스피커의 작동을 시작한다. (API 사용) StartTalk - 실제 통화를 시작한다. StopTalk - 실제 통화를 중지한다. PleaseRecv - 상대방이 전화를 받을 때까지 기다린다. RecvCall - 상대방의 전화를 받는다. OpenDebuglog - 디버그 정보를 생성한다. WriteLog - 로그를 작성한다. GetOption - 환경설정 정보를 가져온다. LoadOption - 환경설정을 파일에서 불러온다. SaveOption - 환경설정을 파일로 저장한다. GetIpByPhoneNumber - 상대방의 IP를 서버로부터 받아온다. RegisterPhoneNumber - 자신의 IP를 서버에 등록한다.

LogOut - 서버에서 로그아웃을 한다.

WriteDisplay - 통화상태에 대한 정보를 출력한다.

PutPhoneNumber - 전화할 상대방의 전화번호를 사용자에게서 입력받는다.

Call - Connect 및 기타 통화에 필요한 함수들을 호출한다. (통화를 하는 궁극적 함수)

End - 통화를 종료한다. (다음 통화를 위한 준비 과정 포함)

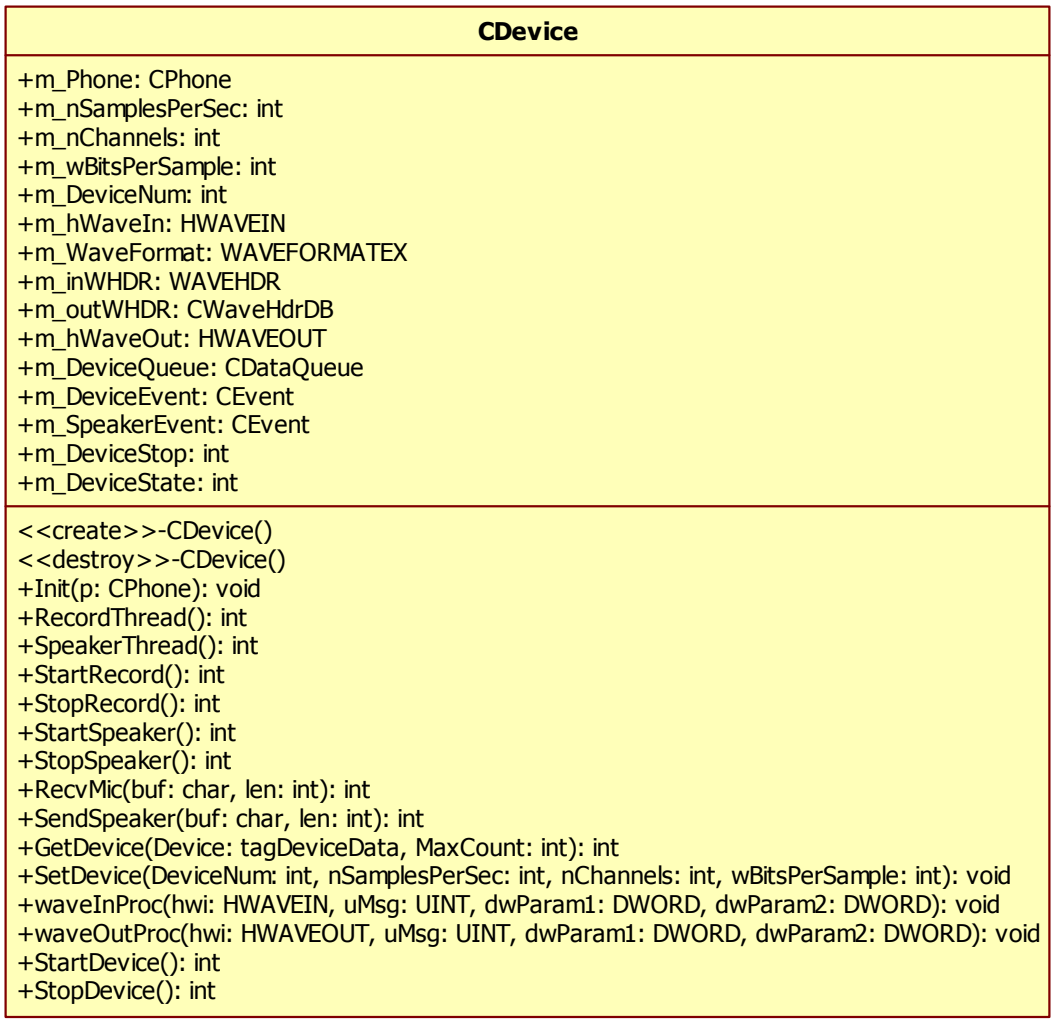
SetConPhoneNumber - 전화할 상대방의 번호를 변수에 입력한다.

LoadSecureFile - 보안통신을 위한 인증서 및 암호키를 메모리로 불러온다.

MakeHello - 보안통신에서 인증을 위해 상대방에게 메시지를 전송한다.

