VideoChat

3조 보안연대
Team David Belasco

Project Members and Role

- Architect, Signaling, Database, Registration, Login
  Robin Kim(김성준), SeKi Park(박세기)

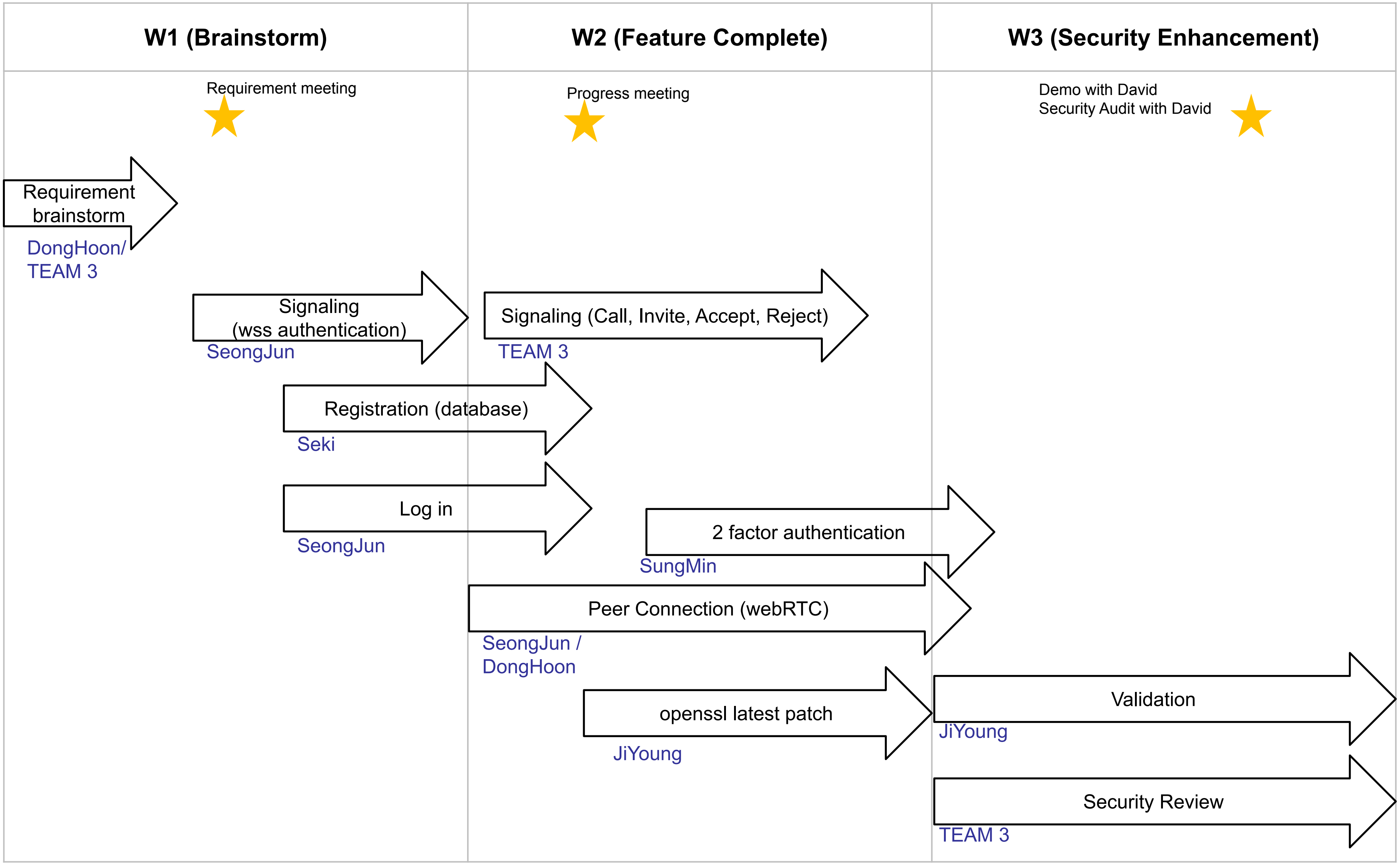
- 2 Factor authentication, Peer connection
  SungMin Kim(김성민), DongHoon Shin(신동훈)

- Test case and Validation
  JiYoung Yoon(윤지영)

- Mentor / Support
  David Belasco

Schedule

★ Meeting with David



W1 (Brainstorm)

W2 (Feature Complete)

W3 (Security Enhancement)

Requirement meeting ★

Progress meeting ★

Demo with David
Security Audit with David ★

Requirement brainstorm
DongHoon/
TEAM 3

Signaling
(wss authentication)
SeongJun

Signaling (Call, Invite, Accept, Reject)
TEAM 3

Registration (database)
Seki

Log in
SeongJun

2 factor authentication
SungMin

Peer Connection (webRTC)
SeongJun /
DongHoon

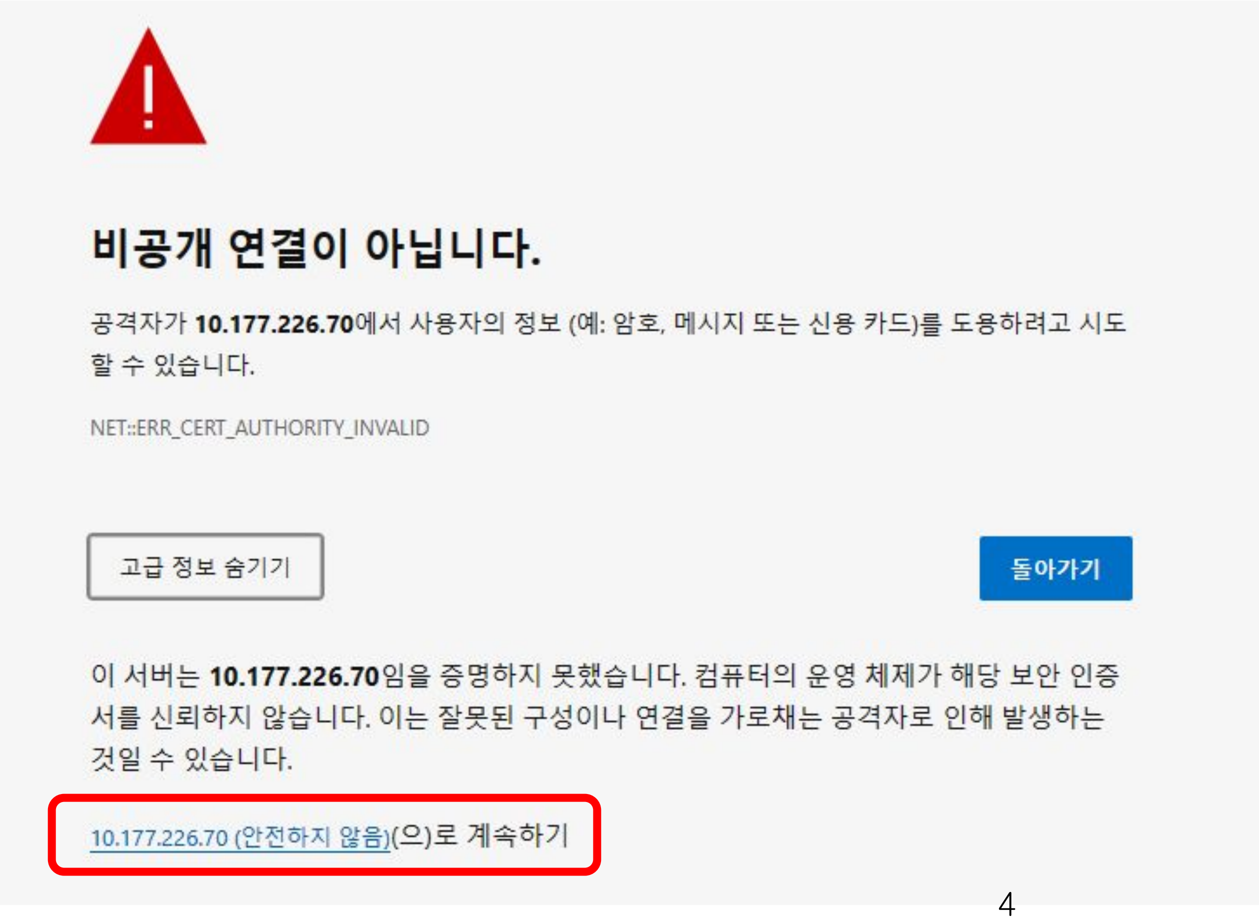
openssl latest patch
JiYoung

Validation
JiYoung

Security Review
TEAM 3

Demo

https://10.177.226.70



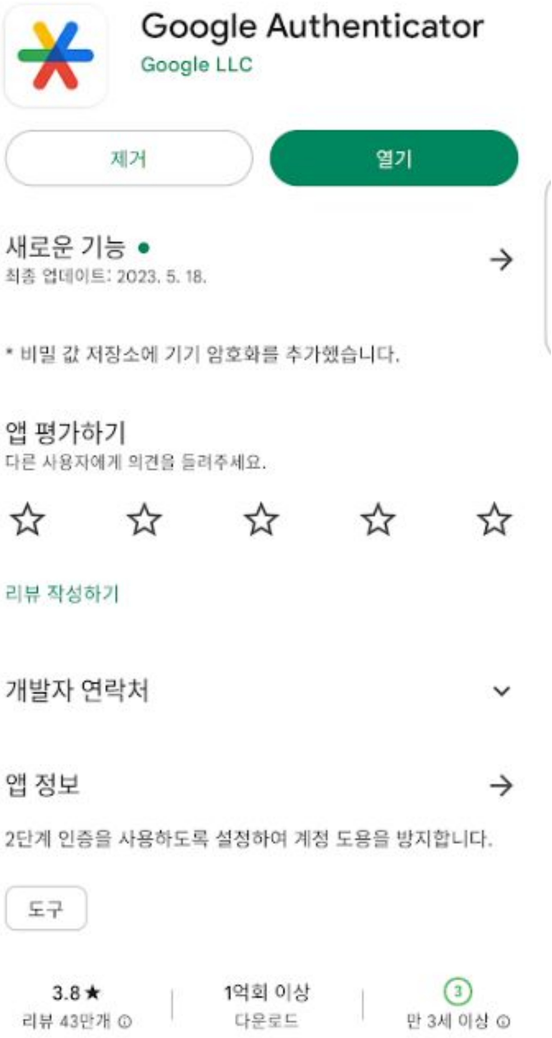
비공개 연결이 아닙니다.

