

SECURE VIDEO CONFERENCING

Design Document

Team : Sapiens
Himanshu Karnatak, Ravibabu M., Mainak Garai

1. Introduction

Secure video conference is a centralized system to provide stream communication between two or more clients connected to a server. It consists of a server, a certification authority, and multiple clients. Clients will obtain certificate of identity from certification authority, and register themselves to the video conferencing server. Video conference server shall publish list of connected users, help establish connection and charge fee based on call length. Clients shall maintain user identity, related keys and provide high speed encryption of audio/video stream.

2. Purpose

Purpose of this document is to describe the top level architecture, design entities, and their interaction with each other.

3. Revision

Revision Number	Description	
1.0	Initial Revision	

4. Architecture

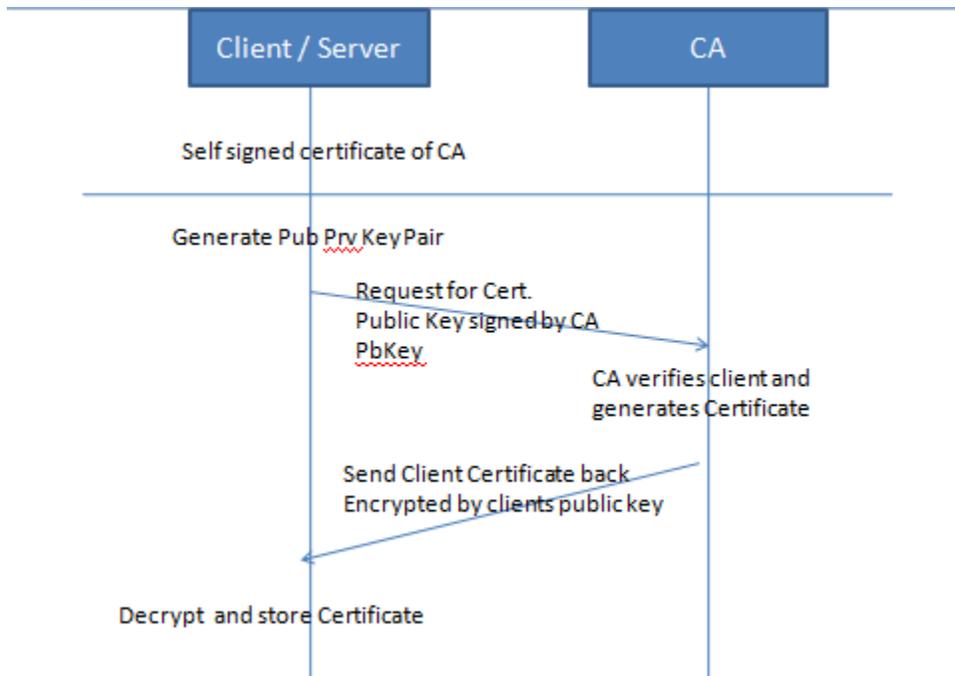
System shall be divided into three entities.

1. **Certification Authority** : CA is a trusted entity that will issue certificates to genuine clients and server based on their request. A self issued certificate of CA will be stored at all other entities. Certificates issued by this CA shall be accepted by all entities of the system.
2. **Video Conferencing Server** : It is a central entity, to help establish connection between clients. It shall verify the client before joining the system using their certificates. It will maintain a list of connected and registered clients and provide it when asked for. It will keep track of call lengths and other parameters to ensure proper billing to the service users.
3. **Client** : Client will connect to the server and register itself. It shall agree with the terms and conditions of use for it to be accepted at server. Once

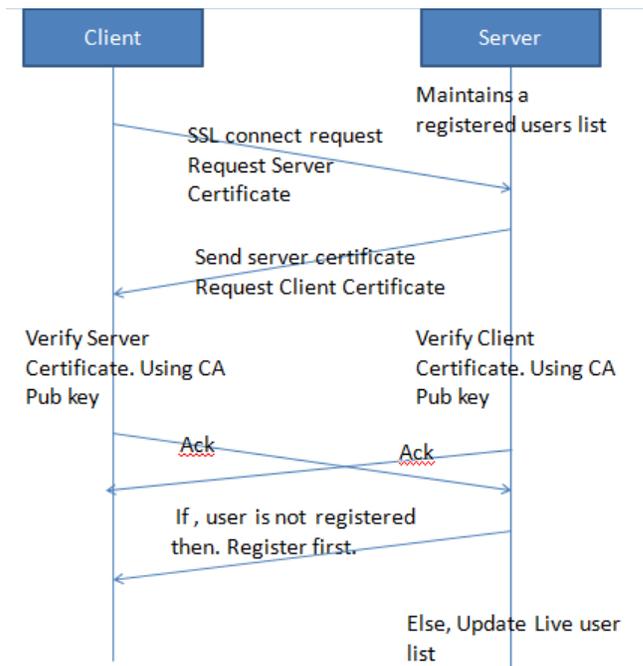
registered, client shall remain in server's registered user list. A client will come to active state when it is online with server. To initiate a communication, it will request a list of active users from the server and then ask to connect to one of them. An active client will listen to the server for any requests that may come. It will respond its willingness to the server for that request.