Verify - 상대방이 올바른 사용자를 확인한다. (인증서를 통한 확인)

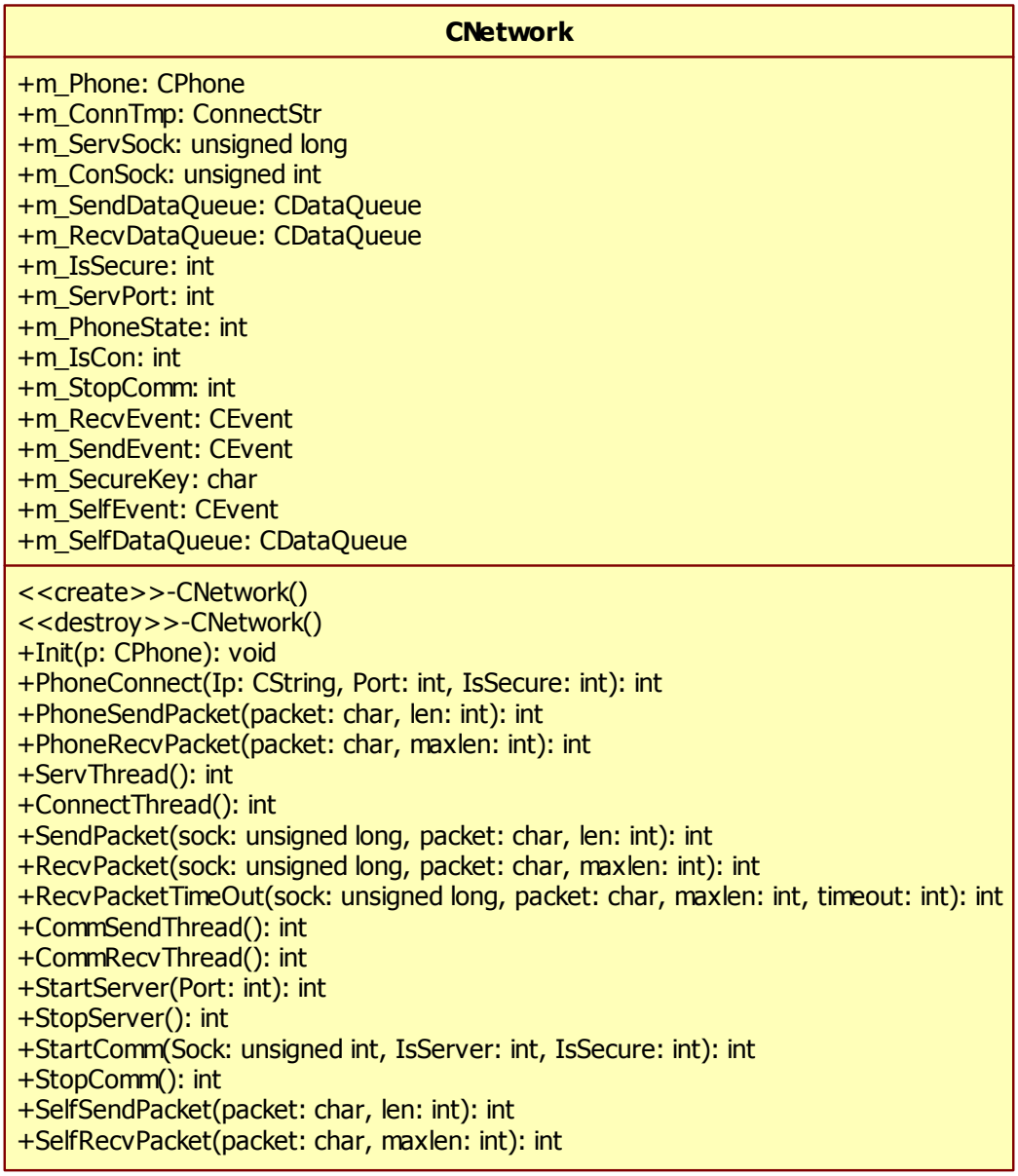
MakeOk - 상대방에게 인증이 되었다는 메시지를 전송한다.



○ CDevice – 마이크 및 스피커 장치를 관리하고 받은 음성을 음성정보로 변환하는 기능을 가진 함수

m_Phone - CPhone의 인스턴스
m_nSamplesPerSec - 음성의 초당 샘플링 수
m_nChannels - 음성의 채널수
m_wBitsPerSample - 샘플당 비트수
m_DeviceNum - 마이크의 디바이스 번호
m_hWaveIn - 웨이브 녹음 핸들러
m_WaveFormat - 웨이브 포맷

m_inWHDR - 웨이브 녹음 헤더 m_outWHDR - 웨이브 출력 헤더 m_hWaveOut - 웨이브 출력 핸들러 m_DeviceQueue - 음성을 저장하는 자료구조(큐) m_DeviceEvent - 음성을 동기화 시키기 위한 이벤트 m_SpeakerEvent - 스피커 출력을 동기화 시키기 위한 이벤트 m_DeviceStop - 1 : 정지 시킴 m_DeviceState - 0 : 중지 1: 작동중
Init - 초기화 함수 RecordThread - 음성을 마이크로부터 받아들이는 쓰레드 SpeakerThread - 음성을 스피커로 출력하는 쓰레드 StartRecord - RecordThread를 실행한다. StopRecord - RecordThread를 정지한다. StartSpeaker - SpeakerThread를 실행한다. StopSpreker - SpeakerThread를 정지한다. RecvMic - 마이크로부터 음성데이터를 전달받는다. PCM형태로 전달받음 SendSpeaker - 스피커로 출력한다. GetDevice - 컴퓨터에 설치된 마이크 장치의 목록을 가져온다. SetDevice - 마이크 장치의 목록중 사용할 장치를 설정한다. waveInProc - 마이크 콜백함수 waveOutPrc - 마이크 콜백함수 StartDevice - 마이크 장치 동작 StopDevice - 마이크 장치 동작 정지