공격자가 **10.177.226.70**에서 사용자의 정보 (예: 암호, 메시지 또는 신용 카드)를 도용하려고 시도할 수 있습니다.

NET::ERR_CERT_AUTHORITY_INVALID

이 서버는 **10.177.226.70**임을 증명하지 못했습니다. 컴퓨터의 운영 체제가 해당 보안 인증서를 신뢰하지 않습니다. 이는 잘못된 구성이나 연결을 가로채는 공격자로 인해 발생하는 것일 수 있습니다.

10.177.226.70 (안전하지 않음)(으)로 계속하기



Google Authenticator
Google LLC

새로운 기능 ●
최종 업데이트: 2023. 5. 18.

* 비밀 값 저장소에 기기 암호화를 추가했습니다.

앱 평가하기
다른 사용자에게 의견을 들려주세요.

☆ ☆ ☆ ☆ ☆

리뷰 작성하기

개발자 연락처

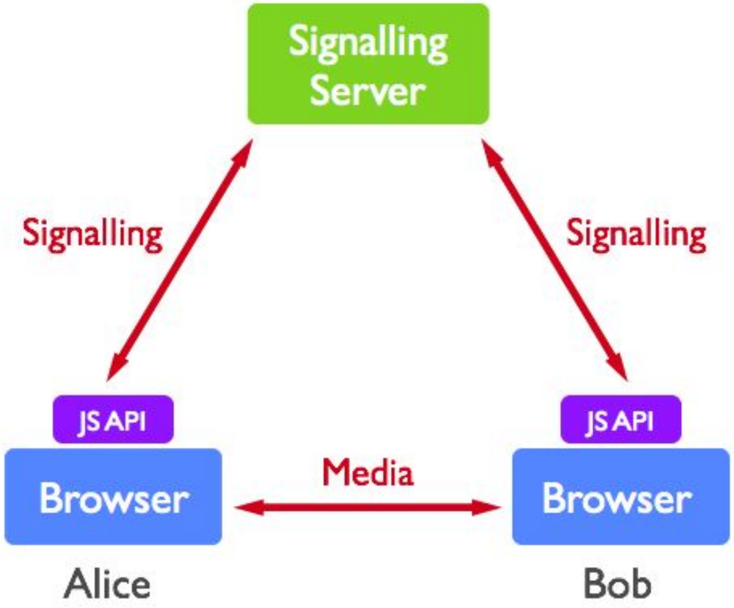
앱 정보

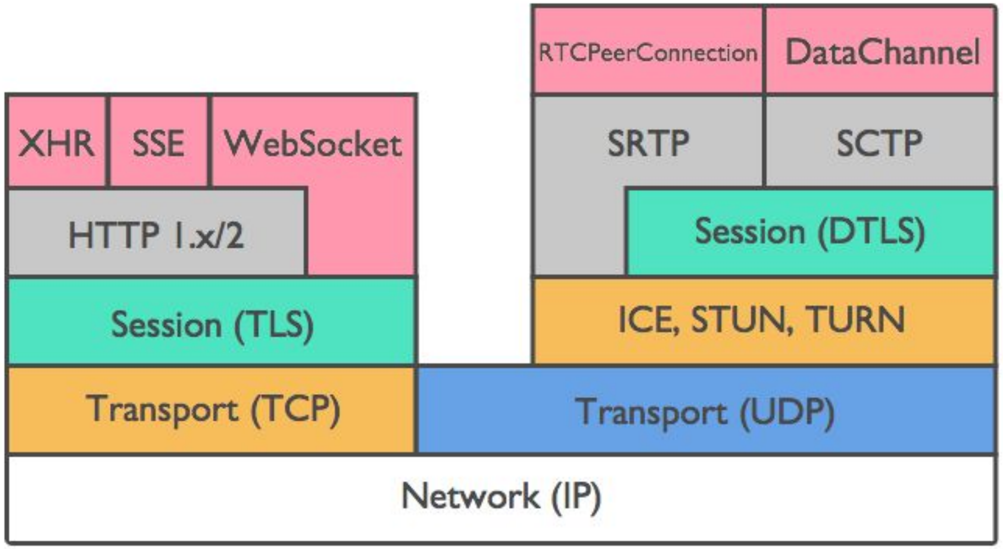
2단계 인증을 사용하도록 설정하여 계정 도용을 방지합니다.

3.8★	1억회 이상	3
리뷰 43만개 ⓘ	다운로드	만 3세 이상 ⓘ

Brainstorm / Background

- Harnessing collective intelligence to find best solution of the requirement
- WebRTC is perfectly matched for the requirement

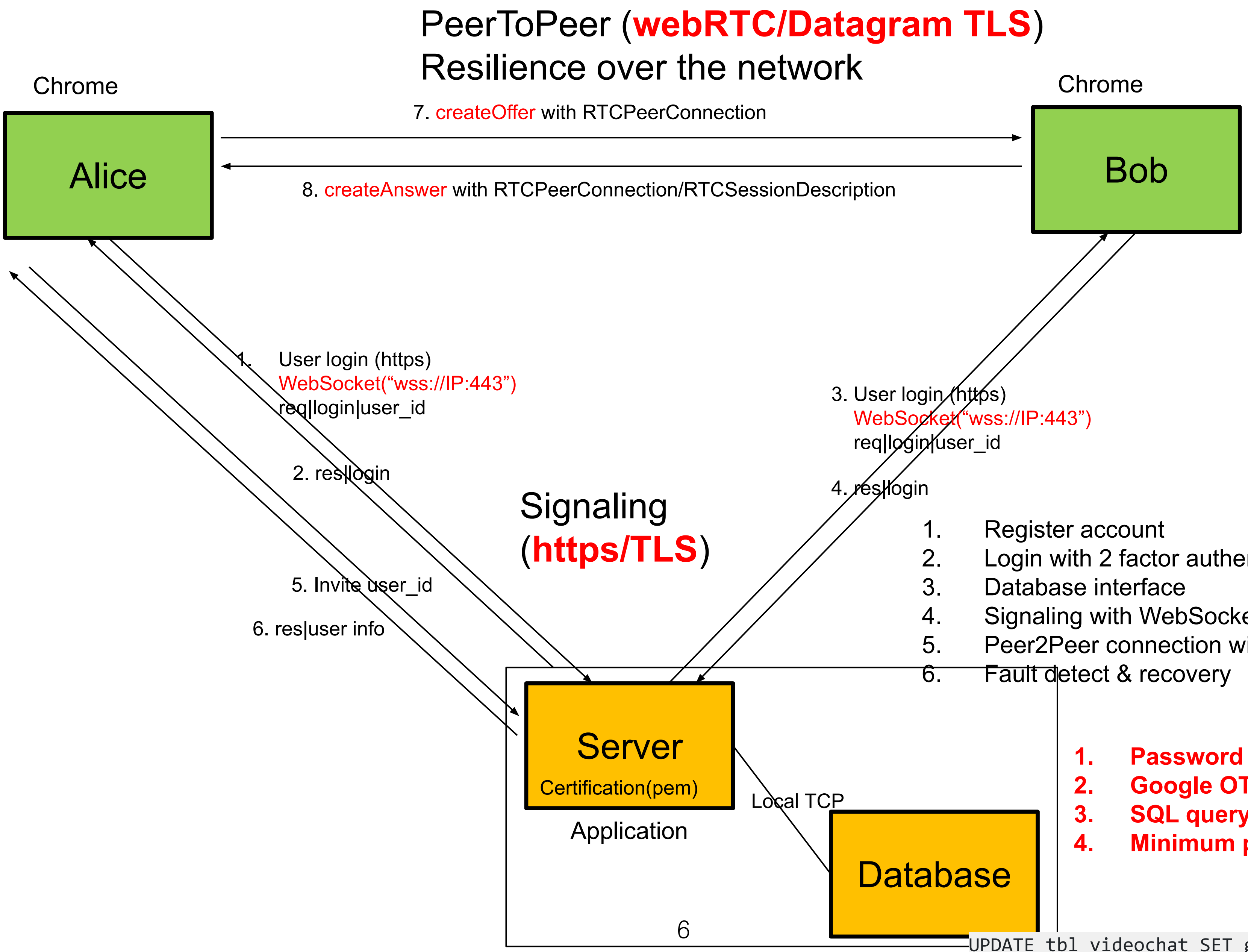




Signaling (TLS)

Media (DTLS)

Overall Architect



PeerToPeer (**webRTC/Datagram TLS**)
Resilience over the network

Chrome

Alice

Chrome

Bob

7. createOffer with RTCPeerConnection

8. createAnswer with RTCPeerConnection/RTCSessionDescription

1. User login (https)
WebSocket(“wss://IP:443”)
req|login|user_id

2. res|login

3. User login (https)
WebSocket(“wss://IP:443”)
req|login|user_id

4. res|login

5. Invite user_id

6. res|user info

Signaling
(**https/TLS**)

Server
Certification(pem)