5. Connection Sequence

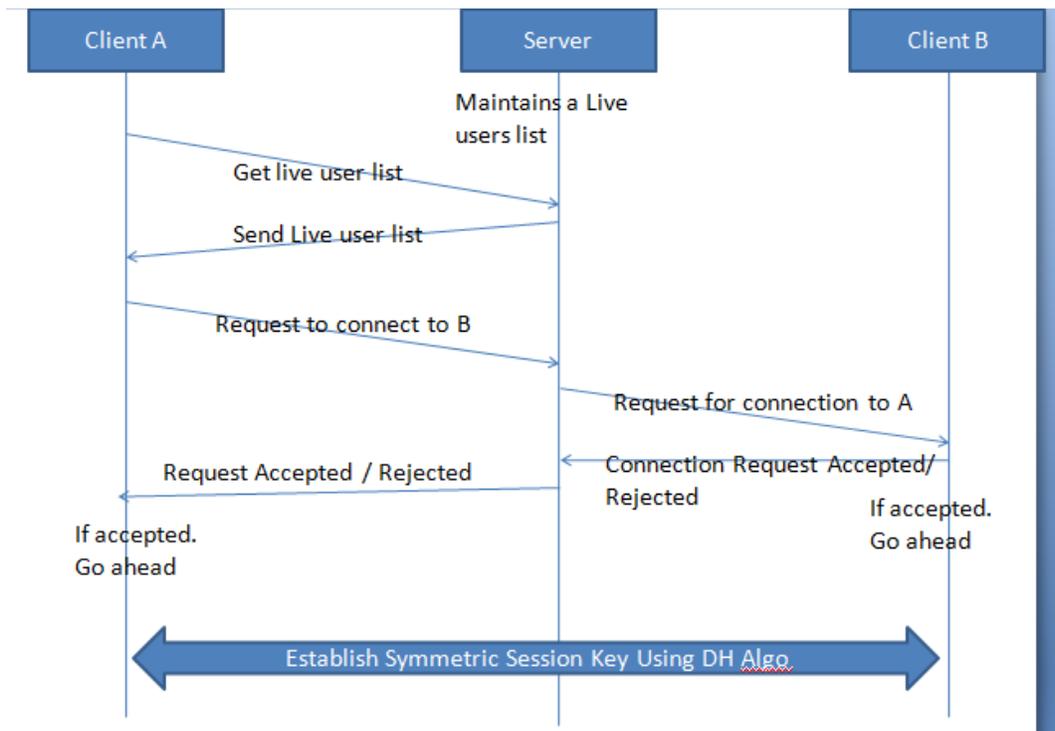
Issuing certificates



Communication between client and server.



Connection establishment between two clients



All messages represented by arrows shall be signed by the sender and encrypted by the public key of the receiver. Symmetric session key establishment will be implemented using client server conversations using asymmetric encryption. In the end it will lead to establishment of symmetric key between client A and client B.

When both A and B agrees to communicate, the initiator A will ask for the certificate of B from the server, server sends the certificate of client B. This certificate will be verified with the CA public key and extract the B's Public key from the certificate. Similarly client B will get the client A's certificate and extract the public key of A. Now both A and B has the asymmetric keys and they establish the symmetric key for the session.

6. APIs

Certification Authority:

API	Return Value	Parameter	Remarks
getNewCert()	Certificate, keyID	Public Key, ClientID	Generates a certificate to the request based on provided public key and id. And sends it back encrypted with requester's public key.
revokeCert()	Int	KeyID	Returns success or error.
updateCert()	Certificate, keyID	Public Key, KeyID, ClientID	Generates a new certificate to the request based on provided public key and id. And sends it back encrypted with requester's public key. It will revoke the obsolete certificate.

Video Conferencing Server:

API	Return Value	Parameter	Remarks
getServerCert())	Certificate		Returns server certificate.
registerNewUser())	Int	ClientID	Returns success or error.
updateCert())	Certificate, keyID	Public Key, KeyID, ClientID	Generates a new certificate to the request based on provided public key and id. And sends it back encrypted with requester's public key. It will revoke the obsolete certificate.
getActiveUsers())	User list	Client ID	Returns active user list
reqConnet())	Int	srcClientID, destClientID	Returns success or error status.
getAccountInfo())	Account info	ClientID	Returns clients account info.
establishSession())	-TBD		
addCredit())	Int	Credit info	
terminateSession())	Int		

closeConnection()			
unregister()	Int	ClientID	

Client

getClientCert()	Certificate		
acceptConnect() ()	Int	srcClientID	
establishSession() on()			
terminateSession() on()			