○ CNetwork – 네트워크 관련 리시버와 커넥터 쓰레드를 생성하고 동작하게 하는 클래스

m_Phone - CPhone의 인스턴스
m_ConnTmp - 연결요청시
m_ServSock - 서버 소켓 핸들러
m_ConSock - 커넥터 소켓 핸들러

m_SendDataQueue - 데이터 전송 큐

m_RecvDataQueue - 데이터 수신 큐

m_IsSecure - 0: 아님 1:보안모드

m_ServPort - 서버 포트

m_PhoneState - 0:서버리스닝중 1: 연결중 2: 종료

m_IsCon - 0:연결안됨 1:연결됨

m_StopComm - 1:스톱

m_RecvEvent - 네트워크 수신시 동기화 이벤트

m_SendEvent - 네트워크 송신시 동기화 이벤트

m_SecureKey - 보안키

m_SelfEvent - 테스트용 이벤트

m_SelfDataQueue - 테스트용 자료구조(큐)

Init - 초기화 함수

PhoneConnect - 연결할 상대의 아이피 정보와 보안통신 여부를 확인후 연결을 하는 함수

PhoneSendPacket - SendPacket을 이용하여 음성 패킷을 발신한다.

PhoneRecvPacket - RecvPacket을 이용하여 음성 패킷을 수신한다.

ServThread - 리시버 쓰레드로 생성한다.

ConnectThread - 커넥터 쓰레드를 생성한다.

SendPacket - 음성 패킷을 네트워크로 보낸다.

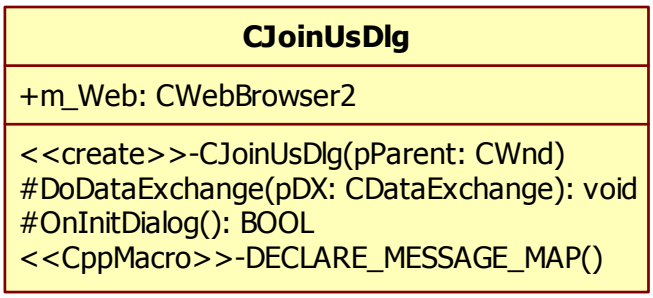
RecvPacket - 네트워크로부터 음성 패킷을 받아온다.

RecvPacketTimeOut - 음성패킷 수신에 대기시간을 갖는 함수로 일정시간이 지나면 연결을 끊는 함수

CommSendThread - 통화시 송신에 쓰이는 쓰레드를 생성한다. 음성정보를 패킷 단위로 잘라 송신한다. ServThread는 다른 통화를 대비해 그대로 대기한다.

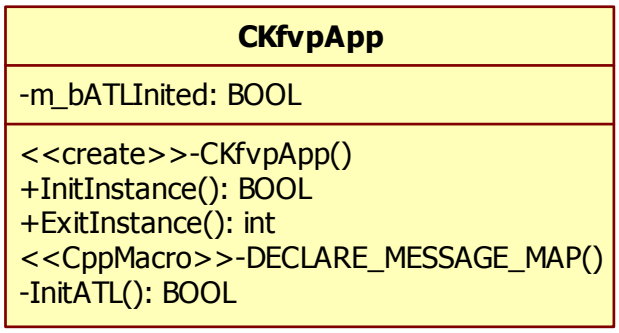
CommRecvThread - 통화시 수신에 쓰이는 쓰레드를 생성한다. 잘라진 음성 패킷을 다시 합친다. RecvThread는 다른 통화를 대비해 그대로

대기한다. StartServer - 통화를 받을 때 StopServer - 통화를 종료 할 때 Startcomm - 보안을 점검하고 CommSendThread와 CommRecvThread를 생성하여 통화를 가능하게 한다. Stopcomm - Startcomm 종료 SelfSendPacket - 에코 테스트용 SelfRecvPacket - 에코 테스트용