Application

Local TCP

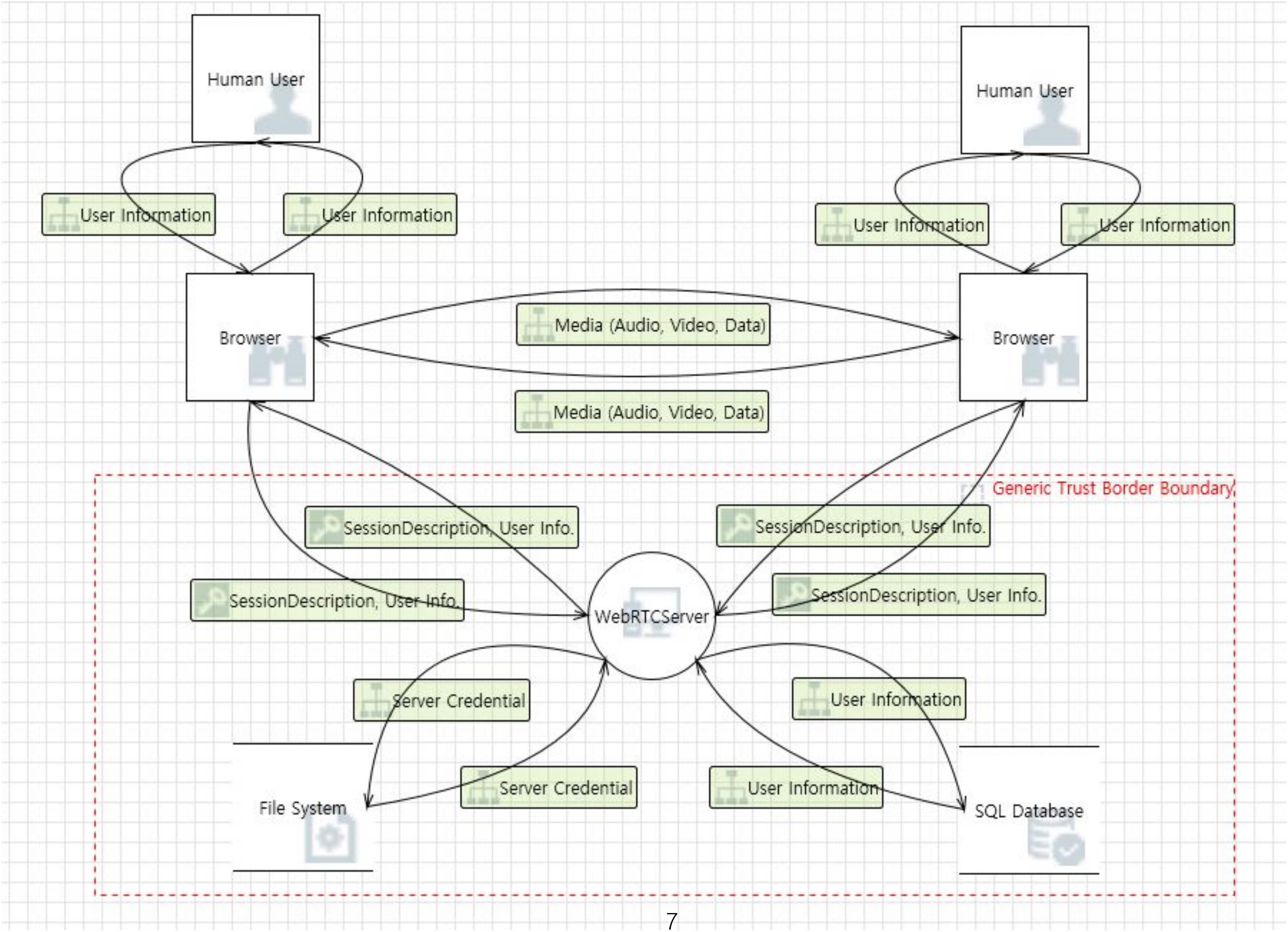
Database

1. Register account
2. Login with 2 factor authentication
3. Database interface
4. Signaling with WebSocket secure & SDP
5. Peer2Peer connection with webRTC
6. Fault detect & recovery

1. **Password - SHA256 with salt**
2. **Google OTP key - AES**
3. **SQL query binding**
4. **Minimum privilege access**

```
UPDATE tbl videochat SET gotp = ? WHERE uniqu
```

Threat Modeling


Human User
Human User
User Information
User Information
User Information
User Information
Browser
Browser
Media (Audio, Video, Data)
Media (Audio, Video, Data)
Generic Trust Border Boundary
SessionDescription, User Info.
SessionDescription, User Info.
SessionDescription, User Info.
SessionDescription, User Info.
WebRTCServer
Server Credential
Server Credential
User Information
User Information
File System
SQL Database

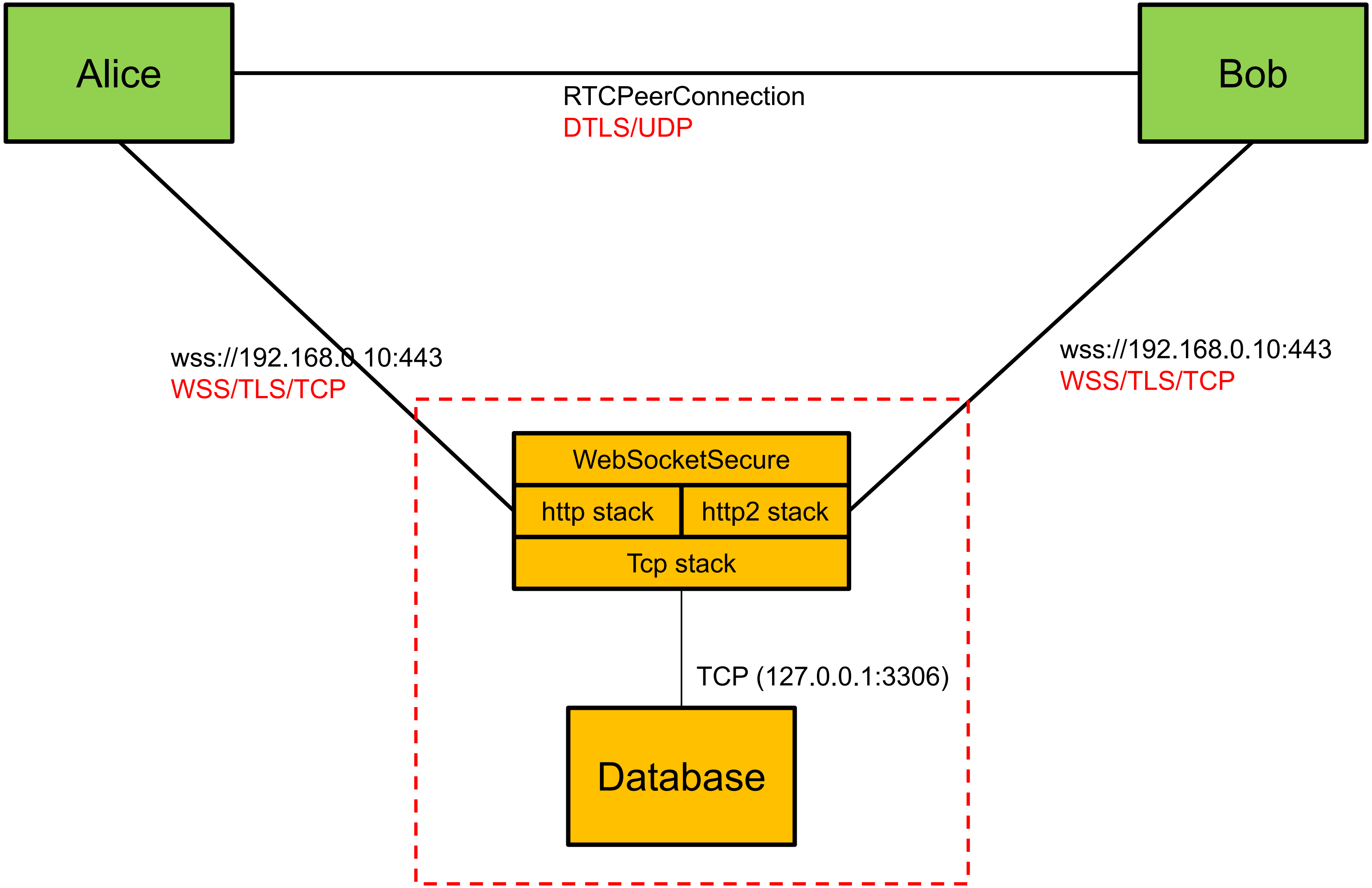
Threat Lists (1/2)

Priority	Category	When/Where	Threat/Requirement Description	Mitigation	Requirement #
1	Tampering	Brower → WebRTCServer	The web server 'WebRTCServer' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.	1. Input validation 2. Input sanitization(Use Regular Expression): Special characters can be blocked or removed	**R1.1.2**
2	Elevation Of Privilege	Brower → WebRTCServer	WebRTCServer may be able to impersonate the context of Browser in order to gain additional privilege.	1. Use 2FA 2. Use Least Privilege principle	**R1.2**
3	Repudiation	Brower → WebRTCServer	WebRTCServer claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	1. Logging 2. TLS (Server Authentication)	**R15**
4	Elevation Of Privilege	Brower → WebRTCServer	An attacker may pass data into WebRTCServer in order to change the flow of program execution within WebRTCServer to the attacker's choosing.	1. Input validation 2. Input sanitization 3. Use Least Privilege principle	**R1.1.2**

Threat Lists (2/2)

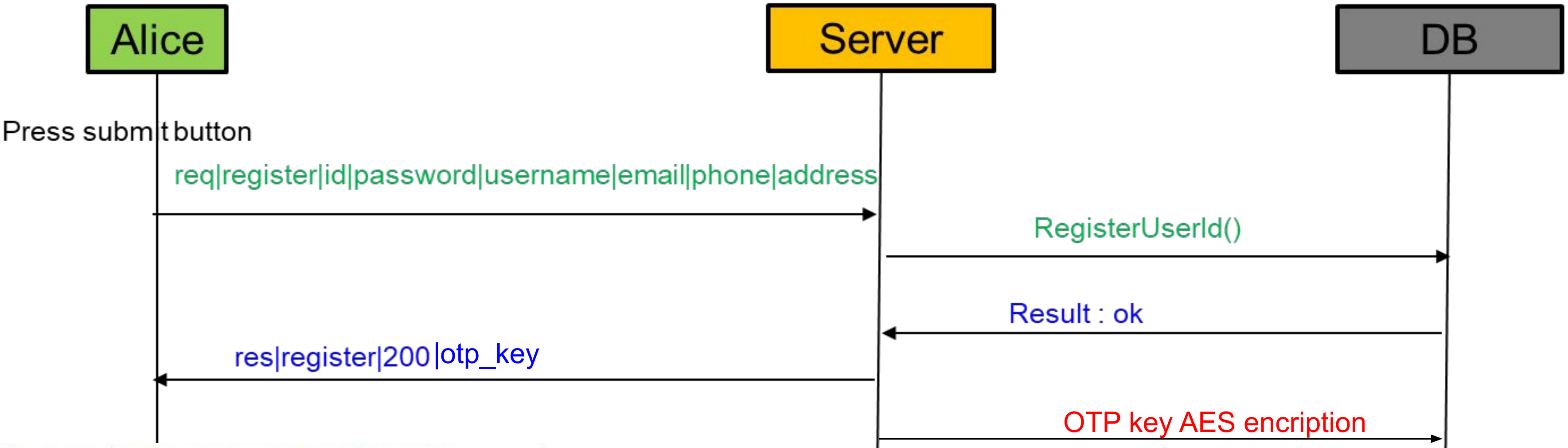
Priority	Category	When/Where	Threat/Requirement Description	Mitigation	Requirement #
5	Spoofing	SQL DB → WebRTCServer	SQL Database may be spoofed by an attacker and this may lead to incorrect data delivered to WebRTCServer. Consider using a standard authentication mechanism to identify the source data store.	1. Block remote access to DB 2. Use strong password 3. TLS: meets both authentication and encryption	R16
6	Information Disclosure	SQL DB	Improper data protection of SQL Database can allow an attacker to read information not intended for disclosure. Review authorization settings.	1. Use SQL bind variabe 2. Block remote access to DB 3. Use strong password 4. DB Encryption	R16
7	Spoofing	WebRTCServer → FileSystem	File System may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of File System. Consider using a standard authentication mechanism to identify the destination data store.	1. Keep server credential in secure storage	R16