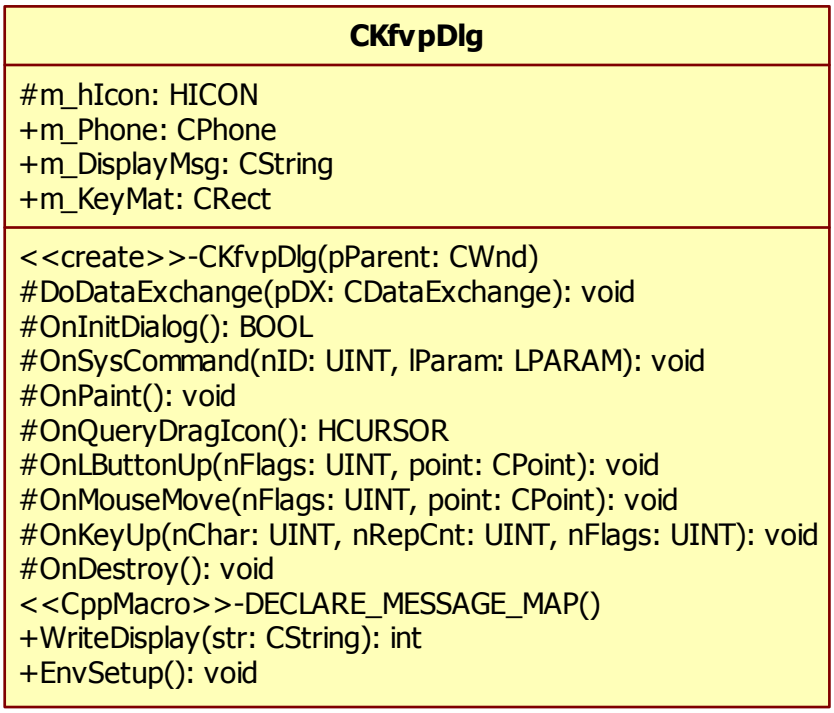
○ CJoinUsDlg – 회원가입 창을 띄우는 기능을 가짐

m_Web - 웹 브라우져 객체
DoDataExchange - MFC 기본 함수 OnInitDialog – 초기화함수



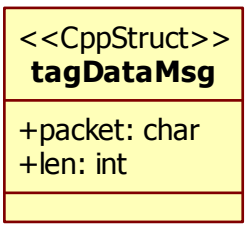
○ CKfvpAPP – 어플리케이션 기본 창에 대한 클래스

m_bATLInited - ATL 관련 변수
InitInstance - 인스턴스 초기화 ExitInstance - 인스턴스 종료 InitATL - ATL초기화



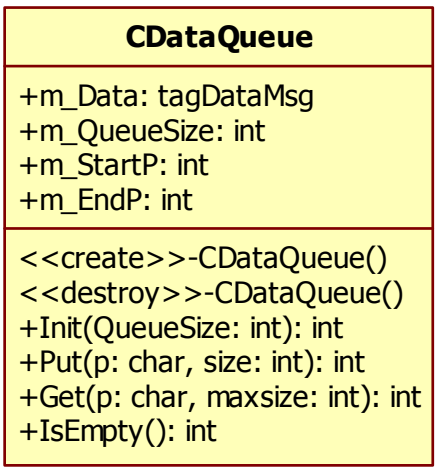
○ CKfvpDlg – 어플리케이션 기본창의 GUI 에 대한 클래스

m_hIcon - 아이콘 핸들러 m_Phone - CPhone의 인스턴스 m_DisplayMsg - 출력 메시지 m_KeyMat - 입력할 숫자 배열
DoDataExchange - MFC 기본 함수 OnInitDialog - 다이얼로그 초기화 OnSysCommand - 시스템메뉴 처리 OnPaint - 배경 BMP 파일 띄우기 OnQueryDragIcon - 윈도우즈 기본 기능 버튼 처리 OnLButtonUp - 마우스 버튼 클릭 처리 OnMouseMove - 마우스 동작 OnKeyUp - 키보드 동작 OnDestroy - 창 종료시 수행 WriteDisplay - 액정 (메시지 출력 부분) EnvSetup - 환경설정 창을 띄우는 버튼



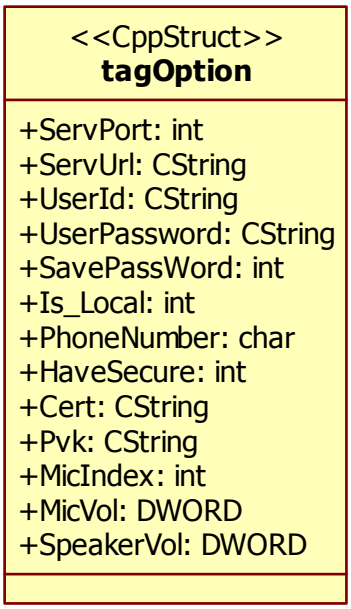
○ tagDataMsg – 디바이스와 네트워크 간의 이동하는 데이터 구조

packet – 패킷 len - 패킷의 길이



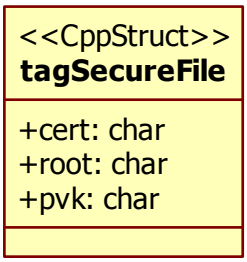
○ CDataQueue – 디바이스와 네트워크 간의 이동하는 데이터를 큐로 랩핑

m_Data – 데이터 m_QueueSize - 큐의 크기 m_StartP - 큐의 시작점 m_EndP - 큐의 끝점
Init - 초기화 Put - 데이터 반환 Get - 비어있는 큐를 가져옴 IsEmpty - 큐가 전부 비어있는지 확인한다.



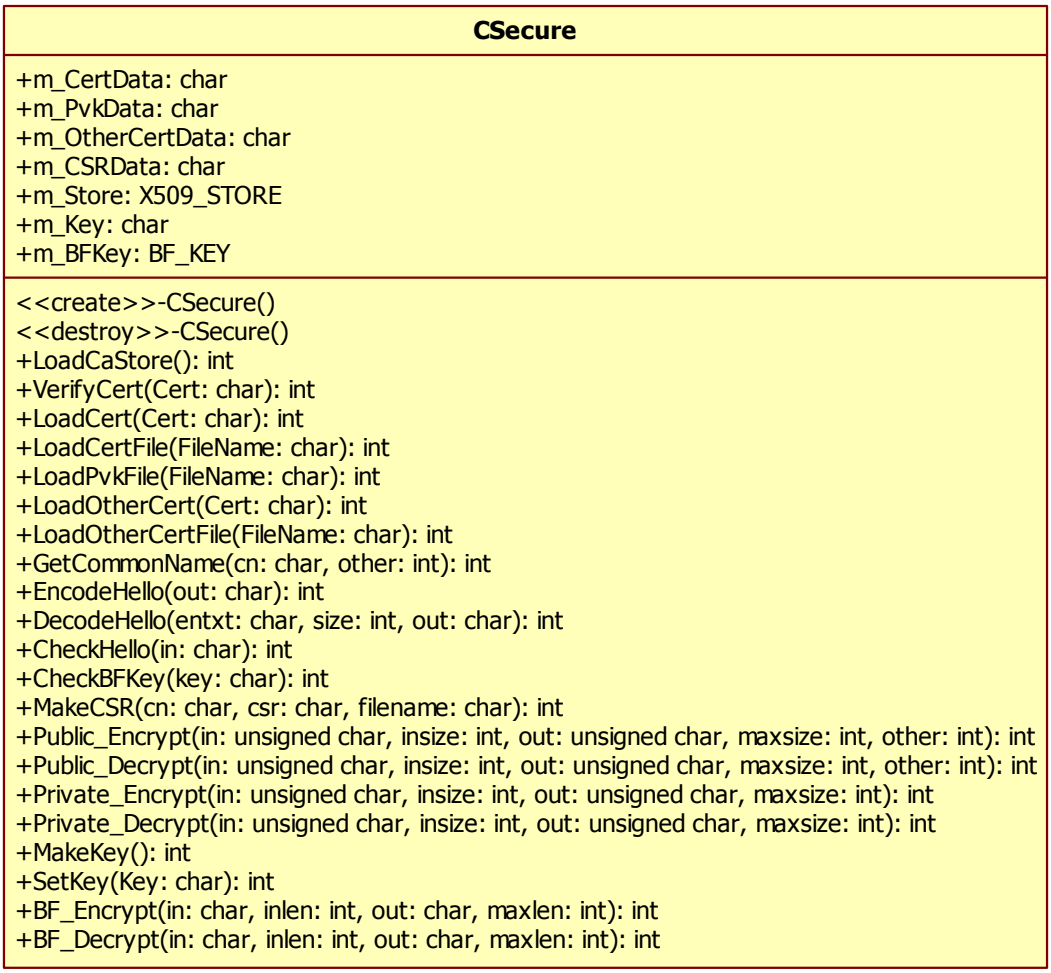
○ tagOption – 사용자 정보와 환경설정 정보를 저장하는 구조체

ServPort - 서버 포트 ServUrl - 서버 URL UserId - 사용자 ID UserPassword - 사용자 암호 SavePassword - 비밀번호 저장 여부 0: 저장안함 1 저장함 Is_Local - 같은 네트워크 내에서의 통신여부 PhoneNumber – 전화번호 HaveSecure - 보안통신 여부 Cert – 인증서 Pvk – 개인키 MicIndex - 마이크 장치 번호 MicVol - 마이크 볼륨 SpeakerVol - 스피커 볼륨