Static View


Alice
Bob
RTCPeerConnection
DTLS/UDP
wss://192.168.0.10:443
WSS/TLS/TCP
wss://192.168.0.10:443
WSS/TLS/TCP
WebSocketSecure
http stack
http2 stack
Tcp stack
TCP (127.0.0.1:3306)
Database

Sequence View (User Registration)

Sequence Diagram



Alice

Server

DB

Press submit button

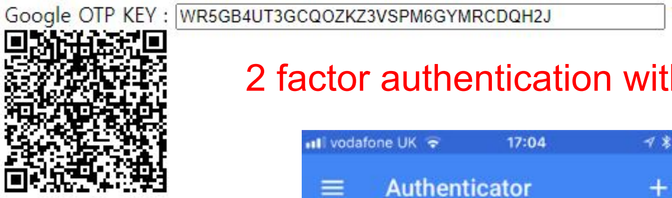
req|register|id|password|username|email|phone|address

RegisterUserId()

Result : ok

res|register|200 |otp_key

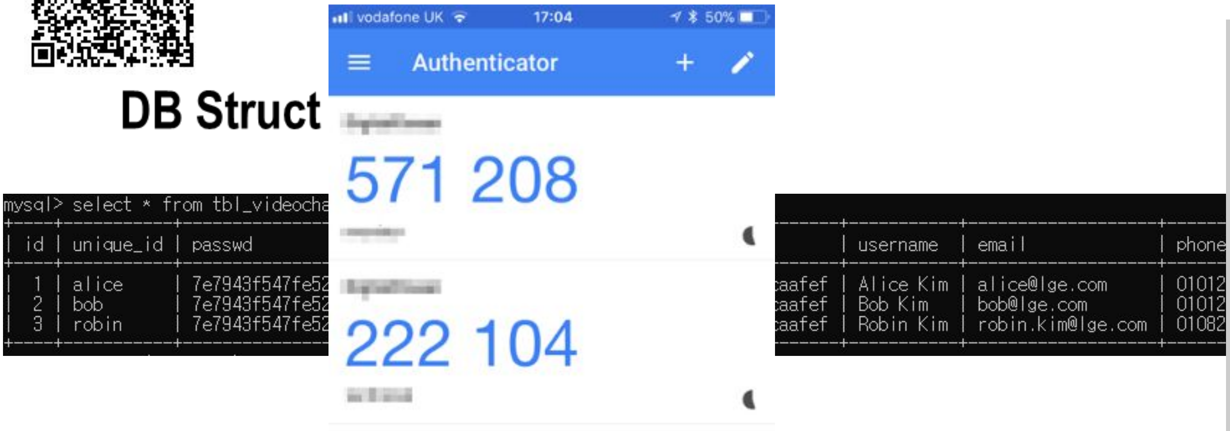
OTP key AES encription



Google OTP KEY : WR5GB4UT3GCQOZKZ3VSPM6GYMRCDQH2J

2 factor authentication with Google OTP

DB Struct



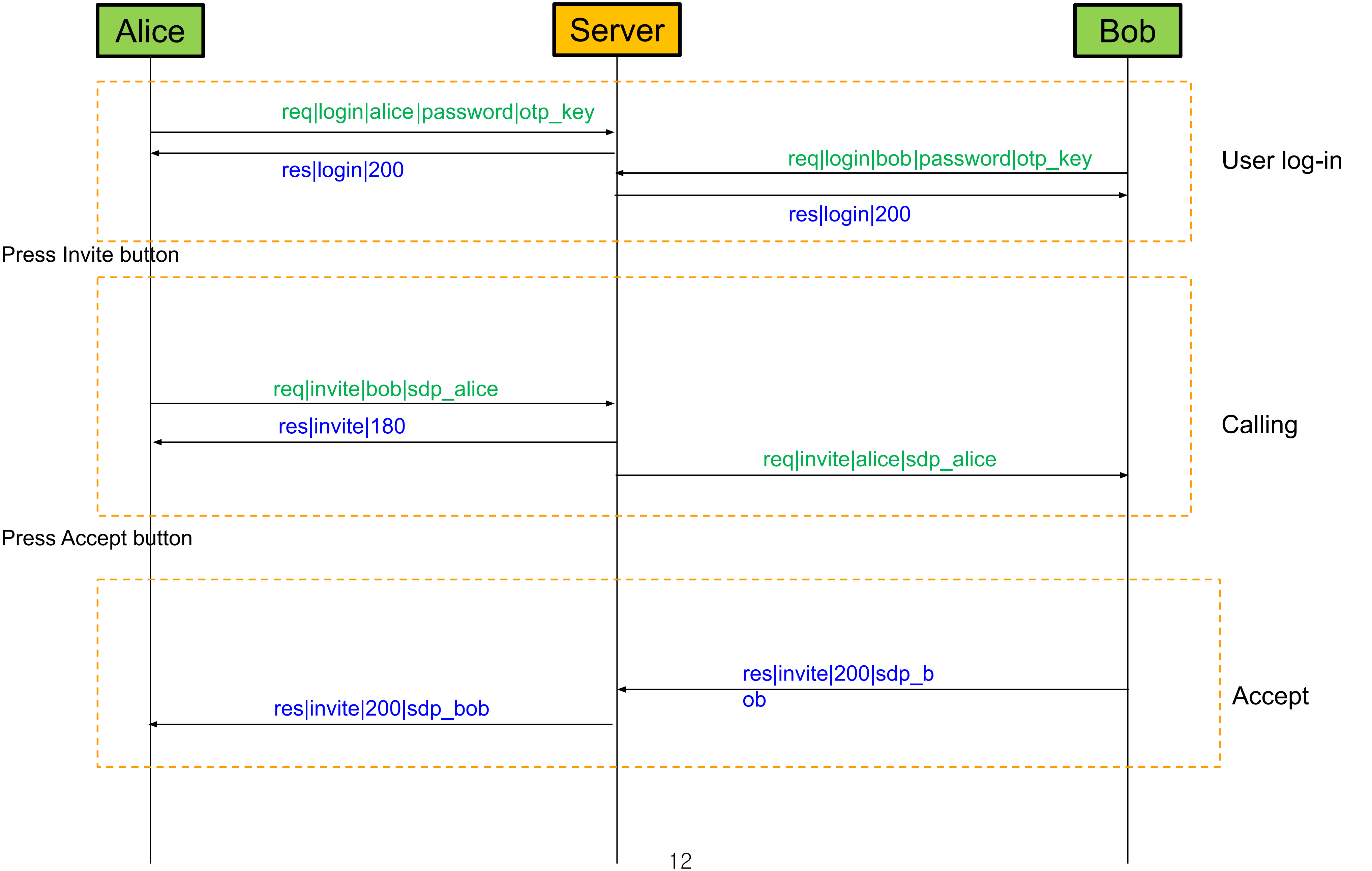
571 208

222 104

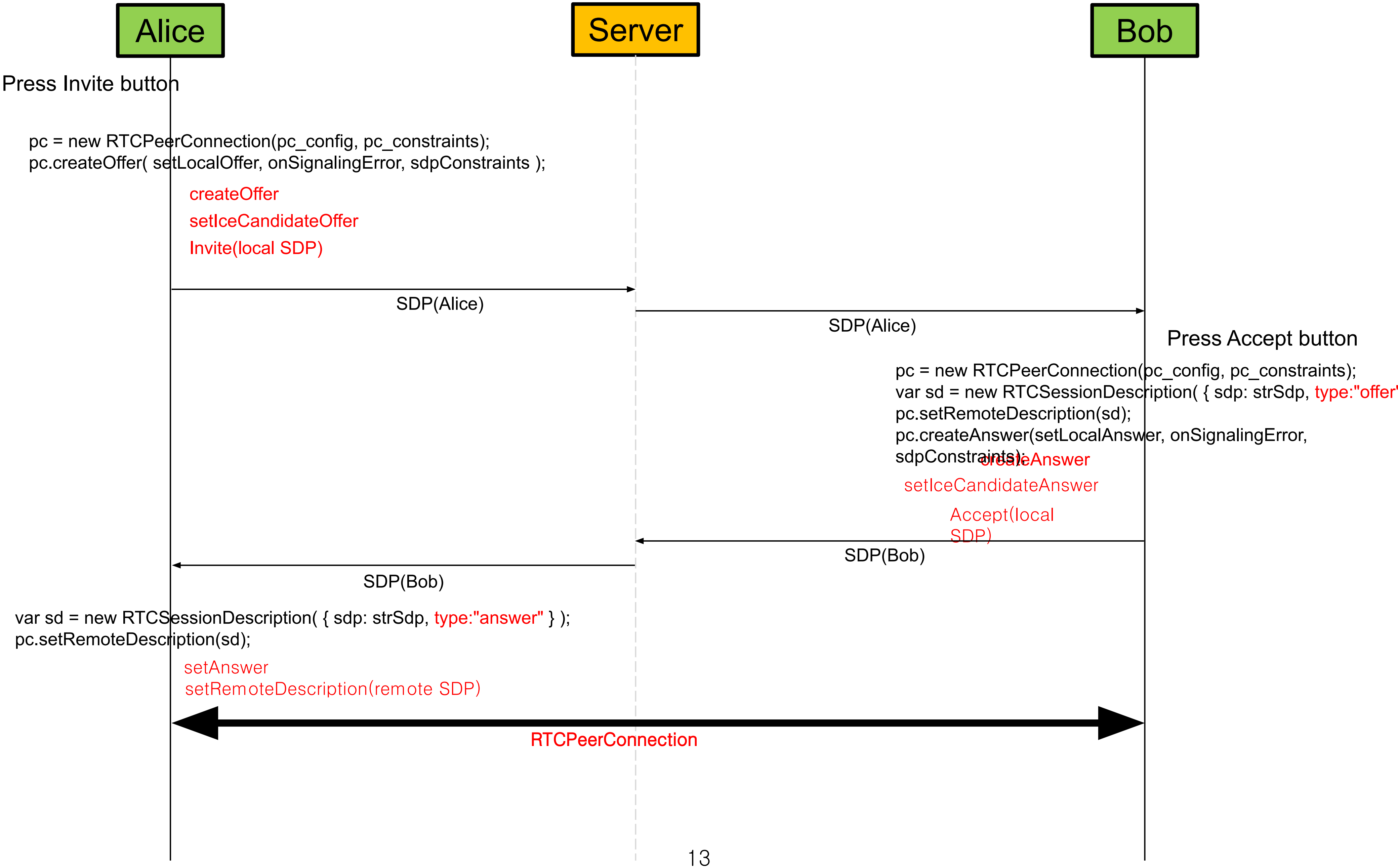
```
mysql> select * from tbl_videocha
+----+-----------+---------------  -------+-----------+-------------------+------
| id | unique_id | passwd                 | username  | email             | phone
+----+-----------+---------------  -------+-----------+-------------------+------
|  1 | alice     | 7e7943f547fe52  caafef | Alice Kim | alice@lge.com     | 01012
|  2 | bob       | 7e7943f547fe52  caafef | Bob Kim   | bob@lge.com       | 01012
|  3 | robin     | 7e7943f547fe52  caafef | Robin Kim | robin.kim@lge.com | 01082
+----+-----------+---------------  -------+-----------+-------------------+------
```

```
#define SQL_CREATE_TBL                    \
            " CREATE TABLE IF NOT EXISTS tbl_videochat( \n \
            id INTEGER PRIMARY KEY AUTO_INCREMENT, \n \
            unique_id  TEXT    NOT NULL, \n \
            passwd     TEXT    NOT NULL, \n \