○ tagSecureFile – 인증서 와 키파일을 저장하는 구조체

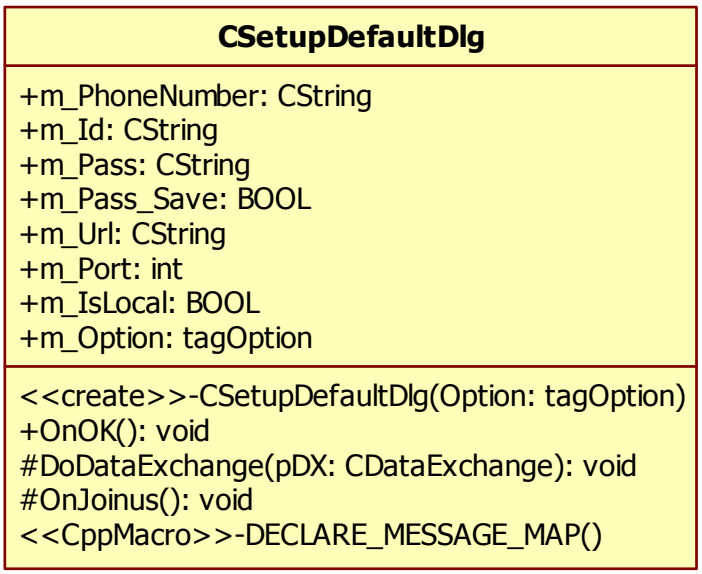
cert – 인증서 root – 루트 인증서 pvk – 개인키



○ CSecure – 보안관련 클래스

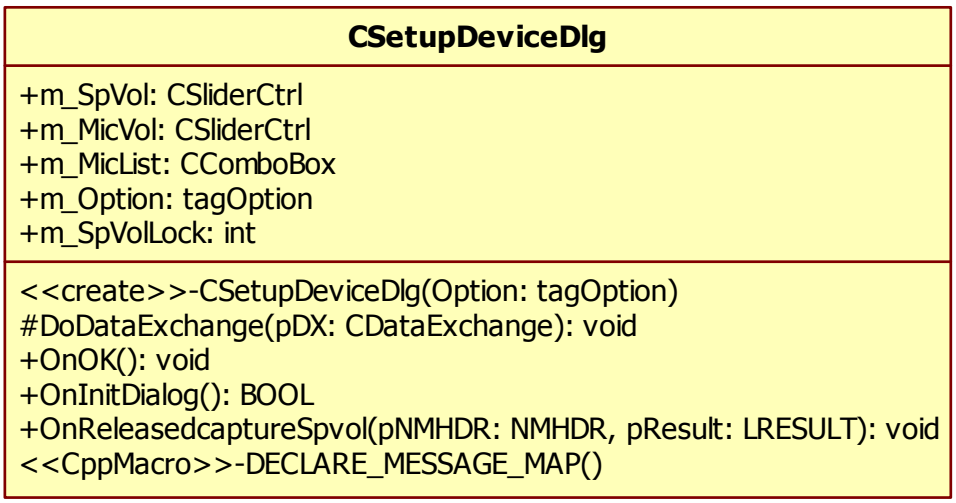
m_CertData - 인증서 데이터 m_PvkData - 비밀키 데이터 m_OtherCertData - 상대 인증서 데이터 m_CSRData - CSR 데이터 m_Store - 인증서 저장영역 m_Key – 대칭키 m_BfKey 보안대칭키 핸들러
LoadCaStore - 인증서를 저장할 Store 구조체 생성하고 파일로부터 CA인증서 읽어드린후 Store에 CA 인증서를 추가한다. VerifyCert - 인증서 인증 LoadCert - 메모리에 있는 상대방의 인증서를 openssl에 올리는 기능 LoadCertFile - 자신의 인증서 파일을 로드함 LoadPvkFile - 자신의 키파일을 로드함 LoadOtherCert - 테스트용 LoadCert LoadOtherCertFile - 테스트용 LoadCertFile GetCommonName - 인증서에서 전화번호를 가져옴 EncodeHello - 첫 연결시 전송되는 패킷을 암호화 DecodeHello - 첫 연결시 수신되는 패킷을 복호화 CheckHello - 복호화된 패킷을 확인 CheckBFKey - 대칭암호화키의 패스워드를 체크한다. MakeCSR - 자신의 전화번호로 CSR 파일을 생성한다. Public_Encrypt - 공개키로 암호화 Public_Decrypt - 공개키로 복호화 Private–Encrypt - 개인키로 암호화 Private Decrypt - 개인키로 복호화 MakeKey – 키생성 SetKey - 키설정

BF_Encrypt - 대칭암호화키로 암호화 BF_Decrypt - 대칭암호화키로 복호화



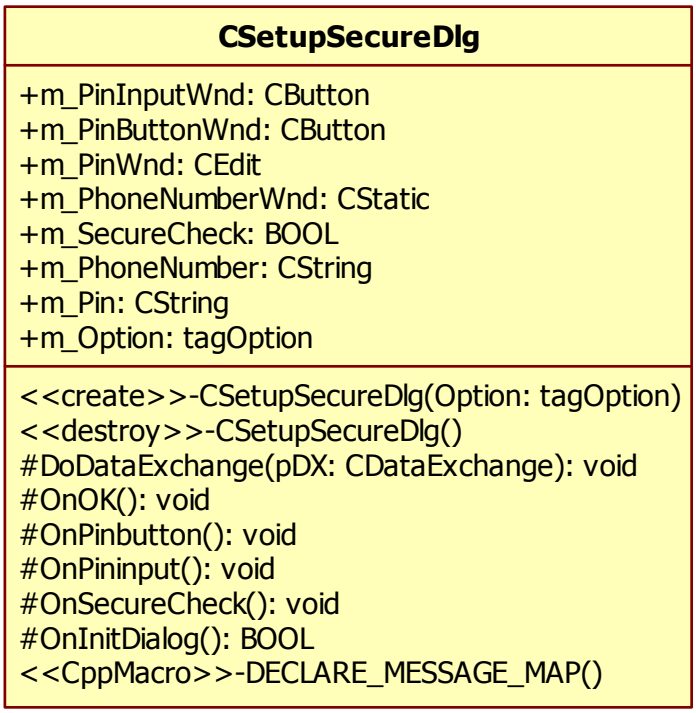
○CSetupDefaultDlg - 환경설정 및 아이디와 비밀번호 입력 창에 대한 클래스

m_PhoneNumber – 전화번호 m_Id – 아이디 m_Pass – 비밀번호 m_Pass_Save - 비밀번호 저장여부 m_Url - 서버 URL m_Port - 리시버 포트 m_IsLocal - 같은 네트워크 내에서의 통신 여부 m_Option - m_Option에 대한 포인터
OnOK - 확인 버튼 DoDataExchange - MFC 기본 함수 OnJoinus - 회원가입 버튼



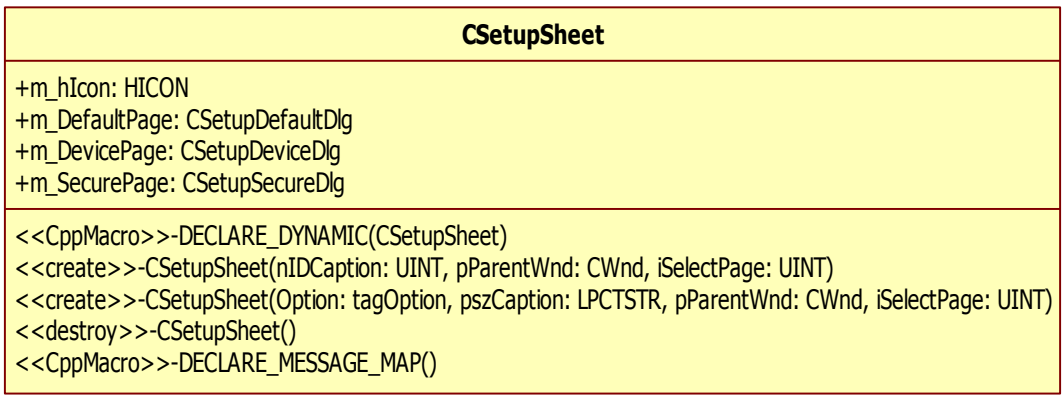
○CSetupDeviceDlg – 장치 설정 창에 대한 클래스

m_SpVol - 스피커 볼륨 m_MicVol - 마이크 볼륨 m_MicList - 마이크 장치 목록 m_Option - m_option 포인터 m_SpVolLock - 스피커 관련 변수
DoDataExchange - MFC 기본 함수 OnOK – 확인버튼 OnInitDialog - 다이얼로그 초기화 OnReleasecaptureSpvol - 스피커 볼륨 조절 바에 대한 함수



○ CSetupSecureDlg – 보안 설정 창에 대한 클래스

m_PinInputWnd - 인증번호 입력 버튼 m_PinButtonWnd - 인증번호 받기 버튼 m_PinWnd - 에디트 박스 m_PhoneNumberWnd - 전화번호 출력을 위한 핸들러 m_SecureCheck - 보안 통신 여부 m_PhoneNumber – 전화번호 m_Pin – 인증번호 m_Option - m_option에 대한 포인터
DoDataExchange - MFC 기본 함수 OnOK - 확인 버튼 OnPinbutton - 인증번호 요청버튼 동작 OnPininput - 인증번호 입력 동작 OnSecureCheck - 보안통신사용 체크 OnInitDialog - 다이얼로그 초기화



○ CSetupSheet – 환경설정 디폴트,디바이스,보안 관련 전체 시트 관련 클래스

m_hIcon - 아이콘 핸들러 m_DefaultPage - 기본 설정 m_DevicePage - 장치 설정 m_SecurePage - 보안 설정

시연

○ 프로그램 실행화면

- 마우스를 이용한 번호 입력이 가능함

- 키보드를 이용한 번호 입력이 가능함

- Talk 버튼은 일반 모드로 전화를 걸때의 버튼

- Secure Talk 버튼은 보안 모드로 전화를 걸때의 버튼

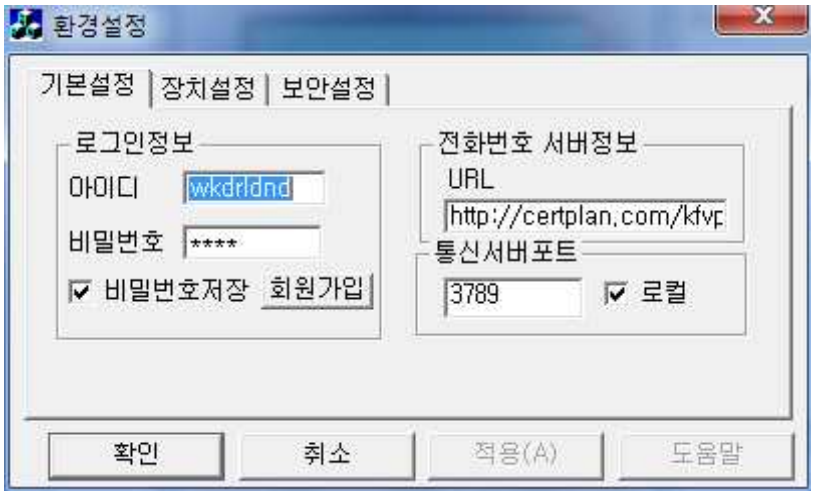
- Option 버튼은 설정 창을 열기위한 버튼

- Stop 버튼은 통화 종료시 이용되는 버튼

- 버튼들 위에 녹색 박스가 프로그램 수행시 메시지가 출력되는 부분

- 오른쪽 상단의 단추를 이용하여 최소화 및 프로그램 종료가 가능함

○ 설정 창 화면



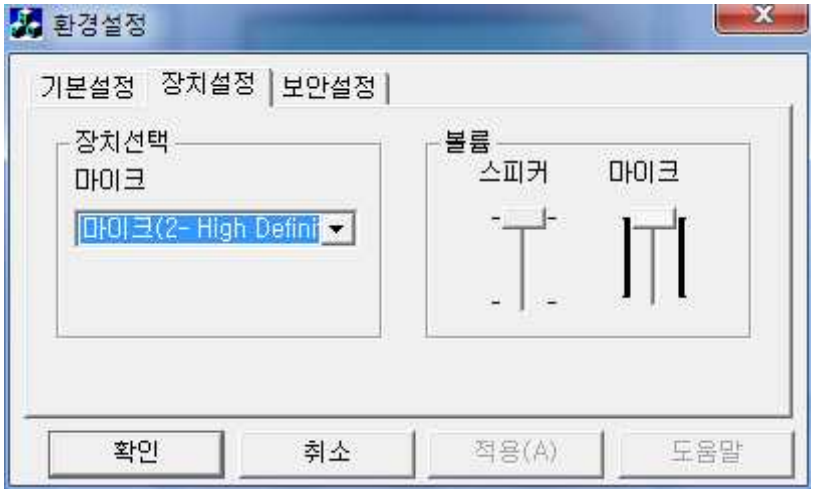
- Option 버튼을 누르면 환경 설정 창이 나오고 기본설정 탭이 선택되어져 나옴