            username   TEXT    NOT NULL, \n \
            email      TEXT    NOT NULL, \n \
            phone      TEXT    NOT NULL, \n \
            address    TEXT    NOT NULL, \n \
            passwd_update_utc INTEGER,   \n \
            passwd_wrong_cnt  INTEGER DEFAULT 0,  \n \
            passwd_lock_utc   INTEGER DEFAULT 0,   \n \
            gotp       TEXT             \n \
            );"
```

Sequence View (Signaling Call Accept)



Sequence View (Peer to Peer)



Press Invite button

```
pc = new RTCPeerConnection(pc_config, pc_constraints);
pc.createOffer( setLocalOffer, onSignalingError, sdpConstraints );
```

createOffer

setIceCandidateOffer

Invite(local SDP)

Press Accept button

```
pc = new RTCPeerConnection(pc_config, pc_constraints);
var sd = new RTCSessionDescription( { sdp: strSdp, type:"offer"
pc.setRemoteDescription(sd);
pc.createAnswer(setLocalAnswer, onSignalingError,
sdpConstraints);
```

createAnswer

setIceCandidateAnswer

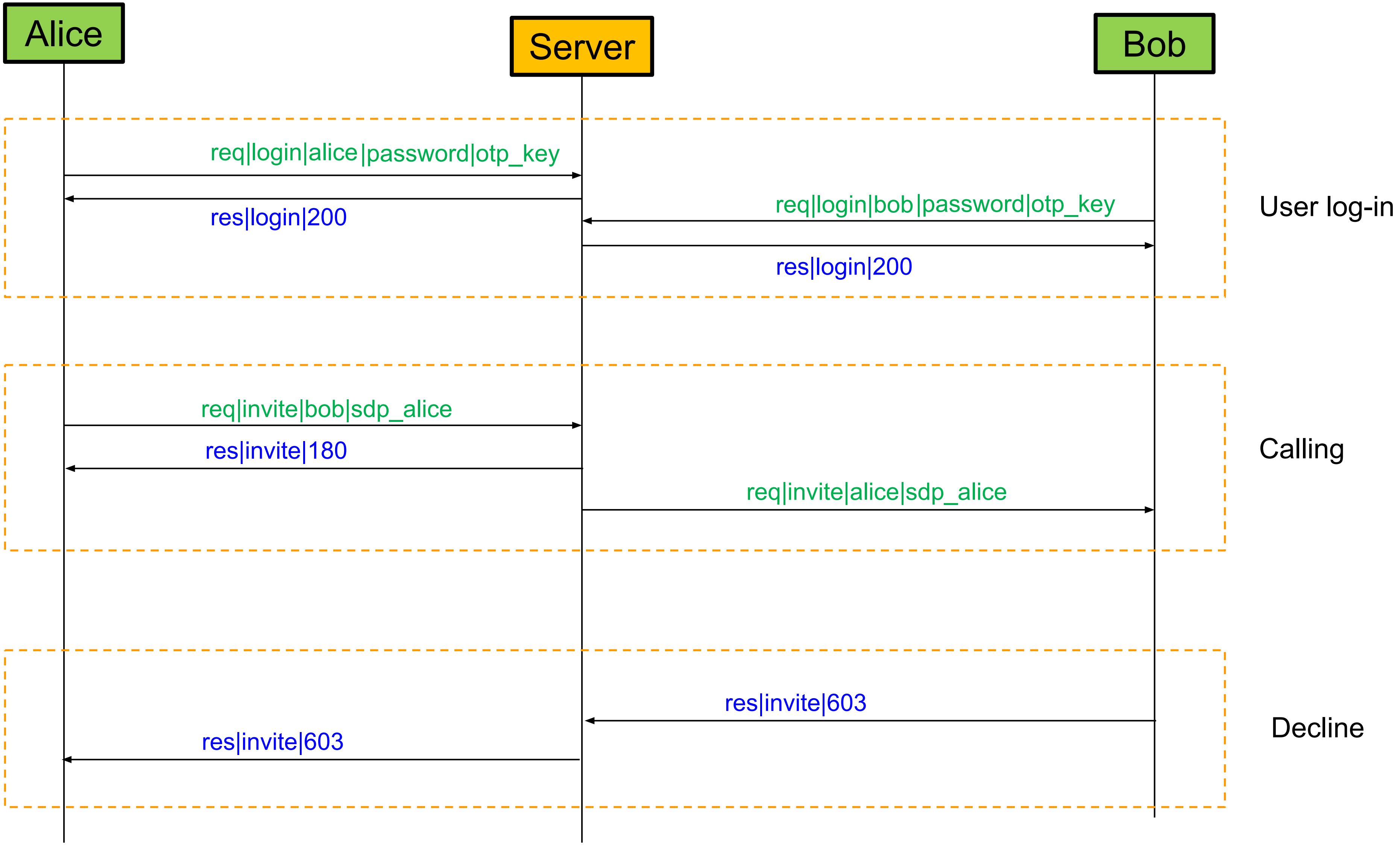
Accept(local SDP)

```
var sd = new RTCSessionDescription( { sdp: strSdp, type:"answer" } );
pc.setRemoteDescription(sd);
```

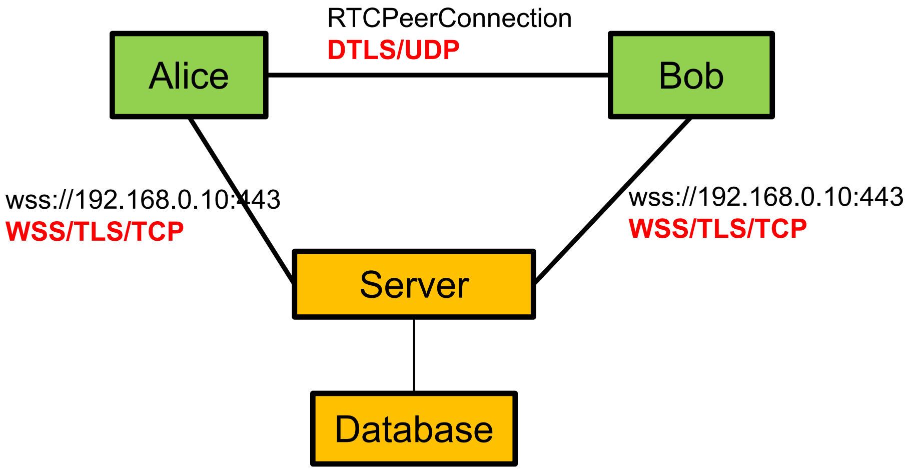
setAnswer

setRemoteDescription(remote SDP)

Sequence View (Signaling Call Decline)



Security View



RTCPeerConnection
DTLS/UDP

Alice

Bob

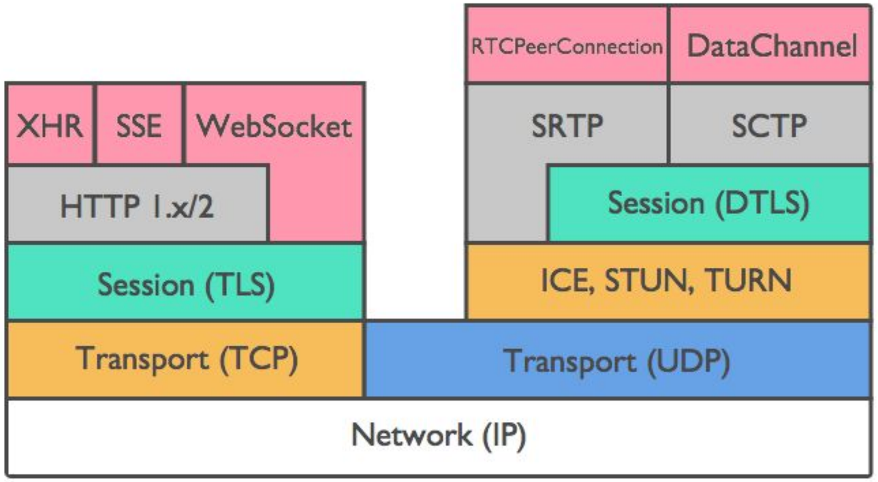
wss://192.168.0.10:443
WSS/TLS/TCP

wss://192.168.0.10:443
WSS/TLS/TCP

Server

Database

1. **Password - SHA256 with salt**
2. **Google OTP key - AES**
3. **SQL bind variable**
4. **Minimum privilege access**