- 로그인 정보에는 사용자의 아이디와 비밀번호를 입력할 수 있게 되어 있고 비밀번호저장 체크박스가 있어 매 실행시마다 비밀번호를 입력하지 않아도 됨

- 아이디가 없을시 회원가입 버튼을 눌러 회원가입 창을 열수가 있음

- 전화번호 서버정보는 서버 URL을 입력할 수 있으며 처음 실행시 기본값이 지정되어 있음

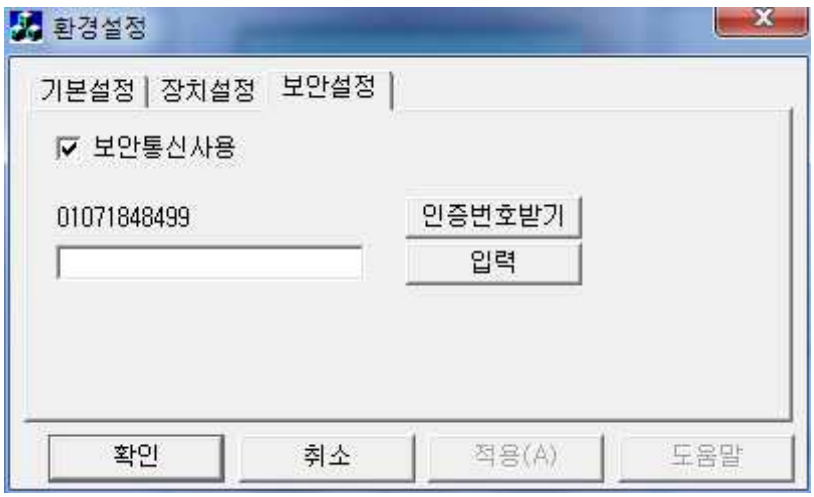
- 통신서버포트는 사용할 포트 번호를 입력할 수 있고 로컬체크는 같은 네트워크를 사용하고 있을 때 체크할 수 있도록 되어있음



- 환경 설정 창 중 장치설정 탭이 활성화 되었을 경우에 화면임

- 사용자의 컴퓨터에 설치된 마이크 장치 목록을 보여주고 선택할 수 있음.

- 볼륨란의 바를 조정하여 스피커의 볼륨을 조절할 수 있음



- 환경 설정 창 중 보안설정 탭이 활성화 되었을 경우에 화면임

- 보안통신사용 체크를 통해 보안통신모드를 사용할 수 있음

- 로그인한 자신의 전화번호를 확인할 수 있음

- 인증번호를 받아 사용자 인증을 할 수 있음

- 인증번호받기 버튼은 클릭 후 사용자의 핸드폰번호로 인증번호가 SMS로 전송됨

- 사용자가 인증번호를 입력후 입력 버튼은 누르면 사용자 인증이 완료됨

○ 통화 화면

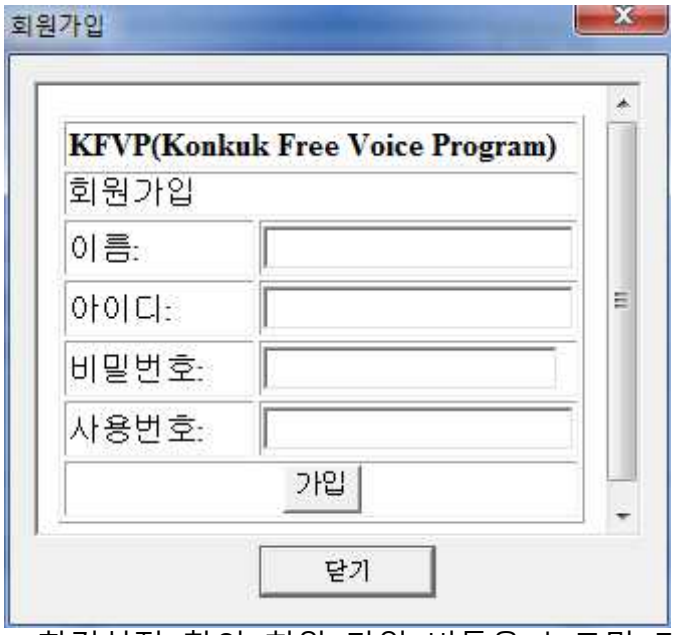
- 전화가 올시 전화온 상대의 전화번호가 뜨고 Talk/Secure Talk 버튼을 누르면 연결이 되었다는 메시지와 함께 연결됨

-

- 상대의 전화번호를 입력 후 Talk/Secure Talk를 누르면 연결중이 뜨면서 통화를 시도한다.

- Stop 버튼 클릭시 연결이 종료되었다는 메시지와 함께 통화가 종료된다.

○ 회원 가입 화면



- 환경설정 창의 회원 가입 버튼을 누르면 다음과 같은 창이 뜨면서 회원가입이 가능해진다.