Confidentiality

- TLS encrypt the data and attacker can only see the encrypted data
- client/server uses SSL/TLS for authentication
- peer to peer uses DTLS for date secure
- password sha256 hash with salt
- Google OTP key is encrypted with AES

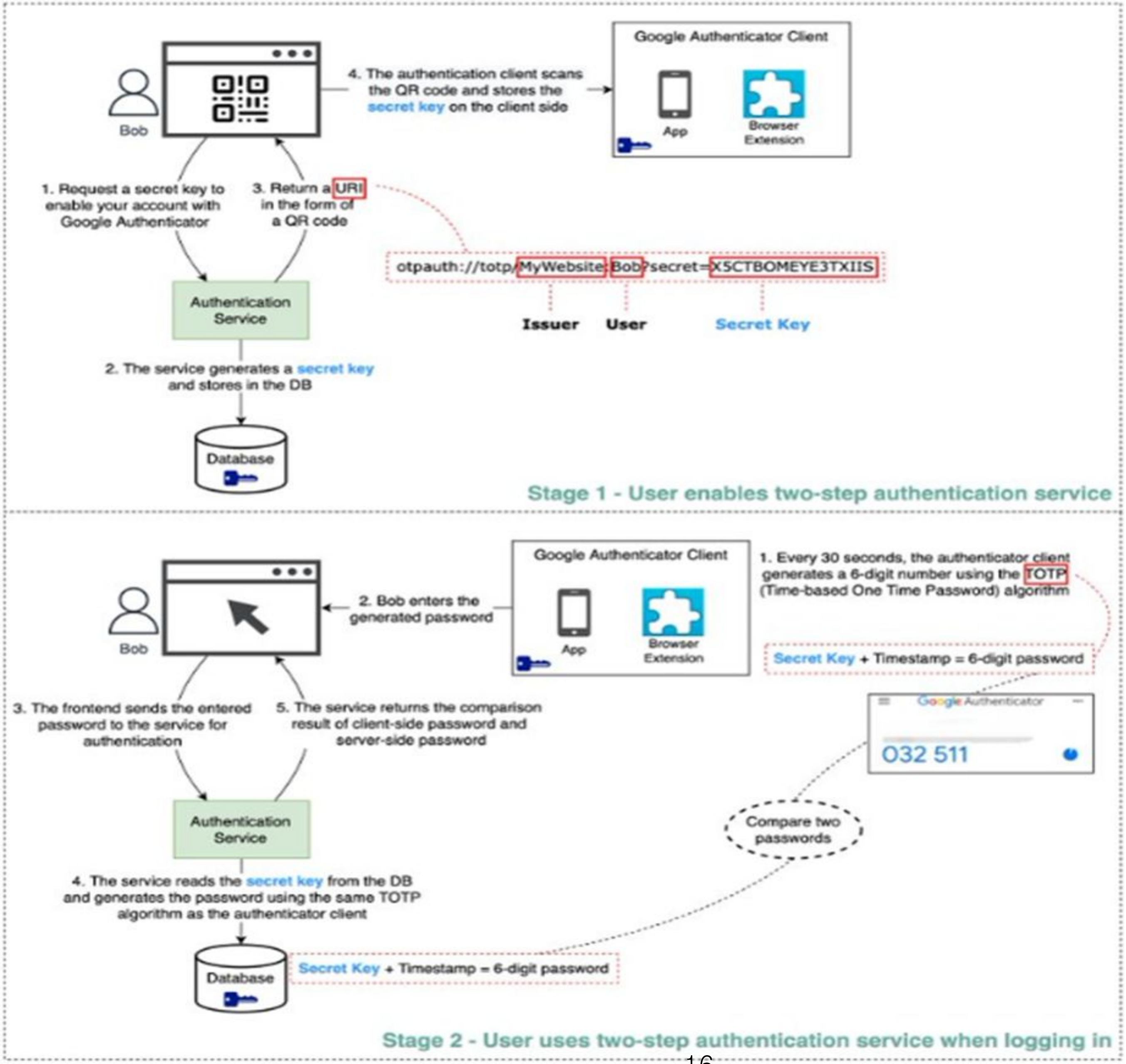
Integrity

- TLS and DTLS can guarantee the integrity as well.

Availability

- connection failed physically, it resume the connection again automatically (QUIC)

2 Factor Authentication (sequence)



Bob

4. The authentication client scans the QR code and stores the secret key on the client side

Google Authenticator Client

App

Browser Extension

1. Request a secret key to enable your account with Google Authenticator

3. Return a URI in the form of a QR code

otpauth://totp/MyWebsite:Bob?secret=X5CTBOMEYE3TXIIS

Issuer

User

Secret Key

Authentication Service

2. The service generates a secret key and stores in the DB

Database

Stage 1 - User enables two-step authentication service

Bob

2. Bob enters the generated password

Google Authenticator Client

App

Browser Extension

1. Every 30 seconds, the authenticator client generates a 6-digit number using the TOTP (Time-based One Time Password) algorithm

Secret Key + Timestamp = 6-digit password

Google Authenticator

032 511

3. The frontend sends the entered password to the service for authentication

5. The service returns the comparison result of client-side password and server-side password

Compare two passwords

Authentication Service

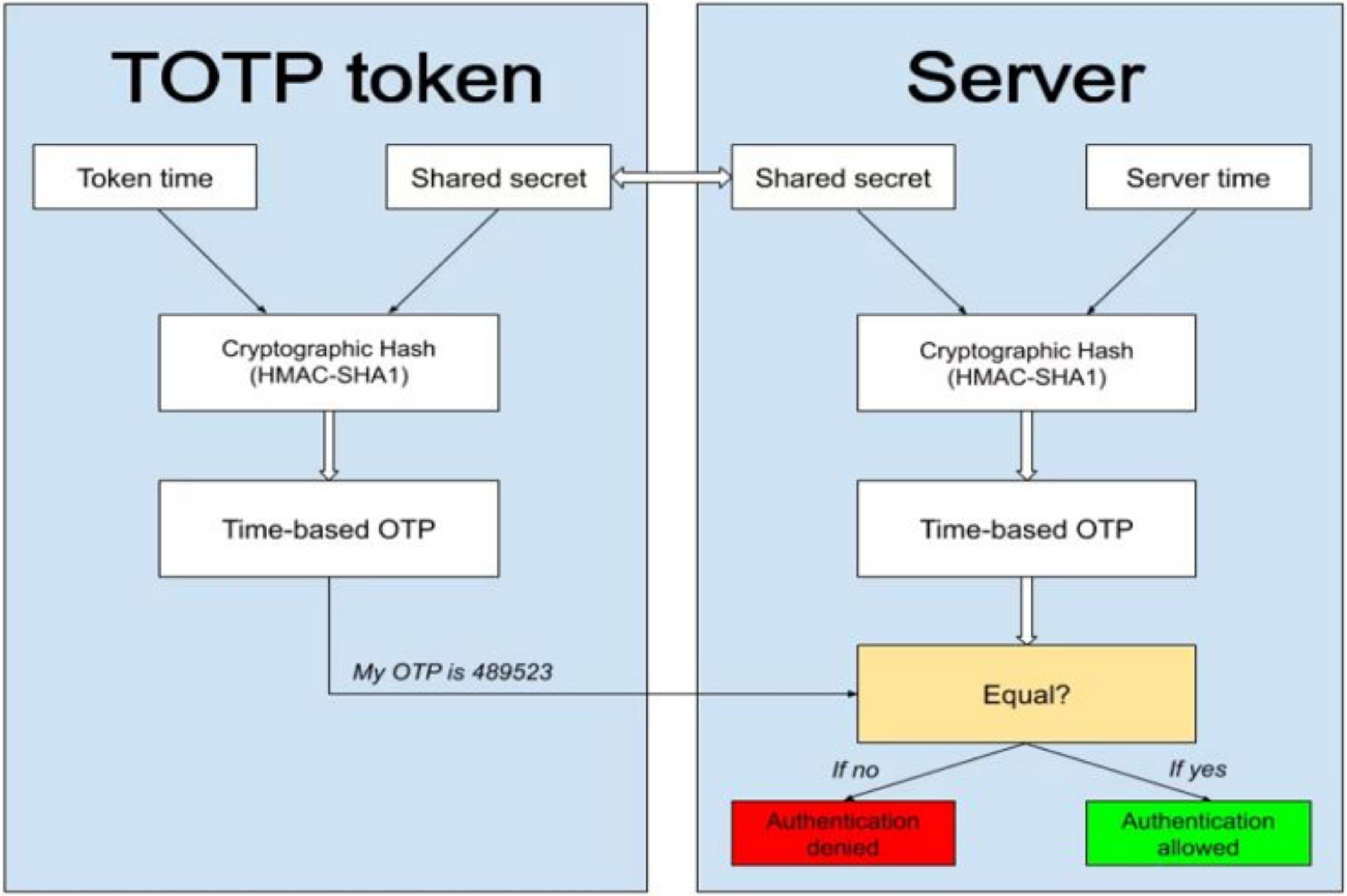
4. The service reads the secret key from the DB and generates the password using the same TOTP algorithm as the authenticator client

Database

Secret Key + Timestamp = 6-digit password

Stage 2 - User uses two-step authentication service when logging in

2 Factor Authentication (internal)



Functional Requirements

No	Requirements	Satisfactory	Security
R1	1. The ability for the user to register with the system	YES	**TLS based wss used with port 443**
R1.1	1.1 The user shall provide the system their email address and password.	YES	
R1.1.1	1.1.1 The system shall ensure that the user's password is secure.	YES	**Password is saved SHA-256 with salt**
R1.1.2	1.1.2 Passwords must be a minimum of 10 characters long and include one number and one symbol.	YES	**LG Password rule is applied 1. at least 1 alphabet 2. at least 1 special character 3. at least 1 number 4. length should be 10 ~ 15**
R1.2	1.2 The system shall use two-factor authentication.	YES	**Google OTP is applied as well as user password Google OTP can be used with "QR code" or "Setup Key"**
R1.3	1.3 The system should force a user to periodically reset their password (at least once a month).	YES	**Detect 1 month past after setting password Password setting UTC time is saved in Database**
R1.4	1.4 If the user enters the incorrect password more than three times, then their account will be locked for one hour.	YES	**Password wrong count and UTC time tried is saved in Database / User blocked**
R1.5	1.5 The system shall allow users to change their email address in a secure way.	YES	**Email address can be changed securely using TLS**
R1.6	1.6 The system shall provide the ability for the user to recover or change their password in the event it is lost.	YES	**Password can be changed securely using TLS. Password can be sent to user's email by SMTP with TLS.**

Functional Requirements

No	Requirements	Satisfactory	Security
R2	2. After successful registration the system shall assign the user a unique contact identification name (contact identifier).	YES	**system has a functionality checking unique_id is unique** **Unique id is used for the system**
R2.1	2.1 this can be the user's email address or some other name chosen by the user if it does not conflict with other user's contact identifiers already in the system.	YES	**system has a functionality checking unique_id is unique**
R3	3. The system shall provide a contact list that associates a person with their contact identifier (last name, first name, address, e-mail, contact identifier).	YES	**System provide the contact list login-ed through TLS connection** **"Retrieved Logined ID" button support this requirement**
R3.1	3.1 . When a contact is associated with a contact identifier the VoIP application shall display the contact's name instead of the contact identifier.	YES	**All login user information is displayed**
R4	4. The system shall provide the ability to initiate a call using a contact identifier or the contacts list.	YES	**unique_id is used for peer connection**
R4.1	4.1 During the call initiation, the user shall be presented with call status and outcome (answered, busy or rejected).	YES	**system presents BUSY, REJECT, ANSWER**
R4.2	4.2 During call initiation the user shall have the ability to end the call at any time.	YES	**system has the ability to Call END**

Functional Requirements

No	Requirements	Satisfactory	Security
R5	5. The system shall provide the ability to accept or reject calls while not in a call.	YES	**System has the ability to Accept or Reject**
R5.1	5.1 Application shall show the caller’s contact identifier or contact name during an incoming call.	YES	**During incoming call, user can see the contact name**
R6	6. The system shall notify the user of missed calls, either because the call was not accepted or because the called entity was in another call.	YES	**system notify missed call**
R7	7. Provide the ability to terminate a call at any time while in a call.	YES	**system can terminate call**
R7.1	7.1 If a call is terminated by one user, the other caller shall be notified.	YES	**system can notify it**
R8	8. Application shall be brought to the foreground during an incoming call.	YES	**We are using web browser(e.g. Google Chrome) as a client, so bringing the browser to the foreground might be challenging. However, we notify users through pop-up alerts.**
R9	9. This application is a point-to-point communication system. That is, each end point of the call should function as both a server and a client.	YES	**WebRTC peer connection is used**

Quality Attribute (Non Functional)

No	Requirements	Details	Satisfactory
R10	Performance	The system must deliver call video/audio as close to real time as possible.	**YES**
R11	Authentication	1. The system must use two factor authentication for sign on and user credentials must be protected. 2. Lost or compromised credentials must be handled in a reasonable way.	**YES**
R12	Communication privacy	1. The system must ensure that calls remain private. 2. No intermediary should be able to snoop or spy on an ongoing call.	**YES**
R13	Proof of identity (nonrepudiation)	Users should be confident that the entity they are on a call with is the one that they believe it is.	**YES**
R14	Reliability	1. The system must ensure that calls are reliable. 2. The system should recover from networking errors and dropped calls as soon as possible. 3. The goal is to maintain a secure, performant connection at all costs.	**YES** **If disconnected network, it recover again with QUIC protocol**

Security Requirements

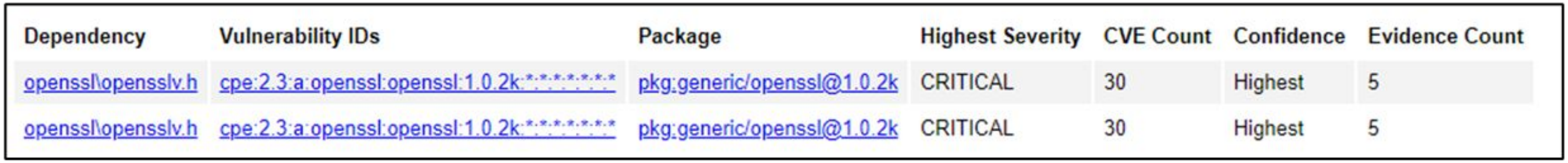
No	Requirements	Details	Satisfactory
R15	Reliability	All transaction should be logged	**YES**
R16	SQL query safe	SQL query should be binded	**YES**

Threat & Security - dependency-check

1st test : refer to dependency-check result file

1st result

- result : openssl 1.0.2k has 30 vulnerabilities
- How to mitigate : openssl upgrade 1.1.1u (latest version)



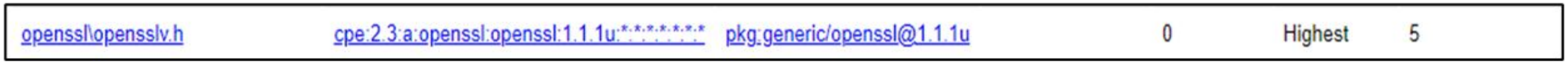
Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
openssl\opensslv.h	cpe:2.3:a:openssl:openssl:1.0.2k:*:*:*:*:*:*:*	pkg:generic/openssl@1.0.2k	CRITICAL	30	Highest	5
openssl\opensslv.h	cpe:2.3:a:openssl:openssl:1.0.2k:*:*:*:*:*:*:*	pkg:generic/openssl@1.0.2k	CRITICAL	30	Highest	5

OpenSSL migration
(1.0.2k -> 1.1.1u)

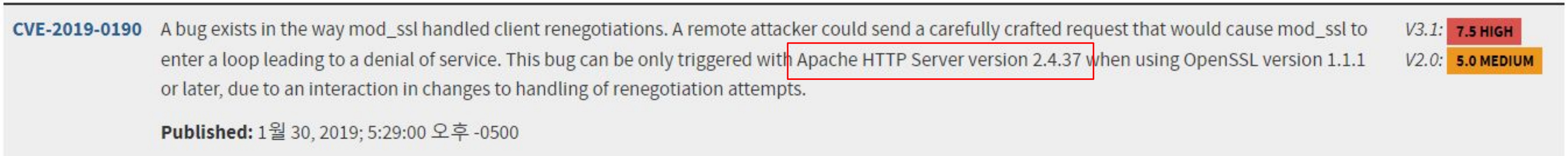
2nd test : refer to dependency-check result file

2nd result

- Result : There's no vulnerabilities



openssl\opensslv.h	cpe:2.3:a:openssl:openssl:1.1.1u:*:*:*:*:*:*:*	pkg:generic/openssl@1.1.1u		0	Highest	5

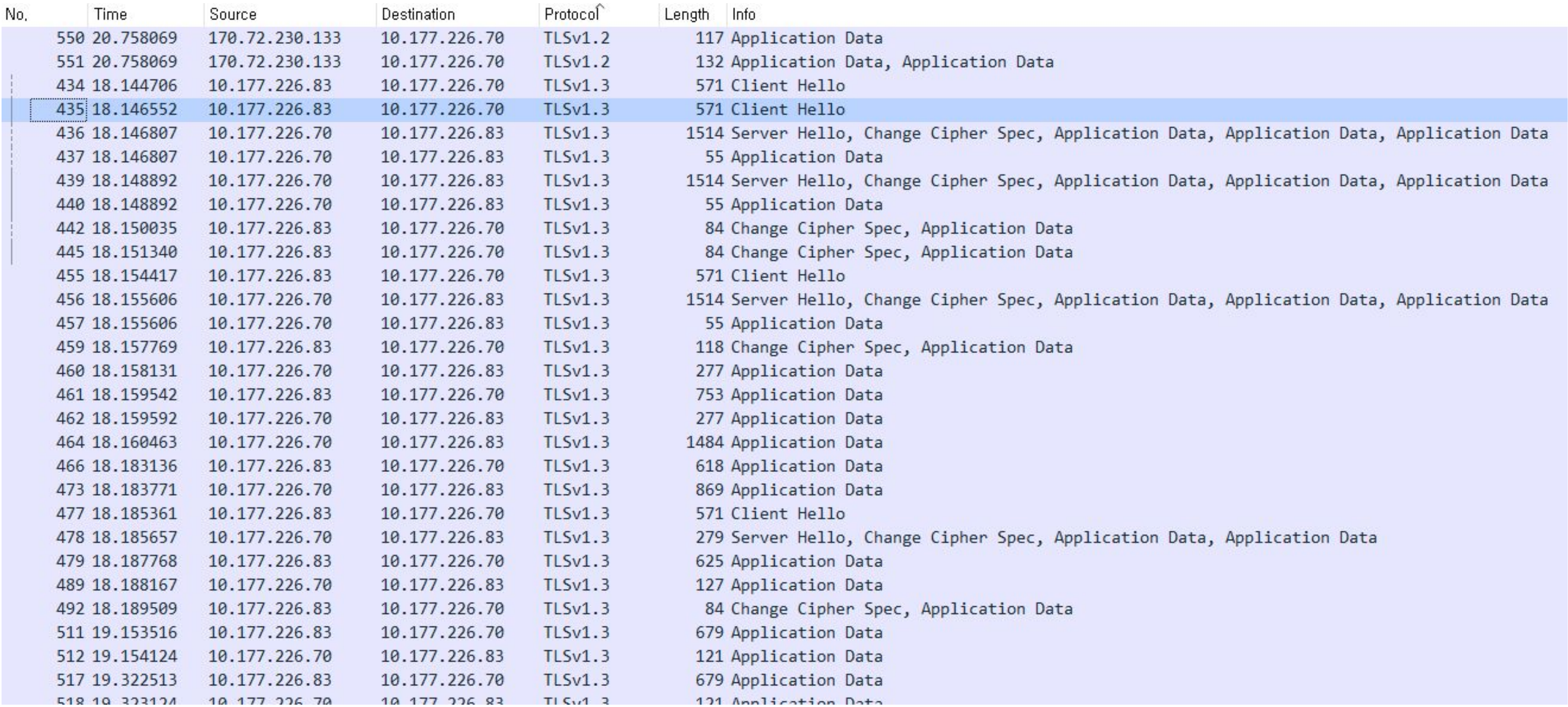


CVE-2019-0190 A bug exists in the way mod_ssl handled client renegotiations. A remote attacker could send a carefully crafted request that would cause mod_ssl to enter a loop leading to a denial of service. This bug can be only triggered with Apache HTTP Server version 2.4.37 when using OpenSSL version 1.1.1 or later, due to an interaction in changes to handling of renegotiation attempts.

V3.1: 7.5 HIGH
V2.0: 5.0 MEDIUM

Published: 1월 30, 2019; 5:29:00 오후 -0500

TLS Validation Test (wireshark)



No.	Time	Source	Destination	Protocol	Length	Info
550	20.758069	170.72.230.133	10.177.226.70	TLSv1.2	117	Application Data
551	20.758069	170.72.230.133	10.177.226.70	TLSv1.2	132	Application Data, Application Data
434	18.144706	10.177.226.83	10.177.226.70	TLSv1.3	571	Client Hello
435	18.146552	10.177.226.83	10.177.226.70	TLSv1.3	571	Client Hello
436	18.146807	10.177.226.70	10.177.226.83	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data
437	18.146807	10.177.226.70	10.177.226.83	TLSv1.3	55	Application Data
439	18.148892	10.177.226.70	10.177.226.83	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data
440	18.148892	10.177.226.70	10.177.226.83	TLSv1.3	55	Application Data
442	18.150035	10.177.226.83	10.177.226.70	TLSv1.3	84	Change Cipher Spec, Application Data
445	18.151340	10.177.226.83	10.177.226.70	TLSv1.3	84	Change Cipher Spec, Application Data
455	18.154417	10.177.226.83	10.177.226.70	TLSv1.3	571	Client Hello
456	18.155606	10.177.226.70	10.177.226.83	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data
457	18.155606	10.177.226.70	10.177.226.83	TLSv1.3	55	Application Data
459	18.157769	10.177.226.83	10.177.226.70	TLSv1.3	118	Change Cipher Spec, Application Data
460	18.158131	10.177.226.70	10.177.226.83	TLSv1.3	277	Application Data
461	18.159542	10.177.226.83	10.177.226.70	TLSv1.3	753	Application Data
462	18.159592	10.177.226.70	10.177.226.83	TLSv1.3	277	Application Data
464	18.160463	10.177.226.70	10.177.226.83	TLSv1.3	1484	Application Data
466	18.183136	10.177.226.83	10.177.226.70	TLSv1.3	618	Application Data
473	18.183771	10.177.226.70	10.177.226.83	TLSv1.3	869	Application Data
477	18.185361	10.177.226.83	10.177.226.70	TLSv1.3	571	Client Hello
478	18.185657	10.177.226.70	10.177.226.83	TLSv1.3	279	Server Hello, Change Cipher Spec, Application Data, Application Data
479	18.187768	10.177.226.83	10.177.226.70	TLSv1.3	625	Application Data
489	18.188167	10.177.226.70	10.177.226.83	TLSv1.3	127	Application Data
492	18.189509	10.177.226.83	10.177.226.70	TLSv1.3	84	Change Cipher Spec, Application Data
511	19.153516	10.177.226.83	10.177.226.70	TLSv1.3	679	Application Data
512	19.154124	10.177.226.70	10.177.226.83	TLSv1.3	121	Application Data
517	19.322513	10.177.226.83	10.177.226.70	TLSv1.3	679	Application Data
518	19.323124	10.177.226.70	10.177.226.83	TLSv1.3	121	Application Data

Work to do / Limitation

- Server Certification Key self-signed is not saved in secure storage of Windows

Test Case -1

Req_ No.	Requirement	Test Case	Result
R1 R1.1	1. The ability for the user to register with the system 1.1 The user shall provide the system their email address and password.	TC000) A registration screen should be provided where you can enter your ID, password, name, email, phone, and address.	OK
		TC001) The ID, password, name, and email information are mandatory, and if they are missing, a separate warning popup will be displayed when the submit button is pressed.	OK
		TC002) After entering the information and clicking the submit button, you should see a success popup.	OK
R1.1.1 R1.1.2	1.1.1 The system shall ensure that the user's password is secure. 1.1.2 Passwords must be a minimum of 10 characters long and include one number and one symbol.	TC003) Provide a guide on the password entry screen (at least 1 alphabet/special character/number and length 10 to 15).	OK
		TC004) Provide the ability to retype passwords, giving users the ability to double-check their passwords.	OK
R1.2	1.2 The system shall use two-factor authentication.	TC005) After entering the user information and clicking the Submit button, the screen should provide a QR code for Google OTP and a setup key value.	OK
		TC006) Using the Google OTP app, you can scan the QR code or enter the setup key to receive the authentication key value.	OK

Test Case -2

Req_ No.	Requirement	Test Case	Result
R1.4	1.4 If the user enters the incorrect password more than three times, then their account will be locked for one hour.	TC007) To log in to the system, you will need to enter the saved user ID, password, and OTP values and click the login button.	OK
		TC008) If you enter an incorrect password and press the login button, a pop-up will be displayed error popup.	OK
		TC009) If you enter the password incorrectly 3 times, a pop-up will be displayed saying 'Too many password is wrong, wait for 60 seconds'.(The requirement says that it should be locked for 1 hour, but we changed it to 1 minute lock for testing convenience)	OK
		TC010) If an incorrect otp value is entered, a 'OTP is WRONG' warning popup should be displayed.	OK
		TC011) If the user id, password, and otp value are entered correctly, the user should be logged in normally by pressing the login button.	OK

Test Case -3

Req_ No.	Requirement	Test Case	Result
R1.5	1.5 The system shall allow users to change their email address in a secure way.	TC012) Once you've logged in, the Change Email button should be active.	OK
		TC013) Click it, enter the email address you want to change and click submit, and your email address should be changed.	OK
		TC014) If you press submit button without entering an email address, it displays error popup.	OK
R2	2. After successful registration the system shall assign the user a unique contact identification name (contact identifier).	TC015) During user registration, a button UI should be provided to enter a unique id.	OK
		TC016) If the user presses the submit button without entering an ID, a warning popup should be displayed stating "userid has not been entered".	OK
		TC017) If the user enters a username and presses the submit button, it should proceed to the next step unless it is a duplicate of a value already in the database.	OK?

Test Case -4

Req_ No.	Requirement	Test Case	Result
R2.1	2.1 this can be the user’s email address or some other name chosen by the user if it does not conflict with other user’s contact identifiers already in the system.	TC018) Provide a button called Check id to allow the user to check for duplicates when registering an ID.	OK
		TC019) If there are no duplicates, when the user clicks the check id button, a popup will display 'user ID is available{userid]'.	OK
		TC020) If a duplicate ID is entered, a warning will be displayed saying 'user ID ALREADY registered. Please use another user ID' warning popup’.	OK
R3 R3.1	3. The system shall provide a contact list that associates a person with their contact identifier (last name, first name, address, e-mail, contact identifier). 3.1 . When a contact is associated with a contact identifier the VoIP application shall display the contact’s name instead of the contact identifier.	TC021) The Retrieve login ID button is disabled when the user is not logged in.	OK
		TC022) If you click the Retrieve login ID button after logging in, the contact list will be displayed at the bottom of the screen.	OK
		TC023) At the bottom, the value (contact identifier, name, e-mail, phone, address) entered by the user during registration should be displayed.	OK

Test Case -5

Req_ No.	Requirement	Test Case	Result
R4	4. The system shall provide the ability to initiate a call using a contact identifier or the contacts list.	TC024) View the Contact list, enter the id of the contact list in the peer ID and click the CALL button to make a call.	OK
		TC025) If you enter a user who is not logged in to the peer ID and make a call, you will see a popup saying 'peer ID is not log-in now' and 'please press 'retrieved login ID' button'.	OK
R4.1 R5 R5.1	4.1 During the call initiation, the user shall be presented with call status and outcome (answered, busy or rejected). 5. The system shall provide the ability to accept or reject calls while not in a call. 5.1 Application shall show the caller’s contact identifier or contact name during an incoming call.	TC026) When a call is received, it should display the following information ’You have a call from userid’	OK
		TC027) When a call is received, a popup should be displayed with the option to 'accept' or 'decline'.	OK
		TC028) Clicking the Accept button should start the call with the other party.	OK
		TC029) Clicking the Decline button should end the call.	OK
		TC030) If a third party calls you during a call, a pop-up will appear to let them know you are busy.	OK

Test Case -6

Req_ No.	Requirement	Test Case	Result
R4.1 R5 R5.1	4.1 During the call initiation, the user shall be presented with call status and outcome (answered, busy or rejected). 5. The system shall provide the ability to accept or reject calls while not in a call. 5.1 Application shall show the caller's contact identifier or contact name during an incoming call.	TC031) When the call is connected, the caller information screen should be shared.	OK
		TC032) When you make a call, the BYE button should be enabled.	OK
		TC033) If you press BYE to end the call, the other party should see a missed call noti pop-up.	OK
R4.2 R6 R7 R7.1	4.2 During call initiation the user shall have the ability to end the call at any time. 6. The system shall notify the user of missed calls, either because the call was not accepted or because the called entity was in another call. 7. Provide the ability to terminate a call at any time while in a call. 7.1 If a call is terminated by one user, the other caller shall be notified.	TC034) When you make a call, the BYE button should be enabled.	OK
		TC035) If you press BYE to end the call, the other party should see a missed call noti pop-up.	OK
		TC036) When the BYE button is pressed during a call, the other party should receive a noti that the call has ended.	OK
		TC037) When a call is received, you should be able to press the decline button to end the call.	OK

Test Case -7

Req_ No.	Requirement	Test Case	Result
R4.2 R6 R7 R7.1	4.2 During call initiation the user shall have the ability to end the call at any time. 6. The system shall notify the user of missed calls, either because the call was not accepted or because the called entity was in another call. 7. Provide the ability to terminate a call at any time while in a call. 7.1 If a call is terminated by one user, the other caller shall be notified.	TC038) The decline button must also be active during the call.	OK
		TC039) In either case, the call should end when the bye or decline button is pressed.	OK
R8	8. Application shall be brought to the foreground during an incoming call.	TC040) When a call is received, it should display the following information ’You have a call from userid’	OK